CLA 2016

Proceedings of the Thirteenth International Conference on Concept Lattices and Their Applications

CLA Conference Series

cla.inf.upol.cz

National Research University Higher School of Economics, Russia

ISBN 978-5-600-01454-1

National Research University Higher School of Economics, Russia

The Thirteenth International Conference on Concept Lattices and Their Applications

CLA 2016

Moscow, Russia
July 18–22, 2016

Edited by

Marianne Huchard
Sergei O. Kuznetsov

Technical Editor:
Dmitry I. Ignatov, dignatov@hse.ru

Cover design:
Daniil Bondarenko, daniil@bondarenko.su

Page count:	xiv+377
Impression:	200
Edition:	1st
First published:	2016

Published and printed by:
National Research University Higher School of Economics, Russia

Organization

CLA 2016 was organized by the National Research University Higher School of Economics.

Steering Committee

Radim Belohlavek	Palacký University, Olomouc, Czech Republic
Sadok Ben Yahia	Faculté des Sciences de Tunis, Tunisia
Jean Diatta	Iremia – Université de la Réunion, Saint-Denis, France
Peter Eklund	University of Wollongong, Wollongong, Australia
Sergei O. Kuznetsov	National Research University Higher School of Economics, Moscow, Russia
Engelbert Mephu Nguifo	LIMOS, University Blaise Pascal, Clermont-Ferrand, France
Amedeo Napoli	INRIA NGE/LORIA, Nancy, France
Manuel Ojeda-Aciego	Universidad de Málaga, Spain
Jan Outrata	Palacký University, Olomouc, Czech Republic

Program Chairs

Marianne Huchard	LIRMM, Université de Montpellier, France
Sergei O. Kuznetsov	National Research University Higher School of Economics, Russia

Program Committee

Cristina Alcalde	Universidad Del Pais Vasco, UPV/EHU, Spain
Simon Andrews	Sheffield Hallam University, United Kingdom
Gabriela Arevalo	Facultad de Ingenieria – Universidad Austral, Argentina
Mikhail A. Babin	Google Inc., USA
Jaume Baixeries	Ciències de la Computació, Catalonia
Radim Belohlavek	Palacký University, Olomouc, Czech Republic
Anne Berry	LIMOS, University Blaise Pascal, Clermont-Ferrand, France
Karell Bertet	Laboratory L3I, University of La Rochelle, France
François Brucker	Ecole Centrale Marseille, France
Ana Burusco	Universidad Publica De Navarra, Spain

Aleksey Buzmakov	Aleksey V. Buzmakov, National Research University Higher School of Economics, Perm, Russia
Peggy Cellier	IRISA/INSA Rennes, France
Pablo Cordero	Universidad de Málaga, Spain
Jean Diatta	Université de la Réunion, France
Stephan Doerfel	University of Kassel, Germany
Xavier Dolques	LHYGES, Université de Strasbourg/ENGEES, CNRS, Strasbourg, France
Florent Domenach	University of Nicosia, Cyprus
Vincent Duquenne	CNRS, Paris, France
Sébastien Ferré	Université de Rennes 1, France
Robert Godin	Université du Québec à Montréal, Canada
Alain Gély	LITA, Université Lorraine, Metz, France
Dmitry I. Ignatov	National Research University Higher School of Economics, Moscow, Russia
Mehdi Kaytoue	LIRIS/INSA de Lyon, France
Jan Konecny	Dept. Computer Science, Palacký University, Olomouc, Czech Republic
Stanislav Krajci	P. J. Safarik University, Slovak Republic
Francesco Kriegel	Technische Universität Dresden, Germany
Michal Krupka	Palacký University, Olomouc, Czech Republic
Marzena Kryszkiewicz	Warsaw University of Technology, Poland
Leonard Kwuida	Bern University of Applied Sciences, Switzerland
Florence Le Ber	ICube, Université de Strasbourg/ENGEES, CNRS, Strasbourg, France
Jesús Medina Moreno	University of Cádiz, Spain
Engelbert Mephu Nguifo	LIMOS, University Blaise Pascal, Clermont-Ferrand, France
Rokia Missaoui	Université du Québec en Outaouais, Canada
Amedeo Napoli	INRIA NGE/LORIA, Nancy, France
Sergei Obiedkov	National Research University Higher School of Economics, Moscow, Russia
Manuel Ojeda-Aciego	University of Malaga, Spain
Petr Osička	Palacký University in Olomouc, Czech Republic
Jan Outrata	Dept. Computer Science, Palacky University, Czech Republic
Jean-Marc Petit	Université de Lyon, INSA Lyon, France
Uta Priss	Ostfalia University of Applied Sciences, Wolfenbüttel, Germany
Sándor Radeleczki	University of Miskolc, Hungary
Olivier Raynaud	LIMOS, University Blaise Pascal, Clermont-Ferrand, France
Artem V. Revenko	Semantic Web Company, Vienna, Austria
François Rioult	GREYC CNRS UMR6072 – Université de Caen, France

Additional Reviewers

Organizing Committee

Proceedings Chair

Dmitry I. Ignatov — National Research University Higher School of Economics, Moscow

Sponsoring Institutions

National Research University Higher School of Economics, Moscow
ExactPro company
Russian Foundation for Basic Research

Table of Contents

Preface

Invited Contributions

Regular Papers

Preface

The 13th International Conference on "Concept Lattices and Applications (CLA 2016)" was held at National Research University Higher School of Economics, Moscow, Russia from July 18 until July 22, 2016. The CLA conference, organized since 2002, aims to provide to everyone interested in Formal Concept Analysis and more generally in Concept Lattices or Galois Lattices, an advanced view on some of the last research trends and applications in this field. It also aims to bring together students, professors, researchers and engineers, involved in all aspects of the study of concept lattices, from theory to implementations and practical applications. As the diversity of the selected papers shows, there is a wide range of research directions, around data and knowledge processing, including data mining, knowledge discovery, knowledge representation, reasoning, pattern recognition, together with logic, algebra and lattice theory.

The program of the conference includes four keynote talks given by the following distinguished researchers: Lev D. Beklemishev (Mathematical Institute of Russian Academy of Science, Moscow), Jérôme Euzenat (INRIA Grenoble Rhône-Alpes), Bernhard Ganter (TU-Dresden), Boris G. Mirkin (National Research University Higher School of Economics, Moscow). This volume includes the selected papers and the abstracts of the invited talks. This year, 46 papers were submitted, from which 28 papers were accepted as regular papers. We would like to thank here the contributing authors for their valuable work, the members of the program committee and the external reviewers who analyzed the papers with care. All of them participated to the continuing quality and importance of CLA, highlighting its key role in the field.

Then we would also like to thank the steering committee of CLA for giving us the occasion of leading this edition of CLA, the conference participants for their participation and support, and people in charge of the organization, especially Larisa I. Antropova, Ekaterina L. Chernyak, Dmitry I. Ignatov, Olga V. Maksimenkova, whose help was very precious in many occasions and that contributed to the success of the event. We would like to thank our sponsors, namely National Research University Higher School of Economics, ExactPro company, Russian Foundation for Basic Research. Finally, we also do not forget that the conference was managed (quite easily) with the Easychair system, for many tasks including paper submission, selection, and reviewing.

July 2016

Marianne Huchard
Sergei O. Kuznetsov

Program Chairs of CLA 2016

Strictly Positive Fragments of Modal and Description Logic

Lev D. Beklemishev

Mathematical Institute of Russian Academy of Science, Moscow, Russia

Abstract. In this talk we will advocate the use of weak systems of modal logic called strictly positive. These can be seen as fragments of polymodal logic consisting of implications of the form $A \to B$, where A and B are formulas built-up from T (truth) and the variables using just conjunction and the diamond modalities. The interest towards such fragments independently emerged around 2010 in two different areas: in description logic and in the area of proof-theoretic applications of modal logic.
From the point of view of description logic, strictly positive fragments correspond to the OWL 2 EL profile of the OWL web ontology language, for which various properties of ontologies can be decided in polynomial time. In the area of proof-theoretic applications, these fragments emerged under the name reflection calculi, as they proved to be a convenient tool to study the independent reflection principles in arithmetic and to calculate proof-theoretic ordinals of formal systems.
Thus, in two different areas strictly positive languages and logics proved to combine both efficiency and simplicity, and sufficient expressive power. In this talk we discuss general problems around weak systems of this kind and describe some of their applications.

Keywords: Modal Logic, Description Logic

Data Interlinking with Formal Concept Analysis and Link Keys

Jérôme Euzenat

INRIA & Univ. Grenoble Alpes, Grenoble, France

Abstract. Data interlinking, the problem of linking pairs of nodes in RDF graphs corresponding to the same resource, is an important task for linked open data. We introduced the notion of link keys as a way to identify such node pairs [1]. Link keys generalise in several ways keys in relational algebra. Thus, we consider how they could be extracted from data with Formal Concept Analysis. We show that an appropriate encoding makes the notion of candidate link keys correspond to formal concepts [2]. However candidate link keys are not yet link keys as they need to be selected through appropriate measures. We discuss how the measurement and concept extraction processes may be interleaved. If time permits we will also discuss extensions of this model to residual link keys and mutually dependent link keys[1].

Keywords: Linked Data, Formal Concept Analysis

References

1. Atencia, M., David, J., Euzenat, J.: Data interlinking through robust linkkey extraction. In: Proc. 21st European Conference on Artificial Intelligence (ECAI), Praha (CZ). (2014) 15–20
2. Atencia, M., David, J., Euzenat, J.: What can FCA do for database linkkey extraction? In: Proc. 3rd ECAI workshop on What can FCA do for Artificial Intelligence? (FCA4AI), Praha (CZ). (2014) 85–92

[1] This work has been developed in collaboration with Manuel Atencia, Jérôme David and Amedeo Napoli.

Things to Do in Formal Concept Analysis

Bernhard Ganter

TU–Dresden, Germany

Abstract. After one third of a century, Formal Concept Analysis still exists. Some of the younger researchers in the field were not even born when Rudolf Wille's seminal paper "Restructuring lattice theory: an approach based on hierarchies of concepts" was published in 1982. In our talk we shall discuss the present status of the field, its strengths and weaknesses, and sketch some possibilities for its future. We discuss several potential projects which we find promising, some in the theoretical foundations, some for applications, and others concerning the necessary philosophical foundations. Our view will be very subjective and hopefully controversial. We hope to induce a discussion which makes the research potential of the field apparent, and which can inspire new research activities.

Keywords: Formal Concept Analysis

References

1. Ganter, B., Obiedkov, S.A.: Conceptual Exploration. Springer (2016)
2. Ganter, B., Wille, R.: Formal Concept Analysis – Mathematical Foundations. Springer (1999)

Approximate Clusters, Biclusters and n-Clusters in the Analysis of Binary and General Data Matrices

Boris G. Mirkin

National Research University Higher School of Economics, Moscow, Russia

Abstract. Approximate cluster structures are those of formal concepts and n-concepts with added numerical intensity weights. The talk presents theoretical results and computational methods for approximate clustering and n-clustering as extensions of the algebraic-geometrical properties of numerical matrices (SVD and the like) to the situations where one or most of elements of the solutions to be found are expressed by binary vectors. The theory embraces such methods as k-means, consensus clustering, network clustering, biclusters and triclusters and provides natural data analysis criteria, effective algorithms and interpretation tools.

Keywords: Approximate clusters, biclusters, n-clusters, Formal Concept Analysis

References

1. Mirkin, B.G., Rostovtsev, P.S.: Method for revealing associated feature subsets,. In Mirkin, B., ed.: Models for Summarization of SocioEconomic Data (Metody Agregirovania Sotsial'no-Economitcheskoi Informatsii), Novosibirsk: Institute of Economics Press (1978) 107–112 (in Russian).
2. Ignatov, D.I., Gnatyshak, D.V., Kuznetsov, S.O., Mirkin, B.G.: Triadic formal concept analysis and triclustering: searching for optimal patterns. Machine Learning **101**(1-3) (2015) 271–302

LatViz: A New Practical Tool for Performing Interactive Exploration over Concept Lattices

Mehwish Alam[1], Thi Nhu Nguyen Le[2], and Amedeo Napoli[2]

1. Laboratoire d'Informatique de Paris-Nord, Université Paris 13, Paris, France
2. LORIA (CNRS – Inria Nancy Grand Est – Université de Lorraine)
BP 239, Vandoeuvre-lès-Nancy, F-54506, France
{alam@lipn.univ-paris13.fr}{thi-nhu-nguyen.le,amedeo.napoli@loria.fr}

Abstract. With the increase in Web of Data (WOD) many new challenges regarding exploration, interaction, analysis and discovery have surfaced. One of the basic building blocks of data analysis is classification. Many studies have been conducted concerning Formal Concept Analysis (FCA) and its variants over WOD. But one fundamental question is, after these concept lattices are obtained on top of WOD, how the user can interactively explore and analyze this data through concept lattices. To achieve this goal, we introduce a new tool called as `LatViz`, which allows the construction of concept lattices and their navigation. `LatViz` proposes some remarkable improvements over existing tools and introduces various new functionalities such as interaction with expert, visualization of Pattern Structures, AOC posets, concept annotations, filtering concept lattice based on several criteria and finally, an intuitive visualization of implications. This way the user can effectively perform an interactive exploration over a concept lattice which is a basis for a strong user interaction with WOD for data analysis.

Keywords: Lattice Visualization, Interactive Exploration, Web of Data, Formal Concept Analysis.

1 Introduction

In the last decade, there has been a huge shift from the Web of Documents to the Web of Data (WOD). Web of Documents represents data in the form of HTML pages which linked with other HTML pages through hyperlinks. This web of documents has evolved into WOD where all the information contained is represented in the form of entity and relations allowing the semantics to be embedded in the representation of the this data. This data are in the form of a (node-arc) labeled graph belonging to several domains such as newspapers, publications, biomedical data etc. The growth in the publication of data sources in WOD has made it an important source of data, which has led towards many challenges pertaining to effective utilization of this data. WOD mainly represents data in the form of Resource Description Framework (RDF)[1]. There are several ways such as data

[1] `http://www.w3.org/RDF/`

ISBN 978-5-600-01454-1, National Research University Higher School of Economics, 2016.

dumps and SPARQL queries to access this data, which can be utilized for many purposes, one of which is visualization and interactive exploration for data analysis purposes. Several visualization tools have been developed for this purpose, one of which is LODLive[2] [1], where user can choose data-sets such as DBpedia and Freebase and specify an entity as a starting point for browsing the node-arc labeled graph. Another tool based on graphical display is RelFinder [2], where given several entities the tool automatically finds the paths connecting these entities. The major drawback of LODLive is that after two hops the number of nodes increase and it is hard to visualize the data. Moreover, these tools are good for getting an insight into what RDF graph contains but they are not built for the purpose of knowledge discovery.

In order to provide the user with the ability to perform data analysis and knowledge discovery over such kind of data, there is a need to perform classification, where the obtained classes are further made available to the user for exploration and subjective interpretation. In the current study we use Formal Concept Analysis as the basis for classification. Several studies have already been conducted using FCA and its variants over RDF graphs or its generalization to knowledge graphs. Out of these studies so far RV-Xplorer [3] is the only tool that actually allows interactive exploration of clustered RDF data [4]. The purpose of this paper is to enhance the functionalities discussed in the previous two studies. In this study we introduce a new tool `LatViz` which increases the interpretability of a concept lattice by remarkably improving the user interaction with the concept lattice as compared to existing tools. Various new functionalities have been introduced such as the visualization of Pattern Structures and AOC-posets, concept annotation, filtering concept lattice and pattern concept lattice based on several criteria and finally, an intuitive visualization of implications. This way the user can effectively perform an interactive exploration over a concept lattice which in turn gives a basis for a strong user interaction with WOD for knowledge discovery purposes. In this paper, we detail the important interaction operations implemented in `LatViz`. In the rest of this paper we refer to "user" as an "expert" as (s)he needs to have some basic knowledge about the lattice structure.

The paper is structured as follows: Section 2 introduces a motivating example, Section 3 introduces the background required for understanding the rest of the paper while Section 4 introduces some of the important functionalities of `LatViz`. Then in Section 5, we discuss some of the related tools already developed and finally Section 6 details the future perspectives of the current work.

2 Motivating Example

Let us consider that an expert is searching for papers published by a particular team in conferences or journals related to his/her field of research. In order to locate the papers of his/her interest (s)he has to search for specific keywords

[2] `http://en.lodlive.it/`

or authors in the local portal. For getting the view of which kind of papers are contained (s)he has to run a broad query and then narrow down his/her query to obtain papers on specific keywords or authors or group of keywords or authors. The expert will end up running several queries to get what (s)he wants. Moreover, if the expert wants to know the collaborations of the team with other members of the research community outside the team, as well as the diversity and the specialization of the team members, this cannot be directly obtained by simple querying. To obtain such kind of knowledge there is a need to introduce a support for data analysis. Based on this scenario, we show how the expert can be guided thanks to an adapted visualization tool to obtain such information of interest with the help of concept lattices.

3 Preliminaries

3.1 Pattern Structures

In this section we provide a brief introduction to pattern structures [5]. A pattern structure is a triple $(G, (D, \bigsqcap), \delta)$, where G is the set of objects, $(D, \bigsqcap)$ is a meet-semilattice of descriptions D equipped with a similarity measure $\bigsqcap$, and $\delta : G \rightarrow D$ maps an object to its description. More intuitively, a pattern structure is a set of objects with their corresponding descriptions, where similarity between descriptions is computed thanks to $\bigsqcap$. This similarity operator $\bigsqcap$ is idempotent, commutative and associative. The derivation operators can be defined as:

$$A^{\square} := \bigsqcap_{g \in A} \delta(g) \qquad for\ A \subseteq G$$
$$d^{\square} := \{g \in G | d \sqsubseteq \delta(g)\} \qquad for\ d \in D$$

Each element in D is referred to as a *pattern*. The subsumption order over these patterns is given as: $c \sqsubseteq d \Leftrightarrow c \sqcap d = c$. The operator $(.)^{\square}$ makes a Galois connection. Then, a *pattern concept* of a pattern structure $(G, (D, \bigsqcap), \delta)$ is a pair (A, d) where $A \subseteq G$ and $d \in D$ such that $A^{\square} = d$ and $A = d^{\square}$, where A is called the concept extent and d is called the concept intent. Pattern concept lattices are defined in the same way as concept lattices in standard FCA.

Interval Pattern Structures. Interval Pattern Structures were first introduced in [6] for dealing with numerical data instead of binary data. 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: $\delta(g_1) \sqcap \delta(g_2) = \langle [min(l_i^1, l_i^2), max(r_i^1, r_i^2)] \rangle$ Following the definition of a pattern concept discussed previously a interval pattern concept lattice can be built. Pattern structures have also been introduced to deal with graphical data [5].

3.2 Web of Data and its Classification

Web of Data (WOD) is represented in the form of entity and relationships. A standard representation of WOD represents data in the form of RDF (Resource Description Framework) triples written as $\langle subject, predicate, object\rangle$. Here, *subject* can be a URI or a blank node, *predicate* can be a URI and *object* can be a URI, a blank node or a literal. Several RDF triples connect together to form an RDF graph. Table 1 shows an example of RDF triple store from DBLP where each row represents one triple. The subject is the title of the paper, predicates are the relations such as dc:subject and dc:creator (interpreted as "has keyword" and " has author" respectively) and the objects are the keywords and the authors. The triple t_1 is read as "*paper* s_1 *has keyword Pattern Structures*".

tid	Subject	Predicate	Object
t_1	s_1	`dc:subject`	Pattern Structures
t_2	s_1	`dc:creator`	$author_1$
t_3	s_2	`dc:subject`	Formal Concept Analysis
t_4	s_2	`dc:creator`	$author_2$
t_5	s_1	`rdf:type`	Publication
t_6	o_{21}	`rdf:type`	Author
⋮	⋮	⋮	⋮

Table 1: RDF triples about papers with their authors and keywords from DBLP.

In order to allow interactive data exploration over RDF data, an initial set of restrictions is posed by the expert by defining the task requirement based on which a SPARQL query is created by the expert to obtain the specific data. Then the most important step for interactive data exploration is to perform classification of RDF data. Finally, the expert is allowed to interact with the obtained classification. In the rest of this paper, we further improve the functionalities of the existing tools by introducing several new interactive operations in a new tool called `LatViz`.

4 `LatViz` for Interactive Exploration of Concept Lattices

4.1 User Interface

The display of `LatViz` resembles Conexp[3], which provides basic functionalities for building a concept lattice. `LatViz` implements two algorithms for building a concept lattice from a binary context, one of which is introduced in [7]. Another, efficient algorithm for building a concept lattice is AddIntent [8]. Demo of LatViz is available on-line through this link: `http://latviz.loria.fr/latviz/`.

The concept lattice for the scenario in section 2 was created by mapping the RDF data to a formal context $\mathcal{K} = (G, M, I)$. Based on Table 1, the subjects of the triples were considered as the set of objects G, the objects in the RDF triples i.e., keywords and authors were considered as the set of attributes M. In this example, the RDF triples were created from the publications of the

[3] `http://conexp.sourceforge.net/`

Knowledge Discovery (KDD) team in LORIA. The number of objects in the context are 343 and attributes are 1516. Figure 1 shows a complete concept lattice built using `LatViz`. The information about a concept can be displayed by selecting the concept. Very often huge concept lattices are obtained based on the context size. `LatViz` provides several interactive operations allowing for reduction of exploration space of the expert. To-date this is the most interactive tool having many unique functionalities such as handling numeric data with the help of interval pattern structures, AOC-posets, filtering concept lattice and implications which provides support for data analysis. Other functionalities such as annotating the lattice, level-wise display of a concept lattice etc. are discussed in many contexts but are not yet directly implemented in the commonly used tools. In the following we detail each of these functionalities for data analysis.

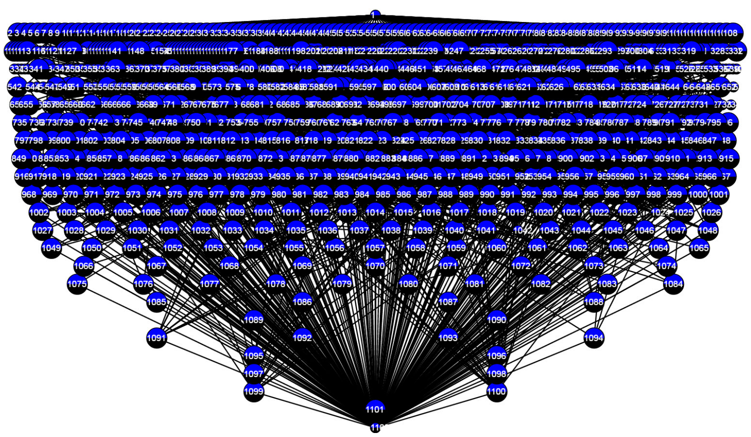


Fig. 1: Complete lattice built from the papers of KDD-Team in LORIA Nancy.

4.2 AOC-Posets

AOC-poset is a partially ordered set of the attribute and object concepts, first introduced in [9, 10]. If (G, M, I) is a formal context then according to the definition in [7], an *object concept* is defined as (g'', g') such that $g \in G$, i.e. (g'', g') is the "lower" concept whose extent includes g. Dually, an attribute concept is defined as (m', m'') where $m \in M$, i.e. (m', m'') is the "highest" concept whose intent includes m. The object and attribute concept are referred to as *introducers* in [11]. Once an attribute is introduced in a concept it is inherited from top to bottom while, dually, an introduced object is "inherited" from bottom to top. During this study, we implement the Hermes Algorithm introduced in [11] for building AOC-Poset from binary context. AOC-posets have been successfully applied to several domain one of which is to classify linguistic data [9]. In the current study we compute AOC Posets of RDF data. Figure 2 shows the AOC Posets of the concept lattice in Figure 1, where object and attribute concepts are shown in green while the other concepts are translucent and the pink color shows the selected concept.

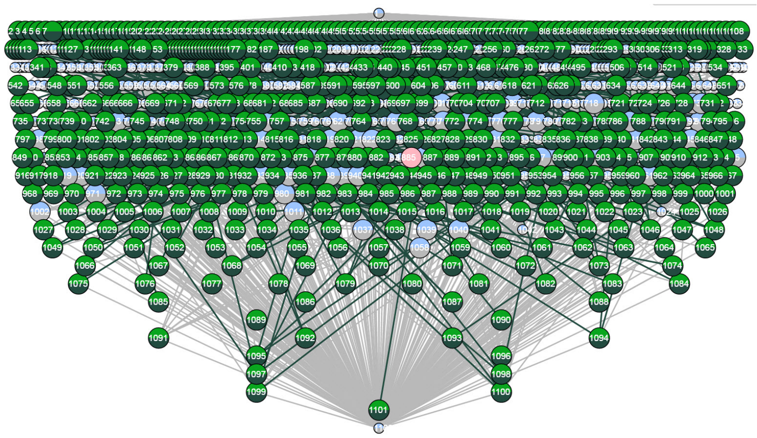


Fig. 2: AOC-Posets.

4.3 Displaying Concept Lattice Level-wise

AOC-Posets actually reduce exploration space but still a huge number of concepts remain to be observed. `LatViz` allows the creation of concept lattice level-wise by interaction. When an expert clicks on the top concept, `LatViz` computes and displays the first level. After that the expert can select the concept for continuing the exploration, then `LatViz` computes the next level for that concept. In Figure 3, the top image shows the first level of the concept lattice built by selecting the top concept. Then the expert can view the contents of each concept on mouseover. In the running example, expert locates the concept with all the papers of Amedeo Napoli (i.e., K#2), which shows that the total number of documents written by Amedeo Napoli are 152. On selecting this concept, the next level of the lattice originating from the selected concept is computed (shown in the bottom image in Figure 3).

4.4 Display Sub/Super Concepts of a Concept

In case of huge concept lattices sometimes it is hard to keep track of the ordering relations between the concepts. `LatViz` allows the expert to only highlight sub-/super concepts of a concept. For example, if the expert wants to display all the publications along with the collaborations of the author Amedeo Napoli, (s)he can highlight the associated sub-lattice of the attribute concept of "Amedeo Napoli". Figure 4 shows the highlighted sub-lattice in brown. An expert can highlight the super-concepts connected to a concept. If the expert is looking for all the papers having some keywords common with the paper *Knowledge Organization and Information Retrieval Using Galois Lattices* having one or more of the keywords in the intent of the concept then (s)he can highlight the sub-lattice of super concepts associated to it (see Figure 5).

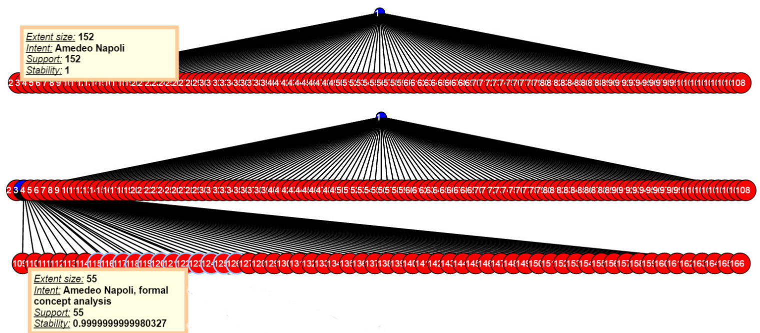


Fig. 3: Top image shows the first level of the concept lattice, the bottom image shows the second level built by interaction.

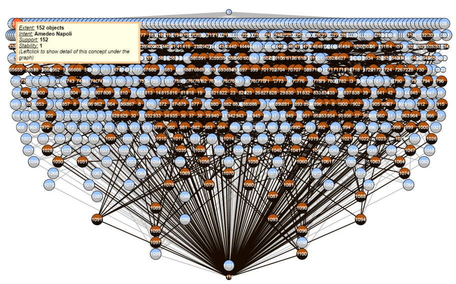


Fig. 4: The sub-lattice highlighted for the author "Amedeo Napoli".

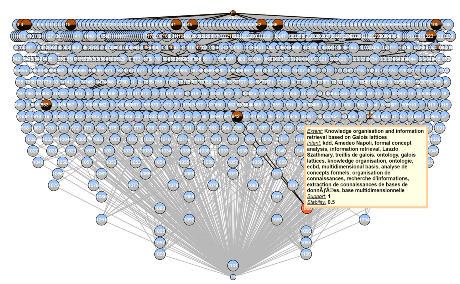


Fig. 5: Highlighting the super lattice of a concept.

4.5 Display/Hide the Sub-lattice

This functionality was partially implemented in RV-Xplorer [3] to reduce the interaction space of the expert. Here the expert can only show the part of the concept lattice in which (s)he is interested. The expert can locate the interesting concept by navigation, containing some intent or extent. If an intent is interesting and the expert marks the concept as interesting then only the sub-concepts of this concept are shown to the expert as the intents are inherited from top to bottom. Dually, if an extent is interesting for the expert then all the super concepts are shown to the expert as the extent is inherited bottom-top. Previously, the expert highlighted the sub-lattice of the concept containing all the papers of Amedeo Napoli, now if the expert is interested in only the papers of Amedeo Napoli on Knowledge Representation then (s)he can navigate downwards and only see this part of concept lattice by marking it interesting (see Figure 6). Similarly, we previously highlighted all the super concepts of the concept having the paper entitled *Knowledge Organization and Information Retrieval Using Galois Lattices* in the extent, Figure 7 only shows the associated sub-lattice to have a clearer view (see Figure 7).

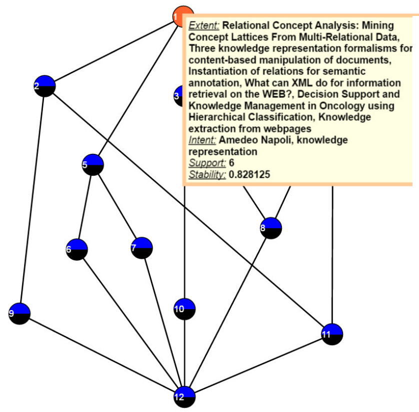


Fig. 6: Showing only sub-lattice of the interesting concept.

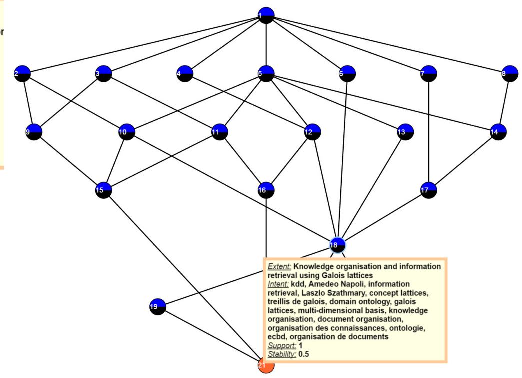


Fig. 7: Showing only super-lattice of the interesting concept.

4.6 Interval Pattern Structures

In the running scenario, we extracted three attributes for the papers i.e., year of publications, rank of the conference in which the paper was published and finally the number of pages. The ranks of the conferences were considered based on COmputing Research and Education (CORE) rankings[4]. The ranks were A*, A, B, C and other which were coded as 1, 2, 3, 4 and 5 respectively. The final concept lattice generated for the last five years of publications of Knowledge Discovery Team is shown in Figure 8.

4.7 Lattice Filtering Criteria

There are two categories of filtering provided by `LatViz`; one is for the concept lattice created with the binary data and the other one is provided for the pattern concept lattice built with the help of interval pattern structures.

Filtering Concept Lattice. After a concept lattice is built by applying FCA, expert is allowed to set several filtering criteria such as stability, lift, extent size, intent size and finally specific object or attribute names. Let us consider that in the running example, the expert is looking for the papers published by Amedeo Napoli on the topic of pattern structures and FCA. A filter on the number of attributes in the intent is set to 3. The filtered concept lattice obtained over the complete lattice in Figure 1 is shown in Figure 9. It further shows the authors with who Amedeo Napoli has worked i.e., Sergei O. Kuznetsov and Mehdi Kaytoue. This part of concept lattice shows the community of authors working with Amedeo Napoli on the topic of pattern structures.

[4] http://portal.core.edu.au/conf-ranks/

Filtering Pattern Concept Lattice. Interval Pattern Concept Lattices can also be filtered by specifying the number of attributes to be considered, the upper and the lower limits for the intervals in the intent of each attribute along with stability, lift and extent size. Let us consider the pattern concept lattice in Figure 8, it can be seen that the concept lattice is hard to interpret. To make it more readable based on what an expert wants, (s)he is allowed to specify filters. For example, if the expert is looking for a paper published in a conference of a rank 1-4 in the year 2012 - 2015 and has the number of pages not less than 2 and no more than 42 then the respective filters can be set for the values of all three attributes. The filtered pattern concept lattice will then only contain the part of lattice needed by the user. Figure 8 shows the concept containing group of papers published from 2014-2015 in conferences with rank 2 having number of pages 2-42.

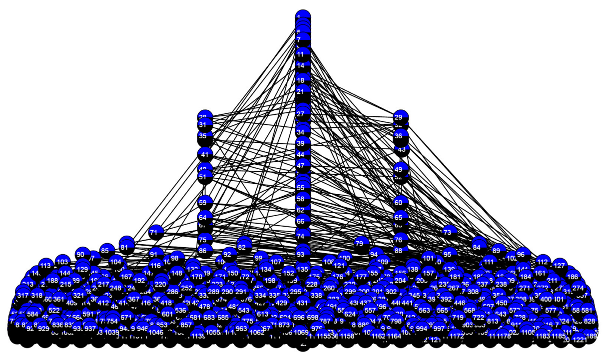


Fig. 8: Interval pattern concept lattice for publications.

4.8 Attribute Implications

One of the many proposed visualization techniques for implications includes *table-based views*. It keeps each column for rule ID, LHS and RHS of the rule, support and confidence measures. These views were used because of the simplicity of storage. However, while expert interaction it is not very convenient to obtain interesting rules at a simple glance as the number of rules can be too many. Another way of visualizing association rules are *Matrix Views*, where rows represent the LHS and columns represent the RHS of the rules. Support and confidence are displayed by different colors in the intersection of the LHS and RHS. In case of a formal context, the number of objects/attributes can be very big leading to problems in displaying the matrix. By carefully taking into account the above drawbacks, we finally settle on visualizing implications with the help of scatter plots, where the x-axis shows the increasing support and the y-axis shows the increasing lift (as we are considering implications the confidence of the rule is always 100%). Such kind of display helps the expert to single-out the rules (s)he wants to visualize based on the values of support and lift. Figure 11 shows implications of the running example, x-axis keeps the support in percentage and y-axis keeps lift. The number on top of the circle shows the number of rules existing in the same point in the plot. On mouse over, expert can view the implications.

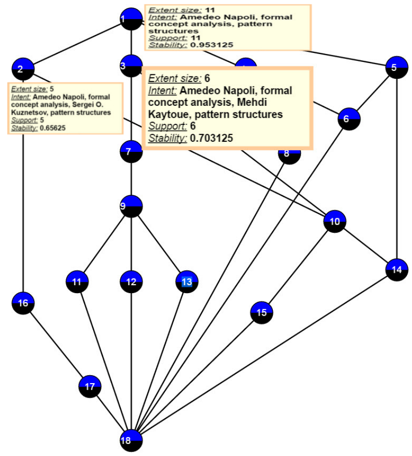


Fig. 9: Filtered concept lattice obtained from binary context.

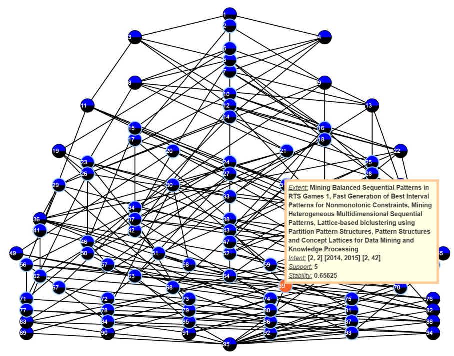


Fig. 10: Filtered Pattern Concept Lattice.

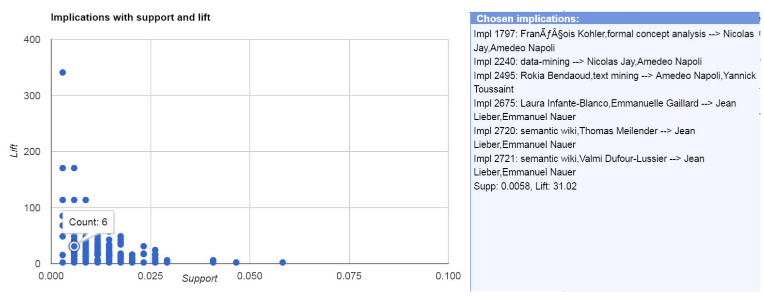


Fig. 11: Attribute implications for the running example.

5 Related Tools

In [4], the authors focus mainly on interactive data exploration over RDF data for interactive knowledge discovery. It clusters RDF triples based on RDF Schema and then allows interactive exploration with the help of RV-Xplorer (Rdf View eXplorer) [3]. It is a tool for visualizing views over RDF graphs mainly for identifying interesting parts of data and allow data analysis. It has also been extended for clustering SPARQL query answers. To-date there have been many other tools developed for reducing the effort of expert in observing and interpreting a concept lattice. Many of the tools have been developed for more specific purposes. CREDO [12] and FooCA [13] are the Web Clustering Engines [14] which take the answers from queries posed against search engines and create a concept lattice which is then displayed to the expert for interaction. CREDO allows only limited interaction, however, FooCA allows the expert to edit the context and iteratively build the concept lattice. CEM [15] is an email manager which allows quick search through the e-mails and usually deals with smaller concept lattices. Camelis [16] is a system based on FCA for the organization of documents allowing several navigation operations. Another set of tools such as Sewelis [17] and Sparklis [18] allows navigation/interaction over knowledge graphs. Many other tools such as Galicia[5], ConExp and ToscanaJ[6] are developed for academic purposes. `LatViz` takes the basic functionalities of ConExp and takes it to the another level by providing visualization for many algorithms introduced over time to increase the readability. Moreover, it re-uses the source-code for building concept lattice with the help of the algorithm in [7] from ToscanaJ [19]. It can not only be applied to WOD but it has been extended for interpreting any kind of data.

6 Discussion and Future Improvements

`LatViz` is a tool built for allowing expert interaction for data analysis purposes. It provides many new functionalities for reducing the exploration space of the expert and enable him to interpret the results. As a future perspective, we also want to implement other variations of pattern structures such as Pattern Structures introduced for structured set of attributes discussed in [20] and Heterogeneous Pattern Structures [21]. We also want to extend the implementation of implications to association rules. Finally, we also want to take into account matrix factorization.

References

1. Camarda, D.V., Mazzini, S., Antonuccio, A.: Lodlive, exploring the web of data. In: I-SEMANTICS 2012 - 8th International Conference on Semantic Systems, I-SEMANTICS '12, Graz, Austria, September 5-7, 2012. (2012)
2. Heim, P., Lohmann, S., Stegemann, T.: Interactive relationship discovery via the semantic web. In: The Semantic Web: Research and Applications, 7th Extended Semantic Web Conference, ESWC 2010, Heraklion, Crete, Greece, May 30 - June 3, 2010, Proceedings, Part I. (2010) 303–317
3. Alam, M., Osmuk, M., Napoli, A.: RV-Xplorer: A way to navigate lattice-based views over RDF graphs. In: Proceedings of the Twelfth International Conference on Concept Lattices and Their Applications. (2015) 23–34
4. Alam, M., Napoli, A.: Interactive exploration over RDF data using formal concept analysis. In: 2015 IEEE International Conference on Data Science and Advanced Analytics, DSAA. (2015) 1–10
5. Ganter, B., Kuznetsov, S.O.: Pattern structures and their projections. In: ICCS. (2001) 129–142

[5] `https://sourceforge.net/projects/galicia/`
[6] `http://toscanaj.sourceforge.net/`

6. Kaytoue, M., Kuznetsov, S.O., Napoli, A.: Revisiting numerical pattern mining with formal concept analysis. In: Proceedings of the 22nd International Joint Conference on Artificial Intelligence. (2011) 1342–1347
7. Ganter, B., Wille, R.: Formal Concept Analysis: Mathematical Foundations. Springer, Berlin/Heidelberg (1999)
8. van der Merwe, D., Obiedkov, S.A., Kourie, D.G.: Addintent: A new incremental algorithm for constructing concept lattices. In: ICFCA. (2004) 372–385
9. Osswald, R., Petersen, W.: A logical approach to data-driven classification. In: KI. Volume 2821 of Lecture Notes in Computer Science., Springer (2003)
10. Petersen, W.: A set-theoretical approach for the induction of inheritance hierarchies. Electr. Notes Theor. Comput. Sci. **53** (2001) 296–308
11. Berry, A., Gutierrez, A., Huchard, M., Napoli, A., Sigayret, A.: Hermes: a simple and efficient algorithm for building the AOC-poset of a binary relation. Ann. Math. Artif. Intell. **72**(1-2) (2014) 45–71
12. Carpineto, C., Romano, G.: Exploiting the potential of concept lattices for information retrieval with CREDO. J. UCS **10**(8) (2004) 985–1013
13. 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, 6th Industrial Conference on Data Mining, ICDM Proceedings. (2006)
14. Carpineto, C., Osiński, S., Romano, G., Weiss, D.: A survey of web clustering engines. ACM Comput. Surv. **41**(3) (2009) 17:1–17:38
15. Cole, R., Stumme, G.: CEM - A conceptual email manager. In: 8th International Conference on Conceptual Structures, ICCS, Proceedings. (2000)
16. Ferré, S.: Camelis: a logical information system to organise and browse a collection of documents. Int. J. General Systems **38**(4) (2009) 379–403
17. Ferré, S., Hermann, A.: Reconciling faceted search and query languages for the semantic web. IJMSO **7**(1) (2012) 37–54
18. Ferré, S.: SPARKLIS: a SPARQL endpoint explorer for expressive question answering. In: Proceedings of the Posters & Demonstrations Track a track within the 13th International Semantic Web Conference, ISWC. (2014)
19. Becker, P., Correia, J.H.: The ToscanaJ suite for implementing conceptual information systems. In: Formal Concept Analysis, Foundations and Applications. (2005) 324–348
20. Alam, M., Buzmakov, A., Napoli, A., Sailanbayev, A.: Revisiting pattern structures for structured attribute sets. In: Proceedings of the Twelfth International Conference on Concept Lattices and Their Applications. (2015) 241–252
21. Codocedo, V., Napoli, A.: A proposition for combining pattern structures and relational concept analysis. In: Formal Concept Analysis - 12th International Conference, ICFCA 2014, Proceedings. (2014) 96–111

Comparing Algorithms for Computing Lower Covers of Implication-closed Sets

Alexandre Bazin

LIMOS, CNRS, Blaise Pascal University, Clermont-Ferrand, France
contact@alexandrebazin.com

Abstract. In this paper, we consider two methods for computing lower cover of elements in closure systems for which we know an implicational basis: intersecting meet-irreducible elements or computing minimal transversals of sets of minimal generators. We provide experimental results on the runtimes for single computations of lower covers and depth-first searches.

1 Introduction

Closed sets are essential objects in many fields such as data mining or database theory. Most of the time, one is interested in computing all or parts of the closure system for a given closure operator [6, 8]. However, we are sometimes faced with the problem of finding a specific closed set that respects some property and computing too much of the rest of the closure system is a waste of time. An example of this is the problem of finding a maximal frequent closed set. As frequency is an anti-monotone property, it suffices to start with the least closed set and perform a depth-first search by repeatedly computing upper covers of sets until we cannot find frequent sets anymore. Given a closure operator and a closed set, it is easy to compute the closed sets immediately above it. Problems start to arise when we are looking for specific closed sets respecting a property that is not anti-monotone. For example, when computing the Duquenne-Guigues basis $\mathcal{B}$ for some closure operator c, a pseudo-closed set P is $\mathcal{I}$-closed but not c-closed for $\mathcal{I} = \mathcal{B} \setminus \{P \rightarrow c(P)\}$. The non-anti-monotonicity of the c-closedness prevents us from using depth-first searches from the bottom.

What we are interested in is the problem of performing depth-first searches starting from the maximal element in closure systems for which we know an implicational basis. This amounts to computing lower covers of closed sets using the implications, a problem shown to be NP-hard [1]. In this work, we compare two algorithms - one based on meet-irreducibles, the other on minimal generators - that solve this problem. Section 2 reviews basic definitions and properties of lower covers that the algorithms use. Sections 3 and 4 respectively describe the methods using meet-irreducible elements and minimal generators. In Section 5, we present experimental results on the runtimes for both algorithms.

© Marianne Huchard, Sergei O. Kuznetsov (Eds.): CLA 2016, pp. 21–32,
ISBN 978-5-600-01454-1, National Research University Higher School of Economics, 2016.

2 Preliminaries

2.1 Basic Definitions

Let us use E to denote a set of elements.

Definition 1 *A closure operator* $c : 2^E \mapsto 2^E$ *is an extensive* $(X \subseteq c(X))$, *increasing* $(X \subseteq Y \Rightarrow c(X) \subseteq c(Y))$ *and idempotent* $(c(c(X)) = c(X))$ *function.*

A set $S \in 2^E$ such that $S = c(S)$ is said to be *closed*. The intersection of two closed sets is closed. The set of all sets closed for a given closure operator c ordered by inclusion forms a lattice that will here be denoted by Φ_c.

Definition 2 *An implication on* E *is a pair* $(A, B) \in 2^E \times 2^E$, *most commonly written* $A \rightarrow B$.

Definition 3 *Let* $\mathcal{I}$ *be a set of implications. We denote by* $\mathcal{I}(\cdot)$ *the closure operator, sometimes called logical closure, that maps a set* X *to its smallest superset* Y *such that*

$$\forall A \rightarrow B \in \mathcal{I}, A \subseteq Y \Rightarrow B \subseteq Y$$

The logical closure is a closure operator. As such, for an implication set $\mathcal{I}$, we will use $\Phi_{\mathcal{I}}$ to denote the lattice of implication-closed sets ordered by inclusion.

Definition 4 *A minimal generator* G *of* $X \in 2^E$ *for a closure operator* c *is an inclusion-minimal set such that* $c(G) = X$.

In the remainder of this paper, we will use the term *minimal generator*, without any more details, to talk about the minimal generators for the logical closure. The set of minimal generators of a set S for an implication set $\mathcal{I}$ will be denoted $Gen_{\mathcal{I}}(S)$.

Definition 5 *Let* $\mathcal{L} = (E, \leq)$ *be a lattice. A meet-irreducible element is an element* $e \subseteq E$ *that has a single upper cover, i.e.* $\{x \in E \mid x > e\}$ *has a single inclusion-minimal element. We use* $\mathcal{M}(\mathcal{L})$ *to denote the set of meet-irreducible elements of the lattice* $\mathcal{L}$.

Any element of the lattice $\mathcal{L}$ is the infimum of a set of meet-irreducible elements.

Definition 6 *Let* $\mathcal{H} = (\mathcal{E}, V)$ *be a hypergraph. A minimal transversal of* $\mathcal{E}$ *is a set* $S \subseteq V$ *such that* $\forall X \in \mathcal{E}$, $X \cap S \neq \emptyset$.

We use $Tr(\mathcal{E})$ to denote the set of minimal transversals of a set of hyperedges $\mathcal{E}$.

2.2 Lower Covers

If we want to perform a depth-first search from the top in a lattice $\Phi_{\mathcal{I}}$ for which we know $\mathcal{I}$, we have to be able to compute the lower covers of a set $S \in \Phi_{\mathcal{I}}$. We present here two methods that can be used to compute them.

Proposition 1 *For any $S \in \Phi_{\mathcal{I}}$, the lower covers of S are the inclusion-maximal elements of $\{S \cap M \mid M \in \mathcal{M}(\Phi_{\mathcal{I}})$ and $S \not\subseteq M\}$.*

Proof Every element of $\Phi_{\mathcal{I}}$ is the infimum of a subset of $\mathcal{M}(\Phi_{\mathcal{I}})$. Additionally, $\Phi_{\mathcal{I}}$ being a closure system, the infimum of two sets is their intersection. As such, for any $M \in \mathcal{M}(\Phi_{\mathcal{I}})$, $S \cap M \subseteq S$ is an $\mathcal{I}$-closed set. $S \cap M$ being strictly contained in S, if it is inclusion-maximal then it is a lower cover of S. □

Using Proposition 1 to compute lower covers requires that we first know the meet-irreducible elements of the lattice. In most case, they must be explicitly computed beforehand. Algorithm 1, presented in Section 3.1, does this.

Proposition 2 *For any $S \in \Phi_{\mathcal{I}}$, the lower covers of S are the sets $S \setminus T$ with $T \in Tr(Gen_{\mathcal{I}}(S))$.*

Proof Let T be a minimal transversal of $Gen_{\mathcal{I}}(S)$. For any $e \in S \setminus T$, $(S \setminus T) \cup \{e\} = S \setminus (T \setminus \{e\})$ contains a minimal generator of S because of the minimality of T. Therefore, there is no closed set between S and $S \setminus T$ and $S \setminus T$ is a lower cover of S. □

Using Proposition 2 to compute the lower covers of S would require that we compute the minimal generators S first, then their minimal transversals. The algorithms we use for these problems are described in Section 4.

3 Computing with Meet-irreducible Elements

3.1 Computing Meet-irreducible Elements from an Implication Base

We use Wild's algorithm [9] to compute the set of meet-irreducible elements of the lattice $\Phi_{\mathcal{I}}$ for which we know an implication base $\mathcal{I}$. It is based on the fact that a set $S \in \Phi_{\mathcal{I}}$ is meet-irreducible if and only if it is maximal among closed sets that do not contain some element e. Let us call $max(e, \mathcal{I})$ the set of maximal elements of $\Phi_{\mathcal{I}}$ that do not contain e and $max'(e, \mathcal{I} \setminus \{I\})$ and $max''(e, \mathcal{I} \setminus \{I\})$ the subsets of $max(e, \mathcal{I} \setminus \{I\})$ for which $I = A \rightarrow B$ respectively holds and does not hold. We then have

$$max(e, \mathcal{I}) \subseteq max'(e, \mathcal{I} \setminus \{I\}) \bigcup_{a \in A} max''(e, \mathcal{I} \setminus \{I\}) * max(a, \mathcal{I} \setminus \{I\})$$

where $X * Y = \{x \cap y \mid x \in X, y \in Y\}$.

Algorithm 1 uses this property to compute the meet-irreducible elements of $\Phi_{\mathcal{I}}$ from $\mathcal{I} = \{A_1 \rightarrow B_1, ..., A_n \rightarrow B_n\}$ by computing the meet irreducibles of every lattice $\Phi_{\mathcal{I}_i}$ such that $\mathcal{I}_i = \{A_1 \rightarrow B_1, ..., A_i \rightarrow B_i\}$ with $0 \leq i \leq n$.

Algorithm 1: Meet-irreducibles

```
Meet-Irreducibles (E, I)
Input  : Implication set I = {A_1 → B_1, ..., A_n → B_n} on E
Output: M(Φ_I)
foreach e ∈ E do
    max(e) = {E \ {e}}
end
foreach i ∈ 1..n do
    foreach e ∈ E do
        max'(e) = {Z ∈ max(e) | A_i ⊈ Z or B_i ⊆ Z}
        max''(e) = max(e) \ max'(e)
        foreach a ∈ A_i do
            foreach X ∈ max''(e) and Y ∈ max(a) do
                T(e) = T(e) ∪ {X ∩ Y}
            end
        end
        max(e) = {Z ∈ T(e) | Z is inclusion-maximal}
    end
end
M(Φ_I) = ⋃_{e∈E} max(e)
```

Algorithm 1 has a runtime exponential in the size of its output, which can itself be exponential in the size of the implication base.

3.2 Intersecting Sets

Once the meet-irreducible elements of the lattice are known, we can compute the lower covers of a set S by intersecting it with the meet-irreducible sets that are not supersets of S and keeping the inclusion-maximal elements. Algorithm 2 runs in time polynomial in the size of $\mathcal{M}(\Phi_{\mathcal{I}})$.

4 Computing with Transversals of Minimal Generators

4.1 Computing Minimal Generators

We compute the minimal generators of an implication-closed set with Algorithms 3 and 4 proposed in [7].

Algorithm 2: Intersection of meet-irreducible elements

```
Input  : A set S and meet-irreducibles M(Φ_I)
Output: The lower covers of S
C = ∅
foreach M ∈ M(Φ_I) do
    C = C ∪ (S ∩ M)
end
Return max(C)
```

Algorithm 3: First minimal generator

```
MinGen (P, L)
Input  : Implications L on the set E and a subset P ⊆ E such that L(P) = P
Output: A minimal generator Q of P
Q ← P
foreach m ∈ P do
    if L(Q \ {m}) = P then
        Q ← Q \ {m}
    end
end
```

Algorithm 3, given a set P and an implication set $\mathcal{L}$, computes a first minimal generator of P. Algorithm 4 computes all the minimal generators of a set P for an implication set $\mathcal{L}$. It has time complexity $O(|\mathcal{L}| \times |\mathcal{G}| \times |\mathcal{P}| \times (|\mathcal{G}| + |\mathcal{P}|))$.

4.2 Computing Minimal Transversals

The problem of computing minimal transversals is a classic that, while extensively studied, still holds many interesting question [4, 5, 3]. It has been shown to be solvable in quasi-polynomial total time. Here, we chose to compute minimal transversals using the, arguably, most simple algorithm, *Berge Multiplication* [2]. Given two hypergraphs $\mathcal{H}_1$ and $\mathcal{H}_2$, their *edgewise union* is defined as :

$$\mathcal{H}_1 \vee \mathcal{H}_2 = \{h_1 \cup h_2 \mid h_1 \in \mathcal{H}_1 \text{ and } h_2 \in \mathcal{H}_2\}$$

We then have

$$Tr(\mathcal{H}_1 \cup \mathcal{H}_2 = min(Tr(\mathcal{H}_1) \vee Tr(\mathcal{H}_2))$$

Which gives rise to Algorithm 5.

5 Experimental Results

We implemented both methods, used them on randomly generated implicational bases and compared their runtimes. Implications were generated by using

Algorithm 4: All minimal generators

Input : Implication set $\mathcal{L}$ on the attribute set E and an $\mathcal{L}$-closed set $P \subseteq E$
Output: All minimal generators $\mathcal{G}$ of P

$\mathcal{G} \leftarrow MinGen(P, \mathcal{L})$
foreach $G \in \mathcal{G}$ **do**
  foreach $L \to R \in \mathcal{L}$ *such that* $L \cup R \cup G \subseteq P$ **do**
    $S \leftarrow L \cup (K \setminus R)$
    $flag \leftarrow true$
    foreach $H \in \mathcal{G}$ **do**
      if $H \subseteq S$ **then**
        $flag \leftarrow false$
      end
    end
    if $flag$ **then**
      $\mathcal{G} \leftarrow \mathcal{G} \cup \{MinGen(S, \mathcal{L})\}$
    end
  end
end

Algorithm 5: All minimal transversals

Input : Hypergraph $\mathcal{H}$
Output: $Tr(\mathcal{H})$

$T = \emptyset$
foreach $E \in \mathcal{H}$ **do**
  $T = min(T \vee \{\{v\} \mid v \in E\})$
end
Return T

NextClosure on formal contexts $(\mathcal{O}, \mathcal{A}, \mathcal{R})$ randomly generated with 50 objects, 12 attributes and a probability d (called *density*) to have $(o, a) \in \mathcal{R}$.

Two cases were considered:

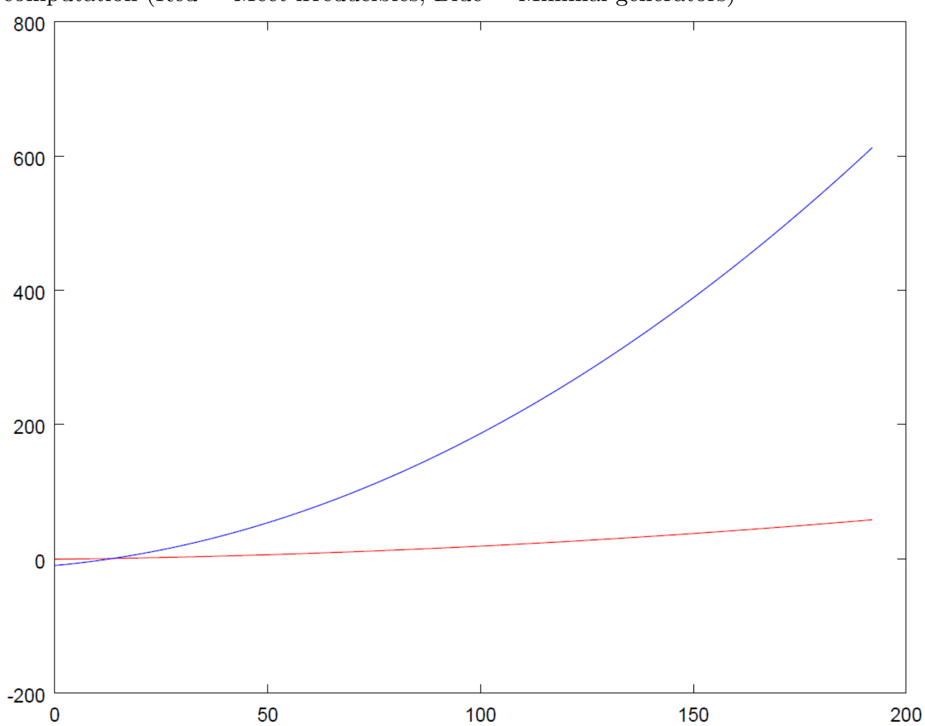
- Single computations of lower covers
- Depth-first searches in lattices involving multiple computations of lower covers

For the first case, we simply computed the lower covers of $\mathcal{A}$. For the second case, we removed some implications as to introduce, in the lattice, sets that are not intents of the formal contexts and we performed the depth-first searches starting from $\mathcal{A}$ and going deeper everytime we encountered a non-intent.

5.1 Single Computations

We generated 1000 contexts with varying numbers of implications and computed the lower covers of $\mathcal{A}$ using the two methods we presented. Figure 1 presents the quadratic interpolation of the runtimes for each method relative to the number of implications in the basis.

Fig. 1. Runtimes for both algorithms relative to the number of implications on a single computation (Red = Meet-irreducibles, Blue = Minimal generators)



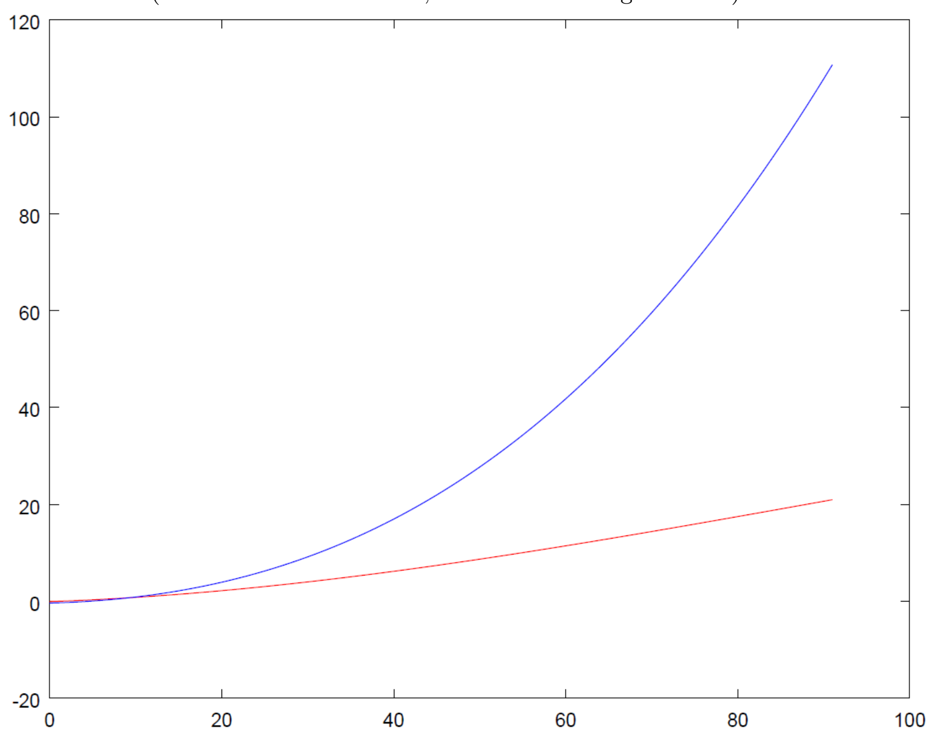
In our experiments, the algorithm using meet-irreducible elements outperformed the one based on minimal generators 79% of the time. Computing the meet-irreducible elements represents 98% of its runtime. The second method devotes 75% of its time to computing minimal generators and 25% on minimal transversals.

5.2 Depth-first Searches

As with single computations, we randomly generated 1000 contexts with varying numbers of implications and performed depth-first searches starting from $\mathcal{A}$.

Runtimes Relative to the Size of the Basis Figure 2 presents the interpolation of the runtimes for each methods relative to the number of implications in the basis.

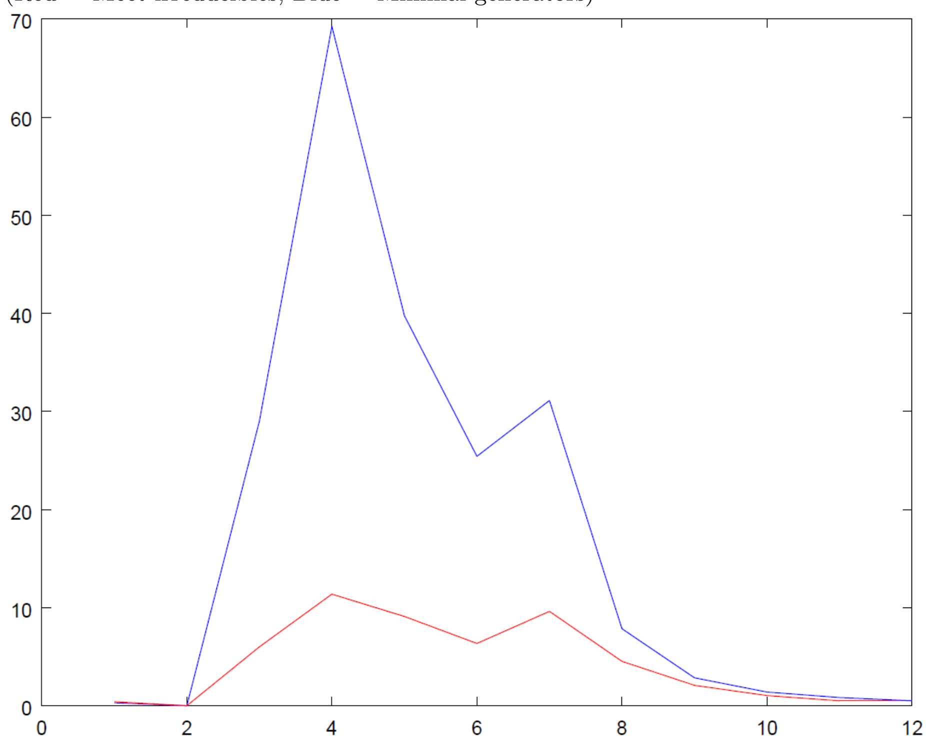
Fig. 2. Runtimes for both algorithms relative to the number of implications for depth-first searches (Red = Meet-irreducibles, Blue = Minimal generators)



Once again, using meet-irreducibles is the most efficient method (it outperformed the minimal generators 72% of the time) and this trend accentuates with the size of the implicational basis. Computing the meet-irreducible elements represents 86% of the total runtime. The second methods devotes 69% of its runtime to the computation of minimal generators and 31% on minimal transversals.

Runtimes Relative to the Depth Figure 3 presents the interpolation of the runtimes for each methods relative to the depth of the search.

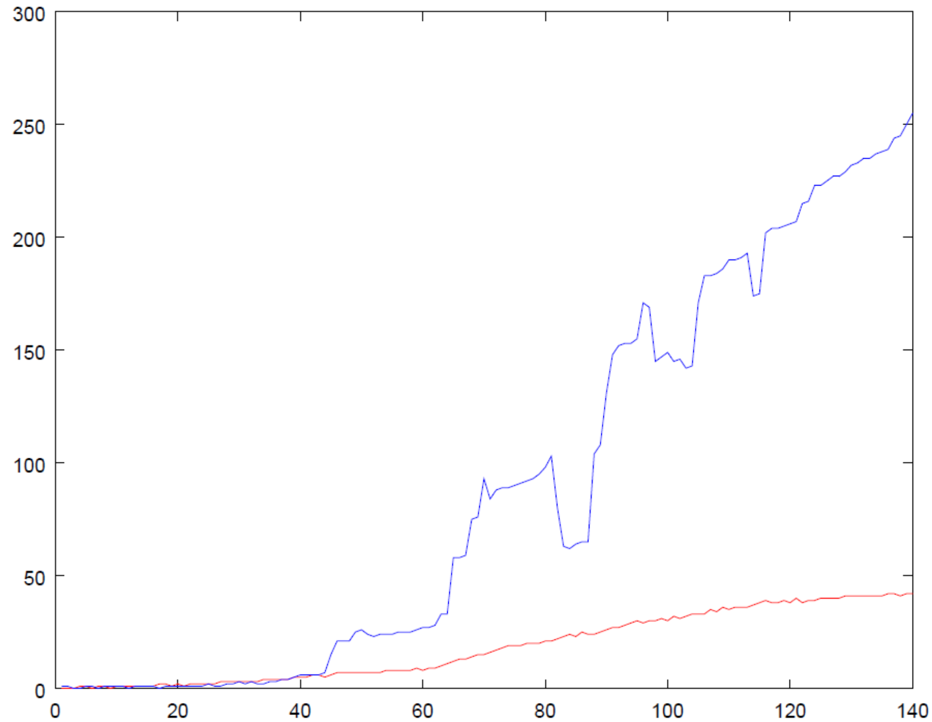
Fig. 3. Runtimes for both algorithms relative to the maximum depth of the search (Red = Meet-irreducibles, Blue = Minimal generators)



We can notice that the algorithm using meet-irreducible elements is much less affected by the depth of the search, most likely due to the fact that most of the computation is done prior to the actual search. Interestingly, the algorithm using minimal generators is least efficient for relatively shallow searches. We believe this is due to the fact that deep searches imply that the lattice is mostly composed of sets that have the property we are looking for. As we used closedness relative to a formal context, this means that those lattices correspond to nearly empty implicational bases.

Runtimes with Increasing Bases We used both methods to perform depth-first searches in a sequence of lattices corresponding to implicational bases $\emptyset \subseteq \mathcal{I}_1 \subseteq ... \subseteq \mathcal{I}_n$. Results are presented in Figure 4.

Fig. 4. Runtimes for both algorithms relative to a growing implicational basis (x-axis = number of implications)(Red = meet-irreducibles, Blue = Minimal generators)



Once again, the runtime of the algorithm using minimal generators skyrockets when the basis grows. The meet-irreducibles method presents much less variance. It should be noted that the meet-irreducible elements are computed from scratch for each basis when Algorithm 1 could be used to speed up the process by using the meet-irreducible elements of the previous lattice to compute the new ones.

6 Discussion

Experimental results indicate that the most efficient way to compute lower covers of implication-closed sets is to first compute the meet-irreducible elements of the lattice and then intersect them. It outperforms the method based on computing minimal transversals of minimal generators even when minimal generators have to be computed only once. During deeper searches, minimal generators have to be computed at each step and the difference is thus more pronounced.

While the algorithm we used for computing minimal transversals is not the most efficient, the fact that this computation represents only a small portion of the second method indicates that, even with the best algorithm, using meet-irreducible elements would be best. Computing these meet-irreducible elements is time-consuming but, once we have them, they are easy to intersect. We believe there is still more room for improvement on the problem of computing meet-irreducible elements from implicational bases.

Acknowledgments The author was supported by the European Union and the French Région Auvergne - Rhône-Alpes.

References

1. Mikhail A. Babin and Sergei O. Kuznetsov. Dualization in lattices given by ordered sets of irreducibles. *Theoretical Computer Science, 2016. CoRR*, abs/1504.01145, 2015.
2. Claude Berge. *Hypergraphs: Combinatorics of Finite Sets*, volume 45. Elsevier, 1984.
3. Thomas Eiter, Georg Gottlob, and Kazuhisa Makino. New Results on Monotone Dualization and Generating Hypergraph Transversals. *SIAM J. Comput.*, 32(2):514–537, 2003.
4. Thomas Eiter, Kazuhisa Makino, and Georg Gottlob. Computational Aspects of Monotone Dualization: A Brief Survey. *Discrete Applied Mathematics*, 156(11):2035–2049, 2008.
5. Michael L. Fredman and Leonid Khachiyan. On the Complexity of Dualization of Monotone Disjunctive Normal Forms. *J. Algorithms*, 21(3):618–628, 1996.
6. Bernhard Ganter and Klaus Reuter. Finding all Closed Cets: A General Approach. *Order*, 8(3):283–290, 1991.
7. Miki Hermann and Baris Sertkaya. On the Complexity of Computing Generators of Closed Sets. In Raoul Medina and Sergei A. Obiedkov, editors, *ICFCA*, volume 4933 of *Lecture Notes in Computer Science*, pages 158–168. Springer, 2008.
8. Petr Krajca, Jan Outrata, and Vilém Vychodil. Advances in Algorithms Based on CbO. In Marzena Kryszkiewicz and Sergei A. Obiedkov, editors, *CLA*, volume 672 of *CEUR Workshop Proceedings*, pages 325–337. CEUR-WS.org, 2010.
9. Marcel Wild. Computations with Finite Closure Systems and Implications. In Ding-Zhu Du and Ming Li 0001, editors, *COCOON*, volume 959 of *Lecture Notes in Computer Science*, pages 111–120. Springer, 1995.

Partial Duplication of Convex Sets in Lattices

Laurent Beaudou and Lhouari Nourine

Clermont-Université, Université Blaise Pascal, LIMOS, CNRS, France
{labeaudo@univ-bpclermont.fr, nourine@isima.fr}

Abstract. In this paper, we generalize the classical duplication of intervals in lattices. Namely, we deal with partial duplication instead of complete convex subsets. We characterize those subsets that guarantee the result to still be a lattice. Moreover, we show that semi-distributive and extremal lattices can be encompassed by such construction where classical duplication fails.

Introduction

The aim of this paper is to give a characterization of several classes of lattices obtained by doubling suborder (not necessary convex) in lattices. This construction generalizes the one that uses convex duplication introduced by Day [1] and followed by several results on the characterizations and algorithmic aspects of these classes of lattices such as: Bounded, Upper Bounded and normal classes of lattices. These results have been obtained by Day on his own [2, 3] or with Nation and Tschantz [4], Bertet and Caspard [5–7] and Geyer [8].

In the opposite of constructing a lattice, decomposing a lattice using properties of duplication to small lattices has been also considered in the literature. Markowsky [9, 10] has shown that extremal lattices can be factorized using prime/coprime property which correspond to the double arrow or perspective relation as introduced in [11]. Janssen and Nourine [12] have given a procedure to decompose a semidistributive lattice according to a simplicial elimination scheme. Others decomposition related to subdirect product construction and congruence can be found in [13–15].

In this paper we give a necessary and sufficient condition for duplications that maintain the lattice structure. We also give other properties that guarantee some combinatorial properties of lattices such as $\wedge$-semidistributivity and extremality. As a by-product of our results and existing ones, we obtain characterizations of some classes of lattices.

1 Preliminaries

In this paper, all considered lattices are finite. For classic definitions of lattices, we refer the reader to the celebrated monograph of Birkhoff [16]. Still, we adress some specific definitions that are of special interest for this documenr. Let $(X, \leqslant)$ be a lattice (denoted $\mathcal{L}$) with $\vee$ and $\wedge$ the usual join and meet operations. An

© Marianne Huchard, Sergei O. Kuznetsov (Eds.): CLA 2016, pp. 33–44,
ISBN 978-5-600-01454-1, National Research University Higher School of Economics, 2016.

element j in X is called join-irreducible in $\mathcal{L}$ if $x = z \vee t$ implies $x = z$ or $x = t$. The set of all join-irreducible elements is denoted by $J(\mathcal{L})$. The set $M(\mathcal{L})$ of all meet-irreducible elements is defined dually. The height $h(\mathcal{L})$ of a lattice $\mathcal{L}$ is the length of the longest chain from $\perp$ to $\top$ (the least and greatest elements of $\mathcal{L}$). Given an element x in X, the set $\uparrow(x, \mathcal{L})$ is the subset of X containing every element y such that $x \leqslant y$. Set $\downarrow(x, \mathcal{L})$ is defined dually.

Given two elements x and y in any lattice $\mathcal{L}$, we use relations $\swarrow$, $\nearrow$ and $\swarrow\!\!\!\nearrow$ defined in [11] as follows,

$$x \swarrow y \text{ if } x \text{ is minimal in } \mathcal{L} - \downarrow(y, \mathcal{L}),$$
$$x \nearrow y \text{ if } y \text{ is maximal in } \mathcal{L} - \uparrow(x, \mathcal{L}),$$
$$x \swarrow\!\!\!\nearrow y \text{ if } x \swarrow y \text{ and } x \nearrow y.$$

Note that whenever $x \swarrow y$, x needs to be a join-irreducible element. Similarly, if $x \nearrow y$, y needs to be a meet-irreducible element.

A lattice is said *meet-semidistributive*, if for all $x, y, z \in \mathcal{L}, x \wedge y = x \wedge z$ implies $x \wedge y = x \wedge (y \vee z)$. It is said *semidistributive* if it is meet-semidistributive and join-semidistributive. We may use $\swarrow\!\!\!\nearrow_{\mathcal{L}}$ to denote the set of pairs (j, m) in $\mathcal{L}$ such that $j \swarrow\!\!\!\nearrow m$. The subscript may be omitted when the context is clear.

2 Doubling construction

We study the possibilities of copying a part of a lattice so that it remains in certain classes of well-known lattices.

The general framework is the following. Let $\mathcal{L}$ be a lattice on some set X with partial order $\leqslant$. Let C be any subset of X that will be copied. We call C' the copy of C (meaning there is a bijection φ from C to C'). The convex closure of C in $\mathcal{L}$ is the set $H(C) = \{y : \exists x, z \in C \text{ with } x \leqslant y \leqslant z\}$. We may now consider the partial order $(X \cup C', \preccurlyeq)$ where the relation $\preccurlyeq$ is defined as follows for any pair (x, y) of elements of $X \cup C'$:

$$x \preccurlyeq y \text{ if } \begin{cases} x \in X, y \in X \text{ and } x \leqslant y \\ x \in X - H(C), y \in C' \text{ and } x \leqslant \varphi^{-1}(y) \\ x \in C', y \in X \text{ and } \varphi^{-1}(x) \leqslant y \\ x \in C', y \in C' \text{ and } \varphi^{-1}(x) \leqslant \varphi^{-1}(y). \end{cases}$$

Note that if x is in $H(C)$, y is in C and $x \leqslant y$, we do not have $x \preccurlyeq \varphi(y)$. It is routine to check that $\preccurlyeq$ defines a partial order on $X \cup C'$. We shall denote this partial order $\mathcal{L}[C]$. If C is the empty set, then this process does not alter $\mathcal{L}$ ($\mathcal{L}[\emptyset] = \mathcal{L}$). We shall distinguish two specific subsets of C, namely the minimal elements $L = \{l_1, l_2, \ldots, l_n\}$ and the maximal elements $U = \{u_1, u_2, \ldots, u_m\}$. Figure 1 depicts an example of a copy with two minimal and two maximal elements in C.

In order to guarantee that the resulting partial order remains a lattice, we need to enforce two properties about C. The first one says that if the join of two

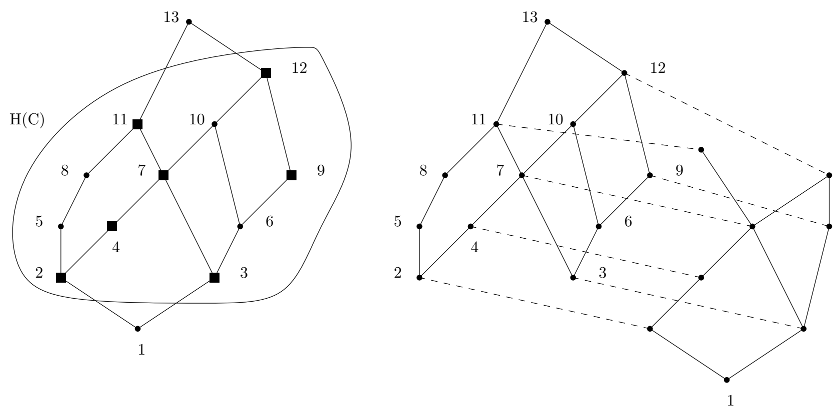


Fig. 1. Depiction of a copy with $U = \{11, 12\}$ and $L = \{2, 3\}$.

copied elements is in the convex closure of C, then it must be copied. The second says that if an element x in $H(C)$ covers an element which is not in $H(C)$, then x must be copied.

$$\forall (x, y) \in C^2, x \vee y \in H(C) \Rightarrow x \vee y \in C, \tag{P_1}$$

$$\forall x \in H(C), \forall y \in X - H(C), x \text{ covers } y \Rightarrow x \in C. \tag{P_2}$$

Remark 1. When U and L are singletons, Property (P_1) says that $(C, \leqslant)$ is a join-sublattice of $\mathcal{L}$.

We shall first prove that properties (P_1) and (P_2) are necessary and sufficient conditions for the resulting partial order to be a lattice.

Proposition 1. *Given a lattice* $\mathcal{L} = (X, \leqslant)$ *and a subset* C *of* X*,* $\mathcal{L}[C]$ *is a lattice if and only if* (P_1) *and* (P_2) *are satisfied.*

Proof. We first show that (P_1) and (P_2) are necessary. Suppose that $\mathcal{L}[C]$ is a lattice. We shall prove both properties separately.

- $\mathbf{(P_1)}$**.** Let x and y be two elements of C such that their join z in $\mathcal{L}$ is in $H(C)$. There is some element u in U such that $z \leqslant u$. By hypothesis, $\mathcal{L}[C]$ is a lattice, so there is a join of $\varphi(x)$ and $\varphi(y)$ in $\mathcal{L}[C]$ let us call it t'. By the definition of $\preccurlyeq$, we have $\varphi(x) \preccurlyeq \varphi(u)$ and $\varphi(y) \preccurlyeq \varphi(u)$. This ensures that t' is between $\varphi(x)$ and $\varphi(u)$. But those elements can only be in C' and there must be an element t in C such that $t' = \varphi(t)$. This element t is then larger than both x and y so that $z \leqslant t$ and by definition of $\preccurlyeq$, $z \preccurlyeq t$. But we also have $\varphi(x) \preccurlyeq z$ and $\varphi(y) \preccurlyeq z$ so that $t \preccurlyeq z$. Finally, $t = z$ and thus the join of x and y is in C.

- **(P_2).** Let x be an element of $H(C)$ and y be an element of $X - H(C)$ which is covered by x in $\mathcal{L}$. Since x is in $H(C)$, there are elements l and u in L and U such that $l \leqslant x \leqslant u$. In turn, $\varphi(l) \preccurlyeq x$. We also know that $y \preccurlyeq x$, thus the join of $\varphi(l)$ and y in $\mathcal{L}[C]$ is less than or equal to x. For a contradiction, suppose that x is not in C. Then x covers y in $\mathcal{L}[C]$ so that the join of y and $\varphi(l)$ in $\mathcal{L}[C]$ must be exactly x. Now $y \preccurlyeq \varphi(u)$ and $\varphi(l) \preccurlyeq \varphi(u)$ so the join of y and $\varphi(l)$ must be below $\varphi(u)$. This is a contradiction since $x \not\preccurlyeq \varphi(u)$.

Let us now prove that (P_1) and (P_2) are sufficient conditions for $\mathcal{L}[C]$ to be a lattice. For this, it suffices to prove that any pair of elements have a least upper bound. From the definition of $\preccurlyeq$, one may check that for any x in $X \cup C'$ we have

$$\uparrow(x, \mathcal{L}[C]) = \begin{cases} \uparrow(x, \mathcal{L}) & \text{if } x \in H(C) \\ \uparrow(x, \mathcal{L}) \cup \varphi(\uparrow(x, \mathcal{L})) & \text{if } x \in X - H(C) \\ \uparrow(\varphi^{-1}(x), \mathcal{L}) \cup \varphi(\uparrow(\varphi^{-1}(x), \mathcal{L})) & \text{if } x \in C', \end{cases}$$

where $\varphi(A)$ denotes all elements that can be written as $\varphi(a)$ for some a in A. In all three cases, there exists an element a in X such that $\uparrow(x, \mathcal{L}[C])$ is exactly $\uparrow(a, \mathcal{L})$ or $\uparrow(a, \mathcal{L}) \cup \varphi(\uparrow(a, \mathcal{L}))$.

Now, notice that for any two subsets of X, A and B, the intersection of A and $\varphi(B)$ is always empty. From this and the distributivity of set operations, we may derive that for any pair (x, y) of elements in $X \cup C'$, there are two elements a and b in X such that

$$\uparrow(x, \mathcal{L}[C]) \cap \uparrow(y, \mathcal{L}[C]) = \begin{cases} \uparrow(a, \mathcal{L}) \cap \uparrow(b, \mathcal{L}) \\ \text{or} \\ (\uparrow(a, \mathcal{L}) \cap \uparrow(b, \mathcal{L})) \cup \varphi(\uparrow(a, \mathcal{L}) \cap \uparrow(b, \mathcal{L})). \end{cases}$$

In the first case, $\uparrow(a, \mathcal{L}) \cap \uparrow(b, \mathcal{L})$ is a subset of X and since $\mathcal{L}$ is a lattice, we know there is a least element. For the second case, let c be the join of a and b in $\mathcal{L}$. We distinguish three subcases.

- If c is in C, then $\varphi(c)$ is definitely less than any element of both $\uparrow(a, \mathcal{L}) \cap \uparrow(b, \mathcal{L})$ and $\varphi(\uparrow(a, \mathcal{L}) \cap \uparrow(b, \mathcal{L}))$. Therefore, $\varphi(c)$ is the least upper bound of a and b.
- If c is not in $H(C)$, then c is a least element of $\uparrow(a, \mathcal{L}) \cap \uparrow(b, \mathcal{L})$. Consider any element x of $\varphi(\uparrow(a, \mathcal{L}) \cap \uparrow(b, \mathcal{L}))$. Then there $\varphi^{-1}(x)$ is an element of C which is greater than a and b. Therefore it is also bigger than c. By the definition of $\preccurlyeq$, x is greater than c in $\mathcal{L}[C]$. Element c is then the least upper bound of a and b.
- If c is in $H(C) - C$, then a and b cannot be both elements of C by (P_1). Property (P_2) basically tells us that for any chain from an element out of $H(C)$ to some element in $H(C)$, there is an element of C. As a consequence, there is a' (respectively b') in C such that $a \leqslant a' \leqslant c$ (respectively $b \leqslant b' \leqslant c$). But then the join of a' and b' must be c and since c is not in C, this leads to a contradiction of (P_1) so that this third subcase can never occur.

Thus any pair of elements in $\mathcal{L}[C]$ has a least upper bound. Since there is also a bottom element in $\mathcal{L}[C]$, we conclude that $\mathcal{L}[C]$ is a lattice. □

As a useful side result, we get that (P_1) and (P_2) imply that the copy of any join-irreducible element in $\mathcal{L}$ is a join-irreducible element of $\mathcal{L}[C]$.

Proposition 2. *Given a lattice $\mathcal{L}$, and a subset C satisfying, (P_1) and (P_2), then for any element j in $J(\mathcal{L}) \cap C$, its copy $\varphi(j)$ is a join-irreducible element of $\mathcal{L}[C]$.*

Proof. Let j be such an element and j_* its unique predecessor in $\mathcal{L}$. For a contradiction, suppose that $\varphi(j)$ is not a join-irreducible element in $\mathcal{L}[C]$. This implies that j_* is in $H(C)$ but has not been copied. Consider two different elements a and b covered by $\varphi(j)$. In particular, $\varphi(j)$ is the join of a and b in $\mathcal{L}[C]$. Furthermore, a and b are less than j in $\mathcal{L}[C]$. Whether they are in C' or in X, this means they are also less than j_*. But j_* is not comparable to $\varphi(j)$ the join of a and b, which is a contradiction since Proposition 1 ensures that $\mathcal{L}[C]$ is a lattice. □

We may first notice that the copying process creates only m new and pairwise distinct meet-irreducible elements.

Remark 2. Given a lattice $\mathcal{L}$ and a subset C satisfying (P_1) and (P_2),

$$M(\mathcal{L}[C]) = M(\mathcal{L}) \cup \{\varphi(u_1), \varphi(u_2), \ldots, \varphi(u_m)\}.$$

Join-irreducible elements can be of four types. Any join-irreducible element of $\mathcal{L}$ which is not copied remains a join-irreducible element in $\mathcal{L}[C]$. By Proposition 2, any join-irreducible element of $\mathcal{L}$ which is copied has its image as a join-irreducible element of $\mathcal{L}[C]$. In addition, any element l in L becomes a join-irreducible element in $\mathcal{L}[C]$. And finally, some elements of C' might be join-irreducible in $\mathcal{L}[C]$ even though their pre-image by φ is not join-irreducible in $\mathcal{L}$. This paragraph is summed up in the following remark.

Remark 3. Given a lattice $\mathcal{L}$ and a subset C satisfying (P_1) and (P_2),

$$J(\mathcal{L}[C]) = (J(\mathcal{L}) - C) \cup \varphi(J(\mathcal{L}) \cap C) \cup \{l_1, l_2, \ldots, l_n\} \cup R,$$

where R denotes the join-irreducible elements of C' which are not the copy of a join-irreducible element.

3 Preserving combinatorial properties

In this paper we want to keep control on the number of join-irreducible elements in order to guarantee that the lattice $\mathcal{L}[C]$ satisfies several combinatorial properties. To this end, we would like the sizes of $J(\mathcal{L}[C])$ and $J(\mathcal{L})$ to differ by only one. Remark 3 ensures that $|J(\mathcal{L}[C])| - |J(\mathcal{L})| = |R| + n$. Since n is at least 1, we need to enforce that R is empty and L is a singleton. Having R as an empty set means that any join-irreducible element of $\mathcal{L}[C]$ in C' is the image of a former join-irreducible of $\mathcal{L}$ in C. We thus states the additional property,

$$\begin{pmatrix} \forall j \in J(\mathcal{L}[C]) \cap C', \varphi^{-1}(j) \in J(\mathcal{L}) \\ L = \{l\} \end{pmatrix}. \tag{P_0}$$

Remark 4. Given a lattice $\mathcal{L}$ and a subset C satisfying (P_0), (P_1) and (P_2),

$$|J(\mathcal{L}[C])| = |J(\mathcal{L})| + 1.$$

In addition we shall consider three properties that will allow us to circumscribe the type of lattice that we want to obtain.

$$L = \{\bot\} \tag{$\bot$}$$

$$U \text{ is a singleton,} \tag{U}$$

$$\forall x \in C, \forall y \in X, \varphi(x) \swarrow y \text{ in } \mathcal{L}[C] \Rightarrow x \swarrow y \text{ in } \mathcal{L}. \tag{$\swarrow$}$$

Each of these properties allows us to control some combinatorial parameter of $\mathcal{L}[C]$. Namely, $(\bot)$ controls the height of the lattice, (U) controls the number of its meet-irreducible elements and $(\swarrow)$ controls the number of pairs related through relation $\swarrow\!\!\!\!\!\!\nearrow$. These are formalized in the following theorem.

Theorem 1. *Given $\mathcal{L}$ a lattice and C a subset satisfying (P_0), (P_1) and (P_2), we have the following implications:*

(i) *if $(\bot)$, then $h(\mathcal{L}[C]) = h(\mathcal{L}) + 1$,*
(ii) *if (U), then $|M(\mathcal{L}[C])| = |M(\mathcal{L})| + 1$,*
(iii) *if $(\swarrow)$, then $|\swarrow\!\!\!\!\!\!\nearrow_{\mathcal{L}[C]}| = |\swarrow\!\!\!\!\!\!\nearrow_{\mathcal{L}}| + |U|$.*

Proof. Fact *(i)* is trivial and *(ii)* is obtained by considering Remark 2. Let us focus on *(iii)*. By Property (P_0), we know that L is a singleton. Let l denote its single element.

Claim 1.1. For any u in U, $l \swarrow\!\!\!\!\!\!\nearrow \varphi(u)$.

The only predecessor of l in $\mathcal{L}[C]$ is $\varphi(l)$ and for any u in U, the only successor of $\varphi(u)$ in $\mathcal{L}[C]$ is u itself. Furthermore $l \preccurlyeq u$, $\varphi(l) \preccurlyeq \varphi(u)$ and $l \not\preccurlyeq \varphi(u)$. This concludes the proof of Claim 1.1.

Claim 1.2. Reciprocally, for any meet-irreducible element m of $\mathcal{L}[C]$, if $l \swarrow\!\!\!\!\!\!\nearrow m$, then there is u in U such that $m = \varphi(u)$.

A stronger statement is that when $l \swarrow m$, then there is u in U such that $m = \varphi(u)$.

We prove the stronger statement for a later use. Let m be an element of $M(\mathcal{L}[C]) - \varphi(U)$, thus m is in X. We shall prove that $l \swarrow m$ cannot occur in $\mathcal{L}[C]$. For a contradiction, suppose that $l \swarrow m$ in $\mathcal{L}[C]$. This means that $\varphi(l) \preccurlyeq m$ and by the definition of $\preccurlyeq$, we get that $l \leqslant m$ in $\mathcal{L}$ and in turn that $\varphi(l) \preccurlyeq m$ which is a contradiction. This concludes the proof of Claim 1.2.

Claim 1.3. Similarily, for any join-irreducible element j of $\mathcal{L}[C]$ and any u in U, if $j \swarrow\!\!\nearrow \varphi(u)$, then $j = l$.

Once again, a stronger statement is obtained when $\swarrow\!\!\nearrow$ is replaced by $\swarrow$.

We also prove the stronger statement. Let u be some element of U and suppose for a contradiction that $\varphi(u)$ is in relation $\swarrow$ with some join-irreducible element j distinct from l. Then $j \not\preccurlyeq \varphi(u)$ and $j \preccurlyeq u$. Thus j cannot be in C' (it would be less than both u and $\varphi(u)$ or not less than both of them). Therefore, j is a join-irreducible element of $\mathcal{L}$ with a single predecessor j_*. In $\mathcal{L}[C]$, j has also a single predecessor j_* which is not in C'. Since we assumed that $j_* \preccurlyeq \varphi(u)$, it cannot be in $H(C)$ (they would be non-comparable). Since j has not been copied, Property (P_2) guarantees that j is not in $H(C)$ either. But by the definition of $\preccurlyeq$, j must be either less than both u and $\varphi(u)$ or not less than both of them. This is a contradiction. This concludes the proof of Claim 1.3.

Claim 1.4. For any j in $J(\mathcal{L}) - C$ and m in $M(\mathcal{L})$, $j \swarrow\!\!\nearrow m$ in $\mathcal{L}$ if and only if $j \swarrow\!\!\nearrow m$ in $\mathcal{L}[C]$.

Let j be an element of $J(\mathcal{L}) - C$ then j is a join-irreducible element of $\mathcal{L}[C]$ and its only predecessor in $\mathcal{L}[C]$ is the same as in $\mathcal{L}$, say j_*. Let m be a meet-irreducible from $\mathcal{L}$. It remains a meet-irreducible element in $\mathcal{L}[C]$. But its only successor in $\mathcal{L}[C]$ can be the same as in $\mathcal{L}$, say m^* or its copy $\varphi(m^*)$. In any case, the comparability of j and m is the same in both $\mathcal{L}$ and $\mathcal{L}[C]$. Same stands for j_* and m. Now if the only successor of m is the same in $\mathcal{L}$ and $\mathcal{L}[C]$, $j \swarrow\!\!\nearrow m$ in $\mathcal{L}$ if and only if $j \swarrow\!\!\nearrow m$ in $\mathcal{L}[C]$. In the case where the only successor of m in $\mathcal{L}[C]$ is $\varphi(m^*)$, if $j \preccurlyeq \varphi(m^*)$, we also have $j \leqslant m^*$. Reciprocally, if $j \leqslant m^*$, we only need to prove that j is not in $H(C)$ to conclude that $j \preccurlyeq \varphi(m^*)$. Suppose that j is in $H(C)$. Since m^* is in C, there is an element u in U such that $m \preccurlyeq \varphi(m^*) \leqslant \varphi(u)$. This implies that m is not in $H(C)$ (otherwise it would not be comparable with $\varphi(m^*)$) so that $j \not\leqslant m$. If $j \swarrow\!\!\nearrow m$ in $\mathcal{L}$, it means that $j_* \leqslant m$ thus j_* is not in $H(C)$ either. In the end, since j has not been copied, Property (P_2) allows us to say that j is not in $H(C)$. So $j \swarrow\!\!\nearrow m$ in $\mathcal{L}$ if and only if $j \swarrow\!\!\nearrow m$ in $\mathcal{L}[C]$, ending the proof of Claim 1.4.

Claim 1.5. For any j in $J(\mathcal{L}) \cap C$ and m in $M(\mathcal{L})$, $j \swarrow\!\!\nearrow m$ in $\mathcal{L}$ if and only if $\varphi(j) \swarrow\!\!\nearrow m$ in $\mathcal{L}[C]$.

We still have to study the case when the join-irreducible element is a copy of a former join-irreducible element. Let j be in $J(\mathcal{L}) \cap C$ and m be a meet-irreducible element of $\mathcal{L}$. We want to prove that $\varphi(j) \swarrow\!\!\nearrow m$ in $\mathcal{L}[C]$ if and only if $j \swarrow\!\!\nearrow m$ in $\mathcal{L}$. In this case, we know that the only predecessor of $\varphi(j)$ is some element between $\varphi(l)$ and $\varphi(j)$. This element can then be written $\varphi(x)$ for some x in C between l and j. Thus, $x \leqslant j_*$. By the definition of $\preccurlyeq$, $\varphi(j) \not\preccurlyeq m$ if and only if $j \not\leqslant m$. Clearly, if $j_* \leqslant m$, we have that $\varphi(x) \preccurlyeq m$. Conversely, if $\varphi(x) \preccurlyeq m$, and $\varphi(j) \not\preccurlyeq m$ it means that $\varphi(j) \swarrow m$. By Property $(\swarrow)$, we have that $j \swarrow m$, thus $j_* \leqslant m$. Let m^* be the only successor of m in $\mathcal{L}$. In $\mathcal{L}[C]$ the only successor of m is either m^* or $\varphi(m^*)$. In the latter case, if $j \leqslant m^*$,

$\varphi(j) \preccurlyeq \varphi(m^*)$ and reciprocally. In the former case, we also have that $\varphi(j) \preccurlyeq m^*$ if and only if $j \leqslant m^*$. Therefore $\varphi(j) \swarrow\!\!\nearrow m$ in $\mathcal{L}[C]$ if and only if $j \swarrow\!\!\nearrow m$ in $\mathcal{L}$. This concludes the proof of Claim 1.5

Summarising the previous results, we obtain that

$$\begin{aligned}\swarrow\!\!\nearrow_{\mathcal{L}[C]} =& \{(l,\varphi(u)) \colon\ u \in U\} \\ & \cup \{(j,m) \in J(\mathcal{L}) \times M(\mathcal{L}) \colon\ j \swarrow\!\!\nearrow m \text{ in } \mathcal{L} \text{ and } j \notin C\} \\ & \cup \{(\varphi(j),m) \in C' \times M(\mathcal{L}) \colon\ j \in J(\mathcal{L}) \cap C \text{ and } j \swarrow\!\!\nearrow m \text{ in } \mathcal{L}\}.\end{aligned}$$

In terms of cardinality, we get that $| \swarrow\!\!\nearrow_{\mathcal{L}[C]} | = | \swarrow\!\!\nearrow_{\mathcal{L}} | + |U|$. □

We may notice that Property ($\swarrow$) is actually only needed for Claim 1.5. Indeed, if this property is not satisfied, we may have new relations between the image of a join-irreducible element and some old meet-irreducible element (see Figure 2).

The proof of the third implication of Theorem 1 can be adapted to prove a fourth implication. We prove separately for an easier reading.

Theorem 2. *Given $\mathcal{L}$ a lattice and C a subset satisfying (P_0), (P_1) and (P_2), we have*

$$(\swarrow) \Rightarrow | \swarrow_{\mathcal{L}[C]} | = | \swarrow_{\mathcal{L}} | + |U|$$

Proof. We use the same ideas as in the proof of Theorem 1.

Claim 2.6. For any u in U, $l \swarrow \varphi(u)$ where $L = \{l\}$.

This a direct consequence of Claim 1.1.

Claim 2.7. For any j in $J(\mathcal{L}) - C$ and m in $M(\mathcal{L})$, $j \swarrow m$ in $\mathcal{L}$ if and only if $j \swarrow m$ in $\mathcal{L}[C]$.

Let j be an element of $J(\mathcal{L}) - C$ then j is a join-irreducible element of $\mathcal{L}[C]$ and its only predecessor in $\mathcal{L}[C]$ is the same as in $\mathcal{L}$, say j_*. Let m be a meet-irreducible from $\mathcal{L}$. It remains a meet-irreducible element in $\mathcal{L}[C]$. But its only successor in $\mathcal{L}[C]$ can be the same as in $\mathcal{L}$, say m^* or its copy $\varphi(m^*)$. In any case, the comparability of j and m is the same in both $\mathcal{L}$ and $\mathcal{L}[C]$. Same stands for j_* and m since j is not copied. Then $j \swarrow m$ in $\mathcal{L}$ if and only if $j \swarrow m$ in $\mathcal{L}[C]$, ending the proof of Claim 2.7.

Claim 2.8. For any j in $J(\mathcal{L}) \cap C$ and m in $M(\mathcal{L})$, $j \swarrow m$ in $\mathcal{L}$ if and only if $\varphi(j) \swarrow m$ in $\mathcal{L}[C]$.

By Property ($\swarrow$), we have for any j in $J(\mathcal{L}) \cap C$ and m in $M(\mathcal{L})$, $\varphi(j) \swarrow m$ in $\mathcal{L}[C]$ imply $j \swarrow m$ in $\mathcal{L}$.

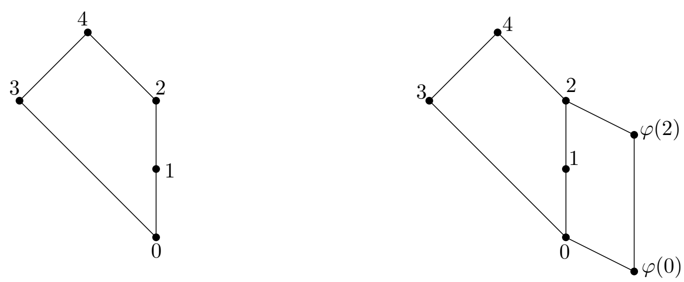
For the converse, let j be in $J(\mathcal{L}) \cap C$ and m be a meet-irreducible element of $\mathcal{L}$ such that $j \swarrow m$ in $\mathcal{L}$. We want to prove that $\varphi(j) \swarrow m$ in $\mathcal{L}[C]$. First, by definition of $\preccurlyeq$, $\varphi(j)$ is incomparable to m. If $j_* \notin H(C)$ then $j_* = \varphi(j)_*$

by definition of $\preccurlyeq$, and then $\varphi(j) \swarrow m$ in $\mathcal{L}[C]$. Now suppose that $j_* \in H(C)$. In this case, we know that the only predecessor $\varphi(j)_*$ of $\varphi(j)$ is some element between $\varphi(l)$ and $\varphi(j)$. This element can then be written $\varphi(x) = \varphi(j)_*$ for some x in C between l and j. Then $x \leqslant j_*$ and thus $\varphi(x) \preccurlyeq m$. So $\varphi(j) \swarrow m$ in $\mathcal{L}[C]$.

Summarising the previous results (and the strong versions of Claims 1.2 and 1.3), we obtain that

$$\begin{aligned}\swarrow_{\mathcal{L}[C]} =& \{(l, \varphi(u)) :\ u \in U\}\\ &\cup \{(j,m) \in J(\mathcal{L}) \times M(\mathcal{L}) :\ j \swarrow m \text{ in } \mathcal{L} \text{ and } j \notin C\}\\ &\cup \{(\varphi(j), m) \in C' \times M(\mathcal{L}) :\ j \in J(\mathcal{L}) \cap C \text{ and } j \swarrow m \text{ in } \mathcal{L}\}.\end{aligned}$$

In terms of cardinality, we get that $|\swarrow_{\mathcal{L}[C]}| = |\swarrow_{\mathcal{L}}| + |U|$. □



$\mathcal{L}$ $\qquad$ $\mathcal{L}[C]$

$C = \{0, 2\}$

Fig. 2. $\varphi(2) \swarrow 3$ in $\mathcal{L}[C]$ while we do not have $2 \swarrow 3$ in $\mathcal{L}$.

Lattices characterizations given in the following theorem can be found in several papers (see for example [10, 17, 18]).

Theorem 3. *Let $\mathcal{L}$ be a finite lattice. Then $\mathcal{L}$ is*

- *meet-semidistributive if and only if $|\swarrow\!\!\!\!\nearrow| = |J(\mathcal{L})|$ [18].*
- *semidistributive if and only if $|\swarrow\!\!\!\!\nearrow| = |J(\mathcal{L})| = |M(\mathcal{L})|$ [18].*
- *meet-extremal if and only if $h(\mathcal{L}) = |M(\mathcal{L})|$ [10].*
- *extremal if and only if $h(\mathcal{L}) = |J(\mathcal{L})| = |M(\mathcal{L})|$ [10].*
- *distributive if and only if $|\swarrow| = |J(\mathcal{L})| = |M(\mathcal{L})|$ [17].*

Remark 5. Notice that a lattice that is semidistributive and extremal does not imply that is distributive (see Figure 3). In fact it is not graded. This explain the property ($\swarrow$).

As a corollary of Theorems 1 2 and 3, we obtain a wider range of possibilities to build specific types of lattices by preserving some combinatorial characterizations.

Corollary 1. *Given a lattice $\mathcal{L}$ and a subset C verifying (P_0), (P_1) and (P_2) the following implications are true:*

1. *$\mathcal{L}$ is distributive, $(\swarrow)$, $(\perp)$ and (U) imply that $\mathcal{L}[C]$ is distributive.*
2. *$\mathcal{L}$ is semidistributive, $(\swarrow)$, and (U) imply that $\mathcal{L}[C]$ is semidistributive*
3. *$\mathcal{L}$ is meet-semidistributive and $(\swarrow)$ imply that $\mathcal{L}[C]$ is meet-semidistributive*
4. *$\mathcal{L}$ is extremal, $(\perp)$ and (U) imply that $\mathcal{L}[C]$ is extremal*
5. *$\mathcal{L}$ is meet-extremal and $(\perp)$ imply that $\mathcal{L}[C]$ is meet-extremal*

One challenging problem is the characterization of contexts where their concepts lattices satisfy the considered properties. For doubling convex sets, there are nice FCA characterization and algorithms that recognize bound, lower (upper) bounded, semidistributive and convex lattices [5–7, 1–4, 8].

Acknowledgment The authors are very grateful to the referee for their constructive comments. This work has been funded by the french national research agency (ANR Graphen, 2015-2018).

References

1. Day, A.: A simple solution to the word problem for lattices. Canad. Math. Bull. **13** (1970) 253–254
2. Day, A.: Doubling constructions in lattice theory. Canadian J. Math **44** (1992) 242–269
3. Day, A.: Congruence normality: the characterization of the doubling class of convex sets. Algebra Universalis **31**(3) (1994) 397–406
4. Day, A., Nation, J.B., Tschantz, S.: Doubling convex sets in lattices and a generalized semidistributivity condition. Order **6(2)** (1989) 175–180
5. Bertet, K.: Sur quelques aspects structurels et algorithmiques des treillis. Phd thesis, Université de Paris 7 (1998)
6. Bertet, K., Caspard, N.: Doubling convex sets in lattices: Characterizations and recognition algorithms. Order **19**(2) (2002) 181–207
7. Caspard, N.: Étude structurelle et algorithmique de classes de treillis obtenus par duplication. Phd thesis, Université de Paris I (1998)
8. Geyer, W.: Generalizing semidistributivity. Order **10**(1) (1993) 77–92
9. Markowsky, G.: The factorization and representation of lattices. Transactions of the American Mathematical Society **203** (1975) 185–200
10. Markowsky, G.: Primes, irreducibles and extremal lattices. Order **9** (1992) 265–290
11. Wille, R.: Subdirect decomposition of concept lattices. Algebra Universalis **17** (1983) 275–287
12. Janssen, P., Nourine, L.: A simplicial elimination scheme for meet-semidistributive lattices and interval collapsing. Algebra Universalis **50**(2) (2003) 171–178
13. Viaud, J., Bertet, K., Demko, C., Missaoui, R.: The reverse doubling construction. In: KDIR 2015 - Proceedings of the International Conference on Knowledge Discovery and Information Retrieval, part of the 7th International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management (IC3K 2015), Volume 1, Lisbon, Portugal, November 12-14, 2015. (2015) 350–357

14. Viaud, J., Bertet, K., Demko, C., Missaoui, R.: Subdirect decomposition of contexts into subdirectly irreducible factors. In: Proceedings of the International Workshop on Formal Concept Analysis and Applications, FCA&A 2015, co-located with 13th International Conference on Formal Concept Analysis (ICFCA 2015), Nerja, Málaga, Spain, June 23-26, 2015. (2015) 49–64
15. Wille, R.: Subdirect product construction of concept lattices. Discrete Mathematics **63**(2) (1987) 305 – 313
16. Birkhoff, G.: Lattice Theory. 3rd edn. Volume 25 of Coll. Publ. XXV. American Mathematical Society, Providence (1967)
17. Monjardet, B.: The presence of lattice theory in discrete problems of mathematical social sciences. why. Mathematical Social Sciences **46**(2) (2003) 103 – 144 Mathematical Theories of Allocation of Discrete Resources: Equilibria, Matchings, Mechanisms.
18. Ganter, B., Wille, R.: Formal Concept Analysis: Mathematical Foundations. Springer-Verlag Berlin (1999)

A Lattice-Based Consensus Clustering Algorithm

Artem Bocharov, Dmitry Gnatyshak, Dmitry I. Ignatov, Boris G. Mirkin, and Andrey Shestakov

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

Abstract. We propose a new FCA-based algorithm for consensus clustering, FCA-Consensus. As the input the algorithm takes T partitions of a certain set of objects obtained by k-means algorithm after its T different executions. The resulting consensus partition is extracted from an antichain of the concept lattice built on a formal context $objects \times classes$, where the classes are the set of all cluster labels from each initial k-means partition. We compare the results of the proposed algorithm in terms of ARI measure with the state-of-the-art algorithms on synthetic datasets. Under certain conditions, the best ARI values are demonstrated by FCA-Consensus.

Keywords: consensus clustering, k-means, Formal Concept Analysis, ensemble clustering, lattice-based clustering

1 Introduction and related work

It seems, consensus clustering approach became popular on the international scene after the paper of A. Strehl and J. Ghosh [1]. Since then consensus clustering is used in such areas as bioinformatics, web-document clustering and categorical data analysis.

As the input the consensus clustering approach usually takes T partitions of a certain set of objects obtained, for example, by k-means algorithm after its T different executions with possibly different k. The resulting consensus partition is build from the matrix $objects \times classes$, where the classes are the set of all cluster labels from each initial k-means partition. Thus, the main goal of consensus clustering is to find (recover) an optimal partition, i.e. to guess the proper number of resulting clusters and put the objects into each block of partition correctly. To evaluate the proposed approach researchers usually hypothesise that if a particular consensus clustering approach is able to guess a proper k and attain high accuracy on labeled datasets, then it can be used in pure unsupervised setting. This task is worth consideration mainly due to two reasons: We do not know a proper k in advance, and k-means is unstable due to randomness of initialisation [2]. However, we can use right guesses of each of the ensemble algorithms to build (recover) a proper partition.

© Marianne Huchard, Sergei O. Kuznetsov (Eds.): CLA 2016, pp. 45–56,
ISBN 978-5-600-01454-1, National Research University Higher School of Economics, 2016.

In [3], consensus clustering algorithms are classified in three main groups: probabilistic approaches [4,5]; direct approaches [1,6,7,8], and pairwise similarity-based approaches [9,10]. In the last category of methods, the (i, j)-th entry a_{ij} of the consensus matrix $A = (a_{ij})$ shows the number of partitions in which objects g_i and g_j belong to the same cluster.

In the previous papers [11,12], a least-squares consensus clustering approach was invoked from the paper [13], to equip it with a more recent clustering procedure for consensus clustering and compare the results on synthetic data of Gaussian clusters with those by the more recent methods. Here, our main goal is to propose a lattice-based consensus clustering algorithm by means of FCA and show its competitive applicability. To the best of our knowledge, a variant of FCA-based consensus approach was firstly proposed to cluster genes into disjoint sets [14]. For those, who are interested theoretical properties of different consensus procedures and its relationship with FCA we could recommend [15].

The paper is organised in five sections. In Section 2, we refresh some definitions from FCA, introduce partitions and their lattice, and prove that any partition lattice can be easily mapped to a concept lattice. In Section 3, we introduce our modification of Close-by-One algorithm for consensus clustering. In Section 4, we provide our experimental results with synthetic data both for individual behaviour of FCA-Consensus and its comparison with the state-of-the-art existing methods. Section 5 concludes the paper and outlines prospective ways of research and developments.

2 Basic definitions

First, we recall several notions related to lattices and partitions.

Definition 1. *A partition of a nonempty set A is a set of its subsets $\sigma = \{B \mid B \subseteq A\}$ such that $\bigcup_{B\in\sigma} B = A$ and $B \cap C = \emptyset$ for all $B, C \in \sigma$. Every element of σ is called* block.

Definition 2. *A partition lattice of set A is an ordered set $(Part(A), \vee, \wedge)$ where $Part(A)$ is a set of all possible partitions of A and for all partitions σ and ρ supremum and infimum are defined as follows:*

$$\sigma \vee \rho = \{N_\rho(B) \cup \bigcup_{C\in N_\rho(B)} N_\sigma(C) | B \in \sigma\},$$

$$\sigma \wedge \rho = \{B \cap C \mid \text{ for all } B \in \sigma \text{ and } C \in \rho\}, \text{where}$$

$N_\rho(B) = \{C \mid B \in \sigma, C \in \rho$ and $B \cap C \neq \emptyset\}$ and $N_\sigma(C) = \{B \mid B \in \sigma, C \in \rho$ and $B \cap C \neq \emptyset\}$.

Definition 3. *Let A be a set and let $\rho, \sigma \in Part(A)$. The partition ρ is finer than the partition σ if every block B of σ is a union of blocks of ρ, that is $\rho \leq \sigma$.*

Equivalently one can use traditional connection between supremum, infimum and partial order in the lattice: $\rho \leq \sigma$ iff $\rho \vee \sigma = \sigma$ ($\rho \wedge \sigma = \rho$).

Now, we recall some basic notions of Formal Concept Analysis (FCA) [16]. 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* $\underline{\mathfrak{B}}(G, M, I)$.

Theorem 1. *(Ganter&Wille [16]) For a given partially ordered set $\mathfrak{P} = (P, \leq)$ the concept lattice of the formal context $\mathbb{K} = (J(P), M(P), \leq)$ is isomorphic to the Dedekind–MacNeille completion of $\mathfrak{P}$, where J(P) and M(P) are set of join-irreducible and meet-irreducible elements of $\mathfrak{P}$.*

Theorem 2. *(this paper) For a given partition lattice $\mathfrak{L} = (Part(A), \vee, \wedge)$ there exist a formal context $\mathbb{K} = (P_2, A_2, I)$, where $P_2 = \{\{a, b\} \mid a, b \in A$ and $a \neq b\}$, $A_2 = \{\sigma \mid \sigma \in Part(A)$ and $|\sigma| = 2\}$ and $\{a, b\}I\sigma$ when a and b belong to the same block of σ. The concept lattice $\underline{\mathfrak{B}}(P_2, A_2, I)$ is isomorphic to the initial lattice $(Part(A), \vee, \wedge)$.*

Proof. According to Theorem 1 the concept lattice of the context $\mathbb{K}_{DM} = (J(\mathfrak{L}), M(\mathfrak{L}), \leq)$ is isomorphic to the Dedekind–McNeille completion of $\mathfrak{L}$. The Dedekind–McNeille completion of a lattice is its isomorphic lattice by the definition (as a minimal completion which forms a lattice). So, we have to show that contexts $\mathbb{K}$ and $\mathbb{K}_{DM}$ (or their concept lattices) are isomorphic.

E.g., from [17] (Lemma 1, Chapter 4, Partition Lattices), we have that the atoms of a partition lattice are those its partitions which have only one block of two elements, the rest are singletons, and its coatoms are partitions into two blocks.

It is evident that all the atoms are meet-irreducible and all the coatoms are join-irreducible and that there are no other irreducible elements of the partition lattice $\mathfrak{L}$.

Let σ and ρ be two partitions from $\mathfrak{L}$, $\sigma \in J(\mathfrak{L})$ and $\rho \in M(\mathfrak{L})$, and $\sigma \leq \rho$. It means that all blocks of σ are subsets of blocks of ρ and the non-trivial block $\{i, j\} \in \sigma$ is a subset of one of the blocks of ρ. Note that A_2 coincides with the coatom set. It directly implies that $\{i, j\}I\rho$ iff an atom σ with block $\{i, j\}$ is finer than a coatom ρ.□

In addition we can show the correspondence between elements of $\mathfrak{L} = (Part(A), \vee, \wedge)$ and formal concepts of $\underline{\mathfrak{B}}(P_2, A_2, I)$. Every $(A, B) \in \underline{\mathfrak{B}}(P_2, A_2, I)$ corresponds to $\sigma = \bigwedge B$ and every pair $\{i, j\}$ from A is in one of σ blocks, where $\sigma \in Part(A)$. Every $(A, B) \in \underline{\mathfrak{B}}_{DM}(J(\mathfrak{L}), M(\mathfrak{L}), \leq)$ corresponds to $\sigma = \bigwedge B = \bigvee A$.

Example 1. In Fig. 1, one can see the diagram of a concept lattice isomorphic to partition lattice of 4-element set.

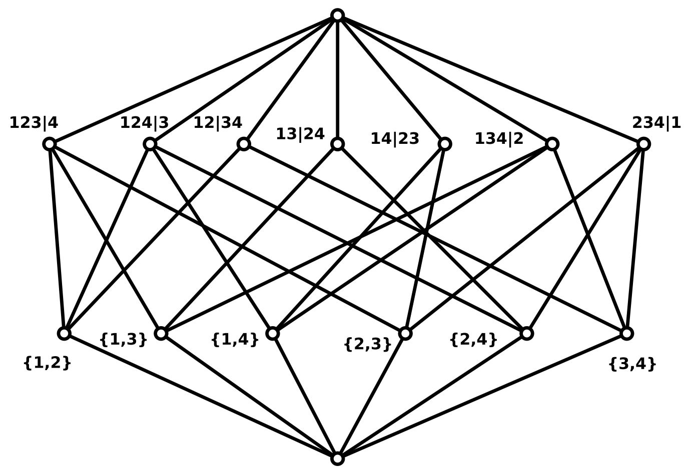


Fig. 1. The line diagram of a concept lattice isomorphic to the partition lattice of 4-element set (reduced labeling).

3 FCA-Consensus: close by object

To work in FCA terms we need to introduce a (formal) *partition context* that corresponds to the matrix X from the previous subsection. Let us consider such a context $\mathbb{K}_{\mathcal{R}} = (G, \sqcup M_t, I \subseteq G \times \sqcup M_t)$, where G is a set of objects, $t = 1, \ldots, T$, and each M_t consists of labels of all clusters in the t-th k-means partition from the ensemble. For example, gIm_{t1} means that object g has been clustered to the first cluster by t-th clustering algorithm in the ensemble.

Our FCA-Consensus algorithm looks for $\mathfrak{S}$, an antichain of concepts of $\mathbb{K}_{\mathcal{R}}$, such that for every (A, B) and (C, D) the condition $A \cap C = \emptyset$ is fulfilled. Here, the concept extent A corresponds to one of the resulting clusters, and its intent contains all labels of the ensemble members that voted for the objects from A being in one cluster. The input cluster sizes may vary, but it is a reasonable consensus hypothesis that at least $\lceil T/2 \rceil$ should vote for a set of objects to be in cluster.

One can prove a theorem below, where by *true partition* we mean the original partition into clusters to be recovered.

Theorem 3. *In the concept lattice of a partition context* $\mathbb{K}_{\mathcal{R}} = (G, \sqcup M_t, I \subseteq G \times \sqcup M_t)$*, there is the antichain of concepts* $\mathfrak{S}$ *such that all extents of its concepts* A_i *coincide with* S_i *from* σ*, the true partition, if and only if* $S_i'' = S_i$ *where* $i = 1, \ldots, |\sigma|$.

Proof. The proof is trivial by noting the fact that blocks of partitions are non-intersected and each block should be closed to form a concept extent.□

In fact, it happens if all ensemble algorithms has voted for all objects from S_i being in one concept (cluster). However, this is rather strong requirement and we should experimentally study good candidates for such an antichain.

The algorithm below works as Close by One (CbO) [18] adding objects one by one and checking a new canonicity conditions. Here it is modified in the following way: we need to stop adding objects to a particular concept in our candidate antichain $\mathfrak{S}$ until $|Y| \geq \lceil T/2 \rceil$, where Y is the intent of this current concept. Moreover, the covered objects at a particular step should not be added with any concept to the antichain $\mathfrak{S}$ further.

Algorithm 1: Main($(G, M, I), T$)

Input: a partition context (G, M, I) and the number of ensemble clusterers T
Output: $\mathfrak{S}$

```
C = ∅
for all g ∈ G do
    if g ∉ C then
        gpp = g''
        gp = g'
        𝔖.enqueue(gpp, gp)
        C = C ∪ gpp
    end if
end for
return Process((G, M, I), k, 𝔖)
```

Thus, the resulting antichain $\mathfrak{S}$ may not cover all objects but we can add each non-covered object g to a concept $(A, B) \in \mathfrak{S}$ with maximal size of the intersection, $|B \cap g'|$. Traditionally, the algorithm consists of two parts, a wrapper procedure, Main, and a recursive procedure, Process.

4 Experimental results

All evaluations are done on synthetic datasets that have been generated using Matlab. Each of the datasets consists of 300 five-dimensional objects comprising three randomly generated spherical Gaussian clusters. The variance of each

Algorithm 2: Process$((G, M, I), T, \mathfrak{S})$

$\mathfrak{T} = \mathfrak{S}$
$Cover = \emptyset$ While $\mathfrak{T} \neq \emptyset$
$\mathfrak{T}.dequeue(A, B)$
if $A \cap Cover = \emptyset$ **then**
    $Cover = Cover \cup A$
    $\mathfrak{P}.enqueue(A, B)$
    for all $g \in min(G \setminus Cover)$ **do**
        $X = A \cup \{g\}$
        $Y = X'$
        if $|Y| \geq \lceil T/2 \rceil$ **then**
            $Z = Y'$
            if $\{h | h \in Z \setminus X,\ h < g\} = \emptyset$ **then**
                $\mathfrak{P}.dequeue(A, B)$
                $\mathfrak{P}.enqueue(Z, Y)$
                $Cover = Cover \cup Z$
            end if
        end if
    end for
end if
if $\mathfrak{S} = \mathfrak{P}$ **then**
    return $\mathfrak{P}$
end if
$\mathfrak{S} = \mathfrak{P}$
return Process$((G, M, I), T, \mathfrak{P})$

cluster lies in $0.1 - 0.3$ and its center components are independently generated from the Gaussian distribution $\mathcal{N}(0, 0.7)$.
Let us denote thus generated partition as λ with k_λ clusters. The *profile* of partitions $\mathcal{R} = \{\rho^1, \rho^2, \ldots, \rho^T\}$ for consensus algorithms is constructed as a result of T runs of k-means clustering algorithm starting from random k centers.

We carry out the experiments in four settings:

1. Investigation of influence of the number of clusters $k_\lambda \in \{2, 3, 5, 9\}$ under various numbers of minimal votes (Fig. 2),
   a) two clusters case $k_\lambda = 2, k \in \{2, 3, 4, 5\}$,
   b) three clusters case $k_\lambda = 3, k \in \{2, 3\}$,
   c) five clusters case $k_\lambda = 5, k \in \{2, 5\}$,
   d) nine clusters case $k_\lambda = 9, k \in \{2, 3, 4, 5, 6, 7, 8, 9\}$;
2. Investigation of the numbers of clusters of ensemble clusterers with fixed number of true clusters $k_\lambda = 5$ (Fig. 3),
   a) $k = 2$,
   b) $k \in \{2, 3, 4, 5\}$,
   c) $k \in \{5\}$,
   d) $k \in \{5, 6, 7, 8, 9\}$
   e) $k = 9$;
3. Investigation of the number of objects $N \in \{100, 300, 500, 1000\}$ (Fig. 4);
4. Comparison with other state-of-the-art algorithms (Fig. 5–8),
   a) two clusters case $k_\lambda = 2, k \in \{2, 3, 4, 5\}$,
   b) three clusters case $k_\lambda = 3, k \in \{2, 3\}$,
   c) five clusters case $k_\lambda = 5, k \in \{2, 5\}$,
   d) nine clusters case $k_\lambda = 9, k \in \{2, 3, 4, 5, 6, 7, 8, 9\}$.

Each experiment encompasses 10 runs for every of 10 generated datasets. Such meta-parameters as the dimension number $p = 3$, the number of partitions (clusterers) in the ensemble $T = 100$, and the parameters of Gaussian distribution have been fixed for each experiment. After applying consensus algorithms, Adjusted Rand Index (ARI) [3] for the obtained consensus partition σ and the generated partition λ is computed as $ARI(\sigma, \lambda)$.

Given two partitions $\rho^a = \{R_1^a, \ldots, R_{k_a}^a\}$ and $\rho^b = \{R_1^b, \ldots, R_{k_b}^b\}$, where $N_h^a = |R_h^a|$ is the cardinality of R_h^a, $N_{hm} = |R_h^a \bigcap R_m^b|$, N is the number of objects, $C_a = \sum\limits_h \binom{N_h^a}{2} = \sum\limits_h \frac{N_h^a(N_h^a - 1)}{2}$.

$$ARI(\rho^a, \rho^b) = \frac{\sum\limits_{hm} \binom{N_{hm}}{2} - C_a C_b \Big/ \binom{N}{2}}{\frac{1}{2}(C_a + C_b) - C_a C_b \Big/ \binom{N}{2}} \tag{2}$$

This criterion expresses similarity of two partitions; its values vary from 0 to 1, where 1 means identical partitions, and 0 means totally different ones.

4.1 Comparing consensus algorithms

The lattice-based consensus results have been compared with the results of the following algorithms (Fig. 5–8):

- AddRemAdd ([19,11])
- Voting Scheme (Dimitriadou, Weingessel and Hornik, 2002) [6]
- cVote (Ayad, 2010) [7]
- Condorcet and Borda Consensus (Dominguez, Carrie and Pujol, 2008) [8]
- Meta-CLustering Algorithm (Strehl and Ghosh, 2002) [1]
- Hyper Graph Partitioning Algorithm [1]
- Cluster-based Similarity Partitioning Algorithm [1]

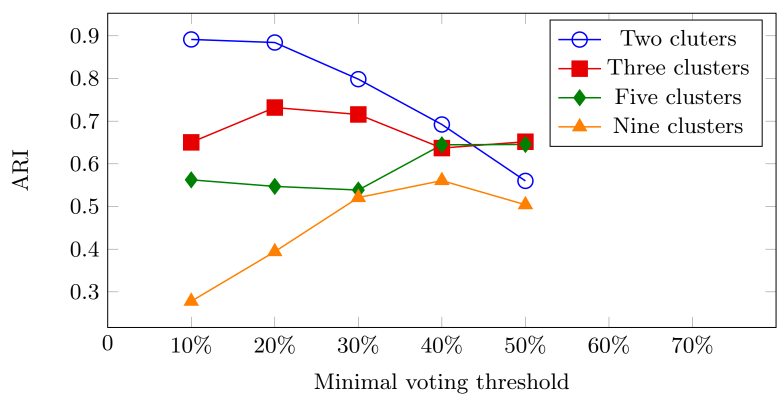


Fig. 2. Influence of minimal voting threshold to ARI for different number of true clusters

To provide the reader with more details we show the values of ARI graphically for each dataset out of ten used. The summarised conclusions are given in the next section.

5 Conclusion

Through experimentation we have draw the following conclusions:

- Optimal voting threshold in terms of minimal intent size for the resulting antichain of concepts is not constant; moreover, it is not usually a majority of votes of ensemble members (see Fig. 2).
- A rather expected conclusion: FCA-based consensus clustering method works better if set the number of blocks for the ensemble clusterers to be equal to the size of the original (true) partition (see Fig. 3).

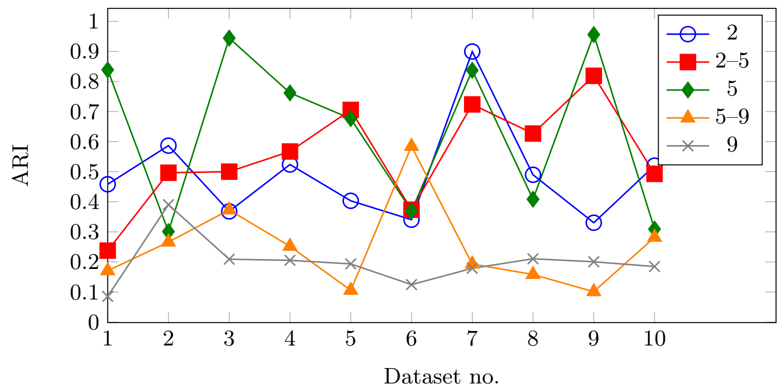


Fig. 3. Influence of minimal voting threshold to ARI for different numbers of clusters of the ensemble clusterers (each point is averaged over 10 datasets)

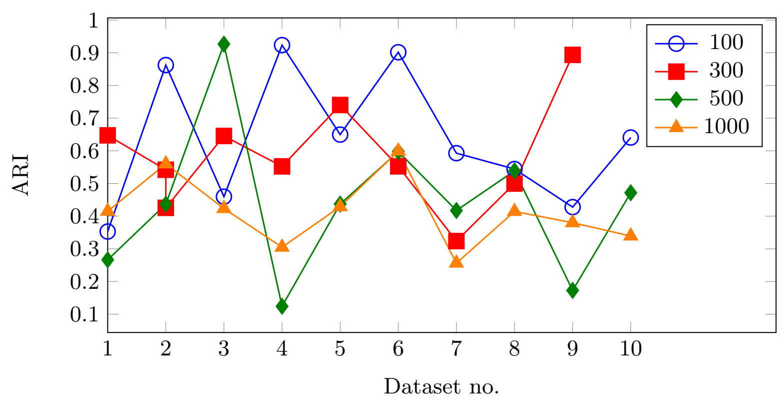


Fig. 4. Influence of different numbers of objects to ARI

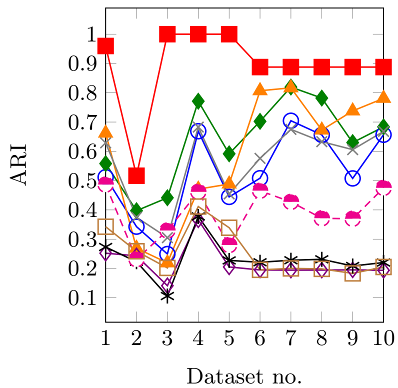


Fig. 5. Two clusters

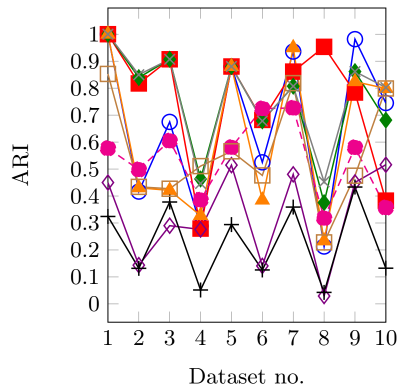


Fig. 6. Three clusters

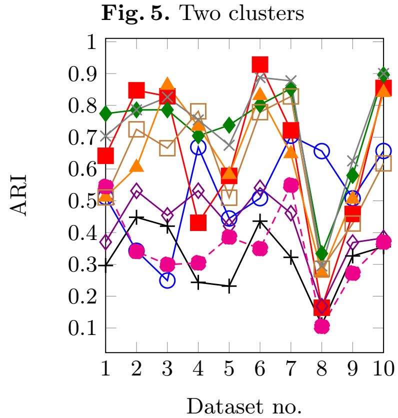


Fig. 7. Five clusters

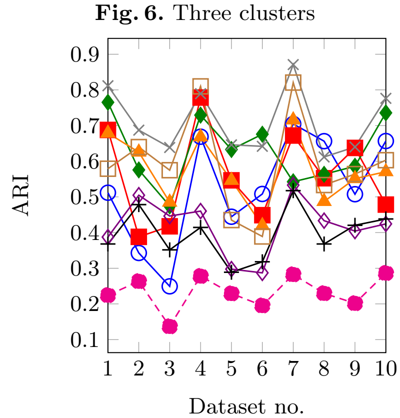


Fig. 8. Nine clusters

– ARI depends on the number of objects: The higher the number, the lower ARI (see Fig. 4).
– For two (and almost for all three) true clusters our method beats the other compared algorithms and in some cases consensus clustering task is solved with 100% accuracy (see Fig. 5–6).
– For larger number of clusters, our method is positioned as the median among the compared methods (see Fig. 7–8).

Thus, the first step on synthetic datasets has been done and we need to test the approach on real datasets. The used version of CbO can be modified for usage on the space of all partition labels for the cases when we have more objects than those labels. The algorithm complexity and time-efficiency should carefully studied and compared with those of the existing algorithms. An interesting venue is to use partition lattices as a search space to find an optimal partition. For example, one can build a pattern structure [20] over partitions similar to one in [21] and analyse the correlation of stability indicies [22] of the partitions as pattern concepts with ARI measure. By so doing it is possible to understand what are the good regions in the lattice for searching an optimal partition that can be built from existing ones via partition union and intersection operations.

Acknowledgments. We would like to thank Jaume Baixeries, Amedeo Napoli, Alexei Buzmakov, Mehdi Kaytoue, and Oleg Anshakov for their comments, remarks and explicit and implicit help during paper preparations. The paper was prepared within the framework of the Basic Research Program at the National Research University Higher School of Economics (HSE) and supported within the framework of a subsidy by the Russian Academic Excellence Project "5-100".

References

1. Strehl, A., Ghosh, J.: Cluster ensembles — A knowledge reuse framework for combining multiple partitions. Journal of Machine Learning Research **3** (2002) 583–617
2. Kuncheva, L.I., Vetrov, D.P.: Evaluation of stability of k-means cluster ensembles with respect to random initialization. IEEE Trans. Pattern Anal. Mach. Intell. **28**(11) (2006) 1798–1808
3. Ghosh, J., Acharya, A.: Cluster ensembles. Wiley Interdisc. Rew.: Data Mining and Knowledge Discovery **1**(4) (2011) 305–315
4. Topchy, A.P., Jain, A.K., Punch, W.F.: A mixture model for clustering ensembles. In: Proceedings of the Fourth SIAM International Conference on Data Mining, Lake Buena Vista, Florida, USA, April 22-24, 2004. (2004) 379–390
5. Wang, H., Shan, H., Banerjee, A.: Bayesian cluster ensembles. Statistical Analysis and Data Mining **4**(1) (2011) 54–70
6. Dimitriadou, E., Weingessel, A., Hornik, K.: A combination scheme for fuzzy clustering. IJPRAI **16**(7) (2002) 901–912
7. Ayad, H., Kamel, M.S.: On voting-based consensus of cluster ensembles. Pattern Recognition **43**(5) (2010) 1943–1953

8. Dominguez, X.S., Carrie, J.S., Pujol, F.A.: Fuzzy clusterers combination by positional voting for robust document clustering. Procesamiento del lenguaje natural **43** (2009) 245–253
9. Mirkin, B.: Clustering: A Data Recovery Approach. CRC Press (2012)
10. Guénoche, A.: Consensus of partitions : a constructive approach. Advances in Data Analysis and Classification **5**(3) (2011) 215–229
11. Mirkin, B.G., Shestakov, A.: Least square consensus clustering: Criteria, methods, experiments. In: Advances in Information Retrieval - 35th European Conference on IR Research, ECIR 2013, Moscow, Russia, March 24-27, 2013. Proceedings. (2013) 764–767
12. Mirkin, B., Shestakov, A. In: A Note on the Effectiveness of the Least Squares Consensus Clustering. Springer New York, New York, NY (2014) 181–185
13. Mirkin, B., Muchnik, I.: Geometrical interpretation of clustering scoring functions. In: Methods for the Analysis of Multivariate Data in Economics, Novosibirsk, Nauka Publisher (1981) 3 – 11 (In Russian).
14. Hristoskova, A., Boeva, V., Tsiporkova, E.: A Formal Concept Analysis Approach to Consensus Clustering of Multi-Experiment Expression Data. BMC Bioinformatics **15** (2014) 151
15. Domenach, F., Tayari, A.: Implications of Axiomatic Consensus Properties. In: Algorithms from and for Nature and Life: Classification and Data Analysis. Springer International Publishing, Cham (2013) 59–67
16. Ganter, B., Wille, R.: Formal Concept Analysis: Mathematical Foundations. 1st edn. Springer-Verlag New York, Inc., Secaucus, NJ, USA (1999)
17. Graetzer, G.: General Lattice Theory. Akademie-Verlag (1978)
18. Kuznetsov, S.O.: A fast algorithm for computing all intersections of objects from an arbitrary semilattice. Nauchno-Tekhnicheskaya Informatsiya Seriya 2 - Informatsionnye protsessy i sistemy (1) (1993) 17–20
19. Mirkin, B.: Core concepts in Data Analysis: summarization, correlation, visualization. Springer (2011)
20. Ganter, B., Kuznetsov, S.O.: Pattern structures and their projections. In: Conceptual Structures: Broadening the Base, 9th International Conference on Conceptual Structures, ICCS 2001, Stanford, CA, USA, July 30-August 3, 2001, Proceedings. (2001) 129–142
21. Codocedo, V., Napoli, A.: Lattice-based biclustering using partition pattern structures. In: ECAI 2014 - 21st European Conference on Artificial Intelligence, 18-22 August 2014, Prague, Czech Republic - Including Prestigious Applications of Intelligent Systems (PAIS 2014). (2014) 213–218
22. Kuznetsov, S.O.: Stability as an estimate of degree of substantiation of hypotheses derived on the basis of operational similarity. Nauchno-Tekhnicheskaya Informatsiya Seriya 2 - Informatsionnye protsessy i sistemy (12) (1990) 21–29

Some Experimental Results on Randomly Generating Formal Contexts

Daniel Borchmann[1,2] and Tom Hanika[3]

[1] Chair of Automata Theory
Technische Universität Dresden, Germany
[2] Center for Advancing Electronics Dresden
Technische Universität Dresden, Germany
[3] Knowledge & Data Engineering Group
University of Kassel, Germany

daniel.borchmann@tu-dresden.de, tom.hanika@cs.uni-kassel.de

Abstract We investigate different simple approaches to generate random formal contexts. To this end, we consider for each approach the empirical correlation between the number of intents and pseudo-intents. We compare the results of these experiments with corresponding observations on real-world use-cases. This comparison yields huge differences between artificially generated and real-world data sets, indicating that using randomly generated formal contexts for applications such as benchmarking may not necessarily be meaningful. In doing so, we additionally show that the previously observed phenomenon of the "Stegosaurus" does not express a real correlation between intents and pseudo-intents, but is an artifact of the way random contexts are generated.

Keywords: Formal Concept Analysis, Pseudo-Intents, Closure Systems

1 Introduction

In the early times of Formal Concept Analysis [1], the study of lattices represented as the concept lattice of a particular formal context $\mathbb{K}$ was one of the main driving motivations. For this one has to solve the computational task of determining all formal concepts of $\mathbb{K}$, one of the first algorithmic challenges in the field of FCA. Since then, many algorithms have been developed to solve this task.

With the rise of a multitude of algorithms it became increasingly important to be able to *compare* these algorithms. One of the first comparisons was done in 2002 by Kuznetsov [2]. The data sets used in this comparison were all "randomly generated", a notion that up to today is not completely understood. Consequently, [2] regrets that there is no deeply investigated algorithm for generating random contexts.

From its original motivation, Formal Concept Analysis has since then evolved into an active research area with many connections to fields outside the scope of this original approach. Nevertheless, the study of properties of lattices in terms

ISBN 978-5-600-01454-1, National Research University Higher School of Economics, 2016.

of corresponding formal contexts is still one of the main lines of research. One of the earliest observations in this direction was that every concept lattice can be understood as the lattice of all closed sets of the valid implications of the underlying formal context. This observation did not only open up connections to fields like data-base theory, data-mining, and logic. It also fostered research on finding efficient algorithms for extracting small *bases* of implications of a given formal context. One of those bases, called the *canonical base*, stands out as base of minimal size for which an explicit construction is known. Recall that for a formal context $\mathbb{K} = (G, M, I)$ the canonical base $\mathcal{L}(\mathbb{K})$ is the set of implications defined by

$$\mathcal{L}(\mathbb{K}) := \{P \to P'' \mid P \text{ is pseudo intent of } \mathbb{K}\},$$

where *pseudo intents of* $\mathbb{K}$ are subsets of M such that $P \neq P''$ and for all pseudo intents $Q \subsetneq P$ it is true that $Q'' \subseteq P$. This recursive definition of pseudo intents makes theoretical investigations of the canonical base rather difficult. Indeed, Babin and Kuznetsov [3] showed that recognizing pseudo-intents is coNP-complete.

Although there are bases whose computation may be more worthwhile in practice, the canonical base is still of major interest for both research and applications. In 2011, Bazhanov and Obiedkov [4] made a performance comparison of the known algorithms to compute canonical bases. For this they used seven distinct real world contexts. More recently is a parallel approach by Borchmann and Kriegel [5]. To evaluate their algorithm they used random contexts as well as real-world contexts from the `fcarepository.com` (which disappeared recently).

It emerges that evaluating the performance of algorithms for computing the set of formal concepts as well as computing the canonical base heavily depends on the choice of the available data sets. Because obtaining real-world data sets may be a challenging endeavor, one often resolve to use artificially-generated "random contexts" instead. However, a thorough theory of randomly generated formal contexts is missing, and even experimental studies are hard to find. This is where this work tries to step in. In particular, it aims to shed some light on a phenomenon we shall call the *Stegosaurus-phenomenon*, a surprising empirically observed correlation between the number of pseudo intents and the number of formal concepts of formal contexts. We shall show that the phenomenon depends strongly on the method used for generating random contexts. Other random context generators show similar, but substantially different phenomena.

Finally, we want to compare our approaches of randomly generating formal contexts with two data sets constructed from real world data, namely from BibSonomy and form the Internet Movie Database. Not surprisingly, the correlation between the number of intents and pseudo-intents in these data sets differs considerably to those observed in the randomly generated contexts. This reminds of an obvious but too rarely stated meme from the early days of formal concept analysis: don't invent data!

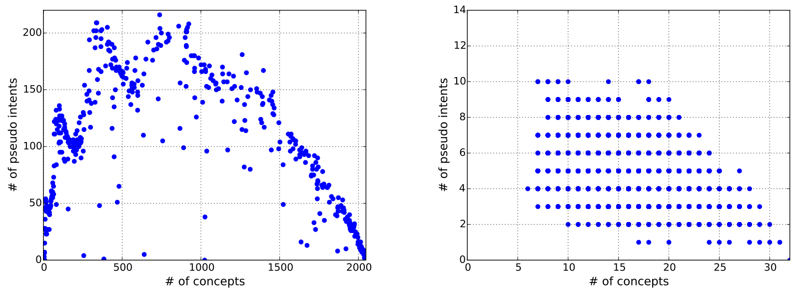


Figure 1. Experimentally observed correlation between the number of intents and pseudo-intents of randomly generated formal contexts on twelve attributes (left), plot of all formal contexts on five attributes (right).

2 Related Work

The original observation of a correlation between the number of intents and pseudo-intents first appeared in [6]. This work was originally not concerned with investigating this relationship, but with representing closure operators on sets by means of formal contexts of minimal size. However, during the experiments on the efficiency of this approach, a correlation between the number of intents and the number of pseudo-intents of randomly generated formal contexts was discovered. The original phenomenon is shown in Figure 1 and has subsequently been called the *Stegosaurus* (because, with some fantasy, the shape of Figure 1 resembles the one of this well-known dinosaur).

Further investigation was conducted in a talk at the in Formal Concept Analysis Workshop in 2011. There not only the experimental setup was discussed in more detail, but also questions were raised that are connected to the experiment. Most importantly, it was asked whether the phenomenon really exists, or whether it was just a programming error or an artifact of the experimental setup. Indeed, using a reimplementation[4], the second author was later able to independently verify the outcome of the experiment.

Another question raised in this investigation was whether the way the formal contexts were generated has an impact on the outcome of the experiment. The problem here is that although in the original experiment the formal contexts were generated in a uniformly random manner, the underlying closure systems were not. This is because closure systems can have multiple representations by means of formal contexts, and the number of those contextual representations may differ widely between different closure systems. Therefore, uniformly choosing a formal contexts does not mean to choose a closure system in a uniform way.

A first attempt to remove the shortcomings of the way random formal contexts are generated was conducted by Ganter [7]. In this work an approach was

[4] `https://github.com/tomhanika/fcatran`

investigated to correctly generate closure systems on a finite set with a uniform random distribution. However, while the proposed algorithm was conceptually simple, it turned out that it is not useful for our experiment. Indeed, it has been shown that the proposed algorithm is only practical for closure systems on sets of up to 7 elements, whereas the original experiment needs a size of at least 9 or 10 to exhibit the characteristic pattern of Figure 1. This is also the reason why an earlier computation of all reduced formal contexts on five attributes, shown in Figure 1, was not helpful to investigate the phenomenon.

3 Experiments

The purpose of this section is to present different experimental approaches to enhance our understanding of the Stegosaurus phenomenon. For this purpose, we shall first recall the original experiment that first exhibited the Stegosaurus. After this, we shall discuss an alternative approach of randomly generating formal contexts that fixes the number of attributes per object. Then we shall consider another method proposed [8]. Finally, we compare our findings against experiments on real world data.

All computations presented in this section were conducted using `conexp-clj`[5].

3.1 Original Experiment

The original experiment that first unveiled the Stegosaurus-phenomenon randomly generated formal contexts as follows. For a given number of attributes N and some $p \in [0, 1]$, first the number of objects is randomly chosen between 1 and 2^N. Then for each pair (g, m) of an object g and an attribute m, a biased coin with probability p was used to determine whether g has attribute m.

Applying this algorithm to generate 1000 formal contexts with $N = 10$ leads to the picture in Figure 2. The result does not change qualitatively by repetition. The provided generating algorithm seems biased towards creating contexts that lie on some idiosyncratic curve. This curve exhibits multiple spikes (in the given picture at least 4 can be identified) and a general skew to the left. Contexts beneath that curve are hit infrequently, above that curve even less. The behavior at the right end of the plot is expected, since when almost every subset of M is an intent, the number of pseudo intents must be low: the number of pseudo-intents of a formal context $\mathbb{K} = (G, M, I)$ is at most $2^{|M|}$ minus the number of intents of $\mathbb{K}$. On the other hand, the behavior in the rest of the picture is not as easily explained and still eludes proper understanding.

We also plotted a histogram in Figure 2 which contains a bin for every occurring number of pseudo intents. By the height of the erected rectangle above each bin we can observe the frequency of appearance of a formal context with that particular number of pseudo intents. The distribution shown in Figure 2 has an expected spike at zero: while generating a random formal context with a high

[5] `https://github.com/exot/conexp-clj`

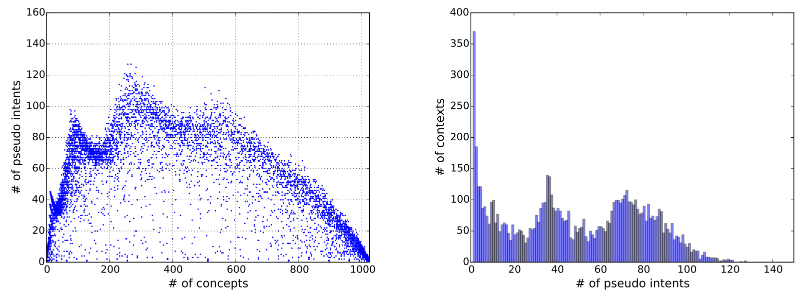


Figure 2. Experimentally observed correlation: Between the number of intents and pseudo-intents (left) and the distribution of the number of contexts having a given number of pseudo intents (right), for 1000 randomly generated formal contexts with ten attributes

probability of crosses, the chances of hitting the ten object-vectors spanning a contra-nominal-scale context is high. Apart from that, there is an cumulation of contexts for approximately 40 and 70 pseudo intents. For some reason the algorithm favors context with those pseudo intent numbers. This could also mean that the same context is generated for multiple times.

These unexpected results lead to many more questions to generate a deeper understanding of the connection between the number of formal concepts and pseudo intents.

One of these questions is what happens if a lot of contexts are generated that way. To address this question, we created five million random contexts using the introduced method. This led to the result shown in Figure 3. In contrast to Figure 2 we see a filled picture. Almost all combinations below the characteristic curve have been realized by at least one context. Only a small seam of not realized combinations is left at the bottom. At a second glance we observe that the whole characteristic curve seems shifted up by approximately ten to twenty pseudo intents. Even more interestingly, a fifth spike can be imagined at about 800 concepts. Furthermore, even in this figure there are still some random context hoovering even above the spikes. This leads to the conjecture that there are contexts with even larger canonical bases that cannot be computed feasibly by the applied method.

In Figure 3 we also plotted the according histogram like we did in Figure 2. The distribution of contexts is of course shifted up since more contexts are generated. But it still resembles the one in Figure 2. In particular, for contexts with about 50 pseudo-intents, a plateau can be observed.

Another question is how far the number of attributes we have chosen for our experiments has an influence on the shape of the Stegosaurus. Since in the first discovery of the Stegosaurus was made with a context that has eleven attributes, the question about the influence of N on the phenomenon is natural. To investigate this question, we computed, still using the same method, several

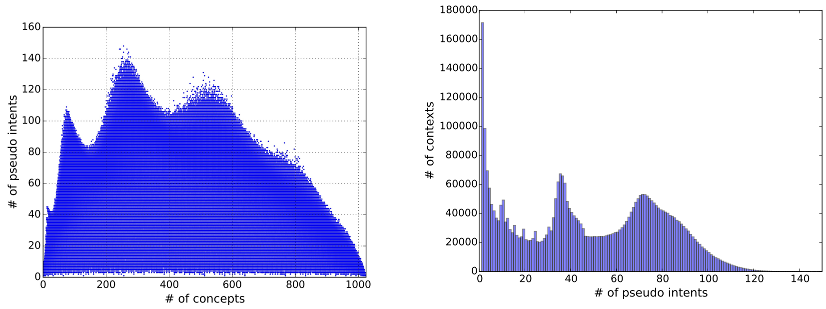


Figure 3. Experimentally observed correlation: Between the number of intents and pseudo-intents (left) and the distribution of the number of contexts having a given number of pseudo intents (right), for five million randomly generated formal contexts with ten attributes, using experiment in Section 3.1.

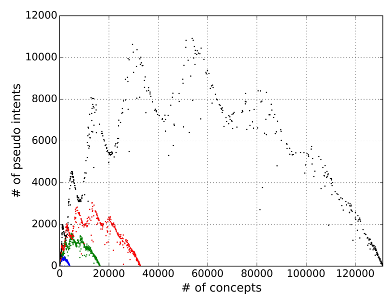


Figure 4. The influence of increasing m for the original experiment.

formal contexts with up to seventeen attributes. As can be see in Figure 4, the characteristic Stegosaurus curve is present in all of them. However, we also can see an increase in spikes.

Therefore, we conjecture that the occurrence of the Stegosaurus phenomenon seems independent from the value of N.

3.2 Increasing the number of pseudo-intents

As described in the previous section, in the original experimental setup the number of pseudo-intents of randomly generated formal contexts increases with the number of iterations. A natural question is whether we can find an upper bound on the number of pseudo-intents a formal context can have given that the number of intents is fixed. For this purpose, we investigate an alternative approach of generating formal contexts that is described in this section.

Let us say that a formal context $\mathbb{K} = (G, M, I)$ has *fixed row-density* if the number of attributes for each object $g \in G$ is the same. In other words, for all $g, h \in G$ we have $|g'| = |h'|$. It is clear how to obtain such formal contexts: let $k, n \in \mathbb{N}$ with $0 \leq k < n$. Let $M = \{1, \ldots, n\}$ and choose $G \subseteq \binom{M}{k}$. Then the formal context (G, M, I), where $(S, i) \in I$ if and only if $i \in S$, has fixed row-density. Let us call a formal context $\mathbb{K}$ with fixed row-density *object maximal* if $\mathbb{K}$ is object clarified and no new object can be added to $\mathbb{K}$ such that the formal context is still object clarified and has fixed row-density. In other words, $\mathbb{K}$ is object maximal with fixed row-density if and only if $\mathbb{K}$ is isomorphic to $\mathbb{K}_{n,k} := (\binom{M}{k}, M, \ni)$, where $M = \{1, \ldots, n\}$.

Formal contexts with fixed row-density have been used by Kuznetsov in his performance comparison of concept lattice generating algorithms [2]. The following observation had already been hinted at (so we suppose) in [9], when it was claimed that constructing formal contexts with as much as $\binom{|M|}{\lfloor |M|/2 \rfloor}$ pseudo-intents is easy.

Proposition 1. *Let $k < n - 1$. The number of pseudo intents of $\mathbb{K}_{n,k}$ is $\binom{n}{k+1}$.*

Proof. Let $M = \{1, \ldots, n\}$. For all $P \subseteq M$ with $|P| = k + 1$ we see that $P \subsetneq P'' = M$. For all proper subsets $Q \subsetneq P$ it is clear that Q is an intent of $\mathbb{K}_{n,k}$, as it can be represented as an intersection of subsets of M of size k. Therefore, the subsets of M of cardinality $k+1$ are in fact pseudo-intents of $\mathbb{K}_{n,k}$, and there are $\binom{n}{k+1}$ many of them. Because each $k+1$-elemental subset $P \subseteq M$ satisfies $P'' = M$, we also have that there are no other pseudo-intents in $\mathbb{K}_{n,k}$.

In fact, for any attribute set M, object maximal formal contexts with fixed row-density are the contexts with the largest canonical base we discovered in our experiments so far. The results of applying this algorithm for $N = 10$ for various k can be seen in Figure 5. We observe the highest peak in the plot for $k = 4$, as Proposition 1 implies. For $k = 1$ we notice the ten possible formal contexts are plotted in between one to ten concepts, as expected, with up to 45 pseudo intents. In contrast to that, we find the ten possible contexts in the $k = 9$ case stringed along the axis for contexts with one pseudo intent, as expected for contexts resembling a contra-nominal scale.

An overlay of all those plots is shown in Figure 6, together with an overlay for $N = 11$ which, despite the thin and high spikes, both are reminiscent of Figure 2.We observe multiple sharp spikes, seven in the case of $N = 10$ and eight in the case of $N = 11$. The top of each spike is the object maximal formal context with fixed row-density for the corresponding k. For every k we observe a hump in the graph before the spike starts. The reasons for that hump as well as for the dale afterwards are unclear.

The curiosity about the Stegosaurus-phenomenon increases even more after overlaying Figure 6 with Figure 3. In contrast to the observation so far, now some spikes seem to "grow" out of dales in the original Stegosaurus plot. In particular, the question if the upper bound in the original Stegosaurus plot states some inherent correlation between the number of pseudo intents and the number of intents can be safely negated at this point.

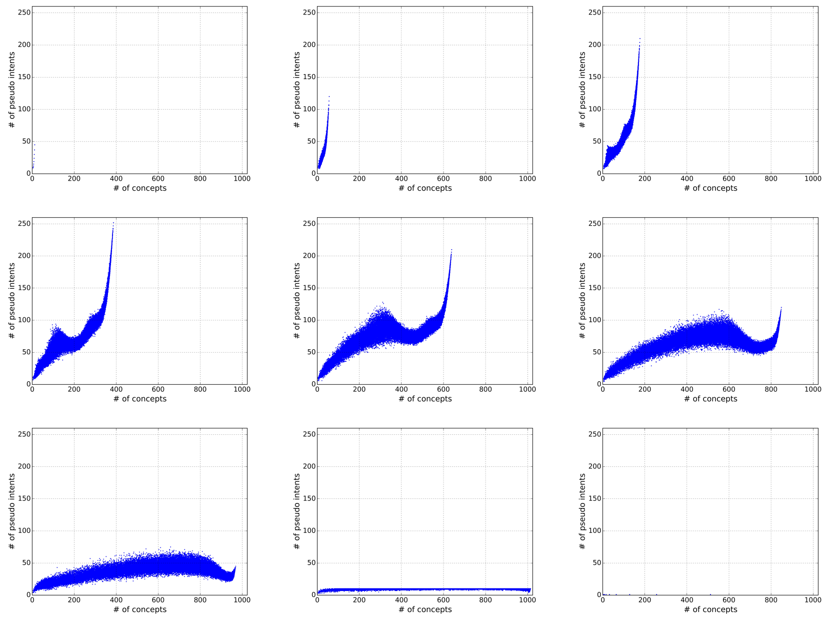


Figure 5. 100,000 random fixed row-density contexts for $|M| = 10$, plotted for $k = 1$ (upper left) up to $k = 9$ (down right).

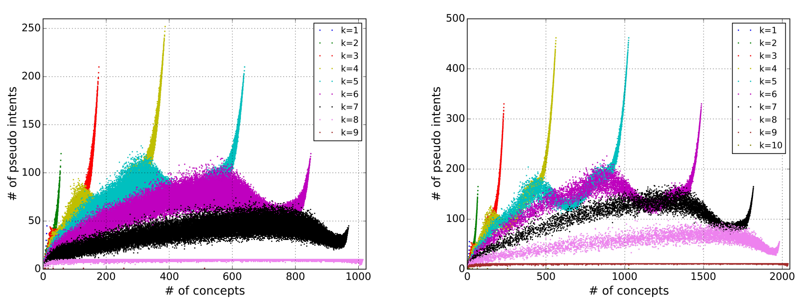


Figure 6. 100,000 random fixed row-density context for $m = 10$ (left) and $m = 11$ (right) for various k (best looked at in color).

3.3 SCGaz-Contexts

In 2013, Rimsa et al. [8] contributed a synthetic formal context generator named SCGaz[6]. The goal for this generator was to create random object irreducable formal contexts with a chosen density that have no full or empty rows and columns. The authors employ four different algorithms for each phase of the generation process, i.e., reaching minimum density, regular filling, coping with problems near the maximum density, and brute force. Since the interactions of these algorithms is rather involved and not possible to describe in short, we refer the reader to [8].

When this tool is invoked with a fixed number of attributes, a number (or an interval) of objects must be provided, as well as a density. In cases when the provided density does not fit with the other parameters, the density is interchanged with 0.5. For example, the request to generate a context with 32 objects, 5 attributes and density 0.9 is impossible, since there is only one object clarified context with those parameters, an object maximal formal context with fixed row-density, which has a density of 0.5.

This particularity in the usage of SCGaz made it tiring to generate a large number of random formal contexts, since a correct density had to be pre-calculated. We did so and generated a set of 3.5 Million contexts for a set of ten attributes, varying number of objects, and three different densities per object-attribute-number combination. The result is shown in Figure 7.

The first thing to observe is again a spike structure. However, the previously observed skew as in Figure 2 is gone, and the upper bound of the plot is significantly higher than in Figure 2. Furthermore, there seems to be a unnatural gap in the plot. This missing piece is an artifact of our parameter generation for invoking SCGaz, in particular the density bound calculations. We verified this by generating a small number of contexts using random densities which led to contexts resembling the same behavior as in Figure 7, but without the missing piece. However, in favor of the more filled plot we decided to include Figure 7 instead of the smaller sample.

Comparing the results from SCGaz with the one obtained in Section 3.2, we observe that some spikes stemming from context with fixed row-density emerge from dales in the SCGaz plot and others adapt closely to the SCGaz spikes. Nevertheless, all spikes we have seen in Section 3.2 outnumber the ones from SCGaz by the number of pseudo intents.

3.4 Real-World Contexts

The purpose of this section is to compare our observations about artificially generated formal contexts with results from experiments based on real-world data sets. The actual experiment is the same as before: we compute for a collection of formal contexts the number of intents and pseudo-intents and plot the result. However, in contrast to our previous experiments, we do not generate the formal

[6] `https://github.com/rimsa/SCGaz`

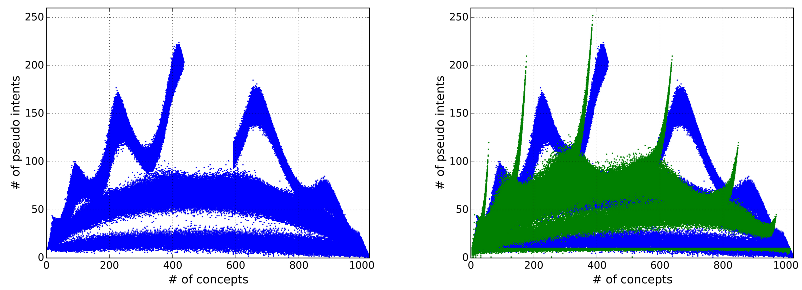


Figure 7. 3.5 Million random contexts generated by SCGaz, using ten attributes and varying density and number of objecst (left) and the same overlain by Figure 6 (right).

contexts using a designated procedure, but use some readily available data sets for this.

The first data sets stems from the BibSonomy project[7]. BibSonomy is a social publication sharing system that allows a user to tag publications with arbitrary tags. Using the publicly available anonymized data sets [10] of BibSonomy[8], we created 2835 contexts as follows. For every user u we defined a set of attributes M_u consisting of the twelve most frequently used tags of the user. The set of objects per user is the set of all the publications stored in BibSonomy. The incidence relation then is the obvious relation between publications and their tags.

The results are depicted in Figure 8. Note that even if the cardinality of the attribute set is twelve, the plot is shown only for up to 1024 intents, because no contexts with more than 1024 intents are contained in the data set.

The majority of the contexts seem to lie near a linear function of the number of concepts. Hence, it looks like the left part of the Stegosaurus phenomenon. Even a first spike can be accounted for with about 60 pseudo intents.

The distribution of contexts, however, behaves very differently. Of course, the first spike for contexts with no pseudo-intents is missing, as the contra-nominal scale is not common in real world data. Furthermore, we can find that there is no wide dale in the graph, like it is observed in Figure 2.

For our second real world data set we chose to use the Internet Movie Database[9]. We created 57582 formal contexts using the following approach. For every actor (context) we took the set of his movies (objects) and the related star-votes. Every movie can be rated from one to ten, and the ten bins of votes were considered as attributes. Every rate-bin that has at least 10% of the total amount of votes was considered as being present for an object. The resulting graphs are shown in Figure 9.

[7] `http://bibsonomy.org`
[8] `http://www.kde.cs.uni-kassel.de/bibsonomy/dumps/`
[9] `http://www.imdb.com`

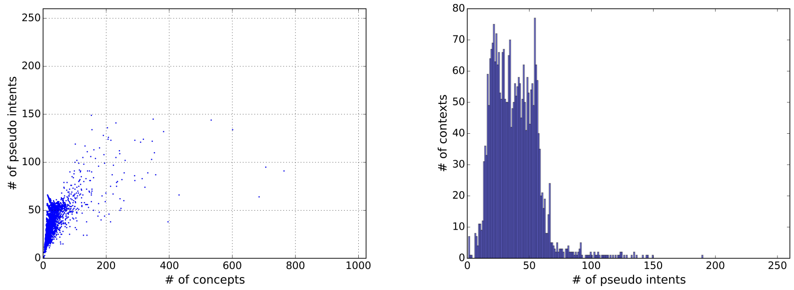


Figure 8. 2835 contexts created using the public BibSonomy data set.

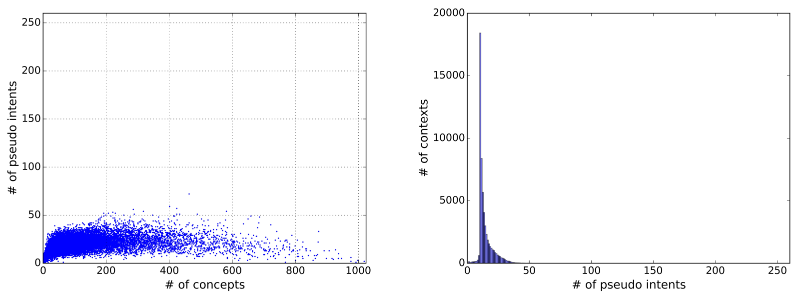


Figure 9. Formal contexts created using the Internet Movie Database.

We observe a quite different behavior to that of the classical Stegosaurus as well as to that of the BibSonomy data set. These contexts fill the area for infrequent contexts of the experiment in Section 3.1. Their canonical bases are mostly below 50 pseudo intents and the number of formal concepts goes up to 400 for a majority, and contexts around 1000 concepts are hit three times.

3.5 Discussion of the Experiments

Throughout our experiments, we observed that the Stegosaurus phenomenon seems to be more associated with the actual algorithm of constructing the formal contexts than with any unknown correlation between the number of pseudo intents and the number of formal concepts. Also, the upper bound which was suggested by the phenomenon appears vacuous for a deeper understanding of the correlation in question.

In particular, the experiments concerning formal contexts with fixed row-density nourished our understanding what actually can be the reason for the original phenomenon. Since the algorithm in Section 3.1 uses a constant probability for generating crosses, the row density in a context does not vary much.

Indeed, with N attributes and a cross probability of p, the expected number of attributes per object is pN. Therefore, in most cases, the algorithm generates an "approximation" of a context with fixed row-density. If one imagines Figure 6 without the thin spikes, the result resembles a lot the one of Figure 2.

At this point we cannot explain the result of the SCGaz context generator with respect to our experimental setup. However, in Figure 7 we see in the overlay plot that the dales are artificial since spikes are running right through them.

The final investigation using real world data sets leads to the question if all discussed random context generators miss the point of creating contexts that behave like real world data, making them unsuitable for real-world benchmarking. For the BibSonomy data set one could still argue that Figure 8 resembles the very left part of Figure 2 and Figure 7. However, in the case of the IMDB data set, strange capping of the number of pseudo intents can be observed that does not appear in any of our approaches of randomly generating formal contexts.

4 Conclusions and Outlook

At his first discovery, the Stegosaurus phenomenon raised a lot of questions. Is it a programming error, is it a systematic error, is it a hint to enhance the understanding of canonical bases? At this point, we feel confident to state that it is "just" a systematic bias in generating the contexts. Therefore, benchmarking FCA-algorithms using random contexts created by the original algorithm seems unreasonable. The SCGaz generator can be tuned to generate more diverse samples. However, this tuning needs some effort and there is still some unaccounted bias. In any way, the question what a truly "random context" is and how it can be sampled remains open.

Recalling the results of the real world data sets, one can conclude that the idea of randomly generating test data for algorithms needs some reconsideration. Like simple random generated graphs in general do not resemble a social network graph, randomly generated contexts might not reproduce real world contexts. In the case of randomly generating social graphs, the method of *preferential attachment* led to better results [11]. Hence, random context generators trying to sample formal contexts with the characteristics of some class of real world contexts would be an improvement in the realm of random contexts.

Still, new algorithmic ideas need to be tested. Therefore, a set of specialized random context generators, as proposed by Kuznetsov [2], producing contexts of a particular class would be an improvement. On the other hand, a standard set of formal contexts to test against should be compiled as well. To this end, the authors have obtained the abandoned domain `fcarepository.com` to revive the idea of a central repository of formal contexts in the next months.

References

1. Ganter, B., Wille, R.: Formal Concept Analysis: Mathematical Foundations. Springer (1999)

2. Kuznetsov, S.O., Obiedkov, S.: Comparing performance of algorithms for generating concept lattices. Journal of Experimental and Theoretical Artificial Intelligence **14** (2002) 189–216
3. Babin, M.A., Kuznetsov, S.O.: Recognizing pseudo-intents is conp-complete. In: Proceedings of the 7th International Conference on Concept Lattices and Their Applications, Sevilla, Spain, October 19-21, 2010. (2010) 294–301
4. Bazhanov, K., Obiedkov, S.A.: Comparing performance of algorithms for generating the duquenne-guigues basis. In: Proceedings of The Eighth International Conference on Concept Lattices and Their Applications, Nancy, France, October 17-20, 2011. (2011) 43–57
5. Kriegel, F., Borchmann, D.: Nextclosures: Parallel computation of the canonical base. CLA 2015 (2015) 181
6. Borchmann, D.: Decomposing Finite Closure Operators by Attribute Exploration, University of Nicosia (May 2011)
7. Ganter, B.: Random extents and random closure systems. In Napoli, A., Vychodil, V., eds.: CLA. Volume 959 of CEUR Workshop Proceedings., CEUR-WS.org (2011) 309–318
8. Rimsa, A., Song, M.A.J., Zárate, L.E.: Scgaz - a synthetic formal context generator with density control for test and evaluation of fca algorithms. In: Systems, Man, and Cybernetics (SMC), 2013 IEEE International Conference on. (Oct 2013) 3464–3470
9. Ganter, B.: Two Basic Algorithms in Concept Analysis. In: Formal Concept Analysis: 8th International Conference, ICFCA 2010, Agadir, Morocco, March 15-18, 2010. Proceedings. Springer Berlin Heidelberg, Berlin, Heidelberg (2010) 312–340
10. Benz, D., Hotho, A., Jäschke, R., Krause, B., Mitzlaff, F., Schmitz, C., Stumme, G.: The social bookmark and publication management system bibsonomy. The VLDB Journal **19**(6) (December 2010) 849–875
11. Watts, D.J., Strogatz, S.H.: Collective dynamics of 'small-world' networks. Nature (393) (1998) 440–442

Approximate Seriation in Formal Concept Analysis

François Brucker and Pascal Préa

École Centrale Marseille,
LIF, Laboratoire d'Informatique Fondamentale de Marseille,
CNRS UMR 7279,
Marseille, France,
francois.brucker@lif.univ-mrs.fr, pascal.prea@lif.univ-mrs.fr

Abstract. In this paper we present a method (linear in the size of the formal context) to solve the seriation problem for formal contexts. We show that any maximal solution can be represented by a PQ-Tree. Moreover, the set of PQ-Trees can be seen as a distributive lattice. This lattice yields a consensus method which deals with the multiple solutions.

Keywords: seriation, PQ-Trees, Consecutive One's Property, lattice, consensus.

1 Introduction

The classical problem of *seriation* in Archeology [9] is the following: we are given a set of *objects* (different kinds of necklace, bracelet, dishes. . .) and a set of *sites* (tombs, houses,. . .). Each object has been used during an interval of time, and each site contains objects. The problem is to order the sites along time. More generally, the seriation problem consists in finding a linear order which underlies a data set, and this problem arises in genetics [2, 6], hypertext browsing [3], philology [4], data visualization [7, 10], musicology [8], . . . ; the order can be time, altitude, dispersion along a river, influence of an author, or even an unknown reason.

Formally speaking, a seriation problem instance can be represented by a formal context (G, M, I) where, for the classical problem in archeology, G is the set of sites, M the set of objects and I the relation which states if an object m has been found in site g.

In *exact seriation*, there exists a linear order (said *compatible*) such that A is an interval for any formal concept (A, B). The problem is to find one (or all the) compatible orders. Since the intersection of two intervals is also an interval, it is sufficient to check the inf-irreducible elements of the associated concept lattice, *i.e.* the columns of the formal context matrix. In this case, the formal context matrix $\mathcal{M}$ is said to have the *Consecutive One's Property* (*C1P*); that is the lines of $\mathcal{M}$ can be reordered in such a way that on every column of $\mathcal{M}$, the *1*s appear in consecutive order. Note tat the C1P is not a symmetrical property. Indeed,

© Marianne Huchard, Sergei O. Kuznetsov (Eds.): CLA 2016, pp. 71–82,
ISBN 978-5-600-01454-1, National Research University Higher School of Economics, 2016.

e.g. for the classical problem in archeology, the objects organize the sites into a linear order (the time), but the converse is false.

An optimal algorithm to find all the compatible orders was introduced in 1976 [5], based on a special data structure: *PQ-Trees*.

In *approximate seriation*, the data is not accurate (*e.g.* a site may not contain an object that was used when it was built); and the formal context matrix $\mathcal{M}$ does not have the C1P. We suppose (for extra reasons) that there exists a linear order which underlies the data set and the problem is to find one (or several) such order. There are two basic approaches for approximate seriation: *(i)* Minimally modify $\mathcal{M}$ so that it has the C1P (this yields to an NP-Hard problem); *(ii)* Find a maximal submatrix which has the C1P. In the formal concept framework, this approach consists in finding a sub-lattice of the formal concept lattice.

The aim of this paper is, following approach (*ii*), to present an efficient algorithm for approximate seriation, also based on PQ-Trees. This paper is organized as follows: in Section 2, we present the PQ-Tree structure and a first algorithm which follows approach *(ii)*. In Section 3, we show that the set of PQ-Trees can be organized as a lattice, which generalizes the semilattice of hierarchies; and we give a second algorithm which constructs a consensus between the possibly multiple solutions of the algorithm given in Section 2. In Section 4, we present a possible workflow of our method on an archeological data set. Actually, this workflow makes several runs of the algorithm of Section 3. The inputs of these runs strongly depends on the data and thus has to be decided by the user.

2 PQ-Trees

Given a finite set X, a *PQ-tree* T on X is a tree that represents a set of permutations on X denoted by S_T. The leaves of T are the elements of X, and the nodes of T are of two types : the *P-nodes* and the *Q-nodes*. We represent P-nodes by ellipses, and Q-nodes by rectangles.

On a P-node, one can apply any permutation of its children (equivalently, its children are not ordered). The children of a Q-node are ordered, and the only permutation we can apply on them is to reverse the order. For instance, the PQ-Tree of Figure 1 represents the set of permutations {(0,1,2,3,4,5), (0,1,3,2,4,5), (0,2,1,3,4,5), (0,2,3,1,4,5), (0,3,1,2,4,5), (0,3,2,1,4,5), (5,4,1,2,3,0), (5,4,1,3,2,0), (5,4,2,1,3,0), (5,4,2,3,1,0), (5,4,3,1,2,0), (5,4,3,2,1)}.

Let $\mathcal{M}$ be a formal context matrix. An order σ on the lines of $\mathcal{M}$ is *compatible* if, when the lines of $\mathcal{M}$ are sorted along σ, on each column of $\mathcal{M}$, the 1's are consecutive. If $\mathcal{M}$ has the Consecutive One's Property (*i.e.* if there exist compatible orders), the set of all compatible orders can be represented by a (unique) PQ-Tree. For instance, the cross-table of Figure 2 has the C1P and its compatible orders are represented by the PQ-Tree of Figure 1.

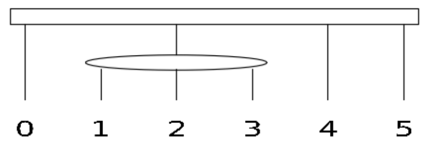


Fig. 1. A PQ-Tree

	0	1	2
0	×		
1	×	×	
2	×	×	
3	×	×	
4		×	×
5			×

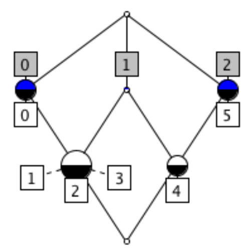


Fig. 2. Example of cross table (left) and its associated lattice (right).

Given a Formal Context $\mathcal{M}$ satisfying the C1P, the associated PQ-Tree is a condensed representation of the associated concept lattice: if we add to $\mathcal{M}$ columns which are nonempty intersections or non-disjoint unions of already existing columns, the associated PQ-Tree remains unchanged. This is why we will use PQ-Trees as a representative of concept lattices to solve seriation problems.

Generally, a formal context does not have the C1P, as that associated with the cross table of Figure 3. We can remark that, although this example was built by adding columns to the example of Figure 2, it is not easy to construct the concept lattice of Figure 2 directly from the one of Figure 3.

	0	1	2	3	4
0	×			×	
1	×	×			×
2	×	×			
3	×	×			×
4		×	×	×	×
5			×		

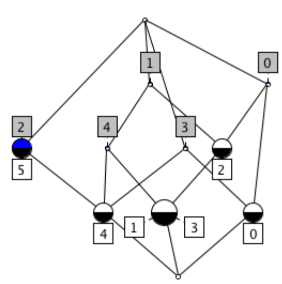
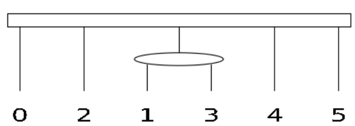


Fig. 3. Extension of the Cross Table 2 (left) and its associated lattice (right).

A first way to solve the approximate seriation problem consists in finding a maximal set of columns M' such that $\mathcal{M}_{|M'}$ has the C1P and exhibit the compatible orders. There exist several maximal sets of attributes having C1P; for instance, the cross-table of Figure 3 admits $\{0, 1, 2, 4\}$ (see Figure 4) or $\{1, 3, 4\}$ (see Figure 6). Remark that the concept lattices of Figures 4 and 6 are sublattices of the one of Figure 3. In addition, for each concept (A, B), A is an interval for any compatible order.

This is made possible by the incremental nature of the Booth and Lueker algorithm, which considers one column of the matrix at each step. In addition, starting with this maximal set M', it is easy to find the associated concepts: their extensions are the columns and the 2-intersections of columns (the intersection of three intervals is the intersection of two of them).

The Booth and Lueker algorithm [5] relies on a function UPDATE_TREE(T, A), where T is a PQ-Tree on X and A a subset of X. UPDATE_TREE returns a PQ-Tree T' where $S_{T'}$ is the set of all permutations σ of S_T such that, when X is sorted along σ, A is an interval of X (if there is no such permutations, T' is None); for instance, with the PQ-Tree of Figure 1 and the set $\{1, 3, 4\}$ (column 4 of Figure 3), UPDATE_TREE returns the PQ-Tree of Figure 4. UPDATE_TREE runs in $O(n)$, where n is the size of the column (*i.e.* the number of lines of the matrix). Given an $n \times m$ $\{0, 1\}$-matrix $\mathcal{M}$, the algorithm of Booth and Lueker



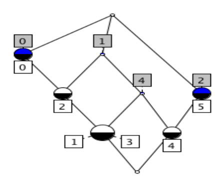


Fig. 4. PQ-Tree built from columns 0, 1, 2 and 4 of Cross-Table of Figure 3 (left) and the associated concept lattice (right).

starts with the PQ-Tree U_n which represents all permutations on $\{1, \ldots n\}$ (U_n has n leaves and one internal node (its root) which is a P-node) and apply UPDATE_TREE for all columns of $\mathcal{M}$. By this way, it determines if $\mathcal{M}$ has the C1P in $O(nm)$. Fo instance, applying this algorithm on the cross table of Figure 2, the algorithm runs as on Figure 5.

More generally, given a subset S of 2^X, we can apply the algorithm of Booth and Lueker on S and obtain a PQ-Tree $T = \mathcal{BL}(S)$ such that, for any permutation σ represented by T (and only for them), when X is sorted along σ, all the elements of S are intervals.

So, given a column order ζ, Algorithm MAXIMAL-C1P-CONSTRUCTION gives a solution to the approximate seriation problem in linear time. For the formal

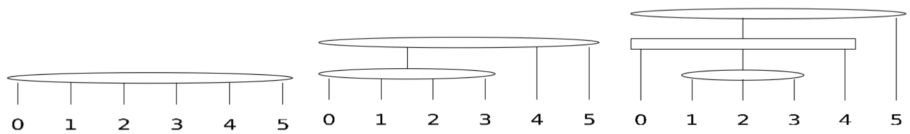


Fig. 5. The intermediate steps of the algorithm of Booth and Lueker, when applied on the cross table of Figure 2: the algorithm starts with the universal tree U_6 (left) and treats the set $\{0,1,2,3\}$ (*i.e.* it forces $\{0,1,2,3\}$ to be an interval); it then gets the PQ-Tree in the middle. Then it treats the set $\{1,2,3,4\}$ and gets the PQ-Tree on the right. By treating the set $\{4,5\}$, it gets the PQ-Tree of Figure 1.

context of Figure 3, if we consider the columns in increasing order, this algorithm returns the PQ-Tree of Figure 4 and rejects the column 3, which is not compatible with the 3 first columns.

Algorithm MAXIMAL-C1P-CONSTRUCTION(M, ζ)
Input A $n \times m$ $\{0,1\}$-matrix M.
  A permutation ζ on the columns of M.
Output A Maximal set C of columns of M such that $M_{|C}$ has C1P;
  A PQ-Tree T representing the compatible permutations.
begin
  $T \leftarrow U_n$ **;**
  $C \leftarrow \emptyset$ **;**
  ForAll *columns* c *of* M *taken along* ζ **Do**
    $T' \leftarrow$ UPDATE_TREE(T, c) **;**
    If $T' \neq$ **None Then**
      $T \leftarrow T'$ **;**
      $C \leftarrow C \cup \{c\}$ **;**
  return T, C **;**
end

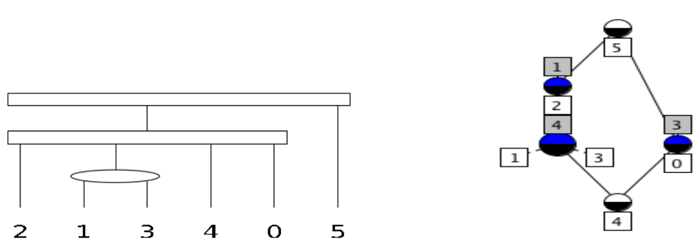


Fig. 6. PQ-Tree built from columns 1, 3 and 4 of Cross-Table of Figure 3 (left) and the associated concept lattice (right).

If the matrix has not the C1P, there are many solutions depending on the order ζ. For instance, if we consider the columns of the formal context of Figure 3 in reverse order, we keep the columns 4, 3, 1 and obtain the PQ-Tree of Figure 6.

We can see that the maximal sets of compatible columns do not have all the same number of elements. Since MAXIMAL-C1P-CONSTRUCTION is very efficient, it is possible to try many orders on the columns and then take the greatest obtained set. The problem is that there may exist many such sets. We will see in next Section that it is possible to go over that by using the lattice structure of PQ-Trees.

3 The Lattice Structure of PQ-Trees

We will show here that the PQ-Trees on a set X can be organized as a distributive lattice. This will allow us to build a consensus (by taking the join) of several PQ-Trees given by Algorithm MAXIMAL_C1P_CONSTRUCTION. For instance, the PQ-Tree of Figure 7 is the join of the PQ-Trees of Figures 4 and 6.

We denote by $\mathcal{T}_X$ the set of all PQ-Trees on a finite set X. Given two elements T_1 and T_2 of $\mathcal{T}_X$, we say that $T_1 \leq T_2$ if $S_{T_1} \subseteq S_{T_2}$. We will show that $(\mathcal{T}_X, \leq)$ is a distributive lattice, which generalizes the semilattice of hierarchies (a hierarchy can be seen as a PQ-Tree with only P-Nodes). This will allow us to define a consensus between the different solutions of MAXIMAL-C1P-CONSTRUCTION.

Given a PQ-Tree T on X, the *Interval Set* of T (denoted by *Int(T)*) is the set of all nonempty subsets S of X such that, for every permutation σ compatible with T, when X is sorted along σ, S is an interval, *i.e.* $Int(T)$ is the greatest subset P of $2^X \setminus \{\emptyset\}$ such that $\mathcal{BL}(P) = T$. Equivalently, $S \in Int(T) \iff$ UPDATE_TREE$(T, S) = T$.

Let α be a node, we denote by $X(\alpha)$ the set of the leaves under α. If α is a Q-node with sons (in this order) $\beta_1, \dots \beta_p$, we denote by $\widehat{X(\alpha)}$ the set $\{\bigcup_{k=i}^{j} X(\beta_k), 1 \leq i < j \leq p\}$ (remark that $\widehat{X(\alpha)}$ is a set of sets). We have:

Property 1 *$Int(T) = \{X(\alpha), \alpha$ node of $T\} \cup \bigcup_{\alpha \text{ Q-node of } T} \widehat{X(\alpha)}$.*

Proof. Let $I(T) = \{X(\alpha), \alpha \text{ node of } T\} \cup \bigcup_{\alpha \text{ Q-node of T}} \widehat{X(\alpha)}$. Clearly, for every permutation represented by T, all subsets of X in $I(T)$ are intervals. So $I(T) \subset Int(T)$.

Conversely, let S be a subset of X not in $I(T)$. We are in one of the following cases:

1. $\exists$ node α s.t. $X(\alpha) \cap S \neq \emptyset$, $X(\alpha) \not\subset S$, $S \not\subset X(\alpha)$.

2. $\exists$ P-node α, with sons $\beta_1, \beta_2, ..., \beta_p$, $p > 2$ s.t. $X(\beta_1) \subset S$, $X(\beta_2) \subset S$, $X(\beta_p) \not\subset S$ (actually, if not in case 1, $X(\beta_p) \cap S = \emptyset$).
3. $\exists$ Q-node α, with sons $\beta_1, \beta_2, ..., \beta_p$, $p > 2$ s.t. $\exists i < j < k$ with $X(\beta_i) \subset S$, $X(\beta_k) \subset S$ and $X(\beta_j) \not\subset S$.

Suppose that there exists an ordering $\sigma = (x_1, x_2, \ldots x_n)$ of X, compatible with T, such that S is a proper interval $x_i, \ldots x_j$ of X. In Case 1, we can suppose that $X(\alpha) = \{x_k, \ldots x_l\}$, with $i < k \leq j < l$. By reversing $X(\alpha)$, we get a compatible ordering of X for which S is not an interval. In Case 2, we can suppose that $X(\beta_1) = \{x_k, \ldots x_{k'}\}$, $X(\beta_3) = \{x_l, \ldots x_{l'}\}$ and $X(\beta_3) = \{x_m, \ldots x_{m'}\}$, with $i \leq k \leq k' < l \leq l' \leq j < m \leq m'$. By "exchanging" β_2 and β_3, we get a compatible permutation for which X is not an interval.

In Case 3, let $x \in X(\beta_i)$, $y \in X(\beta_j) \setminus S$ and $x \in X(\beta_k)$. For any compatible ordering of X, $x < y < z$, and thus X is not an interval. □

Clearly:

Property 2 $T_1 \leq T_2 \iff Int(T_2) \subseteq Int(T_1)$.

Theorem 1 $(\mathcal{T}_X, \leq)$ *is a distributive lattice.*

Proof. By Property 2: $T_1 \wedge T_2 = \mathcal{BL}(Int(T_1) \cup Int(T_2))$ and $T_1 \vee T_2 = \mathcal{BL}(Int(T_1) \cap Int(T_2))$.

In addition, if T_1, T_2 and T_3 are PQ-Trees, $Int(T_1) \cup (Int(T_2) \cap Int(T_3)) = (Int(T_1) \cap Int(T_2)) \cup (Int(T_1) \cap Int(T_3))$, *i.e.* $T_1 \wedge (T_2 \vee T_3) = (T_1 \wedge T_2) \vee (T_1 \wedge T_3)$; so $(\mathcal{T}_X, \leq)$ is a distributive lattice. □

In addition, by Property 1, $T_1 \vee T_2$ and $T_1 \wedge T_2$ can be computed in $O(n^3)$. Remark that, to compute $T_1 \wedge T_2$, we can use the sets $\{X(\beta_i) \cup X(\beta_{i+1}), 1 \leq i < p\}$ instead of $\widehat{X(\alpha)}$ for all Q-nodes α with sons $\beta_1, \ldots \beta_p$. Thus $T_1 \wedge T_2$ can be computed in $O(n^2)$.

The largest element of $(\mathcal{T}_X, \leq)$ is the universal tree $U_{|X|}$ which represents all the permutations on X and the smallest one is None which represents no permutation.

The join of the PQ-Trees of Figure 4 and 6 is represented on Figure 7. The PQ-Trees of Figure 4 and 6 represent respectively 4 and 8 permutations. Their join represents 16 permutations, which is very close to the theoretical minimum of 12, especially when compared to the 720 possible permutations on $\{0, \ldots, 5\}$. In addition, we can see that, for all the permutations represented by this PQ-Tree, the set $\{1, 2, 3, 4\}$ is ordered in $(2, 1, 3, 4), (2, 3, 1, 4), (4, 1, 3, 2)$ or $(4, 3, 1, 2)$, as for the two PQ-Trees of Figure 4 and 6.

Conversely, the meet of the two PQ-Trees of Figure 4 and 6 is None, since these two PQ-Trees are compatible with maximal sets of columns. This situation will occur with any two PQ-Trees obtained with Maximal-C1P-Construction: they are built from maximal sets of columns of $\mathcal{M}$ and thus the permutation sets that they represent are already minimal.

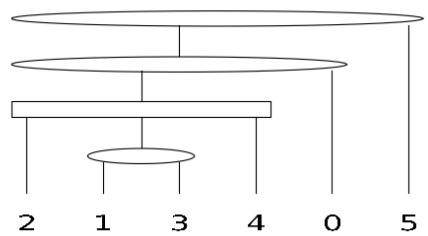


Fig. 7. The join of the PQ-Trees of Figure 4 and 6

We can now improve our algorithm by taking, from the best solutions obtained by MAXIMAL-C1P-CONSTRUCTION a consensus made of the join of these solutions. More precisely:

Algorithm APPROXIMATE_SERIATION(M, κ)
Input A $n \times m$ $\{0,1\}$-matrix M.
  A positive integer $\kappa < m$
  A positive integer Nb_Trials
Output A PQ-Tree T representing the compatible permutations.
  The set $\mathcal{C}$ of columns which have been taken into account.
begin
  $E \leftarrow \emptyset$ **;**
  $\mathcal{C} \leftarrow \emptyset$ **;**
  For $i \leftarrow 1$ **To** Nb_Trials **Do**
    $\zeta \leftarrow$ *random permutation on* $\{1, \dots, m$ **;**
    $(T, C) \leftarrow$ MAXIMAL-C1P-CONSTRUCTION(M, ζ) **;**
    If $Card(C) \geq \kappa$ **Then**
      $E \leftarrow E \cup \{T\}$ **;**
      $\mathcal{C} \leftarrow \mathcal{C} \cup C$ **;**
  $//E = \{T_{i_1}, T_{i_2}, \dots, T_{i_p}\}$
  return $T_{i_1} \vee T_{i_2} \vee \dots \vee T_{i_p}$**,** $\mathcal{C}$**;**
end

This algorithm runs in $O(Nb\_Trials \times n \cdot m + p \cdot n^3)$, where p is the number of column sets of size $\geq \kappa$ having the C1P. The result is a consensus of all "good" PQ-Trees, where "good" means that the PQ-Tree is built on at least κ columns of the matrix. The value of κ must be determined by the user and depends on the data. In next Section, we will apply our algorithm on a real data set and we will see how to choose κ. Moreover, we will see that the use of the algorithm may need other actions of the user.

4 Experimentations

We have experimented our method on a recent archeological data set which is shown in Table 1. This example shows an application where the seriation problem consists in finding trends in the data. In the original paper, the author uses principal component analysis to do that. Our method allows to achieve similar results for binary data, with PQ-Trees replacing principal axes.

The first step is to find maximal sets M' of columns such that the table induced by M' has the C1P. To do that, we have made 200 millions trials of MAXIMAL-C1P-CONSTRUCTION with random order of the columns.[1] We found 1563 maximal sets, of size going from 5 to 11. Their distribution is shown in Table 2.

Table 1. Cross table from Alberti[1]. The columns are indexed by object types and the lines by the huts of the Punta Milazzese (Aeolian Archipelago, Italy) settlement. We have indicated the presence/absence of objects in the different huts.

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
0	×								×		×	×	×	×	×							×	×	×	×					×	
1	×		×			×	×		×		×	×		×	×		×	×		×	×	×		×	×	×	×				
2	×				×	×	×		×		×	×		×			×	×		×	×	×		×	×	×	×	×	×		
3					×	×	×		×			×	×	×			×			×		×		×	×				×	×	
4	×					×	×		×		×	×	×									×			×	×	×		×		
5						×			×			×								×		×		×			×			×	×
6	×		×			×	×		×			×		×		×	×	×		×		×			×	×	×				×
7	×	×			×		×		×	×	×	×	×	×			×	×	×	×	×	×		×	×	×	×		×		
8	×			×			×	×	×	×	×	×		×		×	×	×	×	×	×	×	×	×	×	×	×			×	
9	×	×			×	×	×	×	×			×		×			×			×	×	×			×		×			×	
10	×				×	×	×	×	×	×		×									×	×			×		×				
11	×	×				×	×		×	×	×	×		×	×	×	×	×		×		×		×			×			×	
12	×				×					×		×	×	×			×			×					×						
13	×					×	×	×	×			×		×			×			×	×	×								×	×
14					×									×											×						
15							×		×					×					×			×			×		×			×	
16	×						×			×																				×	
17						×	×		×			×	×				×			×					×						
18	×					×			×	×		×		×					×	×	×	×		×	×					×	

Table 2. Size and number of maximal sets of columns from Table 1 having the C1P.

Number of columns	5	6	7	8	9	10	11
Number of maximal sets	1	28	294	505	514	209	12

At this step, we made a consensus between all the solutions with maximum number of columns, *i.e.* we make APPROXIMATE_SERIATION run with $\kappa = 11$. We obtained the PQ-Tree of Figure 8.

[1] 200 millions is very small when compared to the 31! possible orders of the column set, but actually, we made 20 series of 10 millions of trials. For each of these 20 series, no maximal set M' has been found after trial 50,000. In addition, the maximal sets were the same for the 20 series. It is thus reasonable to suppose that we have found all the maximal sets M'.

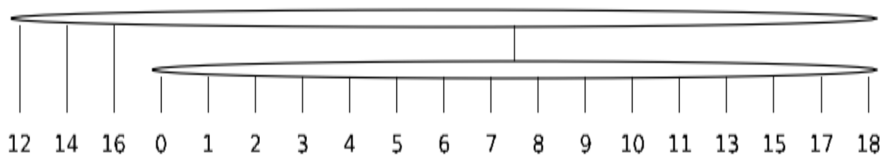


Fig. 8. Consensus PQ-Tree between the twelve maximal PQ-Trees built on 11 columns.

This PQ-Tree takes into account 22 columns, but it represents too many permutations (informally speaking, the corresponding consensus is too "soft"). In addition, since it corresponds to a consensus, we cannot build the associated lattice, but all the possible concepts are intervals of the PQ-Tree.

We can remark that all the lines/huts are grouped together except lines 12, 14 and 16. So we put these lines appart from the others (technically, we filled them with 0) and we determine the maximal sets of columns of the transformed table which have the C1P (in exactly the same way that for the complete table). We get the results of Table 3.

Table 3. Size and number of maximal sets of columns from Table 1 without lines 12, 14 and 16 having the C1P.

Number of columns	7	8	9	10	11	12	Total
Number of maximal sets	3	142	480	579	144	1	1349

The PQ-Tree built on the greatest maximal set of columns (the columns 2, 3, 4, 8, 10, 11, 14, 21, 22, 27, 28 and 30), and all lines except the lines 12, 14 and 16, is shown on Figure 9. Since it is unique, it corresponds to a maximal sub-context (having C1P) of Table 1, whose concept lattice is shown on Figure 10.

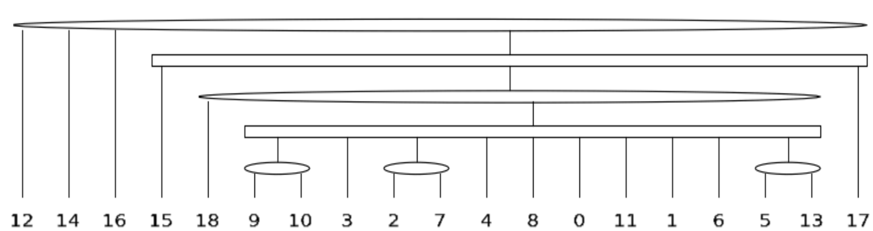


Fig. 9. The PQ-Tree built on 12 columns by putting appart lines 12, 14 and 16

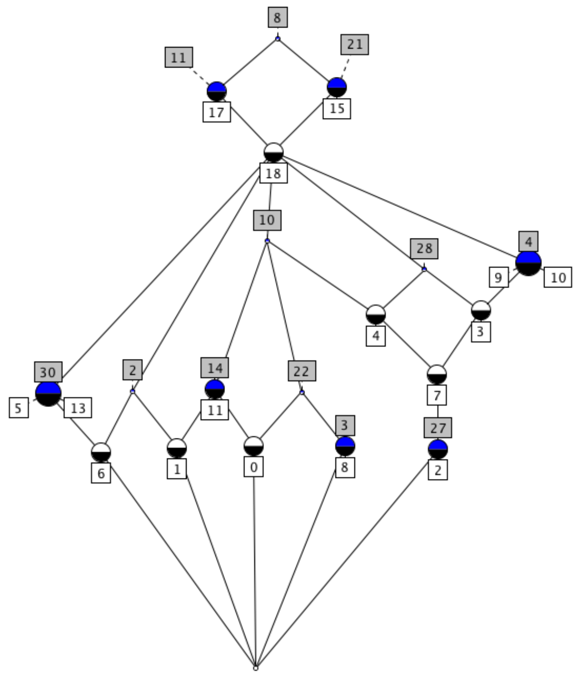


Fig. 10. Concept lattice from the PQ-Tree of Figure 9

This PQ-Tree takes into account only 12 columns, but it is possible to go further that solution. Since their is only one maximal column set of size 12 having C1P, we take the join of all PQ-Trees built on column sets of size ≥ 11 and we get the PQ-Tree of Figure 11. This PQ-Tree represents a lot of permutations, but we can see that lines 15 and 17 are appart from the others. So we put them appart and we get the results of Table 4. The consensus of the PQ-Trees corresponding with sets of size 13 is the one of Figure 12. This PQ-Tree takes into account 22 columns (the columns 0, 1, 2, 3, 6, 7, 8, 9, 11, 12, 13, 15, 16, 17, 18, 19, 21, 25, 26, 27, 28 and 29) and represents 2,985,984,000 permutations, which is 167,000 times less that the PQ-Tree of Figure 8.

Table 4. Size and number of maximal sets of columns from Table 1 without lines 12, 14, 15, 16 and 17 having the C1P.

Number of columns	8	9	10	11	12	13	Total
Number of maximal sets	1	68	405	377	90	10	951

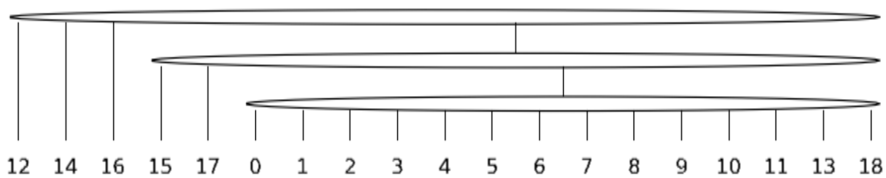


Fig. 11. Consensus PQ-Tree between the maximal PQ-Trees built on more than11 columns with lines 12, 14 and 16 appart.

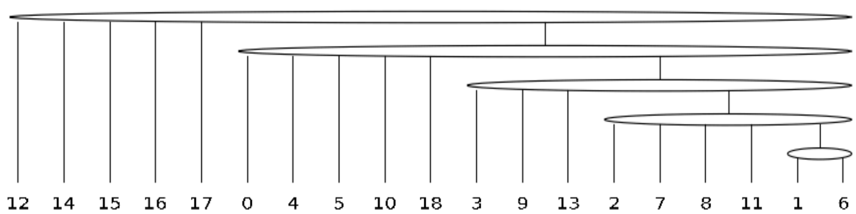


Fig. 12. Consensus PQ-Tree between the PQ-Trees built on 13 columns with lines 12, 14, 15, 16 and 17 appart.

We have seen that one can associate, with each formal context, maximal (in lines and columns) sub-contexts satisfying the C1P, that we could name *Seriation Formal Concepts*. The intersection of two seriation formal concepts $\mathcal{C}_1$ and $\mathcal{C}_2$ is a seriation formal concept (its PQ-Tree is the meet of the two PQ-Trees associated with $\mathcal{C}_1$ and $\mathcal{C}_2$). So, with any formal context, we can associate the semi-lattice of its seriation formal concepts. At the present time, we are able to determine all the seriation formal concepts containing a given set of lines. We are working on an algorithm which computes all the seriation formal concepts and generates the seriation formal lattice.

5 Conclusion

We have presented in this paper an interactive framework to solve the approximate seriation problem for formal contexts. More precisely, this framework uses some runs (3 for our example) of APPROXIMATE_SERIATION, which is in $O(Nb\_Trials \times n \cdot m + p \cdot n^3)$. We have used a very high value for Nb_Trials (up to 200 Millions, but 50,000 trials would have yield the same result). Moreover, since only the large column sets having the C1P are interesting for us, a small number of trials would have been sufficient (≈ 1000 in our case).

As usual in approximation problems, we are dealing with several criteria which are important to determine the quality of the resulting PQ-Tree:

– The number of columns taken into account (the largest possible).
– The number of removed lines (the smallest possible).
– The number of represented permutations (the smallest possible)
– The possibility to build a concept lattice from the solution.

If a formal context admits an exact solution to the seriation problem, then its underlying structure can be represented by a PQ-Tree. If it is not the case, we can build a consensus PQ-Tree which is a solution to the approximate seriation problem but at the present time, we are not able to build an associated concept lattice. Moreover, from this work appears the new notion of seriation formal concepts and semilattices.

References

1. Alberti G.: Making Sense of Contingency Tables in Archeology: the Aid of Correspondence Analysis to Intra-Site Activity Areas Research. Journal of Data Science 11, 479–499 (2013).
2. Benzer S.: The Fine Structure of the Gene. Scientific American 206:1, 70–84 (1962).
3. Berry M.W., Hendrickson B., Raghavan P.: Sparse Matrix Reordering Schemes for Browsing Hypertext. In The Mathematics of Numerical Analysis, J. Renegar, M. Shub and S. Smale Eds., 99–123, American Mathematical Society, Lectures in Applied Mathematics (1996).
4. Boneva L.: Seriation with Applications in Philology. In Mathematical Statistics, R. Bartoszyński, J. Koronacki and R. Zieliński Eds., 73–82, PWN Polish Scientific Publishers (1980).
5. Booth K.S., Lueker G.S.: Testing for the Consecutive Ones Property, Interval Graphs and Graph Planarity Using PQ-Tree Algorithm. Journal of Computer and System Sciences 13, 335–379 (1976).
6. Caraux G., Pinloche S.: PermutMatrix: a Graphical Environment to Arrange Gene Expression Profiles in Optimal Linear Order. Bioinformatics 21, 1280–1281 (2005).
7. Chen C.H., Hwu H.G., Jang W.J., Kao C.H., Tien Y.J., Tzeng S., Wu H.M.: Matrix Visualisation and Information Mining. COMPSTAT'2004, Prague (2004).
8. Halperin, D.: Musical Chronology by Seriation. Computer and the Humanities 28, 13–18 (1994).
9. Petrie, W.M.F.: Sequences in Prehistoric Remains. Journal of the Anthropological Institute of Great Britain and Ireland 29, 295–301 (1899).
10. Strehl, A., Ghosh, J.: Relationship-Based Clustering and Visualization for High-Dimensional Data Mining. INFORMS Journal on Computing 15, 208–230 (2003).

On Scaling of Fuzzy FCA to Pattern Structures

Aleksey Buzmakov[1] and Amedeo Napoli[2]

[1] National Research University Higher School of Economics, Perm, Russia
[2] LORIA (CNRS – Inria – University of Lorraine), Vandœuvre-lès-Nancy, France
avbuzmakov@hse.ru, amedeo.napoli@loria.fr

Abstract. FCA is a mathematical formalism having many applications in data mining and knowledge discovery. Originally it deals with binary data tables. However, there is a number of extensions that enrich standard FCA. In this paper we consider two important extensions: fuzzy FCA and pattern structures, and discuss the relation between them. In particular we introduce a scaling procedure that enables representing a fuzzy context as a pattern structure. Studying the relation between different extensions of FCA is of high importance, since it allows migrating methods from one extension to another. Moreover, it allows for more simple implementation of different extensions within a software.

Keywords: fuzzy FCA, pattern structures, scaling

1 Introduction

In this paper we deal with Formal Concept Analysis (FCA) and its extensions. FCA is a mathematical formalism having many applications in data mining and knowledge discovery. It starts from a binary table, a so-called formal context (G, M, I), where G is the set of objects, M is the set of attributes, and $I \subseteq G \times M$ is a relation between G and M, and proceeds to a lattice of formal concepts [1]. Fuzzy FCA is an extension of standard FCA that allows for fuzzy sets of objects and attributes in order to express uncertainty.

Pattern structures is another extension of FCA that allows processing complex data, e.g., graph or sequence datasets. It is a quite general framework and the question if fuzzy FCA can be represented within Pattern Structures and vice versa is still open. In this paper we make a step in this direction and study the connections between pattern structures and fuzzy FCA.

We show how a fuzzy context can be scaled to a “Minimum Pattern Structure” (MnPS), a special kind of pattern structures, that is close to interval pattern structures when considering numerical data. A scaling is needed, since pattern structures deal with crisp sets of objects and, thus, fuzzy extents cannot be expressed within the formalism of pattern structures. For such a kind of scaling we add new objects to the fuzzy context that express objects with uncertain membership in fuzzy sets, allowing expressing fuzzy sets of objects in the formalism of pattern structures. The resulting context is processed by MnPS. This kind of scaling is applicable to fuzzy FCA based on residuated lattices, a special kind of lattices expressing uncertain membership degrees in fuzzy sets.

© Marianne Huchard, Sergei O. Kuznetsov (Eds.): CLA 2016, pp. 85–96,
ISBN 978-5-600-01454-1, National Research University Higher School of Economics, 2016.

Table 1: A toy dataset of transactions for a supermaket and the related similarity matrix.

(a) A dataset with 5 transactions.

	i_1	i_2	i_3	i_4	i_5	i_6	i_7
t_1	x	x			x	x	x
t_2	x	x		x		x	x
t_3	x	x			x	x	
t_4		x	x				x
t_5	x			x	x	x	

(b) Similarity matrix.

	t_1	t_2	t_3	t_4	t_5
t_1	1.000	0.714	0.857	0.429	0.429
t_2	0.714	1.000	0.571	0.429	0.429
t_3	0.857	0.571	1.000	0.286	0.714
t_4	0.429	0.429	0.286	1.000	0.000
t_5	0.429	0.429	0.714	0.000	1.000

The rest of the paper is organized as follows. Section 2 describes a running example. Later, in Section 3 we introduce main definitions of fuzzy FCA and pattern structures. The main contribution of this paper is located in Section 4, where we introduce and discuss the scaling procedure of fuzzy FCA to pattern structures. Finally, at the end of the paper we discuss some related works.

2 Running Example

Let us consider a toy dataset of transactions within a supermarket. It is shown in Table 1a. Every row corresponds to a basket bought by a customer and every attribute corresponds to an item that can be bought in the supermarket. A cross in a cell (i, j) means that in the basket i there is the item j.

For making the example concrete, let us consider a clustering task. When dealing with clustering one typically needs a similarity or a distance measure. Such distance and similarity measures for the purpose of this example could be the fraction of different items shared by two baskets $\texttt{Dist}(t_1, t_2) = \frac{|t_1' + t_2'|}{|M|}$ and $\texttt{Sim}(t_1, t_2) = 1 - \texttt{Dist}(t_1, t_2)$, where operation '+' between sets is an exclusive OR (a so-called XOR or the symmetric difference, i.e., $A+B = (A \setminus B) \cup (B \setminus A)$). The similarity measure for any pair of transactions is shown in Table 1b. For example, similarity between t_1 and t_2 is equal 0.714. These baskets are different in two items i_4 and i_5. Thus $\texttt{Dist}(t_1, t_2) = \frac{2}{7} = 0.286$, where 7 is the number of items in the supermarket, and $\texttt{Sim}(t_1, t_2) = 1 - \texttt{Dist}(t_1, t_2) = 0.714$.

3 Definitions

Formal Concept Analysis (FCA) is a formalism for dealing with data mining and knowledge discovery tasks. It starts from a binary context (G, M, I), where G is the set of objects, M is the set of attributes and $I \subseteq G \times M$ is a relation between G and M. However, in real tasks such a binary encoding is too limited and several extensions of FCA were introduced. One of them is fuzzy FCA [2] that changes crisp sets into fuzzy sets. By doing so one is able to encode uncertainty. Another extension is pattern structures [3] dealing with crisp sets of objects but replacing binary sets of attributes with arbitrary descriptions allowing processing of many kinds of datasets, e.g., graph or sequential datasets. Below we discuss these two generalizations and their relation.

3.1 Fuzzy FCA

Fuzzy FCA works with fuzzy logic instead of crisp-logic, used in standard FCA. There are several generalizations of FCA to the fuzzy case [2]. Here the approach of Belohlavek is considered [4]. In fuzzy logic formulas can be valid up to a certain degree. It means that the formula can be completely valid, completely invalid, or between these two states. This fuzziness in fuzzy FCA is represented by a so-called residuated lattice, where the top of the lattice $\top$ corresponds to "completely valid" state of the logic and the bottom $\bot$ corresponds to "completely invalid" state.

Definition 1. *A Residuated Lattice is an algebra* $\boldsymbol{L} = \langle L, \vee, \wedge, \otimes, \rightsquigarrow, 0, 1\rangle$*, where* $\langle L, \vee, \wedge, 0, 1\rangle$ *is a complete lattice;* $\langle L, \otimes, 1\rangle$ *is a commutative monoid, i.e.* $\otimes$ *is commutative, associative, and* $\forall a(a \otimes 1 = 1 \otimes a = a)$*;* $\rightsquigarrow$ *and* $\otimes$ *form an adjoint pair, i.e.,* $a \otimes b \leq c \Leftrightarrow a \leq b \rightsquigarrow c$*.*

For the following, L refers to the set of elements of some residuated lattice and $\mathbf{L}$ for the residuated lattice itself. Such a lattice naturally appears when we want to introduce some degree of uncertainty. In the running example we introduced a similarity measure for any two objects. The values of the similarity can be considered as elements of the residuated lattice $[0, 1]$, a linear order of real numbers. The lattice operators $x \wedge y$ and $x \vee y$ are given by $\min(x, y)$ and $\max(x, y)$ correspondingly. The operations $\otimes$ and $\rightsquigarrow$ are "fuzzy conjunction" and "fuzzy implication".

An important residuated lattice based on a linearly ordered set is Göodel residuated lattice, which is used in examples of this paper. In Göodel residuated lattices the fuzzzy implication is defied as following:

$$a \rightsquigarrow b = \begin{cases} \top & a \leq b \\ b & a > b \end{cases} \tag{1}$$

In the crisp logic the implication $\top \rightarrow \bot$ is not valid, i.e., $\top \rightarrow \bot = \bot$, while other three possible implications are valid, i.e., $\top \rightarrow \top = \top$, $\bot \rightarrow \top = \top$, and $\bot \rightarrow \bot = \top$. The formula (1) generalizes this behavior. If the premise is less certain than the conclusion, then the implication is valid ($\top$), otherwise the validity of the implication is equal to the certainty of the conclusion.

In Definition 1 it is required that the fuzzy implication is *adjoint* (related) with an $\otimes$-operation. For Göodel residuated lattices the fuzzy implication is adjoint with $a \otimes b = \min(a, b)$. Indeed, $b \rightsquigarrow c \geq c$ according to (1). If $a \otimes b \leq c$, i.e., $\min(a, b) \leq c$ and $\min(a, b) = a$, then $a \leq c \leq b \rightsquigarrow c$. If $\min(a, b) = b$, then $b \leq c$ and $a \leq 1 = b \rightsquigarrow c$. Accordingly one can check that $a \leq b \rightsquigarrow c \Rightarrow a \otimes b \leq c$.

$\otimes$-operation is called fuzzy conjunction since it is a generalization of the crisp conjunction that is valid only if both arguments are valid, i.e., $\top \otimes \top = \top$.

A fuzzy dataset is encoded by means of a fuzzy context as defined below.

Definition 2. *A Fuzzy Relation between two sets X and Y is a function $I : X \times Y \to L$, for some residuated lattice $\boldsymbol{L}$.*

Definition 3. *A Fuzzy Context is a triple (X, Y, I) where X is a set of objects, Y is a set of attributes, I is a fuzzy relation, $I : X \times Y \to L$.*

Let us consider Table 1b as an example. It describes a fuzzy context where the set of attributes and the set of objects are the same. The cells of this table are similarity measures between objects. In this case the residuated lattice is formed as a linear order on similarity values.

Let us now define what is a fuzzy set, the next building block of fuzzy FCA.

Definition 4. *Given a crisp set X, a fuzzy set A is a function $A : X \to L$, mapping each element of the crisp set to an element of the residuated lattice. A fuzzy set is denoted as $\{{}^{l_i \in L}/_{x_i \in X}\}$, where $\bigcup x_i = X$, and for simplicity elements $A(x \in X) = \bot$ are omitted.*

For example, $\{{}^{1}/_{t_2}, {}^{0.571}/_{t_3}\}$ is a fuzzy set. Object t_2 belongs to this set entirely while object t_3 belongs only partially. The other items do not belong to this set, i.e., $A(g) = \bot$. Given our similarity measure we know that $1 \equiv \top$ and $0 \equiv \bot$.

In the fuzzy case of FCA one also defines Galois connections between a fuzzy set of objects $A : X \to L$ and a fuzzy set of attributes $B : Y \to L$.

Definition 5 (Derivation Operators). *Given a fuzzy context (X, Y, I), a fuzzy set of objects $A : X \to L$, a fuzzy set of attributes $B : Y \to L$, the fuzzy membership for object $x \in X$ and for attribute $y \in Y$ in the corresponding sets $A^\uparrow$ and $B^\downarrow$ are as follows:*

$$A^\uparrow(y) = \bigwedge_{\forall x \in X} (A(x) \rightsquigarrow I(x, y))$$

$$B^\downarrow(x) = \bigwedge_{\forall y \in Y} (B(y) \rightsquigarrow I(x, y))$$

Let us illustrate the derivation operators. First let us introduce the fuzzy set of interest $A = \{{}^{1.0}/_{t_1}, {}^{1.0}/_{t_2}\}$ (all other objects do not belong to this set, i.e., they have "0" membership in that set). Then we would like to find $A^\uparrow$. To compute $A^\uparrow$ we should compute $A(x) \rightsquigarrow I(x, y)$ for every object $x \in X$ and for every attribute $y \in Y$ (in the example we have a particular case where $X \equiv Y$). According to our set of interest $A(x)$ is either 1 or 0. For the function $I(x, y)$, the first argument varies from row to row and the second argument varies from column to column. The result of these operations is shown in Table 2. For example, the result of computing $A(t_1) \rightsquigarrow I(t_1, t_2)$ is given in the first row second column cell. The last row of that table shows the resulting fuzzy set of

Table 2: Example of computations for $\{^{1.0}/_{t_1},^{1.0}/_{t_2}\}^{\uparrow}$.

y	t_1	t_2	t_3	t_4	t_5
$(A(t_1)=1) \rightsquigarrow I(t_1,y)$	1.000	0.714	0.857	0.429	0.429
$(A(t_2)=1) \rightsquigarrow I(t_2,y)$	0.714	1.000	0.571	0.429	0.429
$(A(t_3)=0) \rightsquigarrow I(t_3,y)$	1	1	1	1	1
$(A(t_4)=0 \rightsquigarrow I(t_4,y)$	1	1	1	1	1
$(A(t_5)=0) \rightsquigarrow I(t_5,y)$	1	1	1	1	1
$\{^{1.0}/_{t_1},^{1.0}/_{t_2}\}^{\uparrow}$	0.714	0.714	0.571	0.429	0.429

Table 3: Example of computations for $\{^{1.0}/_{t_1},^{1.0}/_{t_2}\}^{\uparrow\downarrow}$.

	$0.714 \rightsquigarrow I(x,t_1)$	$0.714 \rightsquigarrow I(x,t_2)$	$0.571 \rightsquigarrow I(x,t_3)$	$0.429 \rightsquigarrow I(x,t_4)$	$0.429 \rightsquigarrow I(x,t_5)$	$\{^{1.0}/_{t_1},^{1.0}/_{t_2}\}^{\uparrow\downarrow}$
t_1	1	1	1	1	1	1
t_2	1	1	1	1	1	1
t_3	1	0.571	1	0.286	1	0.286
t_4	0.429	0.429	0.286	1	0	0
t_5	0.429	0.429	1	0	1	0

the attributes (more precisely their membership degree). The value of the last row can be computed by applying $\wedge$ operator of the residuated lattice to the whole column, and in the case of our example $\wedge$ corresponds to the minimum. As the result of $A^{\uparrow}$ we obtained

$$B = A^{\uparrow} = \{^{0.714}/_{t_1},^{0.714}/_{t_2},^{0.571}/_{t_3},^{0.429}/_{t_4},^{0.429}/_{t_5}\}.$$

Let us now apply the derivation operator to the set $A^{\uparrow}$. To find $B^{\downarrow} = A^{\uparrow\downarrow}$ we need to compute $B(x) \rightsquigarrow I(x,y)$. The result of these operations is shown in Table 3 and should be read in the same way as Table 2. The last column corresponds to the result of $B^{\downarrow}$. Then we have:

$$\{^{1.0}/_{t_1},^{1.0}/_{t_2}\}^{\uparrow\downarrow} = \{^{1.0}/_{t_1},^{1.0}/_{t_2},^{0.286}/_{t_3}\}$$

It can be checked that $A^{\uparrow\downarrow\uparrow} = B$ and $A^{\uparrow\downarrow\uparrow\downarrow} = B^{\downarrow} = A^{\uparrow\downarrow}$. These properties of the derivation operators defines fuzzy concepts [2].

Definition 6. *A fuzzy concept is a pair (A,B), where A is a fuzzy set of objects, $A : X \to L$ and B is a fuzzy set of attributes $B : Y \to L$, such that $A^{\uparrow} = B$ and $A = B^{\downarrow}$.*

In particular in the previous example we have found the concept:

$$\left(\{^{1.0}/_{t_1},^{1.0}/_{t_2},^{0.286}/_{t_3}\},\{^{0.714}/_{t_1},^{0.714}/_{t_2},^{0.571}/_{t_3},^{0.429}/_{t_4},^{0.429}/_{t_5}\}\right). \quad (2)$$

This concept can be interpreted in the following way. Every object from the extent is similar (with a certain degree of confidence) to an object from the intent by at least the corresponding membership degree, e.g., object t_1 is similar to object t_2 with confidence 1 (taken from the left membership degree of t_1) by at least 0.714 (taken form the right membership degree of t_2), while object t_3 is similar with t_2 by at least 0.714 with confidence only 0.286. In particular if on the extent side we consider objects with membership degree of 1 ($\top$), e.g., t_1 and t_2 then we would find the similarity of all these objects to the rest of the objects, thus, providing a good description for a cluster around these objects.

The set of fuzzy concepts is ordered such that $(A, B) \leq (X, Y)$ iff $A \subseteq X$ (or dually $B \supseteq Y$) forming a complete lattice, called *fuzzy concept lattice*.

3.2 Pattern Structures

A concept lattice $\mathfrak{L}(G, M, I)$ is constructed from a (binary) formal context (G, M, I) [1]. For non-binary data, such as sequences or graphs, lattices can be constructed in the same way using pattern structures [3].

Definition 7. *A pattern structure $\mathbb{P}$ is a triple $(G, (D, \sqcap), \delta)$, where G, D are sets, called the set of objects and the set of descriptions, and $\delta : G \to D$ maps an object to a description. Respectively, $(D, \sqcap)$ is a meet-semilattice on D w.r.t. $\sqcap$, called similarity operation such that $\delta(G) := \{\delta(g) \mid g \in G\}$ generates a complete subsemilattice $(D_\delta, \sqcap)$ of $(D, \sqcap)$.*

For illustration, let us represent standard FCA in terms of pattern structures. The set of objects G is preserved, the semilattice of descriptions is $(\wp(M), \cap)$, where $\wp(M)$ denotes the powerset of the set of attributes M, a description is a subset of attributes and $\cap$ is the set-theoretic intersection. If $x = \{a, b, c\}$ and $y = \{a, c, d\}$ then $x \sqcap y = x \cap y = \{a, c\}$, and $\delta : G \to \wp(M)$ is given by $\delta(g) = \{m \in M \mid (g, m) \in I\}$.

Derivation operator for a pattern structure $(G, (D, \sqcap), \delta)$, relating sets of objects and descriptions, is defined as follows:

$$A^\diamond := \bigsqcap_{g \in A} \delta(g), \qquad \text{for } A \subseteq G$$

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

Given a subset of objects A, $A^\diamond$ returns the description which is common to all objects in A. Given a description d, $d^\diamond$ is the set of all objects whose description subsumes d. The natural partial order (or subsumption order between descriptions) $\sqsubseteq$ on D is defined w.r.t. the similarity operation $\sqcap$: $c \sqsubseteq d \Leftrightarrow c \sqcap d = c$ (in this case we say that c is subsumed by d). In the case of standard FCA the natural partial order corresponds to the set-theoretical inclusion order, i.e., for two sets of attributes x and y, $x \sqsubseteq y \Leftrightarrow x \subseteq y$.

Definition 8. *A pattern concept of a pattern structure $(G, (D, \sqcap), \delta)$ is a pair (A, d), where $A \subseteq G$ and $d \in D$ such that $A^\diamond = d$ and $d^\diamond = A$; A is called the pattern extent and d is called the pattern intent.*

As in standard FCA, a pattern concept corresponds to the maximal set of objects A whose description subsumes the description d, where d is the maximal common description of objects in A. The set of all pattern concepts is partially ordered w.r.t. inclusion of extents or, dually, w.r.t. subsumption of pattern intents within a concept lattice, these two anti-isomorphic orders form a lattice, called pattern lattice.

Let us return to the example in Table 1b. Let us consider a special case of pattern structures, a so-called Minimum Pattern Structure (MnPS), that is close to interval pattern structures [5]. MnPS is based on the minimum of two numbers as the similarity operation rather than on the convex hull of two intervals. We will show that MnPS is well adapted for formalizing fuzzy FCA within the framework of pattern structures.

In Table 1b we have the set G as both, a set of objects and a set of attributes. Let us first consider only one attribute. Then the set of descriptions D is just the interval $[0,1]$ of real numbers and the similarity operation between two descriptions (numbers) is the minimum. When there are several attributes, the set of descriptions is just an element of $\mathbb{R}^{|N|}$, where $\mathbb{R}$ is the set of real numbers and N is the set of numerical attributes.

In particular, in our example the set of objects is G. The set D of descriptions is $\mathbb{R}^5$, since we have 5 numerical attributes. The mapping function δ is given in Table 1b, e.g., $\delta(t_2) = \langle 0.714, 1, 0.571, 0.429, 0.429\rangle$. The similarity operation is the component-wise minimum, e.g., the similarity between descriptions of t_2 and t_3 is given by

$$\begin{aligned}
\{t_2\}^\diamond \sqcap \{t_3\}^\diamond &= \\
&= \langle 0.714, 1, 0.571, 0.429, 0.429\rangle \sqcap \langle 0.857, 0.571, 1, 0.286, 0.714\rangle = \\
&= \langle \min(0.714, 0.857), \min(1, 0.571), \\
&\quad \min(0.571, 1), \min(0.429, 0.286), \min(0.429, 0.714)\rangle \\
&= \langle 0.714, 0.571, 0.571, 0.286, 0.429\rangle
\end{aligned}$$

4 From Fuzzy FCA to Pattern Structures with Scaling

Let us now discuss a possible connection between fuzzy FCA and pattern structures. A certain connection was already proposed in [6]. In particular, every *crisply* closed subset of objects is an extent of an interval pattern structure. Here, *crisply closed subset of objects* means that the fuzzy closure of this set contains no additional objects g with a membership degree coinciding with the top of the residuated lattice, i.e., $A(g) = \top$.

Here, we discuss a loss-less scaling from a fuzzy formal context (X, Y, I) to a pattern structure, that allows a more efficient processing than the loss-less scaling to crisp formal context and highlights another connection between fuzzy FCA and pattern structures.

Since pattern structures can deal with any kind of descriptions, they should take into account fuzziness on the intent side. However, for the extent side it

Table 4: Scaling of the fuzzy context from table Table 1b to a number-minimum pattern structure.

	t_1	t_2	t_3	t_4	t_5
$\langle t_1, 1.000\rangle$	1.000	0.714	0.857	0.429	0.429
$\langle t_1, 0.857\rangle$	1.000	0.714	1.000	0.429	0.429
$\langle t_1, 0.714\rangle$	1.000	1.000	1.000	0.429	0.429
...					
$\langle t_2, 1.000\rangle$	0.714	1.000	0.571	0.429	0.429
...					
$\langle t_2, 0.571\rangle$	1.000	1.000	1.000	0.429	0.429
...					
$\langle t_3, 1.000\rangle$	0.857	0.571	1.000	0.286	0.714
...					

	t_1	t_2	t_3	t_4	t_5
...					
$\langle t_3, 0.429\rangle$	1.000	1.000	1.000	0.286	1.000
...					
$\langle t_4, 1.000\rangle$	0.429	0.429	0.286	1.000	0.000
...					
$\langle t_4, 0.286\rangle$	1.000	1.000	1.000	1.000	0.000
...					
$\langle t_5, 1.000\rangle$	0.429	0.429	0.714	0.000	1.000
...					
$\langle t_5, 0.286\rangle$	1.000	1.000	1.000	1.000	0.000

is not so straightforward, since pattern structures deal only with crisp sets of objects. Accordingly, we should somehow "scale" object sets, in order to express fuzzy sets of objects.

4.1 On expressing Fuzziness on the Extent Side of Pattern Structures

A natural way is to scale the object set X from a fuzzy context (X, Y, I) by substituting it with the direct product of the crisp set of objects and the residuated lattice (the degrees of confidence), $X \times L$. For every scaled object from this new set, we should compute a description. Let us consider the scaled description for the pair $\langle x, l\rangle$, where $x \in X$ is an object and $l \in L$ is the membership degree of this object. The description of this element should correspond to the description of the fuzzy set $\{{}^{l}/_{x}\}$, since $\langle x, l\rangle$ is "a model of" this fuzzy set.

The derivation operator $\{{}^{l}/_{x}\}^{\uparrow}(y \in Y) = \{{}^{l}/_{x}\}(x) \rightsquigarrow I(x,y) = l \rightsquigarrow I(x,y)$ gives the description of the element $\langle x, l\rangle$ and allows computing the fuzzy relation $\tilde{I}$ between $X \times L$ and Y.

Let us return to our example. Let T be the set of transaction IDs. The scaled fuzzy context is partially shown in Table 4. It consist of $|T| \cdot |L| = 5 \cdot 7 = 35$ objects, 5 attributes and the fuzzy relation between them. Every subset of objects corresponds to a fuzzy set of objects by joining corresponding fuzzy representation for every object. This is made precise in the next subsection.

4.2 Relation between fuzzy and pattern extents and intents

Let (X, Y, I) be a fuzzy context with a residuated lattice $\mathbf{L}$ and (G, D, δ) be a pattern structure, where G is the scaled set of objects $G = X \times L$. Let us formally define the correspondence between fuzzy sets of objects and scaled sets of objects.

Definition 9 (Object sets equivalence). *A fuzzy object set $A : X \to L$ is equivalent to a scaled object set $N \subseteq G$, denoted as $A \sim N$, when*

$$(\forall \langle x, l\rangle \in G)(A(x) \geq l \Leftrightarrow \langle x, l\rangle \in N)$$

Then object sets are equivalent when all scaled objects with membership degree smaller than or equal to $A(x)$ (w.r.t. the residuated lattice) are present in the scaled object set. For example, the fuzzy set $\{^{0.286}/_{t_1}, ^{0.429}/_{t_4}\}$ is equivalent to the scaled set $\{\langle t_1, 0.286\rangle, \langle t_4, 0.429\rangle, \langle t_4, 0.286\rangle\}$[3], where $\langle g, l\rangle \in X \times L$ is an element of the direct product of the set of objects and the residuated lattice.

Given a scaled fuzzy context $(X \times L, Y, \tilde{I})$ we can process it as a minimum pattern structure $(X \times L, D, \delta)$, where $D = L^{|Y|}$ is a tuple of elements from the residuated lattice $\mathbf{L}$ and the semilattice operation is given by the component-wise infimum of $\mathbf{L}$. In particular, we have discussed that for the numerical case, the similarity operation is the component-wise minimum. Indeed, fuzziness on the extent side is expressed by means of scaled object sets, and fuzziness on the intent side is directly processed by the pattern structure. Let us discuss the correspondence between fuzzy intents and patterns.

Definition 10. *A fuzzy attribute set $B : Y \to L$ is equivalent to a pattern $d \in D$, written as $B \sim d$, iff $(\forall y \in Y)(B(y) = d(y))$, where $d(y)$ is the value of the tuple d corresponding to the attribute y.*

A fuzzy attribute set B is equivalent to a pattern d iff for any attribute $y \in Y$, the membership degree $B(y)$ in the fuzzy set is equal to the value in the pattern tuple in the position corresponding to the attribute y, e.g., the pattern $\langle 0.5, 0.7\rangle$ corresponds to the fuzzy set $\{^{0.5}/_{y_1}, ^{0.7}/_{y_2}\}$.

It should be noticed that the definition of equality between fuzzy sets of attributes and patterns is a bijection, while there are scaled sets of objects that have no equivalent fuzzy set of objects. Indeed, there is no equivalent fuzzy set to the scaled set $\{\langle t_1, 0.286\rangle, \langle t_4, 0.429\rangle\}$, since according to Definition 9 all $\langle x, l\rangle$ such that $A(x) \geq l$ should be in this set. And since we have $\langle t_4, 0.429\rangle$ in this set, we should also have $\langle t_4, 0.286\rangle$ in the set. We can notice here that in our particular example the residuated lattice has only the element 0.286 that is smaller than 0.429. By contrast, if we take the real interval $[0, 1]$, then all points smaller than 0.429 should be added to the scaled set.

Let us define equivalence classes of scaled sets of objects in order to have a bijection between the equivalence classes and the fuzzy sets of objects.

Definition 11. *A scaled object set $N \subseteq G$ is* complete *iff a scaled object $\langle x \in X, l \in L\rangle$ belongs to N, then $(\forall l^* \in L, l^* \leq l)\langle x, l^*\rangle \in N$.*

It can be checked that for any scaled object set $N \subseteq G$ there is only one minimal complete superset of N. Let us denote this complete set by $\phi(N)$.

For example, the set $N = \{\langle t_1, 0.286\rangle, \langle t_4, 0.429\rangle\}$ is not complete, since the scaled object $\langle t_4, 0.286\rangle$ is not in N.

By contrast, $N_c = \phi(N) = \{\langle t_1, 0.286\rangle, \langle t_4, 0.429\rangle, \langle t_4, 0.286\rangle\}$ is complete. Moreover, it can be seen that this set is equivalent to $\{^{0.286}/_{t_1}, ^{0.429}/_{t_4}\}$ according to Definition 9. Furthermore, it can be checked, that any complete scaled set of objects is equivalent to a fuzzy set and accordingly the function $\phi(\cdot)$ defines the required equivalence classes.

[3] We notice that $\langle t_4, 0.429\rangle$ and $\langle t_4, 0.286\rangle$ are two different scaled objects.

4.3 Isomorphism of fuzzy and pattern lattices

In this subsection we show that our scaling procedure is correct. And the resulting pattern lattice and the fuzzy lattice are isomorphic. Moreover, the extents and intents of these lattices are connected by means of Definitions 9 and 10. The first lemma (a standard property of residuated lattices) shows that fuzzy implications are related if their premises are comparable.

Lemma 1 *If there are $l_1, l_2, l \in L$ such that $l_1 \leq l_2$ then*

$$l_1 \rightsquigarrow l \geq l_2 \rightsquigarrow l$$

Proof. Let $l_2 \rightsquigarrow l = r$ then according to Def. 1:

$$(\forall f \in \mathbf{L}, f \leq r)(f \otimes l_2 \leq l) \Leftrightarrow (\forall f \leq r)(l_2 \otimes f \leq l)$$
$$\Leftrightarrow (\forall f \leq r)(f \rightsquigarrow l \geq l_2 \geq l_1) \Leftrightarrow (\forall f \leq r)(l_1 \rightsquigarrow l \geq f) \Rightarrow l_1 \rightsquigarrow l \geq r.$$

Let us now show that starting from two (fuzzy and scaled) equivalent sets of objects the resulting descriptions are also equivalent.

Lemma 2 *Given a fuzzy set of objects $A : X \to L$ and a scaled set of objects $N \subseteq G$, such that $A \sim \phi(N)$, we have $A^{\uparrow} \sim N^{\diamond}$.*

Proof. Consider the value of the pattern tuple $N^{\diamond}$ corresponding to an attribute y: $N^{\diamond}(y) = \big(\bigsqcap_{\forall g \in N} \delta(g)\big)(y)$. The semilattice operation of the minimum pattern structure corresponds to the infimum in the residuated lattice:

$$\begin{aligned} N^{\diamond}(y) &= \bigwedge_{\forall \langle x,l\rangle \in N} \tilde{I}(\langle x,l\rangle, y) = \bigwedge_{\forall \langle x,l\rangle \in N} l \rightsquigarrow I(x,y) = \\ &= \Big(\bigwedge_{\forall x \in X} A(x) \rightsquigarrow I(x,y)\Big) \wedge \Big(\bigwedge_{\forall \langle x,l\rangle \in N : l < A(x)} l \rightsquigarrow I(x,y)\Big) = \\ &=_{\text{Lemma 1}} \Big(\bigwedge_{\forall x \in X} A(x) \rightsquigarrow I(x,y)\Big) = A^{\uparrow}(y). \end{aligned}$$

Finally let us show, that starting from equivalent fuzzy set of attributes and pattern, the sets of objects given by the derivation operators are also equivalent.

Lemma 3 *Given a fuzzy set of attributes $B : Y \to L$ and a pattern $d \in D$, such that $B \sim d$, we have $B^{\downarrow} \sim d^{\diamond}$.*

Proof. Let us study when object $\langle x \in X, l \in L\rangle$ can be included into $d^{\diamond}$.

$$\begin{aligned} \langle x \in X, l \in L\rangle \in d^{\diamond} &\Leftrightarrow \delta(\langle x,l\rangle) \sqsupseteq d \Leftrightarrow (\forall y \in Y)(\delta(\langle x,l\rangle)(y) \geq d(y)) \\ &\Leftrightarrow (\forall y \in Y)(l \rightsquigarrow I(x,y) \geq d(y)) \\ &\Leftrightarrow (\forall y \in Y)(d(y) \otimes l \leq I(x,y)) \Leftrightarrow (\forall y \in Y)(l \otimes d(y) \leq I(x,y)) \\ &\Leftrightarrow (\forall y \in Y)(d(y) \rightsquigarrow I(x,y) \geq l) \\ &\Leftrightarrow (\forall y \in Y)(B(y) \rightsquigarrow I(x,y) \geq l) \\ &\Leftrightarrow l \leq \bigwedge_{\forall y \in Y} B(y) \rightsquigarrow I(x,y) \\ &\Leftrightarrow l \leq B^{\downarrow}(x) \end{aligned}$$

Thus an object $\langle x, l\rangle \in G$ is included in $d^{\diamond}$ iff $B^{\downarrow}(x) \geq l$ which is the definition of the equality of a fuzzy set of objects and a scaled set of objects.

Theorem 1 *The fuzzy lattice $\mathfrak{L}_f$ corresponding to the context (X, Y, I) and the pattern lattice $\mathfrak{L}_p$ corresponding to the pattern structure $(G, \underline{D}, \delta)$, where $G = X \times L$, $D = L^{|Y|}$ with component-wise minimum as the semilattice operation, and $\delta(\langle x \in X, l \in L\rangle)(y) = l \rightsquigarrow I(x, y)$ are isomorphic. The extents and intents of the corresponding concepts are equivalent.*

Proof. Let us show, that for any concept in one lattice there is a concept in the other lattice with equivalent extents and intents. Lemmas 2 and 3 are symmetric w.r.t. the type of extents and intents. Accordingly, we can just denote by $\mathfrak{L}_1$ and $\mathfrak{L}_2$ fuzzy and pattern lattices and prove the theorem in both directions. If we take an intent i_1 from $\mathfrak{L}_1$, we can always find an equivalent pattern p (for simplicity, fuzzy set of attributes is also referred as a pattern). Applying appropriate derivation operators to i_1 and p we get equivalent sets of objects according to Lemma 3. Both sets are closed and are extents of $\mathfrak{L}_1$ and $\mathfrak{L}_2$. Applying derivation operators to the extents we get equivalent intents according to Lemma 2. Thus, for any concept of $\mathfrak{L}_1$ there is an equivalent concept in $\mathfrak{L}_2$ and *vice versa*.

4.4 Application of the Theorem

Let us demonstrate how the theorem works in the running example. The proof is based on search of concepts with equivalent extents and intents. Let us find the scaled concept corresponding to the fuzzy concept (2). In the theorem we start from the intent. It can be seen that

$$\{^{0.714}/_{t_1}, ^{0.714}/_{t_2}, ^{0.571}/_{t_3}, ^{0.429}/_{t_4}, ^{0.429}/_{t_5}\} \sim \langle 0.714, 0.714, 0.571, 0.429, 0.429\rangle. \quad (3)$$

For the moment we are not sure that the pattern on the right side is an intent. Accordingly we apply derivation operators to the left and right hand sides and according to Lemma 3 the resulting object sets should be equivalent. Indeed,

$$\{^{1}/_{t_1}, ^{1}/_{t_2}, ^{0.286}/_{t_3}\} \sim$$
$$\sim \{\langle t_1, 1\rangle, \langle t_1, 0.857\rangle, \ldots, \langle t_1, 0.286\rangle, \langle t_2, 1\rangle, \ldots \langle t_2, 0.286\rangle, \langle t_3, 0.286\rangle\}.$$

On the left side we have the extent of the concept, while on the right side we have a closed scaled set of objects, since the result of the derivation operator is always closed. If we apply the derivation operators to these two sets of objects, we have equivalent patterns according to Lemma 2. In fact we have exactly the patterns from (3). Thus, we have found the scaled concept corresponding to the fuzzy concept. Similarly, we can start from a scaled concept and find the corresponding fuzzy concept.

5 Discussion and Conclusion

In this paper we highlighted the relation between fuzzy FCA and pattern structures. Our result is related to the work of [6]. Indeed, the authors have shown that extents of crisply closed fuzzy concepts are also closed in the interval pattern structure. In our work, we used the Minimum Pattern Structure that can be considered as a projection of the interval pattern structure. Indeed, let us consider the following component-wise projection. If $[a, b]$ is an interval, than the projection $\psi([a, b]) = [a, +\inf]$ changes the IPS to the MnPS. Accordingly the set of extents of the MnPS is the subset of the extents of the IPS. However, in our work we have shown, that the MnPS lattice is isomorphic to a fuzzy lattice under the scaling. It can be seen, that if we do not apply the scaling we generate exactly the lattice of the crisply generated fuzzy concepts. And this set of concepts is the subset of the concepts of the corresponding IPS.

The introduced scaling procedure can be useful, first, for migrating results between pattern structure community and fuzzy FCA community, and, second, for efficient implementation of software dealing with both pattern structures and fuzzy FCA at the same time.

Finally, we notice that such a work naturally raises (as it was already mentioned in [6]) the question of a “two-sided” pattern structure as a generalization of both pattern structures and fuzzy FCA. Some suggestions going in this directions can be found in the work of Soldano et al. [7], where the authors discussed projections applied to the extent side.

References

1. Ganter, B., Wille, R.: Formal Concept Analysis: Mathematical Foundations. 1st edn. Springer (1999)
2. Belohlavek, R.: What is a fuzzy concept lattice? II. In: Rough Sets, Fuzzy Sets, Data Min. Granul. Comput. Springer Berlin / Heidelberg (2011) 19–26
3. 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
4. Belohlávek, R.: Lattices generated by binary fuzzy relations (extended abstract). In: Int. Conf. Fuzzy Set Theory Appl. (1998) 11
5. Kaytoue, M., Kuznetsov, S.O., Napoli, A., Duplessis, S.: Mining gene expression data with pattern structures in Formal Concept Analysis. Inf. Sci. **181**(10) (2011) 1989 – 2001
6. Pankratieva, V.V., Kuznetsov, S.O.: Relations between proto-fuzzy concepts, crisply generated fuzzy concepts, and interval pattern structures. Fundam. Informaticae **115**(4) (2012) 265–277
7. Soldano, H., Santini, G.: Graph abstraction for closed pattern mining in attributed network. In: Eur. Conf. Artif. Intell. (ECAI), IOS Press. (2014) 849–854

On the Existence of Right Adjoints for Surjective Mappings between Fuzzy Structures

Inma P. Cabrera[1], Pablo Cordero[1] Francisca García-Pardo[1], Manuel Ojeda-Aciego, and Bernard De Baets[2]

[1] Universidad de Málaga. Dept. Matemática Aplicada. Andalucía Tech. Spain. *
[2] KERMIT. Department of Mathematical Modelling, Statistics and Bioinformatics. Ghent University

Abstract. We continue our study of the characterization of existence of adjunctions (isotone Galois connections) whose codomain is insufficiently structured. This paper focuses on the fuzzy case in which we have a fuzzy ordering ρ_A on A and a surjective mapping $f\colon \langle A, \approx_A\rangle \to \langle B, \approx_B\rangle$ compatible with respect to the fuzzy equivalences $\approx_A$ and $\approx_B$. Specifically, the problem is to find a fuzzy ordering ρ_B and a compatible mapping $g\colon \langle B, \approx_B\rangle \to \langle A, \approx_A\rangle$ such that the pair (f, g) is a fuzzy adjunction.

1 Introduction

Adjunctions, also called isotone Galois connections, are often used in mathematics in order to relate two (apparently disparate) theories, allowing for mutual cooperative advantages.

A number of papers are being published on the applications (both theoretical and practical) of Galois connections and adjunctions. One can find mainly theoretical papers [10,15,17,23], as well as general applications to computer science, some of them dated more than thirty years ago [21] and, obviously, some more recent works on specific applications, such as programming [16, 22], data analysis [26], or logic [18, 25].

The study of new properties of Galois connections found an important niche in the theory of Formal Concept Analysis (FCA) and its generalizations, since the derivation operators which are used to define the formal concepts actually are a Galois connection. Just to name a few, Lumpe and Schmidt [20] consider adjunctions and their concept posets in order to define a convenient notion of morphism between pattern structures; Bělohlávek and Konečný [3] stress on the "duality" between isotone and antitone Galois connections in showing a case of mutual reducibility of the concept lattices generated by using each type of connection; Denniston *et al* [8] show how new results on Galois connection are applied to formal concept analysis, etc.

* Partially supported by Spanish Ministry of Science projects TIN2014-59471-P and TIN2015-70266-C2-1-P, co-funded by the European Regional Development Fund (ERDF).

© Marianne Huchard, Sergei O. Kuznetsov (Eds.): CLA 2016, pp. 97–108, ISBN 978-5-600-01454-1, National Research University Higher School of Economics, 2016.

It is certainly important to detect when an adjunction (or Galois connection) exists between two structured sets, and this problem has been already studied in the abstract setting of category theory. A different problem arises when either the domain or the codomain is unstructured: the authors studied in a previous work [14] the existence and construction of the right adjoint to a given mapping f in the general framework in which a mapping $f\colon A \to B$ from a (pre-)ordered set A into an unstructured set B is considered, aiming at characterizing those situations in which B can be (pre-)ordered and an isotone mapping $g\colon B \to A$ can be built such that the pair (f, g) is an adjunction. The general approach to this problem adopted in [14] was to consider the canonical decomposition of f with respect to the kernel relation, and consider the three resulting cases separately: the projection on the quotient, the isomorphism between the quotient and the image, and the final inclusion of the image into the codomain. The really important parts of the proof were the first and the last ones, since the intermediate part is straightforward.

We consider this work as an extension of the previous problem to a fuzzy framework, in which several papers on fuzzy Galois connections or fuzzy adjunctions have been written since its introduction by Bělohlávek in [1]; consider for instance [4, 9, 19, 27] for some recent generalizations. Some authors have introduced alternative approaches guided by the intended applications: for instance, Shi *et al* [24] introduced a definition of fuzzy adjunction for its use in fuzzy mathematical morphology.

In this paper, on the one hand, we will consider mappings compatible with fuzzy equivalences $\approx_A$ and $\approx_B$ defined on A and B respectively and, on the other hand, we will just focus on the first part of the canonical decomposition. This means that, up to isomorphism, we have a fuzzy ordering ρ_A on A and a *surjective* mapping $f\colon \langle A, \approx_A\rangle \to \langle B, \approx_B\rangle$ compatible with respect to the fuzzy equivalences $\approx_A$ and $\approx_B$. Specifically, the problem is to characterize when there exists a fuzzy ordering ρ_B and a compatible mapping $g\colon \langle B, \approx_B\rangle \to \langle A, \approx_A\rangle$ such that the pair (f, g) is a fuzzy adjunction.

2 Preliminaries

The most usual underlying structure for considering fuzzy extensions of Galois connections is that of complete residuated lattice, $\mathbb{L} = (L, \leq, \top, \bot, \otimes, \to)$. As usual, supremum and infimum will be denoted by $\vee$ and $\wedge$ respectively. An $\mathbb{L}$-fuzzy set in the universe U is a mapping $X\colon U \to L$ where $X(u)$ means the degree in which u belongs to X. Given X and Y two $\mathbb{L}$-fuzzy sets, X is said to *be included in* Y, denoted as $X \subseteq Y$, if $X(u) \leq Y(u)$ for all $u \in U$.

An $\mathbb{L}$-*fuzzy binary relation* on U is an $\mathbb{L}$-fuzzy subset of $U \times U$, that is $R\colon U \times U \to L$, and it is said to be:

- *Reflexive* if $R(a, a) = \top$ for all $a \in U$.
- $\otimes$-*Transitive* if $R(a, b) \otimes R(b, c) \leq R(a, c)$ for all $a, b, c \in U$.
- *Symmetric* if $R(a, b) = R(b, a)$ for all $a, b \in U$.

From now on, when no confusion arises, we will omit the prefix "$\mathbb{L}$-".

Definition 1. *A* ***fuzzy preordered set*** *is a pair* $\mathbb{A} = \langle A, \rho_A \rangle$ *in which* ρ_A *is a reflexive and* $\otimes$*-transitive fuzzy relation on* A.

Definition 2. *Let* $\mathbb{A} = \langle A, \rho_A \rangle$ *be a fuzzy preordered set. The extensions to the fuzzy setting of the notions of* ***upset*** *and* ***downset*** *of an element* $a \in A$ *are defined by* $a^{\uparrow}, a^{\downarrow} \colon A \to L$ *where*

$$a^{\downarrow}(u) = \rho_A(u, a) \quad \text{and} \quad a^{\uparrow}(u) = \rho_A(a, u) \quad \text{for all}\, u \in A.$$

Definition 3. *An element* $m \in A$ *is a* ***maximum*** *for a fuzzy set* $X \colon A \to L$ *if*

1. $X(m) = \top$ *and*
2. $X \subseteq m^{\downarrow}$, *i.e.,* $X(u) \leq \rho_A(u, m)$ *for all* $u \in A$.

The definition of minimum is similar.

Since the maximum (respectively, minimum) of a fuzzy set needs not be unique, we will include special terminology for them: the crisp set of maxima, respectively minima, for X will be denoted p-max(X), respectively p-min(X).

Definition 4. *Let* $\mathbb{A} = \langle A, \rho_A \rangle$ *and* $\mathbb{B} = \langle B, \rho_B \rangle$ *be fuzzy preordered sets.*

1. *A mapping* $f \colon A \to B$ *is said to be* ***isotone*** *if* $\rho_A(a_1, a_2) \leq \rho_B(f(a_1), f(a_2))$ *for all* $a_1, a_2 \in A$.
2. *A mapping* $f \colon A \to A$ *is said to be* ***inflationary*** *if* $\rho_A(a, f(a)) = \top$ *for all* $a \in A$.
   Similarly, f *is* ***deflationary*** *if* $\rho_A(f(a), a) = \top$ *for all* $a \in A$.

From now on, we will use the following notation: For a mapping $f \colon A \to B$ and a fuzzy subset Y of B, the fuzzy set $f^{-1}(Y)$ is defined as $f^{-1}(Y)(a) = Y(f(a))$, for all $a \in A$.

The definition of fuzzy adjunction given in [11] was the expected extension of that in the crisp case. Namely,

Definition 5. *Let* $\mathbb{A} = \langle A, \rho_A \rangle$, $\mathbb{B} = \langle B, \rho_B \rangle$ *be fuzzy orders, and two mappings* $f \colon A \to B$ *and* $g \colon B \to A$. *The pair* (f, g) *forms a* ***fuzzy adjunction*** *between* A *and* B, *denoted* $(f, g) : \mathbb{A} \leftrightharpoons \mathbb{B}$ *if, for all* $a \in A$ *and* $b \in B$, *the equality* $\rho_A(a, g(b)) = \rho_B(f(a), b)$ *holds.*

As in the crisp case, there exist alternative definitions which are summarized in the theorem below:

Theorem 1 (See [11]). *Let* $\mathbb{A} = \langle A, \rho_A \rangle$, $\mathbb{B} = \langle B, \rho_B \rangle$ *be two fuzzy preordered sets, respectively, and* $f \colon A \to B$ *and* $g \colon B \to A$ *be two mappings. The following statements are equivalent:*

1. $(f, g) : \mathbb{A} \leftrightharpoons \mathbb{B}$.
2. f *and* g *are isotone,* $g \circ f$ *is inflationary, and* $f \circ g$ *is deflationary.*
3. $f(a)^{\uparrow} = g^{-1}(a^{\uparrow})$ *for all* $a \in A$.

4. $g(b)^{\downarrow} = f^{-1}(b^{\downarrow})$ *for all* $b \in B$.
5. f *is isotone and* $g(b) \in \text{p-max}\, f^{-1}(b^{\downarrow})$ *for all* $b \in B$.
6. g *is isotone and* $f(a) \in \text{p-min}\, g^{-1}(a^{\uparrow})$ *for all* $a \in A$.

In the rest of this section, we introduce the preliminary definitions and results needed to establish the new structure we will be working on.

Definition 6. *A fuzzy relation* $\approx$ *on* A *is said to be a:*

- ***Fuzzy equivalence relation*** *if* $\approx$ *is a reflexive,* $\otimes$*-transitive and symmetric fuzzy relation on* A.
- ***Fuzzy equality*** *if* $\approx$ *is a fuzzy equivalence relation satisfying that* $\approx(a,b) = \top$ *implies* $a = b$, *for all* $a, b \in A$.

We will use the infix notation for a fuzzy equivalence relation, that is, we will write $a_1 \approx a_2$ instead of $\approx(a_1, a_2)$.

Definition 7. *Given a fuzzy equivalence relation* $\approx\colon A \times A \to L$, *the* ***equivalence class*** *of an element* $a \in A$ *is the fuzzy set* $[a]_{\approx}\colon A \to L$ *defined by* $[a]_{\approx}(u) = (a \approx u)$ *for all* $u \in A$.

Remark 1. Note that $[x]_{\approx} = [y]_{\approx}$ if and only if $(x \approx y) = \top$: on the one hand, if $[x]_{\approx} = [y]_{\approx}$, then $(x \approx y) = [x]_{\approx}(y) = [y]_{\approx}(y) = \top$, by reflexivity; conversely, if $(x \approx y) = \top$, then $[x]_{\approx}(u) = (x \approx u) = (y \approx x) \otimes (x \approx u) \leq (y \approx u) = [y]_{\approx}(u)$, for all $u \in A$; the other inequality follows similarly.

Definition 8 (See [6]). *Given a fuzzy equivalence relation* $\approx_A$ *on* A, *a fuzzy binary relation* $\rho_A\colon A \times A \to L$ *is said to be*

- $\approx_A$***-reflexive*** *if* $(a_1 \approx_A a_2) \leq \rho_A(a_1, a_2)$,
- $\otimes$-$\approx_A$***-antisymmetric*** *if* $\rho_A(a_1, a_2) \otimes \rho_A(a_2, a_1) \leq (a_1 \approx_A a_2)$,

for all $a_1, a_2 \in A$.

Definition 9. *A triplet* $\mathcal{A} = \langle A, \approx_A, \rho_A \rangle$ *in which* $\approx_A$ *is a fuzzy equivalence relation and* ρ_A *is* $\approx_A$*-reflexive,* $\otimes$-$\approx_A$*-antisymmetric and* $\otimes$*-transitive will be called* $\otimes$-$\approx_A$- fuzzy preordered set *or* ***fuzzy preorder with respect to*** $\approx_A$.

Observe that a fuzzy preorder relation wrt $\approx_A$ is a fuzzy preorder relation because $\top = (a \approx_A a) \leq \rho_A(a, a)$, therefore $\rho_A(a, a) = \top$, for all $a \in A$.

Definition 10. *Let* $\approx_A$ *and* $\approx_B$ *be fuzzy equivalence relations on the sets* A *and* B, *respectively. A mapping* $f\colon A \to B$ *is said to be* ***compatible*** *with* $\approx_A$ *and* $\approx_B$ *if* $(a_1 \approx_A a_2) \leq (f(a_1) \approx_B f(a_2))$ *for all* $a_1, a_2 \in A$.

3 On fuzzy adjunctions wrt fuzzy equivalences

The main idea to extend the notion of fuzzy adjunction to take into account fuzzy equivalences, namely, a fuzzy adjunction between $\mathcal{A} = \langle A, \approx_A, \rho_A \rangle$ and $\mathcal{B} = \langle B, \approx_B, \rho_B \rangle$ is, of course, to require f and g to be compatible mappings and include the necessary adjustments due to the use of fuzzy equivalences. A reasonable possibility is the following:

Definition 11. *Let $\mathcal{A} = \langle A, \approx_A, \rho_A \rangle$ and $\mathcal{B} = \langle B, \approx_B, \rho_B \rangle$ be two fuzzy preordered sets wrt $\approx_A$ and $\approx_B$, respectively. Let $f : A \to B$ and $g : B \to A$ be two mappings which are compatible with $\approx_A$ and $\approx_B$. The pair (f, g) is said to be a* ***fuzzy adjunction*** *between $\mathcal{A}$ and $\mathcal{B}$ if the following conditions hold*

(A1) $(a_1 \approx_A a_2) \otimes \rho_A(a_2, g(b)) \leq \rho_B(f(a_1), b)$
(A2) $(b_1 \approx_B b_2) \otimes \rho_B(f(a), b_1) \leq \rho_A(a, g(b_2))$

for all $a, a_1, a_2 \in A$ and $b, b_1, b_2 \in B$.

Surprisingly, it turns out that Definitions 5 and 11 are very closely related, in fact, they are equivalent up to compatibility of the mappings.

Theorem 2. *Let $\mathcal{A} = \langle A, \approx_A, \rho_A \rangle$ and $\mathcal{B} = \langle B, \approx_B, \rho_B \rangle$ be two fuzzy preordered sets wrt $\approx_A$ and $\approx_B$, respectively. Let $f : A \to B$ and $g : B \to A$ be two mappings which are compatible with $\approx_A$ and $\approx_B$, respectively.*

Then, the pair (f, g) is a fuzzy adjunction between $\mathcal{A}$ and $\mathcal{B}$ if and only if $\rho_A(a, g(b)) = \rho_B(f(a), b)$ for all $a \in A$ and $b \in B$.

Proof. Assume that for all $a \in A$ and $b \in B$ the equality $\rho_A(a, g(b)) = \rho_B(f(a), b)$ holds.

Let $a_1, a_2 \in A$ and $b \in B$. Since f is a map which is compatible with $\approx_A$ and $\approx_B$, then

$$(a_1 \approx_A a_2) \otimes \rho_A(a_2, g(b)) \leq (f(a_1) \approx_B f(a_2)) \otimes \rho_A(a_2, g(b)).$$

By the hypothesis, we obtain that

$$(f(a_1) \approx_B f(a_2)) \otimes \rho_A(a_2, g(b)) \leq (f(a_1) \approx_B f(a_2)) \otimes \rho_B(f(a_2), b).$$

As ρ_B is $\approx_B$-reflexive and transitive, we have that

$$(f(a_1) \approx_B f(a_2)) \otimes \rho_B(f(a_2), b) \leq \rho_B(f(a_1), f(a_2)) \otimes \rho_B(f(a_2), b) \leq \rho_B(f(a_1), b).$$

Therefore, $(a_1 \approx_A a_2) \otimes \rho_A(a_2, g(b)) \leq \rho_B(f(a_1), b)$ for all $a_1, a_2 \in A$ and $b \in B$. Analogously, the condition $(A2)$ holds.

Conversely, assume now that conditions $(A1)$ and $(A2)$ hold. Applying condition $(A1)$, for $a \in A$ and $b \in B$, we have that $(a \approx_A a) \otimes \rho_A(a, g(b)) \leq \rho_B(f(a), b)$. Being $\approx_A$ reflexive, it is deduced that $\rho_A(a, g(b)) \leq \rho_B(f(a), b)$ for all $a \in A$ and $b \in B$. Analogously, $\rho_B(f(a), b) \leq \rho_A(a, g(b))$ for all $a \in A$ and $b \in B$. Therefore, $\rho_A(a, g(b)) = \rho_B(f(a), b)$ for all $a \in A$ and $b \in B$. □

Corollary 1. *If a pair (f, g) is a fuzzy adjunction between $\langle A, \approx_A, \rho_A\rangle$ and $\langle B, \approx_B, \rho_B\rangle$ then (f, g) is also a fuzzy adjunction between the two fuzzy preordered sets $\langle A, \rho_A\rangle$ and $\langle B, \rho_B\rangle$.*

Conversely, if a pair (f, g) is a fuzzy adjunction between $\langle A, \rho_A\rangle$ and $\langle B, \rho_B\rangle$ then (f, g) is also a fuzzy adjunction between $\langle A, =, \rho_A\rangle$ and $\langle B, =, \rho_B\rangle$, being $=$ the standard crisp equality.

In the rest of this section, we extend the results in [12, 13] to the framework of fuzzy preordered sets wrt a fuzzy equivalence relation. The underlying idea is similar, but now the mappings f and g need to be compatible with fuzzy equivalence relations $\approx_A$ on A and $\approx_B$ on B, and this makes the development to be much more involved that in the previous case.

To begin with, it is worth to mention that the equivalences in Theorem 1 are valid when considering fuzzy equivalences: obviously, the mappings have to be compatible.

Remark 2. Given two elements $x_1, x_2 \in$ p-max(X), note that $\rho_A(x_1, x_2) = \top = \rho_A(x_2, x_1)$: on the one hand, by $x_1 \in$ p-max(X), we have that $X(x_1) = \top$ and since $x_2 \in$ p-max(X), then $X(u) \leq \rho_A(u, x_2)$ for all $u \in A$. Hence, $\top = X(x_1) \leq \rho_A(x_1, x_2)$ which implies that $\rho_A(x_1, x_2) = \top$.

Likewise, by $\otimes$-$\approx_A$-antisymmetry, also $(x_1 \approx_A x_2) = \top$ for $x_1, x_2 \in \approx_A$-max(X).

Theorem 3. *Let $\mathcal{A} = \langle A, \approx_A, \rho_A\rangle$ and $\mathcal{B} = \langle B, \approx_B, \rho_B\rangle$ be two fuzzy preordered sets. If the pair (f, g) is a fuzzy adjunction between $\mathcal{A}$ and $\mathcal{B}$ then $\big((f \circ g \circ f)(a) \approx_B f(a)\big) = \top$ and $\big((g \circ f \circ g)(b) \approx_A g(b)\big) = \top$, for all $a \in A, b \in B$.*

Proof. Since f is isotone and $g \circ f$ is inflationary we have

$$\top = \rho_A(a, gf(a)) \leq \rho_B(f(a), fgf(a)),$$

therefore, $\rho_B(f(a), fgf(a)) = \top$.

Moreover, $\rho_B(fgf(a), f(a)) = \rho_A(gf(a), gf(a)) = \top$. Therefore, from the $\otimes$-$\approx_B$-antisymmetric property, we obtain $(f \circ g \circ f)(a) \approx_B f(a)) = \top$.

For the other composition, the proof is analogous. □

Corollary 2. *Let $\mathcal{A} = \langle A, \approx_A, \rho_A\rangle$ and $\mathcal{B} = \langle B, \approx_B, \rho_B\rangle$ be two fuzzy preordered sets. If the pair (f, g) is a fuzzy adjunction between $\mathcal{A}$ and $\mathcal{B}$ then, for all $a \in A, b \in B$,*

(i) $\rho_B\big((f \circ g \circ f)(a), f(a)\big) = \rho_B\big(f(a), (f \circ g \circ f)(a)\big) = \top$

(ii) $\rho_A\big((g \circ f \circ g)(b), g(b)\big) = \rho_A\big(g(b), (g \circ f \circ g)(b)\big) = \top$.

Corollary 3. *Let $\mathcal{A} = \langle A, \approx_A, \rho_A\rangle$ and $\mathcal{B} = \langle B, \approx_B, \rho_B\rangle$ be two fuzzy preordered sets. If the pair (f, g) is a fuzzy adjunction between $\mathcal{A}$ and $\mathcal{B}$ then, for all $a_1, a_2 \in A$ and $b_1, b_2 \in B$, the following equalities hold:*

(i) $\big(f(a_1) \approx_B f(a_2)\big) = \big((g \circ f)(a_1) \approx_A (g \circ f)(a_2)\big)$.

(ii) $\big(g(b_1) \approx_A g(b_2)\big) = \big((f \circ g)(b_1) \approx_B (f \circ g)(b_2)\big)$.

Proof. We will prove just the first item, since the second one is similar.

Given $a_1, a_2 \in A$, since g is compatible, we have that $\big(f(a_1) \approx_B f(a_2)\big) \le \big((g \circ f)(a_1) \approx_A (g \circ f)(a_2)\big)$. On the other hand, since f is compatible, we have that

$$\big(g(f(a_1)) \approx_A g(f(a_2))\big) \le \big(f(g(f(a_1))) \approx_B f(g(f(a_2)))\big).$$

Now, by Theorem 3, we have that $\big(f(a) \approx_B f(g(f(a)))\big) = \top$, for all $a \in A$. Finally, the $\otimes$-transitivity of $\approx_B$ leads to

$$\begin{aligned}
&\big(f(g(f(a_1))) \approx_B f(g(f(a_2)))\big)\\
&\qquad = \big(f(a_1) \approx_B f(g(f(a_1)))\big) \otimes \big(f(g(f(a_1))) \approx_B f(g(f(a_2)))\big)\\
&\qquad \le \big(f(a_1) \approx_B f(g(f(a_2)))\big)\\
&\qquad = \big(f(a_1) \approx_B f(g(f(a_2)))\big) \otimes \big(f(g(f(a_2))) \approx_B f(a_2)\big)\\
&\qquad \le \big(f(a_1) \approx_B f(a_2)\big)
\end{aligned}$$

□

4 Characterization and construction of the adjunction

Some more definitions are needed in order to solve the problem in the case of surjective mappings.

Definition 12. *Let $\mathcal{A} = \langle A, \approx_A, \rho_A\rangle$ and $\mathcal{B} = \langle B, \approx_B, \rho_B\rangle$ be two fuzzy preordered sets wrt $\approx_A$ and $\approx_B$, respectively, and let $f\colon \mathcal{A} \to \mathcal{B}$ be a compatible mapping. The* ***fuzzy kernel relation*** *$\equiv_f\colon A \times A \to L$ associated to f is defined as follows for $a_1, a_2 \in A$,*

$$(a_1 \equiv_f a_2) = (f(a_1) \approx_B f(a_2)).$$

Trivially, the fuzzy kernel relation is a fuzzy equivalence relation. The equivalence class of an element $a \in A$ is a fuzzy set denoted by $[a]_f\colon A \to L$ defined by $[a]_f(u) = (f(a) \approx_B f(u))$ for all $u \in A$.

The following definitions recall the notion of Hoare ordering between crisp subsets, and then introduces an alternative statement in the subsequent lemma:

Definition 13. *Let $\mathcal{A} = \langle A, \approx_A, \rho_A\rangle$ be a fuzzy preordered set wrt a fuzzy equivalence relation $\approx_A$. For C, D crisp subsets of A, consider the following notation*

- $(C \sqsubseteq_W D) = \bigvee_{c\in C} \bigvee_{d\in D} \rho_A(c, d)$
- $(C \sqsubseteq_H D) = \bigwedge_{c\in C} \bigvee_{d\in D} \rho_A(c, d)$
- $(C \sqsubseteq_S D) = \bigwedge_{c\in C} \bigwedge_{d\in D} \rho_A(c, d)$

Lemma 1. *Let $\mathcal{A} = \langle A, \approx_A, \rho_A \rangle$ be a fuzzy preordered set wrt a fuzzy equivalence relation $\approx_A$, $X, Y \subseteq A$ such that* $\text{p-max}(X) \neq \varnothing \neq \text{p-max}(Y)$*, then*

$$\begin{aligned}(\text{p-max}(X) \sqsubseteq_W \ \text{p-max}(Y)) &= (\text{p-max}(X) \sqsubseteq_H \ \text{p-max}(Y)) \\ &= (\text{p-max}(X) \sqsubseteq_S \text{p-max}(Y)) = \rho_A(x, y)\end{aligned}$$

for any $x \in$ p-max(X) and $y \in$ p-max(Y).

Proof. Let us show that $\rho_A(x, y) = \rho_A(\bar{x}, \bar{y})$, for any $x, \bar{x} \in \text{p-max}(X)$ and $y, \bar{y} \in \text{p-max}(Y)$: Indeed, using the transitive property of ρ_A and Remark 2 we have that

$$\rho_A(x, y) \geq \rho_A(x, \bar{x}) \otimes \rho_A(\bar{x}, y) = \top \otimes \rho_A(\bar{x}, y) \geq \rho_A(\bar{x}, \bar{y}) \otimes \rho_A(\bar{y}, y) = \rho_A(\bar{x}, \bar{y}).$$

Analogously, $\rho_A(\bar{x}, \bar{y}) \geq \rho_A(x, y)$. Therefore, $\rho_A(\bar{x}, \bar{y}) = \rho_A(x, y)$ for any $x, \bar{x} \in \text{p-max}(X)$ and $y, \bar{y} \in \text{p-max}(Y)$. □

Notice that, by Lemma 1, when both sets are non-empty, for any $x \in \text{p-max}(X)$ and $y \in \text{p-max}(Y)$, $\big(\text{p-max}(X) \sqsubseteq_H \text{p-max}(Y)\ \big) = \rho_A(x, y)$ and this justifies the following notation.

Notation 1 *Let $\mathcal{A} = \langle A, \approx_A, \rho_A \rangle$ be a fuzzy preorder wrt a fuzzy equivalence relation $\approx_A$. Let X, Y be crisp subsets of A such that* $\text{p-max}(X) \neq \varnothing \neq \text{p-max}(Y)$*, then $\rho_A(\text{p-max}(X), \text{p-max}(Y))$ denotes $\big(\text{p-max}(X) \sqsubseteq_H \text{p-max}(Y)\big)$.*

Remark 3. Let $\mathcal{A} = \langle A, \approx_A, \rho_A \rangle$ be a fuzzy preorder wrt a fuzzy equivalence relation $\approx_A$ and $X, Y \subseteq A$. Observe that for all $x_1, x_2 \in \text{p-max}(X)$ and $y_1, y_2 \in \text{p-max}(Y)$, we have that $(x_1 \approx_A y_1) = (x_2 \approx_A y_2)$:

Indeed, recall that $(x_1 \approx_A x_2) = \top = (y_1 \approx_A y_2)$, then $(x_1 \approx_A y_1) = (x_2 \approx_A x_1) \otimes (x_1 \approx_A y_1) \leq (x_2 \approx_A y_1) = (x_2 \approx_A y_1) \otimes (y_1 \approx_A y_2) \leq (x_2 \approx_A y_2)$.

Therefore, we can use the notation

$$\big(\text{p-max}(X) \approx_A \text{p-max}(Y)\big) = (x \approx_A y)$$

for any $x \in \text{p-max}(X), y \in \text{p-max}(Y)$.

Theorem 4 (Necessary conditions). *Let $\mathcal{A} = \langle A, \approx_A, \rho_A \rangle$, $\mathcal{B} = \langle B, \approx_B, \rho_B \rangle$ be two fuzzy preorders and $f \colon A \to B, g \colon B \to A$ two mappings which are compatible with the equivalence relations $\approx_A$ and $\approx_B$. If (f, g) is a fuzzy adjunction between $\mathcal{A}$ and $\mathcal{B}$ then*

1. $\text{p-max}([a]_f)$ *is non-empty for all $a \in A$.*
2. $\rho_A(a_1, a_2) \leq \rho_A(\text{p-max}([a_1]_f), \text{p-max}([a_2]_f))$*, for all $a_1, a_2 \in A$.*
3. $(a_1 \equiv_f a_2) \leq (\text{p-max}([a_1]_f) \approx_A \text{p-max}([a_2]_f))$*, for all $a_1, a_2 \in A$.*

Proof.

- *Condition* 1. We will show that $g(f(a)) \in$ p-max$([a]_f)$:
  By Theorem 3, we have $(f(a) \approx_B f(g(f(a)))) = \top$.
  On the other hand, using the $\approx_B$-reflexivity and that (f, g) is a fuzzy adjunction, for all $u \in A$,

  $$[a]_f(u) = (f(u) \approx_B f(a)) \leq \rho_B(f(u), f(a)) = \rho_A(u, g(f(a))) = g(f(a)) \downarrow (u)$$

- *Condition* 2. By Theorem 1, f and g are isotone maps, thus

  $$\rho_A(a_1, a_2) \leq \rho_A(g(f(a_1)), g(f(a_2)))$$

  for all $a_1, a_2 \in A$. We have just shown that $g(f(a)) \in$ p-max$([a]_f)$ for all $a \in A$, thus, from Lemma 1, we obtain that $\rho_A(a_1, a_2) \leq \rho_A($p-max$([a_1]_f)$, p-max$([a_2]_f))$ for all $a_1, a_2 \in A$.
- *Condition* 3. Since g is compatible with $\approx_B$ and $\approx_A$, then $(a_1 \equiv_f a_2) = (f(a_1) \approx_B f(a_2)) \leq (g(f(a_1)) \approx_A g(f(a_2)))$. But, by Condition 1, $g(f(a_i)) \in$ p-max$([a_i]_f)$.

□

Given $\mathcal{A} = \langle A, \approx_A, \rho_A \rangle$ a fuzzy preordered set wrt $\approx_A$ and a surjective mapping $f: A \to B$ compatible with $\approx_A$ and $\approx_B$, our first goal is to find sufficient conditions to define a suitable fuzzy preordering wrt $\approx_B$ on B and a mapping $g: B \to A$ compatible with $\approx_B$ and $\approx_A$ such that (f, g) is an adjoint pair.

Lemma 2. *Let $\mathcal{A} = \langle A, \approx_A, \rho_A \rangle$ be a fuzzy preorder and $\approx_B$ be a fuzzy equivalence relation on B together with a surjective mapping $f: A \to B$ compatible with $\approx_A$ and $\approx_B$. Suppose that* p-max$([a]_f) \neq \varnothing$ *for all $a \in A$. Then, $\mathcal{B} = \langle B, \approx_B, \rho_B \rangle$ is a fuzzy preorder wrt $\approx_B$, where ρ_B is the fuzzy relation defined as follows*

$$\rho_B(b_1, b_2) = \rho_A(\text{p-max}([a_1]_f), \text{p-max}([a_2]_f))$$

where $a_i \in f^{-1}(b_i)$ for each $i \in \{1, 2\}$.

Theorem 5 (Sufficient conditions). *Let $\mathcal{A} = \langle A, \approx_A, \rho_A \rangle$ be a fuzzy preorder wrt $\approx_A$ and $\approx_B$ be a fuzzy equivalence relation on B together with a surjective mapping $f: A \to B$ compatible with $\approx_A$ and $\approx_B$.*

Suppose that the following conditions hold:

1. p-max$([a]_f)$ *is non-empty for all $a \in A$.*
2. $\rho_A(a_1, a_2) \leq \rho_A($p-max$([a_1]_f)$,p-max$([a_2]_f))$*, for all $a_1, a_2 \in A$.*
3. $(a_1 \equiv_f a_2) \leq ($p-max$([a_1]_f) \approx_A ($p-max$([a_2]_f))$*, for all $a_1, a_2 \in A$.*

Then, there exists a mapping $g: B \to A$ compatible with $\approx_A$ and $\approx_B$ such that (f, g) is a fuzzy adjunction between the fuzzy preorders $\mathcal{A}$ and $\mathcal{B} = \langle B, \approx_B, \rho_B \rangle$, where ρ_B is the fuzzy relation introduced in Lemma 2.

Proof. Following Lemma 2, by Condition 1, there exists a fuzzy preordering ρ_B defined as follows:

$$\rho_B(b_1, b_2) = \rho_A(\text{p-max}([a_1]_f), \text{p-max}([a_2]_f))$$

where $a_i \in f^{-1}(b_i)$ for each $i \in \{1,2\}$.

There is a number of suitable definitions of $g\colon B \to A$, and all of them can be specified as follows: given $b \in B$, we choose $g(b)$ as an element $x_b \in \text{p-max}([x]_f)$, where x is any element of $f^{-1}(b)$.

The existence of g is guaranteed by the axiom of choice, since f is surjective and for all $b \in B$ and for all $x \in f^{-1}(b)$, the set $\text{p-max}([x]_f)$ is nonempty. Moreover, $g(b)$ does not depend on the preimage of b, because $f(x) = f(y) = b$ implies $[x]_f = [y]_f$.

The compatibility of g with $\approx_B$ and $\approx_A$ follows from Condition 3:

$$(b_1 \approx_B b_2) = (f(a_1) \approx_B f(a_2)) = (a_1 \equiv_f a_2) \leq (c_1 \approx_A c_2)$$

for all $a_i \in f^{-1}(b_i)$ and $c_i \in \text{p-max}([a_i]_f)$, for $i \in \{1,2\}$. In particular, $(b_1 \approx_B b_2) \leq (g(b_1) \approx_A g(b_2))$.

Now, due to Theorem 2, it suffices to prove that $\rho_A(a, g(b)) = \rho_B(f(a), b)$, for all $a \in A, b \in B$:

Firstly, by Lemma 1, $\rho_B(f(a), b) = \rho_A(u, v)$ for all $u \in \text{p-max}([a]_f)$ and $v \in \text{p-max}([z]_f)$ where $z \in f^{-1}(b)$. Since, by its definition, we have that $g(b) \in \text{p-max}([z]_f)$, we obtain $\rho_B(f(a), b) = \rho_A(u, g(b))$. Thus, we have to prove just that

$$\rho_A(u, g(b)) = \rho_A(a, g(b))$$

for all $u \in \text{p-max}([a]_f)$.

Given $u \in \text{p-max}([a]_f)$, we have $(f(a) \approx_B f(u)) = \top$ and $(f(a) \approx_B f(x)) \leq \rho_A(x, u)$, for all $x \in A$. In particular, $(f(a) \approx_B f(a)) \leq \rho_A(a, u)$, and then, since $\approx_A$ is reflexive, we obtain $\rho_A(a, u) = \top$. Therefore,

$$\rho_A(u, g(b)) = \rho_A(a, u) \otimes \rho_A(u, g(b)) \leq \rho_A(a, g(b))$$

On the other hand, for any $x \in f^{-1}(b)$, we have that $g(b) \in \text{p-max}([x]_f)$, then $(f(x) \approx_B f(g(b))) = \top$ which implies that $[g(b)]_f = [x]_f$, by Remark 1. Applying Condition 2,

$$\begin{aligned}\rho_A(a, g(b)) &\leq \rho_A(\text{p-max}([a]_f), \text{p-max}([g(b)]_f)) = \\ &= \rho_A(\text{p-max}([a]_f), \text{p-max}([x]_f)) = \rho_B(f(a), b).\end{aligned}$$

□

5 Conclusions

This work continues the research line initiated in [12–14] on the characterization of existence of adjunctions (and Galois connections) for mappings with unstructured codomain.

We have found necessary and sufficient conditions under which, given a fuzzy ordering ρ_A on A and a surjective mapping $f\colon \langle A, \approx_A\rangle \to \langle B, \approx_B\rangle$ compatible with respect to the fuzzy equivalences $\approx_A$ and $\approx_B$, there exists a fuzzy ordering

ρ_B and a compatible mapping $g\colon \langle B, \approx_B\rangle \to \langle A, \approx_A\rangle$ such that the pair (f, g) is a fuzzy adjunction.

As pieces of future work, on the one hand, the use of fuzzy equivalences can be taken into account in order to weaken the notion of surjective function and obtain an alternative proof based on this weaker notion. On the other hand, as stated in the introduction, considering surjective mappings is just the first step in the canonical decomposition of a general mapping $f\colon \langle A, \approx_A\rangle \to \langle B, \approx_B\rangle$, therefore we will study how to extend the obtained ordering to the whole codomain in the case that f is not surjective.

Finally, as a midterm goal, we would like to study possible links of our constructions with some recent efforts to develop a so-called theory of constructive Galois connections [7] aimed at introducing adjunctions and Galois connections within automated proof checkers.

References

1. R. Bělohlávek. Fuzzy Galois connections. *Mathematical Logic Quarterly*, 45(4):497–504, 1999.
2. R. Bělohlávek. Fuzzy relational systems: foundations and principles. Kluwer, 2002.
3. R. Bělohlávek and J. Konecny Concept lattices of isotone vs. antitone Galois connections in graded setting: Mutual reducibility revisited. *Information Sciences* 199:133–137, 2012.
4. R. Bělohlávek. and P. Osička. Triadic fuzzy Galois connections as ordinary connections. In *IEEE Intl Conf on Fuzzy Systems*, 2012.
5. U. Bodenhofer and F. Klawonn. A formal study of linearity axioms for fuzzy orderings. *Fuzzy Sets and Systems* 145(3):323–354, 2004.
6. U. Bodenhofer, B. De Baets and J. Fodor. A compendium of fuzzy weak orders: Representations and constructions. *Fuzzy Sets and Systems* 158(8):811–829, 2007.
7. D. Darais and D. Van Horn. Constructive Galois Connections: Taming the Galois Connection Framework for Mechanized Metatheory. `arXiv:1511.06965`, 2015.
8. J. Denniston, A. Melton, and S.E. Rodabaugh. Formal Contexts, Formal Concept Analysis, and Galois Connections. *Electronic Proc. in Theoretical Computer Science* 129:105–120, 2013.
9. A. Frascella. Fuzzy Galois connections under weak conditions. *Fuzzy Sets and Systems*, 172(1):33–50, 2011.
10. J. G. García, I. Mardones-Pérez, M. A. de Prada-Vicente, and D. Zhang. Fuzzy Galois connections categorically. *Math. Log. Q.*, 56(2):131–147, 2010.
11. F. García-Pardo, I.P. Cabrera, P. Cordero, and M. Ojeda-Aciego. On Galois connections and Soft Computing. *Lect. Notes in Computer Science*, 7903:224–235, 2013.
12. F. García-Pardo, I.P. Cabrera, P. Cordero, and M. Ojeda-Aciego. On the construction of fuzzy Galois connections. *Proc. of XVII Spanish Conference on Fuzzy Logic and Technology*, pages 99-102, 2014.
13. F. García-Pardo, I.P. Cabrera, P. Cordero, and M. Ojeda-Aciego. On adjunctions between fuzzy preordered sets: necessary conditions. *Lect. Notes in Computer Science*, 8536,pp. 211–221, 2014.
14. F. García-Pardo, I.P. Cabrera, P. Cordero, M. Ojeda-Aciego, and F.J. Rodríguez. On the definition of suitable orderings to generate adjunctions over an unstructured codomain. *Information Sciences* 286:173–187, 2014.

15. G. Georgescu and A. Popescu. Non-commutative fuzzy Galois connections. *Soft Computing*, 7(7):458–467, 2003.
16. J. Gibbons, F. Henglein, R. Hinze, and N. Wu. Relational Algebra by way of Adjunctions. In *Proc of Databases and Programming Languages (DBLP)*, 2015
17. J. Järvinen. Pawlak's information systems in terms of Galois connections and functional dependencies. *Fundamenta Informaticae*, 75:315–330, 2007.
18. J. Järvinen, M. Kondo, and J. Kortelainen. Logics from Galois connections. *Int. J. Approx. Reasoning*, 49(3):595–606, 2008.
19. J. Konecny. Isotone fuzzy Galois connections with hedges. *Information Sciences*, 181(10):1804–1817, 2011.
20. L. Lumpe and S.E. Schmidt. Pattern Structures and Their Morphisms. In *Proc. of Concept Lattices and Their Applications (CLA)*, CEUR Workshop Proceedings 1466:171–179, 2015
21. A. Melton, D.A. Schmidt, and G.E. Strecker. Galois connections and computer science applications. *Lect. Notes in Computer Science*, 240:299–312, 1986.
22. S.-C. Mu and J. Oliveira. Programming from Galois connections. *Journal of Logic and Algebraic Programming*, 81(6):680–704, 2012.
23. M. Nowak A Proof of Tarski's Fixed Point Theorem by Application of Galois Connections. *Studia Logica* 103(2):287–301, 2014
24. Y. Shi, M. Nachtegael, D. Ruan, and E. Kerre Fuzzy adjunctions and fuzzy morphological operations based on implications. *Intl J of Intelligent Systems* 24(12):1280–1296, 2009.
25. F.J. Valverde-Albacete, C. Peláez-Moreno, and C. del Campo. Activating Generalized Fuzzy Implications from Galois Connections. In *Enric Trillas: A passion for fuzzy sets*, Studies in Fuzzy Sets and Soft Computing 322:201–212, 2015.
26. M. Wolski. Galois connections and data analysis. *Fundamenta Informaticae*, 60:401–415, 2004.
27. W. Yao and L.-X. Lu. Fuzzy Galois connections on fuzzy posets. *Mathematical Logic Quarterly 55:105–112, 2009*

FCA for Software Product Lines Representation: Mixing Configuration and Feature Relationships in a Unique Canonical Representation

Jessie Carbonnel[1], Karell Bertet[2], Marianne Huchard[1] and Clémentine Nebut[1]

[1] LIRMM, CNRS and Université de Montpellier, France
[2] L3i, Université de La Rochelle, France
{jcarbonnel,huchard,nebut}@lirmm.fr,bertet@univ-lr.fr

Abstract. Software Product Line Engineering (SPLE) is a software engineering domain in which families of similar softwares (called products) are built reusing common artifacts. This requires to analyze commonalities and variabilities, for example to detect which parts are common to several products and which parts differ from one product to another. Such software characteristics that may be present or not in a product are called features. Several approaches in the literature exist to organize features and product configurations in terms of features. In this paper we review those approaches and show that concept lattices are a relevant structure to organize features and product configurations. We also address scaling issues related to formal context computation in the domain of SPLE.

Keywords: Software Product Lines, Feature Model, Formal Concept Analysis, Concept Lattice

1 Introduction

Software Product Line Engineering (SPLE) focuses on the reuse of common software pieces to reduce the building and maintenance cost of similar software systems (called products). An important step of this methodology consists in analyzing and modeling variability, i.e. mainly extracting "features", a feature being a discriminating characteristic common to several products or specific to a product. A product configuration is then a set of these features. Different formalisms are used in SPLE to organize features and product configurations. Some of these formalisms focus on features, while others represent product configurations. Some are canonical, while others are not, and depend on the designer point of view.

In this paper, we review the main used formalisms and we show that concept lattices might be a relevant (canonical) structure for representing variability, while highlighting information on relationships between product configurations, and between product configurations and features, that other formalisms hardly represent. Besides explaining what is the contribution of concept lattices to variability representation, we propose a solution to address some scaling issues of

ISBN 978-5-600-01454-1, National Research University Higher School of Economics, 2016.

concept lattices in this domain. Actually, scaling issues can occur at two levels when computing: (1) the formal context, and (2) the concept lattice. Here, we focus on scaling issues related to formal context computation; we investigate implicative systems on attributes as closure operators to build a feature closed sets lattice without building the formal context. We show that implicative systems are another representation of the variability that can be useful for designers.

The remaining of the paper is organized as follows. Section 2 presents the various formalisms found in the literature to capture the variability of a software product line. Section 3 shows that concept lattices are an interesting formalism to analyse variability, and presents related work concerning the use of formal concept analysis for product lines. Section 4 explains how using implicative systems allows to face scaling issues related to formal context computation for variability management.

2 Existing Formalisms for variability representation

To capture and describe the variability of a software product line, almost all approaches in the literature use feature-oriented representations [11, 12, 20]. Features describe and discriminate the products. As an example, features for an e-commerce website may include displaying a *catalog*, proposing to fill a *basket* of products, or offering a *payment_method*. In our context, we consider a feature set F. A product configuration (or simply a configuration) is a subset of F.

Feature models (FMs) are graphical representations that include a decorated feature tree and a set of textual cross-tree constraints which complements information given in the tree. The vertices of the tree are the features (from F), while the edges (in $F \times F$) correspond to refinement or sub-feature (part of) relationships in the domain. Edges can be decorated by a symbol meaning that if the parent feature is selected, the child feature can be selected or not (*optional*). Another symbol indicates that if the parent feature is selected, the child feature is necessarily selected (*mandatory*). Groups of edges rooted in a feature represent: *xor-groups* (if the parent feature is selected, exactly one feature has to be selected in the group), and *or-groups* (if the parent feature is selected, at least one feature has to be selected in the group). Fig. 1 shows a simple FM for e-commerce websites.

Such a software necessarily includes a *catalog* for proposing products, and this catalog is displayed using a *grid* or (exclusive) using a *list*. Optionnally, a *basket* functionality is proposed. A *payment_method* may also be optionally proposed. Two payment methods are proposed: *credit_card* or (inclusive) *check*. A cross-tree constraint, written below the tree, indicates that if a basket is proposed, a payment method is also proposed (and reciprocally).

A variability representation conveys *ontological* information (*ontological* semantics): the edges of the feature tree and the groups correspond to domain knowledge, e.g. the group *grid, list* indicates a semantic refinement of catalog; the edge *(e_commerce, catalog)* indicates that catalog is a subpart of the website.

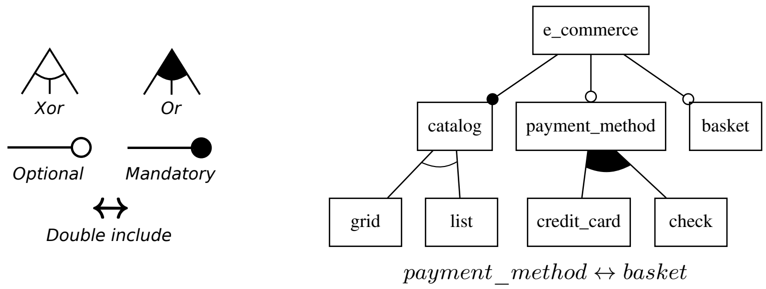


Fig. 1: (Left) Basic feature model's relationships and their corresponding edges; (Right) A basic feature model for an e-commerce website

A variability representation also has a *logical* semantics: for example an alternative representation of the FM is an equivalent propositional formula with $|F|$ variables and constraints defined using propositional connectives ($\wedge, \vee, \rightarrow, \leftrightarrow$ and $\neg$) [5, 8]. Automated analysis can then be performed using SAT-solvers, generally on the Conjunctive or Disjunctive Normal Forms (CNF or DNF). Fig. 2 shows the propositional formula equivalent to the FM of Fig. 1. The down side of the propositional formula is that *ontological* semantics is lost, e.g. an implication in the formula may represent a subpart relationship or a cross-tree constraint.

$$
\begin{array}{ll}
\mathit{hierarchy}: & (Ca \rightarrow Ec) \wedge (G \rightarrow Ca) \wedge (L \rightarrow Ca) \wedge \\
 & (Pm \rightarrow Ec) \wedge (Cc \rightarrow Pm) \wedge (Ch \rightarrow Pm) \wedge (B \rightarrow Ec) \wedge \\
\mathit{xor\text{-}groups}: & (Ca \rightarrow (G \oplus L)) \wedge \\
\mathit{or\text{-}groups}: & (Pm \rightarrow (Cc \vee Ch)) \wedge \\
\mathit{mandatory}: & (Ec \rightarrow Ca) \wedge \\
\mathit{cross\text{-}tree}: & (Pm \leftrightarrow B)
\end{array}
$$

Fig. 2: Propositional formula corresponding to the feature model of Fig. 1

The third semantics is the *configuration-semantics* that associates to any variability representation the set of its valid configurations. The set of the 8 valid configurations for the FM of Fig. 1 is given in Table 1. For the sake of space, it is shown using the Formal Context representation, which is equivalent.

An important property of a formalism is *canonicity*. Given a set of configurations that are to be represented, and considering a chosen formalism, there are often different ways of writing a representation of a given set of configurations following this formalism. For example different feature models can have the same *configuration-semantics*. *Concision* is also an interesting property: a variability representation can be extensional if it enumerates all the possible configurations, or intensional if it represents these configurations in a more compact way. For example, the formal context of Table 1 is an extensional representation of the

Table 1: Set of valid configurations of the FM of Fig. 1

	e_com.(Ec)	catalog(Ca)	grid(G)	list(L)	pay.met.(Pm)	cred.card(Cc)	check(Ch)	basket(B)
1	x	x	x					
2	x	x		x				
3	x	x	x		x	x		x
4	x	x	x		x		x	x
5	x	x	x		x	x	x	x
6	x	x		x	x	x		x
7	x	x		x	x		x	x
8	x	x		x	x	x	x	x

FM of Fig. 1, whereas the FM is an intensional representation of variability.

In this section, we study graph-like representations which have been used in the literature to express software product line variability of a feature model starting from a propositional formula. For each representation, we give a definition and discuss its canonicity, concision, configuration semantics and ontological semantics.

A **binary decision tree** (BDT) is a tree-like graph used to represent the truth table of a boolean function equivalent to a propositional formula: it is an extensional representation. This representation has redundancies which can be avoided by node sharing, which results in a graph called **binary decision diagram** [8, 6] (BDD): this representation is more concise than the BDT. BDD usually refers to ROBDD (for reduced ordered binary decision diagram), which is unique for a given propositional formula. A BDD depicts the same set of configurations as the original feature model, but the ontological semantics is lost in the transformation. A propositional formula can also be represented as an **implication hypergraph** [8]. As the implication set for a given formula is not necessarily unique (except if it is a canonical basis), neither is the obtained hypergraph. The hypergraph depicts exactly the same configuration set. It also keeps a part of the ontological semantics, as feature groups patterns can be extracted from the hyperedges. Another similar representation is the **implication graph** [8], which only depicts binary implications, and thus does not express feature groups. For a given propositional formula, several implication graphs can be constructed, but two induced structures are unique: the transitive closure and the transitive reduction of the graph. Its configuration semantics is not always the same as the original feature model, because an implication graph can eventually depict more configurations, as it expresses less constraints than the original feature model or propositional formula. Finally, a **feature graph** [19] is a diagram-like representation which seeks to describe all feature models which depict a same set of configurations. Because configuration semantics do not formulate mandatory relationships between features, they are not expressed in feature graphs either. As for FMs, a feature graph is not necessarily unique for a given set of configurations, but the transitive reduction and the transitive closure of the feature graph are canonical. All these representations express variability in a compact way.

Table 2: Properties of the different formalisms

Domain	Representation	Canonicity	Same Conf. Sem. as the FM	Same Logical Sem. as the FM	Ontol. sem.: Groups	Ontol. sem.: include vs. refin.	Ontol. sem.: double include vs. mand	$F \times F$	$Config. \times F$	$Config. \times Config.$	textual representation	graphical representation	extensional representation	intensional representation
SPLE	Set of configurations	x	x	x					x		x		x	
SPLE	Feature model		x	x	x	x	x	x			x	x		x
SPLE	Propositional formula		x	x				x			x			x
SPLE	Binary decision tree	x	x	x				x				x	x	
SPLE	Binary decision diagram	x	x	x				x				x		x
SPLE	Implication hypergraph		x	x	x			x				x		x
SPLE	Implication graph (IG)							x				x		x
SPLE	IG $\rightarrow$ Transitive reduction	x						x				x		x
SPLE	IG $\rightarrow$ Transitive closure	x						x				x		x
SPLE	Feature graph (FG)		x	x	x			x			x	x		x
SPLE	FG $\rightarrow$ Transitive reduction	x	x	x	x			x			x	x		x
SPLE	FG $\rightarrow$ Transitive closure	x	x	x	x			x			x	x		x
FCA	**Formal Context**	x	x	x					x		x		x	
FCA	**Concept lattice**	**x**	**x**	**x**				**x**	**x**	**x**		x	x	
FCA	**Labelled feature closed set lattice**	**x**	**x**	**x**	**x**			**x**	**x**	**x**		x	x	

The upper part of Table 2 compares the different formalisms used in SPLE domain with respect to canonicity and their ability to encompass or highlight the different semantics. Besides, it shows which kinds of relationships can be read in the formalism: between features only, between configurations and features, or between configurations. Then it indicates if this is a textual or a graphical formalism, and if this is an intensional or an extensional representation. In SPLE domain, all representations (except the set of configurations and the BDT) consider an intensional point of view with only feature organization. FM is the only representation which clearly expresses all ontological information, but it is not canonical, since many relevant FMs can be built from domain information. Implication hypergraph and feature graph preserve the notion of groups, but refinement and mandatory information of features are lost.

To sum up, these formalisms concentrate on feature organization (except the set of configurations and the BDT), are more or less respectful of initial semantics of the FM they represent and none of them considers a mixed representation of features and configurations. In the next section, we show the benefits of having such a mixed representation and in general, the contributions that a concept lattice based representation may bring to the SPLE domain as a complement to the existing representations.

3 Contribution of concept lattices to variability representation and related work

Formal Concept Analysis [10] provides an alternative framework for variability representation, based on a configuration list, given in the form of a formal context (as in Table 1). Formal objects are the configurations, while formal attributes are the features. Fig. 3 presents the corresponding concept lattice. A concept groups a maximal set of configurations sharing a maximal set of features. In the representation, configurations appear in the lower part of the concepts and are inherited from bottom to top. Features appear in the upper part of the concepts and are inherited from top to bottom. This representation includes the FM, in the sense that if there is an edge indicating F_2 sub-feature of F_1 in the tree, these features are respectively introduced in two comparable concepts $C_2 \leq C_1$, furthermore, the cross-tree constraints are verified by the logic formula that describes the concept lattice.

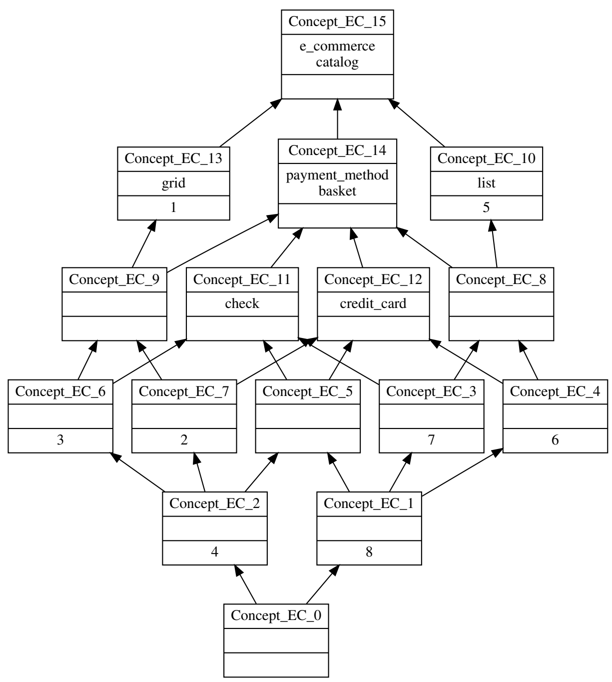


Fig. 3: Concept lattice for the formal context of Table 1, built with RCAExplore[3]

[3] http://dolques.free.fr/rcaexplore.php

A concept lattice organizes configurations and features in a single structure, which has a canonical form (only one concept lattice can be associated with a formal context). If the configurations in the formal context are the valid configurations of a feature model, the configuration semantics of the concept lattice is the same and the configurations can be read from the lattice. The logical semantics is the same too. However, the ontological semantics is incomplete as in the structure, we cannot distinguish ontological relationships: for example, when a feature F_2 is in a sub-concept of a concept that introduces another feature F_1, we cannot know whether F_2 implies F_1 (having a basket *implies* having a payment_method) or F_2 refines F_1 (pay by check *is a kind of* payment_method).

The concept lattice has many qualities regarding the variability representation and relationships between configurations, features, as well as between configurations and features, including highlighting:

- bottom features that are present in all configurations (e.g. `catalog, e_commerce`)
- mutually exclusive features (in concepts whose supremum is the top)
- feature co-occurrence (introduced in the same concept, e.g. `basket` and `payment_method`)
- feature implication (one is introduced in a sub-concept of another one, e.g. `credit_card` implies `basket`)
- how a configuration is closed to or specializes another one, or a merge of other configurations. E.g. `8` is a specialization of `5,6,7`.
- features that are specific to a configuration, or shared by many.

The concept lattice is also an interesting structure to navigate between these features and configurations, and is a theoretical support for association rule extraction, a domain that has not been explored yet in SPLE, as far as we know.

Besides, lattice theory defines irreducible elements, useful for identifying irreducible features and configurations (in a polynomial time), that are used for defining canonical representations of a context or a rule basis. In lattice theory, an element is called join-irreducible if it cannot be represented as the supremum of strictly lower elements. They are easily identifiable in a lattice because they have only one predecessor in lattice transitive reduction. All join-irreducible elements are present in the formal context, so they all correspond to valid configurations.

Research work done in the framework of reverse engineering exploits some of the relevant properties of the concept lattice. Formal Concept Analysis has been used to organize products, features, scenarios, or to synthesize information on the product line. In [13], the authors classify the usage of variable features in existing products of a product line through FCA. The analysis of the concept lattice reveals information on features that are present in all the products, none of the products, on groups of features that are always present together, and so on. Such information can be used to drive modifications on the variability management. In the same range of idea, the authors of [2] explore concept lattices as a way to represent variability in products, and revisit existing approaches to handle variability through making explicit hidden FCA aspects existing in them. The authors of [7] go a step further in the analysis of the usage of FCA, by studying

Relational Concept Analysis (RCA) as a way to analyze variability in product lines in which a feature can be a product of another product family.

Different artifacts are classified in [15]: the authors organize scenarios of a product line by functional requirements, and by quality attributes. They identify groups of functional requirements that contribute to a quality attribute, detect interferences between requirements and quality attributes, and analyze the impact of a change in the product line w.r.t functional requirement fulfillment.

Several proposals investigate with FCA the relationships between features and source code of existing products. References [4, 21] aim at locating features in source code: existing products described by source code are classified though FCA, and an analysis of the resulting concepts can detect groups of source code elements that may be candidates to reveal a feature. In the same idea, traceability links from source code to features are mined in reference [17]. In reference [9], the authors mine source code in order to identify pieces of code corresponding to a feature implementation through an FCA analysis with pieces of source code, scenarios executing those pieces of source code, and features.

FCA is also used in several approaches to study the feature organization in feature models. Concept structures (lattices or AOC-poset) are used to detect constraints in feature models, and propose a decomposition of features into sub-features. The authors of [16] extract implication rules among features, and covering properties (e.g. sets of features covering all the products). References [3, 18] produce logical relationships between the features of a FM, as well as cross-tree constraints.

Concept lattice could also be a tool in the framework of forward engineering, using a transformation chain starting from a FM, building with the existing tools, as FAMILIAR [1], the configuration set (which is equivalent to having a formal context), then building the corresponding lattice. But applying in practice this approach to the FMs repository SPLOT[4] [14], we noticed that tools hardly compute more than 1000 configurations, thus we faced a scaling problem.

4 Addressing scaling issues

4.1 From feature models dependencies to implicative systems

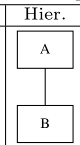
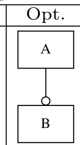
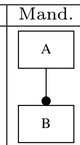
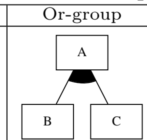
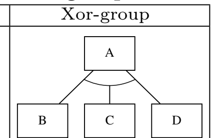
The set of all valid attribute implications of a formal context represent a closure operator, which produces attribute closed sets corresponding to concept intents of the context. The associated attribute closed set lattice is thus isomorphic to the concept lattice of the formal context. It is noteworthy that (1) FMs represent features interaction by graphically depicting a set of features and dependencies between them, and that (2) the set of all valid implications also describes dependencies between attributes (i.e. features). Thus, an analogy can be done between implicative systems and FMs dependencies.

We have previously mentioned a method to build a concept lattice from a FM, which consists in enumerating all the FM configurations (i.e. all combinations of

[4] http://www.splot-research.org/

features w.r.t its dependencies) in a formal context. However, FMs are intensional representations which can potentially depict a large number of configurations, making difficult their enumeration and the context computation. In order to avoid this enumeration, we propose a way to express FMs dependencies as sets of implications $\mathcal{P}(F) \times F$, without building the formal context. We made an experiment in which we generated several FMs of small size (< 10 features) and built their equivalent formal contexts, from which we extracted a complete set of valid implications with the tool Concept Explorer[5] [22]. When comparing these FMs to their corresponding set of implications, we noticed that each type of FM dependency generates the same kind of implications, as presented in Table 3.

Table 3: FM dependencies and their corresponding implications

	Root	Hier.	Opt.	Mand.	Or-group	Xor-group
dependencies	R					
impl.	$\emptyset \to R$	$B \to A$	None	$A \to B$	None	$BC \to ABCD$ $BD \to ABCD$ $DC \to ABCD$

The root feature traditionally represents the name of the modeled software system, and thus is present in all configurations. This peculiarity is translated by $\emptyset \to root$, requiring the presence of *root* in all closed sets. Hierarchy constraints (subpart relationships) require that a child feature can be selected only if its parent feature is already selected, and thus produce a $child \to parent$ implication. Optional relationships actually express the absence of dependencies between a feature and its child, and do not generate any implication. Mandatory relationships imply that a child feature is necessarily selected with its parent and produce a $parent \to child$ implication. Or-groups behave as optional relationships with an obligation to select at least one feature: this kind of constraints do not produce any implication. Finally, xor-groups require that two of their features cannot appear together in any configuration: each pair of features thus implies the set of all features.

We can also determine implications for cross-tree constraints, i.e. include and exclude constraints. Let F be the set of all features and $f_1, f_2 \in F$ two features. f_1 *includes* f_2 can naturally be translated by $f_1 \to f_2$, and f_1 *excludes* f_2 can be translated by $f_1\ f_2 \to F$, as in xor-groups.

Table 3 thus permits to translate FM dependencies in implicative systems without building a formal context. The fact that the obtained implicative system is exactly the system corresponding to the original FM can be proved by construction. When adding a new feature (resp. feature group) to a FM, this adds

[5] http://conexp.sourceforge.net/

new dependencies which only involve the added feature (resp. feature group) and its parent. It does not change the previous dependencies expressed in the FM, but only adds new ones. In our approach, we first identify the implications corresponding to each type of feature groups and optional/mandatory relationships. Then, if we construct the FM step by step, we can create the implications corresponding to each added feature (resp. feature group), and thus no implication is missing, nor needs to be changed afterward. We applied our method on the FM of Fig. 1 and obtained the implicative system presented in Fig. 4.

root :	$\varnothing \rightarrow Ec$
hierarchy :	$Ca \rightarrow Ec$; $G \rightarrow Ca$; $L \rightarrow Ca$;
	$Pm \rightarrow Ec$; $Cc \rightarrow Pm$; $Ch \rightarrow Pm$; $B \rightarrow Ec$
mandatory :	$Ec \rightarrow Ca$
xor-group :	$G, L \rightarrow Ec, Ca, G, L, Pm, Cc, Ch, B$
cross-tree :	$Pm \rightarrow B$; $B \rightarrow Pm$

Fig. 4: Implicative system corresponding to the feature model of Fig. 1

These implications can be extracted by performing a graph search on the FM, and their number can be predicted by analysing its dependencies (# stands for "number of"):

$1 + \#child\text{-}parent\ relationships + \#mandatory\ relationships$
$+\#pairs\ of\ features\ in\ each\ xor\text{-}group + \#cross\text{-}tree\ constraints$

For example, a representative FM of SPLOT (e-commerce) with 19 features and 768 configurations is equivalent (with the configuration-semantics) to an implicative system with 27 implications.

4.2 Identification of the set of possible configurations

In a concept lattice representing a FM, an object introduced in a concept extent represents a valid configuration, which corresponds to the feature set of the concept intent. Because each configuration in a FM is unique, a concept can introduce at most one object. Thus, for SPLE, a concept intent represents either a valid configuration or an invalid one. In the isomorphic feature closed set lattice, each closed set corresponds to a concept intent from the context: therefore, each valid configuration of the FM matches a feature closed set. However, the feature closed set lattice does not display objects and thus we cannot identify which closed set corresponds to a valid configuration. To be able to retrieve knowledge about configurations as in concept lattices, their identification in feature closed set lattice is necessary.

As previously said, all join-irreducibles correspond to valid configurations. But there can exist valid configurations which do not correspond to join-irreducibles, and thus they cannot be discerned from invalid ones in feature closed set lattices. A solution is to add a unique attribute for each configuration, as an identifier. Thus, we change the lattice structure to make each configuration correspond to a join-irreducible element, which can be detected. However, the obtained lattice is not isomorphic to the original concept lattice, and its size is larger. A way to keep the isomorphism is to add "reducible" attributes, which do not modify the lattice structure and which can label the lattice's elements.

In what follows, we investigate a way to label feature closed sets to help the identification of valid configurations. We seek to produce a labelled implicative system that generates a labelled feature closed set lattice, isomorphic to the concept lattice associated with the formal context. We recall that a valid configuration is a combination of features w.r.t. all the FM dependencies: thus, we seek to retrieve valid configurations by detecting which feature closed sets respect all these dependencies.

Features linked by mandatory relationships always appear together in closed sets: this type of dependencies is respected. Optional relationships express the absence of dependencies and do not create difficulties. Or-groups and xor-groups, however, are more problematic. Let us consider the or-group in Table 3, composed of B and C, which are two sub-features of A. $\{A, B\}$ and $\{A, C\}$ are two valid combinations of features of this group. Because our feature closed set family is closed under intersection/join, $\{A, B\} \cap \{A, C\} = \{A\}$ is also a feature closed set of the family, but it does not respect the dependencies induced by the or-group (i.e. contains at least B or C). The same reasoning can be applied to xor-groups. To identify if a feature closed set respects the dependencies induced by or-groups and xor-groups, we choose to make *constraints* related to feature groups appear directly in feature closed sets, as labels.

Let $\{f_1, \ldots, f_n\}$ be a subset of features involved in a feature group. If they form an or-group, each feature closed set containing the parent feature of this group will be labeled $(f_1, \ldots, f_n)$, defining the constraint: "this feature closed set must have at least one feature from $\{f_1, \ldots, f_n\}$ to correspond to a valid configuration". If they form a xor-group, each feature closed set containing the parent feature of the group will be labeled $[f_1, \ldots, f_n]$, defining the constraint: "this feature closed set must have exactly one feature from $\{f_1, \ldots, f_n\}$ to correspond to a valid configuration". As example, the FM of Fig. 1 produces two different labels: one for the xor-group of the feature *catalog* (Ca), and another for the or-group of the feature *payment_method* (Pm). Each feature closed set possessing *catalog* has to be labeled $[grid, list]$, and each feature closed set possessing *payment_method* has to be labeled $(check, credit\_card)$.

We choose to represent these labels in the labelled implicative system as attributes. A label is attached to a feature by adding to the original implicative system a double implication between the feature and the corresponding label-attribute. Fig. 5 presents the implications added to the implicative system of Fig. 4 in order to take into account labels $[grid, list]$ ($[G, L]$) and $(check, credit\_card)$ ((Ch, Cc)).

A feature closed set with a $(check, credit\_card)$ label is a valid configuration if it contains at least features *credit_card* or *check*. A feature closed set with a

$$labels: \quad \begin{array}{l} Pm \rightarrow (Ch, Cc)\ ;\ (Ch, Cc) \rightarrow Pm\ ; \\ Ca \rightarrow [G, L]\ ;\ [G, L] \rightarrow Ca \end{array}$$

Fig. 5: Adding labels in the implicative system of Fig. 4

$[grid, list]$ label is a valid configuration if it contains *grid* or *list*, but not both. A feature closed set respecting the constraints expressed by all its labels represents a valid configuration. A label is associated with the parent feature of the group, and thus does not change the original lattice structure.

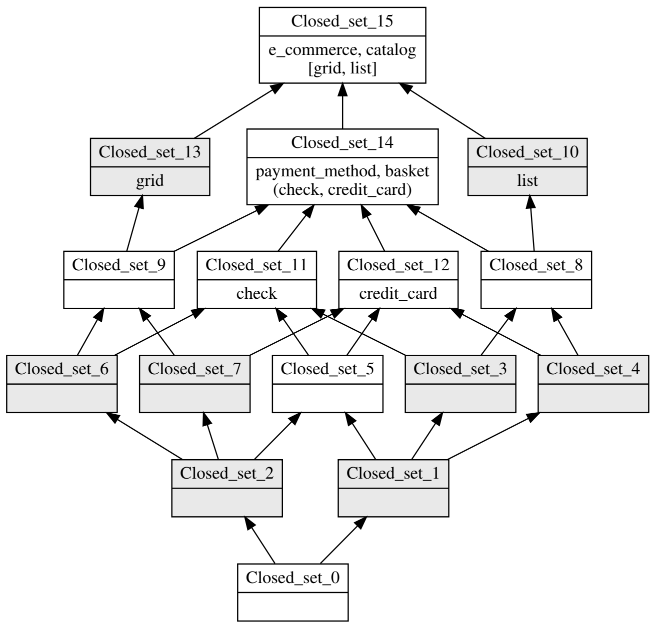


Fig. 6: Feature closed set lattice built with the implicative system of Fig. 4, labeled with implications of Fig. 5

Fig. 6 represents the feature closed set lattice associated with the labeled implicative system of Fig. 4 plus Fig. 5: feature closed sets which respect all the constraints defined by their labels are colored, and correspond to the 8 configurations of the formal context of Table 1. In the lattice, "label features" are inherited from top to bottom, as usual features. For example, *Closed_set_3* possesses features $\{e\_commerce, catalog, list, basket, payment\_method, check\}$ and the two labels $[grid, list]$ and $(check, credit\_card)$. This feature closed set possesses feature *list* and not feature *grid*, and thus respects the constraint of label $[grid, list]$. Moreover, it possesses feature *check*, and thus also respects the constraint of label $(check, credit\_card)$. The constraints corresponding to all its labels are respected: *Closed_ set_ 3* is thus a valid configuration of the software product line. Note that in this particular case, the valid configurations are all irreducible.

To conclude, the labelled implicative system permits to construct a lattice from a FM without enumerating all its configurations: the obtained feature closed set lattice is a canonical representation, isomorphic to the concept lattice of a formal context, in which one can retrieve exactly the same information about features and configurations.

5 Conclusion

In this paper, we compare the various formalisms used in the literature to represent and manage the variability of a software product line. Especially, we study their different semantics, their canonicity and the type of information they can highlight. We investigate formal concept analysis and concept lattices to represent a software product line originally described by a feature model. Contrary to FMs, concept lattices represent commonalities and variabilities in a canonical form. Moreover, they permit to extract relationships between features, between features and configurations and between configurations.

Constructing a concept lattice from a FM requires to enumerate all its configurations in a formal context, but this method can be difficult to realize when their number is too high. We propose a method to extract feature implications directly from feature models dependencies. The obtained implicative system produces a feature closed set lattice isomorphic to the concept lattice which can be built from the context. We also propose a method to label these implicative systems in order to identify the set of valid configurations, and thus retrieve the same informations as in concept lattices.

In the future, we will make experiment on the existing FMs repositories in order to assess the size of FMs, implicative systems, and closed set lattices and how frequent are the FMs that have reducible configurations. We will also expand our study to multiple software product lines. We will study relational concept analysis to connect several software product lines represented by concept lattices, and analyze their properties and the issues they permit to answer.

References

1. Acher, M., Collet, P., Lahire, P., France, R.B.: FAMILIAR: A domain-specific language for large scale management of feature models. Sci. Comput. Program. 78(6), 657–681 (2013)
2. Al-Msie 'deen, R., Seriai, A.D., Huchard, M., Urtado, C., Vauttier, S., Al-Khlifat, A.: Concept lattices: A representation space to structure software variability. In: 5th Int. Conf. on Inf. and Comm. Systems (ICICS). pp. 1 – 6 (2014)
3. Al-Msie'deen, R., Huchard, M., Seriai, A., Urtado, C., Vauttier, S.: Reverse Engineering Feature Models from Software Configurations using Formal Concept Analysis. In: 11th Int. Conf. on Concept Lattices and Their Applications (ICFCA). pp. 95–106 (2014)

4. Al-Msie'deen, R., Seriai, A., Huchard, M., Urtado, C., Vauttier, S., Salman, H.E.: Mining Features from the Object-Oriented Source Code of a Collection of Software Variants Using Formal Concept Analysis and Latent Semantic Indexing. In: 25th Conf. on Soft. Eng. and Know. Eng. (SEKE). pp. 244–249 (2013)
5. Batory, D.S.: Feature Models, Grammars, and Propositional Formulas. In: 9th Int. Conf. on Soft. Product Lines (SPLC). pp. 7–20 (2005)
6. Bryant, R.E.: Graph-Based Algorithms for Boolean Function Manipulation. IEEE Trans. Computers 35(8), 677–691 (1986)
7. Carbonnel, J., Huchard, M., Gutierrez, A.: Variability representation in product lines using concept lattices: Feasibility study with descriptions from wikipedia's product comparison matrices. In: Int. Works. on Formal Concept Analysis and Applications, FCA&A 2015, co-located with ICFCA 2015). pp. 93–108 (2015)
8. Czarnecki, K., Wasowski, A.: Feature Diagrams and Logics: There and Back Again. In: 11th Int. Conf. on Soft. Product Lines (SPLC). pp. 23–34 (2007)
9. Eisenbarth, T., Koschke, R., Simon, D.: Locating features in source code. IEEE Trans. Softw. Eng. 29(3), 210–224 (2003)
10. Ganter, B., Wille, R.: Formal Concept Analysis – Mathematical Foundations. Springer (1999)
11. Kang, K.C., Cohen, S.G., Hess, J.A., Novak, W.E., Peterson, A.S.: Feature-Oriented Domain Analysis (FODA): Feasibility Study (1990)
12. Kang, K.C., Lee, J., Donohoe, P.: Feature-oriented product line engineering. IEEE software 19(4), 58–65 (2002)
13. Loesch, F., Ploedereder, E.: Restructuring Variability in Software Product Lines using Concept Analysis of Product Configurations. In: 11th Eur. Conf. on Soft. Maintenance and Reengineering (CSMR). pp. 159–170 (2007)
14. Mendonça, M., Branco, M., Cowan, D.D.: S.P.L.O.T.: software product lines online tools. In: Companion to the 24th Annual ACM SIGPLAN Conference on Object-Oriented Programming, Systems, Languages, and Applications, OOPSLA 2009, October 25-29, 2009, Orlando, Florida, USA. pp. 761–762 (2009)
15. Niu, N., Easterbrook, S.M.: Concept analysis for product line requirements. In: 8th Int. Conf. on Aspect-Oriented Software Development (AOSD). pp. 137–148 (2009)
16. Ryssel, U., Ploennigs, J., Kabitzsch, K.: Extraction of feature models from formal contexts. In: 15th Int. Conf. on Soft. Product Lines (SPLC) Workshop Proceedings (Vol. 2). p. 4 (2011)
17. Salman, H.E., Seriai, A., Dony, C.: Feature-to-code traceability in a collection of software variants: Combining formal concept analysis and information retrieval. In: 14th Conf. on Inf. Reuse and Integration (IRI). pp. 209–216 (2013)
18. Shatnawi, A., Seriai, A.D., Sahraoui, H.: Recovering architectural variability of a family of product variants. In: 14th Int. Conf. on Soft. Reuse (ICSR). pp. 17–33 (2015)
19. She, S., Ryssel, U., Andersen, N., Wasowski, A., Czarnecki, K.: Efficient synthesis of feature models. Information & Software Technology 56(9), 1122–1143 (2014)
20. Van Gurp, J., Bosch, J., Svahnberg, M.: On the notion of variability in software product lines. In: Work. IEEE/IFIP Conf. on Soft. Arch. (WICSA). pp. 45–54 (2001)
21. Xue, Y., Xing, Z., Jarzabek, S.: Feature location in a collection of product variants. In: 19th Working Conf. on Reverse Engineering (WCRE). pp. 145–154 (2012)
22. Yevtushenko, S.A.: System of data analysis "Concept Explorer". In: 7th national conference on Artificial Intelligence KII-2000. pp. 127–134 (2000)

Characterization of Order-like Dependencies with Formal Concept Analysis

Victor Codocedo[1], Jaume Baixeries[2], Mehdi Kaytoue[1], and Amedeo Napoli[3]

[1] Université de Lyon. CNRS, INSA-Lyon, LIRIS. UMR5205, F-69621, France.
[2] Universitat Politècnica de Catalunya. 08032, Barcelona. Catalonia.
[3] LORIA (CNRS - Inria Nancy Grand Est - Université de Lorraine), B.P. 239, F-54506, Vandœuvre-lès-Nancy.

Corresponding author: mehdi.kaytoue@insa-lyon.fr

Abstract. Functional Dependencies (FDs) play a key role in many fields of the relational database model, one of the most widely used database systems. FDs have also been applied in data analysis, data quality, knowledge discovery and the like, but in a very limited scope, because of their fixed semantics. To overcome this limitation, many generalizations have been defined to relax the crisp definition of FDs. FDs and a few of their generalizations have been characterized with Formal Concept Analysis which reveals itself to be an interesting unified framework for characterizing dependencies, that is, understanding and computing them in a formal way. In this paper, we extend this work by taking into account order-like dependencies. Such dependencies, well defined in the database field, consider an ordering on the domain of each attribute, and not simply an equality relation as with standard FDs.

Keywords: functional dependencies, order dependencies, formal concept analysis

1 Introduction

Functional dependencies (FDs) are well-known constraints in the relational model used to show a functional relation between sets of attributes [12], i.e. when the values of a set of attributes are determined by the values of another set of attributes. They are also used in different tasks within the relational data model, as for instance, to check the consistency of a database, or to guide the design of a data model [10].

Different generalizations of FDs have been defined in order to deal with imprecision, errors and uncertainty in real-world data, or simply, to mine and discover more complex patterns and constraints within data when the semantics of FDs have shown to be too restrictive for modeling certain attribute domains.

id	Month	Year	Av. Temp.	City
t_1	1	1995	36.4	Milan
t_2	1	1996	33.8	Milan
t_3	5	1996	63.1	Rome
t_4	5	1997	59.6	Rome
t_5	1	1998	41.4	Dallas
t_6	1	1999	46.8	Dallas
t_7	5	1996	84.5	Houston
t_8	5	1998	80.2	Houston

Table 1

© Marianne Huchard, Sergei O. Kuznetsov (Eds.): CLA 2016, pp. 123–134, ISBN 978-5-600-01454-1, National Research University Higher School of Economics, 2016.

For example, consider the database in Table 1 as an example[4]. Attributes of these 8 tuples are city names, month identifiers, years and average temperatures. From this table, we could expect that the value for average temperature is determined by a city name and a month of the year (e.g. the month of May in Houston is hot, whereas the month of January in Dallas is cold). Therefore, we would expect that this relationship should be somehow expressed as a (functional) dependency in the form *city name, month* $\rightarrow$ *average temperature*. However, while the average temperature is truly determined by a city and a time of the year, it is very hard that it will be exactly the same from one year to another. Instead, we can expect that the value will be *similar*, or *close* throughout different years, but rarely the same. Unfortunately, semantics of FDs is based on an equivalence relation and fail to grasp the dependencies among these attributes.

To overcome the limitations of FDs while keeping the idea that some attributes are functionally determined by other attributes, different generalizations of functional dependencies have been defined, as recently deeply reviewed in a comprehensive survey [4]. Actually, the example presented in the last paragraph is a so-called *similarity dependency* [2,4]. Several other families of dependencies exist and allow relaxing the definition of FDs on the extent part (e.g. the dependency must hold only in a subset of the tuples in a database table) or on the intent part (equality between attribute values is relaxed to a similarity or *tolerance* relation).

The definition of a variation of a functional dependency shows different problems: characterization, axiomatization and computation. Formal Concept Analysis (FCA [7]) has already been used to characterize and compute functional dependencies. Moreover, in order to overcome some of the limitations of FCA to discover FDs, a more sophisticated formalization is presented in [1] and [3] where pattern structures ([6]) were used. The same framework is used in [2] to compute similarity dependencies.

In this paper we present an FCA-based characterization of order-like dependencies, a generalization of functional dependencies in which the equality of values is replaced by the notion of order. Firstly, we show that the characterization of *order dependencies* in their general definition [8] can be achieved through a particular use of general ordinal scaling [7]. Secondly, we extend our characterization in order to support *restricted order dependencies* through which other FDs generalizations can be modeled, namely sequential dependencies and trend dependencies [4]. Finally, we present a characterization to a complex FD generalization named *lexicographical ordered dependencies* [11] showing the flexibility of our approach.

The rest of this paper is organized as follows. In Section 2 we formally introduce the definition of functional dependencies, formal concept analysis and the principle of the characterization of FDs with FCA. In Section 3, we characterize *order dependencies* in their general definition. We show that our formalization can be adapted to *restricted ordered dependencies* in Section 4 and *lexicographical ordered dependencies* [11] in Section 5 before presenting our conclusions.

[4] `http://academic.udayton.edu/kissock/http/Weather/`

2 Preliminaries

2.1 Functional dependencies

We deal with datasets which are sets of tuples. Let $\mathcal{U}$ be a set of attributes and Dom be a set of values (a domain). For the sake of simplicity, we assume that Dom is a numerical set. A tuple t is a function $t : \mathcal{U} \mapsto Dom$ and then a table T is a set of tuples. We define the functional notation of a tuple for a set of attributes $X \subseteq \mathcal{U}$ as follows, assuming that there exists a total ordering on $\mathcal{U}$. Given a tuple $t \in T$ and $X = \{x_1, x_2, \ldots, x_n\}$, we have: $t(X) = \langle t(x_1), t(x_2), \ldots, t(x_n)\rangle$.

Definition 1 (Functional dependency [12]). *Let T be a set of tuples (data table), and $X, Y \subseteq \mathcal{U}$. A* functional dependency (FD) *$X \rightarrow Y$ holds in T if:*

$$\forall t, t' \in T : t(X) = t'(X) \rightarrow t(Y) = t'(Y)$$

Example. The table on the right presents 4 tuples $T = \{t_1, t_2, t_3, t_4\}$ over attributes $\mathcal{U} = \{a, b, c, d\}$. We have that $t_2(\{a, c\}) = \langle t_2(a), t_2(c)\rangle = \langle 4, 4\rangle$. Note that the set notation is usually omitted and we write ab instead of $\{a, b\}$. In this example, the functional dependency $d \rightarrow c$ holds and $a \rightarrow c$ does not hold.

id	a	b	c	d
t_1	1	3	4	1
t_2	4	3	4	3
t_3	1	8	4	1
t_4	4	3	7	8

Table 2

2.2 Formal Concept Analysis (FCA)

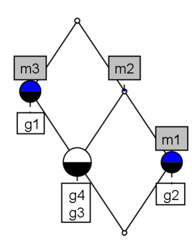
Let G and M be arbitrary sets, respectively called *objects* and *attributes*, and $I \subseteq G \times M$ an arbitrary binary relation: $(g, m) \in I$ is interpreted as "g has attribute m". (G, M, I) is called a formal context. The two following derivation operators $(\cdot)'$ define a Galois connection between the powersets of G and M.

$$A' = \{m \in M \mid \forall g \in A : gIm\} \qquad for\ A \subseteq G,$$
$$B' = \{g \in G \mid \forall m \in B : gIm\} \qquad for\ B \subseteq M$$

For $A \subseteq G$, $B \subseteq M$, a pair (A, B) such that $A' = B$ and $B' = A$, is called a *(formal) concept* while A is called the *extent* and the set B the *intent* of the concept. Concepts are partially ordered by $(A_1, B_1) \leq (A_2, B_2) \Leftrightarrow A_1 \subseteq A_2$ ($\Leftrightarrow B_2 \subseteq B_1$): the set of all formal concepts forms a complete lattice called the *concept lattice* of the formal context (G, M, I). An implication of a formal context (G, M, I) is denoted by $X \rightarrow Y$, $X, Y \subseteq M$ and means that all objects from G having the attributes in X also have the attributes in Y, i.e. $X' \subseteq Y'$. Implications obey the Armstrong rules (reflexivity, augmentation, transitivity).

Example. The table on the left presents a formal context: we have $(\{g_3\}'', \{g_3\}') = (\{g_3, g_4\}, \{m_2, m_3\})$ and the implication $m_1 \rightarrow m_2$. Its concept lattice representation involves *reduced labeling*: each node is a concept, lines represent partial ordering while an attribute (resp. object) label is inherited from the top (resp. the bottom).

	m_1	m_2	m_3
g_1			x
g_2	x	x	
g_3		x	x
g_4		x	x



2.3 Characterization of Functional Dependencies with FCA

It has been shown in previous work that functional dependencies, can be characterized with FCA. For example, Ganter & Wille [7] presented a data transformation of the initial set of tuples into a formal context. In this context, implications are in 1-to-1 correspondence with the functional dependencies of the initial dataset. In Figure 1, we illustrate this characterization with the set of tuples of Table 2. Each possible pair of tuples gives rise to an object in the formal context. Attributes remain the same. An object, say (t_i, t_j), has an attribute m iff $t_i(m) = t_j(m)$. The concept lattice is given on the right hand side of this figure: there are two implications, namely $d \to c$ and $d \to a$, which are also the functional dependencies in the original set of tuples.

However, this approach implies that a formal context much larger than the original dataset must be processed. It was then shown that this formal context can actually be encoded with a pattern structure [6]: each attribute of the original dataset becomes an object of the pattern structure and is described by a partition on the tuple set. Actually, each block of the partition is composed of tuples taking the same value for the given attribute [9]. For example, in Table 2, the partition describing a is $\{\{t_1, t_3\}, \{t_2, t_4\}\}$. Then, the implications in the pattern concept lattice are here again in 1-to-1 correspondence with the functional dependencies of the initial dataset [3]. What is important to notice is that this formalization is possible as a partition is an *equivalence relation*: a symmetric, reflexive and transitive binary relation.

In [2], another kind of dependencies was formalized in a similar way, i.e. similarity dependencies, where the equality relation is relaxed to a similarity relation when comparing two tuples. An attribute is not anymore described by a partition, but by a tolerance relation, i.e. a symmetric, reflexive, but not necessarily transitive binary relation (*"the friends of my friends are not necessarily my friends"*). Each original attribute is then described by a set of tolerance blocks, each being a maximal set of tuples that have pairwise similar values (instead of equal values for classical dependencies).

This way of characterizing FDs and similarity dependencies actually fails for order dependencies, as the relation in this case is not symmetric: it is neither an equality nor a similarity but a partial order in the general case.

3 Characterization of Order Dependencies with FCA

Although functional dependencies are used in several domains they cannot be used to express some relationships that exist in data. Many generalizations have been proposed and we focus in this article on order dependencies [8,4]. Such dependencies are based on the *attribute-wise* order on tuples. This order assumes that each attribute follows a partial order associated to the values of its domain. For the sake of generality, we represent this order with the symbol $\sqsubseteq_x$ for all $x \in \mathcal{U}$. In practice, this symbol will be instantiated by intersections of any partial order on the domain of this attribute, as, for instance, $<, \leq, >, \geq$, etc.

id	a	b	c	d
t_1	1	3	4	1
t_2	4	3	4	3
t_3	1	8	4	1
t_4	4	3	7	8

$\mathbb{K}$	a	b	c	d
(t_1, t_2)		×	×	
(t_1, t_3)	×		×	×
(t_1, t_4)		×		
(t_2, t_3)			×	
(t_2, t_4)	×	×		
(t_3, t_4)				

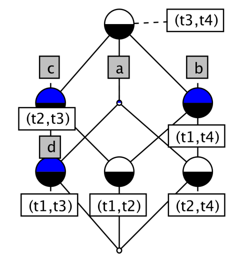


Fig. 1: Characterizing functional dependencies with FCA.

We remark that this order on the set of values of a *single* attribute does not need to be a total order, although in many different instances, like numeric or character strings domains, this will be the case. In the following, we formalize operator $\sqsubseteq_x$ (Definition 2) and define accordingly order dependencies (Definition 3).

Definition 2 (Attribute-wise ordering). *Given two tuples $t_i, t_j \in T$ and a set of attributes $X \subseteq \mathcal{U}$, the attribute-wise order of these two tuples on X is:*

$$t_i \sqsubseteq_X t_j \Leftrightarrow \forall x \in X : t_i[x] \sqsubseteq_x t_j[x]$$

This definition states that one tuple is *greater* –in a sense involving the order of all attributes– than another tuple if their attribute-wise values meet this order. This operator induces a partial order $\Pi_X = (T, \prec_X)$ on the set T of tuples.

Definition 3 (Order dependency). *Let $X, Y \subseteq \mathcal{U}$ be two subsets of attributes in a dataset T. An order dependency $X \to Y$ holds in T if and only if:*

$$\forall t_i, t_j \in T : t_i \sqsubseteq_X t_j \to t_i \sqsubseteq_Y t_j$$

Example. Consider the table on the right with six tuples and three attributes. Taking $\sqsubseteq_a$, $\sqsubseteq_b$ and $\sqsubseteq_c$ defined as the ordering $\leq$. The orders induced by the sets of attributes $\{a\}$,$\{b\}$,$\{c\}$ and $\{a, b\}$ are:

$\Pi_a = (T, \prec_a) \quad = \{\{t_1\} \prec \{t_2\} \prec \{t_3, t_6\} \prec \{t_5\} \prec \{t_4\}\}$
$\Pi_b = (T, \prec_b) \quad = \{\{t_5\} \prec \{t_1, t_4\} \prec \{t_3\} \prec \{t_2\} \prec \{t_6\}\}$
$\Pi_{ab} = (T, \prec_{ab}) = \{\{t_1\} \prec \{t_2\} \prec \{t_6\};$
$\{t_1\} \prec \{t_3\};$
$\{t_1\} \prec \{t_4\};$
$\{t_5\} \prec \{t_4\}\}$
$\Pi_c = (T, \prec_c) \quad = \{\{t_1\} \prec \{t_2\} \prec \{t_3, t_6\} \prec \{t_5\} \prec \{t_4\}\}$

id	a	b	c
t_1	1	3	1
t_2	2	7	2
t_3	3	4	4
t_4	5	3	9
t_5	4	2	5
t_6	3	8	4

Table 3

These orders are such that the order dependency $\{a, b\} \to \{c\}$ holds. Remark that Definition 3 is generic since the orders that are assumed for each attribute need to be instantiated: we chose $\leq$ in this example for all attributes, while taking the equality would produce standard *functional dependencies*.

To achieve the characterization of order dependencies with FCA, we propose to represent the partial order $\Pi_X = (T, \prec_X)$ associated to each subset of attribute $X \subseteq \mathcal{U}$ as a formal context $\mathbb{K}_X$ (a binary relation on $T \times T$ thanks to a general ordinal scaling [7]). Then, we show that an order dependency $X \to Y$ holds iff $\mathbb{K}_X = \mathbb{K}_{XY}$.

Definition 4 (General ordinal scaling of the tuple set). *Given a subset of attributes $X \subseteq \mathcal{U}$ and a table dataset T, we define a formal context for $\Pi_X = (T, \prec_X)$ (the partial order it induces) as follows:*

$$\mathbb{K}_X = (T, T, \sqsubset_X)$$

where $\sqsubset_X = \{(t_i, t_j) \mid t_i, t_j \in T, t_i \sqsubseteq_X t_j\}$. This formal context is the ***general ordinal scale*** *of Π_X [7]. All formal concepts $(A, B) \in \mathbb{K}_X$ are such that A is the set of lower bounds of B and B is the set of upper bounds of A. Its concept lattice is the smallest complete lattice in which the order Π_X can be order embedded.*

This way to characterize a partial order is only one among several possibilities. However, the choice of formal contexts is due to their versatility, since they can characterize binary relations, hierarchies, dependencies, different orders [7] and graphs [5]. In the next section we will see how this versatility allows us to generalize similarity dependencies. Given the set of attributes $X \subseteq \mathcal{U}$, an associated partial order $\Pi_X = (T, \prec_X)$ and the formal context $(T, T, \sqsubset_X)$, it is easy to show that the later is a composition of contexts defined as: $(T, T, \sqsubset_X) = (T, T, \bigcap_{x \in X} \sqsubset_x)$.

We can now propose a characterization of order dependencies with FCA.

Proposition 1. *An order dependency $X \to Y$ holds in T iff $\mathbb{K}_X = \mathbb{K}_{XY}$.*

Proof. Recall that $\mathbb{K}_{XY} = (T, T, \sqsubset_{XY}) = (T, T, \sqsubset_X \cap \sqsubset_Y)$. We have that

$$\begin{aligned} X \to Y &\iff \sqsubset_X = \sqsubset_X \cap \sqsubset_Y \\ &\iff \sqsubset_X \subseteq \sqsubset_Y \\ &\iff \forall t_i, t_j \in T,\ t_i \sqsubseteq_X t_j \to t_i \sqsubseteq_Y t_j \end{aligned}$$

Example. To calculate $(T, T, \sqsubset_{ab})$, we just need to calculate $(T, T, \sqsubset_a \cap \sqsubset_b)$, as illustrated in the example below.

$\sqsubset_a$	t_1	t_2	t_3	t_4	t_5	t_6
t_1		×	×	×	×	×
t_2			×	×	×	×
t_3				×	×	
t_4						
t_5				×		
t_6				×	×	

Table 4: $(T, T, \sqsubset_a)$

$\sqsubset_b$	t_1	t_2	t_3	t_4	t_5	t_6
t_1		×	×			×
t_2						×
t_3		×				×
t_4		×	×			×
t_5	×	×	×	×		×
t_6						

Table 5: $(T, T, \sqsubset_b)$

$\sqsubset_{ab}$	t_1	t_2	t_3	t_4	t_5	t_6
t_1		×	×			×
t_2						×
t_3						
t_4						
t_5				×		
t_6						

Table 6: $(T, T, \sqsubset_{ab})$

The fact that the order dependency $\{a, b\} \rightarrow \{c\}$ holds can be illustrated with the formal contexts in Tables 7,8 and 9. We have indeed that $\mathbb{K}_{ab} = \mathbb{K}_{abc}$.

$\sqsubset_c$	t_1	t_2	t_3	t_4	t_5	t_6
t_1		×	×	×	×	×
t_2			×	×	×	×
t_3				×	×	×
t_4						
t_5				×		
t_6				×	×	

Table 7: $(T, T, \sqsubset_c)$

$\sqsubset_{ab}$	t_1	t_2	t_3	t_4	t_5	t_6
t_1		×	×			×
t_2						×
t_3						
t_4						
t_5				×		
t_6						

Table 8: $(T, T, \sqsubset_{ab})$

$\sqsubset_{abc}$	t_1	t_2	t_3	t_4	t_5	t_6
t_1		×	×			×
t_2						×
t_3						
t_4						
t_5				×		
t_6						

Table 9: $(T, T, \sqsubset_{abc})$

Order dependencies and other FDs generalizations. We have seen that the definition of order dependencies replaces the equality condition present in FDs or other similarity measures present in other dependencies, by an order relation. This may suggest that order dependencies and other kinds of FDs generalizations are structurally very similar, whereas this is not the case. Functional dependencies generate a reflexive, symmetric and transitive relation in the set of tuples, i.e. an equivalence relation. Then the set of tuples can be partitioned into *equivalence classes* that are used to characterize and compute the set of FDs holding in a dataset, as presented in a previous work [3].

In the generalization of functional dependencies that replaces the equality condition by a *similarity* measure or a distance function, this measure generates a symmetric relation in the set of tuples, but not necessarily a transitive relation. In turn, this implies that the set of tuples can be partitioned into *blocks of tolerance* instead of equivalence classes, as shown in [2].

In this article, the novelty is that we are dealing with a transitive relation, but not necessarily a symmetric relation. That means that we are not dealing with equivalence classes nor blocks of tolerance any longer, but, precisely, with orders. And since the characterization of these dependencies cannot be performed in terms of equivalence classes nor blocks of tolerance, will use a more general approach: *general ordinal scaling.*

4 Characterization of Restricted Order Dependencies

Order dependencies allow taking into account the ordering of the values of each attribute when looking for dependencies in data. However, violations of the ordering due to value variations should sometimes not be considered in many real world scenarios. Consider the example given in Table 10: it gives variations on the number of people waiting at a bus station over time. In such a scenario we can expect that more people will be waiting in the station as time moves on ($People\_waiting \rightarrow Time$). However, at some point, a bus arrives and the number of people waiting decreases and starts increasing again. It is easy to observe that the order dependency $People\_waiting \rightarrow Time$

	Time	*People_waiting*
t_1	10:00	101
t_2	10:20	103
t_3	10:40	105
t_4	11:00	77
t_5	11:20	80
t_6	11:40	85

Table 10

does not hold as we have the counter-example:

$$t_4 \sqsubseteq_{People\_waiting} t_3 \text{ and } t_3 \sqsubseteq_{Time} t_4$$

However, the gap between the values 77 and 105 is significant enough to be considered as a different instance of the ordering. We can formalize this idea by introducing a *similarity threshold* $\theta = 10$ for the attribute $People\_waiting$ such that the ordering between values is checked iff the difference is smaller than θ. In this way, the previous counter-example is avoided (*restricting* the binary relation) along with any other counter-example and we have that the restricted order dependency $People\_waiting \rightarrow Time$ holds.

We now formalize the tuple ordering relation, and consequently the notion of restricted order dependencies.

Definition 5. *Given two tuples $t_i, t_j \in T$ and a set of attributes $X \subseteq \mathcal{U}$, the attribute-wise order of these two tuples on X is:*

$$t_i \sqsubseteq^*_x t_j \Leftrightarrow \forall x \in X : 0 \leq t_j[x] - t_i[x] \leq \theta_x$$

Definition 6. *Let $X, Y \subseteq \mathcal{U}$ two sets of attributes in a table T such that $|T| = n$, and let θ_X, θ_Y be thresholds values of tuples in X and Y respectively. A restricted order dependency $X \rightarrow Y$ holds in T if and only if:*

$$t[X] \sqsubseteq^*_X t'[X] \rightarrow t[Y] \sqsubseteq^*_Y t'[Y]$$

Using these definitions we can encode the tuple ordering relations as formal contexts for any subset of attributes $X \subseteq \mathcal{U}$. Indeed, the binary relations between tuples by operator $\sqsubseteq^*_X$ can be encoded in a formal context $\mathbb{K}^*_X = (T, T, \sqsubseteq^*_X)$ which in turn, can be composed from single attributes $x \in \mathcal{U}$ as follows:

$$\sqsubseteq^*_X = \bigcap_{x \in X} \sqsubseteq^*_x$$

Moreover, we can use the same rationale we used to mine order dependencies to find restricted order dependencies.

Proposition 2. *A restricted order dependency $X \rightarrow Y$ holds in T iff*

$$X \rightarrow Y \iff \mathbb{K}^*_X = \mathbb{K}^*_{XY}$$

Proof. This proposition can be proved similarly to Proposition 1.

Example. For the previous example, we calculate the corresponding formal contexts shown in Tables 11 and 12 ($\sqsubseteq^*_{Tm}$ for *Time*, and $\sqsubseteq^*_{Pp}$ for *People_waiting*). It is easy to observe that the restricted order dependency $People\_waiting \rightarrow Time$ holds as we have that $\mathbb{K}^*_{Pp} = \mathbb{K}^*_{Pp,Tm}$.

$\sqsubseteq^*_{Tm}$	t_1	t_2	t_3	t_4	t_5	t_6
t_1	×	×	×	×	×	×
t_2		×	×	×	×	×
t_3			×	×	×	×
t_4				×	×	×
t_5					×	×
t_6						×

Table 11: $(T, T, \sqsubseteq^*_{Tm})$

$\sqsubseteq^*_{Pp}$	t_1	t_2	t_3	t_4	t_5	t_6
t_1	×	×	×			
t_2		×	×			
t_3			×			
t_4				×	×	×
t_5					×	×
t_6						×

Table 12: $(T, T, \sqsubseteq^*_{Pp})$

$\sqsubseteq^*_{Tm,Pp}$	t_1	t_2	t_3	t_4	t_5	t_6
t_1	×	×	×			
t_2		×	×			
t_3			×			
t_4				×	×	×
t_5					×	×
t_6						×

Table 13: $(T, T, \sqsubseteq^*_{Tm,Pp})$

Restricted order dependencies and other FDs generalizations. Similarity dependencies (SDs) generalize functional dependencies through the use of a tolerance relation instead of an equivalence relation between values of tuples for a given set of attributes. A tolerance relation is a reflexive, symmetric and non-transitive binary association between two tuples given a threshold θ. In a nutshell, a SD is established between two tuples if their values are within a given *distance* controlled by the threshold. Such dependencies were studied in a previous work [2]. However, from the perspective of order dependencies, we can request that such distance has a certain polarity. As we have previously discussed, order dependencies arise from anti-symmetric, not necessarily reflexive, and transitive binary relations ($<, \leq$). Then, it can be expected that using a *threshold of distance* θ between tuple values for a given set of attributes requires an antisymmetric, non-transitive relation between the values of tuples w.r.t. a set of attributes X, that we have defined as $\sqsubseteq^*_X$.

Observe that the difference between Definition 5 and tolerance relations is the drop of the absolute value for $t_j[x] - t_i[x]$ and the requirement that this is a positive number, i.e. $t_i[x] < t_j[x], \forall x \in X$. There is an important difference between this setting and SDs. While in SDs the threshold θ is used to *relax* the strict equivalence condition of standard functional dependencies, from the perspective of order dependencies the threshold is actually used to *restrict* tuple relations.

Restricted order dependencies have the potential to implement some other generalizations of FDs such as sequential dependencies and trend dependencies [4]. The latter is actually a particular case of restricted order dependencies where the threshold is applied to an attribute not contained in the attributes of the dependency. Instead, it is applied to a *time* attribute that allows defining a *snapshot* of the database. In sequential dependencies, the antecedent is a mapping (ρ) of a set of attributes with a *complete unrestricted* order (without a threshold). Currently, we are able to support some instances of sequential dependencies when the mapping is symmetric ($\rho(XY) = \rho(YX)$). Details on this matter has been left out from this paper for space reasons. Rather, in the following section we describe another dependency which is not symmetric, namely lexicographical ordered dependencies (LODs), that exemplifies the flexibility of our approach to support complex dependency definitions.

5 Lexicographical ordered dependencies LODs

LODs use the notion of lexicographical ordering in a rather unconventional manner[5]. While it could be expected that they compare the values of different attributes using lexicographical order, instead new descriptions (or *projections*) are composed from the Cartesian product of attribute domains on which the lexicographical order is applied [11]. Consequently, the order in which we compose new descriptions becomes relevant.

[5] Consider lexicographical order as the order of words in a dictionary.

For example, in Table 14 we compose a description using the ordered set $X = \langle b, e \rangle$ such that the new descriptions or *projections* of tuples t_1, t_2 on X are $t_1^X =< 3, 5 >$ and $t_2^X =< 4, 0 >$ respectively. It is clear that t_1^X is lexicographically lower than t_2^X. However, considering $Y = \langle e, b \rangle$ which is the inverse of X, we have $t_1^Y = 53$ and $t_2^Y = 04$ where t_2^Y is lexicographically lower than t_1^Y. Definition 7 formalizes lexicographical ordering for tuple projections.

Definition 7 ([11]). *Let $X \subseteq M$ be an ordered set, such that $n = |X|$, and let $[1, n]$ be the set of indices of the ordered set X. A lexicographical ordering on X, denoted by $\leq_X^l$ is defined for t_1^X, t_2^X as $t_1^X \leq_X^l t_2^X$, if either:*

1. $\exists k \in [1, n]$ *s.t.* $t_1^X[k] < t_2^X[k]$ *and* $t_1^X[j] = t_2^X[j]$ *with* $j \in [1, k[$.
2. $t_1^X[i] = t_2^X[i], \forall i \in [1, n]$

The main difference between LODs and standard order dependencies is that LODs are established over *ordered sets* and thus, the LOD $\langle a, b \rangle \rightsquigarrow \langle c \rangle$ does not imply that $\langle b, a \rangle \rightsquigarrow \langle c \rangle$ holds, where $\rightsquigarrow$ is used to denote a LOD. Definition 8 formalizes lexicographical order dependencies.

Definition 8 ([11]). *Let $X, Y \subseteq \mathcal{U}$ be two ordered attribute sets. A LOD, $X \rightsquigarrow Y$ is satisfied iff for all $t_1, t_2 \in r$, $t_1^X \leq_X^l t_2^X$ implies that $t_1^Y \leq_Y^l t_2^Y$.*

In Table 14, we have the LOD $\langle c, a, b \rangle \rightsquigarrow \langle d, e \rangle$ which can be verified as follows. Let $X = \langle c, a, b \rangle$ and $Y = \langle d, e \rangle$.

$$(t_1^X = 123) \leq_X^l (t_2^X = 124) \rightarrow (t_1^Y = 45) \leq_X^l (t_2^Y = 60)$$

	a	b	c	d	e
t_1	2	3	1	4	5
t_2	2	4	1	6	0

Table 14: Example

$\leq_{\langle e,d \rangle}^l$	t_1	t_2
t_1	×	
t_2	×	×

Table 15: $\mathbb{K}_{ed}^l$

$\leq_{\langle d,e \rangle}^l$	t_1	t_2
t_1	×	×
t_2		×

Table 16: $\mathbb{K}_{de}^l$

As pointed out in [11], a LOD between single attributes is necessarily an order dependency (with a single attribute there is only one order). Furthermore, given point 2 of Definition 7, all functional dependencies are also LODs. This includes the permutations of the antecedent and the consequent of a FD.

In our setting, we have described that a context can be build to encode tuple relations for a given order operator (e.g. $\sqsubseteq_X, \sqsubseteq_X^*$) w.r.t. a set of attributes X. Regarding LODs, this cannot be the case as the set of attributes X is required to be *ordered*, meaning that the context $\mathbb{K}_{xy}^l$ is not necessarily the same as the context $\mathbb{K}_{yx}^l$ for $x, y \in \mathcal{U}$ ($\mathbb{K}^l$ indicates a formal context encoding a lexicographical ordering $\leq^l$). For example, from Table 14 we can build a formal context encoding $\leq_{\langle e,d \rangle}^l$ (shown in Table 15) where $t_2 \leq_{\langle e,d \rangle}^l t_1$, and a different formal context encoding $\leq_{\langle d,e \rangle}^l$ (shown in Table 16) where $t_1 \leq_{\langle d,e \rangle}^l t_2$.

Nevertheless, a close inspection to a generic LOD $X \rightsquigarrow Y$ reveals that it requires a series of order-like dependencies to hold to be satisfied. For example, if $X \rightsquigarrow Y$ then the functional dependency $X \rightarrow Y$ holds. We can prove this as follows. Consider two tuples such that $t_i^X = t_j^X$ (their projections w.r.t. X are equivalent). Then, $t_i \leq_X^l t_j$ and $t_j \leq_X^l t_i$ which, if $X \rightsquigarrow Y$ holds, implies that $t_i \leq_Y^l t_j$ and $t_j \leq_Y^l t_i$ which is the same as $t_i^Y = t_j^Y$. Clearly, $t_i^X = t_j^X$ regardless the order of attributes in X and thus we have a functional dependency $X \to Y$. Now, consider the first attribute of X, x_1 and the first attribute of Y, y_1. Necessarily, $t[x_1] < t'[x_1] \to t[y_1] \leq t'[y_1]$ which is an order-like dependency between x_1 and y_1. Similar analysis can be used to obtain sufficient rules so $X \rightsquigarrow Y$.

For the sake of brevity, we describe a simple algorithm verifying that the LOD $X \rightsquigarrow Y$ holds by checking a *cascade* of order-like dependencies obtained using the previous analysis. Table 5 presents an example of this algorithm applied to a table containing a LOD. For each attribute $x \in \mathcal{U}$ we generate three different formal contexts, namely $\mathbb{K}_x^{=} = (T, T, =_x)$, $\mathbb{K}_x^{<} = (T, T, <_x)$ and $\mathbb{K}_x^{\leq} = (T, T, \leq_x)$. Then, we proceed as follows:

– Check functional dependency $X \to Y$
– For the i-th element of Y, y_i:
  • Build $\mathbb{K}_\psi = \mathbb{K}_{y_1}^{=} \cap \mathbb{K}_{y_2}^{=} \cap \mathbb{K}_{y_3}^{=} \cap \cdots \cap \mathbb{K}_{y_{i-1}}^{=}$
  • For the j-th element of X, x_j:
    ∗ Build $\mathbb{K}_\chi = \mathbb{K}_{x_1}^{=} \cap \mathbb{K}_{x_2}^{=} \cap \mathbb{K}_{x_3}^{=} \cap \cdots \cap \mathbb{K}_{x_{j-1}}^{=}$
    ∗ Check the order-like dependency $(\mathbb{K}_\chi \cap \mathbb{K}_{x_j}^{<} \cap \mathbb{K}_\psi) \subseteq \mathbb{K}_{y_i}^{\leq}$

Checking $\langle a, b\rangle \rightsquigarrow \langle c, d\rangle$:

– $\mathbb{K}_{ab}^{=}$ is empty and the FD $ab \to cd$ holds.
– For the first element of $\langle c, d\rangle$:
  • $\mathbb{K}_a^{<} = \{(t_1, t_2)\} \subseteq \mathbb{K}_c^{\leq} = \{(t_1, t_2), (t_1, t_3), (t_2, t_3), (t_3, t_2)\}$
  • $\mathbb{K}_a^{=} \cap \mathbb{K}_b^{<} = \{(t_2, t_3)\} \subseteq \mathbb{K}_c^{\leq} = \{(t_1, t_2), (t_1, t_3), (t_2, t_3), (t_3, t_2)\}$
– For the second element of element of $\langle c, d\rangle$:
  • $\mathbb{K}_a^{<} \cap \mathbb{K}_c^{=} = \emptyset \subseteq \mathbb{K}_d^{\leq} = \{(t_2, t_3)\}$
  • $\mathbb{K}_a^{=} \cap \mathbb{K}_b^{<} \cap \mathbb{K}_c^{=} = \{(t_2, t_3)\} \subseteq \mathbb{K}_d^{\leq} = \{(t_2, t_3)\}$

	a	b	c	d
t_1	1	?	1	?
t_2	2	1	2	1
t_3	2	2	2	2

Table 17: Example

6 Conclusion

Different generalizations of functional dependencies have been defined. The definition of a new kind of generalization of functional dependencies needs to cover two different aspects: axiomatization and computation.

We have presented a characterization of order dependencies with FCA, which can be potentially extended to other types of order-like dependencies, and are used in many fields in database theory, knowledge discovery and data quality.

These dependencies are part of a set of functional dependencies generalizations where equality condition is replaced with a more general relation. In some cases, the equality is replaced by an approximate measure, in other cases, like in order dependencies, by an order relation.

We have seen that order dependencies are based on a transitive, but not necessarily symmetric relation, contrasting approximate dependencies, which are based on a symmetric, but not necessarily transitive relation. It is precisely this formalization in terms of FCA that allows us to find these structural differences between these types of dependencies. We have also seen that this same characterization can be extended to other kinds of approximate dependencies.

This characterization allows us to cover also the computation of these dependencies: they can use the different algorithms that already exist in FCA.

This present work needs to be extended to other kinds of order-like dependencies, and some experimentation needs to be performed in order to verify the computational feasibility of this approach.

Acknowledgments. This research work has been supported by the SGR2014-890 (MACDA) project of the Generalitat de Catalunya, and MINECO project APCOM (TIN2014-57226-P) and partially funded by the French National Project FUI AAP 14 Tracaverre 2012-2016.

References

1. J. Baixeries, M. Kaytoue, and A. Napoli. Computing functional dependencies with pattern structures. In L. Szathmary and U. Priss, editors, *CLA*, volume 972 of *CEUR Workshop Proceedings*, pages 175–186. CEUR-WS.org, 2012.
2. J. Baixeries, M. Kaytoue, and A. Napoli. Computing similarity dependencies with pattern structures. In M. Ojeda-Aciego and J. Outrata, editors, *CLA*, volume 1062 of *CEUR Workshop Proceedings*, pages 33–44. CEUR-WS.org, 2013.
3. J. Baixeries, M. Kaytoue, and A. Napoli. Characterizing Functional Dependencies in Formal Concept Analysis with Pattern Structures. *Annals of Mathematics and Artificial Intelligence*, 72(1-2):129–149, Oct. 2014.
4. L. Caruccio, V. Deufemia, and G. Polese. Relaxed functional dependencies - A survey of approaches. *IEEE Trans. Knowl. Data Eng.*, 28(1):147–165, 2016.
5. S. Ferré. A Proposal for Extending Formal Concept Analysis to Knowledge Graphs. In *Formal Concept Analysis*, volume LNCS 9113, pages 271–286, Nerja, Spain, June 2015.
6. B. Ganter and S. O. Kuznetsov. Pattern structures and their projections. In H. S. Delugach and G. Stumme, editors, *Conceptual Structures: Broadening the Base, Proceedings of the 9th International Conference on Conceptual Structures (ICCS 2001)*, LNCS 2120, pages 129–142. Springer, 2001.
7. B. Ganter and R. Wille. *Formal Concept Analysis*. Springer, Berlin, 1999.
8. S. Ginsburg and R. Hull. Order dependency in the relational model. *Theoretical Computer Science*, 26(1):149 – 195, 1983.
9. Y. Huhtala, J. Kärkkäinen, P. Porkka, and H. Toivonen. Tane: An efficient algorithm for discovering functional and approximate dependencies. *Computer Journal*, 42(2):100–111, 1999.
10. H. Mannila and K.-J. Räihä. *The Design of Relational Databases*. Addison-Wesley, Reading (MA), USA, 1992.
11. W. Ng. Ordered functional dependencies in relational databases. *Information Systems*, 24(7):535 – 554, 1999.
12. J. Ullman. *Principles of Database Systems and Knowledge-Based Systems, volumes 1–2*. Computer Science Press, Rockville (MD), USA, 1989.

Reducts in Multi-Adjoint Concept Lattices

Maria Eugenia Cornejo[1], Jesús Medina[2], and Eloísa Ramírez-Poussa[2]

[1] Department of Statistic and O.R., University of Cádiz. Spain
Email: mariaeugenia.cornejo@uca.es
[2] Department of Mathematics, University of Cádiz. Spain
Email: {jesus.medina,eloisa.ramirez}@uca.es

Abstract. Removing redundant information in databases is a key issue in Formal Concept Analysis. This paper introduces several results on the attributes that generate the meet-irreducible elements of a multi-adjoint concept lattice, in order to provide different properties of the reducts in this framework. Moreover, the reducts of particular multi-adjoint concept lattices have been computed in different examples.
Keywords: attribute reduction, reduct, multi-adjoint concept lattice

1 Introduction

Attribute reduction is an important research topic in Formal Concept Analysis (FCA) [1, 4, 10, 15]. Reducts are the minimal subsets of attributes needed in order to compute a lattice isomorphic to the original one, that is, that preserve the whole information of the original database. Hence, the computation of these sets is very interesting. For example, they are useful in order to obtain attribute implications and, since the complexity to build concept lattices directly depend on the number of attributes and objects, if a reduct can be detected before computing the whole concept lattice, the complexity will significantly be decreased.

Different fuzzy extensions of FCA have been introduced [2, 3, 9, 14]. One of the most general is the multi-adjoint concept lattice framework [11, 12]. Based on a characterization of the meet-irreducible elements of a multi-adjoint concept lattice, a suitable attribute reduction method has recently been presented in [6]. In this paper the notions of absolutely necessary, relatively necessary and absolutely unnecessary attribute, as in Rough Set Theory (RST) [13], have been considered in order to classify the set of attributes. This classification provides a procedure to know whether an attribute should be considered or not. Consequently, it can be used to extract reducts. In addition, when the attribute classification verifies that the set of relatively necessary attributes is not empty several reducts can be obtained.

Due to the relation between the given attribute classification and the meet-irreducible elements of a concept lattice, this paper studies the attributes that generate the meet-irreducible elements of a multi-adjoint concept lattice. From the introduced results, different properties of the corresponding reducts have

© Marianne Huchard, Sergei O. Kuznetsov (Eds.): CLA 2016, pp. 135–146,
ISBN 978-5-600-01454-1, National Research University Higher School of Economics, 2016.

been presented. In addition, two examples in which the reducts of particular multi-adjoint concept lattices have been included.

2 Preliminaries

A brief summary with the basic notions and results related to attribute classification in the fuzzy framework of multi-adjoint concept lattices is presented.

2.1 Multi-adjoint concept lattices

First of all, we will recall the definitions of multi-adjoint frame and context where the operators to carry out the calculus are adjoint triples [7, 8].

Definition 1. *A* multi-adjoint frame *is a tuple* $(L_1, L_2, P, \&_1, \dots, \&_n)$ *where* $(L_1, \preceq_1)$ *and* $(L_2, \preceq_2)$ *are complete lattices,* $(P, \leq)$ *is a poset and* $(\&_i, \swarrow^i, \nwarrow_i)$ *is an adjoint triple with respect to* L_1, L_2, P*, for all* $i \in \{1, \dots, n\}$.

Definition 2. *Let* $(L_1, L_2, P, \&_1, \dots, \&_n)$ *be a multi-adjoint frame, a* context *is a tuple* (A, B, R, σ) *such that* A *and* B *are nonempty sets (usually interpreted as attributes and objects, respectively),* R *is a* P*-fuzzy relation* $R\colon A \times B \to P$ *and* $\sigma\colon A \times B \to \{1, \dots, n\}$ *is a mapping which associates any element in* $A \times B$ *with some particular adjoint triple in the frame.*

In order to introduce the *multi-adjoint concept lattice* associated with this frame and this context, two concept-forming operators ${}^\uparrow\colon L_2^B \to L_1^A$ and ${}^\downarrow\colon L_1^A \to L_2^B$ are considered. These operators are defined as

$$g^\uparrow(a) = \inf\{R(a, b) \swarrow^{\sigma(a,b)} g(b) \mid b \in B\} \tag{1}$$

$$f^\downarrow(b) = \inf\{R(a, b) \nwarrow_{\sigma(a,b)} f(a) \mid a \in A\} \tag{2}$$

for all $g \in L_2^B$, $f \in L_1^A$ and $a \in A$, $b \in B$, where L_2^B and L_1^A denote the set of mappings $g\colon B \to L_2$ and $f\colon A \to L_1$, respectively, which form a Galois connection [12].

By using the concept-forming operators, a *multi-adjoint concept* is defined as a pair $\langle g, f\rangle$ with $g \in L_2^B$, $f \in L_1^A$ satisfying $g^\uparrow = f$ and $f^\downarrow = g$. The fuzzy subsets of objects g (resp. fuzzy subsets of attributes f) are called *extensions* (resp. *intensions*) of the concepts.

Definition 3. *The* multi-adjoint concept lattice *associated with a multi-adjoint frame* $(L_1, L_2, P, \&_1, \dots, \&_n)$ *and a context* (A, B, R, σ) *given, is the set*

$$\mathcal{M} = \{\langle g, f\rangle \mid g \in L_2^B, f \in L_1^A \text{ and } g^\uparrow = f, f^\downarrow = g\}$$

where the ordering is defined by $\langle g_1, f_1\rangle \preceq \langle g_2, f_2\rangle$ *if and only if* $g_1 \preceq_2 g_2$ *(equivalently* $f_2 \preceq_1 f_1$*).*

A classification of the attributes of a multi-adjoint context from a characterization of the $\wedge$-irreducible elements of the corresponding concept lattice $(\mathcal{M}, \preceq)$ was given in [5, 6]. Before introducing this classification, the characterization theorem must be recalled. First and foremost, it is necessary to define the following specific family of fuzzy subsets of attributes.

Definition 4. *For each $a \in A$, the fuzzy subsets of attributes $\phi_{a,x} \in L_1^A$ defined, for all $x \in L_1$, as*

$$\phi_{a,x}(a') = \begin{cases} x & \text{if } a' = a \\ \bot_1 & \text{if } a' \neq a \end{cases}$$

will be called fuzzy-attributes*, where $\bot_1$ is the minimum element in L_1. The set of all fuzzy-attributes will be denoted as $\Phi = \{\phi_{a,x} \mid a \in A, x \in L_1\}$.*

Theorem 1 ([5]). *The set of $\wedge$-irreducible elements of $\mathcal{M}$, $M_F(A)$, is formed by the pairs $\langle \phi_{a,x}^{\downarrow}, \phi_{a,x}^{\downarrow\uparrow} \rangle$ in $\mathcal{M}$, with $a \in A$ and $x \in L_1$, such that*

$$\phi_{a,x}^{\downarrow} \neq \bigwedge \{\phi_{a_i,x_i}^{\downarrow} \mid \phi_{a_i,x_i} \in \Phi, \phi_{a,x}^{\downarrow} \prec_2 \phi_{a_i,x_i}^{\downarrow}\}$$

and $\phi_{a,x}^{\downarrow} \neq g_{\top_2}$, where $\top_2$ is the maximum element in L_2 and $g_{\top_2} \colon B \to L_2$ is the fuzzy subset defined as $g_{\top_2}(b) = \top_2$, for all $b \in B$.

2.2 Attribute classification

The main results, related to the attribute classification in a multi-adjoint concept lattice framework, were established by meet-irreducible elements of the concept lattice and the notions of consistent set and reduct [6]. For that reason, we will recall the following definitions.

Definition 5. *A set of attributes $Y \subseteq A$ is a* consistent *set of (A, B, R, σ) if the following isomorphism holds:*

$$\mathcal{M}(Y, B, R_Y, \sigma_{Y\times B}) \cong_E \mathcal{M}(A, B, R, \sigma)$$

This is equivalent to say that, for all $\langle g, f\rangle \in \mathcal{M}(A, B, R, \sigma)$, there exists a concept $\langle g', f'\rangle \in \mathcal{M}(Y, B, R_Y, \sigma_{Y\times B})$ such that $g = g'$.

Moreover, if $\mathcal{M}(Y \setminus \{a\}, B, R_{Y\setminus\{a\}}, \sigma_{Y\setminus\{a\}\times B}) \ncong_E \mathcal{M}(A, B, R, \sigma)$, for all $a \in Y$, then Y is called a reduct *of (A, B, R, σ).*

The core *of (A, B, R, σ) is the intersection of all the reducts of (A, B, R, σ).*

A classification of the attributes can be given from the reducts of a context.

Definition 6. *Given a formal context (A, B, R, σ) and the set $\mathcal{Y} = \{Y \subseteq A \mid Y$ is a reduct$\}$ of all reducts of (A, B, R, σ). The set of attributes A can be divided into the following three parts:*

1. *Absolutely necessary attributes (core attribute) $C_f = \bigcap_{Y\in\mathcal{Y}} Y$.*
2. *Relatively necessary attributes $K_f = (\bigcup_{Y\in\mathcal{Y}} Y) \setminus (\bigcap_{Y\in\mathcal{Y}} Y)$.*

3. *Absolutely unnecessary attributes* $I_f = A \setminus (\bigcup_{Y\in\mathcal{Y}} Y)$.

The attribute classification theorems introduced in [6] are based on the previous notions and are recalled below.

Theorem 2 ([6]). *Given* $a_i \in A$, *we have that* $a_i \in C_f$ *if and only if there exists* $x_i \in L_1$, *such that* $\langle \phi^{\downarrow}_{a_i,x_i}, \phi^{\downarrow\uparrow}_{a_i,x_i}\rangle \in M_F(A)$, *satisfying that* $\langle \phi^{\downarrow}_{a_i,x_i}, \phi^{\downarrow\uparrow}_{a_i,x_i}\rangle \neq \langle \phi^{\downarrow}_{a_j,x_j}, \phi^{\downarrow\uparrow}_{a_j,x_j}\rangle$, *for all* $x_j \in L_1$ *and* $a_j \in A$, *with* $a_j \neq a_i$.

Theorem 3 ([6]). *Given* $a_i \in A$, *we have that* $a_i \in K_f$ *if and only if* $a_i \notin C_f$ *and there exists* $\langle \phi^{\downarrow}_{a_i,x_i}, \phi^{\downarrow\uparrow}_{a_i,x_i}\rangle \in M_F(A)$ *satisfying that* E_{a_i,x_i} *is not empty and* $A \setminus E_{a_i,x_i}$ *is a consistent set, where the sets* $E_{a_i,x}$ *with* $a_i \in A$ *and* $x \in L_1$ *are defined as:*

$$E_{a_i,x} = \{a_j \in A \setminus \{a_i\} \mid \textit{there exist } x' \in L_1, \textit{ satisfying } \phi^{\downarrow}_{a_i,x} = \phi^{\downarrow}_{a_j,x'}\}$$

Theorem 4 ([6]). *Given* $a_i \in A$, *it is absolutely unnecessary,* $a_i \in I_f$, *if and only if, for each* $x_i \in L_1$, *we have that* $\langle \phi^{\downarrow}_{a_i,x_i}, \phi^{\downarrow\uparrow}_{a_i,x_i}\rangle \notin M_F(A)$, *or in the case that* $\langle \phi^{\downarrow}_{a_i,x_i}, \phi^{\downarrow\uparrow}_{a_i,x_i}\rangle \in M_F(A)$, *then* $A \setminus E_{a_i,x_i}$ *is not a consistent set.*

The classification of the set of attributes in absolutely necessary, relatively necessary and absolutely unnecessary attributes, provided by the previous theorems, will allow us to obtain reducts (minimal sets of attributes) in the following section. Determining the reducts can entail an important reduction of the computational complexity of the concept lattice.

3 Computing the reducts of a multi-adjoint concept lattice

This section is focused on analyzing the construction process of reducts from the attribute classification shown in the previous section. To begin with, the attributes in the core, that is, the absolutely necessary attributes, are included in all reducts and the unnecessary attributes must be removed.

The choice of the relatively necessary attributes is the main task in the process, because several reducts are obtained when the set of relatively necessary attributes is nonempty.

Hence, several issues raise, such as, how should we select the set of relatively necessary attributes? What is the most efficient way to perform this process? Do all the reducts have the same cardinality? How can we get a reduct with a minimal number of attributes? This work establishes the first steps in order to answer these questions.

Regarding a simplification in the selection of the relatively necessary attributes, a subset of attributes associated with each concept will be considered.

Definition 7. *Given a multi-adjoint frame* $(L_1, L_2, P, \&_1, \dots, \&_n)$ *and a context* (A, B, R, σ) *with the associated concept lattice* $(\mathcal{M}, \preceq)$. *Let* C *be a concept of* $(\mathcal{M}, \preceq)$, *we define* the set of attributes generating C *as the set:*

$$Atg(C) = \{a_i \in A \mid \textit{there exists} \quad \phi_{a_i,x} \in \Phi \quad \textit{such that} \quad \langle \phi^{\downarrow}_{a_i,x}, \phi^{\downarrow\uparrow}_{a_i,x}\rangle = C\}$$

Now, we will present several properties about the attributes of the context which will be useful to build reducts in our context, together with some example which illustrate them.

Proposition 1. *If C is a meet-irreducible concept of $(\mathcal{M}, \leq)$, then $Atg(C)$ is a nonempty set.*

The following example was introduced in [6], in which an attribute classification was given. Now, we will use it in order to clarify the previous result.

Example 1. Let $(L, \preceq, \&_G)$ be a multi-adjoint frame, where $\&_G$ is the Gödel conjunctor with respect to $L = \{0, 0.5, 1\}$. In this framework, the context is (A, B, R, σ), where $A = \{a_1, a_2, a_3, a_4, a_5\}$, $B = \{b_1, b_2, b_3\}$, $R \colon A \times B \to L$ is given by the table in Figure 1, and σ is constant.

R	b_1	b_2	b_3
a_1	1	1	0
a_2	0.5	1	0
a_3	0.5	1	0
a_4	1	1	0.5
a_5	1	1	1

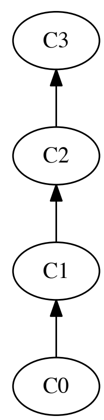


Fig. 1. Relation R and Hasse diagram of Example 1.

The concept lattice of the considered framework and context are displayed in Figure 1, from which it is easy to see that the meet-irreducible elements are C_0, C_1 and C_2. Now, we will show that the sets $\mathrm{Atg}(C_0)$, $\mathrm{Atg}(C_1)$ and $\mathrm{Atg}(C_2)$ are not empty. For that, the fuzzy-attributes associated with the meet-irreducible concepts need to be obtained. Applying the concept-forming operators to the fuzzy-attributes we have

$$\begin{aligned} & \langle \phi^{\downarrow}_{a_4,1.0}, \phi^{\downarrow\uparrow}_{a_4,1.0} \rangle = C_2 \\ \langle \phi^{\downarrow}_{a_1,0.5}, \phi^{\downarrow\uparrow}_{a_1,0.5} \rangle = \langle \phi^{\downarrow}_{a_1,1.0}, \phi^{\downarrow\uparrow}_{a_1,1.0} \rangle = \langle \phi^{\downarrow}_{a_2,0.5}, \phi^{\downarrow\uparrow}_{a_2,0.5} \rangle = {} & \langle \phi^{\downarrow}_{a_3,0.5}, \phi^{\downarrow\uparrow}_{a_3,0.5} \rangle = C_1 \\ \langle \phi^{\downarrow}_{a_2,1.0}, \phi^{\downarrow\uparrow}_{a_2,1.0} \rangle = {} & \langle \phi^{\downarrow}_{a_3,1.0}, \phi^{\downarrow\uparrow}_{a_3,1.0} \rangle = C_0 \end{aligned}$$

obtaining the association which is written in Table 1.

$M_F(A)$	Fuzzy-attributes generating the meet-irreducible concept
C_0	$\phi_{a_2,1}, \phi_{a_3,1}$
C_1	$\phi_{a_1,0.5}, \phi_{a_1,1}, \phi_{a_2,0.5}, \phi_{a_3,0.5}$
C_2	$\phi_{a_4,1}$

Table 1. Fuzzy-attributes generating the meet-irreducible concepts of Example 1.

From this table, the sets of attributes generating these concepts are straightforwardly determined:

$$\begin{aligned}\text{Atg}(C_0) &= \{a_2, a_3\}\\ \text{Atg}(C_1) &= \{a_1, a_2, a_3\}\\ \text{Atg}(C_2) &= \{a_4\}\end{aligned}$$

Hence, these subsets of attributes are nonempty as Proposition 1 shows. □

The following proposition characterizes the singleton sets of attributes generating a concept.

Proposition 2. *If C is a meet-irreducible concept of $(\mathcal{M}, \leq)$ satisfying that card($Atg(C)$) $= 1$, then $Atg(C) \subseteq C_f$.*

Example 2. In the framework of Example 1, if we consider the concept C_2 then we see that the hypothesis given in Proposition 2 are satisfied, that is $\text{card}(\text{Atg}(C_2)) = 1$, and consequently $\text{Atg}(C_2) = \{a_4\} \subseteq C_f$.

This can be checked from the attribute classification given from Table 1 and the classification theorems:

$$\begin{aligned}I_f &= \{a_1, a_5\}\\ K_f &= \{a_2, a_3\}\\ C_f &= \{a_4\}\end{aligned}$$

□

Note that the counterpart of the previous proposition is not true, in general. That is, we can find $a \in C_f$ such that $a \in \text{Atg}(C)$ and satisfying that $\text{card}(\text{Atg}(C)) \geq 1$. What we can assert is that we can always find a meet-irreducible element C satisfying that $\text{card}(\text{Atg}(C)) = 1$, if the core is nonempty, as the following proposition explains.

Proposition 3. *If the attribute $a \in C_f$ then there exists $C \in M_F(A)$ such that $a \in Atg(C)$ and card($Atg(C)$) $= 1$.*

Example 3. Coming back to Example 1, we can ensure that the attribute a_4 belongs to C_f and, as Proposition 3 shows, there exists a concept in $M_F(A)$, which is C_2, verifying that $a_4 \in \text{Atg}(C_2)$ and $\text{card}(\text{Atg}(C_2)) = 1$. □

As a consequence of the above properties, the following corollary holds.

Corollary 1. *If C is a meet-irreducible concept of $(\mathcal{M}, \leq)$ and $Atg(C) \cap K_f \neq \varnothing$ then $card(Atg(C)) \geq 2$.*

Example 4. In Example 1, the concept C_1 is a meet-irreducible element such that $\mathrm{Atg}(C_1) \cap K_f = \{a_1, a_2, a_3\} \cap \{a_2, a_3\} = \{a_2, a_3\} \neq \varnothing$. As a consequence, we have that $\mathrm{card}(\mathrm{Atg}(C_1)) = 3 \geq 2$ as Corollary 1 shows. □

The next proposition guarantees that, if a meet-irreducible concept C is obtained from a relatively necessary attribute, then there does not exist an attribute in the core belonging to $\mathrm{Atg}(C)$.

Proposition 4. *Let C be a meet-irreducible concept. $Atg(C) \cap K_f \neq \varnothing$ if and only if $Atg(C) \cap C_f = \varnothing$.*

Example 5. Considering the meet-irreducible concept C_0 of Example 1, we have that $\mathrm{Atg}(C_0) \cap K_f = \{a_2, a_3\}$. Since this intersection is nonempty, applying Proposition 4, we obtain that $\mathrm{Atg}(C_0) \cap C_f = \{a_2, a_3\} \cap \{a_4\} = \varnothing$. A similar situation is given if we take into account C_1. □

A lower bound and a upper bound of the cardinality of the reducts in a multi-adjoint concept lattice framework are provided.

Proposition 5. *Given $\mathcal{G}_K = \{Atg(C) \mid C \in M_F(A) \quad and \quad Atg(C) \cap K_f \neq \varnothing\}$ and any reduct Y of the context (A, B, R, σ). Then, the following chain is always satisfied:*

$$card(C_f) \leq card(Y) \leq card(C_f) + card(\mathcal{G}_K)$$

Example 6. From Example 1, we can ensure that either attribute a_2 or a_3 is needed (the attribute a_1 is absolutely unnecessary) in order to obtain the meet-irreducible concepts C_0 and C_1. Hence, since $a_4 \in C_f$, two reducts $Y_1 = \{a_2, a_4\}$ and $Y_2 = \{a_3, a_4\}$ exist. Thus, only two attributes are needed in order to consider a concept lattice isomorphic to the original one. Now, we will see that these reducts satisfy the previous proposition.

Since the set $\mathcal{G}_K$ is composed by the attributes generating C_0 and C_1, we have that $\mathcal{G}_K = \{\{a_2, a_3\}, \{a_1, a_2, a_3\}\}$. Therefore, both reducts Y_1 and Y_2 satisfy the inequalities in Proposition 5:

$$1 = \mathrm{card}(C_f) \leq \mathrm{card}(Y_1) = \mathrm{card}(Y_2) \leq \mathrm{card}(C_f) + \mathrm{card}(\mathcal{G}_K) = 3$$

□

The proposition below is fundamental in order to provide a sufficient condition to ensure that all the reducts have the same cardinality.

Proposition 6. *If $\mathcal{G}_K = \{Atg(C) \mid C \in M_F(A) \quad and \quad Atg(C) \cap K_f \neq \varnothing\}$ is a partition of K_f, each attribute in K_f generates only one meet-irreducible element of the concept lattice.*

The following result states several conditions to guarantee that all the reducts have the same cardinality.

Theorem 5. *When the set*

$$\mathcal{G}_K = \{Atg(C) \mid C \in M_F(A) \quad and \quad Atg(C) \cap K_f \neq \varnothing\}$$

is a partition of K_f, then:

(a) All the reducts $Y \subseteq A$ have the same cardinality and, specifically, the cardinality is:

$$card(Y) = card(C_f) + card(\mathcal{G}_K)$$

(b) The number of different reducts obtained from the multi-adjoint context is

$$\prod_{Atg(C)\in\mathcal{G}_K} card(Atg(C))$$

Note that the previous theorem provides a sufficient condition in order to ensure that the cardinality of the reducts is the same, however it is not a necessary condition as Example 6 reveals.

4 Worked out examples

This section begins with an illustrative example of Proposition 6 and Theorem 5 that computes the reducts of a particular multi-adjoint concept lattice framework, and shows that these reducts have the same cardinality.

Example 7. Let $(L_1, L_2, L_3, \preceq, \&_G^*)$ be a multi-adjoint frame, where $L_1 = [0,1]_{10}$, $L_2 = [0,1]_4$ and $L_3 = [0,1]_5$ are regular partitions of $[0,1]$ in 10, 4 and 5 pieces, respectively, and $\&_G^*$ is the discretization of the Gödel conjunctor defined on $L_1 \times L_2$. We consider a context (A, B, R, σ), where $A = \{a_1, a_2, a_3, a_4, a_5, a_6\}$, $B = \{b_1, b_2, b_3\}$, $R\colon A \times B \to L_3$ is given by the table shown in the left side of Figure 2 and σ is constantly $\&_G^*$.

In order to obtain reducts, we will study the meet-irreducible elements of the concept lattice displayed in the right side of Figure 2 and the fuzzy-attributes associated with them. From the corresponding Hasse diagram, we can assert that $M_F(A) = \{C_1, C_8, C_9, C_{10}, C_{13}, C_{14}\}$. The fuzzy-attributes related to these concepts are shown in Table 2.

Applying the attribute classification theorems, we obtain:

$$C_f = \{a_1, a_2\}$$
$$K_f = \{a_3, a_4, a_5, a_6\}$$

Once we have classified the attributes, we are going to construct all possible reducts. Clearly, the attributes a_1 and a_2 must be included in all reducts. Hence, it only remains to choose the relatively necessary attributes that should

R	a_1	a_2	a_3	a_4	a_5	a_6
b_1	0.6	0.2	0.2	0	1	0.6
b_2	0.8	0.4	0.6	0.6	1	0.8
b_3	0.6	0.6	0.2	0	0	0

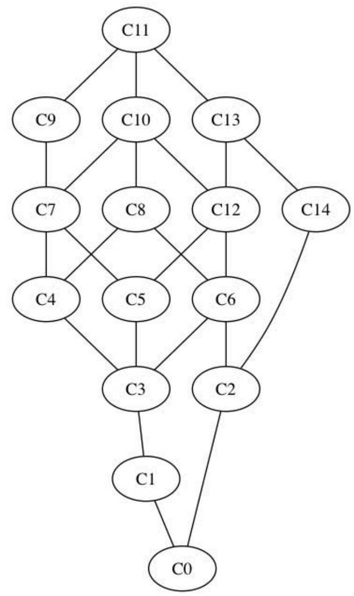


Fig. 2. Relation R (left side) and Hasse diagram of $(\mathcal{M}, \preceq)$ (right side) of Example 7.

be contained in each reduct. For that purpose, we will analyze the attributes generating each meet-irreducible concept:

$$\begin{aligned}
\mathrm{Atg}(C_1) &= \{a_3, a_4\} \\
\mathrm{Atg}(C_8) &= \{a_1\} \\
\mathrm{Atg}(C_9) &= \{a_5, a_6\} \\
\mathrm{Atg}(C_{10}) &= \{a_1\} \\
\mathrm{Atg}(C_{13}) &= \{a_2\} \\
\mathrm{Atg}(C_{14}) &= \{a_2\}
\end{aligned}$$

Since $\mathrm{Atg}(C_1)$ and $\mathrm{Atg}(C_9)$ are disjoint subsets of K_f, we can guarantee that $\mathcal{G}_K$ is a partition of K_f and therefore:

(1) By Proposition 6, each attribute in K_f generates only one meet-irreducible element of the concept lattice. From Table 2, it is easy to prove that the attributes a_3 and a_4 only generate the meet-irreducible concept C_1. The concept C_9 is uniquely generated by a_5 and a_6.

$M_F(A)$	Fuzzy-attributes generating the meet-irreducible concept
C_1	$\phi_{a_3,0.7}, \phi_{a_3,0.8}, \phi_{a_3,0.9}, \phi_{a_3,1}$ $\phi_{a_4,0.7}, \phi_{a_4,0.8}, \phi_{a_4,0.9}, \phi_{a_4,1}$
C_8	$\phi_{a_1,0.9}, \phi_{a_1,1}$
C_9	$\phi_{a_5,0.1}, \phi_{a_5,0.2}, \phi_{a_5,0.3}, \phi_{a_5,0.4}, \phi_{a_5,0.5}, \phi_{a_5,0.6}, \phi_{a_5,0.7}, \phi_{a_5,0.8}, \phi_{a_5,0.9}, \phi_{a_5,1}$ $\phi_{a_6,0.1}, \phi_{a_6,0.2}, \phi_{a_6,0.3}, \phi_{a_6,0.4}, \phi_{a_6,0.5}, \phi_{a_6,0.6}$
C_{10}	$\phi_{a_1,0.7}, \phi_{a_1,0.8}$
C_{13}	$\phi_{a_2,0.3}, \phi_{a_2,0.4}$
C_{14}	$\phi_{a_2,0.5}, \phi_{a_2,0.6}$

Table 2. Fuzzy-attributes generating the meet-irreducible concepts of Example 7.

(2) By Theorem 5, all the reducts have the same cardinality. Thus, since

$$\begin{aligned}\mathcal{G}_K &= \{\text{Atg}(C) \mid C \in M_F(A) \quad \text{and} \quad \text{Atg}(C) \cap K_f \neq \varnothing\} \\ &= \{\text{Atg}(C_1), \text{Atg}(C_9)\} \\ &= \{\{a_3, a_4\}, \{a_5, a_6\}\}\end{aligned}$$

we have that $\text{card}(Y) = \text{card}(C_f) + \text{card}(\mathcal{G}_K) = 2 + 2 = 4$, for any reduct Y of the context. Moreover, the number of reducts that we obtain from this context is

$$\prod_{\text{Atg}(C) \in \mathcal{G}_K} \text{card}(\text{Atg}(C)) = 2 \cdot 2 = 4$$

Specifically, the whole set of reducts are listed below:

$$\begin{aligned}Y_1 &= \{a_1, a_2, a_3, a_5\} \\ Y_2 &= \{a_1, a_2, a_3, a_6\} \\ Y_3 &= \{a_1, a_2, a_4, a_5\} \\ Y_4 &= \{a_1, a_2, a_4, a_6\}\end{aligned}$$

From the previous reducts, we obtain the following isomorphic concept lattices:

$$(\mathcal{M}, \preceq) \cong (\mathcal{M}^{Y_1}, \preceq) \cong (\mathcal{M}^{Y_2}, \preceq) \cong (\mathcal{M}^{Y_3}, \preceq) \cong (\mathcal{M}^{Y_4}, \preceq)$$

□

Now, we will present a situation where the elements belonging to the set $\mathcal{G}_K$ are not a partition of K_f, and we will see that in this particular example several reducts with different cardinality are obtained.

Example 8. Considering the same framework that in the previous example, we fix a context (A, B, R, σ) where the set A consists of seven attributes, the set B contains three objects and R is obtained from the relation of the previous example with a few of changes shown in Table 3. Hence, we obtain an isomorphic concept lattice to the one shown in Figure 2, but a different attribute classification arises.

Table 3. Definition of R

R	a_1	a_2	a_3	a_4	a_5	a_6	a_7
b_1	0.6	0.2	0.2	1	0.6	0.2	0
b_2	0.8	0.4	0.4	1	0.8	0.6	0.6
b_3	0.6	0.6	0.2	0	0	0.2	0

The attributes are classified as follows:

$$C_f = \{a_1, a_2\}$$
$$K_f = \{a_4, a_5, a_6, a_7\}$$
$$I_f = \{a_3\}$$

As a consequence, a_1 and a_2 must belong to all the reducts and a_3 should be removed. Analyzing the meet-irreducible elements and the fuzzy-attributes generating them, we obtain:

$$\mathrm{Atg}(C_1) = \{a_6, a_7\}$$
$$\mathrm{Atg}(C_8) = \{a_1\}$$
$$\mathrm{Atg}(C_9) = \{a_4, a_5, a_6\}$$
$$\mathrm{Atg}(C_{10}) = \{a_1\}$$
$$\mathrm{Atg}(C_{13}) = \{a_2\}$$
$$\mathrm{Atg}(C_{14}) = \{a_2\}$$

Now, we have to select one attribute of $\mathrm{Atg}(C_1)$ and another one of $\mathrm{Atg}(C_9)$ in order to obtain the whole set of meet-irreducible concepts and compute the reducts. However, in this case, $\mathrm{Atg}(C_1) \subseteq K_f$ and $\mathrm{Atg}(C_9) \subseteq K_f$ and the intersection $\mathrm{Atg}(C_1) \cap \mathrm{Atg}(C_9) = a_6$ is nonempty. Therefore, the set $\mathcal{G}_K = \{\mathrm{Atg}(C_1), \mathrm{Atg}(C_9)\}$ is not a partition of K_f.

Consequently, we can obtain the following different reducts whose sizes depend on the chosen attributes as we can see below:

$$Y_1 = \{a_1, a_2, a_6\}$$
$$Y_2 = \{a_1, a_2, a_4, a_7\}$$
$$Y_3 = \{a_1, a_2, a_5, a_7\}$$

□

This example provides the idea that, in order to compute a minimal reduct, with respect to the number of attributes, the relatively necessary attributes to be taken into account must be the ones given in the intersection of the sets $\mathrm{Atg}(C)$, with $\mathrm{Atg}(C) \in \mathcal{G}_K$.

5 Conclusion and future work

Based on the attribute classification introduced in [6], a construction process of the reducts of a multi-adjoint concept lattice has been shown. Several properties have been stated together with examples that illustrate the shown results. The importance of the choice of the relatively necessary attributes for computing the reducts has also been highlighted.

More properties related to reducts will be investigated in the future in order to find the most profitable way to generate them. We are also interested in obtaining an algorithm that provides a reduct with a minimal number of attributes for any multi-adjoint concept lattice framework given.

References

1. L. Antoni, S. Krajči, and O. Krídlo. Constraint heterogeneous concept lattices and concept lattices with heterogeneous hedges. *Fuzzy Sets and Systems*, 2015, In Press.
2. R. Bělohlávek. Lattices of fixed points of fuzzy Galois connections. *Mathematical Logic Quartely*, 47(1):111–116, 2001.
3. A. Burusco and R. Fuentes-González. Construction of the L-fuzzy concept lattice. *Fuzzy Sets and Systems*, 97(1):109–114, 1998.
4. J. Chen, J. Li, Y. Lin, G. Lin, and Z. Ma. Relations of reduction between covering generalized rough sets and concept lattices. *Information Sciences*, 304:16 – 27, 2015.
5. M. E. Cornejo, J. Medina, and E. Ramírez-Poussa. On the classification of fuzzy-attributes in multi-adjoint concept lattices. *Lecture Notes in Computer Science*, 7903:266–277, 2013.
6. M. E. Cornejo, J. Medina, and E. Ramírez-Poussa. Attribute reduction in multi-adjoint concept lattices. *Information Sciences*, 294:41 – 56, 2015.
7. M. E. Cornejo, J. Medina, and E. Ramírez-Poussa. Multi-adjoint algebras versus extended-order algebras. *Applied Mathematics & Information Sciences*, 9(2L):365–372, 2015.
8. M. E. Cornejo, J. Medina, and E. Ramírez-Poussa. Multi-adjoint algebras versus non-commutative residuated structures. *International Journal of Approximate Reasoning*, 66:119–138, 2015.
9. S. Krajči. A generalized concept lattice. *Logic Journal of IGPL*, 13(5):543–550, 2005.
10. J. Medina. Relating attribute reduction in formal, object-oriented and property-oriented concept lattices. *Computers & Mathematics with Applications*, 64(6):1992–2002, 2012.
11. J. Medina and M. Ojeda-Aciego. Multi-adjoint t-concept lattices. *Information Sciences*, 180(5):712–725, 2010.
12. J. Medina, M. Ojeda-Aciego, and J. Ruiz-Calviño. Formal concept analysis via multi-adjoint concept lattices. *Fuzzy Sets and Systems*, 160(2):130–144, 2009.
13. Z. Pawlak. Rough sets. *International Journal of Computer and Information Science*, 11:341–356, 1982.
14. S. Pollandt. *Fuzzy Begriffe*. Springer, Berlin, 1997.
15. L. Wei and J.-J. Qi. Relation between concept lattice reduction and rough set reduction. *Knowledge-Based Systems*, 23(8):934–938, 2010.

Conceptual Methods for Identifying Needs of Mobile Network Subscribers

Dmitry Palchunov[1,2], Gulnara Yakhyaeva[2], Ekaterina Dolgusheva[2]

[1] Sobolev Institute of Mathematics, Novosibirsk, Russian Federation
palch@math.nsc.ru
[2] Novosibirsk State University, Novosibirsk, Russian Federation
gul_nara@mail.ru, cralya_cher9299@mail.ru

Abstract. The paper is devoted to methods for identifying payment plans and services by mobile operators which are the best for the given subscribers. We base our research on the model-theoretic approach to domain formalization. We use Formal Concept Analysis for processing the mobile subscriber data. An Ontological Model of the domain "Mobile Networks" is constructed in the scope of this research. The Ontological Model of the domain is constructed by integration of data extracted from depersonalized subscriber profiles. The signature of this Ontological Model contains unary predicates which describe subscriber behavior and features of payment plans and services. We consider formal contexts where objects are subscriber models and attributes are formulas of predicate logic. We investigate concept lattices and association rules of these formal contexts. Knowledge about optimal payment plans and services for a given subscriber is generated automatically with the help of the association rules.

Keywords: mobile networks, mobile network subscribers, formal context, concept lattice of formal context, association rules, ontology, ontological model.

1 Introduction

Mobile connection is a very important part of our life. Mobile operators provide the possibility to be in touch for people in different countries. Operators provide access to USSD-applications and to the Internet.

Mobile operators develop various payment plans and services to satisfy their clients' needs. However it is difficult for mobile network subscribers to get up-to-date information about new payment plans or services. Mobile operators send SMS messages to inform clients about news. But it is very expensive to inform all subscribers about every small change or update of services. A possible solution of this problem is sending personal recommendations about services and payment plans that could be useful for a given subscriber.

A visualization approach based on a graph of calls made by subscribers was used in [1] for mining behavior patterns of mobile network subscribers. A behavior pattern discovered during the graph exploration resulted in developing and applying a new payment plan. Development of methods for increasing the number of subscribers

© Marianne Huchard, Sergei O. Kuznetsov (Eds.): CLA 2016, pp. 147–158, ISBN 978-5-600-01454-1, National Research University Higher School of Economics, 2016.

using services by a mobile network is studied in [2]. An algorithm called Frequent Pattern-Growth Strategy is used for mining patterns in how subscribers use mobile network services. Optimization strategies are suggested by experts based on series of 'frequent' sets.

Formal Concept Analysis is a well-known formalism in data analysis and knowledge engineering, see recent surveys [3, 4]. Formal Concept Analysis is used to develop user behavior templates [5, 6]. These results are applied to planning and running marketing campaigns.

Association rules for optimizing structures of menus for accessing mobile network services were constructed in [7]. The Apriori algorithm was used in [8] to develop association rules patterns in services visited during a single subscriber session. Today we have more effective algorithms for mining association rules, e.g. see [9].

Fuzzy concept lattices were first introduced in [10]. Papers [11-13] are devoted to definitions of fuzzy transaction, support and confidence of fuzzy association rules. The authors of [11] used an algorithm developed in [14] for building sets of fuzzy rules which describe dependencies between popular telecom services provided by mobile networks in Taiwan.

Our research is devoted to methods for identifying payment plans and services which would be optimal for a given mobile network subscriber. Such knowledge allows mobile operator to make really useful recommendations for subscribers.

We base our research on the model-theoretic approach to domain formalization [15-18]. We use methods and techniques of Formal Concept Analysis for processing the mobile subscriber data. Now a lot of attention is paid to the relationships between FCA and models of knowledge representation and processing [19].

The ontological model of the domain "Mobile Networks" is constructed by integration of data extracted from depersonalized subscriber profiles. The signature of this ontological model contains unary predicates which describe subscriber behavior and features of payment plans and services. To generate meaningful recommendation of alternative services and payment plans, we define formal contexts where objects are subscriber models, and attributes are formulas of predicate logic. We investigate concept lattices and association rules of these formal contexts to get high-quality recommendation. To do this, we consider extensions of attribute sets of formal contexts.

In [20] extensions of infinite attribute sets were considered, it was suggested to use concept descriptions of bounded depth. In [21] a new approach to reduce the number of attributes was presented.

In this paper we consider finite extensions of the initial finite context. We use interrelation between axiomatizable classes and FCA [22]. Section 2.1 is devoted to isomorphisms between lattices of relatively axiomatizable classes of one-element models and lattices of formal concepts of formal contexts generated by these classes. Section 2.2 describes extensions of such formal contexts having distributive concept lattices.

The main purpose of this paper is to develop methods of identifying payment plans and services which would be optimal for the given mobile network subscriber. To do this, firstly, we construct Case Model based on the known information about behavior patterns of mobile network subscribers (Section 2.2). We represent the Case Model as a relatively axiomatizable class of one-element models. On the base of this Case Model we define a formal context.

Secondly, we move from the Case Model to Ontology Model (Section 3.1). We construct the set of ontological projections which is the basis of extensions of attribute set of the formal context under consideration (Section 3.2).

And finally we mine association rules with high confidence and support in the extended formal context. Computer experiments show that the methods presented in the paper allow us to find association rules which can be used for recommendations.

2 Case Model

2.1 Relatively axiomatizable classes and formal contexts

Here we introduce some definitions and results on the relationship between relatively axiomatizable classes and formal contexts. The main result of this section is Proposition 2 which is necessary for proofs of Propositions 4 and 5 in Section 2.2. The proofs of the statements are based on [22].

An algebraic system (a model) is a tuple $\mathfrak{A} = \langle A;\ P_1, \dots, P_n, f_1, \dots, f_m, c_1, \dots, c_k \rangle$, where the set $|\mathfrak{A}| = A$ is called universe, $P_1, \dots, P_n$ are predicates defined on the set A, $f_1, \dots, f_m$ are functions defined on the set A and $c_1, \dots, c_k$ are constants. The tuple $\sigma = \langle P_1, \dots, P_n, f_1, \dots, f_m, c_1, \dots, c_k \rangle$ is called signature of the algebraic system $\mathfrak{A}$.

Denote by $FV(\varphi)$ the set of all free variables of a formula φ. A formula having no free variables is called sentence. For a signature σ we denote:

$F(\sigma) \leftrightharpoons \{\varphi \mid \varphi\ is\ a\ formula\ of\ the\ signature\ \sigma\}$,

$F_1(\sigma) \leftrightharpoons \{\varphi \mid \varphi \in F(\sigma)\ and\ FV(\varphi) = \{x\}\}$,

$S(\sigma) \leftrightharpoons \{\varphi \mid \varphi\ is\ a\ sentence\ of\ the\ signature\ \sigma\}$ and

$K(\sigma) \leftrightharpoons \{\mathfrak{A} \mid \mathfrak{A}\ is\ a\ model\ of\ the\ signature\ \sigma\}$.

Here $FV(\varphi) = \{x\}$ means that each formula $\varphi \in F_1(\sigma)$ has just one free variable, which is the fixed variable x.

Consider a signature σ and a model $\mathfrak{A} \in K(\sigma)$. For a sentence $\psi \in S(\sigma)$ we denote $\mathfrak{A} \vDash \psi$ if ψ is true in the model $\mathfrak{A}$. For a formula $\varphi(x_1, \dots, x_n) \in F(\sigma)$ we write $\mathfrak{A} \vDash \varphi$ if $\mathfrak{A} \vDash \forall x_1 \dots \forall x_n \varphi(x_1, \dots, x_n)$.

Definition 1. *Let $K \subseteq K(\sigma)$. For a formula $\varphi \in F(\sigma)$ we denote $K \vDash \varphi$ if $\mathfrak{A} \vDash \varphi$ for any $\mathfrak{A} \in K$. For a set of formulas $\Gamma \subseteq F(\sigma)$ we denote $K \vDash \Gamma$ if $\mathfrak{A} \vDash \varphi$ for any $\mathfrak{A} \in K$ and $\varphi \in \Gamma$. For a set of formulas $\Gamma \subseteq F(\sigma)$ we denote*

$$K(\Gamma) \leftrightharpoons K_\sigma(\Gamma) \leftrightharpoons \{\mathfrak{A} \in K(\sigma) | \mathfrak{A} \vDash \varphi \quad for\ \ any\ \varphi \in \Gamma\}.$$

A class $K \subseteq K(\sigma)$ is called axiomatizable if there exists a set $\Gamma \subseteq S(\sigma)$ such that $K = \{\mathfrak{A} \in K(\sigma) \mid \mathfrak{A} \vDash \Gamma\}$.

For the aims of our research we need to generalize the notion of relatively axiomatizable class [22] to the case of arbitrary sets of formulas Δ.

Definition 2. *Let K, $K_1 \subseteq K(\sigma)$ and $\Delta \subseteq F(\sigma)$. We say that the class K_1 is axiomatizable in the class K relatively to the set of formulas Δ if there exists a set $\Gamma \subseteq \Delta$ such that $K_1 = \{\mathfrak{A} \in K \mid \mathfrak{A} \vDash \Gamma\}$.*

Notice that the class $K_1 \subseteq K(\sigma)$ is axiomatizable if and only if K_1 is axiomatizable in the class $K = K(\sigma)$ relatively to the set of formulas $\Delta = S(\sigma)$.

Definition 3. *For $K \subseteq K(\sigma)$ and $\Delta \subseteq F(\sigma)$ we denote* $\mathbb{B}(K,\Delta) \leftrightharpoons \{K_1 \mid K_1 \textit{ is axiomatizable in } K \textit{ relatively to the set of formulas } \Delta\}$ *and* $T_\Delta(K) \leftrightharpoons \{\varphi \in \Delta \mid K \vDash \varphi\}$. *The set of formulas $T_\Delta(K)$ is call **Δ-type** of K.*

Note that $K_1 \in \mathbb{B}(K,\Delta)$ if and only if $K_1 = \{\mathfrak{A} \in K \mid \mathfrak{A} \vDash T_\Delta(K_1)\}$.

For each class $K \subseteq K(\sigma)$ and set $\Delta \subseteq S(\sigma)$ we consider the formal context $(K,\Delta,\vDash)$, with derivation operator $()'$ [23].

Remark 1. *Let $K \subseteq K(\sigma)$, $\Delta \subseteq F(\sigma)$ and $A \subseteq K$. Then $A' = T_\Delta(K)$.*

For a formal context $(\mathrm{G},\mathrm{M},\mathrm{I})$ by $\underline{\mathfrak{B}}(\mathrm{G},\mathrm{M},\mathrm{I})$ we denote the lattice of formal concepts of the formal context $(\mathrm{G},\mathrm{M},\mathrm{I})$.

Proposition 1. *Let $K \subseteq K(\sigma)$, $\Delta \subseteq F(\sigma)$, $A \subseteq K$ and $B \subseteq \Delta$. Then $(A,B) \in \underline{\mathfrak{B}}(K,\Delta,\vDash)$ if and only if A is axiomatizable in the class K relatively to the set of formulas Δ and $B = T_\Delta(A)$.*

Proof. ($\Rightarrow$) Let $(A,B) \in \underline{\mathfrak{B}}(K,\Delta,\vDash)$. Then $B = A'$, so $B = T_\Delta(A)$ by Remark 1. We have $A = B'$, hence $A = \{\mathfrak{A} \in K \mid \mathfrak{A} \vDash B\}$ and $B \subseteq \Delta$. Therefore, by Definition 2, the class A is axiomatizable in the class K relatively to the set of formulas Δ.

($\Leftarrow$) Let the class A be axiomatizable in the class K relatively to the set of formulas Δ and $B \subseteq \Delta$. So there exists $\Gamma \subseteq \Delta$ such that $A = \{\mathfrak{A} \in K \mid \mathfrak{A} \vDash \Gamma\}$. Then in the formal context $(K,\Delta,\vDash)$ we have $\Gamma' = A$. So $A'' = A$. The set $B = T_\Delta(A)$, thus $B = A'$ by Remark 1. Therefore, $(A,B) \in \underline{\mathfrak{B}}(K,\Delta,\vDash)$.

Corollary 1. *Let $K, K_1 \subseteq K(\sigma)$ and $\Delta \subseteq F(\sigma)$.*

1. $K_1 \in \mathbb{B}(K,\Delta)$ if and only if $(K_1, T_\Delta(K_1)) \in \underline{\mathfrak{B}}(K,\Delta,\vDash)$.

2. $K_1 = K_1''$ if and only if K_1 is axiomatizable in the class K relatively to the set of formulas Δ.

Therefore, the classes which are axiomatizable in a class K relatively to a set of formulas Δ are exactly extents of the formal concepts of the formal context $(K,\Delta,\vDash)$.

We consider $\mathbb{B}(K,\Delta)$ as a set ordered by inclusion $\subseteq$. So $\mathbb{B}(K,\Delta)$ is a lattice.

Proposition 2. *The lattices $\underline{\mathfrak{B}}(K,\Delta,\vDash)$ and $\mathbb{B}(K,\Delta)$ are isomorphic, i.e., $\underline{\mathfrak{B}}(K,\Delta,\vDash) \cong \mathbb{B}(K,\Delta)$, for any $K \subseteq K(\sigma)$ and $\Delta \subseteq F(\sigma)$.*

Proof. Let us consider the mapping $h: \underline{\mathfrak{B}}(K,\Delta,\vDash) \to \mathbb{B}(K,\Delta)$ defined as follows: $h(A,B) = A$ for any $(A,B) \in \underline{\mathfrak{B}}(K,\Delta,\vDash)$. By Proposition 1 for any $(A,B) \in \underline{\mathfrak{B}}(K,\Delta,\vDash)$ we have $h(A,B) = A \in \mathbb{B}(K,\Delta)$. For each $A \in \mathbb{B}(K,\Delta)$ it is true that $(A, T_\Delta(A)) \in \underline{\mathfrak{B}}(K,\Delta,\vDash)$, so $h(A, T_\Delta(A)) = A$. Thus the mapping h is onto.

For any (A_1,B_1), $(A_2,B_2) \in \underline{\mathfrak{B}}(K,\Delta,\vDash)$ we have: $(A_1,B_1) \leq (A_2,B_2)$ iff $A_1 \subseteq A_2$. Hence the mapping h preserves the partial order.

Therefore, the mapping h is an isomorphism.

2.2 Description of the Case Model

Further we consider signatures consisting of a finite set of unary predicate symbols, i.e. $\sigma =< P_1, \ldots, P_n >$. We consider the set $\Delta \subseteq S(\sigma)$ for different signatures σ which means that the original signature is enriched by new unary predicate symbols. From a model-theoretic point of view we may assume that there is some covering signature σ^U and all considered signatures are its subsets.

Consider a finite set $A = \{e_1, \ldots, e_n\}$ of subscribers of a given mobile network and fix a signature $\sigma = \sigma_{\mathbb{P}} \cup \sigma_{\mathbb{Q}}$ where $\sigma_{\mathbb{P}}$ is a set of personal characteristics of subscribers and $\sigma_{\mathbb{Q}}$ is a set of payments plans, services and options. Each of these sets has a hierarchical structure. There are more details about the signatures $\sigma_{\mathbb{P}}$ and $\sigma_{\mathbb{Q}}$ below. For each subscriber e_i we know which characteristics (presented by signature predicates from σ) are true and which characteristics are false. Thus, for each subscriber e_i there is a one-element model $\boldsymbol{e_i} = \langle \{e_i\}, \sigma \rangle$ which is called a case of the domain $\mathbb{M}$.

Consider the ***Case Model*** $\mathfrak{A} = \langle A, \sigma \rangle$ defined by a set of cases $\{\boldsymbol{e_1}, \ldots, \boldsymbol{e_n}\}$ [20]. On the model $\mathfrak{A}$ for each signature predicate $P \in \sigma$ and for every element $e \in A$ we have $\mathfrak{A} \vDash P(e)$ if and only if the predicate $P(x)$ is true in the model (case) $\boldsymbol{e}$ (i.e., $\boldsymbol{e} \vDash P(x)$). Here $\boldsymbol{e} \vDash P(x)$ means that $\vDash \exists x P(x)$. On the base of the Case Model $\mathfrak{A} = \langle A, \sigma \rangle$ in the section 3.5 we will define the ontological model.

Denote by $K_{\mathfrak{A}} = \{\boldsymbol{e_1}, \ldots, \boldsymbol{e_n}\}$ the class of cases (one-element models) generated by the set of subscribers $\{e_1, \ldots, e_n\}$.

Note that $K_{\mathfrak{A}} = \{\langle \{e\};\ \sigma \rangle \mid e \in A \ and\ \langle \{e\};\ \sigma \rangle \leq \mathfrak{A}\}$. Here the notation $\boldsymbol{e} = \langle \{e\};\ \sigma \rangle \leq \mathfrak{A}$ means that the model $\boldsymbol{e}$ is a submodel of the model $\mathfrak{A}$. Recall that in pure predicate signature each subset of a model is the universe of its submodel.

Here we consider different sets of formulas $\Delta \subseteq F_1(\sigma)$. In particular, we consider $\Delta_\sigma = \{P(x) \mid P \in \sigma\} \subseteq F_1(\sigma)$. Denote by $C_{\mathfrak{A}}^{\Delta} = (K_{\mathfrak{A}}, \Delta, \vDash)$ the formal context having the set of objects $K_{\mathfrak{A}}$, the set of attributes Δ and the incidence relation $\vDash$. Denote $C_{\mathfrak{A}}^{\sigma} = (K_{\mathfrak{A}}, \Delta_\sigma, \vDash)$.

$\Delta \subseteq F_1(\sigma)$ is a set of properties of the cases $\boldsymbol{e} \in K_{\mathfrak{A}}$, which are definable by formulas of the signature σ. When we change the set Δ we change the set of attributes of the formal context keeping fixed the set of objects $K_{\mathfrak{A}}$. Reductions and expansions of formal contexts were studied in [25].

Let us consider two formal contexts $C_1 = (G, M_1, I)$ and $C_2 = (G, M_2, I)$. Suppose that $M_1 \subseteq M_2$, $A \subseteq G$ and $A = A''$ в C_1. Then $A = A''$ in C_2.

We define a mapping $i: \underline{\mathfrak{B}}(G, M_1, I) \to \underline{\mathfrak{B}}(G, M_2, I)$ as follows: $i(A, B_1) = (A, B_2)$, where $A \subseteq G$, $B_1 \subseteq M_1$, $B_2 \subseteq M_2$, $A = A''$, $B_1 = A'$ in the context C_1 and $B_2 = A'$ in the context C_2.

Remark 2. *The mapping* $i: \underline{\mathfrak{B}}(G, M_1, I) \to \underline{\mathfrak{B}}(G, M_2, I)$ *is an isomorphic embedding of the lattice* $\underline{\mathfrak{B}}(G, M_1, I)$ *into the lattice* $\underline{\mathfrak{B}}(G, M_2, I)$.

Next consider an arbitrary signature σ_0 and an arbitrary class $K_0 \subseteq K(\sigma_0)$.

Remark 3. *Let* $\Delta \subseteq F(\sigma_0)$ *and* $\varphi_1, \ldots, \varphi_n \in F(\sigma_0)$. *Then the mapping* $i: \underline{\mathfrak{B}}(K_0, \Delta, \vDash) \to \underline{\mathfrak{B}}(K_0, \Delta \cup \{\varphi_1 \& \ldots \& \varphi_n\}, \vDash)$ *is an isomorphism of lattices.*

Corollary 2. *a) The sets of association rules of the formal contexts* $(K_0, \Delta, \vDash)$ *and* $(K_0, \Delta \cup \{\varphi_1 \& \dots \& \varphi_n\}, \vDash)$ *coincide up to the substitution of the formula* $(\varphi_1 \& \dots \& \varphi_n)$ *by the set* $\{\varphi_1, \dots, \varphi_n\}$.

b) The sets of attribute implications of the formal contexts $(K_0, \Delta, \vDash)$ *u* $(K_0, \Delta \cup \{\varphi_1 \& \dots \& \varphi_n\}, \vDash)$ *coincide up to the substitution of the formula* $(\varphi_1 \& \dots \& \varphi_n)$ *by the set* $\{\varphi_1, \dots, \varphi_n\}$.

Corollary 3. *If* $\Delta, \Delta_1 \subseteq F(\sigma_0)$, $\Delta \subseteq \Delta_1$ *and the set* $\Delta_1 \backslash \Delta$ *consists of some conjunctions of formulas from* Δ *then the sets of attribute implications as well as the sets of association rules of the formal contexts* $(K_0, \Delta, \vDash)$ *and* $(K_0, \Delta_1, \vDash)$ *coincide up to the substitution of the conjunctions from* $\Delta_1 \backslash \Delta$ *by the corresponding sets of formulas.*

Let us go back to the formal context $(K_{\mathfrak{A}}, \Delta_\sigma, \vDash)$.

Remark 4. *Let* $P_1, P_2 \in \sigma$ *. Then the mapping* $i: \underline{\mathfrak{B}}(K_{\mathfrak{A}}, \Delta_\sigma, \vDash) \to \underline{\mathfrak{B}}(K_{\mathfrak{A}}, \Delta_\sigma \cup \{P_1(x) \vee P_2(x)\}, \vDash)$ *is an isomorphic embedding of lattices; in the general case this mapping is not an isomorphism. Moreover, in the general case* $\underline{\mathfrak{B}}(K_{\mathfrak{A}}, \Delta_\sigma, \vDash) \ncong \underline{\mathfrak{B}}(K_{\mathfrak{A}}, \Delta_\sigma \cup \{P_1(x) \vee P_2(x)\}, \vDash)$.

Corollary 4. *In the general case if we add a disjunction* $(P_1(x) \vee \dots \vee P_k(x))$ *to the set of formulas* Δ_σ*, where* $P_1, \dots, P_k \in \sigma$*, then the set of association rules of the formal context* $(K_{\mathfrak{A}}, \Delta_\sigma, \vDash)$ *will be changed.*

Denote $\Delta_\sigma^\vee = \Delta_\sigma \cup \{(P_1(x) \vee \dots \vee P_k(x)), | P_i \in \sigma\}$.

We will be adding disjunctions of signature predicates into the set Δ_σ for improving association rules based on an algorithm for subscribers' behavior prediction. It means that we will consider the set of formulas $\Delta_\sigma^\vee$ instead of the set of formulas Δ_σ and the formal context $(K_{\mathfrak{A}}, \Delta_\sigma^\vee, \vDash)$ instead of the formal context $(K_{\mathfrak{A}}, \Delta_\sigma, \vDash)$.

Definition 4. We say that a set of formulas $\Delta \subseteq F(\sigma_0)$ is closed under disjunction if $(\varphi \vee \psi) \in \Delta$ for any $\varphi, \psi \in \Delta$.

Proposition 3. *Let* $K \subseteq K_{\mathfrak{A}}$ *and* $\Delta \subseteq F(\sigma)$*. If the set of formulas* Δ *is closed under disjunction then the lattice* $\mathbb{B}(K, \Delta)$ *is distributive.*

Proof. Assume that $\Delta \subseteq F(\sigma)$ and $K_1, K_2 \in \mathbb{B}(K, \Delta)$. Then $K_1, K_2 \subseteq K$ and there exist $\Gamma_1, \Gamma_2 \in \Delta$ such that $K_1 = \{\mathfrak{A} \in K \mid \mathfrak{A} \vDash \Gamma_1\}$ and $K_2 = \{\mathfrak{A} \in K \mid \mathfrak{A} \vDash \Gamma_2\}$.

Denote $\Gamma_3 = \Gamma_1 \cup \Gamma_2$ and $\Gamma_4 = \{(\varphi \vee \psi) \mid \varphi \in \Gamma_1 \; and \; \psi \in \Gamma_2)\}$. Then $K_1 \cap K_2 = \{\mathfrak{A} \in K \mid \mathfrak{A} \vDash \Gamma_3\}$, hence $(K_1 \cap K_2) \in \mathbb{B}(K, \Delta)$.

Let $\mathfrak{A} \in K$. Then $\mathfrak{A}$ is a one-element model. Therefore for any $\varphi \in \Gamma_1$ and $\psi \in \Gamma_2$ we have: $\mathfrak{A} \vDash (\varphi \vee \psi) \Leftrightarrow \mathfrak{A} \vDash \forall x_1 \dots \forall x_n (\varphi(x_1, \dots, x_n) \vee \psi(x_1, \dots, x_n)) \Leftrightarrow$

$$\Leftrightarrow \mathfrak{A} \vDash \forall x_1 \dots \forall x_n \varphi(x_1, \dots, x_n) \; or \; \mathfrak{A} \vDash \forall x_1 \dots \forall x_n \psi(x_1, \dots, x_n) \Leftrightarrow$$
$$\Leftrightarrow \mathfrak{A} \vDash \varphi \; or \; \mathfrak{A} \vDash \psi, \text{ where } FV(\varphi \vee \psi) = \{x_1, \dots, x_n\}.$$

Assume that $\mathfrak{A} \in (K_1 \cup K_2)$, then $\mathfrak{A} \in K_1 \; or \; \mathfrak{A} \in K_2 \Rightarrow$

$$\Rightarrow (\mathfrak{A} \in K \; and \; \mathfrak{A} \vDash \Gamma_1) \; or \; (\mathfrak{A} \in K \; and \; \mathfrak{A} \vDash \Gamma_2) \Rightarrow$$
$$\Rightarrow \mathfrak{A} \in K \; and \; (\mathfrak{A} \vDash \Gamma_1 \; or \; \mathfrak{A} \vDash \Gamma_2) \Rightarrow \mathfrak{A} \in K \; and \; \mathfrak{A} \vDash \Gamma_4.$$

Next, suppose that $\mathfrak{A} \nvDash \Gamma_1$ and $\mathfrak{A} \nvDash \Gamma_2$. So there exist $\varphi \in \Gamma_1$ and $\psi \in \Gamma_2$ such that $\mathfrak{A} \nvDash \varphi$ and $\mathfrak{A} \nvDash \psi$. Then $\mathfrak{A} \nvDash (\varphi \vee \psi)$, so $\mathfrak{A} \nvDash \Gamma_4$.

Thus, if $\mathfrak{A} \vDash \Gamma_4$ then $(\mathfrak{A} \vDash \Gamma_1 \; or \; \mathfrak{A} \vDash \Gamma_2)$. Hence, if $\mathfrak{A} \in K$ and $\mathfrak{A} \vDash \Gamma_4$ then $\mathfrak{A} \in K_1$ or $\mathfrak{A} \in K_2$, so $\mathfrak{A} \in (K_1 \cup K_2)$.

Therefore, $K_1 \cup K_2 = \{\mathfrak{A} \in K \mid \mathfrak{A} \vDash \Gamma_4\}$ and $(K_1 \cup K_2) \in \mathbb{B}(K, \Delta)$.

We proved that $(K_1 \cap K_2), (K_1 \cup K_2) \in \mathbb{B}(K, \Delta)$ for any $K_1, K_2 \in \mathbb{B}(K, \Delta)$. Hence, the lattice $\mathbb{B}(K, \Delta)$ is distributive.

Proposition 4. *The lattice of formal concepts* $\underline{\mathfrak{B}}(K_{\mathfrak{A}}, \Delta_{\sigma}^{\vee}, \vDash)$ *is distributive.*

Proof: in virtue of Proposition 2 and Proposition 3.

However the initial formal context $(K_{\mathfrak{A}}, \Delta_{\sigma}, \vDash)$ does not have this good property.

Remark 5. *In the general case the lattice of formal concepts* $\underline{\mathfrak{B}}(K_{\mathfrak{A}}, \Delta_{\sigma}, \vDash)$ *is not distributive. It means that there exists a class* $K_{\mathfrak{A}}$ *such that the lattice* $\underline{\mathfrak{B}}(K_{\mathfrak{A}}, \Delta_{\sigma}, \vDash)$ *is not distributive.*

Remark 6. *Let* $\Delta \subseteq F_1(\sigma)$, $\Delta_\sigma \subseteq \Delta$ *and the set* $\Delta \backslash \Delta_\sigma$ *consists of some conjunctions of formulas from* Δ_σ. *Then there exists a class* $K_{\mathfrak{A}}$ *such that the lattice* $(K_{\mathfrak{A}}, \Delta, \vDash)$ *is not distributive.*

For the set of all formulas the situation is better.

Proposition 5. *1) The lattice of formal concepts* $\underline{\mathfrak{B}}(K_{\mathfrak{A}}, F_1(\sigma), \vDash)$ *is distributive.*
2) The lattice of formal concepts $\underline{\mathfrak{B}}(K_{\mathfrak{A}}, F(\sigma), \vDash)$ *is distributive.*

Proof: in virtue of Proposition 2 and Proposition 3.

Association rule mining for the original context $(K_{\mathfrak{A}}, \Delta_{\sigma}, \vDash)$ does not produce a lot of rules with high confidence. A lot of various payment plans and services exist, and commonly more than one service can be useful for the subscriber. The service that will be preferred by the user depends on many factors. Some of these factors can change time to time. So we cannot detect such factors in scope of formal context $(K_{\mathfrak{A}}, \Delta_{\sigma}, \vDash)$ because the context is based on long users' history.

Moreover, mobile operator can suggest 2-3 possible services and the subscriber may select himself the most useful service. That is why it makes sense to add disjunctions of signature predicates to Δ_σ and use context $(K_{\mathfrak{A}}, \Delta_{\sigma}^{\vee}, \vDash)$ on next steps.

There are two problems with association rules that were mined using formal context $(K_{\mathfrak{A}}, \Delta_{\sigma}^{\vee}, \vDash)$. First of all some of rules have high confidence, but their conclusions are disjunctions of meaningfully nonrelated services. Such association rules could not be used for recommendations. It will be looking like spam for mobile network subscribers. So experts should process all rules and select only meaningful rules. Second, processing the whole formal context $(K_{\mathfrak{A}}, \Delta_{\sigma}^{\vee}, \vDash)$ is very laborious computational procedure.

To solve these problems we are moving from the Case Model $\mathfrak{A} = \langle A, \sigma \rangle$ to the Ontological Model $\langle \mathfrak{A}, T^a, T^s, T^f \rangle$ of the domain. We add new unary predicates to the signature σ to describe meaning of payment plans and services. Using new predicates (from the signature $\sigma_{\mathbb{R}}$) we generate automatically meaningful disjunctions of original predicates from the signature σ.

3 Ontological Model of the domain

3.1 Ontology

Ontological Model of the domain consists of four parts [15]:

(1) The domain ontology, i.e. description of the structure and the meaning of the domain concepts.

(2) General knowledge and domain regularities, sentences which are true for every case.

(3) The set of cases from the domain, that we consider in the given moment. This is empirical knowledge about the domain; the set of cases that we are looking at in this article is represented by the model $\mathfrak{A} = \langle A, \sigma \rangle$.

(4) Estimated and probabilistic knowledge: probabilistic and confidential estimates, fuzzy values of sentences [16].

In this section we describe construction of the domain ontology.

From a model-theoretic point of view the domain ontology construction consists of description of the signature and creation of a set of axioms that describe the meaning of the concepts of the domain [17, 18]. To define the signature $\sigma_{\mathbb{M}}$ of the domain $\mathbb{M}$ = "Mobile networks" we consider two sets of attributes: $\sigma_{\mathbb{P}}$, the set of individual subscriber's features and $\sigma_{\mathbb{Q}}$, the set of various payment plans and services.

The set of attributes $\sigma_{\mathbb{P}}$, "Individual subscribers' feature" consists of two parts: $\sigma_{\mathbb{P}_1}$, "payment plans" and $\sigma_{\mathbb{P}_2}$, "accrual". Every part $\sigma_{\mathbb{P}_i}$ $(i = \underline{1, 2})$ consists of two subparts, such as $\sigma_{\mathbb{P}_{i1}}$, "traffic (and accrual) without roaming inside operator network",..., $\sigma_{P_{i6}}$, "traffic (and accrual) in common roaming", … . Each of the listed signatures consists of more detailed categories, e.g., $\sigma_{P_{111}}$, "Traffic SMS without roaming inside operator network". Every category $\sigma_{P_{ijk}}$ contains finite number of signature symbols $P_{1ij}^{1}, \dots, P_{1ij}^{n}$. For example, $P_{111}^{1}(x) =$ "Traffic of SMS without roaming inside network for subscriber x is not more than 50 SMS in month" and $P_{112}^{1}(x) =$ "Traffic of SMS without roaming inside network for subscriber x is more than 50 SMS in month".

Signature $\sigma_{\mathbb{Q}}$ consists of two parts: $\sigma_{\mathbb{Q}_1}$ and $\sigma_{\mathbb{Q}_2}$. Part $\sigma_{\mathbb{Q}_1}$ is "payment plans", it has hierarchical structure and consists of symbols of unary predicates. Each unary predicate describes the presence or absence of connected payment plan for subscribers. Signature $\sigma_{\mathbb{Q}_2}$, "services and options", consists of symbols of unary predicates. Each unary predicate describes the presence or absence of connected service or option.

To describe the domain ontology, we define a finite set of ontological axioms $\mathcal{A}x_a \subseteq F_1(\sigma_{\mathbb{P}} \cup \sigma_{\mathbb{Q}})$. We introduce the following axioms.

Axioms of hyponym-hyperonym. Hierarchical structure of the signature $\sigma_{\mathbb{Q}_1}$ is represented by axioms such as:

$$(Q_{ijk}^{n}(x) \to Q_{ij}^{n}(x)) \ and \ (Q_{ij}^{n}(x) \to Q_i(x)).$$

Axioms of completeness. For each predicates inside every class $\sigma_{\mathbb{P}_{ijk}}$ and class $\sigma_{\mathbb{Q}_1}$ for a given subscriber there must be at least one true predicate. The schemes of such axioms are the following:

$$\vee \left\{P(x) \mid P(x) \in \sigma_{\mathbb{P}_{ijk}}\right\} \; and \; \vee \left\{P(x) \mid P(x) \in \sigma_{\mathbb{Q}_1}\right\}.$$

Axioms of including. For example, if payment plan t contains "more than 100 free SMS" then it contains "more 50 free SMS". The schemes of such axioms are the following:

$$(Q_{ijk}^{n_1}(x) \rightarrow Q_{ijk}^{n_2}(x)), \text{ where } n_2 < n_1.$$

Next we construct an extension $(\sigma_{\mathbb{P}} \cup \sigma_{\mathbb{Q}})$ of the signature $\sigma_{\mathbb{M}}$ by additional unary predicates that describe properties of payment plans and interests of subscribers. For that we introduce two types of concepts:

1) Concepts $\sigma_{\mathbb{R}}$. This is a set of features for different payment plans, services, and options. For example, amount of free calls time, volume of SMS package or of Internet package and etc. With the help of $\sigma_{\mathbb{R}}$ we can give formal definition for payment plans and services, i.e., formal definition of predicates of the signature $\sigma_{\mathbb{Q}}$.

2) Concepts $\sigma_{\mathbb{I}}$ describing subscriber's interests, e.g., reducing the costs of calls, SMS, etc.

Concepts from $(\sigma_{\mathbb{R}} \cup \sigma_{\mathbb{I}})$ are used for automation of construction formulas as attributes in formal contexts for association rules mining. Notice that the pair $\langle \sigma_{\mathbb{M}}, \mathcal{A}x \rangle$ forms the ontology of the domain $\mathbb{M}$ [18].

In the next step we introduce a new set of axioms $\mathcal{A}x_S \subseteq F_1(\sigma_{\mathbb{M}})$ and call it the domain axioms. This set will be used for describing various characteristics of payment plans and services provided at present moment of time by a mobile network.

Among other things, these axioms relate personal parameters of a subscriber. The range of parameters contains subscriber traffics denoted by predicates from $\sigma_{\mathbb{P}_1}$ and payments denoted by predicates from $\sigma_{\mathbb{P}_2}$, with regard to activated payment plans from $\sigma_{\mathbb{Q}_1}$ and services from $\sigma_{\mathbb{Q}_2}$.

Axioms $\mathcal{A}x_S$ are true for any case from the domain, and the same statement is true for ontological axioms as well. However, there is a difference between ontological and domain axioms, as the second ones might change over time. Consider the following formula as an example of a domain axiom:

$$(Q_1(x) \rightarrow \neg Q_2(x)), \qquad where \;\; Q_1 \in \sigma_{\mathbb{Q}_1}, Q_2 \in \sigma_{\mathbb{Q}_2}.$$

This formula declares the following: if a subscriber has payment plan Q_1 activated, then service Q_2 cannot be activated for this subscriber. Note that a mobile network company can naturally change its decision for not supporting simultaneously the precise payment plan along with the specific service, at any moment.

3.2 Ontological projections

In order to automate development of the formula set Δ for the sake of finding association rules, we use the Ontological Model of the domain.

Definition 4. An Ontological Model of a domain is a tuple $\langle \mathfrak{A}, T^a, T^s, T^f \rangle$, where T^a is an analytical theory of the domain, T^s is a theory of the domain, and T^f is a fuzzy theory of the model $\mathfrak{A}_{\mathbb{M}}$.

The analytical theory T^a of the domain under consideration is axiomatized by the sentences $\mathcal{A}x_a$ which are axioms of the domain ontology. A theory T^s of the domain is axiomatized by the axioms $\mathcal{A}x_S$ of the domain.

Formula definitions of predicates from $\sigma_{\mathbb{Q}}$ (which present payment plans and services) are defined by construction of ontological projection.

Definition 5. Consider the Ontological Model $\langle \mathfrak{A}, T^a, T^S, T^f \rangle$, let $Q \in \sigma_{\mathbb{Q}}$. Denote

$$S_Q = \{\varphi \in F_1(\sigma_{\mathbb{R}}) \mid T^a \vdash (Q(x) \to \varphi(x))\}.$$

An ontological projection of the predicate Q on the signature $\sigma_{\mathbb{R}}$ is the formula

$$\varphi_Q^{\sigma_{\mathbb{R}}}(x) = \&\{P(x) \mid P \in \sigma_{\mathbb{R}} \text{ and } P(x) \in S_Q\}.$$

A projection of the predicate Q on the set of formulas $F_1(\sigma_{\mathbb{R}})$ is the formula

$$\psi_Q^{\sigma_{\mathbb{R}}}(x) = \&S_Q = \&\{\varphi(x) \mid \varphi(x) \in S_Q\}.$$

Let us consider the formal context $C_{\mathfrak{A}}^{\Delta} = (K_{\mathfrak{A}}, \Delta, \vDash)$. We search association rules with the following requirements:

1) a) the premise of the association rule is included in the set $\Delta \upharpoonright \sigma_{\mathbb{P}}$ or
   b) the premise of the association rule is included in the set $\Delta \upharpoonright (\sigma_{\mathbb{P}} \cup \sigma_{\mathbb{Q}})$;
2) a) the conclusion of the association rule belongs to the set $\Delta \upharpoonright \sigma_{\mathbb{Q}}$ or
   b) the conclusion of the association rule belongs to the set $\Delta \upharpoonright \sigma_{\mathbb{R}}$ or
   c) the conclusion of the association rule belongs to the set $\Delta \upharpoonright (\sigma_{\mathbb{R}} \cup \sigma_{\mathbb{I}})$;
3) the support and the confidence of the rules are higher than specified limits.

Notice that the set of association rules of the formal context $C_{\mathfrak{A}}^{\Delta} = (K_{\mathfrak{A}}, \Delta, \vDash)$ is included in the fuzzy theory T^f of the model $\mathfrak{A}_{\mathbb{M}}$ [26, 27].

Then the software system automatically processes obtained association rules. For example, consider an association rule with one-element conclusion P belonging to $\sigma_{\mathbb{R}}$. This rule will be transformed into association rule with the same premise, but the conclusion of the new rule will be one-variable formula from Δ which is a disjunction of all predicates $Q_i \in \sigma_{\mathbb{Q}}$ such that P belongs to the ontological projection of Q_i.

4 Software Implementation

Using the results of the presented investigation, we have developed a software for mining association rules in the formal context $(K_{\mathfrak{A}}, \Delta_{\sigma}^{\vee}, \vDash)$. We have found out that adding predicates from $\sigma_{\mathbb{R}}$ to the formal context gives us the possibility to find association rules with high confidence and support. Conclusions of such rules are transformed into disjunctions of predicates from $\sigma_{\mathbb{Q}}$ with the help of the operator of ontology projection. Obtained association rules seem to be useful for mobile network companies. The software processes the impersonal data for more than 10 million subscribers. This is information for one month of mobile network using by subscribers.

The set of characteristics of subscribers contains more than 90 different items[1]:

1) Personal features of subscriber,
2) Attributes that describe calls made by subscriber,
3) Attributes that describe the mode of using Internet
4) Attributes that describe the mode of using SMS,
5) Attributes that describe the mode of using MMS,
6) Attributes that describe the mode of using LBS (Location Based Services),
7) List of mobile services that were connected to subscriber,
8) Payment plan that is used by subscriber.

The total amount of services that can be connected to subscriber is more than 90. The total count of different payment plans is more than 1200.

Thus, we have more than 10 million objects and nearly 1400 attributes. Part of attributes is quantitative, most part of attributes (more than 1200) are binary.

Let us notice that attributes of connected payment plans, services and personal attributes are always filled. That is why we use only quantitative attributes for density calculation. We calculate data density as follows: $\frac{P}{M*N}$, where P is the number of non-zero subscribers' attributes, M is the total number of subscribers, N is the number of quantitative attributes. For our data the data density is equal to 0.043.

The data is stored in a file with Basket format. Basket is one of standard formats for storing data of "objects-attributes" type in R.

Let us consider an example of association rules which have conclusions consisting of payment plans providing access to the Internet. The predicate $P(x) \in \sigma_{\mathbb{R}}$ denotes that subscriber's payment plan includes unlimited Internet traffic of the special kind[1]. The payment plans having the unlimited Internet traffic of this kind are $Q_1, Q_2, Q_3 \in \sigma_{\mathbb{Q}_1}$, where Q_1 is "Unlimited", Q_2 is "United", and Q_3 is "Online". These payment plans provide unlimited access to Internet with different connection speed and different price. Formally, in terms of ontological projections, it means that $P(x) \in S_{Q_1}$, $P(x) \in S_{Q_2}$, $P(x) \in S_{Q_3}$ and $P(x) \notin S_Q$ for every $Q \in \sigma_{\mathbb{Q}} \backslash \{Q_1, Q_2, Q_3\}$.

Mined association rules have premises with various sets of personal features of subscribers from $\sigma_{\mathbb{P}}$ and the conclusion $P(x)$. The automatically chosen rules have rather high confidence and support (see examples 1 and 2, table 1).

After that the predicate $P(x)$ is substituted by the equivalent disjunction $(Q_1 \vee Q_2 \vee Q_3)$ in the conclusions of the association rules. Table 1 shows that substituting the disjunction $(Q_1 \vee Q_2 \vee Q_3)$ by any of these predicates Q_i notably decreases both confidence and support of the association rules.

Thus, the new association rules generated by the algorithm in the extended formal context $(K_{\mathfrak{A}}, \Delta_\sigma^\vee, \vDash)$ have higher support and confidence as compared to rules with the same premise which may be found in the original formal context $(K_{\mathfrak{A}}, \Delta_\sigma^\vee, \vDash)$.

Table 1. Examples of association rules[1].

	Rule	Support	Confidence
Example 1	$\{\boldsymbol{P_1}, \dots, \boldsymbol{P_n}\} \to \boldsymbol{P}$	**11%**	**91%**
	$\{P_1, \dots, P_n\} \to Q_1$	6%	50%
	$\{P_1, \dots, P_n\} \to Q_2$	3%	23%
	$\{P_1, \dots, P_n\} \to Q_3$	2%	24%
Example 2	$\{\boldsymbol{P_1''}, \dots, \boldsymbol{P_l''}\} \to \boldsymbol{P}$	**11%**	**89%**
	$\{P_1'', \dots, P_l''\} \to Q_1$	4%	35%
	$\{P_1'', \dots, P_l''\} \to Q_2$	5%	38%

[1] Due to NDA, the details of the attribute list and characteristics P_i, Q_i cannot be given. So in the examples below, the real names of characteristics P_i and Q_i have been changed.

	$\{P_1'', \ldots, P_l''\} \to Q_3$	2%	18%
Example 3	$\{\boldsymbol{P_1'''}, \ldots, \boldsymbol{P_l'''}\} \to \boldsymbol{\varphi}$	**10%**	**82%**
	$\{P_1''', \ldots, P_l'''\} \to Q_1$	5%	40%
	$\{P_1''', \ldots, P_l'''\} \to Q_2$	4%	38%
	$\{P_1''', \ldots, P_l'''\} \to Q_3$	0.01%	0.08%
	$\{P_1''', \ldots, P_l'''\} \to T_1$	4%	45%
	$\{P_1''', \ldots, P_l'''\} \to T_2$	1%	5%
	$\{P_1''', \ldots, P_l'''\} \to T_3$	0.01%	0.06%
	$\{P_1''', \ldots, P_l'''\} \to T_4$	1%	3%
	$\{P_1''', \ldots, P_l'''\} \to (T_1 \vee T_2 \vee T_3 \vee T_4)$	6%	51%

If we would process association rules just in the formal context $(K_{\mathfrak{A}}, \Delta_\sigma^\vee, \models)$ without using the signature $\sigma_{\mathbb{R}}$, then many conclusions of mined rules will be non-meaningful disjunctions. Let us consider Example 3 in Table 1. Here $\varphi = (Q_1 \vee Q_2 \vee Q_3 \vee T_1 \vee T_2 \vee T_3 \vee T_4) \in \Delta_\sigma^\vee$, services Q_i provide unlimited Internet, and services T_i provide unlimited SMS. Here T_1 is "unlimited free SMS for month with a fixed price", T_2 is "1000 free SMS for month with a fixed price", T_3 is "unlimited cheap SMS", and T_4 is "discount for SMS, using with special conditions". The confidence of the association rule $\{P_1''', \ldots, P_l'''\} \to \varphi$ is high enough. The value is much greater than the confidence of rules $\{P_1''', \ldots, P_l'''\} \to Q_i$ and $\{P_1''', \ldots, P_l'''\} \to T_i$, but the mobile operator cannot use this association rule for recommendations, because it contains non-related services Q_i and T_i in the conclusion. However, if we consider the association rule $\{P_1''', \ldots, P_l'''\} \to (T_1 \vee T_2 \vee T_3 \vee T_4)$, we can see that this rule has low confidence.

5 Conclusion

The paper is devoted to methods for identifying payment plans and services by mobile operators which would be most useful for the given mobile network subscribers. We use the Case Model $\langle K_{\mathfrak{A}}, \sigma \rangle$ for mobile subscriber's behavior description. The Case Model is based on depersonalized subscribers' data provided by mobile operator. Objects (elements of the model) are mobile subscribers. The signature of the Case Model consists of unary predicates. These predicates describe individual subscriber's features (accruals, traffics) or features of payment plans and services. We construct the formal context $(K_{\mathfrak{A}}, \Delta, \models)$ based on the Case Model. Then we mine association rules describing payment plans and services that are commonly used by subscribers with given features. After that we consider the formal context $(K_{\mathfrak{A}}, \Delta_\sigma, \models)$. Our experiments show that interesting association rules have low confidence values in this context. That is why they cannot be used by mobile operator for any recommendations.

To improve association rules quality we deal with an extension of this formal context, the formal context $(K_{\mathfrak{A}}, \Delta_\sigma^V, \models)$. Using this context we can find association rules with high confidence. However, a big part of mined rules have conclusions which are disjunctions of non-related services, e.g. 'Song instead of Beep' and 'Unlimited Internet'. That is why such association rules could not be used for recommendations

Finally, we consider enriched signature $\sigma_{\mathbb{M}}$ instead of the signature σ to find semantically useful disjunctions. Signature $\sigma_{\mathbb{M}}$ contains predicates that describe specific features of payment plans and services. Using the formal context $(K_{\mathfrak{A}}, \Delta_{\sigma_{\mathbb{M}}}, \vDash)$ we compute association rules such that their conclusions are predicates of the signature $\sigma_{\mathbb{R}}$. We transform the obtained association rules into association rules of the formal context $(K_{\mathfrak{A}}, \Delta_{\sigma}^{V}, \vDash)$ using the Ontological Model $\langle K_{\mathfrak{A}}, T^{a}, T^{S}, T^{f}\rangle$. We substitute the predicate in the conclusion of an association rule by disjunction of predicates of the initial signature σ. As the result we obtain association rules of the formal context $(K_{\mathfrak{A}}, \Delta_{\sigma}^{V}, \vDash)$. These rules have high confidence and support, and the conclusions of these rules are completely meaningful for the mobile network operator as well as for mobile network subscribers. Mined association rules allow making recommendations for customers who will be interested in information about these services and tariffs.

Acknowledgment The reported study was partially supported by Russian Foundation for Basic Research, research project No. 14-07-00903-a.

References

1. Han, J., Pregibon, D., Mannila, H., Kumar, V., Altman, R. B.: Emerging scientific applications in data mining. Communications of the ACM, 45(8), 54–58 (2002)
2. Surendiran, R., Rajan, K.P., Sathishkumar, M.: Study on the Customer targeting using Association Rule Mining. International Journal on Computer Science and Engineering, 2(7), 2483–2484 (2010)
3. Priss, U.: Formal concept analysis in information science. In: Blaise C, ed. Annual Review of Information Science and Technology. ASIST. John Wiley & Sons (2006)
4. Kuznetsov, S. O.: Galois Connections in Data Analysis: Contributions from the Soviet Era and Modern Russian Research. In: Formal Concept Analysis: Foundations and Applications (B. Ganter, G. Stumme, R. Wille, Eds.), Lecture Notes in Artificial Intelligence (Springer), State-of-the Art Ser., 3626, 196-225 (2005)
5. Üstündag, A., Bal, M.: Evaluating Market Basket Data with Formal Concept Analysis. International Symposium Chaos, Complexity and Leadership, Springer Proceedings in Complexity (2012)
6. Furletti, B., Gabrielli, L., Renso, C., Rinzivillo, S.: Analysis of GSM calls data for understanding user mobility behavior. Proceedings of the IEEE International Conference on Big Data, 550–555 (2013)
7. Pawar, P., Aggarwal, A. K.: Associative Rule Mining of Mobile Data Services Usage for Preference Analysis, Personalization & Promotion. Proceedings of WSEAS International Conference on Simulation, Modeling and Optimization, Izmir, Turkey (2004)
8. Agrawal, R., Imieliński, T., Swami, A.: Mining association rules between sets of items in large databases. Proceedings of the 1993 ACM SIGMOD international conference on Management of data - SIGMOD '93 (1993)
9. Szathmary, L., Napoli, A., Kuznetsov, S.O.: ZART: A multifunctional itemset mining algorithm. CLA (2007)

10. Burusco, A., Fuentes-Gonzalez, R.: The study of L-fuzzy concept lattice. Mathw Soft Comput, 3, 209–218 (1994)
11. Chueh, H.-E.: Mining target-oriented fuzzy correlation rules to optimize telecom service management. International Journal of Computer Science & Information Technology, 3(1), 74 – 83 (2011)
12. Bosc, P., Pivert, O., Prade, H.: On fuzzy association rules based on fuzzy cardinalities. Proceedings of the IEEE International Fuzzy Systems Conference, 461-464 (2001)
13. Chueh, H.E., Lin, N.P., Jan, N.Y.: Mining Target-oriented Fuzzy Correlation Rules in Telecom Database. Proceedings of the 2009 International Conference on Advances in Social Networks Analysis and Mining, 399-404 (2009)
14. Lin, N. P., Chueh, H.-E.: Fuzzy Correlation Rules Mining. Proceedings of the 6^{th} WSEAS International Conference on Applied Computer Science, 13-18 (2007)
15. Naydanov, Ch.A., Palchunov, D.E., Sazonova, P.A.: Development of automated methods for the prevention of risks of critical conditions, based on the analysis of the knowledge extracted from the medical histories. Proceedings of the International Conference on Biomedical Engineering and Computational Technologies, Novosibirsk, 47-52 (2015)
16. Palchunov, D.E., Yakhyaeva, G.E., Yasinskaya, O.V.: Software system for the diagnosis of the spine diseases using case-based reasoning. Proceedings of the International Conference on Biomedical Engineering and Computational Technologies, Novosibirsk, 150-155 (2015)
17. Palchunov, D. E.: The solution of the problem of information retrieval based on ontologies. Biznes-informatika, 1, 3–13 (2008) (In Russian)
18. Palchunov, D.E.: Virtual catalog: the ontology-based technology for information retrieval. In: Knowledge Processing and Data Analysis. LNAI 6581, Springer-Verlag Berlin Heidelberg, 2011, pp. 164–183.
19. Kuznetsov, S.O., Poelmans, J.: Knowledge representation and processing with formal concept analysis. In: Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery, 3(3), 200-215 (2013).
20. Rudolph, S.: Exploring relational structures via FLE. In: Wolff K.E., Pfeiffer H.D., Delugach H.S., Eds. ICCS, LNAI 3127, 196–212 (2004)
21. Baader, F., Ganter, B., Sertkaya, B., Sattler, U.: Completing description logic knowledge bases using formal concept analysis. In: Eklund P, eds. Proceedings of the International Joint Conference on Artificial Intelligence, 230–235 (2007)
22. Palchunov, D. E.: Lattices of relatively axiomatizable classes. In: Lecture Notes in Artificial Intelligence, 4390, 221–239 (2007)
23. Ganter, B., Wille, R.: Formal Concept Analysis. Mathematical Foundations. Berlin: Springer (1999)
24. Yakhyaeva, G.E., Yasinskaya, O.V.: Application of Case-based Methodology for Early Diagnosis of Computer Attacks. Journal of Computing and Information Technology, 22(3), 145–150 (2014)
25. Kuznetsov, S. O., Mahalova, T.P.: Concept interestingness measures: a comparative study. Proceedings of the Twelfth International Conference on Concept Lattices and Their Applications Clermont-Ferrand, France, 1466, 59-72 (2015)
26. Palchunov, D.E., Yakhyaeva, G.E.: Fuzzy logics and fuzzy model theory. Algebra and Logic, 54(1), 74-80 (2015)
27. Yakhyaeva, G.E.: Fuzzy model truth values., Proceedings of the 6-th International Conference Aplimat, Bratislava, Slovak Republic, 423-431 (2007)

A Hybrid Data Mining Approach for the Identification of Biomarkers in Metabolomic Data

Dhouha Grissa[1,3], Blandine Comte[1], Estelle Pujos-Guillot[2], and Amedeo Napoli[3]

[1] INRA, UMR1019, UNH-MAPPING, F-63000 Clermont-Ferrand, France,
[2] INRA, UMR1019, Plateforme d'Exploration du Métabolisme, F-63000 Clermont-Ferrand, France
[3] LORIA, B.P. 239, F-54506 Vandoeuvre-lès-Nancy, France

Abstract. In this paper, we introduce an approach for analyzing complex biological data obtained from metabolomic analytical platforms. Such platforms generate massive and complex data that need appropriate methods for discovering meaningful biological information. The datasets to analyze consist in a limited set of individuals and a large set of attributes (variables). In this study, we are interested in mining metabolomic data to identify predictive biomarkers of metabolic diseases, such as type 2 diabetes. Our experiments show that a combination of numerical methods, e.g. SVM, Random Forests (RF), and ANOVA, with a symbolic method such as FCA, can be successfully used for discovering the best combination of predictive features. Our results show that RF and ANOVA seem to be the best suited methods for feature selection and discovery. We then use FCA for visualizing the markers in a suggestive and interpretable concept lattice. The outputs of our experiments consist in a short list of the 10 best potential predictive biomarkers.

Keywords: hybrid knowledge discovery, random forest, SVM, ANOVA, formal concept analysis, feature selection, biological data analysis, lattice-based visualization

1 Introduction

In the analysis of biological data, one of the challenges of metabolomics[1] is to identify, among thousands of features, predictive biomarkers[2] of disease development [13]. However, such a mining task is difficult as data generated by metabolomic platforms are massive, complex and noisy. In the current study,

[1] Metabolomics is the characterization of a biological system by the simultaneous measurement of metabolites (small molecules) present in the system and accessible for analysis. Data obtained are provided from different techniques and different analytical instruments.

[2] A biomarker, or biological marker, generally refers to a measurable indicator of some biological status or condition.

© Marianne Huchard, Sergei O. Kuznetsov (Eds.): CLA 2016, pp. 161–172, ISBN 978-5-600-01454-1, National Research University Higher School of Economics, 2016.

we aim at identifying from a large metabolomic dataset, predictive metabolic biomarkers of future T2D (type 2 diabetes) development, a few years before occurrence, in an homogeneous population considered healthy at the time of the analysis. The datasets include a rather limited number of individuals and a quite large set of variables. Specific data processing is required, e.g., feature selection. Accordingly, we propose a knowledge discovery process based on data mining methods for biomarker discovery from metabolomic data. The approach focuses on evaluating a combination of numeric-symbolic techniques for feature selection and evaluates their capacity to select relevant features for further use in predictive models. Actually, we need to apply feature selection for reducing dimension and avoid over-fitting[3]. The resulting reduced dataset is then used as a context for applying FCA [5] for visualization and interpretation. More precisely, we develop a hybrid data mining process which combines FCA with several numerical classifiers including Random Forest (RF) [3], Support Vector Machine (SVM) [16], and the Analysis of Variance (ANOVA) [4]. The dataset relies on a large number of numerical variables, e.g. molecules or fragments of molecules, a limited numbers of individuals, and one binary target variable, i.e. developing or not the disease a few years after the analysis. RF, SVM and ANOVA are used to discover discriminant biological patterns which are then organized and visualized thanks to FCA. Because it is known that the most discriminant[4] features may not be necessarily the best predictive[5] ones, it is essential to be able to compare different feature selection methods and to evaluate their capacity to select relevant features for further use in predictive models. The initial problem statement based on a data table of $individuals \times features$ is transformed into a binary table $features \times classification\,process$. Data preparation for feature selection is carried out using filter methods based on the correlation coefficient and mutual information to eliminate redundant/dependent features, to reduce the size of the data table and to prepare the application of RF, SVM and ANOVA.

A comparative study of the best k features from the combination of these different classification process (CP) –10 combinations of CP are considered– is performed. Then a binary data table is built consisting of $N\,features \times 10\,CP$. This binary table is considered as a formal context and as a starting point for the application of FCA and the construction of concept lattices. The features shared by all CP combinations can be interpreted as potential biomarkers of disease development. However, it is essential for biological experts to evaluate and compute the performances of the proposed biomarkers in models predicting the disease development a few years before occurrence. The performance of prediction models can be assessed using different methods. One classical method used by biologists for binary outcomes is the receiver operating characteristic

[3] The problem of over-fitting occurs when a statistical model describes random error or noise instead of the underlying relationship.

[4] A feature is said to be discriminant if it separates individuals in distinct classes (as, healthy vs not healthy).

[5] A feature is said to be predictive if it enables predicting the evolution of individuals towards the disease a few years later.

(ROC) curve [11], where the TPR (True positive rate) is plotted in function of the FDR (False discovery rate) for different cut-off points. A short list of the best predictive features is selected as the core set of biomarkers. Based on this selection, FCA is used to identify the top list of feature selection methods that provide the best ranking of these core set of biomarkers. This additional visualisation is essential for experts to discover the few best predictive biomarkers from the massive metabolomic dataset.

The remainder of this paper is organized as follows. Section 2 provides a description of related works. Section 3 presents the proposed approach and explains the methodological analysis of biomarker identification. Section 4 describes the experiments performed on a real-world metabolomic data set and discusses the results, while section 5 concludes the paper.

2 State of the art

In [14], the authors discuss the main research topics related to FCA and focus on works using FCA for knowledge discovery and ontology engineering in various application domains, such as text mining and web mining. They also discuss recent papers on applying FCA in bio-informatics, chemistry and medicine. Bartel et al. [1] are one of the first papers which apply FCA in chemistry. They use FCA to analyze the structure-activity relationships to predict the toxicity of chemical compounds. Gebert et al. [6] use an FCA-based model to identify combinatorial biomarkers of breast cancer from gene expression values. Since, the structure of gene expression data (GED) differs from metabolomic data, we can approve according to literature that FCA is never applied on metabolomic data. Indeed, the GED data tables include genes which are more or less expressed. Each gene is represented by a vector of values that explain the relative expression of the gene. This is totally different from metabolomic data where input data tables contain samples in rows and thousands of metabolites (small molecules) or feature in columns expressed as signal intensities. The goal is to identify metabolites that predict the evolution towards a clinical outcome. The processing of such metabolomic data is usually performed within different supervised learning techniques, such as PLS-DA (partial least squares discriminant analysis), PC-DFA (Principal component discriminant function analysis), LDA (Linear discriminant analysis), RF and SVM. Standard univariate statistical methodologies (as ANOVA or Student's t-test[6]) are also frequently used to analyze the metabolomic data [10]. In [8], authors show that there is no universal choice of method which is superior in all cases, even if they show that PLS-DA methods outperform the other approaches in terms of feature selection and classification. In a more detailed study [7], authors compare different variable selection approaches (LDA, PLS-DA with Variable Importance in Projection

[6] t-test or Student's t-test is a statistical hypothesis test which can be used to determine if two sets of data are significantly different from each other. If the p-value is below the threshold chosen for statistical significance (usually the 0.10, the 0.05, or 0.01 level), then the null hypothesis is rejected in favor of the alternative hypothesis.

(VIP), SVM-Recursive Feature Elimination (RFE), RF with Accuracy and Gini scores) in order to identify which of these methods are ideally suited to analyze a common set of metabolomic data, capable of classifying the Gram-positive bacteria Bacillus. They conclude that RF with its feature ranking techniques (mean decrease gini/accuracy) and SVM combined with SVM-RFE [9] as a variable selection method display the best results in comparison to other approaches. All these studies show that the choice of the appropriate algorithms is highly dependent on the dataset characteristics and the objective of the data mining process. In the field of biomarker discovery, SVM and RF algorithms prove to be robust for extracting relevant chemical and biological knowledge from complex data, in particular in metabolomics [7]. RF is a highly accurate classifier, based on a robust model to outlier detection (a sample point that is distant from other samples). Its main advantage [2] includes essentially its power to deal with overfitting and missing data, as well as its capacity to handle large datasets without variable elimination in terms of feature selection. Nevertheless, it generates unstable and volatile results, contrary to SVM which delivers a unique solution. These alternative approaches may be useful for data dimensionality reduction and feature selection purposes, and may be suitable to combine with FCA.

3 Design approach for Metabolomic data analysis

In this study, we design a hybrid data mining strategy based on the combination of numerical classifiers including RF, SVM, the univariate analysis ANOVA with the symbolic method FCA, to discover the best combination of biological features. In this work, we aim to find, from a large dataset, predictive metabolomic biomarkers of future T2D development.

We evaluate the proposed approach from a performance point of view. For this, we use Dell machine with ubuntu 14.04 LTS, a 3.60 GHZ $\times 8$ CPU and 15,6 GBi RAM. We perform all data analyses using the RStudio software (Version 0.98.1103, R 3.1.1) environment. Rstudio is available for free and offers a selection of packages suitable for different types of data.

3.1 Dataset description and pre-processing

Dataset description: we use a biological data set obtained from a case-control study within the GAZEL French population-based cohort (20 000 subjects). The data set includes the measurements (signal intensities) of 111 male subjects (54-64 years old) free of T2D at baseline. It consists in continuous numerical (semi quantitative) data which represent measurements performed on for each individual. Cases (55 subjects) who developed T2D at the follow-up belong to class '1' (diabetes) and are compared to Controls (56 subjects) which belong to class '-1' (healthy controls). A total of about three thousand features is generated after carrying out mass spectrometry (MS) analysis. But after noise filtration, each subject is described by 1195 features. In the rest of this paper, we consider this new filtered dataset of 1195 features, the original dataset.

The obtained dataset is then the result of an analysis performed on homogeneous individuals considered healthy at that time. However, the binary target variable describing the data classes is introduced based on the health status of the same individuals five years after the first analysis. Some of these individuals developed the disease at the follow-up. For this reason, we can not consider the discriminant features as the predictive ones, since features enabling a good separation between data classes (healthy vs not healthy) are not necessarily the same that predict the disease development a few years later.

Data pre-processing: the metabolomic database contains thousands of features with a wide intensity value range. A data preprocessing step is mandatory for adjusting the importance weights allocated to the features. Thus, before applying any FS method, except ANOVA, data are transformed using a Unit-Variance scaling method. It divides each feature value by its standard deviation; so that all features have the same chance to contribute to the model as they have an equal unit variance. The transformed dataset of 1195 features is used as input for all FS methods, except for ANOVA.

3.2 Feature selection for data dimensionality reduction

Only a few features (a small part of the original dataset) allow a good separation between data classes. Therefore, it is necessary to reduce data dimension to select a small number of relevant features for further use in predictive models. Reducing the dimensionality of the data is a challenging step, requiring a careful choice of appropriate feature selection techniques [15]. Filter and embedded methods are used for this purpose. We discarded wrapper approaches since they are greedy in computational cost.

The metabolomic data contain highly correlated features, which may impact the calculation of feature importance and ranking features [8]. To overcome this problem, we use two filter methods, the coefficient of correlation (Cor) and mutual information (MI). The first filter (Cor) is used to discard very highly correlated features, and the second filter (MI) is used to remove very dependent features. As embedded methods [12], we retain two FS techniques that are widely used on biological data, which are RF and SVM.

Figure 1 describes the feature selection workflow we propose to obtain a reduced set of relevant features. This workflow considers at the beginning the filter methods 'Cor' and 'MI' to eliminate redundant/dependent features. In order to limit the loss of information, very highly correlated features are discarded (one feature per group of correlated ones is kept) to keep a reasonable number of features to work with. All the features whose MI average values are smaller than the threshold are selected, since it is known that high mutual information is indicating a large reduction of uncertainty [17]. We then set correlation and mutual information thresholds to 0.95 and 0.02, respectively. Consequently, two reduced subsets are generated: the first subset contains 963 features after 'Cor' filter, and the second one contains 590 features after 'MI' filter. When we fix

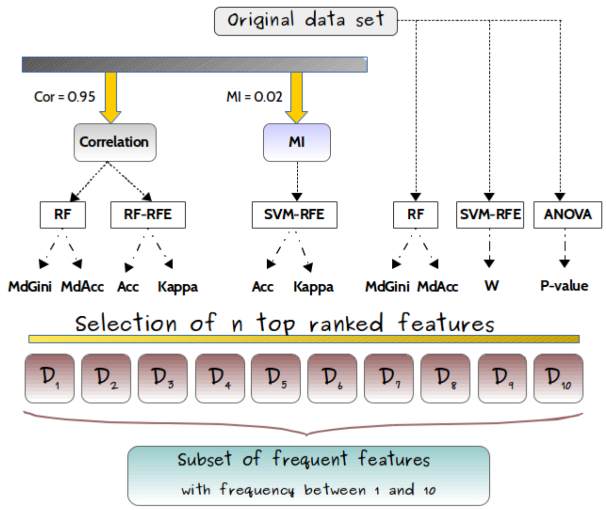


Fig. 1. Feature selection and dimensionality reduction process.

a lower threshold of correlation, we remove a lot of features since the original dataset is very correlated. When we set the MI threshold to a lower value, we keep only a small number of features and consequently we may loose a lot of information.

Both reduced subsets are used as input for the application of RF and SVM classifiers. Nonetheless, as correlation values between variables are still high, we furthermore adapt the RFE[7] approach with RF and SVM. To cover various possible classification results, we apply the embedded methods RF, RF-RFE and SVM-RFE on both filtered subsets. We also apply the ANOVA method on the original data set (not transformed) since it is commonly applied on metabolomic data. Three different classification models are respectively obtained. The first model is built from the application of RF on data filtered with Cor. The second classification model is fitted according to RF-RFE also on the subset of data filtered with 'Cor'. The third model is built from the application of SVM-RFE on the subset of data filtered with 'MI'. Based on these three classification models, we use several accuracy metrics to measure the importance of each feature in the overall result. These measures include MdGini[8], MdAcc[9], Accuracy, and

[7] Recursive Feature Elimination (RFE) is a backward elimination method, originally proposed by Guyon et al. [9] for binary classification. This is one of the classical embedded methods for feature selection with SVM.

[8] Mean decrease in Gini index (MdGini) provides a measure of the internal structure of the data.

[9] Mean decrease in accuracy (MdAcc) measures the importance/performance of each feature to the classification. The general idea of these metrics is to permute the values of each variable and measure the decrease in the accuracy of the model.

Kappa[10]. The scores given by these metrics enable ranking the features by means of the classification models already built.

When no filter is used, three feature selection techniques (SVM-RFE, RF and ANOVA) are applied directly to the original dataset using the feature weight values 'W' (i.e. the weight magnitude of features), p-value[11], MdGini and MdAcc scores to sort the features and identify those with the highest discriminative power. Various forms of results (feature ranking, feature weighting, etc.) and multiple (sub)sets of ranked features are obtained as output. In total, 10 (sub)sets are generated, corresponding to the different CP and ranking scores (Figure 1). For each CP, we give a corresponding name that well describe the whole classification process. The first CP is called 'Cor-RF-MdAcc', which means that we apply firstly the correlation coefficient 'Cor', then we apply RF on the obtained set and rank features according to MdAcc. We follow the same logic to name the other CP: (2) 'Cor-RF-MdGini', (3) 'Cor-RF-RFE-Acc', (4) 'Cor-RF-RFE-Kap', (5) 'MI-SVM-RFE-Acc', (6) 'MI-SVM-RFE-Kap', (7) 'RF-MdAcc', (8) 'RF-MdGini', (9) 'SVM-RFE-W' and (10) 'ANOVA-pValue'. To preserve only important features, we retain the 200 first ranked ones from each of the 10 (sub)sets, except the set 'ANOVA-pValue' from which we select only 107 features that have a reasonable p-value (lower than 0.1). Ten reduced sets of ranked features are consequently obtained, named D_i, where $i \in \{1, \ldots, 10\}$. Then, to analyze the relative importance of individual features and to enable a comprehensive interpretation of the results, these reduced sets of ranked features are combined for comparison.

3.3 Visualization with FCA

This section focuses on comparing all the reduced sets (D_i, where $i \in \{1, \ldots, 10\}$) of highly ranked features (Figure 1). The combination of these subsets resulting from different CP, enables covering several possible results and yields to a stable unique reduced output. For the comparison propose, a binary table of *features* $\times$ *CP* is built (e.g., Table 1), where the objects (rows) are the features and the variables (columns) are the 10 CP. We put '1' if the feature exists in the reduced set of a corresponding CP; otherwise, we put '0'. Each feature has then a support[12] calculated from the obtained binary table, where the most frequent features are those existing in all the reduced sets (support =10). Nevertheless, since we are looking for frequent features according to the different CP, a subset of features common to at least 6 techniques is selected (i.e., features belonging to D_i, where $i \in \{1, \ldots, 10\}$ and identified by at least 6 CP), and a new subset of 48 frequent features is obtained. The choice of this value (6) is not random,

[10] Cohens Kappa (Kappa) is a statistical measure which compares an Observed Accuracy with an Expected Accuracy (random chance)

[11] A p-value helps determining the statistical significance of the results when a hypothesis test is performed.

[12] The support is the number of times we have '1' in each row, according to the binary table.

but it enables obtaining results from complementary FS methods. It ensures the selection of some relevant features that could have been removed by filters, while keeping a reasonable dataset size (48 features). A new binary table of the form 48 $features \times 10\, CP$ is obtained and presented in Table 1. It describes features in rows by the CP in columns and transforms then the initial problem statement from a data table of 111 $individuals \times 1195\, features$ to 48 $features \times 10\, CP$. The labels of the features start with the word 'm/z' which corresponds to the mass per charge value.

From this (48×10) binary table, we apply FCA with the help of ConExp tool [18]). Two seventy six concepts are obtained from the derived concept lattice (Figure 2). The combination of FCA with the results of the numerical methods and the transformation of the problem statement bring new light to the generated data. Four features 'm/z 383', 'm/z 227', 'm/z 114' and 'm/z 165' of the subconcept are identified as the most frequent (maximum rectangle full of 1 in Table 1). Most of the 44 remaining features highlight strong relationships between each others, such as 'm/z 284', 'm/z 204', 'm/z 132', 'm/z 187', 'm/z 219', 'm/z 203', 'm/z 109', 'm/z 97' and 'm/z 145'. Among the 48 frequent features, 39 are significant w.r.t. ANOVA (have a pvalue$<$0.05). The generated lattice highlights then the potential of the proposed feature selection approach for analyzing metabolomic data. It enables discriminating direct and indirect associations: highly linked metabolites belonging to the same concept. The links between the concepts in the lattice represent the degree of interdependencies between concept and metabolites belonging to the same concept. These 48 frequent features are then proposed as candidate for prediction.

4 Evaluation and discussion

4.1 Predictive performance evaluation and interpretation

Considering the 48 most frequent features previously identified, we would like to evaluate their predictive capacities. Accordingly, we start the performance evaluation using the ROC curves (Figure 3) of the 48 features with associated confidence intervals. These analyses are performed using the ROCCET tool (http://www.roccet.ca), with calculation of the area under the curve (AUC) and confidence intervals (CI), calculation of the true positive rate (TPR), where $TPR = TP/(TP + FN)$, and the false discovery rate (FDR), where $FDR = TN/(TN+FP)$. The p-values of these relevant features are also computed using t-test.

ROC curve is a non-parametric analysis, which is considered to be one of the most objective and statistically valid method for biomarker performance evaluation [11]. They are commonly used to evaluate the prediction performance of a set of features, or their accuracy to discriminate diseased cases from normal cases. Since the number of features to propose as biomarkers requires to be quite limited (because of clinical constraints), we rely on the ROC curves of the top 2, 3, 5, 10, 20 and 48 of important features ranked based on their AUC values. These small sets of features are used to build the RF classification models based on the

Table 1. Input binary table describing the 48 frequent features with the 10 CP.

Features	Cor-RF-MdGini	Cor-RF-MdAcc	Cor-RF-RFE-Acc	Cor-RF-RFE-Kap	RF-MdGini	RF-MdAcc	MI-SVM-RFE-Acc	MI-SVM-RFE-Kap	SVM-RFE-W	ANOVA-pValue
m/z 383	1	1	1	1	1	1	1	1	1	1
m/z 227	1	1	1	1	1	1	1	1	1	1
m/z 114	1	1	1	1	1	1	1	1	1	1
m/z 165	1	1	1	1	1	1	1	1	1	1
m/z 145	1	1	1	1	1	1	1	1		1
m/z 97	1	1	1	1	1	1	1	1		1
m/z 441	1	1	1	1	1	1	1		1	1
m/z 109	1	1	1	1	1	1	1	1		1
m/z 203	1	1	1	1	1	1	1	1		1
m/z 219	1	1	1	1	1	1	1	1		1
m/z 198	1		1	1	1	1	1	1	1	1
m/z 263	1	1	1		1	1	1	1	1	1
m/z 187	1	1	1	1	1	1	1	1		1
m/z 132	1	1	1	1	1	1	1	1		1
m/z 204	1	1	1	1	1	1	1	1		1
m/z 261	1	1	1	1	1	1	1	1	1	
m/z 162	1	1			1	1	1	1	1	1
m/z 284	1	1	1	1	1	1	1	1		1
m/z 603	1	1	1	1	1	1			1	1
m/z 148	1	1	1	1	1	1		1		1
m/z 575	1	1	1	1	1	1	1	1		
m/z 69	1	1			1	1	1	1		1
m/z 325	1	1			1	1	1	1		1
m/z 405	1	1			1	1	1		1	1
m/z 929	1	1	1	1	1	1	1	1		
m/z 58	1	1	1	1	1	1	1	1		
m/z 336	1	1	1	1	1	1	1			1
m/z 146	1		1	1	1		1	1		1
m/z 104	1		1	1	1		1	1		1
m/z 120			1	1	1	1	1	1	1	1
m/z 558	1	1			1	1	1	1	1	
m/z 231					1	1	1	1	1	1
m/z 132*	1	1	1	1	1		1	1		
m/z 93		1	1	1	1	1	1	1		
m/z 907		1	1	1		1	1	1	1	
m/z 279		1	1	1	1	1	1	1		
m/z 104*	1	1			1	1	1	1		1
m/z 90	1	1			1	1	1	1		1
m/z 268					1	1	1	1	1	1
m/z 288*	1	1	1	1	1	1				1
m/z 287	1	1	1	1	1	1				1
m/z 167	1	1			1	1	1	1		1
m/z 288	1	1	1	1	1	1				1
m/z 252	1	1			1	1	1	1		1
m/z 141	1		1	1	1		1	1		1
m/z 275			1	1	1		1	1		1
m/z 148*			1	1	1		1	1		1
m/z 92	1	1			1	1	1	1		1

cross validation (CV) performance. The ROC curves enable identifying this best combination of predictive features. Figure 3 shows that the best performance is

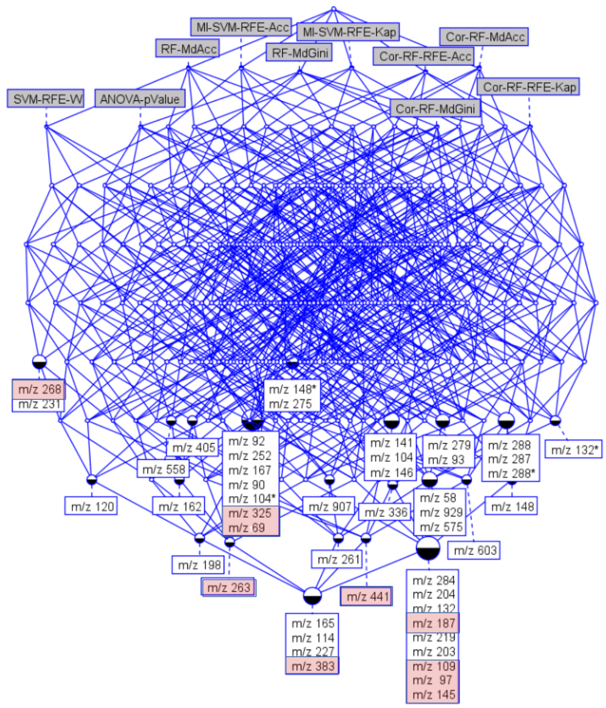


Fig. 2. The concept lattice derived from the 48 × 10 binary table (Table 1).

given to the 48 features together (AUC=0.867). But a predictive model with 48 metabolites is not useable in clinical practices. The set of best features with the smallest p-values and the highest accuracy values is selected to finally obtain a short list of potential biomarkers. When we select the ten first features (Table 3), we have an AUC equals to 0.79, and a CI=0.71-0.9. When we select the first four features, we obtain an AUC close to 0.75. These high AUC values show a good predictive performance.

In sight of these results, it is more advisable to select the 10 first features which have an AUC greater than 0.74 and a significant small t-test values (Table 3) as potential biomarkers. We compare this subset of 10 best predictive features with the four most frequent features (features with full of '1' in Table 1), we find that only one feature is in common, 'm/z 383'. We conclude that the core set of most frequent features is not the best predictive set, as expected biologically because the metabolomic analyses are performed 5 years before disease occurrence. Moreover, these best predictive features (or potential biomarkers) are not belonging to the same concept. Figure 2 highlights this conclusion and shows that the best predictive biomarkers have different extents and belong to concepts with different intents. They are depicted by the red squares in the lat-

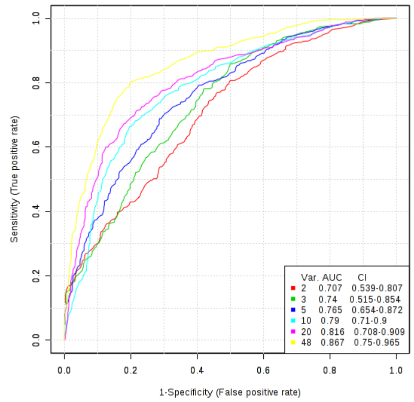


Fig. 3. The ROC curves of at least 2 and max 48 combined frequent features based on RF model and AUC ranking.

tice. For example, the features 'm/z 145', 'm/z 97', 'm/z 109' and 'm/z 187' are part of the intent of a concept including all the CP, except 'SVM-RFE-W', in extent. By contrast, the feature 'm/z 268' belongs to another concept including 6 CP in extent (*'RF-MdGini', 'RF-MdAcc', 'MI-SVM-RFE-Acc', 'MI-SVM-RFE-Kap', 'SVM-RFE-W', 'ANOVA-pValue'*). Here again, the simple visualization of the lattice comes to highlight the position of the predictive features among the discriminant ones and shows the associations with selection methods. This information is interesting for the expert domain since this visualization allows choosing the best combination of feature selection methods.

4.2 Selection of the best FS method(s)

As some feature selection methods do not retain the ten best predictive ones as their highly ranked, it remains essential to identify the methods that provide the best selection from metabolomic data. Here again, FCA comes to highlight and to assist information retrieval and visualization of the results. We then retain only the subset of ten best features (*'m/z 145', 'm/z 441', 'm/z 383', 'm/z 97', 'm/z 325', 'm/z 69', 'm/z 268', 'm/z 263', 'm/z 187' and 'm/z 109'*) identified previously due to the ROC curve, and apply FCA another time on their corresponding binary Table 2. A new concept lattice is generated (Figure 4) showing a superconcept with 4 feature selection methods, 'ANOVA-pValue', 'MI-SVM-RFE-Acc', 'RF-MdAcc' and 'RF-MdGini', verified by all features.

This is a very interesting result which needs a deeper interpretation before validation. We then consider these 4 methods and look for their ranking w.r.t. the 10 best predictive features (Table 3). Table 4 shows that RF-based techniques and Anova provide a good ranking to the 10 features contrarily to 'MI-SVM-RFE-Acc'. For example, 'm/z 145' is ranked first according to 'RF-MdAcc', 'RF-

Table 2. Input binary table describing the 6 best predictive features with the 10 CP.

Features	Cor-RF-MdGini	Cor-RF-MdAcc	Cor-RF-RFE-Acc	Cor-RF-RFE-Kap	RF-MdGini	RF-MdAcc	MI-SVM-RFE-Acc	MI-SVM-RFE-Kap	SVM-RFE-W	ANOVA-pValue
m/z 383	1	1	1	1	1	1	1	1	1	1
m/z 145	1	1	1	1	1	1	1	1		1
m/z 97	1	1	1	1	1	1	1	1		1
m/z 263	1	1	1		1	1	1	1	1	1
m/z 325	1	1			1	1	1	1		1
m/z 268					1	1	1	1	1	1

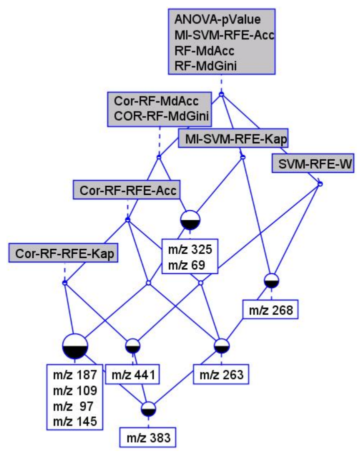


Fig. 4. The concept lattice of the 10 best predictive variables.

MdGini', second according to 'ANOVA-pvalue' and hundredth within 'MI-SVM-RFE-Acc'. The feature 'm/z 441' is ranked 6th according to 'RF-MdAcc', 8th within 'RF-MdGini', 172th within 'MI-SVM-RFE-Acc', and 11th according to 'ANOVA-pvalue'. Consequently, the toplist methods for biomarker identification from metabolomic data are RF-based and ANOVA.

5 Conclusion and future works

In this paper, we presented a new approach for the identification of predictive biomarkers from complex metabolomic dataset. Due to the nature of metabolomic data (highly correlated and noisy), the results highlighted the importance of working on reduced datasets to identify important variables related to the observed discrimination between case and control subjects and candidate for pre-

Name	AUC	T-tests
m/z 145	0.79	1.4483E-6
m/z 383	0.79	5.0394E-7
m/z 97	0.78	1.5972E-6
m/z 325	0.77	2.2332E-5
m/z 69	0.76	1.2361E-5
m/z 268	0.75	4.564E-6
m/z 441	0.75	9.0409E-5
m/z 263	0.75	5.996E-6
m/z 187	0.74	9.0708E-6
m/z 109	0.74	2.6369E-5

Table 3. Table of performance of the best 10 AUC ranked features.

Feature	RF-MdAcc	RF-MdGini	MI-SVM-RFE-Acc	ANOVA-pValue
m/z 145	1	1	100	2
m/z 383	3	3	40	1
m/z 97	2	2	63	3
m/z 325	5	5	38	8
m/z 69	4	4	65	7
m/z 268	9	6	168	4
m/z 441	6	8	172	11
m/z 263	8	7	28	5
m/z 187	14	10	27	6
m/z 109	7	9	37	9

Table 4. Ranking of the 10 features with respect to 4 CP.

diction. Indeed, a combination of numerical (supervised) and symbolic (unsupervised) methods remains the best approach, as it allows combining the strengths of both techniques.

In this study, we used machine learning methods, RF and SVM, that we combined with FCA, to select a subset of good candidate biological features for prediction diseases. Our results showed the interest of this association to reveal subtle effects (hidden information) in such high dimensional datasets and how FCA highlighted the relationship between the best predictive features and the selection methods. RF-based methods as well as ANOVA gave the toplist of relevant features that best predict the disease development. With this help, the experts in biology will go deeper in interpretation, attesting the success of the knowledge discovery process. Additional experiments on other metabolomic datasets are required to attest the success of the proposed approach.

References

1. Bartel, H.G., Brüggemann, R.: Application of formal concept analysis to structure-activity relationships. Fresenius' Journal of Analytical Chemistry 361(1), 23–28 (1998)
2. Biau, G.: Analysis of a random forests model. J. Mach. Learn. Res. 13(1), 1063–1095 (2012)
3. Breiman, L.: Random forests. In: Machine Learning. pp. 5–32 (2001)
4. Cho, H., Kim, S.B., Jeong, M.K., Park, Y., Miller, N.G., Ziegler, T.R., Jones, D.P.: Discovery of metabolite features for the modelling and analysis of high-resolution nmr spectra. International Journal of Data Mining and Bioinformatics 2(2), 176–192 (2008)

5. Ganter, B., Wille, R.: Formal Concept Analysis – Mathematical Foundations. Springer (1999)
6. Gebert, J., Motameny, S., Faigle, U., Forst, C., Schrader, R.: Identifying Genes of Gene Regulatory Networks Using Formal Concept Analysis. Journal of Computational Biology 2, 185–194 (2008)
7. Gromski, P.S., Xu, Y., Correa, E., Ellis, D.I., Turner, M.L., Goodacre, R.: A comparative investigation of modern feature selection and classification approaches for the analysis of mass spectrometry data. Analytica Chimica Acta 829, 1–8 (2014)
8. Gromski, P., Muhamadali, H., Ellis, D., Xu, Y., Correa, E., Turner, M., Goodacre, R.: A tutorial review: Metabolomics and partial least squares-discriminant analysis–a marriage of convenience or a shotgun wedding. Anal Chim Acta. 879, 10–23 (2015)
9. Guyon, I., Weston, J., Barnhill, S., Vapnik, V.: Gene selection for cancer classification using support vector machines. Mach. Learn. 46, 389–422 (2002)
10. Jansen, J., Hoefsloot, H., van der Greef, J., Timmerman, M., Westerhuis, J., Smilde, A.: Asca: analysis of multivariate data obtained from an experimental design. Journal of Chemometrics 19(9), 469–481 (2005)
11. Jianguo Xia, David I. Broadhurst, M.W., author, D.S.W.: Translational biomarker discovery in clinical metabolomics: an introductory tutorial. Metabolomics 9(2), 280–99 (2013)
12. Lal, T.N., Chapelle, O., Weston, J., Elisseeff, A.: Feature Extraction: Foundations and Applications, chap. Embedded Methods, pp. 137–165. Springer Berlin Heidelberg (2006)
13. Mamas, M., Dunn, W., Neyses, L., Goodacre, R.: The role of metabolites and metabolomics in clinically applicable biomarkers of disease. Arch Toxicol. 85(1), 5–17 (2011)
14. Poelmans, J., Ignatov, D.I., Kuznetsov, S.O., Dedene, G.: Formal concept analysis in knowledge processing: A survey on applications. Expert Systems with Applications 40(16), 6538 – 6560 (2013)
15. Saeys, Y., Inza, I., Larraaga, P.: A review of feature selection techniques in bioinformatics. Bioinformatics 23(19), 2507–2517 (2007)
16. Vapnik, V.: Statistical Learning Theory. Wiley-Interscience, John Willey & Sons (1998)
17. Wang, H., Khoshgoftaar, T.M., Wald, R.: Measuring Stability of Feature Selection Techniques on Real-World Software Datasets. Information Reuse and Integration in Academia and Industry, pp. 113–132. Springer Vienna (2013)
18. Yevtushenko, S.A.: System of data analysis "concept explorer". In: Proceedings of the 7th national conference on Artificial Intelligence. pp. 127–134. KII'2000 (2000)

Generalized Metrics and Their Relevance for FCA and Closure Operators

Tobias Gäbel-Hökenschnieder[1], Thorsten Pfeiffer[2], Mosadak Al Salamat[1], and Stefan E. Schmidt[1]

[1] Institut für Algebra, Technische Universität Dresden,
[2] FINMA, Bern, Schweiz
tgh@mail.de, thorsten_pfeiffer@yahoo.com, mosadak@gmx.de, midt1@msn.com

Abstract. We provide an approach to generalized metrics that covers various concepts of distance. In particular, we consider functorial maps which are weakly positive. Here, we focus on the supermodular case which generalizes dimension functions. We give a lattice-theoretically based construction for supermodular functorial maps, which generalize those arising from Dempster-Shafer-Theory. Within this framework, generalized metrics relevant for FCA and closure operators are discussed.

Keywords: Generalized metric, supermodular, formal concept analysis, Dempster-Shafer-Theory, closure operators

1 Introduction

Generalized metrics recently have become of increased interest for modelling a concept of directed distances with values in a qualitative measurement space. In particular, they allow to distinguish between *deletion* and *error* within the context of transferred information. We propose a general modeling including lattices and ordered monoids. Here, our goal is to construct *generalized metrics* relevant for FCA and closure operators [3, 8, 7]. For our approach it turns out that *supermodularity* plays an important role, which goes beyond ideas of measurement accociated with Dempster-Shafer-Theory [8].

Our modeling of *generalized metrics* can be very helpful to improve and better understand the mapping of ratings, i. e. compare the rating methodologies of different rating agencies with different result scales.

2 Motivation

Before we present *generalized metrics* in an abstract setting, we want to discuss a motivating special situation where we collect properties relevant for our general approach.

© Marianne Huchard, Sergei O. Kuznetsov (Eds.): CLA 2016, pp. 175–186,
ISBN 978-5-600-01454-1, National Research University Higher School of Economics, 2016.

We start with a lattice $\mathbb{L} = (L, \leq_{\mathbb{L}})$. Then, we consider the ordered monoid $\mathcal{M} = (M, \cup, \emptyset, \subseteq)$ with $M = 2^L$. Furthermore, we set

$$\downarrow x := \{t \in L \mid t \leq_{\mathbb{L}} x\}$$

and define the maps

$$\lambda\colon L \longrightarrow M : x \mapsto \downarrow x$$

and

$$D_\lambda\colon\ \leq_{\mathbb{L}} \longrightarrow M : (x, y) \mapsto \downarrow y - \downarrow x. \tag{1}$$

We can easily see that $D_\lambda(x, y)$ is equal to the set $\{t \in L \mid t \leq_{\mathbb{L}} y \text{ and } t \not\leq_{\mathbb{L}} x\}$, where $x, y \in L$ and $x \leq_{\mathbb{L}} y$.

Observation Obviously, D_λ fulfils the following properties:

- $D_\lambda(x, x) = \emptyset$ holds for all $x \in L$, since $D_\lambda(x, x) = \downarrow x - \downarrow x = \emptyset$.
- $D_\lambda(x, y) \cup D_\lambda(y, z) = D_\lambda(x, z)$ holds for all $x, y, z \in L$ with $x \leq_{\mathbb{L}} y \leq_{\mathbb{L}} z$, since

$$\begin{aligned} D_\lambda(x, y) \cup D_\lambda(y, z) &= (\downarrow y - \downarrow x)\ \cup\ (\downarrow y - \downarrow z) \\ &= \downarrow z - \downarrow x \\ &= D_\lambda(x, z). \end{aligned}$$

Satisfying these conditions, D_λ will be called *functorial* (see definition 2).

Now we want to put the previous considerations in a slightly more general setting. As above, the starting point is a lattice $\mathbb{L} = (L, \leq_{\mathbb{L}})$. For a given subset A of L we consider the ordered monoid $\mathcal{M} = (M, \cup, \emptyset, \subseteq)$ with $M = 2^A$. Then, for all $x \in L$ we define

$$Ax := \{a \in A \mid a \leq_{\mathbb{L}} x\}.$$

Based on this setup, we consider the following maps:

$$\begin{aligned} \lambda\colon L &\longrightarrow M : x \mapsto Ax, \\ D_\lambda\colon\ \leq_{\mathbb{L}} &\longrightarrow M : (x, y) \mapsto Ay - Ax. \end{aligned} \tag{2}$$

Claim 1 *D_λ is functorial w. r. t. $(\mathbb{L}, \mathcal{M})$.*

Proof. – Firstly, $D_\lambda(x, x) = \emptyset$ obviously holds for all $x \in L$.

- Secondly, we have to show that $D_\lambda(x, y) \cup D_\lambda(y, z) = D_\lambda(x, z)$ holds for all $x, y, z \in L$ with $x \leq_{\mathbb{L}} y \leq_{\mathbb{L}} z$:

  Let x, y, z be elements in L such that $x \leq_{\mathbb{L}} y \leq_{\mathbb{L}} z$. For all $a \in A$, $a \in D_\lambda(x, z)$ is equivalent to

  $$a \not\leq_{\mathbb{L}} x \quad \text{and} \quad a \leq_{\mathbb{L}} z.$$

  Let $a \in D_\lambda(x, z)$. We distinguish two situations:

Case 1: $\boxed{a \leq_{\mathbb{L}} y}$ Then, $a \nleq_{\mathbb{L}} x$ and $a \leq_{\mathbb{L}} y$, that is $a \in D_\lambda(x, y)$.

Case 2: $\boxed{a \nleq_{\mathbb{L}} y}$ Then, $a \nleq_{\mathbb{L}} y$ and $a \leq_{\mathbb{L}} z$, that is $a \in D_\lambda(y, z)$.

Hence, $a \in D_\lambda(x, y) \cup D_\lambda(y, z)$.

On the other hand, assume $a \in D_\lambda(x, y) \cup D_\lambda(y, z)$. Hence, $a \nleq_{\mathbb{L}} x$ and $a \leq_{\mathbb{L}} y$, or $a \nleq_{\mathbb{L}} y$ and $a \leq_{\mathbb{L}} z$. Then, $a \nleq_{\mathbb{L}} x$ (since $x \leq_{\mathbb{L}} y$) and $a \leq_{\mathbb{L}} z$ (since $y \leq_{\mathbb{L}} z$) which yields $a \in D_\lambda(x, z)$. □

In (2), we introduced D_λ as a function with domain $\leq_{\mathbb{L}}$. Next, we want to look for an extension of D_λ onto $L \times L$. We achieve this by the following map:

$$d_\lambda \colon L \times L \longrightarrow M : (x, y) \mapsto D_\lambda(x \wedge y, y) \tag{3}$$

Claim 2 *The map* d_λ *is a* generalized quasi metric (GQM) *w. r. t.* $(L, \mathcal{M})$*, that is, the subsequent conditions are satisfied:*

(A0) *for all* $x, y \in L: \quad \emptyset \subseteq d_\lambda(x, y)$,

(A1) *for all* $x \in L: \quad d_\lambda(x, x) = \emptyset$,

(A2) *for all* $x, y, z \in L: \quad d_\lambda(x, z) \subseteq d_\lambda(x, y) \cup d_\lambda(y, z)$.

We remind the reader that A is called *join-dense* in $\mathbb{L}$ if for all $x, y \in L$ with $x \nleq_{\mathbb{L}} y$ there exists $a \in A$ such that $a \nleq_{\mathbb{L}} y$ and $a \leq_{\mathbb{L}} x$.

Claim 3 *Let* A *be join-dense in* $\mathbb{L}$*. Then* d_λ *is a* generalized metric (GM) *w. r. t.* $(L, \mathcal{M})$*, that is,* d_λ *is a GQM which additionally satisfies:*

(A3) *For all* $x, y \in L: \quad d_\lambda(x, y) = \emptyset = d_\lambda(y, x) \quad \Longrightarrow \quad x = y$.

A more general definition for the underlying concepts will be given in definition 3.

Proof. (A0) Obviously, for all $x, y \in L$, the condition $\emptyset \subseteq d_\lambda(x, y)$ is satisfied.

(A1) Clear, since for all $x \in L : d_\lambda(x, x) = \emptyset$.

(A2) We have to show that $d_\lambda(x, z) \subseteq d_\lambda(x, y) \cup d_\lambda(y, z)$ holds for all $x, y, z \in L$. This is equivalent to

$$D_\lambda(x \wedge z, z) \subseteq D_\lambda(x \wedge y, y) \cup D_\lambda(y \wedge z, z).$$

To do so, let $a \in D_\lambda(x \wedge z, z)$. Hence, $a \nleq_{\mathbb{L}} x \wedge z$ and $a \leq_{\mathbb{L}} z$, which implies

$$a \nleq_{\mathbb{L}} x \quad \text{and} \quad a \leq_{\mathbb{L}} z.$$

We have to examine two cases:

Case 1: $\boxed{a \leq_{\mathbb{L}} y}$ Hence, $a \nleq_{\mathbb{L}} x \wedge y$ and $a \leq_{\mathbb{L}} y$. It follows $a \in D_\lambda(x \wedge y, y)$.

Case 2: $\boxed{a \nleq_{\mathbb{L}} y}$ Hence, $a \nleq_{\mathbb{L}} y \wedge z$ and $a \leq_{\mathbb{L}} z$. It follows $a \in D_\lambda(y \wedge z, z)$.

All in all, also (A2) is satisfied. Consequently, d_λ is a GQM.

(A3) Let $x, y \in L$. We suppose $d_\lambda(x, y) = \emptyset$. This is equivalent to

$$\begin{aligned} & Ay - A(x \wedge y) = \emptyset \\ \iff \quad & Ay = A(x \wedge y). \end{aligned}$$

Taking advantage of the precondition $d_\lambda(x, y) = \emptyset = d_\lambda(y, x)$, we follow that

$$Ay = A(x \wedge y) = A(y \wedge x) = Ax.$$

Hence, $y = x$, as A is join-dense.

All in all, d_λ is a GM w. r. t. $(L, \mathcal{M})$. □

Claim 4 *The map D_λ is* supermodular *w. r. t.* $(\mathbb{L}, \mathcal{M})$ [10]*, that is, for all $x, y \in L$, the following condition holds:*

(A4) $D_\lambda(x \wedge y, y) \subseteq D_\lambda(x, x \vee y)$

Proof. Let $a \in D_\lambda(x \wedge y, y)$. Since $D_\lambda(x \wedge y, y)$ equals $Ay - A(x \wedge y)$, we know that

$$a \in Ay \quad \text{and} \quad a \notin A(x \wedge y). \tag{4}$$

According to the definition of A, we obtain $a \leq y$ and $a \nleq x \wedge y$. Hence, $a \leq x \vee y$.

Suppose $a \leq x$. As $a \leq y$, it follows $a \leq x \wedge y$ which is a contradiction to (4). Therefore, $a \in A(x \vee y) - Ax = D_\lambda(x, x \vee y)$. □

3 Abstract Approach

We want to put our recent examinations from the special case into a more general setting. For that, we start with some necessary definitions [1, 3, 10].

Definition 1 *$\mathcal{M} = (M, *, \varepsilon, \leq)$ is an* **ordered monoid** *if $\mathbb{M} := (M, *, \varepsilon)$ is a monoid and $(M, \leq)$ is a poset such that $a \leq b$ implies $c * a \leq c * b$ and $a * c \leq b * c$, for all $a, b, c \in M$.*

Definition 2 *Let $\mathbb{P} = (P, \leq_{\mathbb{P}})$ be a poset and $\mathcal{M} = (M, *, \varepsilon, \leq)$ be an ordered monoid. A function*

$$\Delta\colon \leq_{\mathbb{P}} \longrightarrow M$$

is called **functorial w. r. t.** $(\mathbb{P}, \mathcal{M})$*, if*

– *for all* $p \in P: \quad \Delta(p,p) = \varepsilon$,

– *for all* $p,t,q \in P$ *with* $p \leq_{\mathbb{P}} t \leq_{\mathbb{P}} q: \quad \Delta(p,t) * \Delta(t,q) = \Delta(p,q)$.

Furthermore, Δ *is called* **weakly positive**, *if* $\varepsilon \leq \Delta(p,q)$ *for all* $(p,q) \in\ \leq_{\mathbb{P}}$.

Δ *is called* **supermodular** *w. r. t.* $(\mathbb{P},\mathcal{M})$, *if* $\Delta(p \wedge q, q) \leq \Delta(p, p \vee q)$ *holds for all* $(p,q) \in\ \leq_{\mathbb{P}}$.

Furthermore, Δ *is called* **submodular** *w. r. t.* $(\mathbb{P},\mathcal{M})$, *if* $\Delta(p \wedge q, q) \geq \Delta(p, p \vee q)$ *holds for all* $(p,q) \in\ \leq_{\mathbb{P}}$.

Definition 3 *Let* P *be a set, and* $\mathcal{M} = (M, *, \varepsilon, \leq)$ *be an ordered monoid. A function* $d\colon P \times P \longrightarrow M$ *is called* **generalized quasi-metric (GQM) w. r. t.** $(P,\mathcal{M})$, *if*

(A0) *for all* $(p,q) \in\ \leq_{\mathbb{P}}\ : \quad \varepsilon \leq d(p,q)$

(A1) *for all* $p \in P: \quad d(p,p) = \varepsilon$

(A2) *for all* $p,t,q \in P: \quad d(p,t) * d(t,q) \leq d(p,q)$

If in addition, (A3) holds, d *is a* **generalized metric (GM) w. r. t.** $(P,\mathcal{M})$:

(A3) *for all* $(p,q) \in P \times P: \quad d(p,q) = \varepsilon = d(q,p) \quad \Longrightarrow \quad p = q$

For a given $\Delta\colon \leq_{\mathbb{P}} \longrightarrow M$, does there exist a generalized quasi-metric $d\colon P \times P \longrightarrow M$ w. r. t. $(P,\mathcal{M})$ which extends Δ such that $d|_{\leq_{\mathbb{P}}} = \Delta$?

Theorem 1 *Let* $\mathbb{P} = (P, \leq_{\mathbb{P}})$ *be a lattice. If a map* $\Delta\colon \leq_{\mathbb{P}}\ \longrightarrow M$ *is weakly positive, supermodular and functorial w. r. t.* $(\mathbb{P},\mathcal{M})$, *then*

$$d\colon P \times P \longrightarrow M, (p,q) \mapsto \Delta(p \wedge q, q)$$

is a GQM w. r. t. $(P,\mathcal{M})$.

Proof. Obviously, conditions (A0) and (A1) from definition 3 hold for d.

For (A2), we have to show that $d(p,q) \leq_{\mathbb{P}} d(p,t) * d(t,q)$ holds for all $p,t,q \in P$. According to the definition of d this means

$$\Delta(p \wedge q, q) \leq_{\mathbb{P}} \Delta(p \wedge t, t) * \Delta(t \wedge q, q).$$

We will prove this inequality immediately in Claim 3 below. However, first of all, we need to show two properties in preparation for that.

Claim 1 (Interval Property)

$$t \leq_{\mathbb{P}} x \leq_{\mathbb{P}} y \leq_{\mathbb{P}} z \quad \Longrightarrow \quad \Delta(x,y) \leq \Delta(t,z).$$

Proof. Since Δ is functorial, we obtain

$$\Delta(t,z) = \Delta(t,x) * \Delta(x,y) * \Delta(y,z).$$

As Δ is weakly positive, we get

$$\Delta(x,y) = \varepsilon * \Delta(x,y) * \varepsilon \leq \Delta(t,z).$$

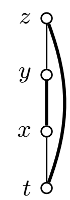


Fig. 1

◇

Claim 2 (Meet Property)

$$x \leq_{\mathbb{P}} y \quad \Longrightarrow \quad \Delta(x \wedge z, y \wedge z) \leq \Delta(x,y).$$

Proof. To show this implication, we rewrite the right hand side:

$$\Delta(x \wedge z, y \wedge z) = \Delta(x \wedge (y \wedge z), y \wedge z)$$

We continue denoting $y \wedge z$ by y' and derive

$$\begin{aligned}\Delta(x \wedge (y \wedge z), y \wedge z) &= \Delta(x \wedge y', y')\\ &\leq \Delta(x, x \vee y'),\end{aligned}$$

due to supermodularity of Δ. We know that $x \vee y' \leq_{\mathbb{P}} y$, since $x \leq_{\mathbb{P}} y$ and $y' \leq_{\mathbb{P}} y$. Hence, with Claim 1, we get

$$\Delta(x \wedge z, y \wedge z) \leq \Delta(x,y).$$

◇

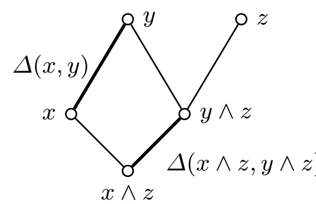


Fig. 2

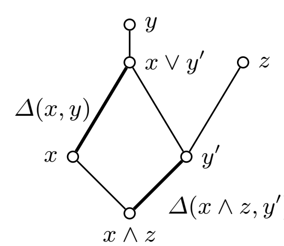


Fig. 3

Claim 3

$$\Delta(p \wedge q, q) \leq \Delta(p \wedge t, t) * \Delta(t \wedge q, q).$$

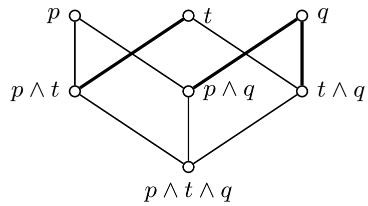


Fig. 4

Proof. Taking advantage of Claim 1, we know that

$$\Delta(p \wedge q, q) \leq \Delta(p \wedge t \wedge q, q).$$

Since Δ is functorial, it follows

$$\Delta(p \wedge t \wedge q, q) = \Delta(p \wedge t \wedge q, t \wedge q) * \Delta(t \wedge q, q).$$

With Claim 2, we finally receive

$$\Delta(p \wedge t \wedge q, t \wedge q) * \Delta(t \wedge q, q) \leq \Delta(p \wedge t, t) * \Delta(t \wedge q, q).$$

Therefore, $\Delta(p \wedge q, q) \leq \Delta(p \wedge t, t) * \Delta(t \wedge q, q)$. □

The latter theorem can be applied to various concepts of distance between objects of a given lattice. In the following, we will study some interesting applications in different context, starting with formal concept analysis.

4 Application to FCA

Let $\mathbb{K} = (G, M, I)$ be a finite formal context. Then the set of formal concepts of $\mathbb{K}$ is given by

$$B\mathbb{K} := \{(X, Y) \in 2^G \times 2^M \mid X' = Y \text{ and } , Y' = X\}$$

and the formal concept lattice of $\mathbb{K}$ is defined as

$$\mathfrak{B}\mathbb{K} := (B\mathbb{K}, \leq_{\mathfrak{B}\mathbb{K}})$$

with $c_1 \leq_{\mathfrak{B}\mathbb{K}} c_2$ iff $A_1 \subseteq A_2$ holds for all $c_1 = (A_1, B_1), c_2 = (A_2, B_2) \in B\mathbb{K}$.

Remarkably, the map

$$d_{ext} \colon B\mathbb{K} \times B\mathbb{K} \longrightarrow \mathbb{N} \quad \text{such that} \quad (c_1, c_2) \mapsto \#(A_2 - A_1)$$

is a GM w. r. t. $(B\mathbb{K}, \mathcal{M})$ with $\mathcal{M} := (\mathbb{N}, +, 0, \leq)$. The reason for this is based in Theorem 1, as we will outline below.

For all $c_1 = (A_1, B_1), c_2 = (A_2, B_2) \in B\mathbb{K}$ it follows

$$d_{ext}(c_1, c_2) = \#A_2 - \#(A_1 \cap A_2),$$

since $A_2 - A_1 = A_2 - (A_1 \cap A_2)$ and $\#$ is the counting measure.

To verify that d_{ext} is a GM, we define

$$D_{ext}\colon \leq_{\mathfrak{B}\mathbb{K}} \longrightarrow \mathbb{N} \quad \text{such that} (c_1, c_2) \mapsto \#A_2 - \#A_1.$$

Claim 4 *D_{ext} is functorial w. r. t. $(\mathfrak{B}\mathbb{K}, \mathcal{M})$, weakly positive and supermodular.*

Proof. The properties of being weakly positive and functorial are clear due to the definition of D_{ext} via the counting measure. Let us have a closer look at the supermodularity:

Let $c_1, c_2 \in B\mathbb{K}$. We have to show that

$$D_{ext}(c_1 \wedge c_2, c_2) \overset{!}{\leq} D_{ext}(c_1, c_1 \vee c_2).$$

Transforming the left hand side, we obtain

$$\begin{aligned} D_{ext}(c_1 \wedge c_2, c_2) &= D_{ext}\Big(\big(A_1 \cap A_2, (A_1 \cap A_2)'\big), \big(A_2, B_2\big)\Big) \\ &= \#A_2 - \#(A_1 \cap A_2) \\ &= d_{ext}(c_1, c_2). \end{aligned}$$

On the right hand side, we get

$$\begin{aligned} D_{ext}(c_1, c_1 \vee c_2) &= D_{ext}\Big(\big(A_1, A_2\big), \big(\underbrace{(B_1 \cap B_2)'}_{=(A_1 \cup A_2)''}, B_1 \cap B_2\big)\Big) \\ &= \#\big((A_1 \cup A_2)''\big) - \#A_1 \\ &\geq \#(A_1 \cup A_2) - \#A_1 \\ &= \#A_2 - \#(A_1 \cap A_2) \\ &= d_{ext}(c_1, c_2). \end{aligned}$$

Hence,

$$D_{ext}(c_1 \wedge c_2, c_2) \leq D_{ext}(c_1, c_1 \vee c_2)$$

and the supermodularity is shown. □

Obviously, by theorem 1 together with claim 4 it immediately follows that d_{ext} is a GM w. r. t. $(B\mathbb{K}, \mathcal{M})$.

Remark. In analogy to the above, the map

$$d_{int}\colon B\mathbb{K} \times B\mathbb{K} \longrightarrow \mathbb{N} \quad \text{such that} \quad (c_1, c_2) \mapsto \#(B_1 - B_2)$$

is a GM w. r. t. $(B\mathbb{K}, \mathcal{M})$.

5 Application to Dempster-Shafer-Theory

Choosing $L := 2^U$ and $A := L$, where U is a finite set, allows us a link to *Dempster-Shafer-Theory.*

Let m be a *mass function on* L, that is

$$m\colon L \longrightarrow \mathbb{R}_{\geq 0} : X \mapsto mX \quad \text{is a map such that } m(\emptyset) = 0 \text{ and } \sum_{X \in L} mX = 1.$$

We define

$$\mathrm{Bel}_m\colon L \longrightarrow \mathbb{R}_{\geq 0} : X \mapsto \sum_{T \subseteq X} mT$$

as the so-called *belief map* w. r. t. m and

$$\mathrm{Pl}_m\colon L \longrightarrow \mathbb{R}_{\geq 0} : X \mapsto \sum_{T \in L : T \cap X \neq \emptyset} mT$$

as the so-called *plausibility map* w. r. t. m.

Obviously, for all $X \in \mathrm{Bel}_m$, the equation $\mathrm{Bel}_m X + \mathrm{Pl}_m(U - X) = 1$ holds. That is:

$$\begin{aligned} 1 - \mathrm{Pl}_m(U - X) &= \mathrm{Bel}_m X \\ 1 - \mathrm{Bel}_m X &= \mathrm{Pl}_m(U - X). \end{aligned} \tag{5}$$

Claim 5 *Let Δ be the function which maps every pair (X, Y) with $X \subseteq Y \subseteq U$ to*

$$\Delta(X, Y) := \mathrm{Bel}_m Y - \mathrm{Bel}_m X. \tag{6}$$

Δ is functorial, weakly positive, and supermodular w. r. t. the power set lattice of U into the naturally ordered additive monoid of non-negative real numbers.

Applying this claim to theorem 1, we receive that

$$d\colon L \times L \longrightarrow \mathbb{R}_{\geq 0},\ (X, Y) \mapsto \Delta(X \cap Y, Y) = \mathrm{Bel}_m Y - \mathrm{Bel}_m(X \cap Y)$$

is a GM w. r. t. the power set of U into the naturally ordered additive monoid of non-negative real numbers.

Remark. With the *plausibility map* introduced above, a submodular pendant to the supermodular map Δ in (6) can be constructed via

$$\widetilde{\Delta}(X, Y) := \mathrm{Pl}_m Y - \mathrm{Pl}_m X \qquad \text{where } X \subseteq Y \subseteq U.$$

$\widetilde{\Delta}$ is indeed submodular, as the following inequation holds:

$$\mathrm{Pl}_m(X \cup Y) + \mathrm{Pl}_m(X \cap Y) \leq \mathrm{Pl}_m X + \mathrm{Pl}_m Y.$$

This can be shown by using the equality

$$\mathrm{Pl}_m X = 1 - \mathrm{Bel}_m(U - X)$$

that we have already observed in (5).

Remark. Dually to theorem 1, $\widetilde{\Delta}$ induces a GM $\tilde{d}$ via

$$\tilde{d}\colon L \times L \longrightarrow \mathbb{R}_{\geq 0},\ (X, Y) \mapsto \widetilde{\Delta}(X, X \cup Y) = \mathrm{Pl}_m(X \cup Y) - \mathrm{Pl}_m X.$$

6 A Fundamental Construction of Generalized Quasi Metrics

Let $\mathbb{P} = (P, \leq_{\mathbb{P}})$ be a poset, $\mathbb{M} = (M, *, \varepsilon)$ be a monoid, and $*\colon P \times M \longrightarrow P$ be a map such that the following properties are satisfied:

① For all $p \in P$ and all $x, y \in M$: $\quad p * (x * y) = (p * x) * x$

② For all $p \in P$: $\quad p * \varepsilon = p$

③ For all $p, y \in P, x \in M$: $\quad p \leq_{\mathbb{P}} q \implies p * x \leq_{\mathbb{P}} q * x$

Then we call the triple $(\mathbb{P}, \mathbb{M}, *)$ a *poset right monoid action.*

In this setup, we consider the map $\nabla\colon P \times P \longrightarrow M$ defined by

$$\nabla(p, q) \coloneqq \{x \in M \mid q \leq_{\mathbb{P}} p * x\} \text{ for all } p, q \in P.$$

Claim 6 *For all $p, q, r \in P$ the following reverse triangle inequality holds:*

$$\nabla(p, q) * \nabla(q, r) \subseteq \nabla(p, r)$$

Proof. We choose $z \in \nabla(p, q) * \nabla(q, r)$. That is, there exits $x \in \nabla(p, y)$ and $y \in \nabla(q, r)$ such that $z = x * y$.

Since $x \in \nabla(p, q)$, we know that $q \leq_{\mathbb{P}} p * x$. Analogously, $y \in \nabla(q, r)$ implies $r \leq_{\mathbb{P}} q * y$.

Using property ③, we get

$$\begin{aligned} r \leq_{\mathbb{P}} q * y &\overset{③}{\leq_{\mathbb{P}}} (p * x) * y \\ &\overset{①}{=} p * (x * y) \\ &= p * z. \end{aligned}$$

All in all, $r \leq_{\mathbb{P}} p * z$ which implies $z \in \nabla(p, r)$. □

Let $(L, *, \epsilon, \leq)$ be a *residual complete lattice*, that is an ordered monoid for which $*$ preserves arbitrary infima in each component. Furthermore, we consider a map $\nu\colon M \longrightarrow L$ which satisfies the condition

$$\nu(x * y) \leq \nu x * \nu y \quad \text{for all } x, y \in M. \tag{7}$$

On this basis, we construct the following map

$$d\colon P \times P \longrightarrow L \quad \text{via} \quad (p, q) \mapsto \inf \nu\big(\nabla(p, q)\big).$$

Claim 7 *The map d satisfies the triangle inequality, that is*

$$d(p, r) \leq d(p, q) * d(q, r) \quad \textit{holds for all } p, q, r \in P.$$

Proof. Let $p, q, r \in P$. First, we transform the inequality's right hand side:

$$\begin{aligned} d(p, q) * d(q, r) &= \inf \nu\big(\nabla(p, q)\big) * \inf \nu\big(\nabla(q, r)\big) \\ &= \inf\Big(\nu\big(\nabla(p, q)\big) * \nu\big(\nabla(q, r)\big)\Big). \end{aligned}$$

Hence, we have to show that $\inf \nu\big(\nabla(p, q)\big) \leq \inf\Big(\nu\big(\nabla(p, q)\big) * \nu\big(\nabla(q, r)\big)\Big)$.

Let $t \in \nu\big(\nabla(p, q)\big) * \nu\big(\nabla(q, r)\big)$. It follows that there exists $x \in \nabla(p, q)$ and $y \in \nabla(q, r)$ such that $t = \nu x * \nu y$, which is greater than or equal to $\nu(x * y)$ due to property, i. e. we obtain

$$\nu(x * y) \leq t. \tag{8}$$

Consequently, with claim 6, we get

$$x * y \in \nabla(p, q) * \nabla(q, r) \subseteq \nabla(p, r).$$

Applying ν on both sides yields

$$\nu(x * y) \in \nu\big(\nabla(p, r)\big).$$

Hence,

$$\inf \nu\big(\nabla(p, r)\big) \leq \nu(x * y) \overset{(8)}{\leq} t$$

and the triangle inequality of d is shown. □

Definition 4 *Let $\mathbb{M} = (M, *, \varepsilon)$ be a monoid and let $\mathcal{L} = (L, *, \epsilon, \leq)$ be an ordered monoid. Then, a map $\nu\colon M \longrightarrow L$ is a* **monoid norm w. r. t.** $(\mathbb{M}, \mathcal{L})$ *if $\nu(\varepsilon) = \epsilon$ and $\nu(x * y) \leq \nu x * \nu y$ holds for all $x, y \in M$.*

Theorem 2 *Let $(\mathbb{P}, \mathbb{M}, *)$ be a poset right monoid action with $\mathbb{P} = (P, \leq_{\mathbb{P}})$ and $\mathbb{M} = (M, *, \varepsilon)$. Further, let $\mathcal{L} = (L, *, \epsilon, \leq)$ be a residual complete lattice and $\nu\colon M \longrightarrow L$ be a monoid norm w. r. t. $(\mathbb{M}, \mathcal{L})$.*

Then

$$d\colon P \times P \longrightarrow L, (p, q) \mapsto \inf \nu\big(\nabla(p, q)\big)$$

is a GQM w. r. t. $(P, \mathcal{L})$.

7 Application to join geometries

Let $\mathbb{P} = (P, \leq)$ be a complete lattice. Then an element $x \in P$ is called *compact in* $\mathbb{P}$ if for every subset T of P with $x \leq \sup T$ there exists a finite subset U of T such that $x \leq \sup U$.

Definition 5 *A* **join geometry** *is defined as a pair* $(\mathbb{P}, E)$ *consisting of a complete lattice* $\mathbb{P}$ *and a join-dense subset* E *consisting of compact elements in* $\mathbb{P}$.

For the following, let $(\mathbb{P}, E)$ be a join geometry such that for all $p, q \in P$ there exists a compact element $r \in P$ such that $q \leq p \vee r$.

Then the triple $(\mathbb{P}, \mathbb{M}, *)$ is a *poset right monoid action* for $\mathbb{M} = (M, \cup, \emptyset)$ with $M := 2^E_{fin}$ and

$$*\colon P \times M \longrightarrow P, \quad (x, D) \mapsto x \vee \sup D.$$

Moreover, the map

$$\nu\colon M \longrightarrow \mathbb{N} \cup \{\infty\}, \quad D \mapsto \#D$$

is a monoid norm w. r. t. $(\mathbb{M}, \mathcal{L})$ for

$$\mathcal{L} := \big(\mathbb{N} \cup \{\infty\}, +, 0, \leq\big)$$

(which forms a residual complete lattice). Obviously, by theorem 2 it follows that

$$\begin{aligned} d\colon P \times P \longrightarrow \mathbb{N} \cup \{\infty\}, \quad (p, q) &\mapsto \inf \nu\big(\nabla(p, q)\big) \\ &= \min\{\#D \mid D \in 2^E_{fin} : q \leq p \vee \sup D\} \end{aligned}$$

is a GM w. r. t. $(P, \mathcal{L})$.

This result has an important specialisation for closure operators on power sets of finite sets.

8 Application to Closure Operators

Let U be a finite set and γ be a closure operator on $\mathbb{P} := \big(P, \subseteq\big)$ with $P := 2^U$. Further, let $\mathcal{M} := \big(\mathbb{N}, +, 0, \leq\big)$. Then the map

$$d\colon P \times P \longrightarrow \mathbb{N}, (X, Y) \mapsto \min\big\{\#T \mid T \in P : Y \subseteq \gamma(X \cup T)\big\}$$

is a GQM w. r. t. $(P, \mathcal{M})$, which we want to call the *closure distance*.

In particular, the restriction of d onto $\gamma P \times \gamma P$ is a GM w. r. t. $(\gamma P, \mathcal{M})$.

In context of information pooling, for a group of received elements, we can construct the corresponding closure and with the *closure distance* d from above, the distance to a given closure can be evaluated. This works for arbitrary closure operators, which also includes closure systems of a *matroid*, for instance.

References

[1] Birkhoff, G.: *Lattice Theory.* American Mathematical Society Colloquium Publications, Vol. 25, Providence, 1967

[2] Blyth, T.S., Janowitz, M.F.: *Residuation Theory.* Pergamon Press, pp. 1-382, 1972

[3] Carpineto, C., Romano, G.: *Concept Data Analysis: Theory and Applications.* John Wiley & Sons, 2004

[4] Davey, B. A., Priestly, H. A.: *Introduction to Lattices and Order* Cambridge University Press, 1990

[5] Deza, M., Deza, E.: *Encyclopedia of Distances.* Springer, Berlin–Heidelberg, 2009

[6] Deza, M., Grishukhin, V. P., Deza, E.: *Cones of weighted quasimetrics, weighted quasihypermetrics and of oriented cuts.* Abstracts of the International Conference "Mathematics of Distances and Applications", July 02-05, 2012, Varna, Bulgaria. ITHEA, Sofia, Bulgaria, 2012

[7] Ganter, B., Wille, R.: *Formal Concept Analysis: Mathematical Foundations.* Springer-Verlag, Berlin Heidelberg, 1999

[8] Kwuida, L., Schmidt, S.E.: *Valuations and closure operators on finite lattices.* Discrete Applied Mathematics. Vol 159, Issue 10., 2011

[9] Lengnink, K.: *Formalisierungen von Ähnlichkeit aus Sicht der Formalen Begriffsanalyse.* Shaker Verlag, Aachen, 1996

[10] Lenk, Markus: *Generalized Metrics and Preordered Sets. Master Thesis.* Technische Universität Dresden, 2015

Global Optimization in Learning with Important Data: an FCA-Based Approach

Yury Kashnitsky, Sergei O. Kuznetsov

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

Abstract. Nowadays decision tree learning is one of the most popular classification and regression techniques. Though decision trees are not accurate on their own, they make very good base learners for advanced tree-based methods such as random forests and gradient boosted trees. However, applying ensembles of trees deteriorates interpretability of the final model. Another problem is that decision tree learning can be seen as a greedy search for a good classification hypothesis in terms of some information-based criterion such as Gini impurity or information gain. But in case of small data sets the global search might be possible.
In this paper, we propose an FCA-based lazy classification technique where each test instance is classified with a set of the best (in terms of some information-based criterion) rules. In a set of benchmarking experiments, the proposed strategy is compared with decision tree and nearest neighbor learning.

Keywords: Formal Concept Analysis, lazy learning, global optimization

1 Introduction

The classification task in machine learning aims to use some historical data (a training set) to predict unknown discrete variables in unknown data (a test set). While there are dozens of popular methods for solving the classification problem, usually there is an accuracy-interpretability trade-off when choosing a method for a particular task. Neural networks, random forests and ensemble techniques (boosting, bagging, stacking etc.) are known to outperform simple methods in difficult tasks. Kaggle competitions also bear testimony for that – usually, winners resort to ensemble techniques, mainly to gradient boosting [1]. The mentioned algorithms are widely spread in those application scenarios where classification performance is the main objective. In Optical Character Recognition, voice recognition, information retrieval and many other tasks typically we are satisfied with a trained model if it has a low generalization error.

However, in lots of applications we need a model to be interpretable as well as accurate. Some classification rules, built from data and examined by experts, may be justified or proved. In medical diagnostics, when making highly responsible decisions, e.g., predicting whether a patient has cancer (i.e., dealing with "important data"), experts prefer to extract readable rules from a machine learning model in order to "understand" it and justify the decision. In credit scoring,

© Marianne Huchard, Sergei O. Kuznetsov (Eds.): CLA 2016, pp. 189–200,
ISBN 978-5-600-01454-1, National Research University Higher School of Economics, 2016.

for instance, applying ensemble techniques can be very effective, but the model is often obliged to have "sound business logic", that is, to be interpretable [2].

2 Related work

Eager (non-lazy) algorithms construct classifiers that contain an explicit hypothesis mapping unlabelled test instances to their predicted labels. A decision tree classifier, for example, uses a stored model to classify instances by tracing the instance through the tests at the interior nodes until a leaf containing the label is reached. In eager algorithms, the main work is done at the phase of building a classifier.

In lazy classification paradigm [3], however, no explicit model is constructed, and the inductive process is done by a classifier which maps each test instance to a label using a training set.

2.1 Lazy decision trees

The authors of [4] point the following problem with decision tree learning: while entropy measures used in C4.5 and ID3 are guaranteed to decrease on average, the entropy of a specific child may not change or may increase. In other words, a single decision tree may find a locally optimal hypothesis in terms of entropy measure such as Gini impurity or pairwise mutual information. But using a single tree may lead to many irrelevant splits for a given test instance. A decision tree built for each test instance individually can avoid splits on attributes that are irrelevant for the specific instance. Thus, such "customized" decision trees (actually classification paths) built for a specific test instance may be much shorter and hence may provide a short explanation for the classification.

2.2 Lazy associative classification

Associative classifiers build a classifier using association rules mined from training data. Such rules have the class attribute as a conclusion. This approach was shown to yield improved accuracy over decision trees as they perform a global search for rules satisfying some quality constraints [5]. Decision trees, on the contrary, perform greedy search for rules by selecting the most promising attributes.

Unfortunately, associative classifiers tend to output too many rules while many of them even might not be used for classification of a test instance. Lazy associative classification algorithm overcomes these problems of associative classifiers by generating only the rules with premises being subsets of test instance attributes [5]. Thus, in lazy associative classification paradigm only those rules are generated that might be used in classification of a test instance. This leads to a reduced set of classification rules for each test instance.

2.3 Decision trees in terms of Formal Concept Analysis

In [6] the authors utilize concept lattices to represent each concept intent (a closed set of attributes) as a decision tree node and a concept lattice itself – as a set of overlapping decision trees. The construction of a decision tree is thus reduced to selecting one of the downward paths in a concept lattice via some information criterion.

2.4 Lazy classification for complex structure data

The modification of the lazy classification algorithm capable of handling complex structure data was first proposed in [7]. The main difference from the Lazy Associative Classification algorithm is that the method is designed to analyze arbitrary objects with complex descriptions (intervals, sequences, graphs etc.). This setting was implemented for interval credit scoring data [8] and for graphs in a toxicology prediction task [9].

3 Definitions

Here we introduce some notions from Formal Concept Analysis [10] which help us to organize the search space for classification hypotheses.

Definition 1. *A formal context in FCA is a triple $K = (G, M, I)$ where G is a set of objects, M is a set of attributes, and the binary relation $I \subseteq G \times M$ shows which object possesses which attribute. gIm denotes that object g has attribute m. For subsets of objects and attributes $A \subseteq G$ and $B \subseteq M$ Galois operators are defined as follows:*

$$A' = \{m \in M \mid gIm \ \forall g \in A\},$$
$$B' = \{g \in G \mid gIm \ \forall m \in B\}.$$

A pair (A, B) such that $A \subseteq G, B \subseteq M, A' = B$ and $B' = A$, is called a formal concept of a context K. The sets A and B are closed and called the extent and the intent of a formal concept (A, B) respectively.

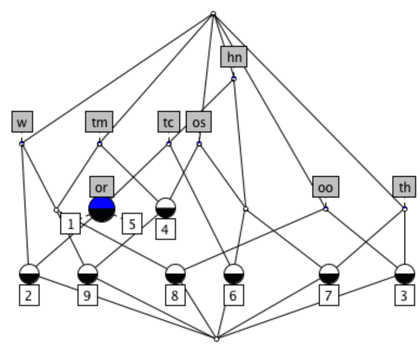
Example 1. Let us consider a "classical" toy example of a classification task. The training set is represented in Table 1. All categorical attributes are binarized into "dummy" attributes. The table shows a formal context $K = (G, M, I)$ with $G = \{1, \ldots, 10\}$, $M = \{or, oo, os, tc, tm, th, hn, w\}$ (let us omit a class attribute "play") and I – a binary relation defined on $G \times M$ where an element of a relation is represented with a cross ($\times$) in a corresponding cell of a table.

A concept lattice for this formal context is depicted to the right from1. It should be read as follows: for a given element (formal concept) of the lattice its intent (closed set of attributes) is given by all attributes which labels can be reached in ascending lattice traversal. Similarly, the extent (a closed set of objects) of a certain lattice element (formal concept) can be traced in a downward lattice traversal from a given point. For instance, a big blue-and-black circle depicts a formal concept $(\{1, 2, 5\}, \{or, tc, hn\})$.

Such concept lattice is a concise way of representing all closed itemsets (formal concepts' intents) of a formal context. Closed itemsets, further, can serve as a condensed representation of classification rules [11]. In what follows, we develop the idea of a hypotheses search space represented with a concept lattice.

Table 1. A toy classification problem and a concept lattice of the corresponding formal context. Attributes: *or* – outlook = rainy, *oo* – outlook = overcast, *os* – outlook = sunny, *tc* – temperature = cold, *tm* – temperature = mild, *th* – temperature = high, *hn* – humidity = normal, *w* – windy, *play* – whether to play tennis or not (class attribute).

№	or	oo	os	tc	tm	th	hn	w	play
1	×			×			×		×
2	×			×			×	×	
3		×				×			×
4			×		×				
5	×			×			×		×
6			×	×			×		×
7			×			×	×		×
8		×			×			×	×
9			×		×			×	
10			×	×					?



4 Concept lattice a hypothesis search space

Further we describe and illustrate the proposed approach in binary- and numeric-attribute cases when dealing with binary classification. The approach is naturally extended to multiclass case with the corresponding adjustments to information criteria formulas.

4.1 Binary-attribute case

In case of training and test data represented as binary tables, the proposed algorithm is described as Algorithm 1.

Let $K_{train} = (G_{train}, M_0 \cup \overline{M}_0 \cup c_{train}, I_{train})$ and $K_{test} = (G_{train}, M_0 \cup \overline{M}_0, I_{test})$ be formal contexts representing a training set and a test set correspondingly. We state clearly that the set of attributes is dichotomized:
$M = M_0 \cup \overline{M}_0$ where $\forall g \in G_{train}, m \in M_0\ \exists\ \overline{m} \in \overline{M}_0 : gI_{train}m \rightarrow \neg gI_{train}\overline{m}$.
Let $CbO(K, min\_supp)$ be the algorithm used to find all formal concepts of a

formal context K with support greater or equal to min_supp (by default we use a modification of the InClose-2 program implementation [12] of the CloseByOne algorithm [13]). Let $inf : M \cup c_{train} \rightarrow \mathbb{R}$ be an information criterion used to rate classification rules (we use Gini impurity by default). Finally, let min_supp and n_rules be the parameters of the algorithm (the minimal support of each classification rule's premise and the number of rules to be used for prediction of each test instance's class attribute).

With these designations, the main steps of the proposed algorithm for each test instance are the following:

1. For each test object we leave only its attributes in the training set (step 1 in Algorithm 1). Or, formally, we build a new formal context $K_t = \{G_{train}, g_t', I_{train}\}$ with the same objects G_{train} as in the training context K_{train} and with attributes of a test object $g_t' \cup c_{train}$. We clarify what it means in case of real-valued attributes in subsection 4.2.
2. With $CbO(K, min\_supp)$, find all formal concepts of a formal context K_t satisfying the constraint on minimal support. We build formal concepts in a top-down manner (increasing the number of attributes) and backtrack when the support of a formal concept intent is less than min_supp. The parameter min_supp refines the support of any possible hypothesis mined to classify the test object and is therefore analogous to the parameter $min\_samples\_leaf$ of a decision tree. While generating formal concepts, we keep track of the values of the class attributes for all training objects having all corresponding attributes (i.e. for all objects in formal concept extent). We calculate the value of an information criterion inf (we use Gini impurity by default) for each formal concept intent.
3. Then the mined formal concepts are sorted by the value of the criterion inf from the "best" to the "worse".
4. Retaining first n concepts with the best values of the chosen information criterion, we have a set of rules to classify the current test object. For each concept we define a classification rule with concept intent as an antecedent and the most common value of class attribute among the objects of concept extent as a consequent.
5. Finally, we predict the value of the class attribute for current test object simply via majority rule among n "best" classification rules' antecedents. We also save the rules for each test object in a dictionary r_{test}.

4.2 Numeric-attribute case

In our approach, we deal with numeric attributes similarly to what is done in the CART algorithm [14]. We sort the values of a numeric attribute and identify the thresholds to binarize numeric attributes where the target attribute changes. Let us demonstrate step 1 of Algorithm 1 in case of binary and numeric attributes with a sample from Kaggle "Titanic: Machine Learning from Disaster" competition dataset.[1]

[1] https://www.kaggle.com/c/titanic

Algorithm 1 Lazy Lattice-based Optimization (LLO)

Input: $K_{train} = (G_{train}, M_0 \cup \overline{M}_0 \cup c_{train}, I_{train})$
$K_{test} = (G_{test}, M_0 \cup \overline{M}_0, I_{test})$
$min\_supp \in \mathbb{R}^+,\ n_{rules} \in \mathbb{N};$
$CbO(K, min\_supp) : K \to \mathcal{S};$
$sort(\mathcal{S},\ inf) : \mathcal{S} \to \mathcal{S}$
$inf : M \cup c_{train} \to \mathbb{R};$
Output: c_{test}, r_{test}

$c_{test} = \emptyset, r_{test} = \emptyset$
for $g_t \in G_{test}$ **do**
1. $K_t = \{G_{train}, g_t', I_{train}\}$
2. $\mathcal{S}_t = \{(A, B) \mid A \subseteq G_{train}, B \subseteq g_t', A' = B, B' = A, \frac{|A|}{G_{train}} \geq min\_supp\} = CbO(K_t, min\_supp)$
3. $\mathcal{S}_t = sort(\mathcal{S}_t,\ inf)$
4. $\{B_i\}_{i \in [1, n_{rules}]} = \{B_j \mid (A_j, B_j) \in \mathcal{S}_t\}, j \in [1, n\_rules]$
5. $c_i = argmax(\{count(c_{train_j}) \mid j \in B_i'\})$
6. $r_{test}[i] = \{B_i \to c_i\}, i = 1, \ldots, n_{rules}$
7. $c_{test}[i] = argmax(\{count(c_j) \mid j = 1, \ldots, n_{rules})\}$

end for

Example 2. Table 2 shows a sample from the Titanic dataset. Let us build a formal context to classify passenger no. 7 with attributes *Pclass=2, Age=28, City=C*. If we sort the data by age in ascending order we see where the target attribute "Survived" switches from 0 to 1 or vice versa.

Age	16	18	30	39	42	62
Survived	1	0	0	1	0	1

Thus we have a set of thresholds to discretize the attribute "Age": $T = \{17, 34.5, 40.5, 52\}$. The formal context K_7 (corresponding to K_t for $t = 7$ in Algorithm 1) is presented in Table 3.

4.3 Complexity

The algorithm is based on the CloseByOne lattice-building algorithm with time complexity shown [15] to be equal to $O(|G||M|^2|\mathcal{L}|)$ for a formal context (G, M, I) and a corresponding lattice $\mathcal{L}$. To put it simply, the complexity is linear in the number of objects, quadratic in the number of attributes and linear in the number of built formal concepts.

In the proposed algorithm CloseByOne is run for each test object (step 3 in Algorithm 1), and for each formal concept information criterion values are calculated. Calculating entropy or Gini index is linear in the number of objects as it requires calculating supports of attribute sets. This is done "on-the-go" while building a lattice (step 4 in Algorithm 1).

Table 2. A sample from the Titanic dataset. Attributes: “Pclass” – passenger’s class, “City” – boarding city (here Cherburg or Southhampton), “Age” – passenger’s age, “Survived” – whether a passenger survived in the Titanic disaster.

Id	**Pclass**	**Age**	**City**	**Survived**
1	3	39	S	1
2	3	16	S	1
3	1	62	C	1
4	3	42	S	0
5	2	30	C	0
6	2	18	C	0
7	2	28	C	?
8	1	47	C	?

Table 3. A formal context built to classify a test passenger no. 7.

Id	$Pclass! = 1$	$Pclass == 2$	$Pclass! = 3$	$Age \geq 17$	$Age \leq 34.5$	$Age \leq 40.5$	$Age \leq 52$	$City == C$	**Survived**
1	×			×		×	×		1
2	×				×	×	×		1
3			×	×				×	1
4	×			×			×		0
5	×	×	×	×	×	×	×	×	0
6	×	×	×	×	×	×	×	×	0

Therefore, the time complexity of classifying $|G_t|$ test instances with the proposed algorithm based on a training formal context (G, M, I) is approximately $O(|G_t||G||M|^2|\bar{\mathcal{L}}|)$ where $|\bar{\mathcal{L}}|$ is an average lattice size for formal contexts described in step 2 in Algorithm 1.

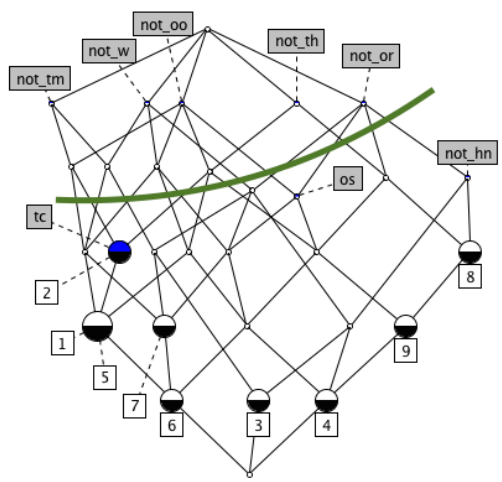
5 Example

Let us illustrate the proposed algorithm with a toy example from Table 1. To classify the object no. 10, we do the following steps according to Algorithm 1:

1. Let us fix Gini impurity as an information criterion of interest and the parameters $min\_supp = 0.5$ and $n = 3$. Thus, we are going to classify a test instance with 3 rules supporting at least 5 objects and having highest gain in Gini impurity.
2. The case *Outlook=sunny, Temperature=cool, Humidity=high, Windy=false* corresponds to a set of attributes $\{os,\ tc,\ hh, w\}$ describing the test instance. Or, if we consider the negations of the attributes, such case is described with a set of attributes: $\{\bar{or}, \bar{oo}, os, tc, \bar{tm}, \bar{th}, \bar{hn}, \bar{w}\}$
3. We build a formal context with objects being the training set instances and attributes of a test instance – $\{\bar{or}, \bar{oo}, os, tc, \bar{tm}, \bar{th}, \bar{hn}, \bar{w}\}$. The corresponding binary table is shown in Table 4.

Table 4. The training set instances with attributes of a test instance *Outlook=sunny, Temperature=cool, Humidity=high, Windy=false*. Attributes: $\bar{or}$ – outlook is not rainy, $\bar{oo}$ – outlook is not overcast, os – outlook = sunny, tc – temperature = cool, $\bar{tm}$ – temperature is not mild, $\bar{th}$ – temperature is not high, $\bar{hn}$ – humidity is not normal, $\bar{w}$ – not windy, *play* – whether to play tennis or not (class attribute). A concept lattice on the right-hand side is build with the corresponding formal context. The horizontal line separates the concepts with extents comprised of at least 5 objects.

№	$\bar{or}$	$\bar{oo}$	**os**	**tc**	$\bar{tm}$	$\bar{th}$	$\bar{hn}$	$\bar{w}$	**play**
1		×		×	×	×		×	×
2		×		×	×	×			
3	×				×		×	×	×
4	×	×	×			×	×	×	
5		×		×	×	×		×	×
6	×	×	×	×	×	×		×	×
7	×	×	×		×			×	×
8	×					×	×		×
9	×	×	×			×	×		



4. A concept lattice, organizing all formal concepts for a formal context is shown to the right from Table 4. The horizontal line separates the concepts with extents having at least 5 objects (above, $min\_supp \geq 0.5$).
5. 9 formal concepts satisfying $min\_supp \geq 0.5$ give rise to 9 classification rules. Top 3 rules having the highest gain in Gini impurity are given in Table 5.

Table 5. Top 3 rules to classify the test instance *Outlook=sunny, Temperature=cool, Humidity=high, Windy=false*

Rule	**Gini gain**
$\{not\ windy, temperature\ not\ mild\} \rightarrow play$	0.278
$\{outlook\ not\ overcast, temperature\ not\ high\} \rightarrow play$	0.111
$\{outlook\ not\ overcast, temperature\ not\ mild\} \rightarrow play$	0.044

6. The "best" rules mined in the previous step unanimously classify the test instance *Outlook=sunny, Temperature=cool, Humidity=high, Windy=false* as appropriate for playing tennis.

6 Experiments

As we have stated, in this paper we deal with "important data" problems, those where accurate and interpretable results are needed. We compare the proposed classification algorithm (denoted as LLO for "Lazy Lattice-based Optimization") with Scikit-learn [16] implementations of CART [14] and kNN on several datasets from the UCI machine learning repository.[2]

We used pairwise mutual information as a criterion for rule selection. CART and kNN parameters were chosen in stratified 5-fold cross-validation and are given in Table 7.

Parameter min_supp for LLO was taken equal to CART min_sample_leaf for each dataset divided by the number of objects. We used $n = 5$ classification rules to vote for a test instance label.

As it can be seen, the proposed approach performs better than CART on most of the datasets while kNN is often better when the number of attributes is not high. Obviously, the running times of LLO are far from perfect. That is due to the computationally demanding nature of the algorithm.

Conclusions and further work

In this paper, we have shown how searching for classification hypotheses in a formal concept lattice for each test instance individually may yield accurate

[2] http://repository.seasr.org/Datasets/UCI/csv/

Table 6. Accuracy and F1-score in classification experiments with the UCI machine learning datasets. "CART acc" stands for "5-fold cross-validation accuracy of the CART algorithm", ... , "LLO F1" stands for "5-fold cross-validation F1 score of the Lazy Lattice Optimization algorithm".

dataset	CART acc	kNN acc	LLO acc	CART F1	kNN F1	LLO F1
audiology	0.743	0.442	**0.758**	0.725	0.336	**0.736**
breast-cancer	0.738	0.727	**0.769**	0.477	0.66	**0.694**
breast-w	0.936	0.773	**0.942**	0.909	0.734	**0.921**
colic	0.647	0.644	**0.653**	0.619	0.569	**0.664**
heart-h	0.782	**0.837**	0.791	0.664	**0.831**	0.787
heart-statlog	0.804	**0.848**	0.816	0.761	**0.846**	0.823
hepatitis	**0.794**	**0.794**	0.782	**0.867**	0.702	0.755
hypothyroid	**0.975**	0.923	0.968	**0.974**	0.886	0.948
ionosphere	0.9	0.783	**0.924**	0.923	0.757	**0.938**
kr-vs-kp	**0.98**	0.761	**0.98**	0.981	0.756	**0.984**
letter	0.769	0.711	**0.774**	0.769	0.645	**0.771**
lymph	0.818	**0.831**	0.82	0.806	0.813	**0.85**
primary-tumor	0.425	**0.469**	0.457	0.376	**0.418**	0.409
segment	0.938	0.872	**0.947**	**0.938**	0.869	0.928
sonar	0.697	0.663	**0.73**	0.665	0.658	**0.718**
soybean	0.877	**0.89**	0.88	0.868	0.883	**0.879**
splice	0.943	0.833	**0.956**	0.943	0.832	**0.948**
vehicle	**0.708**	0.677	0.692	**0.708**	0.667	0.62
vote	0.956	0.929	**0.968**	0.946	0.929	**0.955**
vowel	0.436	0.405	**0.442**	**0.428**	0.387	0.406
waveform-5000	0.761	**0.834**	0.783	0.761	0.583	**0.774**

Table 7. Parameters and runtimes in classification experiments with the UCI machine learning datasets. "CART msl" stands for for the minimal required number of objects in each node of a CART tree ($min\_samples\_leaf$), "kNN k" is the number of neighbors used by the kNN algorithm.

dataset	**# objects**	**# attr**	**CART msl**	**kNN k**	**CART time**	**kNN time**	**LLO time**
audiology	226	94	1	10	0.31	0.52	9.97
breast-cancer	286	39	4	20	0.29	0.52	5.35
breast-w	699	89	6	10	0.29	0.83	258.37
colic	368	59	1	30	0.3	0.52	6.41
heart-h	294	24	2	20	0.3	0.52	0.89
heart-statlog	270	13	5	45	0.3	0.53	3.76
hepatitis	155	285	2	10	0.29	0.55	62.9
hypothyroid	3772	126	7	15	0.63	1.39	298.84
ionosphere	351	34	4	10	0.41	0.54	2.03
kr-vs-kp	3196	38	1	50	0.4	2.03	23.15
letter	20000	256	1	71	38.04	67.85	6607.2
lymph	148	50	1	10	0.29	0.52	4.7
primary-tumor	339	26	4	15	0.3	0.52	1.59
segment	2310	19	1	10	1.05	0.83	4.17
sonar	208	60	3	15	0.41	0.53	3.79
soybean	683	98	1	10	0.3	0.73	32.6
splice	3190	287	6	65	1.3	11.29	1302.57
vehicle	846	18	4	10	0.62	0.63	1.34
vote	435	32	2	10	0.31	0.53	2.65
vowel	990	26	2	35	0.63	0.63	3.29
waveform-5000	5000	40	5	82	3.79	1.34	40.2

results while keeping the classification model interpretable. The proposed strategy is computationally demanding but may be used for “small data” problems where prediction delay is not as important as classification accuracy and interpretability.

Further we plan to implement the idea of searching for classification hypotheses in a concept lattice for complex structure data such as molecular graphs We plan to implement the same strategy of lazy classification by searching for succinct classification rules in a pattern concept lattice. The designed framework might help to learn sets of rules for tasks such as biological activity (toxicology, mutagenicity, etc.) prediction. We are also going to interpret random forests as a search for an optimal hypothesis in a concept lattice and try to compete with this popular classification method.

References

1. Tsoumakas, G., Papadopoulos, A., Qian, W., Vologiannidis, S., D'yakonov, A., Puurula, A., Read, J., Svec, J., Semenov, S.: Wise 2014 challenge: Multi-label classification of print media articles to topics. In: 15th International Conference on Web Information Systems Engineering (WISE 2014). Proceedings Part II. Volume 8787 of Lecture Notes in Computer Science., Springer (October 12-14 2014) 541–548
2. Li, X., Zhong, Y.: An overview of personal credit scoring: Techniques and future work. International Journal of Intelligence Science **2**(4A) (2012) 181–189
3. Aha, D.W., ed.: Lazy Learning. Kluwer Academic Publishers, Norwell, MA, USA (1997)
4. Friedman, J.H.: Lazy decision trees. In: Proceedings of the Thirteenth National Conference on Artificial Intelligence - Volume 1. AAAI'96, AAAI Press (1996) 717–724
5. Veloso, A., Meira Jr., W., Zaki, M.J.: Lazy Associative Classification. In: Proceedings of the Sixth International Conference on Data Mining. ICDM '06, Washington, DC, USA, IEEE Computer Society (2006) 645–654
6. Belohlavek, R., De Baets, B., Outrata, J., Vychodil, V.: Inducing decision trees via concept lattices. International Journal of General Systems **38**(4) (2009) 455–467
7. Kuznetsov, S.O.: Scalable knowledge discovery in complex data with pattern structures. In Maji, P., Ghosh, A., Murty, M.N., Ghosh, K., Pal, S.K., eds.: PReMI. Volume 8251 of Lecture Notes in Computer Science., Springer (2013) 30–39
8. Masyutin, A., Kashnitsky, Y., Kuznetsov, S.O.: Lazy classification with interval pattern structures: Application to credit scoring. In: CEUR Workshop Proceedings. Volume 1430. (2015) 43–54
9. Kashnitsky, Y., Kuznetsov, S.O.: Lazy associative graph classification. In: CEUR Workshop Proceedings. Volume 1430. (2015) 63–74
10. Ganter, B., Wille, R.: Formal Concept Analysis: Mathematical Foundations. 1st edn. Springer-Verlag New York, Inc., Secaucus, NJ, USA (1997)
11. Hata, I., Veloso, A., Ziviani, N.: Learning accurate and interpretable classifiers using optimal multi-criteria rules. JIDM **4**(3) (2013) 204–219
12. Andrews, S.: In-close2, a high performance formal concept miner. In: Conceptual Structures for Discovering Knowledge - 19th International Conference on Conceptual Structures, ICCS 2011, Derby, UK. Proceedings. (2011) 50–62

13. Kuznetsov, S.O.: A fast algorithm for computing all intersections of objects from an arbitrary semilattice. Nauchno-Tekhnicheskaya Informatsiya Seriya 2 – Informatsionnye protsessy i sistemy (1) (1993) 17–20
14. Breiman, L., Friedman, J., Stone, C., Olshen, R.: Classification and Regression Trees. The Wadsworth and Brooks-Cole statistics-probability series. Taylor & Francis (1984)
15. Kuznetsov, S.O., Obiedkov, S.A.: Comparing performance of algorithms for generating concept lattices. Journal of Experimental & Theoretical Artificial Intelligence **14**(2-3) (2002) 189–216
16. Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., Duchesnay, E.: Scikit-learn: Machine Learning in Python. Journal of Machine Learning Research **12** (2011) 2825–2830

Intension Graphs as Patterns over Power Context Families

Jens Kötters

data2knowledge GmbH, Bremen, Germany

Abstract. Intension graphs are introduced as an intensional variant of Wille's concept graphs. Windowed intension graphs are then introduced as formalizations of conjunctive queries. Realizations describe pattern matching over power context families, which have been introduced with concept graphs as representations of relational data using a sequence of formal contexts. Using windowed intension graphs as patterns within pattern structures, we can define concept lattices, where power context families take the role of formal contexts. Relational Context Families, used in Relational Concept Analysis (RCA), correspond to power context families using sorts and only binary relations, and the lattices generated by the RCA algorithm (using wide scaling) can be represented using rooted trees as intents, which are introduced as a subclass of windowed intension graphs. Consequently, projections of the previously introduced pattern structure can be used as an alternative to the RCA algorithm.

Keywords: Conjunctive Queries, Pattern Structures, Power Context Families, Relational Concept Analysis

1 Introduction

In the terminology of philosophers and linguists, a concept has an *extension* and an *intension*. We may say that "extension" refers to the things belonging to a concept, whereas "intension" refers to the meaning of a concept. Formal Concept Analysis [6] (FCA) provides a mathematical formalization of concepts which represents the extension by a set of formal objects (the *extent*) and the intension by a set of formal attributes (the *intent*). The notion of intension is however a vague one and different representations can be thought of.

Many real-world concepts describe objects in terms of their relations to other objects (e.g. visitor, grandfather, ticket), and this may suggest a different representation of intensions using graphs. It turns out that conjunctive queries offer a rich notational framework to support this kind of representation. In [8], concepts have been defined in terms of family relations, and windowed relational structures have been used to represent conjunctive queries. The qualifier "windowed" is used here to express that a number of designated elements have been chosen from the underlying structure. These are the elements being described. The current paper introduces intension graphs (IGs), which are attribute-labeled graphs, and uses them in place of relational structures. In contrast to relational

© Marianne Huchard, Sergei O. Kuznetsov (Eds.): CLA 2016, pp. 203–214, ISBN 978-5-600-01454-1, National Research University Higher School of Economics, 2016.

structures, IGs formally represent information in the same way it is drawn (i.e. centered around objects) and are supposed to be more intuitive to work with. Conjunctive queries are accordingly represented by windowed IGs. Section 2 defines IGs, describes pattern matching over power context families [12] (PCFs) and shows that IGs can be represented by PCFs and vice versa.

Sections 3 and 4 define the sum and product of IGs and windowed IGs, respectively. In both cases, sum and product realize the supremum and infimum operations. These operations are called sum and product because they realize certain universal properties (coproduct and product) defined in category theory. Also, Sect. 4 briefly states connections of windowed IGs to primitive positive formulas and relational algebra operations, which are known to exist because windowed IGs model conjunctive queries.

The concept lattice depends not only on the PCF (which plays the role of a formal context), but also depends on the chosen formalization of intension. Section 5.1 states the pattern structure (see [5]) for building the lattice of windowed IGs over a PCF, which contains the formalization of intension in its definition. From there, any general algorithm for pattern structures can be used to build the lattice. Section 5.2 provides an example and illustrations.

Finally, Sect. 6.1 shows that the Relational Concept Analysis (RCA) algorithm, used with the wide scaling operator, generates lattices of *rooted trees*, which form a subclass of conjunctive queries. It is shown that essentially the same lattices can be generated from projections [5] of the pattern structure of Sect. 5.1.

2 Intension Graphs and Power Context Families

2.1 Intension Graphs

A *simple relational graph* is a pair (V, E) consisting of a set V of vertices and a set $E \subseteq \bigcup_{k \geq 1} V^k$ of edges. The edges in $E^{(k)} := E \cap V^k$ are said to have *arity* k $(k \geq 1)$.

Definition 1. *An* intension graph *over a family* $(M_k)_{k \in \mathbb{N}}$ *of attribute sets is a triple* (V, E, κ)*, where* (V, E) *is a simple relational graph and* κ *is a map defined on* $V \cup E$ *with* $\kappa(V) \subseteq \mathcal{P}(M_0)$ *and* $\kappa(E^{(k)}) \subseteq \mathcal{P}(M_k) \setminus \{\emptyset\}$ *for* $k \geq 1$*.*

A *homomorphism* $\varphi : (V_G, E_G, \kappa_G) \to (V_H, E_H, \kappa_H)$ of intension graphs over the same family M of intents is a map $\varphi : V_G \to V_H$, extended to V^k, $k \geq 1$, by setting

$$\varphi((v_1, \ldots, v_k)) := (\varphi(v_1), \ldots, \varphi(v_k)), \tag{1}$$

which preserves edges and intents, i.e.

$$\varphi(e) \in E_H, \tag{2}$$

$$\kappa_G(u) \subseteq \kappa_H(\varphi(u)) \tag{3}$$

must hold for all $e \in E_G$ and $u \in V_G \cup E_G$. We define $\mathbf{IG}_M$ as the category of intension graphs over M.

2.2 Power Context Families

Definition 2. *A* power context family *is a sequence* $(\mathbb{K}_i)_{i\in\mathbb{N}}$ *of formal contexts* $\mathbb{K}_i =: (G_i, M_i, I_i)$ *such that* $G_i \subseteq (G_0)^i$ *for all* $i \geq 1$. *We say that* $(\mathbb{K}_i)_{i\in\mathbb{N}}$ *is a power context family over the family* $(M_i)_{i\in\mathbb{N}}$ *of attribute sets if, in addition,* $g^{I_k} \neq \emptyset$ *for all* $g \in G_k$, $k \geq 1$.

A *homomorphism* $\varphi : ((G_i, M_i, I_i))_{i\in\mathbb{N}} \to ((H_i, M_i, J_i))_{i\in\mathbb{N}}$ of power context families over the same family $(M_i)_{i\in\mathbb{N}}$ of attribute sets is a map $\varphi : G_0 \to H_0$, extended to $(G_0)^k$, $k \geq 1$, by setting

$$\varphi((g_1, \ldots, g_k)) := (\varphi(g_1), \ldots, \varphi(g_k)), \tag{4}$$

which preserves incidences, i.e.

$$gI_k m \Rightarrow \varphi(g)J_k m \tag{5}$$

must hold for all $k \in \mathbb{N}$, $g \in G_k$ and $m \in M_k$. We define $\mathbf{PCF}_M$ as the category of power context families over M.

2.3 Isofunctors

Let $(M_i)_{i\in\mathbb{N}} =: M$ be a family of attribute sets. We may represent a power context family $(\mathbb{K}_i)_{i\in I}$ in $\mathbf{PCF}_M$ by an intension graph

$$\mathrm{ig}_M((\mathbb{K})_{i\in\mathbb{N}}) := (G_0, \bigcup_{k\geq 1} G_k, \{u \mapsto u^{I_k} \mid (k, u) \in \bigcup_{k\in\mathbb{N}} \{k\} \times G_k\}). \tag{6}$$

in $\mathbf{IG}_M$. Conversely, each intension graph G in $\mathbf{IG}_M$ is represented in $\mathbf{PCF}_M$ by the power context family

$$\begin{gathered}\mathrm{pcf}_M(G) := ((E_G^{(k)}, M_k, \ni_G^{(k)}))_{k\in\mathbb{N}}, \\ \text{where } u \ni_{\mathfrak{G}}^{(k)} m :\Leftrightarrow m \in \kappa(u).\end{gathered} \tag{7}$$

It is easy to see that $\mathrm{pcf}_M(\mathrm{ig}_M(\vec{\mathbb{K}})) = \vec{\mathbb{K}}$ and $\mathrm{ig}_M(\mathrm{pcf}_M(G)) = G$ for all $\vec{\mathbb{K}} \in \mathbf{PCF}_M$ and $G \in \mathbf{IG}_M$. Moreover, every homomorphism $\varphi : G \to H$ of intension graphs is also a homomorphism $\varphi : \mathrm{pcf}_M(G) \to \mathrm{pcf}_M(H)$ and vice versa. This means that the categories $\mathbf{IG}_M$ and $\mathbf{PCF}_M$ are essentially the same.

2.4 Interpretations

Power context families can be used to model factual knowledge about objects and their relations to each other. The objects are collected in a set G_0, and the formal contexts (G_0, M_0, I_0) and (G_1, M_1, I_1) describe the objects by attributes. Finally, the contexts (G_k, M_k, I_k), $k \geq 2$, describe how the objects are related to each other.

Intension graphs can be used to model patterns. The nodes describe some unspecified objects, and the map κ describes them in terms of attributes. An edge is used to indicate that the objects involved are related in some way, and the map κ specifies the relation(s) between the objects. A pattern match is formalized by the following definition:

Definition 3. *Let* $G \in \mathbf{IG}_M$ *and* $\vec{\mathbb{K}} \in \mathbf{PCF}_M$. *A* realization $\rho : G \to \vec{\mathbb{K}}$ *is a map* $\rho : V_G \to G_0$ *with* $\rho(u) \in \kappa(u)^{I_k}$ *for all* $u \in E_G^{(k)}$ *and* $k \in \mathbb{N}$.

Since $\mathbf{IG}_M$ and $\mathbf{PCF}_M$ are isomorphic, we may represent patterns and data in the same category. A realization then becomes a homomorphism:

Proposition 1. *Let* $G \in \mathbf{IG}_M$ *and* $\vec{\mathbb{K}} =: (G_i, M_i, I_i)_{i \in I} \in \mathbf{PCF}_M$. *A map* $\varphi : V_G \to G_0$ *is a realization* $\varphi : G \to \vec{\mathbb{K}}$ *iff it is a homomorphism* $\varphi : G \to \mathrm{ig}(\vec{\mathbb{K}})$.

Some remarks are in order why intension graphs and power context families were defined the way they are. First, if edge labels of intension graphs were permitted, we could create more specific patterns by adding edges with empty labels. This could be justified by saying that an edge with an empty label means that the incident vertices are related in some unspecified way. However, it seems better to model this explicitly by adding "is related" attributes. Adding empty rows to a context (G_k, M_k, I_k), $k \geq 1$, of a power context family $\vec{\mathbb{K}}$, on the other hand, results in an equivalent power context family (as per the homomorphism definition). To make $\mathbf{PCF}_M$ and $\mathbf{IG}_M$ isomorphic, empty rows are not permitted in contexts (G_k, M_k, I_k), $k \geq 1$.

3 Graph Operations and Graph Construction

The main result of Sections 3 and 4 is the definition of the product and the sum for IGs and windowed IGs. These operations define infima and suprema in the respective morphism preorders. Moreover, in category theoretical terms, the stated operations realize (categorical) products and coproducts[1]. This means that, given graphs G_1 and G_2, there are morphisms $\pi_1 : G_1 \times G_2 \to G_1$, $\pi_2 : G_1 \times G_2 \to G_2$ such that for any other graph X and morphisms $\varphi_1 : X \to G_1$, $\varphi_2 : X \to G_2$ there is a unique $\varphi : X \to G_1 \times G_2$ with $\varphi_1 = \pi_1 \circ \varphi$ and $\varphi_2 = \pi_2 \circ \varphi$ (Fig. 1), and likewise for the coproduct (Fig. 2). Infinite (co-)products are defined accordingly. Every product is an infimum in the morphism preorder, but the opposite does not hold: products are unique up to isomorphism[1], but infima are only unique up to hom-equivalence (i.e. equivalence in the morphism preorder). The stronger product property is not needed in this paper, but when looking for infima of patterns compared by morphisms, one may check for categorical products as they can often be derived from well-known products. It may seem unfortunate that in Fig. 1 the infimum $G_1 \times G_2$ is drawn *above* G_1 and G_2, but this arrangement seems to be prevalent in drawings of categorical products and is also in line with how patterns are arranged in the concept lattice.

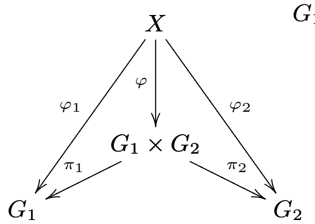


Fig. 1. Product

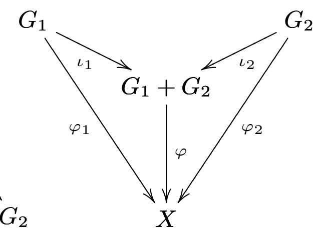


Fig. 2. Coproduct

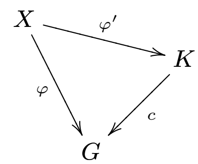


Fig. 3. Co-reflection

3.1 Graph Operations

The *product* of a family $((V_i, E_i, \kappa_i))_{i \in I}$ of intension graphs is the intension graph $\bigtimes_{i \in I}(V_i, E_i, \kappa_i) =: (V, E, \kappa)$ given by

$$V := \bigtimes_{i \in I} V_i, \tag{8}$$

$$\begin{aligned}(v_1, \ldots, v_k) \in E^{(k)} :\Leftrightarrow ((v_1(i), \ldots, v_k(i)))_{i \in I} \in \bigtimes_{i \in I} E_i^{(k)} \\ \text{and} \bigcap_{i \in I} \kappa_i((v_1(i), \ldots, v_k(i))) \neq \emptyset,\end{aligned} \tag{9}$$

$$\kappa(u) := \bigcap_{i \in I} \kappa_i(u) \tag{10}$$

for $u \in V \cup E$ and $v_1, \ldots, v_k \in V$, $k \geq 1$. Given a set X, an intension graph G and a bijection $\varphi : V_G \to X$, we call the graph

$$\varphi \circ G := (\varphi \circ V_G, \varphi \circ E_G, \kappa_G \circ \varphi^{-1}) \tag{11}$$

a *renaming* of G (cf. (1)). This amounts to a renaming of graph nodes. The *union* of graphs G and H with $V_G \cap V_H \neq \emptyset$ is the graph

$$G_1 \cup G_2 := (V_{G_1} \cup V_{G_2}, E_{G_1} \cup E_{G_2}, \kappa_{G_1} \cup \kappa_{G_2}), \tag{12}$$

and the *disjoint union* or *sum* of two arbitrary graphs G_1 and G_2 is given by

$$G_1 \sqcup G_2 := (\varphi_1 \circ G_1) \cup (\varphi_2 \circ G_2), \tag{13}$$

where $\varphi_i(v) := (i, v)$ for $i \in \{1, 2\}$ and $v \in V_{G_i}$. The disjoint union is the coproduct in $\mathbf{IG}_M$. Now let $G \in \mathbf{IG}_M$ and $\theta \subseteq V_G \times V_G$ an equivalence relation. The *quotient* of G w.r.t. θ is the graph

$$G \setminus \theta := (V_G \setminus \theta, E_G \setminus \theta, \kappa_\theta), \tag{14}$$

$$\text{where } E \setminus \theta := \{([v_1], \ldots, [v_n]) \mid (v_1, \ldots, v_n) \in E_G\}, \tag{15}$$

$$\text{and } \kappa_\theta([u]_\theta) := \bigcup_{x \theta u} \kappa(x) \tag{16}$$

for $u \in V_G$. The operation can be visualized as a merging of nodes within the same graph. For an arbitrary relation $\theta \subseteq V_G \times V_G$, we define $V \setminus \theta := V \setminus \bar{\theta}$, where $\bar{\theta}$ is the smallest equivalence relation with $\theta \subseteq \bar{\theta}$. Finally, for graphs $G, H \in \mathbf{IG}_M$ and $\theta \subseteq V_G \times V_H$, we define the *amalgam*

$$G_1 \underset{\theta}{+} G_2 := (G_1 \sqcup G_2) \setminus \{(\varphi_1(x), \varphi_2(y)) \mid (x, y) \in \theta\}, \tag{17}$$

where φ_1 and φ_2 are given as in (13). The amalgam can be visualized as a merging of two graphs by their nodes.

3.2 Graph Construction

An intension graph $G \in \mathbf{IG}_M$ with $|V_G| < \infty$ is called finite. For attribute sets $B \subseteq M_0$ and $R \subseteq M_k, k \geq 1$, we define the following structurally minimal graphs.

$$\mathcal{E}_B(x) := (\{x\}, \emptyset, \{x \mapsto B\}) \tag{18}$$

$$\begin{aligned} \mathcal{S}_R(x_1, \ldots, x_n) &:= (\{x_1, \ldots, x_n\}, \{(x_1, \ldots, x_n)\}, \kappa_R), \\ &\text{where } \kappa_R := \{(x_1, \ldots, x_n) \mapsto R, x_1 \mapsto \emptyset, \ldots, x_n \mapsto \emptyset\}) \end{aligned} \tag{19}$$

Every finite $G \in \mathbf{IG}_M$ can be constructed from these graphs in a finite number of steps, using the amalgam and renaming operations.

4 Windowed Intension Graphs

Definition 4. *A windowed intension graph is a pair (α, G) consisting of an intension graph G and a partial map $\alpha : \mathbb{N} \nrightarrow V_G$.*

A homomorphism $\varphi : (\alpha_1, G_1) \to (\alpha_2, G_2)$ of windowed intension graphs is a homomorphism $\varphi : G_1 \to G_2$ with $\varphi \circ \alpha_1 \leq \alpha_2$.

While a pattern match for an intension graph G in a power context family $\vec{\mathbb{K}}$ has been defined by a realization $\rho : G \to \vec{\mathbb{K}}$, the set

$$(\alpha, G)^\diamond := \{\alpha \circ \varphi \mid \varphi : G \to \vec{\mathbb{K}}\} \tag{20}$$

defines the set of all pattern matches for the windowed intension graph (α, G). A finite windowed intension graph corresponds to a primitive positive formula (pp formula), i.e. a predicate logical formula which is built from atoms using conjunction ($\wedge$) and existence quantification ($\exists$) only (atoms may contain the equals sign). For finite graphs, the $(\cdot)^\diamond$ operation can be inductively defined, starting with

$$(\mathrm{id}_{\{0\}}, \mathcal{E}_B(0))^\diamond = B^{I_0}, \quad B \subseteq M_0, \tag{21}$$

$$(\mathrm{id}_{\{1,\ldots,n\}}, \mathcal{S}_R(1, \ldots, n))^\diamond = R^{I_n}, \quad R \subseteq M_k, k \geq 1, \tag{22}$$

which correspond to the *select* operation on databases, and proceeding with similar rules for the *join* and *project* operations.

When viewing a windowed intension graph (α, G) as a pp formula, the set $\alpha^{-1}(V_G)$ corresponds to the free variables, and the set $V_G \setminus \alpha(\mathbb{N})$ corresponds to the existentially quantified variables.

Let us denote by C_n^S the set of all primitive positive formulas in the free variables $x_0, \dots, x_{n-1}$ for a given signature S. The lattice of all n-ary relations which can be defined in a given S-structure by formulas in C_n^S can be defined as the concept lattice of the context $((G_0)^n, C_n^S, \models)$, where $\models$ is the satisfaction relation. In Sect. 5.1, an equivalent construction is done using windowed intension graphs as patterns over a power context family.

4.1 Product and Sum

The product of a family of windowed intension graphs is given by

$$\bigtimes_{i \in I} (\alpha_i, G_i) := (\langle \vec{\alpha} \rangle, \bigtimes_{i \in I} G_i), \tag{23}$$

$$\langle \vec{\alpha} \rangle(n) := \begin{cases} (\alpha_i)_{i \in I} & \text{if } \alpha_i(n) \text{ is defined for all } i \in I, \\ \text{undefined} & \text{otherwise} \end{cases}. \tag{24}$$

The sum of windowed intension graphs is given by

$$(\alpha_1, G_1) + (\alpha_2, G_2) := ([\alpha_1, \alpha_2], G_1 \underset{\theta_{(\alpha_1, \alpha_2)}}{+} G_2), \tag{25}$$

where

$$\theta_{(\alpha_1, \alpha_2)} := \{(\alpha_1(k), \alpha_2(k)) \mid k \in \mathbb{N} \wedge \alpha_1(k) \text{ defined } \wedge \alpha_2(k) \text{ defined } \}, \tag{26}$$

$$[\alpha_1, \alpha_2](k) := \begin{cases} [(1, \alpha_1(k))]_{\theta_{(\alpha_1, \alpha_2)}} & \text{if } \alpha_1(k) \text{ is defined,} \\ [(2, \alpha_2(k))]_{\theta_{(\alpha_1, \alpha_2)}} & \text{if } \alpha_2(k) \text{ is defined, .} \\ \text{undefined} & \text{otherwise} \end{cases} \tag{27}$$

The sum can also be defined for arbitrary families $(\alpha_i, G_i)_{i \in I}$, but this is even more tedious and not needed in the following. We denote by $\mathbf{IG}_M^X$ the category of all windowed intension graphs where the first component has domain of definition X.

5 Pattern Concepts

5.1 Concept Lattices of Power Context Families

Let $\vec{\mathbb{K}} \in \mathbf{PCF}_M$ and $n \in \mathbb{N}$. We want to create the lattice which has as its extents all n-ary relations definable by windowed intension graphs (α, G), where α is defined on $n := \{0, \dots, n-1\}$ (i.e., α is an n-tuple). The most specific description for an n-tuple α is the windowed intension graph

$$\delta_{\vec{\mathbb{K}}}^n(\alpha) := (\alpha, \Delta), \tag{28}$$

where $\Delta := \mathbf{ig}(\vec{\mathbb{K}})$. We state the pattern structure as a triple $((G_0)^n, \mathbf{IG}_M^n, \delta_{\vec{\mathbb{K}}}^n)$, where the second component is a category instead of, as usual, a semilattice. As noted before, the infimum operation in the morphism preorder is realized - up to pattern equivalence - by the categorical product (Sect. 4.1).

The Galois connection which arises from the pattern structure can be stated as follows:

$$A^\diamond := \bigtimes_{\lambda \in A} (\lambda, \Delta), \tag{29}$$

$$(\alpha, G)^\diamond := \{\lambda \in (G_0)^n \mid \exists \varphi\, \varphi : (\alpha, G) \to (\lambda, \Delta)\}. \tag{30}$$

The definitions in (20) and (30) coincide. The pattern concepts are the pairs $((\alpha, G)^\diamond, (\alpha, G)^{\diamond\diamond})$ for $G \in \mathbf{IG}_M$ and $\alpha \in (G_0)^n$. The same concepts arise as the pairs $(A^{\diamond\diamond}, A^{\diamond\diamond\diamond})$ for $A \subseteq (G_0)^n$; the patterns $A^\diamond$ and $A^{\diamond\diamond\diamond}$ are hom-equivalent, but generally not identical.

5.2 Example

We define a family $M := (\{a, b\}, \emptyset, \{r, s\}, \emptyset, \dots)$ of attribute sets. Figure 4 shows a power context family $\vec{\mathbb{K}}$ over M. The intension graph $\mathbf{ig}(\vec{\mathbb{K}})$ is shown in Fig. 5. It has three components, which are individually listed in Fig. 6 as components C_1, C_2 and C_3.

Let us construct the concept lattice for patterns in $\mathbf{IG}_M^1$ (Fig. 7). First of all, a pattern in $\mathbf{IG}_M^1$ can be stated as (x, G) with $x \in V_G$ (by writing $(\alpha(0), G)$ instead of (α, G)), and we may alternatively state this as (x, C), where C is the component of x. The object intents, given by $\delta_{\vec{\mathbb{K}}}^1$, are the patterns $(1, C_1)$,$(2, C_3)$,$(3, C_3)$,$(4, C_3)$,$(5, C_2)$ and $(6, C_2)$. In Fig. 7, they can be found directly on top of the pattern $((), C_0)$ for the bottom concept, which is generated by the empty product. The product $C_1 \times C_3$ has a single component, which is denoted C_5 (Fig. 6). This yields $(1, C_1) \times (j, C_3) = ((1, j), C_5)$ for $j = 2, 3, 4$. As we can see, when we multiply intension graphs, several products of windowed intension graphs are obtained at once. In this case, all three products are hom-equivalent, and yield the topmost circle pattern in Fig.7.

Let us generate all 2-generated concepts in lectic order. The next concept would be $(\{1, 5\}^{\diamond\diamond}, \{1, 5\}^\diamond)$ up to hom-equivalence. Formally, Sect. 5.1 states the intent as $\{1, 5\}^{\diamond\diamond\diamond}$, but this is not practically relevant. The product $C_1 \times C_2$ provides $\{1, 5\}^\diamond$ and $\{1, 6\}^\diamond$, which are different patterns (the co-atoms in Fig. 7) with the same underlying component C_7. To obtain $\{2, 3\}^\diamond$, we compute $C_3 \times C_3$, which is disconnected ($C_3 \times C_3 \cong C_3 \sqcup C_4 \sqcup C_4$). The new component C_4 yields the three remaining circle patterns in Fig. 7. Computing $\{2, 5\}^\diamond$ leads to $C_3 \times C_2 = C_6 \sqcup C_8$, which gives six new pattern concepts, and finally $\{5, 6\}^\diamond$ (the only missing combination) yields the top concept. There are two more patterns, which are 3-generated and have undelying component C_{10}.

All patterns produced were minimal (i.e. they have no proper hom-equivalent subpattern). The reason for this is that the nodes in the generating patterns

$\mathbb{K}_0$:

	a	b
1		
2		×
3	×	
4	×	
5		
6	×	×

$\mathbb{K}_2$:

	r	s
(1,1)	×	
(2,3)	×	×
(3,4)	×	
(4,2)	×	
(5,6)		×
(6,5)	×	

Fig. 4. Power context family

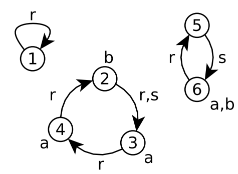


Fig. 5. Intension graph

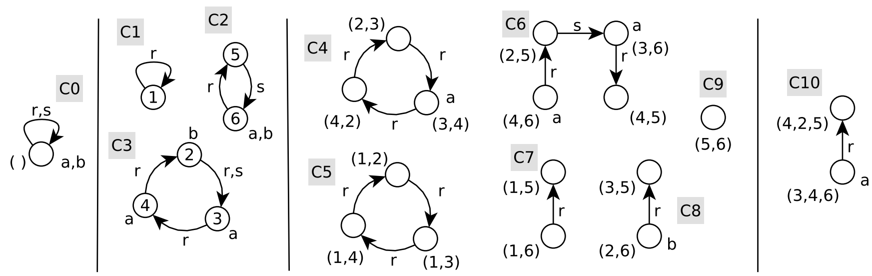


Fig. 6. Components

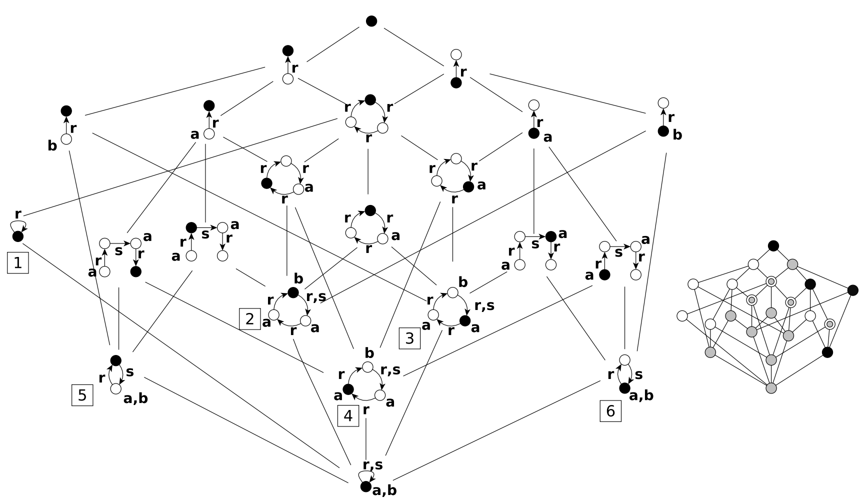


Fig. 7. Concept lattice (left), tree $\bigwedge$-sublattices (right)

have indegree and outdegree bounded by 1. Otherwise it may happen that a component contains two concepts (i.e. nodes), say X and Y, such that X is necessary to describe Y, but a minimal description of Y does not contain X, which has to be taken care of during lattice construction. Another point is that hom-equivalent patterns need to be discovered, which generally requires homomorphism checks. The inherent complexity can be avoided if patterns are restricted to trees (see Sect.6). An implementation of lattice construction, which currently uses a variant of Ganter's NextConcept algorithm [6], is available at `https://github.com/koetters/cgnav`.

6 Relational Concept Analysis and Tree Patterns

6.1 Relational Concept Analysis

Relational Concept Analysis uses relational scaling to express relations between objects by means of formal attributes. A number of scaling operators are defined, but only the wide scaling operator is covered here. The RCA algorithm builds a lattice from a Relational Context Family (RCF), which can be seen as a many-sorted PCF with binary relations only. We only deal with the one-sorted case, because the general case is implied. In this case, an RCF can be likened to a PCF with two contexts $\mathbb{K}_0$ and $\mathbb{K}_2$. The *RCA algorithm* definies an iterative procedure which incrementally adds new attributes to $\mathbb{K}_0$. The sequence of contexts can be described as follows:

$$\mathbb{K}^{(0)} := \mathbb{K}_0, \tag{31}$$

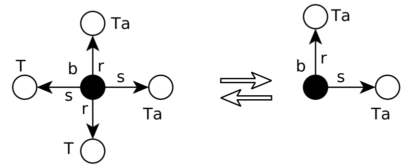
$$\mathbb{K}^{(i+1)} := \mathbb{K}_0 \mid (G_0, M_2 \times \underline{\mathcal{B}}(\mathbb{K}^{(i)}), J^{(i+1)}), \tag{32}$$

$$\text{where } gJ^{(i+1)}(r, C) :\Leftrightarrow \exists h : h \in \text{ext}(C) \wedge (g, h) \in r. \tag{33}$$

Consider the PCF from Fig. 2. To obtain $\mathbb{K}^{(1)}$, we first have to generate $\mathcal{B}(\mathbb{K}^{(0)})$. The concept lattice consists of the four black nodes shown in the miniature lattice in Fig. 7. Relational scaling produces eight new attributes. Fig.8 shows the context $\mathbb{K}^{(1)}$. The left tree in Fig. 9 represents the intent of the object 2 in $\mathbb{K}^{(1)}$ by a tree pattern. The neighbors of the black node are supposed to represent concepts of $\mathcal{B}(\mathbb{K}^{(0)})$, and the full tree pattern is obtained by substituting these with their pattern intents (this adds two occurrences of a). The right tree in Fig. 9 represents a minimal hom-equivalent subpattern. The lattice $\mathcal{B}(\mathbb{K}^{(1)})$ can be generated by intersecting all object intents (as attribute sets) or alternatively, by computing the tree products. The lattice $\mathcal{B}(\mathbb{K}^{(2)})$ consists of the gray nodes in addition to the black nodes (Fig. 7). The context $\mathbb{K}^{(3)}$ is a fixed point of the RCA algorithm, the final lattice additionally contains the dotted nodes. The white nodes are not discovered by the RCA algorithm (although five of them can be discovered by adding r^{-1} and s^{-1} to $\mathbb{K}_2$). The tree intents of the objects in $\mathbb{K}^{(i)}$ can be obtained directly from $\text{ig}_M(\vec{\kappa})$ using the splice$^{(i)}$ operation from Sect. 6.3. It is also shown that splice^i is a pattern projection(cf.[5]), which enables the use of pattern structure algorithms for RCA. The rest of the section proves the relevant claims.

	a	b	$\exists r:\top$	$\exists r:T_a$	$\exists r:T_b$	$\exists r:\bot$	$\exists s:\top$	$\exists s:T_a$	$\exists s:T_b$	$\exists s:\bot$
1			×							
2		×	×	×			×	×		
3	×		×	×						
4	×		×		×					
5							×	×	×	×
6	×	×	×							

Fig. 8. Scaled Context



$\top := ((5,6), C_9)$ $T_a := ((3,4,6), C_{10})$
$T_b := ((2,6), C_8)$ $\bot := (6, C_2)$

Fig. 9. Equivalent Object Patterns

6.2 Rooted Trees

Definition 5. *A* rooted tree *is a windowed intension graph which can be constructed by the following rules:*

(RT1) *For a given set $B \subseteq M_0$ of attributes, the windowed intension graph $(0, \mathcal{E}_B(0))$ is a rooted tree with*

$$\operatorname{depth}((0, \mathcal{E}_B(0))) := 0. \tag{34}$$

(RT2) *For a given set $B \subseteq M_0$ of attributes, an index set $I \neq \emptyset$, a family $(R_i)_{i \in I}$ of attribute sets $R_i \subseteq M_2$, $R_i \neq \emptyset$, and a family $(x_i, T_i)_{i \in I}$ of rooted trees such that $\sup_{i \in I} \operatorname{depth}((x_i, T_i)) < \infty$, the windowed intension graph*

$$(x, T) := (0, \mathcal{E}_B(0)) + \sum_{i \in I} (0, \mathcal{S}_{R_i}(0,1) \underset{\{(1,x_i)\}}{+} T_i) \tag{35}$$

is a rooted tree with

$$\operatorname{depth}((x,T)) := 1 + \max_{i \in I} \operatorname{depth}((x_i, T_i)). \tag{36}$$

A rooted tree is called thin *if it can be constructed by rules* (**RT1**) *and* (**RT2′**), *where* (**RT2′**) *is obtained from* (**RT2**) *by adding the additional requirement that $|R_i| = 1$ for all $i \in I$.*

We denote by $\mathbf{It}_{M,n}$ the subcategory of $\mathbf{IG}_M$ that consists of the thin rooted trees of depth at most n.

Proposition 2. *Let $n \in \mathbb{N}$. The concept extents of $\underline{\mathcal{B}}(\mathbb{K}^{(n)})$ are precisely the sets $(x,T)^\diamond$ described by thin rooted trees (x,T) of depth $\leq n$.*

Proof. This is proved by induction over $n \in \mathbb{N}$. For $n = 0$, the claim follows from (21). If T is a thin rooted tree with $\operatorname{depth}(T) = n+1$, there is a family $(x_i, T_i)_{i \in I}$ of thin rooted trees of depth $\leq n$ and a family $(r_i)_{i \in I}$ of attributes in M_2, such that $(x,T) = (0, \mathcal{E}_B(0)) + \sum_{i \in I} (0, \mathcal{S}_{\{r_i\}}(0,1) \underset{\{(1,x_i)\}}{+} T_i)$. By the induction hypothesis, there exists a family $(C_i)_{i\in I}$ of concepts $C_i \in \underline{\mathcal{B}}(\mathbb{K}^{(n)})$ with $(x_i, T_i)^\diamond = \text{ext}(C_i)$. Then

$$(x,T)^\diamond = ((0,\mathcal{E}_B(0)) + \sum_{i\in I}(0, \mathcal{S}_{\{r_i\}}(0,1) \underset{\{(1,x_i)\}}{+} T_i))^\diamond \tag{37}$$

$$= B' \cap \bigcap_{i\in I} r_i^{-1}((x_i,T_i)^\diamond) \tag{38}$$

$$= B' \cap \bigcap_{i\in I} r_i^{-1}(\text{ext}(C_i)) \tag{39}$$

$$= (B \cup \{(r_i, C_i) \mid i \in I\})'. \tag{40}$$

This shows that $(x, T^\diamond)$ is a concept extent in $\underline{\mathcal{B}}(\mathbb{K}^{(n+1)})$. Conversely, a concept extent in $\underline{\mathcal{B}}(\mathbb{K}^{(n+1)})$ is defined by an attribute set as in (40), and the induction hypothesis is used in (39) to obtain the family $(x_i, T_i)_{i\in I}$ for given $(C_i)_{i\in I}$. □

6.3 Graph Splicing

A graph $G \in \mathbf{IG}_M$ can be unfolded into a (possibly infinite) tree, starting at any given vertex which becomes the root of the tree. It is easy to see that, among all trees more general than G, the unfolding is the most specific one. In other words, the unfolding is a kernel operation (the dual of a closure operation). This implies that an $\bigwedge$-sublattice is obtained if patterns are restricted to trees. A similar operation constructs a thin rooted tree from a graph: In addition to unfolding, every edge carrying multiple relation attributes is spliced into several edges, each carrying exactly one of the relation attributes. The operation is formalized by the following inductive definition, where $G \in \mathbf{IG}_M$ and $x \in V_G$:

$$\text{splice}^{(0)}(x,G) := (x, \mathcal{E}_{\kappa(x)}(x)), \tag{41}$$

$$\text{splice}^{(i+1)}(x,G) := (x, \mathcal{E}_{\kappa(x)}(x)) + \sum_{(x,y)I_2 r} (x, \mathcal{S}_{\{r\}}(x,y) \underset{\{(y,\tilde{y})\}}{+} T_y^{(i)}), \quad \text{where } (\tilde{y}, T_y^{(i)}) := \text{splice}^{(i)}(y,G) \tag{42}$$

The following proposition states, in category theoretical terms, that the splice operation maps each $(x,G) \in \mathbf{IG}^1_M$ to its coreflection in $\mathbf{It_M}$ (cf. Fig. 3). As can be seem from Fig. 3, this implies that splicing is a kernel operation (or pattern projection). This means that the pattern structure $(G_0, \mathbf{It}_M, \text{splice} \circ \delta^1_{\vec{\mathbb{K}}})$ creates the concepts of the RCA algorithm.

Proposition 3. *For each $(x,G) \in \mathbf{IG}^1_M$, there exists a morphism $\varphi_{(x,G)}$: $\text{splice}((x,G)) \to (x,G)$ such that for every $(y,T) \in \mathbf{It_M}$ and $\varphi : (y,T) \to (x,G)$ there exists a unique morphism $\psi : (y,T) \to \text{splice}((x,G))$ such that $\varphi = \varphi_{(x,G)} \circ \psi$.*

Proof sketch: We inductively prove a unique morphism $\psi^{(i)} : \text{splice}^{(i)}((y,T)) \to \text{splice}((x,G))$. In the induction step, the image of each neighbor of the root node is uniquely determined. The union of the $\psi := \psi^{(i)}$ is well-defined (because of the uniqueness). Since $\operatorname{splice}((x,T)) = (x,T)$ holds, φ is the required morphism. □

From a given graph $G \in \mathbf{IG}^1_M$, we can determine for each $x \in V_G$ the extent $\operatorname{ext}_\Delta(x)$ in $\operatorname{splice}((x,G))$ without actually splicing the graph. Let us denote this as the tree extent $\operatorname{tex}_\Delta(x)$ of x in G. The tree extent can be computed as follows:

$$\operatorname{tex}^{(0)}_\Delta(x) := \kappa_G(x)^{I_k} \text{ for } x \in E^{(k)}_G, \tag{43}$$

$$\operatorname{tex}^{(i+1)}_\Delta(x) := \operatorname{tex}^{(0)}_\Delta(x) \cap \bigcap_{(x,y) I_2 r} r^{-1}(\operatorname{tex}^{(i)}_\Delta(y)). \tag{44}$$

This can be proven by inductively showing $\operatorname{splice}^{(i)}((x,G))^\diamond = \operatorname{tex}^{(i)}_\Delta(x)$.

7 Related Work

Power context families and concept graphs have been introduced by Rudolf Wille in [11]. Concept graphs have been presented as a mathematical formalization of Conceptual Graphs [10]. Different kinds of concept graphs are presented in [12] but, to the knowledge of the author, abstract concept graphs mentioned in introductory paper [11] are the only kind of concept graphs defined without a realization. Abstract concept graphs use symbols as node labels rather than sets of attributes.

In [8], windowed structures have been introduced as triples $(X, \nu, \mathcal{G})$, and a Galois connection into a complete lattice of data tables (where the infimum is realized by the *join* for database tables) has been presented. A follow-up paper [9] addresses the connection to logic, features sorts, uses a "relational structure with concept labels" hybrid and shows the connection to pattern structures by representing extensions as sets of partial interpretations. Pattern structures were introduced in [5], and the use of Conceptual Graphs as patterns is suggested in there. The representation of conjunctive queries (and thus pp formulas) by graphs, and of entailment by graph homomorphism, is credited to [2]. In [3], these relationships are stated for λ-BGs, which are Basic Conceptual Graphs with distinguished concepts given by a mapping λ, and this representation directly corresponds to the windowed abstract concept graphs (and their homomorphisms) in the paper at hand. Moreover, in [3, Chapter 8], the categorical product is used to describe the least generalization of Conceptual Graphs. The Projected Graph Patterns (PGPs) in [4], their inclusion and intersection, corresponds to λ-BGs and windowed abstract concept graphs and their respective notions of homomorphism and product. In [4], as in [8], concept lattices are generated, with intents realized using the respective formalizations.

Relational Context Families and the construction algorithm are described in [7], and the RCA algorithm has been described for different kinds of inter-object relations which are not covered here.

8 Conclusion

The paper has introduced windowed intension graphs as a formalization of conjunctive queries. Intension graphs correspond to concept graphs without the realization component. Some notation has been introduced which establishes connections to logic and database theory. The lattices generated by the RCA algorithm have been characterized as $\bigwedge$-sublattices of conjunctive queries. The results concerning rooted trees still have to be implemented and compared with the RCA algorithm. While a bound for the maximum number of iterations of the RCA algorithm can be given by $|G|$, the pattern structures algorithms might benefit from a better bound computed in advance from the context family.

References

1. Adámek, J., Herrlich, H., Strecker, G.E.: Abstract and concrete categories: the joy of cats. Pure and applied mathematics, Wiley, New York (1990)
2. Chandra, A.K., Merlin, P.M.: Optimal implementation of conjunctive queries in relational databases. In: Proceedings of the ninth annual ACM symposium on Theory of computing. pp. 77–90. STOC '77, ACM, New York, NY, USA (1977)
3. Chein, M., Mugnier, M.L.: Graph-based Knowledge Representation: Computational Foundations of Conceptual Graphs. Advanced Information and Knowledge Processing, Springer, London (2009)
4. Ferré, S.: A proposal for extending formal concept analysis to knowledge graphs. In: Baixeries, J., Sacarea, C., Ojeda-Aciego, M. (eds.) Formal Concept Analysis - 13th International Conference, ICFCA 2015, Nerja, Spain, June 23-26, 2015, Proceedings. LNAI, vol. 9113, pp. 271–286. Springer (2015)
5. Ganter, B., Kuznetsov, S.O.: Pattern structures and their projections. In: Delugach, H.S., Stumme, G. (eds.) Proceedings of ICCS 2001. LNCS, vol. 2120, pp. 129–142. Springer (2001)
6. Ganter, B., Wille, R.: Formal Concept Analysis: Mathematical Foundations. Springer, Berlin (1999)
7. Huchard, M., Hacene, M.R., Roume, C., Valtchev, P.: Relational concept discovery in structured datasets. Annals of Mathematics and Artificial Intelligence 49(1-4), 39–76 (2007)
8. Kötters, J.: Concept lattices of a relational structure. In: Pfeiffer, H.D., Ignatov, D.I., Poelmans, J., Gadiraju, N. (eds.) Proceedings of ICCS 2013. LNCS, vol. 7735, pp. 301–310. Springer (2013)
9. Kötters, J., Schmidt, H.W.: A database browser based on pattern concepts. In: Carpineto, C., Kuznetsov, S.O., Napoli, A. (eds.) Proceedings of FCAIR 2013. CEUR Workshop Proceedings, vol. 977. CEUR-WS.org (2013), `http://ceur-ws.org/Vol-977`
10. Sowa, J.F.: Conceptual Structures: Information Processing in Mind and Machine. Addison-Wesley (1984)
11. Wille, R.: Conceptual Graphs and Formal Concept Analysis. In: Lukose, D., Delugach, H.S., Keeler, M., Searle, L., Sowa, J.F. (eds.) Proceedings of ICCS 1997. LNCS, vol. 1257, pp. 290–303. Springer, Heidelberg (1997)
12. Wille, R.: Formal Concept Analysis and Contextual Logic. In: Hitzler, P., Schärfe, H. (eds.) Conceptual Structures in Practice, pp. 137–173. Chapman & Hall/CRC (2009)

Pattern Structures for Treatment Optimization in Subgroups of Patients

Natalia V. Korepanova and Sergei O. Kuznetsov

National Research University Higher School of Economics, Moscow, Russia
korepanova.natalia@gmail.com, skuznetsov@hse.ru

Abstract. A comparison of different treatment strategies does not always result in determining the best one for all patients, one needs to study subgroups of patients with significant difference in efficiency between treatment strategies. To solve this problem an approach to subgroups generation is proposed, where data are described in terms of a pattern structure and pattern concepts stay for patient subgroups and their descriptions. To find the most promising pattern concepts in terms of the difference of treatment strategies in efficiency a version of CbO algorithm is proposed. An application to the analysis of data on childhood acute lymphoblastic leukemia is considered.

Keywords: pattern structure, subgroup analysis, acute lymphoblastic leukaemia

1 Introduction

Randomized controlled trial (RCT)[1] is a common approach in evidence-based medicine to prove the superiority of a disease treatment over another one. There are three main types of hypotheses which can be tested with the help of RCT: superiority, noninferiority or equivalence. The main goal of all intervention, drug, or therapy inventions is to find a better way of treating patients. So, superiority trials are possibly the most popular ones, and they are the subject of interest of this paper. A superiority trial allows physicians to find the optimal treatment and improve the curability of the disease. However if comparing treatment strategies are similar enough it is a big success to find and prove the superiority of any of them for all patients. But the effect of treatment strategies may depend on patients initial features (physiological characteristics and/or results of diagnostics). For example, if we compare dosages of a toxic drug a small dosage may be more suitable for patients with light disease manifestation because it throws off a disease and reduces the negative consequences of toxicity while for patients with intense disease manifestation a small dosage is not enough to cure them of the disease. The question is how to find such subgroups of quite similar patients where the efficiency difference of treatment strategies is significant.

Several approaches were proposed in [2–11]. Most of them [5–11] are based on the idea of decision or regression trees which locally optimize some measure at every iteration of the algorithm. This approach is more suitable when we operate

© Marianne Huchard, Sergei O. Kuznetsov (Eds.): CLA 2016, pp. 217–228, ISBN 978-5-600-01454-1, National Research University Higher School of Economics, 2016.

on the big datasets or constantly increasing dataset because they allow to find some subgroups quickly. In the case of treatment optimization the datasets, as a rule, are not very big and do not increase in size rapidly. Moreover, collecting such datasets demands a lot of time and efforts. So, it is more important to carry out more detailed analysis of the data then to make it fast. Also, in RCT on cancer, heart conditions or chronic diseases the outcome of the therapy can be censored, while only few papers like [9, 11] report on analysis of censored data. In this paper we propose a universal approach to finding subgroups of patients with significantly different responses to different treatment strategies, which is not biased by any local optimization criterion. Within this approach subgroups of patients are generated, which are determined by subsets of patients' features.The approach in based on computing closed patterns [14–17] that satisfy criteria of treatment efficiency. The approach was proposed for the analysis of the database of randomized controlled trial on childhood acute lymphoblastic leukemia (ALL) [12, 13] which was performed in several hospitals in Russia and Byelorussia. In this dataset each patient under study is assigned one of the studied treatment strategies, he/she is described by a set of initial features that can be nominal or numerical, and some outcome which is used to estimate treatment efficiency.

The rest of the paper is organized as follows. In section 2 we recall basic definitions of pattern structures and give examples of pattern structures [14–17] relevant to the analyzed data. In section 3 a version of Close-by-One (CbO) algorithm [18] performing on the attributes is proposed. Section 4 presents stopping criterion added to the version of CbO to generate only subgroups with the difference in the efficiency of treatment strategies. Section 5 presents an application of the proposed approach to the ALL dataset, and we conclude in section 6.

2 Pattern Structures

2.1 Main Definitions

In this section we recall pattern structures and examples of pattern structures used for nominal and numerical features.

Let G be a set (of objects), $(D, \sqcap)$ be a meet-semilattice (of all possible object descriptions), and $\delta : G \to D$ be a mapping. Then $(G, (D, \sqcap), \delta)$ is called a *pattern structure*, provided that the set $\delta(G) = \{\delta(g) \mid g \in G\}$ generates a complete subsemilattice $(D_\delta, \sqcap)$ of $(D, \sqcap)$, i.e. every subset X of $\delta(G)$ has an infimum $\bigsqcap X$ in $(D, \sqcap)$. Elements of D are called *patterns* and are naturally ordered by subsumption relation $\sqsubseteq$: $c \sqsubseteq d \Leftrightarrow c \sqcap d = c$, where $c, d \in D$. Operation $\sqcap$ is also called a *similarity operation*. If $(G, (D, \sqcap), \delta)$ is a pattern structure we define the derivation operators which form a Galois connection between the powerset of G and $(D, \sqcap)$ as:

$$\begin{aligned} A^\diamond &= \textstyle\bigsqcap_{g \in A} \delta(g) && \text{for } A \subseteq G \\ d^\diamond &= \{g \in G \mid d \sqsubseteq \delta(g)\} && \text{for } d \in D \end{aligned} \tag{1}$$

The pairs (A, d) satisfying $A \subseteq G$, $d \in D$, $A^\diamond = d$, and $A = d^\diamond$ are called *pattern concepts* of $(G, (D, \sqcap), \delta)$, with *pattern extent* A and *pattern intent* d. Pattern concepts are ordered with respect to set inclusion on extents. The ordered set of pattern concepts makes a lattice, called *pattern concept lattice*. Operator $(\cdot)^{\diamond\diamond}$ is an algebraical closure operator on patterns, since it is idempotent, extensive, and monotone.

If objects are described by binary attributes from set M, then $D = \wp(M)$, the powerset of M, and $\delta(g)$ is prime operator $(\cdot)'$ in the context (G, M, I): $\delta(g) = \{m \in M \mid gIm\}$, and $d_1 \sqcap d_2 = d_1 \cap d_2$ where $d_1, d_2 \in D$. So, subsumption corresponds to set inclusion: $d_1 \sqsubseteq d_2 \Leftrightarrow d_1 \sqcap d_2 = d_1 \Leftrightarrow d_1 \cap d_2 = d_1 \Leftrightarrow d_1 \subseteq d_2$.

So, if all patients' initial features are binary we can use them as binary attributes directly. However we also aim at dealing with nominal initial features. Consider patients are described by k initial features $\{\psi_1, ..., \psi_k\}$, all of them are nominal (binary is a particular case) and their values are coded as natural numbers. So, if ψ_i takes l_i values we assume that the range of ψ_i is $\{1, ..., l_i\}$. For each ψ_i we construct l_i binary attributes $\{\beta_i^1, ..., \beta_i^{l_i}\}$ such that $\beta_i^j : G \to \{0, 1\}$ and $\beta_i^j(g) : g \to \psi_i(g) = j$ where $g \in G, j = 1, ..., l_i$. As a result we get $\sum_{i=1,...,n} l_i$ binary attributes to which pattern structures can be applied as it is shown above.

2.2 Pattern Structures on Intervals

To operate with numerical features *interval pattern structures* [15–17] can be applied. Let us consider each patient is described by n numerical and no nominal initial features. In our notation G corresponds to the set of patients. So, let $\{\varphi_1, \varphi_2, ..., \varphi_n\}$ be a set of functions represented patients' initial features such that $\varphi_i : G \to \mathbb{R}$ for $i = 1, ..., n$. For each feature φ_i we construct a corresponding interval attribute $\alpha_i : G \to [\mathbb{R}, \mathbb{R}]$ such that if $\varphi_i(g) = x$ for $g \in G$, then $\alpha_i(g) = [x, x]$, where $x \in \mathbb{R}$. These attributes are used for pattern structures construction.

Each object is described by a n-dimensional tuple of intervals. Let a and b be tuples of n intervals, so $a = \langle [v_i, w_i] \rangle_{i=1,...,n}$ and $b = \langle [x_i, y_i] \rangle_{i=1,...,n}$, where $v_i, w_i, x_i, y_i \in \mathbb{R}\ \forall i = 1, ..., n$. In this case the similarity operation $\sqcap$ is defined by the meet of tuple components:

$$a \sqcap b = \langle [v_i, w_i] \rangle_{i=1,...,n} \sqcap \langle [x_i, y_i] \rangle_{i=1,...,n} = \langle [v_i, w_i] \sqcap [x_i, y_i] \rangle_{i=1,...,n}, \qquad (2)$$

where $[v_i, w_i] \sqcap [x_i, y_i] = [\min(v_i, x_i), \max(w_i, y_i)]$.

Hence, subsumption on tuples of interval is defined as:

$$\begin{aligned} a \sqsubseteq b \Leftrightarrow [v_i, w_i] \sqsubseteq [x_i, y_i]_{i=1,...,n} \Leftrightarrow [v_i, w_i] \sqcap [x_i, y_i] = [v_i, w_i]_{i=1,...,n} \Leftrightarrow \\ \Leftrightarrow [\min(v_i, x_i), \max(w_i, y_i)] = [v_i, w_i]_{i=1,...,n} \Leftrightarrow [v_i, w_i] \supseteq [x_i, y_i]_{i=1,...,n}. \end{aligned} \qquad (3)$$

For example, $\langle [2, 6], [4, 5] \rangle \sqsubseteq \langle [3, 4], [5, 5] \rangle$ as $[2, 6] \sqsubseteq [3, 4]$ and $[4, 5] \sqsubseteq [5, 5]$.

2.3 Pattern Structures on Mixed Tuples

In the previous sections we consider separately the cases of nominal and numerical initial features. However, the situation when patients have both nominal and numerical initial features seems more natural. As it is described above we associate the set of binary attributes with each nominal feature and an interval attribute - with each numerical one. So, $d \in D$ is a tuple, where components are intervals and binary attributes. Let us have k binary and n interval attributes. Assume $d = \langle \alpha, \beta \rangle$ where α is the tuple of intervals of length n, and β is the subset of binary attributes. If $d_1, d_2 \in D$, $d_1 = \langle \alpha_1, \beta_1 \rangle$, and $d_2 = \langle \alpha_2, \beta_2 \rangle$ similarity operator can be set as $d_1 \sqcap d_2 = \langle \alpha_1 \sqcap \alpha_2, \beta_1 \sqcap \beta_2 \rangle$ where similarity operators for the tuples of intervals and the sets of binary attributes are defined above. The subsumption is also defined by subsumption on the tuples of intervals and the sets of binary attributes:

$$\begin{aligned} d_1 \sqsubseteq d_2 \iff d_1 \sqcap d_2 = d_1 \iff \langle \alpha_1, \beta_1 \rangle \sqcap \langle \alpha_2, \beta_2 \rangle = \langle \alpha_1, \beta_1 \rangle \iff \\ \iff \alpha_1 \sqcap \alpha_2 = \alpha_1, \beta_1 \sqcap \beta_2 = \beta_1 \iff \alpha_1 \sqsubseteq \alpha_2, \beta_1 \sqsubseteq \beta_2. \end{aligned} \tag{4}$$

3 Pattern Concepts Generation

Pattern concepts can be computed by Close by One (CbO) algorithm. It produces a tree structure on pattern concepts where edges represent a subset of lattice edges. For the following analysis we do not need the whole lattice but the tree-structure provided by CbO is helpful for implementation of stopping criterion. Considering the top of the lattice is the pair $(G, G^\diamond)$, and the bottom is $(\emptyset, \emptyset^\diamond)$ CbO starts from the bottom and proceeds "object-wise". However, when the number of objects is larger than the number of attributes processing top-down allows one to reduce computation time. Moreover, for the given problem we need to generate pattern concepts with as large extents as possible to detect the difference in treatment efficiency. So, it is more reasonable to start generation process from the top of the lattice. As we operate on descriptions consisting from interval and binary attributes classical CbO must be adapted to such descriptions. For this purpose a version of CbO is proposed below. The idea is to order elements of descriptions, and start to reduce descriptions by changing its elements in this order.

Let objects be described by n interval and k binary attributes. So, we can rewrite description as $d = \langle v_1, w_1, ..., v_n, w_n, b_1, ..., b_k \rangle$, where v_i and w_i are the left and right bounds of the i-th interval attribute for $i = 1, ..., n$, and b_j indicates whether description d contains the j-th binary attribute for $j = 1, ..., k$. So, if b_j is 0 $\delta(g)$ does not contain the j-th binary attribute for all $g \in d^\diamond$, and if b_j is 1, then $\delta(g)$ may or may not contain the j-th binary attribute for all $g \in d^\diamond$. In other words, $1 \sqcap 0 = 0$ or $0 \sqsubseteq 1$. The definition of similarity operator remains the same: we take minimum of left bounds, maximum of right bounds, and set intersection, which in the given notation can be written as element-wise conjunction of indicator vectors:

$$d_1 \sqcap d_2 = \langle \langle \min(v_{1,i}, v_{2,i}), \max(w_{1,i}, w_{2,i}) \rangle_{i=1,...,n}, \langle b_{1,j} \wedge b_{2,j} \rangle_{j=1,...,k} \rangle . \tag{5}$$

The introduced version of CbO starts from the most general description and specifies it by reducing intervals and adding binary features. For interval reduction it is necessary to choose some step value s_i for each interval attribute ($i = 1, ...n,$). If we aim at some sort of scaling we can set these values by ourselves, or if scaling is unwanted s_i is set to the smallest difference between values of the initial feature corresponding to the i-th interval attribute.

Further we denote $d_{all} = \bigsqcap_{g \in G} \delta(g)$, $min(d_1, d_2)$ denotes the minimum position of unequal elements of tuples d_1 and d_2 in element-wise comparison. Let $suc(d)$ denote the set of all children of the node corresponding to the description d. Let $prev(d)$ return the parent of d, $address(d)$ return the address of d, and $nexti(d)$ store the position of the description tuple d which must be changed at the next algorithm returning to d. Let $\div$ denote integer division, % denote residue of division, and $[\cdot]$ be an operator of taking the element of the tuple. Function AddConcept set required links between tree nodes when a new node is added. Function OneIteration changes the description d_{curr} given as argument in position $nexti(d_{curr})$ and takes closure of the changed description by $(\cdot)^{\diamond\diamond}$. If min of the closure and d_{curr} is not less than $nexti(d_{curr})$ then the function returns the closure, otherwise it returns d_{curr}.

def AddConcept($parent, child$)
  $suc(parent) \leftarrow address(child)$
  $prev(child) := parent$
  $nexti(child) := nexti(parent)$

def OneIteration(d_{curr})
  $d_{new} := d_{curr}$
  $i := nexti(d_{curr})$
  if $i \leq 2n$ **then**
    $q = i \div 2$
    $r = i\%2$
    $d_{new}[i] := d_{new}[i] - s_q(2r - 1)$
    $d_{add} := d_{new}^{\diamond\diamond}$
    if $d_{new}[2q] \leq d_{new}[2q + 1]$ **and** $min(d_{curr}, d_{add}) \geq i$ **then**
      AddConcept(d_{curr}, d_{add})
      return d_{add}
    else return d_{curr}
  else
    $d_{new}[i] := d[i] \wedge 1$
    $d_{add} := d_{new}^{\diamond\diamond}$
    if $d_{new}[i] \neq d[i]$ **and** $min(d, d_{add}) \geq i$ **then**
      AddConcept(d_{curr}, d_{add})
      return d_{add}
    else return d_{curr}

$d := d_{all}, nexti(d) := 1, prev(d) := \emptyset, suc(d) := \emptyset$
until $d = d_{all}$ and $nexti(d) > 2n + k$ **do**
  until $nexti(d) > 2n + k$ **do**
    d_{add}:=OneIteration(d)
    if $d \neq d_{add}$ **then**

```
5.            d := d_add
6.            output(d, d^◇)
7.        else nexti(d) := nexti(d) + 1
8.    d := prev(d)
```

Lines 7 and 14 of function OneIteration has complexity $O((2n+k)|G|)$, and all lines 3–8 of the main part of the algorithm are performed at most in this time. The loop starting at line 2 is repeated $2n|G|+k$ times at worst (as in the worst case each boundary of all interval attributes can take $|G|$ values), while the loop starting at line 1 is repeated $|L|$ times exactly, where $|L|$ is the number of pattern concepts. So, the algorithm has complexity $O((2n+k)|G|(2n|G|+k)|L|)$. The complexity is higher than that of CbO, $O((2n+k)|G|^2|L|)$, but in practice our algorithm may become faster when n and k are small, and the number of numerical values in data is less than $|G|$.

4 Stopping Criterion

As it is mentioned above it may not be required to generate all pattern concepts, and the version of CbO may stop when subgroups with difference in treatment efficiency and maximal possible extent are generated. To estimate the difference in efficiency for some description d we define a difference measure which depends on the sets of outcomes of patients who match d and have received the same treatment, and if the value of this measure satisfies some criterion of significance the proposed version of CbO stops to generate specification of the description which is currently in work.

Let p be the number of comparing treatment strategies, and d is the description currently processed by the algorithm. Assume $Q_i = \{outcome(g) \mid g \in d^\diamond, treatment(g) = i\}$ for all $i = 1, ..., p$, where $outcome(g)$ is the outcome of g, and $treatment(g)$ is the treatment assigned to g. So, $|Q_i|$ denotes the number of patients received treatment i. We define difference measure μ which takes the sets of outcomes corresponding to each treatment strategy and returns the estimation of difference, and set threshold ε.The criterion looks like if $\mu(Q_1, ..., Q_p) > \varepsilon$ the algorithm does not generate the children of the currently processed node and returns to its parent node.

Except the stopping criterion itself several additional restrictions are necessary. So, if a subgroup does not contain patients per each of p treatment strategies we are not able to compute μ and need to return to the parent subgroup without generating children nodes, since the antimonotonicity of operator $(\cdot)^\diamond$ ensures this property for all descendants of the current node. Also, it may be important to result in descriptions with approximately equal number of patients per treatment strategy in corresponding subgroups. Therefore, additional parameter $\lambda \in [0, 1]$ is provided to control the ratio of each treatment strategy in a subgroup (see line 4 in function Restrictions). If λ is set to one this restriction is deactivated, if it is set to zero the numbers of patients per each treatment strategy must be equal. Let $outc(d)$ be $\langle Q_i \rangle_{i=1,...,n}$, $outc(d)[i]$ be Q_i, and $|\cdot|$

return the power of the set. Function Restrictions checks fulfillment of these restrictions for a particular description. Let $isempty(d)$ be $False$ if the subgroups corresponding to description d do not contain patients from every treatment strategy, and $True$ otherwise. Let also $notBalanced(d)$ be $False$ if the subgroup corresponding to d is not balanced (do not satisfy restriction on proportion of treatment strategies in the subgroup), and $True$ otherwise.

```
def Restrictions(d)
isempty(d) := False, notBalanced(d) := False
for j from 1 to p do
    isempty(d) := isempty(d) ∨ (|outc(d)[j]| = 0)
    share := |outc(d)[j]| / |d◇|
    notBalanced(d) := notBalanced(d) ∨ ((1−λ)/p ≥ share) ∨ (share ≥ (1+λ)/p)
```

```
d := d_all, nexti(d) := 1, prev(d) := ∅, suc(d) := ∅
Restrictions(d)
until d = d_all and nexti(d) > 2n + k do
    if isempty(d) then
        d := prev(d)
        continue
    if notBalanced(d) or μ(outc(d)) ≤ ε then
        d_add:=OneIteration(d)
        if d ≠ d_add then
            d := d_add
            Restrictions(d)
        else nexti(d) := nexti(d) + 1
    else
        output(d, d◇)
        d := prev(d)
```

The algorithm outputs the set of maximal size subgroups (i.e. maximal extents) with significant difference in treatment efficiency. Since we do not construct the whole pattern lattice the resulting set may contain subgroups which subsumed under the other subgroups from this set. Smaller subgroups should be excluded from the output by post-processing.

5 Application to ALL Dataset

5.1 Dataset

The dataset consists of more than 2000 patients from 1 to 18 years old with newly diagnosed ALL. All of them were included into the standard risk group (SRG) or into the intermediate risk group (ImRG) of randomized clinical trial MB-ALL-2008 [19]. The protocol of this trial contains three stages of treatment for SRG and ImRG: induction (36 days), consolidation (25 weeks), and maintenance (2-3 years). In this paper we only focus on the induction stage. Induction therapy aims at bringing a patient into remission and is very toxic. At this stage of treatment patients from SRG and ImRG are randomized into 3 and 2 treatment strategies correspondingly. Let us code them as T_1, T_2, and T_3 in SRG and T_4 and T_5 in ImRG. SRG and ImRG parts of the dataset may be considered as two independent datasets because SRG and ImRG therapies differ considerably.

Each patient from the dataset is described by the set of initial features, treatment strategy which he or she was assigned at randomization, and outcome features. From all initial features 8 were chosen for the analysis: sex (male or female), age (in years from the birth date to the start of the therapy), initial white blood count (per nl) (WBC), immunophenotype (B- or T-ALL), central nervous system (CNS) status (normal, cytosis is less than 5 per mcl and blast cells, neuroleukemia), liver enlargement (in cm), spleen enlargement (in cm), mediastinum status (normal or pathological). As for outcome features of the dataset each outcome feature consists of two parts: the result of the therapy and time from the beginning of the therapy to the date when the result was fixed (in years). In this dataset we have two such features. One of them is fixing the time before patient's death.This feature has three possible states: alive, lost to follow-up or death. When a patient is alive or lost to follow-up we say that censoring happened because we cannot measure exactly the time before death. The other outcome feature represents the time before a negative event . This feature can possess the following states: alive in remission, lost to follow-up, death in remission, secondary tumor, relapse or metastases, nonresponse to the therapy or disease progression, or death in induction. Alive in remission and lost to follow-up events correspond to censoring. Two types of outcomes are used for different variants of treatment efficiency estimation which are presented below.

Finally, we exclude patients with missed values from the further analysis and result in 1221 SRG patients: 387, 366, and 368 patients received T_1, T_2, and T_3 respectively. 929 patients in ImRG: 467 and 462 patients received T_4 and T_5 respectively.

5.2 Data Preprocessing

As it was shown above the initial features of the patients should be transformed into the tuple of interval and binary attributes. Age, WBC, liver enlargement, and spleen enlargement are numerical features, so for each of them an interval attribute is created. Moreover, we scale them a little to obtain subgroup descriptions which make sense for physicians. For example, it is not correct enough in medical terms that one treatment is more effective for patients, let's say, up to 5.4 years old. Similar limitations should also be applied to other 3 numerical features. To answer these limitations we propose the following way of interval attributes construction:

1. Let one set the steps of interval reduction in CbO to $s_{age} = s_{liver} = s_{spleen} = 1$ and $s_{WBC} = 10$.
2. For each $name \in \{age, liver, spleen, WBC\}$ and for every $g \in D$, where D is the set of all patients, if $name(g) = x$ then the value of the corresponding interval attribute is set to $[s_{name} \cdot \lfloor x/s_{name} \rfloor, s_{name} \cdot \lceil x/s_{name} \rceil]$.

All nominal features (sex, immunophenotype, CNS status, and mediastinum status) are converted to the set of binary attributes in the way presented in section 2.1. For instance, we construct two binary attributes corresponding to sex: one indicating males and another indicating females.

5.3 Methodology of Generating Hypotheses

The algorithm of subgroup descriptions generation is proposed in Section 4. It requires to set the difference measure. For the data of the childhood ALL we choose log-rank statistics [20]. As it is a statistical method of detecting the difference between two (or several) survival curves [20–22] the threshold is naturally chosen to satisfy 95% level of confidence of log-rank test. So, if p-value of two-sided log-rank test is less than 5%, we output current description as a potential subgroup description and do not generate its children, otherwise we continue to generate children descriptions in accordance with the introduced algorithm. Finally, from the set of potential subgroup descriptions we delete descriptions subsumed under other potential subgroup descriptions.

As in SRG three treatment strategies (T_1, T_2, T_3) are compared, the algorithm selected those subgroups where the difference between any pair of the treatment strategies is significant. Assume one of descriptions we get is d. For the subgroup described by this description we should compare every pair of treatment strategies: T_1 and T_2, T_1 and T_3, and T_2 and T_3. For each pair where log-rank test detects the difference with confidence 95% and power 80% we make a hypothesis. For instance, if pair T_i and T_j satisfies these requirements then a hypothesis says that treatment strategies T_i and T_j affect patients described by d differently. At the same time if the survival curve for T_i, for instance, is located above the survival curve for T_j we can even say that T_i is better for patients described by d than T_j. For patients from ImRG only two treatment strategies are compared. Therefore it is enough to estimate power, and if it is more than 80% we make a hypothesis in the same way.

5.4 Summary of Results

The proposed algorithm was run to compare separately overall survival (OS)[23], event-free survival (EFS)[24], and relapse-free survival (RFS)[25] in each risk group with λ set to $\frac{1}{3}$ and $\frac{1}{5}$ for SRG and ImRG respectively. We also add restrictions on the size of the subgroups: not less than 20 and 200 patients per each treatment strategy for SRG and ImRG, respectively (the choice is explained by the greater number of patients per treatment and possible descriptions for ImRG). So, as a result three sets of subgroup descriptions were obtained for each risk group (SRG and ImRG), one per each type of survival.

The results of experiments for SRG are presented in Table 1. OS, EFS, and RFS stand for three types of survival described above, CbO stands for the proposed approach, and IT stands for Interaction Tree from [9]. To construct an interaction tree the same restriction on the size of subgroups (i.e. leaves) was set: not less than 20 patients per each treatment strategy. Performing IT with pruning results in no subgroups, therefore we compared to unpruned trees. To estimate subgroups in each of them paired logrank test p-values for every pair of compared treatment strategies are estimated. We have also performed bootstrap sampling on 1000 samples, and for each subgroup we estimate p-value median and 0.95 unpivotal confidence interval for the difference between long-term survival estimations for every pair of compared treatment strategies. We count the number of subgroups where the result of comparison of even one pair of compared treatment strategies in this subgroups satisfies restrictions at the heading. So, p corresponds to p-value of paired logrank test, p_m is a median estimated by bootstrap, d_l and d_r are the left and the right boundaries of 0.95 bootstrap confidence interval for the difference in long-term survival.

Table 1. Number of subgroups obtained for SRG corresponding to different types of survival and applied apprioach to subgroup detection.

survival type	censoring rate	approach	number of subgroups	$p < 0.05$	$p_m < 0.05$	$p < 0.05$ & $p_m < 0.05$	$p < 0.05$ & $d_l \cdot d_r > 0$	$p < 0.05$ & $p_m < 0.05$ & $d_l \cdot d_r > 0$
OS	0.945	CbO	67	55	50	49	50	46
		IT	10	1	1	1	1	1
EFS	0.923	CbO	166	101	104	100	87	87
		IT	12	1	0	0	0	0
RFS	0.963	CbO	89	54	47	45	21	20
		IT	11	1	1	1	1	1

For ImRG we obtained 2559 and 153 subgroups based on OS and EFS respectively and no subgroups based on RFS since the superiority of T_5 over T_4 in RFS holds for the whole set of ImRG patients. Moreover, all obtained subgroups based on OS and EFS confirm that T_5 is better than T_4. For this reason we did not carry out an experiments on ImRG by applying Interaction Trees.

5.5 An Example of Generated Hypotheses

Given all hypotheses for ImRG propose the superiority of T_5 over T_4 it is more interesting to look at the hypotheses for SRG. Short-term estimation of the treatment strategy efficiencies carried out by physicians shows that T_1 is significantly worse than T_2 and T_3 for all patients from SRG while long-term estimations show no significant difference. However, by applying the proposed algorithm we found several subgroups where T_1 is better than T_2 and T_3 in long-term. For instance, the description was obtained on the basis of difference in OS: $4 \leq$ *age* , $3 \leq$ *liver enlargement* ≤ 7*, and normal mediastinum status.* Testing T_1 vs T_2 and T_1 vs T_3 we got p-values 1.4% and 1.9% and power estimations 86% and 94%. There are approximately 60 patients per treatment strategy in the subgroup. OS curves for the patients which can be described by even one of these descriptions is presented in Fig. 1. Confidence intervals of p-values obtained from 1000 sample bootstrap: [0.3%, 1.6%] and [0.7%, 1.7%]. So, the advantage of strategy

T_1 over T_2 and T_3 for patients matching the description seems confident and independent from the certain data.

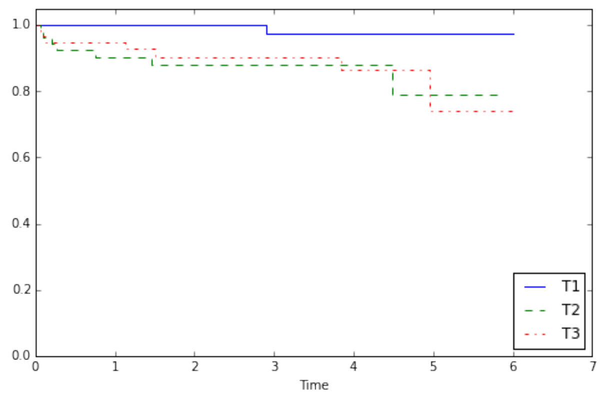


Fig. 1. OS curves for subgroups showing the superiority of T_1 over T_2 and T_3.

6 Conclusion

In this paper we have introduced an approach to solving the problem of determining relevant subgroups of patients for therapy optimization. The approach is based on representing data by numerical pattern structures and applying the version of CbO algorithm. The algorithm computes the pattern lattice top-down (starting with the most general descriptions) and its stopping criterion allows one to generate subgroups with significant differences in the efficiency of treatment strategies containing the maximal possible number of patients to satisfy statistical power restrictions. This approach allows one to avoid binarization or using similarity measures on patients, which can result in artifacts. The approach is also not biased by local optimization heuristics used for constructing decision trees and random forests. The situations when various subgroups are not disjoint will be the subject of further study.

Acknowledgments

We would like to thank Prof. Alexander I. Karachunskiy and his colleagues from the Federal Research Institute of Pediatric Hematology, Oncology and Immunology, Moscow, Russia for helpful discussions and for providing us with data on MB-ALL-2008 protocol.

References

1. Levin, K.A.: Study Design VII. Randomised Controlled Trials. Evident-Based Dentistry 8, 22-23, Nature Publishing Group (2007)
2. Zeileis, A., Hothorn, T., Hornik, K. Model-Based Recursive Partitioning. J. Comput. Graph. Stat. 17(2), 492-514 (2008)
3. Foster, J., Taylor, J., Ruberg, S. Subgroup Identification from Randomized Clinical Trial Data. Stat. Med. 30(24), 2867-2880 (2011)
4. Zhao, Y., Zeng, D., Rush A.J., Kosorok, M.R. Estimating Individualized Treatment Rules Using Outcome Weighted Learning. J. Am. Stat. Assoc. 107(449), 1106-1118 (2012)
5. Korepanova, N., Kuznetsov, S.O., Karachunskiy, A.I.: Matchings and Decision Trees for Determining Optimal Therapy. In: Ignatov D.I. et al. Analysis of Images, Social Networks and Texts, Proc. 3rd International Conference (AIST2014), Communications in Computer and Information Science, Vol. 436, pp. 101-110. Springer (2014)
6. Dusseldorp, E., Conversano ,C., Van Os, B.J. Combining an Additive and Tree-Based Regression Model Simultaneously: STIMA. J. Comput. Graph. Stat. 19(3), 514-530 (2010)
7. Dusseldorp, E., Van Mechelen, I. Qualitative Interaction Trees: a Tool to Identify Qualitative Treatment-Subgroup Interactions. Stat. Med. 33, 219–237 (2014)
8. Lipkovich, I., Dmitrienko, A., Denne, J., Enas, G. Subgroup Identification Based on Differential Effect Search a Recursive Partitioning Method for Establishing Response to Treatment in Patient Subpopulations. Stat. Med. 30(21), 2601-2621 (2011)
9. Su, X., Zhou, T., Yan, X., Fan, J., Yang, S. Interaction Trees with Censored Survival Data. Int. J. Biostat. 4(1), 2 (2008)
10. Su, X., Tsai, C.L., Wang, H., Nickerson, D.M., Li, B. Subgroup Analysis via Recursive Partitioning. J. Mach. Learn. Res. 10, 141-158 (2009)
11. Loh, W.-Y., He, X., Man, M. A Regression Tree Approach to Identifying Subgroups with Differential Treatment Effects. Stst. Med. 34, 1818 – 1833 (2015)
12. Ching-Hon, P., Robison, L.L., Look, A.T. Acute Lymphoblastic Leukaemia. The Lancet. 9617(371), 1030 – 1043, The Lancet (2008)
13. Karachunskiy, A., Roumiantseva, J., Lagoiko, S. (eds.) Efficacy and Toxicity of Dexamethasone vs Methylprednisolone Long-Term Results in More than 1000 Patients from the Russian Randomized Multicentric trial ALL-MB 2002, Letter to the Editor. Leukemia. 29, 1955–1958, Nature Publishing Group (2015)
14. Ganter, B., Kuznetsov, S.O. Pattern Structures and Their Projections. In: Proc. Stumme, G., Delugach, H. (eds.) 9th International Conference on Conceptual Structures (ICCS 2001). Lecture Notes in Artificial Intelligence, vol. 2120, pp. 129–142, Springer (2001)
15. Kuznetsov, S.O. Pattern Structures for Analyzing Complex Data. In: Sakai H. et al. (eds.) Proc. 12th International Conference on Rough Sets, Fuzzy Sets, Data Mining and Granular Computing (RSFDGrC 2009). Lecture Notes in Artificial Intelligence, vol. 5908, pp. 33–44, Springer (2009)
16. Kaytoue, M., Kuznetsov, S.O., Napoli, A., Duplessis, S. Mining Gene Expression Data with Pattern Structures in Formal Concept Analysis. Information Sciences, 10(181), 1989–2001, Elsevier, New York (2011)
17. Kuznetsov, S.O. Scalable Knowledge Discovery in Complex Data with Pattern Structures. In: Maji, P., Ghosh, A., et al. (eds.) Proc. 5th International Conference

Pattern Recognition and Machine Intelligence (PReMI'2013). Lecture Notes in Computer Science, vol. 8251, pp. 30–41, Springer (2013)

18. Kuznetsov, S.O. A Fast Algorithm for Computing All Intersections of Objects from an Arbitrary Semilattice. Nauchno-Tekhnicheskaya Informatsiya Seriya 2 - Informatsionnye Protsessy i Sistemy, 1, 17–20 (1993).
19. A service of the U.S. National Institutes of Health, https://clinicaltrials.gov/ct2/show/NCT01953770
20. Kleinbaum, D.G., Klein, M.: Kaplan-Meier Survival Curves and the Log-Rank Test. In: Survival Analysis, pp. 55–96, Springer New York (2012)
21. Kaplan, E.L., Meier, P. Nonparametric Estimation from Incomplete Observations. Journal of the American Statistical Association, 282(53), 457–481 (1958)
22. May, W.L. Kaplan-Meier Survival Analysis. In: Encyclopedia of Cancer, pp. 1590-1593, Springer Berlin Heidelberg (2009)
23. National Cancer Institute, http://www.cancer.gov/publications/dictionaries/cancer-terms?cdrid=655245
24. National Cancer Institute, http://www.cancer.gov/publications/dictionaries/cancer-terms?cdrid=655269
25. National Cancer Institute, http://www.cancer.gov/publications/dictionaries/cancer-terms?cdrid=655254

NextClosures with Constraints

Francesco Kriegel

Institute of Theoretical Computer Science
Technische Universität Dresden, Dresden, Germany
francesco.kriegel@tu-dresden.de

Abstract. In a former paper, the algorithm *NextClosures* for computing the set of all formal concepts as well as the canonical base for a given formal context has been introduced. Here, this algorithm shall be generalized to a setting where the data-set is described by means of a closure operator in a complete lattice, and furthermore it shall be extended with the possibility to handle constraints that are given in form of a second closure operator. As a special case, constraints may be predefined as implicational background knowledge. Additionally, we show how the algorithm can be modified in order to do parallel *Attribute Exploration* for unconstrained closure operators, as well as give a reason for the impossibility of (parallel) *Attribute Exploration* for constrained closure operators if the constraint is not compatible with the data-set.

Keywords: Formal Concept Analysis, Implication, Canonical Base, Constraint, Closure Operator, Parallel Computation, Parallel Exploration

1 Introduction

Recently, in [18, 20] a parallel algorithm for the computation of canonical bases (and the set of formal concepts as a byproduct) for formal contexts has been introduced. Furthermore, in [18, 19] some extensions have been provided that allow for the parallel exploration of canonical bases by means of experts. However, not all data-sets occurring in practical applications can efficiently be described as a formal context. Henceforth, it can be useful to describe a generalization of the *NextClosures* algorithm which allows to compute canonical bases for implications valid in a closure operator in a complete lattice. Of course, powersets are always complete lattices, and the composition of the two derivation operators of a formal context is a closure operator in the powerset lattice – hence this is indeed a generalization. Since the algorithm proceeds in a level-wise order that is compatible with the set-theoretic subset ordering, the generalization presupposes a strict order-homomorphism of the underlying lattice into the ordered set $(\mathbb{N}, \leq)$ of the natural numbers with their usual ordering (called a *quasi-rank function*). Furthermore, this paper presents some possible applications, e.g., in *Formal Concept Analysis* [13], for interpretations in *Description Logics* [2], and for *Pattern Structures* [12].

First, the notion of a closure operator in a complete lattice is defined, and it is proven that the set of all closure operators in a fixed complete lattice forms a complete lattice itself. Consequently, it is possible to construct infima and suprema of sets of closure operators. As applications, we discuss what it means for implications to be valid in infima and suprema, and furthermore generalize the notion of C-*implications* (*constrained implications*, where C is a closure operator) in [4] from the special case of *Formal Concept Analysis* to the more general case of closure operators.

© Marianne Huchard, Sergei O. Kuznetsov (Eds.): CLA 2016, pp. 231–242, ISBN 978-5-600-01454-1, National Research University Higher School of Economics, 2016.

In [24], Stumme introduced an extension of Ganter's *Attribute Exploration* [10, 11] that can handle background knowledge in form of an implication set. A further generalization [9] allows for arbitrary propositional background knowledge. The input data is given as a formal context $\mathbb{K} = (G, M, I)$ as well as an implication set $\mathcal{L} \subseteq \mathsf{Imp}(M)$ that is valid in $\mathbb{K}$. Then by generalizing the notions of pseudo-intents and canonical bases, it is possible to compute a minimal extension of $\mathcal{L}$ which is sound and complete for $\mathbb{K}$. This minimal extension is called *canonical base* of $\mathbb{K}$ *relative* to $\mathcal{L}$. However, it has not been addressed what can be done in the case where $\mathcal{L}$ is not valid in $\mathbb{K}$. There are at least two possibilities to handle such cases. One may either compute a base for the implications that are valid in $\mathbb{K}$ as well as are entailed by $\mathcal{L}$, or one can compute a base for the constrained implications of $(\mathbb{K}, \mathcal{L})$. We will discuss some details here, and provide solutions by utilizing infima and suprema of closure operators in lattices.

Higuchi has shown in [16] that the set of all closure operators on a set constitutes a complete lattice. More generally, this is also true for the set of all closure operators in a complete lattice. The order of closure operators is defined as follows: A closure operator ϕ is smaller than another closure operator ψ if each closure of ψ is also a closure of ϕ, i.e., if ϕ is more *fine-grained* than ψ which means that each closure of ψ can be expressed as an infimum of closures of ϕ. Furthermore, it is easy to verify that implications are valid in an infimum of two closure operators if, and only if, they are valid in each of the closure operators. The dual case considers implications valid in suprema, which are exactly those implications valid for all models that are closures of all involved closure operators. In general, one may think of infima describing unions of data-sets, and suprema as describing intersections (or mergings) of data-sets.

The last case investigated in this document considers the constrained implications. As input two closure operators are necessary, one describing the data-set and the other one describing the constraints to be met. Then a base for all those implications may be computed, for which both premise and conclusion is a closure of the constraining closure operator, and all common closures of both closure operators are models. In particular, we will show an application where we have some implications which have been proven to be correct, and a possibly faulty data-set. As a result, we may construct implications that are compatible with the background knowledge, and furthermore are valid w.r.t. all closures of the data-set which are also models of the background implications.

This document is structured as follows. In Section 2, the notion of a closure operator in a complete lattice is defined, and it is shown that the set of all closure operators constitute a complete lattice itself, i.e., we present an order on closure operators as well as give the equations for the corresponding infimum and supremum operations. Then in Section 3, we consider implications in a complete lattice and define the notion of validity w.r.t. closure operators, as well as entailment for implication sets. Next, Section 4 generalizes the parallel *NextClosures* algorithm from [20] to the case of closure operators (with constraints) in complete lattices. Proofs are ommited, as on the one hand they have already been given in [18], and on the other hand could be generalized from those given in [19, 20] without too much effort. Before finishing this document with some conclusions in Section 6, several possible applications are described, in particular, in *Formal Concept Analysis*, in *Description Logics*, and for *Pattern Structures*.

2 The Complete Lattice of Closure Operators in a Complete Lattice

Throughout the whole section, we assume that $\underline{M} = (M, \leq, \bigwedge, \bigvee, \top, \bot)$ is a *complete lattice*, i.e., M is a set; $\leq$ is an order relation on M, i.e., it is reflexive, antisymmetric, and transitive; for each subset $X \subseteq M$, $\bigwedge X$ is the *infimum* of X, i.e., $\bigwedge X \leq x$ for all $x \in X$, and $y \leq x$ for all $x \in X$ implies $y \leq \bigwedge X$; dually, $\bigvee X$ is the *supremum* of X; $\top$ is the greatest element and $\bot$ is the smallest element, i.e., $\bot \leq x \leq \top$ for all $x \in M$. We set $x \wedge y \coloneqq \bigwedge\{x, y\}$ as well as $x \vee y \coloneqq \bigvee\{x, y\}$. For $x \in M$, its *(prime) ideal* is $\downarrow x \coloneqq \{\, y \in M \mid y \leq x \,\}$, and its *(prime) filter* is $\uparrow x \coloneqq \{\, y \in M \mid x \leq y \,\}$. For further information on (complete) lattices, the interested reader is referred to [5, 7, 13, 14].

A *closure operator* in $\underline{M}$ is a mapping $\phi \colon M \to M$ that is extensive, monotone, and idempotent, cf. [6, 7, 16]. $\mathsf{ClOp}(\underline{M})$ denotes the set of all closure operators in $\underline{M}$. An element $x \in M$ is a *closure* of ϕ if $x = x^{\phi}$, and we shall denote the set of all closures of ϕ by $\mathsf{Clo}(\phi)$. Furthermore, it is well-known that the following statements are equivalent characterizations of a closure operator ϕ:

1. $x \leq x^{\phi}$, $x \leq y \Rightarrow x^{\phi} \leq y^{\phi}$, and $x^{\phi\phi} = x^{\phi}$, for all $x, y \in M$.
2. $x \leq y^{\phi} \Leftrightarrow x^{\phi} \leq y^{\phi}$ for all $x, y \in M$.
3. $x \vee y^{\phi\phi} \leq (x \vee y)^{\phi}$ for all $x, y \in M$.
4. $x \leq x^{\phi}$, and $(x \vee y)^{\phi} = (x^{\phi} \vee y^{\phi})^{\phi}$, for all $x, y \in M$.

It is easy to verify that the following statements hold for a closure operator ϕ:

1. $(x \wedge y)^{\phi} \leq x^{\phi} \wedge y^{\phi}$ for all $x, y \in M$.
2. $(x^{\phi} \wedge y^{\phi})^{\phi} = x^{\phi} \wedge y^{\phi}$ for all $x, y \in M$.

A *closure system* in $\underline{M}$ is a $\bigwedge$-closed subset of M. A subset $P \subseteq M$ is a closure system in $\underline{M}$ if, and only if, $\{\, p \in P \mid x \leq p \,\} = \uparrow x \cap P$ has a smallest element for all $x \in M$. Then there is a one-to-one correspondence between closure operators and closure systems as follows. For each closure operator in $\underline{M}$, the set of its closures is a closure system in $\underline{M}$. Vice versa, if P is a closure system in $\underline{M}$, then $\phi_P \colon x \mapsto \bigwedge(\uparrow x \cap P)$ is a closure operator in $\underline{M}$. The operations are mutually inverse.

It turns out that the set of all closure operators in a complete lattice constitutes a complete lattice itself, see also [16, 23]. Indeed, closure operators can be ordered by $\preceq$, where $\phi \preceq \psi$ if $\mathsf{Clo}(\phi) \supseteq \mathsf{Clo}(\psi)$. This condition is equivalent to both $\phi \circ \psi = \psi$ and $\phi \leq \psi$ (w.r.t. pointwise order, i.e., $x^{\phi} \leq x^{\psi}$ for all $x \in M$). Obviously, the smallest closure operator is given by the identity mapping $\underline{\bot} \colon x \mapsto x$, and the greatest closure operator is given by the constant mapping $\overline{\top} \colon x \mapsto \top$. If Φ is a set of closure operators in $\underline{M}$, then its *infimum* $\curlywedge \Phi$ and *supremum* $\curlyvee \Phi$ are given by the following equations:

$$\curlywedge \Phi \colon x \mapsto \bigwedge\{\, x^{\phi} \mid \phi \in \Phi \,\},$$

$$\text{and} \quad \curlyvee \Phi \colon x \mapsto \bigwedge\{\, y \mid x \leq y \text{ and } y = y^{\phi} \text{ for all } \phi \in \Phi \,\}.$$

Of course, all complete lattices are (isomorphic to) a formal concept lattice, cf. [13]. In particular, the lattice of closure operators is isomorphic to the concept lattice of $(\mathsf{ClOp}(\underline{M}), \mathsf{ClOp}(\underline{M}), \preceq)$, but also to the concept lattice of the smaller formal context $(\mathsf{ClOp}(\underline{M}), M, C)$ where $\phi \; C \; x$ if x is a closure of ϕ. The corresponding isomorphisms are

$$\phi \mapsto (\downarrow \phi, \mathsf{Clo}(\phi)) \quad \text{and} \quad (\Phi, X) \mapsto \curlyvee \Phi \colon x \mapsto \bigwedge(\uparrow x \cap X).$$

3 Implications in Closure Operators

An *implication* in M is an expression of the form $p \to c$ with *premise* $p \in M$ and *conclusion* $c \in M$. An element $m \in M$ is a *model* of $p \to c$ if $p \leq m$ implies $c \leq m$. It is *valid* in ϕ if all closures of ϕ are models of $p \to c$, and we shall denote this by $\phi \models p \to c$. Furthermore, it can easily be shown that $\phi \models p \to c$ is equivalent to $c \leq p^{\phi}$. We shall denote the set of all implications in M by $\mathsf{Imp}(M)$. For an implication set $\mathcal{L} \cup \{p \to c\} \subseteq \mathsf{Imp}(M)$, we say that $\mathcal{L}$ *entails* $p \to c$, symbolized as $\mathcal{L} \models p \to c$, if each model of $\mathcal{L}$, i.e., each model of all implications in $\mathcal{L}$, is a model of $p \to c$. For each element $x \in M$, there is a smallest element $x^{\mathcal{L}}$ above x that is a model of $\mathcal{L}$, and the corresponding closure operator satisfies $x^{\mathcal{L}} = \bigvee \{ x^{\mathcal{L},n} \mid n \geq 1 \}$ where

$$x^{\mathcal{L},1} \coloneqq x \vee \bigvee \{ c \mid \exists p\colon p \to c \in \mathcal{L} \text{ and } p \leq x \},$$
$$\text{and} \quad x^{\mathcal{L},n+1} \coloneqq (x^{\mathcal{L},1})^{\mathcal{L},n} \text{ for each } n \in \mathbb{N}.$$

Then, $\mathcal{L}$ entails $p \to c$ if, and only if, $c \leq p^{\mathcal{L}}$. Additionally, we define a similar closure operator $\mathcal{L}^*$ where $p \leq x$ is replaced with $p \lneq x$ in the definition of $x^{\mathcal{L},1}$. For an implication set $\mathcal{L}$, we symbolize by $\mathcal{L}{\restriction_k}$ the subset consisting of all implications from $\mathcal{L}$ the premises of which have a quasi-rank not exceeding k.

A *pseudo-closure* of ϕ is an element $p \in M$ which is not a closure of ϕ, but contains the closure of each strictly smaller pseudo-closure. $\mathsf{PsClo}(\phi)$ denotes the set of all pseudo-closures of ϕ, and then $\mathcal{B}_{\mathsf{can}}(\phi) \coloneqq \{ p \to p^{\phi} \mid p \in \mathsf{PsClo}(\phi) \}$ constitutes an *implicational base* for ϕ, i.e., for all implications $p \to c \in \mathsf{Imp}(M)$, $\phi \models p \to c$ if, and only if, $\mathcal{B}_{\mathsf{can}}(\phi) \models p \to c$. We call $\mathcal{B}_{\mathsf{can}}(\phi)$ the *canonical base* of ϕ, and furthermore it can be shown that it is a *minimal* implicational base, i.e., there is no base of smaller cardinality. Algorithm 1 can be utilized to compute the canonical base of ϕ, for which the constraining closure operator ψ must be set to the identity mapping $\perp$ that imposes no constraints at all.

3.1 Implications in Infima

As we have seen in Section 2, the infimum of two closure operators ϕ and ψ in a complete lattice $\underline{M}$ is given as $\phi \curlywedge \psi\colon x \mapsto x^{\phi} \wedge x^{\psi}$. The corresponding closure system is the smallest that contains all closures of ϕ as well as all closures of ψ. Consequently, an implication is valid in the infimum if, and only if, it is valid in both closure operators. For formal contexts $\mathbb{K}_1$ and $\mathbb{K}_2$, the infimum of their intent closure operators can be obtained as the intent closure operator of the subposition $\frac{\mathbb{K}_1}{\mathbb{K}_2}$. In general, the infimum somehow corresponds to the union of two data-sets.

Using the infimum operation it is furthermore possible to construct implicational bases from streams. Suppose that $(\phi_n)_{n \in \mathbb{N}}$ is a sequence of closure operators in $\underline{M}$ such that ϕ_n is only available at time point n, i.e., we do not have access to the whole sequence at once, but only to the most recent closure operator. However, for each time point $n \in \mathbb{N}$, it is desired to have a base for the implications that are valid in all closure operators ϕ_ℓ with $\ell \leq n$. This can easily be achieved as follows:

1. Set $\mathcal{B}_0 \coloneqq \mathcal{B}_{\mathsf{can}}(\phi_0)$.
2. Set $\mathcal{B}_{n+1} \coloneqq \mathcal{B}_{\mathsf{can}}(\mathcal{B}_n \curlywedge \phi_{n+1})$ for each $n \in \mathbb{N}$.

By construction, then each $\mathcal{B}_n$ is an implicational base for $\bigcurlywedge_{\ell=0}^{n} \phi_\ell$, i.e., for the implications which are valid in $\phi_0, \dots, \phi_n$.

3.2 Implications in Suprema

Consider two closure operators ϕ and ψ in a complete lattice $\underline{M}$. Then, their supremum $\phi \curlyvee \psi$ maps each element $x \in M$ to the smallest element of M that is greater than x, and both a closure of ϕ and ψ. The corresponding closure system is the smallest that contains the intersection $\mathsf{Clo}(\phi) \cap \mathsf{Clo}(\psi)$. Hence, the supremum corresponds to the intersection (or merging) of data-sets. An implication is valid in $\phi \curlyvee \psi$ if, and only if, it has all common closures of ϕ and ψ as models. Note that there may be implications valid in $\phi \curlyvee \psi$ which are neither valid in ϕ, nor valid in ψ.

3.3 Constrained Implications

As a special case of implications being valid in a supremum, we consider so-called *constrained implications*. This notion has first been defined by Belohlávek and Vychodil in [4] for formal contexts. Consider a formal context $\mathbb{K} = (G, M, I)$ and a closure operator C on M, then a C*-implication* over M is an implication the premise as well as the conclusion of which are closures of C. Furthermore, a C-implication is valid in $\mathbb{K}$ if it has as models all intents of $\mathbb{K}$ that are closures of C. We can generalize this to the case of closure operators in complete lattices as follows.

Again, suppose that ϕ and ψ are closure operators in a complete lattice $\underline{M}$. We then call the pair (ϕ, ψ) a *closure operator with constraint* where ϕ is the *constrained*, and ψ is the *constraining* closure operator. An implication $p \to c$ is *constrained* by ψ if both its premise p and conclusion c are closures of ψ. Furthermore, $p \to c$ is *valid* in (ϕ, ψ) if it has all closures of the supremum $\phi \curlyvee \psi$ as models, i.e., if $\phi \curlyvee \psi \models p \to c$. In order to construct bases for constrained implications in a closure operator with constraint, we shall adapt the notions of pseudo-closures and the canonical base accordingly. An element $p \in M$ is a pseudo-closure of (ϕ, ψ) if p is not a closure of ϕ, but a closure of ψ, and furthermore $q^{\phi \curlyvee \psi} \leq p$ for all pseudo-closures $q \lneq p$ of (ϕ, ψ). Then, the set $\mathcal{B}_{\mathsf{can}}(\phi, \psi) \coloneqq \{\, p \to p^{\phi \curlyvee \psi} \mid p \in \mathsf{PsClo}(\phi, \psi) \,\}$ is a minimal implicational base of (ϕ, ψ) where $\mathsf{PsClo}(\phi, \psi)$ denotes the set of all pseudo-closures of (ϕ, ψ).

Comparing the two approaches for computing implications in a supremum, the first one in Section 3.2 only restricts the possible models, in particular, only closures of ϕ are considered that are also closures of ψ. The second approach in Section 3.3 also imposes constraints on the implications, i.e., only implications are considered where both premise and conclusion are closures of ψ.

4 A Generalized NextClosures Algorithm

Let $\underline{M}$ be a complete lattice. Throughout the whole section, we assume that there is a strict order-preserving function $|\cdot| \colon (M, \leq) \to (\mathbb{N}, \leq)$, i.e., $x \lneq y$ implies $|x| \lneq |y|$ for all $x, y \in M$. W.l.o.g. $|\bot| = 0$. (If $|\bot| = n$ for $n > 0$, then $||\cdot|| \colon x \mapsto |x| - n$ is also a strict order-homomorphism, and $||\bot|| = 0$.) Then $|\cdot|$ is a *quasi-rank function* on $(M, \leq)$, and we say that an element $x \in M$ has *quasi-rank* $|x|$. In particular, for all *graded* complete lattices, the corresponding rank function $|\cdot|$ is a quasi-rank function such that furthermore $|x| + 1 = |y|$ if x is a lower neighbor of y. For our purposes, we do not need the additional property that the ranks of neighboring elements only differ by an amount of 1.

Consider two closure operators ϕ and ψ in $\underline{M}$. Then the set containing all closures of $\phi \curlyvee \psi$ and all pseudo-closures of (ϕ, ψ) is a closure system in $\underline{M}$, and the corresponding closure operator is $(\phi, \psi)^* := \mathcal{B}_{\mathsf{can}}(\phi, \psi)^*$. Our aim is to find an automatic method for constructing all closures of $(\phi, \psi)^*$. The following Algorithm 1 has been introduced in [18], and the special case of ϕ being the intent closure operator induced by a formal context, and the constraining closure operator ψ being the identity $\underline{\perp}$ (i.e., no constraints are given), has been handled in [20]. The proof is left out here, as it can be found in [18], or can be generalized from the proof in [20]. We will describe Algorithm 1 in the following text.

According to the definition of a pseudo-closure, for all elements $p \in M$, it suffices to know all strictly smaller pseudo-closures $q \lneq p$ in order to correctly determine whether p is pseudo-closed. Hence, we may compute them in a level-wise approach w.r.t. increasing quasi-rank, since if we have computed all pseudo-closures q where $|q| \lneq |p|$, then we can find those pseudo-closures with $q \lneq p$ among them. In particular, if $\mathcal{L}$ is a set of ψ-implications that contains exactly all implications $p \to p^{\phi \curlyvee \psi}$ where p is a (ϕ, ψ)-pseudo-closure with quasi-rank not exceeding k, and some arbitrary implications with premise quasi-rank $k + 1$, then the closure operators $(\phi, \psi)^*$ and $\mathcal{L}^*$ coincide on all elements with a quasi-rank of at most $k + 1$, i.e., $|x| \leq k + 1$ implies $x^{(\phi,\psi)^*} = x^{\mathcal{L}^*}$. Furthermore, it can be proven that the following statements are satisfied:

1. If $p \in M$ is a (ϕ, ψ)-pseudo-closure, then there is neither a $\phi \curlyvee \psi$-closure nor a (ϕ, ψ)-pseudo-closure strictly between p and $p^{\phi \curlyvee \psi}$.
2. If $m \in M$ is a $\phi \curlyvee \psi$-closure, then the next $\phi \curlyvee \psi$-closures or (ϕ, ψ)-pseudo-closures are of the form $n^{(\phi,\psi)^*}$ for upper neighbors $n \succ m$.
3. If $x, y \in M$ with $x \lneq y$ are neighboring $(\phi, \psi)^*$-closures, then $y = z^{(\phi,\psi)^*}$ for all upper neighbors $z \succ x$ with $z \leq y$.

Algorithm 1. `NextClosures`

`Input:` a complete lattice $\underline{M} = (M, \leq, \bigwedge, \bigvee, \top, \perp)$
`Input:` a quasi-rank function $|\cdot| \colon (M, \leq) \to (\mathbb{N}, \leq)$
`Input:` a closure operator with constraint (ϕ, ψ) in $\underline{M}$
`Initialize:` a set $\mathcal{I} := \emptyset$
`Initialize:` an implication set $\mathcal{L} := \emptyset$
`Initialize:` a candidate set $\mathbf{C} := \{\perp\}$
`for` $k = 0, 1, \ldots, |\top|$ `do`
  `for all` $c \in \mathbf{C}$ with $|c| = k$ `do in parallel`
    `if` $c = c^{\mathcal{L}^*}$ `then`
      `if` $c \neq c^{\phi \curlyvee \psi}$ `then`
        $\mathcal{L} := \mathcal{L} \cup \{c \to c^{\phi \curlyvee \psi}\}$
      $\mathcal{I} := \mathcal{I} \cup \{c^{\phi \curlyvee \psi}\}$
      $\mathbf{C} := \mathbf{C} \cup \{\, d \mid c^{\phi \curlyvee \psi} \prec d \,\}$
    `else`
      $\mathbf{C} := \mathbf{C} \cup \{c^{\mathcal{L}^*}\}$
  Wait for termination of all parallel processes.
`Output:` the set $\mathcal{I}$ of all (ϕ, ψ)-closures
`Output:` the canonical base $\mathcal{L}$ of (ϕ, ψ)

Algorithm 1 manages a set $\mathbf{C}$ of candidates, an implication set $\mathcal{L}$, and a set $\mathcal{I}$ of closures. Initially, $\perp$ is the only candidate in $\mathbf{C}$. Algorithm 1 is *in state* k if it has processed all candidates with a quasi-rank $\leq k$, but none of quasi-rank $> k$. In state k, $\mathbf{C}_k$ denotes the set of candidates, $\mathcal{L}_k$ denotes the implication set, and $\mathcal{I}_k$ denotes the set of closures. It can be shown that the following invariants are always satisfied during Algorithm 1's run:

1. $\mathbf{C}_k$ contains all pseudo-closures of (ϕ, ψ) with quasi-rank $k+1$, and all closures of $\phi \curlyvee \psi$ with quasi-rank $k+1$ that are not already contained in $\mathcal{I}_k$.
2. $\mathcal{I}_k$ contains exactly all closures of $\phi \curlyvee \psi$ with a quasi-rank not exceeding k.
3. $\mathcal{L}_k$ contains exactly all implications $p \to p^{\phi \curlyvee \psi}$ for which the premise p is a pseudo-closure of (ϕ, ψ) with a quasi-rank of at most k.
4. Between the states k and $k+1$, the closure operators $(\phi, \psi)^*$ and $\mathcal{L}^*$ coincide on all elements with quasi-rank $k+1$, i.e., an element with quasi-rank $k+1$ is either a closure of $\phi \curlyvee \psi$ or a pseudo-closure of (ϕ, ψ) if, and only if, it is a closure of $\mathcal{L}^*$.

As a consequence, we obtain that in the final state $|\top|$, Algorithm 1 outputs the set of all closures of $\phi \curlyvee \psi$ as well as the canonical base of (ϕ, ψ).

4.1 A Generalized Parallel Attribute Exploration

As an extension, we suppose that the given closure operator only describes a part of the data-set, and furthermore there is an expert available (very much in the same way as for *Attribute Exploration* [10, 11] in the case of *Formal Concept Analysis*). The exploration of a closure operator in a lattice is explained in full detail in [18], and the special case of exploring a formal context is considered in [19] (including extensive proofs). However, we will give a short summary here.

An *expert* in a lattice $\underline{M}$ is a partial mapping $\chi\colon \mathsf{Imp}(M) \to_{\mathsf{p}} M$ such that the following conditions hold:

1. If $\chi(p \to c)$ is defined, then the value is a counterexample against $p \to c$, i.e., $\chi(p \to c) = m$ implies $p \leq m$ and $c \nleq m$.
2. If $\chi(p \to c)$ is undefined, then all counterexamples of χ are models of $p \to c$, i.e., whenever $\chi(q \to d) = m$, then $p \leq m$ implies $c \leq m$.

We say that χ *accepts* $p \to c$ if it does not provide a counterexample against $p \to c$, i.e., $\chi(p \to c)$ is undefined, and that χ *refutes* $p \to c$ otherwise. Furthermore, χ is *induced* by a closure operator ϕ if for all implications $p \to c$, $\chi(p \to c)$ is undefined if, and only if, $p \to c$ is valid in ϕ. The expert χ is *optimal* if it accepts $p \to c \wedge x$ whenever it refutes $p \to c$ with counterexample x. For an arbitrary expert χ, it is always possible to construct an optimal expert $\widehat{\chi}$ that accepts the same implications as follows: Let $p \to c$ be an implication. Then $\widehat{\chi}$ accepts $p \to c$ if χ accepts $p \to c$. Otherwise, let $c_0 \coloneqq c$ and $n \coloneqq 0$. While χ rejects c_n with counterexample x_n, set $c_{n+1} \coloneqq c_n \wedge x_n$, and increase n. Eventually, define $\widehat{\chi}(p \to c) \coloneqq \bigwedge_{k=0}^{n} x_k$.

As input the generalized *Parallel Attribute Exploration* requires a closure operator ϕ, an expert χ, as well as an implication set $\mathcal{L}$ in $\underline{M}$, such that there is an inaccessible closure operator δ describing the domain of interest where χ is induced by δ, $\delta \preceq \phi$, i.e., $\mathsf{Clo}(\phi) \subseteq \mathsf{Clo}(\delta)$, and $\mathcal{L}$ is valid in δ. W.l.o.g. we may assume that χ is optimal. The goal is to compute an implicational base for the domain closure operator δ by utilizing the background knowledge $\mathcal{L}$, and querying the expert χ.

In order to insert new closures into the closure operator ϕ, we define an operation $\downarrow\colon \mathsf{ClOp}(\underline{M}) \times M \to \mathsf{ClOp}(\underline{M})$ by $\phi\downarrow x \coloneqq \phi \curlywedge \rho_x$ where $\rho_x(m) \coloneqq x$ if $m \leq x$, and $\rho_x(m) \coloneqq \top$ otherwise. It is readily verified that $\phi\downarrow x$ is the greatest closure operator below ϕ that has x as a closure. Then $x \models y \to y^{\phi}$ implies $y^{\phi\downarrow x} = y^{\phi}$. Furthermore, if x is a model of all implications $p \to p^{\phi}$ where p is a pseudo-closure of $(\phi, \mathcal{L})$ with $|p| \leq k$, then the pseudo-closures of $(\phi, \mathcal{L})$ and $(\phi\downarrow x, \mathcal{L})$ with a quasi-rank of at most k are the same. As a corollary, we get that then also the canonical bases of $(\phi, \mathcal{L})$ and $(\phi\downarrow x, \mathcal{L})$ coincide for the implications the premises of which have a quasi-rank not exceeding k. By an inductive application, it follows that $\mathcal{B}_{\mathsf{can}}(\phi, \mathcal{L})\restriction_k = \mathcal{B}_{\mathsf{can}}((\phi\downarrow x_1, \ldots, x_n), \mathcal{L})\restriction_k$ if each x_i is a model of $\mathcal{B}_{\mathsf{can}}(\phi, \mathcal{L})\restriction_k$. This fact enables us to execute a parallel attribute exploration.

Algorithm 2. `ParallelAttributeExploration`

`Input:` a complete lattice $\underline{M} = (M, \leq, \bigwedge, \bigvee, \top, \bot)$
`Input:` a quasi-rank function $|\cdot|\colon (M, \leq) \to (\mathbb{N}, \leq)$
`Input:` a closure operator ϕ in $\underline{M}$
`Input:` an expert χ in $\underline{M}$
`Input:` a set $\mathcal{L}$ of background implications that are valid in ϕ
`Initialize:` an implication set $\mathcal{B} \coloneqq \emptyset$
`Initialize:` a candidate set $\mathbf{C} \coloneqq \{\bot\}$
for $k = 0, \ldots, |\top| - 1$ **do**
  for all $c \in \mathbf{C}$ **with** $|c| = k$ **do in parallel**
    if $c = c^{\mathcal{B}^*}$ and $c = c^{\mathcal{L}}$ **then**
      while $c \neq c^{\phi}$ **and** $\chi(c \to c^{\phi}) = x$ **do**
        $\phi \coloneqq \phi\downarrow x$
      if $c \neq c^{\phi}$ **then**
        $\mathcal{B} \coloneqq \mathcal{B} \cup \{c \to c^{\phi}\}$
      $\mathbf{C} \coloneqq \mathbf{C} \cup \{\, d \mid d \succ c^{\phi} \,\}$
    else
      $\mathbf{C} \coloneqq \mathbf{C} \cup \{c^{\mathcal{B}^* \curlyvee \mathcal{L}}\}$
  Wait for termination of all parallel processes.
`Output:` the refined closure operator ϕ
`Output:` the canonical base $\mathcal{B}$ of ϕ relative to $\mathcal{L}$

Algorithm 2 is *in state* k if it has processed all candidates of a quasi-rank not exceeding k, but none of a greater quasi-rank. Let $\mathbf{C}_k$ denote the set of candidates in state k, and let $\mathcal{B}_k$ denote the implication set in state k. Furthermore, $x_k^1, \ldots, x_k^{n_k}$ denote all counterexamples provided by the expert between states k and $k{+}1$, and ϕ_k is the closure operator in state k, i.e., $\phi_k\downarrow x_k^1, \ldots, x_k^{n_k} = \phi_{k+1}$. Then the following statements are always satisfied.

1. $\mathbf{C}_k$ contains all pseudo-closures of $(\phi_{k+1}, \mathcal{L})$ with quasi-rank $k + 1$.
2. $\mathcal{B}_k$ consists of all implications $p \to p^{\phi \curlyvee \psi}$ where p is a pseudo-closure of $(\phi_k, \mathcal{L})$ with a quasi-rank of at most k, i.e., $\mathcal{B}_{\mathsf{can}}(\phi_k, \mathcal{L})\restriction_k = \mathcal{B}_k$.
3. Between the states k and $k + 1$, every element with quasi-rank $k + 1$ is a closure of $\mathcal{B}^*$ if, and only if, it is either a closure of $\phi_{k+1} \curlyvee \mathcal{L}$ or a pseudo-closure of $(\phi_{k+1}, \mathcal{L})$.

As a corollary, we infer that in the final state $|\top|$, Algorithm 2 returns a refinement of ϕ that has the same closures as δ, and a minimal implicational base of δ relative to $\mathcal{L}$, i.e., a minimal superset of $\mathcal{L}$ that constitutes an implicational base of δ. Additionally, there is no algorithm that computes a minimal relative implicational base of δ, but poses less questions to χ than Algorithm 2.

4.2 A Problem for the Exploration of Constrained Implicational Bases

A further generalization of Algorithm 2 to arbitrary closure operators ψ instead of implication sets $\mathcal{L}$ valid in ϕ is not easily possible. We may only replace $\mathcal{L}$ by a closure operator ψ such that $\psi \preceq \phi$, otherwise when receiving counterexamples from the expert, they may not be inserted into the closure operator ϕ, but must be inserted into the supremum $\phi \curlyvee \psi$, since we cannot ensure that $y^{(\phi \downarrow x) \curlyvee \psi} = y^{\phi \curlyvee \psi}$ if x is a ψ-model of the ψ-implication $y \to y^{\phi \curlyvee \psi}$. This is due to the fact that the lattice of closure operators is not distributive, and this fact can be proven by means of [22, Lemmata 1 and 2].

5 Applications

In this section, we will present some applications of the generalized *NextClosures* algorithm. It is trivial that there is an application for *Formal Concept Analysis*, since the powerset of the attribute set constitutes a complete lattice, and the composition of the derivation operators is a closure operator. Further applications can be found for interpretations in *Description Logics*, and for *Pattern Structures*. We will shortly describe each of the cases in the following subsections.

5.1 Application: Formal Concept Analysis

For each formal context $\mathbb{K} = (G, M, I)$, the mapping $II\colon X \mapsto X^{II}$ is a closure operator in the powerset $\wp(M)$. The closure system induced by II is the set of all intents of $\mathbb{K}$, and thus it is easy to verify that an implication is valid in $\mathbb{K}$ if, and only if, it is valid in II. Hence, Algorithm 1 can be utilized to compute both the set of intents of $\mathbb{K}$ and the canonical base of $\mathbb{K}$. If furthermore a constraining closure operator C in $\wp(M)$ is given, cf. [4] as well as Section 3.3, then Algorithm 1 is also applicable, and computes all C-intents as well as a base of the C-implications.

As a special case, we may consider the case where the constraining closure operator C is given by means of an implication set $\mathcal{L} \subseteq \mathsf{Imp}(M)$, i.e., $C\colon X \mapsto X^{\mathcal{L}}$ assigns to each attribute set $X \subseteq M$ the smallest superset that is a model of $\mathcal{L}$. Then there are two cases to consider: either $\mathbb{K} \models \mathcal{L}$, or $\mathbb{K} \not\models \mathcal{L}$.

First, assume that $\mathcal{L}$ is valid in $\mathbb{K}$. Then each intent of $\mathbb{K}$ is a model of $\mathcal{L}$, i.e., $\mathcal{L} \preceq II$, and thus $II \curlywedge \mathcal{L} = \mathcal{L}$ as well as $II \curlyvee \mathcal{L} = II$. Consequently, it is uninteresting to consider the infimum or supremum. Furthermore, it can be shown that the canonical base of $\mathbb{K}$ with constraint $\mathcal{L}$ from Section 3.3 coincides with the canonical base of $\mathbb{K}$ relative to $\mathcal{L}$ from [9, 24], i.e., is a minimal extension of $\mathcal{L}$ which constitutes an implicational base for $\mathbb{K}$. However, both documents [9, 24] have not addressed the remaining case where $\mathcal{L}$ is not valid in $\mathbb{K}$.

Second, let $\mathbb{K} \not\models \mathcal{L}$. Then, an implication is valid in the infimum $II \curlywedge \mathcal{L}$ if, and only if, it is valid in $\mathbb{K}$ as well as is entailed by $\mathcal{L}$. This is useful for the case where a sequence $(\mathbb{K}_n)_{n \in \mathbb{N}}$ of formal contexts is observed, and for each time point n, a base for the subposition of $\mathbb{K}_0, \ldots, \mathbb{K}_n$ shall be constructed. Note that this has already been discussed in Section 3.1 for the general case.

For the supremum, an implication is valid if, and only if, it has as models all intents of $\mathbb{K}$ that are also models of $\mathcal{L}$. Consider the case where $\mathbb{K}$ contains faulty objects, e.g., due to wrong observations inserted to $\mathbb{K}$, and where the implication set $\mathcal{L}$ has been manually created or verified to ensure that all its implications are valid in the domain of interest. Then, constructing implications that are valid in $II \curlyvee \mathcal{L}$ is a suitable method, since only those intents of $\mathbb{K}$ are allowed as models that are already models of $\mathcal{L}$, i.e., satisfy the existing implicational knowledge. A further restriction is imposed by also requiring the premises and conclusions to be models of $\mathcal{L}$, i.e., only considering implications that are $\mathcal{L}$-constrained. For both cases, Algorithm 1 can be used to compute a canonical base in parallel. However, it remains to investigate whether the implications valid in $II \curlyvee \mathcal{L}$, or the $\mathcal{L}$-constrained implications of $\mathbb{K}$, are more useful in practise.

5.2 Application: Pattern Structures

A *pattern structure*, as introduced by Ganter and Kuznetsov in [12], is a triple $(G, (M, \sqcap), \delta)$ consisting of a set G of *objects*, a semi-lattice $(M, \sqcap)$ of *patterns*, and a *description function* $\delta \colon G \to M$, such that the set $\{\, \delta(g) | g \in G \,\}$ induces a complete sub-semi-lattice $(M_\delta, \sqcap)$ of $(M, \sqcap)$. Furthermore, then a galois connection between the powerset $\wp(G)$ and the pattern lattice $(M, \sqsubseteq)$ (where $x \sqsubseteq y$ iff $x \sqcap y = x$) is given as follows:

$$A^\square := \bigsqcap \{\, \delta(g) \mid g \in A \,\} \quad \text{for all } A \subseteq G,$$
$$d^\square := \{\, g \in G \mid \delta(g) \sqsubseteq d \,\} \text{ for all } d \in M.$$

A *pattern implication* is a term $x \to y$ where $x, y \in M$, and is valid in $(G, (M, \sqcap), \delta)$ if $x^\square \subseteq y^\square$. Since $(\cdot^\square, \cdot^\square)$ is a galois connection, it follows that the composition $\cdot^{\square\square}$ is a closure operator in the lattice of all descriptions. Hence, it is possible to apply the generalized *NextClosures* algorithm for the case of pattern structures, too, provided that there is a quasi-rank function $|\cdot| \colon (M, \sqsubseteq) \to (\mathbb{N}, \leq)$.

5.3 Application: Description Logics

As a special case of pattern structures, we consider interpretations in *Description Logics* [2] as input data-sets. First, fix a finite signature (N_C, N_R), i.e., N_C is a set of *concept names*, and N_R is a set of *role names*. An *interpretation* is a pair $\mathcal{I} = (\Delta^\mathcal{I}, \cdot^\mathcal{I})$ consisting of a non-empty set $\Delta^\mathcal{I}$, called *domain*, and an *extension function* $\cdot^\mathcal{I}$ such that $A^\mathcal{I} \subseteq \Delta^\mathcal{I}$ for all concept names $A \in N_C$, and $r^\mathcal{I} \subseteq \Delta^\mathcal{I} \times \Delta^\mathcal{I}$ for all role names $r \in N_R$. An $\mathcal{EL}^\bot$*-concept description* may be built according to the following inductive rule:

$$C ::= \bot \mid \top \mid A \mid C \sqcap C \mid \exists r.\, C.$$

Then, the extension function $\cdot^\mathcal{I}$ is extended to all $\mathcal{EL}^\bot$-concept descriptions by means of the following definitions:

$$\begin{aligned} \bot^\mathcal{I} &:= \emptyset, \\ \top^\mathcal{I} &:= \Delta^\mathcal{I}, \\ (C \sqcap D)^\mathcal{I} &:= C^\mathcal{I} \cap D^\mathcal{I}, \\ \text{and} \quad (\exists r.\, C)^\mathcal{I} &:= \{\, d \in \Delta^\mathcal{I} \mid \exists e \in \Delta^\mathcal{I} \colon (d, e) \in r^\mathcal{I} \text{ and } e \in C^\mathcal{I} \,\}. \end{aligned}$$

Please note that an implication between concept descriptions is rather called a *general concept inclusion*. An implicational base for the closure operator $\cdot^{\mathcal{I}\mathcal{I}}$, where the first $\cdot^{\mathcal{I}}$ is the extension mapping, and the second $\cdot^{\mathcal{I}}$ is the model-based most-specific concept description mapping, is a base of GCIs for $\mathcal{I}$, as it has been investigated by Baader and Distel in [1, 8].

The set of all $\mathcal{EL}^{\bot}$-concept descriptions constitutes a bounded lattice. First we define a quasi-order $\sqsubseteq$, the *subsumption order*, by $C \sqsubseteq D$ if $C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$ for all interpretations $\mathcal{I}$. The corresponding equivalence relation is denoted as $\equiv$. For simplicity, we do not distinguish between equivalence classes and their representatives. The set of all equivalence classes is a bounded lattice with the smallest element $\bot$, the greatest element $\top$, and where the *conjunction* $\sqcap$ is the infimum operation, and the *least common subsumer mapping* $\sqcup$ is the supremum operation. It is easy to verify that both $\sqcap$ and $\sqcup$ are compatible with $\equiv$, hence they are well-defined for the equivalence classes.

If we do not restrict the role-depth, and use descriptive semantics (as defined above) instead of greatest-fixpoint semantics, then however the lattice of concept descriptions is not complete, as there are infinitely many mutually distinct $\mathcal{EL}^{\bot}$-concept descriptions, e.g., the concept descriptions $(\exists\, r.)^n A$ for $n \in \mathbb{N}$, and conjunction is only allowed for a finite number of concept descriptions. It turns out, that this is no restriction for the application of Algorithm 1, since for finite interpretations (i.e., with a finite domain), the canonical base is finite, and hence for the computation of closures w.r.t. the current implication set in Algorithm 1 only finitary suprema are necessary.

For defining a quasi-rank function, we may first observe that the dual lattice of $\mathcal{EL}^{\bot}$-concept descriptions is well-founded, i.e., there are no infinite strictly ascending chains $C_0 \sqsubsetneq C_1 \sqsubsetneq C_2 \sqsubsetneq \ldots$, cf. Baader and Morawska in [3, Proposition 3.5]. Hence, a quasi-rank function could be defined by $|C| \coloneqq n$ where n is the maximal length of a chain starting at C and ending at $\top$. Since each concept description has only finitely many subsumers (for a finite signature), there are only finitely many mutually distinct chains in the interval $[C, \top]$, and thus $|\cdot|$ as above is well-defined. For the same reason, the set of upper neighbors of an arbitrary concept description is computable.

6 Conclusion

In this document, the *NextClosures* algorithm has been generalized to arbitrary closure operators in complete lattices, and has been extended to handle constraints that are given in form of another closure operator. Furthermore, some exemplary applications have been presented, e.g., in the field of *Formal Concept Analysis*, *Description Logics*, and for *Pattern Structures*. While some experiments of an implementation specialized to formal contexts as input structures have shown that there is an almost inverse linear correlation between the computation time and the number of available processor cores, experiments for other types of input structures are outstanding. A complexity analysis taking into account the complexities of the lattice operations as well as of the closure operator could be interesting.

Acknowledgements The author gratefully thanks the anonymous reviewers for their constructive hints and helpful remarks.

References

[1] Franz Baader and Felix Distel. "A Finite Basis for the Set of EL-Implications Holding in a Finite Model". In: *Formal Concept Analysis, 6th International Conference, ICFCA 2008, Montreal, Canada, February 25-28, 2008, Proceedings*. 2008, pp. 46–61.

[2] Franz Baader and Carsten Lutz. "Description Logic". In: *Handbook of Modal Logic*. Ed. by Patrick Blackburn, Johan van Benthem, and Frank Wolter. Elsevier, 2006. Chap. 13.

[3] Franz Baader and Barbara Morawska. "Unification in the Description Logic EL". In: *Logical Methods in Computer Science* 6.3 (2010).

[4] Radim Belohlávek and Vilém Vychodil. "Formal Concept Analysis with Constraints by Closure Operators". In: *Conceptual Structures: Inspiration and Application, 14th International Conference on Conceptual Structures, ICCS 2006, Aalborg, Denmark, July 16-21, 2006, Proceedings*. 2006, pp. 131–143.

[5] Garrett Birkhoff. "Lattice theory". In: *Colloquium Publications*. 3rd ed. Vol. 25. Amer. Math. Soc., 1967.

[6] Nathalie Caspard and Bernard Monjardet. "The Lattices of Closure Systems, Closure Operators, and Implicational Systems on a Finite Set: A Survey". In: *Discrete Applied Mathematics* 127.2 (2003), pp. 241–269.

[7] Brian A. Davey and Hilary Ann Priestley. *Lattices and Order*. 2nd ed. Cambridge University Press, Apr. 2002.

[8] Felix Distel. "Learning description logic knowledge bases from data using methods from formal concept analysis". PhD thesis. Dresden University of Technology, 2011.

[9] Bernhard Ganter. "Attribute Exploration with Background Knowledge". In: *Theor. Comput. Sci.* 217.2 (1999), pp. 215–233.

[10] Bernhard Ganter. *Two Basic Algorithms in Concept Analysis*. FB4-Preprint 831. Darmstadt, Germany: Technische Hochschule Darmstadt, 1984.

[11] Bernhard Ganter. "Two Basic Algorithms in Concept Analysis". In: *Formal Concept Analysis, 8th International Conference, ICFCA 2010, Agadir, Morocco, March 15-18, 2010. Proceedings*. 2010, pp. 312–340.

[12] Bernhard Ganter and Sergei O. Kuznetsov. "Pattern Structures and Their Projections". In: *Conceptual Structures: Broadening the Base, 9th International Conference on Conceptual Structures, ICCS 2001, Stanford, CA, USA, July 30-August 3, 2001, Proceedings*. 2001, pp. 129–142.

[13] Bernhard Ganter and Rudolf Wille. *Formal Concept Analysis: Mathematical Foundations*. 1st ed. Springer, Dec. 1999.

[14] George Grätzer. *General Lattice Theory*. 2nd ed. Birkhäuser Verlag, 1998.

[15] Jean-Luc Guigues and Vincent Duquenne. "Famille minimale d'implications informatives résultant d'un tableau de données binaires". In: *Mathématiques et Sciences Humaines* 95 (1986), pp. 5–18.

[16] Akira Higuchi. "Lattices of closure operators". In: *Discrete Mathematics* 179.1-3 (1998), pp. 267–272.

[17] Francesco Kriegel. *Concept Explorer FX*. Software for Formal Concept Analysis, available at `https://github.com/francesco-kriegel/conexp-fx`. 2010–2016.

[18] Francesco Kriegel. *NextClosures – Parallel Exploration of Constrained Closure Operators*. LTCS-Report 15-01. Dresden, Germany: Chair of Automata Theory, Institute of Theoretical Computer Science, Technische Universität Dresden, 2015.

[19] Francesco Kriegel. "Parallel Attribute Exploration". In: *Graph-based Modeling of Conceptual Structures - 22nd International Conference on Conceptual Structures, ICCS 2016, Annecy, France, July 5-7, 2016, Proceedings*. 2016.

[20] Francesco Kriegel and Daniel Borchmann. "NextClosures: Parallel Computation of the Canonical Base". In: *Proceedings of the 12th International Conference on Concept*

Lattices and Their Applications (CLA 2015), Clermont-Ferrand, France, October 13-16, 2015. 2015, pp. 181–192.

[21] David Maier. *The Theory of Relational Databases*. Computer Science Press, 1983.

[22] José Morgado. "Note on the distributive closure operators of a complete lattice". In: *Portugaliae Mathematica* 23 (1964), pp. 11–25.

[23] Sebastian Rudolph. "On the Succinctness of Closure Operator Representations". In: *Formal Concept Analysis - 12th International Conference, ICFCA 2014, Cluj-Napoca, Romania, June 10-13, 2014. Proceedings*. 2014, pp. 15–36.

[24] Gerd Stumme. "Attribute Exploration with Background Implications and Exceptions". In: *Data Analysis and Information Systems*. Ed. by Hans-Hermann Bock and Wolfgang Polasek. Studies in Classification, Data Analysis, and Knowledge Organization. Springer, 1996, pp. 457–469.

[25] Marcel Wild. "A Theory of Finite Closure Spaces Based on Implications". In: *Advances in Mathematics* 108.1 (1994), pp. 118 –139.

On Stability of Triadic Concepts

Sergei O. Kuznetsov[1] and Tatiana P. Makhalova[1,2]

[1]National Research University Higher School of Economics, Kochnovsky pr. 3, Moscow 125319, Russia
[2]ISIMA, Complexe scientifique des Cézeaux, 63177 Aubière Cedex, France

skuznetsov@hse.ru,t.makhalova@gmail.com

Abstract. Triadic formal concept analysis has become a popular research direction, since triadic relations give natural models of many data collections. In this paper we address the problem of selecting most interesting concepts by proposing triadic stability indices.

1 Introduction

Triadic formal concept analysis (3FCA) was introduced by Rudolf Wille and Fritz Lehmann [1] to model hierarchies of classes and dependencies arising from ternary relations. Recently, several algorithms for computing frequent triconcepts were proposed [2, 3]. It is noticed that some infrequent concepts are still interesting, since they represent extraordinary or uncommon data. In this paper we propose triadic stability for selecting interesting triadic concepts. Together with exact stability indices we suggest their efficient approximations analogous to Δ-stability introduced in [4].

2 Main definitions

2.1 Formal Concept Analysis

First, we briefly recall some basic definitions of the Formal Concept Analysis (FCA) [5]. A formal context is a triple (G, M, I). G and M are sets of objects and attributes respectively, and I is an incidence relation. It is defined as the Cartesian product $G \times M$, i.e. $(g, m) \in I$ if the object $g \in G$ has the attribute $m \in M$. The derivation operators $(\cdot)'$ are defined for $A \subseteq G$ and $B \subseteq M$ as follows:

$$A' = \{m \in M \mid \forall g \in A : gIm\}$$
$$B' = \{g \in G \mid \forall m \in B : gIm\}$$

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. Subsets $A \subseteq G$, $B \subseteq M$ such that $A = A''$ and $B = B''$ are called *closed*.

© Marianne Huchard, Sergei O. Kuznetsov (Eds.): CLA 2016, pp. 245–256, ISBN 978-5-600-01454-1, National Research University Higher School of Economics, 2016.

A (formal) concept is a pair (A, B), where $A \subseteq G$, $B \subseteq M$ and $A' = B$, $B' = A$. A is called the (formal) extent, and B is called the (formal) intent of the concept (A, B).

A concept lattice (or Galois lattice) is a partial ordered set of concepts, the order $\leqslant$ on the set of concepts is defined as follows: $(A, B) \leq (C, D)$ iff $A \subseteq C\,(D \subseteq B)$, a pair (A, B) is a subconcept of (C, D), while (C, D) is a superconcept of (A, B). Each finite lattice has the highest element with $A = G$, called the top element, and the lowest element with $B = M$, called the bottom element.

2.2 Triadic Concept Analysis

In the case of a triadic relation one deals with a quadruple (G, M, B, Y), called a triadic context. G, M, B are sets and Y is a ternary relation between G, M and B, i.e. $Y \subseteq G \times M \times B$; the elements of G, M and B are called objects, attributes and conditions respectively, and $(g, m, b) \in Y$ is read: object g has attribute m under condition b.

The dyadic derivation operators can be used to construct triadic concepts. A triadic context can be represented as follows: $\mathbb{K} := (K_1, K_2, K_3, Y)$, where K_1 is a set of objects G, K_2 is a set of attributes and K_3 is a set of conditions, and each element of K_i may be seen as an instance of Peirce's i-th category [1]. For every triadic context one defines the following dyadic contexts:

$\mathbb{K}^1 := \left(K_1, K_2 \times K_3, Y^{(1)}\right)$ with $gY^{(1)}\,(m, b) :\Leftrightarrow (g, m, b) \in Y$

$\mathbb{K}^2 := \left(K_2, K_1 \times K_3, Y^{(2)}\right)$ with $mY^{(2)}\,(g, b) :\Leftrightarrow (g, m, b) \in Y$

$\mathbb{K}^3 := \left(K_3, K_1 \times K_2, Y^{(3)}\right)$ with $bY^{(3)}\,(g, m) :\Leftrightarrow (g, m, b) \in Y$

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

Put differently, the context $\mathbb{K}^{(i)}$ is a flattened representation of the original triadic context, while $\mathbb{K}^{(i,j)}_{A_k}$ corresponds to the relation between elements of K_i and K_j that belong to A_k.

(i)-derivation operator 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

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

$$Z \longmapsto Z^{(i)} := \{a_i \in K_i \mid (a_i, a_j, a_k) \in Y \text{ for all}\,(a_j, a_k) \in Z\}$$

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

$$X_i \longmapsto X_i^{(i,j,A_k)} := \{a_j \in K_j \mid (a_i, a_j, a_k) \in Y \text{ for all}\,(a_i, a_k) \in X_i \times A_k\}$$

$$X_j \longmapsto X_j^{(i,j,A_k)} := \{a_i \in K_i \mid (a_i, a_j, a_k) \in Y \text{ for all}\,(a_j, a_k) \in X_j \times A_k\}$$

A triadic concept (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 $\mathfrak{T}(K)$ denote the set of all triadic concepts of $\mathbb{K}$.

A triadic concept lattice has three maximal elements, namely $((K_2 \times K_3)^{(1)}, K_2, K_3)$, $(K_1, (K_1 \times K_3)^{(2)}, K_3)$, and $(K_1, K_2, (K_1 \times K_2)^{(3)})$. For any two elements of a lattice one defines tree types of set inclusion/exclusion relations, which satisfy the following antiordinal dependencies: $(A_1, A_2, A_3) \preceq_G (B_1, B_2, B_3)$ iff $A_1 \subseteq B_1, A_2 \supseteq B_2, A_3 \supseteq B_3$, $(A_1, A_2, A_3) \preceq_M (B_1, B_2, B_3)$ iff $A_1 \supseteq B_1, A_2 \subseteq B_2, A_3 \supseteq B_3$ or $(A_1, A_2, A_3) \preceq_C (B_1, B_2, B_3)$ iff $A_1 \supseteq B_1, A_2 \supseteq B_2, A_3 \subseteq B_3$.

3 Stability Indices For Triadic Concepts

Stability indices for formal concepts were introduced in [6, 7] and modified in [8]. We define stability indices for the triadic case in a similar way. We describe two types of stability that correspond to the derivation operators defined above.

(i)-stability For a triadic concept (A_1, A_2, A_3) the (i)-stability is defined by:

$$Stab^{(i)}(A_1, A_2, A_3) := \frac{|\left\{X \subseteq A_i | X^{(i)} = (A_j \times A_k)\right\}|}{2^{|A_i|}}$$

This index shows how much the binary relation on sets X_j and X_k is dependent on particular elements of a subset A_i.

(i, j, X_k)-stability For a triadic concept (A_1, A_2, A_3) the (i, j, X_k)-stability is defined by:

$$Stab^{(i,j,X_k)}(A_1, A_2, A_3) := \frac{|\left\{X \subseteq (A_i \times A_j) | X^{(k)} = A_k\right\}|}{2^{|A_i|+|A_j|}}$$

The (i, j, X_k)-stability allows us to estimate the dependence of a subset A_k on elements of the (X_i, X_j)-relation.

Example Below we consider a small examples of computing stability indices for a concept. The formal context is given in the table 1.

Table 1. Triadic context

	α				β				γ				δ			
	a	b	c	d	a	b	c	d	a	b	c	d	a	b	c	d
1	×				×			×				×				
2	×				×	×	×			×	×			×	×	
3			×	×	×	×	×			×	×					
4								×								

Let us consider a triconcept $C = (\{2,3\}, \{b,c\}, \{\beta,\gamma\})$ with (1) - stability and $(1,3,X_2)$-stability.

$Stab^{(1)}(C) = \frac{1}{2}$. Since the numerator is comprised by $\{3\}^{(1)} = (\{b,c\} \times \{\beta,\gamma\})$ and $\{2,3\}^{(1)} = (\{b,c\} \times \{\beta,\gamma\})$.

Table 2. $I \subseteq M \times C$ corresponding to all possible subsets of the extent $\{2,3\}$

$\{\emptyset\}$	a	b	c	d	$\{2\}$	a	b	c	d	$\{3\}$	a	b	c	d	$\{2,3\}$	a	b	c	d
α	×	×	×	×	α	×				α			×	×	α				
β	×	×	×	×	β	×	×	×		β	×	×	×		β	×	×	×	
γ	×	×	×	×	γ		×	×		γ		×	×		γ		×	×	
δ	×	×	×	×	δ		×	×		δ					δ				

$Stab^{(1,3,X_2)}(C) = \frac{3}{8}$. To compute this value one needs to check 16 subsets of $X_1 \times X_3$ and corresponding subsets of X_2. The following sets occur in the numerator: $\{\emptyset, 2, 3, 23\} \times \{\emptyset, \beta, \gamma, \beta\gamma\}$.

$$\{b,c\} = (\{2,3\}, \{\beta,\gamma\})^{(1,3,A_2)} = (\{2,3\}, \{\gamma\})^{(1,3,A_2)} = (\{3\}, \{\gamma\})^{(1,3,A_2)}$$
$$(\{3\}, \{\beta,\gamma\})^{(1,3,A_2)} = (\{2\}, \{\gamma\})^{(1,3,A_2)} = (\{2\}, \{\beta,\gamma\})^{(1,3,A_2)}$$

4 Estimates of stability

The problem of computing stability is $\#P$-complete [6, 7], therefore, in practice, when one deals with a big context and with the huge amount of generated concepts, it is very difficult to apply these indices. That's why, estimates of the stability for dyadic concepts have been proposed [9, 4].

We have expanded the Δ-stability [4] for the case of triadic stability indices. In this regard, it is important to note that the estimates derived from the direct descendants of a triconcept can be useless owing to the defined quasiorders, because the number of direct neighbors is usually small. In figure 1 the distributions of the descendants number with respect to different inclusion/exclusion relations are represented.

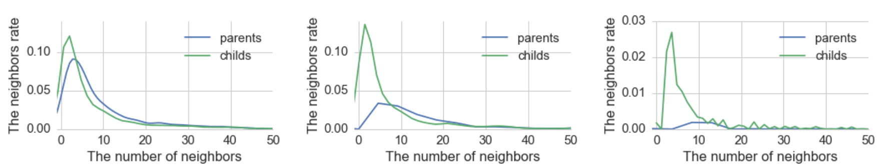


Fig. 1. The number of neighbors distributions

Instead of considering the set difference between (i)-th components of a triconcept and each direct descendant, we consider the set difference between (i)-th components of a triconcept $c = (A_1, A_2, A_3)$ and other, possibly, unclosed concepts derived by adding new elements from $K_j \setminus A_j$, $j \neq i$ or $K_j \times K_k \setminus A_j \times A_k$, $j, k \neq i$. Put differently, the lower and upper bounds estimates of stability index look as follows:

$$-log_2 \sum_{d \in Enl(c)} 2^{-\Delta(c,d)} \leq LStab\,(c) \leq \Delta_{min}\,(c)\,,$$

where $\Delta_{min}\,(c) = min_{d \in Enl(c)} \Delta\,(c, d)$,

$$Enl\,(c) = \left\{ X \mid X = \{A_k \cup x\}\,, x \in K_k \setminus A_k, X^{(k)} \subseteq (A_i \times A_j) \right\}$$

and $\Delta\,(c, d)$ is the difference between $|A_j| \cdot |A_k|$ and the number of elements in $X^{(k)}$ for estimates of (i, j, X_k)-stability.

$$Enl\,(c) = \left\{ X \mid X = \{A_j \times A_k \cup x\}\,, x \in K_j \times K_k \setminus A_j \times A_k, X^{(i)} \subseteq A_i \right\}$$

and $\Delta\,(c, d)$ is the difference between $|A_i|$ and the number of elements in $X^{(i)}$ for estimates of (i)-stability.

Example. Consider upper and lower bounds of stability estimates for $C = (\{2, 3\}, \{b, c\}, \{\beta, \gamma\})$ from the running example (Table 1). To get an estimate of the (1)-stability we consider elements of the following set $\{a, b, c, d\} \times \{\alpha, \beta, \gamma, \delta\} \setminus \{b, c\} \times \{\beta, \gamma\}$. Subsets of A_1 derived from those elements are $\{\emptyset\}$ and $\{2\}$, which give us $-log_2(7 \cdot 2^{-2} + 2^{-1})$ and 1 for lower and upper bounds, respectively. To get estimates of $(1, 3, X_2)$-stability one needs to expand the intent by elements from $\{a, d\}$. Adding the first element a reduces the $\{2, 3\} \times \{\beta, \gamma\}$ to $\{2, 3\} \times \{\beta\}$, while expanding the intent by d results in the empty set. Thus, the lower and upper bounds take values 1.678 and 2, respectively.

5 Experimental Results

In this section we explore some empirical properties of the introduced indices using synthetic data. We generated four groups of 100 random $10 \times 10 \times 10$ contexts with densities 0.1, 0.2, 0.4, 0.6. The features of the data are given in Figure 2.

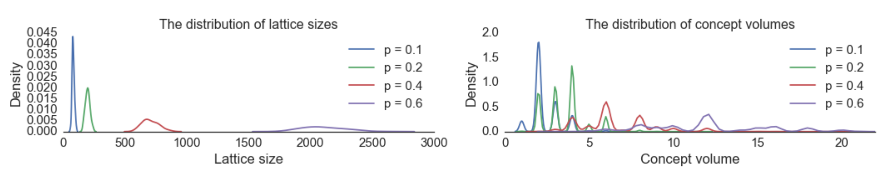


Fig. 2. Parameters of lattices constructed on $10 \times 10 \times 10$ contexts.

The choice of a subset of indices for data exploration can be motivated by the following properties: the indices should be pairwise uncorrelated (to avoid biased results when combining indices) and efficiently computable (if possible). The density function of an index may be a multimodal mixture of two or more distributions. In this case one needs a special justification for the choice of a threshold value separating two distributions.

We consider Pearson's correlation between all pairs of stability indices and cardinalities of sets that comprise a triadic concept (extent, intent, modus). In Figure 3 the values of the coefficient are represented. The sizes of the extent, intent and modus are denoted by $|A_1|$, $|A_2|$, $|A_3|$, respectively. The sizes of dyadic subcontexts are denoted in a similar way. The corresponding stability estimates are referred to by the *log* prefix.

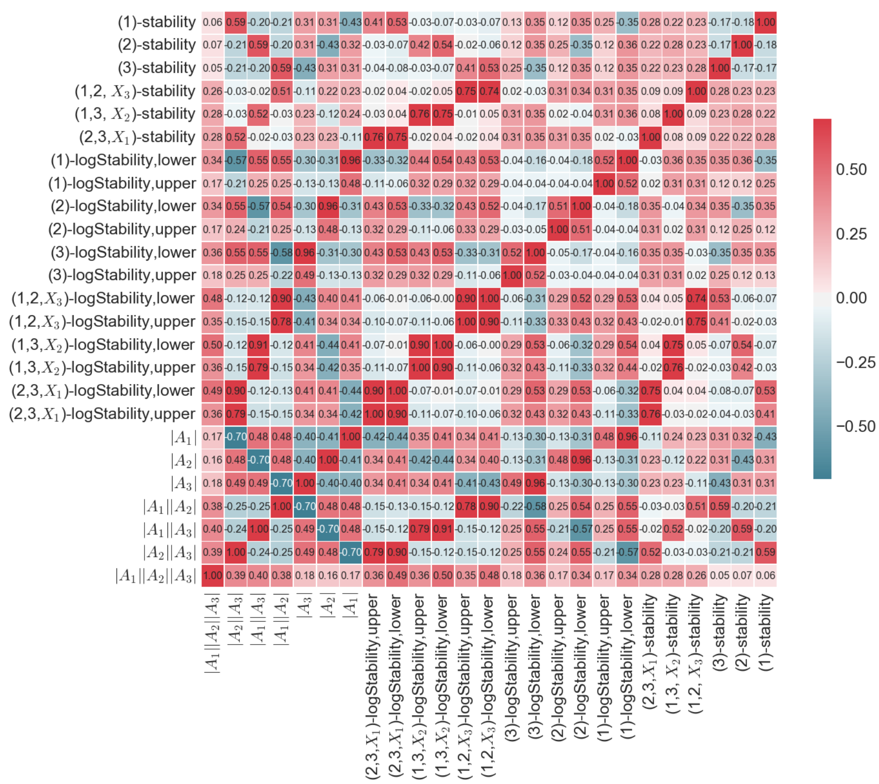


Fig. 3. The Pearson's correlation coefficient among 100 contexts with the density 0.4

As can be seen from Figure 3, there is a correlation between (i)-stability and $|A_j| \cdot |A_k|$. The index of (i)-stability correlates less strongly with the estimates of (j, k, X_i)-stability and the size of the set A_i. These types of correlation become stronger as the density of a context increases. In fact, these indices can be replaced by the size of a particular set in the case of a dense context. A correlation is observed between (i, j, X_k)-stability and its estimates, a less strong correlation is observed between (i, j, X_k)-stability and $|A_i| \cdot |A_j|$. It is important to note that the strong correlation between (i, j, X_k)-stability and its estimates, as well as very small correlation between (i, j, X_k)-stability and estimates of (k)-stability remains the same with different context densities. The pairwise correlation between stability indices is weak, hence it is preferable to use these indices together.

For selecting interesting concepts based on values of an index it is important to choose a correct threshold. This choice can be based on the distribution of index values. Figure 4 shows that the distribution of values (2)-stability and $(1, 3, X_k)$-stability (other (i)-stabilities and (i, j, X_k)-stabilities have similar distributions). The distribution of (i)-stability values allows us to identify a threshold easily, since some picks exist in the distribution, while for (i, j, X_k)-stability the distribution varies from density to density, in case of a dense context it motivates further study of the index and the way one selects thresholds for it. For values of the lower bound of stability estimates the modes of the distribution become less distinct or the distribution becomes unimodal (Figures 5,6). The upper bound for (i)-stability estimate (or (i, j, X_k)) in most cases corresponds to $|A_i|$ (or $|A_i| \cdot |A_j|$), since the closure of a superset of $A_j \times A_k$ (or A_k) results in the empty set.

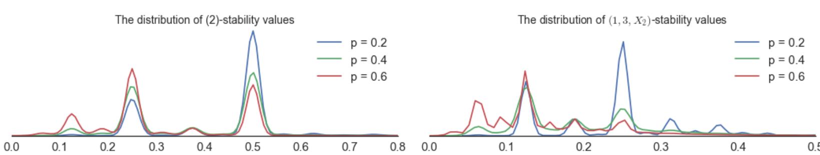


Fig. 4. The distribution of values of (2)-stability and $(1, 3, X_2)$-stability

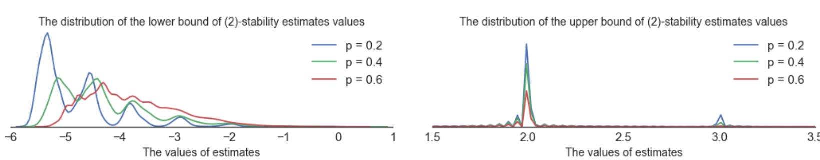


Fig. 5. The values distribution of (2)-stability estimates

The lower bound of (i)-stability is also strongly correlated with the size of set i. This is due to the fact that a big size of the set i leads to larger difference between the sizes of A_i and $(X_j \times X_k)^{(i)}$, where $X_j \times X_k$ is a superset of $A_j \times A_k$, and the sum under logarithm. The estimate of (i, j, X_k)-stability are correlated

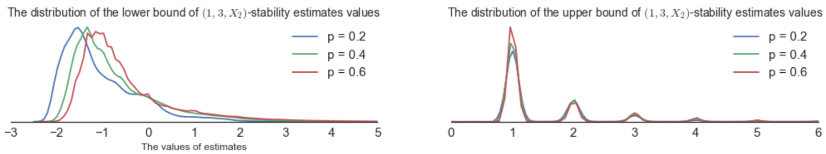


Fig. 6. The values distribution of (i, j, X_k)-stability estimates

with the corresponding indices. There is roughly the same correlation between the estimate and the value $|A_i| \cdot |A_j|$, which results from the bigger difference between $|A_i| \cdot |A_j|$ and $|X_i| \cdot |X_j|$, which correspond to a superset of A_k. The upper and lower estimates of (i, j, X_k)-stability also correlate, in this case, the correlation could be related to set-difference between the set $A_i \times A_j$ and the volume of the rectangular subarea of $X_i \times X_k$ for the corresponding superset of A_k.

It is noteworthy that the calculation of stability estimates in practice could take more time then the stability calculation itself. It is typical for (i)-stability, where the number $2^{|A_i|}$ is lower then the number of all possible subsets obtained by adding elements from $K_j \times K_k \setminus A_j \times A_k$.

6 Conclusion

In this paper we have introduced two stability indices for triadic concepts, based on two derivation operators, and studied their empirical behavior. We have proposed to compute stability using two derivation operators. We have studied correlation of stability indices and their distributions, which is important in practical data analysis. As it was shown, the introduced stability indices are not pairwise correlated and therefore can be used in some combinations for selecting interesting concepts. Moreover, (i)-stability correlates with $|A_i|$(for dense contexts) and $|A_j||A_k|$, and hence these indices should not be combined together.

The values of (i)-stability for all concepts are characterized by the presence of groups of values with high frequency, which facilitates selection of interesting concepts based on threshold values, while the distribution of (i, j, X_k)-stability does not give clearly defined groups of interesting concepts.

We have also introduced the estimates of stability indices, which correlate both with the corresponding stability indices and some of stability estimates. This is due to the fact that the estimates of (i)-stability (or (i, j, X_k)-stability) are based on the elements from $K_j \times K_k \setminus A_j \times A_k$ (or $K_k \setminus A_k$). Hence, the choice between stability and its estimates must be guided by the comparison of the sizes of sets involved in calculation, e.g. in the case of (i)-stability the number of subsets $2^{|A_i|}$, most probably, will be less then the number of elements in $K_j \times K_k \setminus A_j \times A_k$.

The proposed indices characterize triconcepts differently, in general they do not agree in the top-n selected concepts, which allow us to use either their combination to set up the strictest selection criteria, or to take some of them depending on the meaning behind a stability index.

References

1. Lehmann, F., Wille, R.: A triadic approach to formal concept analysis. In: Conceptual Structures: Applications, Implementation and Theory: Third International Conference on Conceptual Structures, ICCS '95 Santa Cruz, CA, USA, August 14–18, 1995 Proceedings. Springer Berlin Heidelberg, Berlin, Heidelberg (1995) 32–43
2. Jäschke, R., Hotho, A., Schmitz, C., Ganter, B., Stumme, G.: Trias–an algorithm for mining iceberg tri-lattices. In: Proceedings of the Sixth International Conference on Data Mining. ICDM '06, Washington, DC, USA, IEEE Computer Society (2006) 907–911
3. Ji, L., Tan, K.L., Tung, A.K.H.: Mining frequent closed cubes in 3d datasets. In: Proceedings of the 32Nd International Conference on Very Large Data Bases. VLDB '06, VLDB Endowment (2006) 811–822
4. Buzmakov, A., Kuznetsov, S.O., Napoli, A.: Scalable estimates of concept stability. In Glodeanu, C., Kaytoue, M., Sacarea, C., eds.: Formal Concept Analysis. Lecture Notes in Computer Science, Springer International Publishing (ICFCA, 2014) 157–172
5. Ganter, B., Wille, R.: Contextual attribute logic. In Tepfenhart, W., Cyre, W., eds.: Conceptual Structures: Standards and Practices. Volume 1640 of Lecture Notes in Computer Science. Springer Berlin Heidelberg (1999) 377–388
6. Kuznetsov, S.O.: Stability as an estimate of degree of substantiation of hypotheses derived on the basis of operational similarity. Nauchn. Tekh. Inf., Ser. 2 (12) (1990) 21–29
7. Kuznetsov, S.O.: On stability of a formal concept. Annals of Mathematics and Artificial Intelligence **49**(1-4) (2007) 101–115
8. Kuznetsov, S.O., Obiedkov, S., Roth, C.: Reducing the representation complexity of lattice-based taxonomies. In: Conceptual Structures: Knowledge Architectures for Smart Applications. Springer Berlin Heidelberg (2007) 241–254
9. Babin, M.A., Kuznetsov, S.O.: Approximating concept stability. In Domenach, F., Ignatov, D., Poelmans, J., eds.: Formal Concept Analysis. Volume 7278 of Lecture Notes in Computer Science., Springer Berlin Heidelberg (2012) 7–15

A Fresh View on Fuzzy FCA and Mathematical Morphology

Tim B. Kaiser[1], Lars Lumpe[2], and Stefan E. Schmidt[2]

[1] SAP AG, Walldorf
[2] Institut für Algebra, Technische Universität Dresden
tbkaiser@gmx.de,larslumpe@gmail.com, midt1@msn.com

Abstract. In this paper we discuss how biresiduations provide a unifying paradigm for fuzzy formal concept analysis and mathematical morphology. In particular we provide constructions of morphological operators such as dilation and erosion within the framework of convolution algebras of action networks (also known as small covariant categories) over join complete semirings. This unifies and generalizes previous contributions in mathematical morphology, such as the work of Isabelle Bloch. Further we show how decomposition, factorization, and hedges naturally align themselves in the framework of fuzzy formal concept analysis.

Keywords: biresiduation, fuzzy formal concept analysis, hedges, factorization, decomposition, linear algebra, complete monoids

1 Introduction

In our paper we outline the fundamental role of biresiduation in combination with biadditivity for a common view on fuzzy FCA and mathematical morphology. While the concept of biresiduation is an important paradigm of algebraic logic, the concept of biadditivity is rooted in linear algebra (over complete monoids and complete semirings). The close relationship between both approaches is reflected in the bijective correspondence of the residuated complete lattices and the join complete semirings on a fixed set. Our considerations yield a major application for a better understanding of mathematical morphology: We achieve this by constructing and investigating convolution algebras of action networks over join complete semirings. Finally, as a tool for information reduction, we also discuss the role of hedges on residuated complete lattices and their relationship with substructures of join complete semirings, and more generally within the category of biadditive setups.
Important literature for (fuzzy) FCA is given by [1,2,3,4,5], regarding factor analysis, especially by [6], regarding hedges [7]. Belohlávek gives an overview on approaches to fuzzy concept analysis in [8] which is an update of [9].
We assume the reader to be familiar with ordered sets and formal concept analysis as exposed in [5] and [1]. Profound information on residuation theory can be found, for instance, in [10].

Over the last two decades, major contributions to mathematical morphology go back to Isabelle Bloch [11,12,13].
This paper is based on [14].

2 Biresiduation

One of the key concepts for accessing FCA and its generalizations is that of a biresiduated map. We start by recalling the definition of a residuated map.

Definition 1 (residuated map, adjunction)**.** Let $\mathbb{P}_1 := (P_1, \leq)$ and $\mathbb{P}_2 := (P_2, \leq)$ be ordered sets. Then a map $f : P_1 \to P_2$ is *residuated* from $\mathbb{P}_1$ to $\mathbb{P}_2$ if there exists a map $f^+ : P_2 \to P_1$ such that

$$fp_1 \leq p_2 \iff p_1 \leq f^+(p_2).$$

Here, the map f^+ is uniquely determined by f and is called the residual of f. The pair (f, f^+) is called an *adjunction* w.r.t. $(\mathbb{P}_1, \mathbb{P}_2)$. In case $\mathbb{P}_1 = \mathbb{P}_2$, we will say that (f, f^+) forms an adjunction on $\mathbb{P}_1$.

Remark 1. A map between complete lattices is residuated if and only if it is *completely join preserving*, that is, $f(\sup X) = \sup(fX)$ holds for all $X \subseteq P_1$ in the above setting.

The following definition is central for this paper.

Definition 2 (biresiduation)**.** Let $\mathbb{P}_1 := (P_1, \leq)$, $\mathbb{P}_2 := (P_2, \leq)$, and $\mathbb{P} := (P, \leq)$ be ordered sets. Then a map

$$\otimes : P_1 \times P_2 \to P$$

will be called a *biresiduation* w.r.t. $\mathfrak{P} := (\mathbb{P}_1, \mathbb{P}_2, \mathbb{P})$ if there exist maps

$$\overset{\otimes}{\rightarrow} : P_1 \times P \to P_2$$
$$\overset{\otimes}{\leftarrow} : P \times P_2 \to P_1$$

such that

$$p_1 \leq (p \overset{\otimes}{\leftarrow} p_2) \iff (p_1 \otimes p_2) \leq p \iff p_2 \leq (p_1 \overset{\otimes}{\rightarrow} p)$$

holds for all $p_1 \in P_1$, $p_2 \in P_2$, and $p \in P$.
In case $\mathbb{P}_1 = \mathbb{P}_2 = \mathbb{P}$, we say that $\otimes$ forms a biresiduation w.r.t. $\mathbb{P}$. If in addition $(P, \otimes, \varepsilon)$ is a monoid then $(\mathbb{P}, \otimes, \varepsilon)$ will be referred to as *residuated ordered set.* If in particular $\mathbb{P}$ forms a complete lattice, $(\mathbb{P}, \otimes, \varepsilon)$ is a *residuated complete lattice.*

Remark 2. A biresiduation is characterized as a map $\otimes$ which is residuated in both arguments. In particular a biresiduation is isotone in each argument.

Example 1. Any two sets G and M give rise to a biresiduation as follows: The map

$$\otimes : 2^G \times 2^M \to 2^{G\times M}, (A, B) \mapsto A \times B$$

is a biresiduation w.r.t. $(\mathbf{2}^G, \mathbf{2}^M, \mathbf{2}^{G\times M})$, where $\mathbf{2}^G$ denotes the power set lattice of G. This follows immediately since power set lattices are complete with the supremum operation as set union. In particular for $I \subseteq G \times M$ we consider the context (G, M, I): For all $A \subseteq G$ and all $B \subseteq M$ we have

$$(A \times B) \subseteq \alpha \iff A \subseteq (B \overset{\otimes}{\to} \alpha)$$

where

$$(B \overset{\otimes}{\to} \alpha) = \bigcup \{H \subseteq G \mid H \times B \subseteq \alpha\} = B'.$$

Dually, we get

$$(\alpha \overset{\otimes}{\leftarrow} A) = \bigcup \{N \subseteq M \mid A \times N \subseteq \alpha\} = A'.$$

We have recaptured the derivation operators of classical formal concept analysis as residuals of a specific biresiduation, the cartesian product.
Now, it is worth noting the connection between cartesian product and dyadic product as used in linear algebra. We recall the definition of a dyadic product. Given two n-dimensional vectors $\boldsymbol{u}, \boldsymbol{v}$ over a semiring S we can define the dyadic product as

$$\boldsymbol{u} \otimes \boldsymbol{v} := \boldsymbol{u} \cdot \boldsymbol{v}^T$$

where the second multiplication is simply matrix multiplication. If S is the well-known boolean 2-element semiring, the dyadic product resembles exactly the cartesian product. So, in a sense, the dyadic product generalizes the cartesian product. From our point of view, this is key to understanding fuzzy concept analysis in terms of biresiduations.

Example 2 (t-Norm)**.** A binary operation $\otimes$ on the real unit interval $[0, 1]$, which is isotone in both arguments, is called *t-Norm* if for all $a, b, c \in [0, 1]$ the following hold:

(1) $1 \otimes a = a$
(2) $a \otimes b = b \otimes a$
(3) $a \otimes (b \otimes c) = (a \otimes b) \otimes c$

The t-norm $\otimes$ is *left continuous* if for all $a \in [0, 1]$ and $\alpha \in [0, 1]^I$ (where I is an arbitrary index set) the following holds:

$$a \otimes (\sup_{i\in I} \alpha i) = \sup_{i\in I}(a \otimes \alpha i)$$

From the terminology introduced above, a map $\otimes : [0,1] \times [0,1] \to [0,1]$ is a left continuous t-norm if and only if $([0,1], \leq, \otimes, 1)$ forms a residuated complete lattice. Among the various left continuous t-norms we point out the Gödel t-notm

$$\begin{aligned} \otimes : [0,1] \times [0,1] &\to [0,1], \\ (a,b) &\mapsto \min(a,b) \end{aligned}$$

and the Łukasiewicz t-norm

$$\begin{aligned} \otimes : [0,1] \times [0,1] &\to [0,1], \\ (a,b) &\mapsto \max(a+b-1,0) \end{aligned}$$

These t-norms will be used in our visualizations for mathematical morphology in figure 1 and 2.

In the next section, we will highlight that residuals of a biresiduation play a crucial role in FCA and its abstractions.

3 Abstract Concepts and Maximal Rectangles

Let $\mathbb{P}_1 := (P_1, \leq), \mathbb{P}_2 := (P_2, \leq)$ and $\mathbb{P} = (P, \leq)$ be ordered sets and $\otimes$ a biresiduation w.r.t. $(\mathbb{P}_1, \mathbb{P}_2, \mathbb{P})$. We define

$$\boxed{\mathrm{f}}_{\otimes} := P_1 \times P_2$$

to be its set of *formal rectangles*. If $\alpha \in P$ then

$$\mathcal{K} := (\otimes, \alpha)$$

is called an *abstract context* w.r.t. $(\mathbb{P}_1, \mathbb{P}_2, \mathbb{P})$. Given such an abstract context we can define the set of formal rectangles w.r.t. $\mathcal{K}$ as

$$\boxed{\mathrm{f}}_{\mathcal{K}} := \{(u,w) \in \boxed{\mathrm{f}}_{\otimes} \mid u \otimes w \leq \alpha\}.$$

On a formal rectangle, we can apply our biresiduation operation to yield an *(actual) rectangle*. We define

$$\square_{\mathcal{K}} := \{u \otimes w \mid (u,w) \in \boxed{\mathrm{f}}_{\mathcal{K}}\}$$

to be the set of *(actual) rectangles* w.r.t. $\mathcal{K}$ and

$$\boxed{\mathrm{mf}}_{\mathcal{K}} := \max \boxed{\mathrm{f}}_{\mathcal{K}}$$

to be the set of maximal rectangles regarding the product order on $P_1 \times P_2$. We define the set of abstract concepts as

$$\mathfrak{B}\mathcal{K} := \{(u,w) \in P_1 \times P_2 \mid u \overset{\otimes}{\to} \alpha = w \ \& \ \alpha \overset{\otimes}{\leftarrow} w = u\}.$$

We order the set of maximal rectangles of an abstract context. For $b = (u_b, w_b), c = (u_c, w_c) \in \boxed{\text{mf}}_{\mathcal{K}}$ we set

$$b \leq_{\mathcal{K}} c :\iff u_b \leq_{P_1} u_c \iff w_c \leq_{P_2} w_b.$$

Now we can define an *abstract concept order* as

$$\underline{\mathfrak{B}}\mathcal{K} := (\mathfrak{B}\mathcal{K}, \leq_{\mathcal{K}}).$$

In case $(\mathbb{P}_1, \mathbb{P}_2, \mathbb{P})$ is a triple of complete lattices, $\underline{\mathfrak{B}}\mathcal{K}$ forms a complete lattice, called the abstract concept lattice of $\mathcal{K}$.
As usual in FCA, let us abbreviate $u \overset{\otimes}{\rightarrow} \alpha$ as u', the derivation of u w.r.t. $\mathcal{K} := (\otimes, \alpha)$, and similarly, $\alpha \overset{\otimes}{\leftarrow} w$ as w'. We show that even in our rather abstract setting we can talk about maximal rectangles being the abstract concepts.

Proposition 1. *Let $\mathcal{K} := (\otimes, \alpha)$ be an abstract context w.r.t. $(\mathbb{P}_1, \mathbb{P}_2, \mathbb{P})$ and define $\gamma : P_1 \to P, x \mapsto (x'', x')$ and $\mu : P_2 \to P, x \mapsto (x', x'')$. Then*

1. $\mathcal{BK} = im(\gamma) = im(\mu)$
2. $\boxed{mf}_{\mathcal{K}} = \mathcal{BK}$

We call $(p_1, p_2) \in \boxed{\text{f}}_{\mathcal{K}}$ a *decomposition of* $\mathcal{K}$ if $p_1 \otimes p_2 = \alpha$. If additionally $(p_1, p_2) \in \boxed{\text{mf}}_{\mathcal{K}}$ we call (p_1, p_2) a *conceptual decomposition of* $\mathcal{K}$.

Corollary 1. *Let $\mathcal{K} = (\otimes, \alpha)$ be an abstract context w.r.t. $(\mathbb{P}_1, \mathbb{P}_2, \mathbb{P})$. If (p_1, p_2) is a decomposition of $\mathcal{K}$ there exists a conceptual decomposition (q_1, q_2) of $\mathcal{K}$ with $p_1 \leq q_1$ and $p_2 \leq q_2$.*

Proof. For instance, set $(q_1, q_2) := \gamma(p_1)$. □

4 Biadditivity

We complement the approach sketched above (where ordered sets are used as basic structures) by using complete monoids [15] as basic structures.

Definition 3 (complete monoid)**.** A quadruple $\mathcal{A} := (A, +, 0, \Sigma)$ is called a complete monoid if $(A, +, 0)$ is a commutative monoid and Σ assigns to every $\alpha \in A^I$ (for an arbitrary index set I) an element $\Sigma\alpha =: \Sigma_{i \in I}\alpha i$ of A such that

(1) $\Sigma\alpha = 0$ if $\alpha i = 0$ for all $i \in I$
(2) $\Sigma\alpha = \alpha i$ if $I = \{i\}$
(3) $\Sigma\alpha = \alpha i + \alpha j$ if $I = \{i, j\}$ and $i \neq j$
(4) $\Sigma\alpha = \Sigma\beta$ for every partition $\mathcal{T}$ of I and β given by $\mathcal{T} \to A, T \mapsto \Sigma\alpha_{|T}$

Definition 4 (join complete monoid)**.** A *join complete monoid* is a complete monoid $\mathcal{A} := (A, +, 0, \Sigma)$ such that for every non-empty set I and every $a \in A$ it follows $\sum_{i \in I} a = a$.

Remark 3. For every complete lattice $\mathbb{L} := (L, \leq)$ let $\mathcal{A}(\mathbb{L}) := (L, +, 0, \Sigma)$ be the join complete monoid, where $x + y := sup_{\mathbb{L}}\{x, y\}$ for all $x, y \in L$ and $0 := sup_{\mathbb{L}}\emptyset$ and $\Sigma\alpha := sup_{\mathbb{L}}\{\alpha(i) \mid i \in I\}$ for every index set I and all $\alpha \in L^I$. If on the other hand, $\mathcal{A} := (A, +, 0, \sum)$ is a join complete monoid then $\mathbb{L}(\mathcal{A}) := (A, \leq)$ is a complete lattice, provided $x \leq y :\Leftrightarrow x + y = y$ for all $x, y \in A$. This observation establishes for every set A a bijective correspondence between all complete lattices on A and all join complete monoids on A.

$\mathfrak{L} := (\mathbb{L}_1, \mathbb{L}_2, \mathbb{L})$ is a triple of complete lattices then $\mathcal{A}(\mathfrak{L}) := (\mathcal{A}(\mathbb{L}_1), \mathcal{A}(\mathbb{L}_2), \mathcal{A}(\mathbb{L}))$. We define the analogue of biresiduations for complete monoids.

Definition 5 (biadditive map)**.** Let $\mathfrak{A} := (\mathcal{A}_1, \mathcal{A}_2, \mathcal{A})$ be a triple of complete monoids. Then a biadditive map w.r.t. $\mathfrak{A}$ is defined as a map $\otimes : A_1 \times A_2 \to A$ such that

$$\Sigma\beta \otimes \Sigma\gamma = \Sigma_{(i,j)\in I\times J}\beta i \otimes \gamma j$$

holds for all $\beta \in A_1^I$ and $\gamma \in A_2^J$ for arbitrary index sets I, J.

The next definition will also be useful in the section concerning hedges.

Definition 6 (biadditive setup, morphism)**.** Let $\mathfrak{A} := (\mathcal{A}_1, \mathcal{A}_2, \mathcal{A})$ be a triple of complete monoids and let $\otimes$ be a biadditive map on $\mathfrak{A}$. Then we call the pair $(\mathfrak{A}, \otimes)$ a *biadditive setup*. In case $\mathcal{A}_1 = \mathcal{A}_2 = \mathcal{A}$, we say that $\otimes$ is a *biadditive operation* on $\mathcal{A}$, and $(\mathcal{A}, \otimes)$ forms a *biadditive setup*.

Given two biadditive setups $(\mathfrak{A}, \otimes)$ and $(\mathfrak{U}, \otimes)$ we call $\Phi := (\varphi_1, \varphi_2, \varphi)$ a morphism from $(\mathfrak{U}, \otimes)$ to $(\mathfrak{A}, \otimes)$ if $\varphi_1, \varphi_2, \varphi$ are morphisms from U_1, U_2, and U to A_1, A_2, and A, respectively, and for all $u_1 \in U_1$ and $u_2 \in U_2$ we have

$$\varphi_1(u_1) \otimes \varphi_2(u_2) = \varphi(u_1 \otimes u_2).$$

Biadditive setups and their morphisms obviously form a category. In particular, $(\mathfrak{U}, \otimes)$ is a *substructure* of $(\mathfrak{A}, \otimes)$ if U_1, U_2, and U are complete submonoids of A_1, A_2, and A, respectively, and $U_1 \otimes U_2 \subseteq U$. Clearly, The image of a morphism induces a substructure.

Definition 7 (complete semiring)**.** A tuple $\mathfrak{R} := (R, +, \otimes, 0, 1, \sum)$ is a complete semiring if the following properties are satisfied:

(1) $\mathfrak{R}_{add} := (R, +, 0, \sum)$ is a complete monoid,

(2) $\mathfrak{R}_{mult} := (R, \otimes, 1)$ is a monoid with $1 \neq 0$,

(3) the following distributive laws hold for all $a \in R$, $\alpha \in R^I$:

$$a \otimes (\sum_{i\in I} \alpha i) = \sum_{i\in I} (a \otimes \alpha i)$$
$$(\sum_{i\in I} \alpha i) \otimes a = \sum_{i\in I} (\alpha i \otimes a).$$

Remark 4. If $\mathfrak{R} := (R, +, \otimes, 0, 1, \sum)$ is a complete semiring then $(\mathfrak{R}_{add}, \otimes)$ forms a biadditive setup.

Definition 8. A *join complete semiring* is a complete semiring $\mathfrak{R}$ such that $\mathfrak{R}_{add}$ is a join complete monoid.

Remark 5. If $\mathfrak{R} := (R, +, \otimes, 0, 1, \sum)$ is a given join complete semiring then $\mathcal{L}(\mathfrak{R}) := (\mathbb{L}(\mathfrak{R}_{add}), \otimes, 1)$ forms a complete residuated lattice. If on the other hand $\mathcal{L} := (L, \otimes, \varepsilon)$ with $\mathbb{L} = (L, \leq)$ is a given residuated complete lattice then $\mathfrak{R}(\mathcal{L}) := (L, +, \otimes, 0, \varepsilon, \sum)$ with $(L, +, 0, \sum) := \mathcal{A}(\mathbb{L})$ is a join complete semiring. This establishes a bijection between all join complete semirings on a set R and all complete residuated lattices on R. In particular for every join complete semiring $\mathfrak{R} := (R, +, \otimes, 0, 1, \sum)$ there exist maps $\overset{\otimes}{\rightarrow}$ and $\overset{\otimes}{\leftarrow}$ from $R \times R$ to R such that for all $r, s, t \in R$ the following holds:

$$r \leq (t \overset{\otimes}{\leftarrow} s) \iff (r \otimes s) \leq t \iff s \leq (r \overset{\otimes}{\rightarrow} t)$$

A more extensive discussion of the interplay between biresiduation and biadditivity will be presented in section 6.

Proposition 2. *Let $(\mathfrak{A}, \otimes)$ be a biadditive setup where $\mathfrak{A} := (\mathcal{A}_1, \mathcal{A}_2, \mathcal{A})$. For sets G and M define $\boxtimes : A_1^G \times A_2^M \to A^{G \times M}$ where*

$$(u \boxtimes w)(g, m) := u(g) \otimes w(m).$$

Then $\boxtimes$ forms a biadditive map w.r.t. $(A_1^G, A_2^M, A^{G\times M})$ which is also known as the dyadic product.

More generally, we have the following construction.

Proposition 3. *Let $(\mathfrak{A}, \otimes)$ be a biadditive setup where $\mathfrak{A} := (\mathcal{A}_1, \mathcal{A}_2, \mathcal{A})$. For sets G, H, and M define*

$$* : A_1^{G\times H} \times A_2^{H \times M} \to A^{G \times M}$$

where

$$(\beta * \eta)(g, m) := \Sigma_{h \in H} \beta(g, h) \otimes \eta(h, m).$$

Then $*$ forms a biadditive map w.r.t. $(A_1^{G\times H}, A_2^{H\times M}, A^{G\times M})$ – which, as a matter of fact, is the matrix product.

If $(\mathfrak{A}, \otimes)$ is a biadditive setup where $\mathfrak{A} := (\mathcal{A}_1, \mathcal{A}_2, \mathcal{A})$ and $\alpha \in A$ then a familiy $(u_h, w_h)_{h \in H} \in (A_1 \times A_2)^H$ will be called a sum-decomposition of $(\otimes, \alpha)$ if

$$\alpha = \Sigma_{h \in H} u_h \otimes w_h.$$

An important observation is the following

Proposition 4. *Let $(\mathfrak{A}, \otimes)$ be a biadditive setup where $\mathfrak{A} := (\mathcal{A}_1, \mathcal{A}_2, \mathcal{A})$. Then for sets G, H, M and $\alpha \in A^{G\times M}$ the following holds*

1. *If* (β, η) *is a decomposition of* $(*, \alpha)$*, that is,* $\beta \in A_1^{G \times H}$ *and* $\eta \in A_2^{H \times M}$ *such that* $\alpha = \beta * \eta$*, then* $(u_h, w_h)_{h \in H}$ *is a sum-decomposition of* $(\boxtimes, \alpha)$ *where* $u_h := \beta(\cdot, h)$ *and* $w_h := \eta(h, \cdot)$*.*
2. *Conversely, if* $(u_h, w_h)_{h \in H}$ *is a sum-decomposition of* $(\boxtimes, \alpha)$ *then* (β, η) *is a decomposition of* $(*, \alpha)$ *where* $\beta : G \times H \to A_1, (g, h) \mapsto u_h g$ *and* $\eta : H \times M \to A_2, (h, m) \mapsto w_h m$*.*

Remark 6. An important situation where this proposition can be applied occurs when $\mathfrak{R} = (R, +, \otimes, 0, 1, \Sigma)$ is a complete semiring, since then $(\mathfrak{R}_{add}, \otimes)$ is a biadditive setup as already mentioned in remark 4.

In mathematical morphology the concept of dilation is fundamental, and crucially involves *convolution* in a general setting which will be layed out in the following.

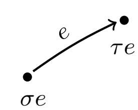
5 Construction of Convolution Algebras

The ingredients of our modelling approach for the construction of convolution algebras are action networks and complete semirings. For this we still need to introduce networks and action networks.

Definition 9 (network)**.** A triple $\mathcal{G} := (V, E, \varrho)$ will be called a *network* (*directed multigraph*) if V and E are sets and $\varrho : E \to V \times V$ is a map. In this context we interpret V as the *set of nodes*, E as the *set of edges*, and ϱ as the *structure map* of $\mathcal{G}$.

Additional Remark: The structure map ϱ of $\mathcal{G}$ induces two maps

$$\sigma : E \to V, e \mapsto \sigma e \quad \text{and}$$
$$\tau : E \to V, e \mapsto \tau e$$



satisfying $\varrho e = (\sigma e, \tau e)$ for all $e \in E$; here we consider σe as *source node* and τe as *target node* of e.

A pair of edges (c, d) will be called *sequential* if the target node of c is equal to the source node of d. The set of all sequential pairs of edges will be denoted by $E^{\langle 2 \rangle}$. Similarly $E^{\langle 3 \rangle}$ will denote the set of all triples of edges (c, d, e) such that (c, d) and (d, e) are sequential pairs.
Next we consider networks with additional structure. In this situation, edges will be interpreted as actions which allow concatenation of sequential pairs of actions.

Definition 10 (action network)**.** A triple $\mathbb{G} := (\mathcal{G}, *, id)$ will be called an *action network* (*small covariant category*) if $\mathcal{G} := (V, E, \varrho)$ is a network and

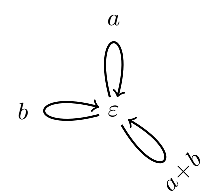
$$* : E^{\langle 2 \rangle} \to E, (c, d) \mapsto c * d \quad \text{and} \quad id : V \to E$$

are maps satisfying:

$$\begin{aligned}&\varrho(c * d) = (\sigma c, \tau d) \text{ for all } (c, d) \in E^{\langle 2\rangle}\\&(c * d) * e = c * (d * e) \text{ for all } (c, d, e) \in E^{\langle 3\rangle}\\&\varrho(idp) = (p, p) \text{ for all } p \in V\\&id(\sigma c) * c = c = c * id(\tau c) \text{ for all } c \in E.\end{aligned}$$

We interprete E as set of *actions* and $*$ as *concatenation map*, which maps every sequential pair of actions (c, d) to its concatenation $c*d$. Furthermore, idp denotes the *passive action* at node p.

Example 3. Any monoid $\mathbb{M} := (M, *, \varepsilon)$ can be interpreted as action network with $\{\varepsilon\}$ as the singleton node set, M as the set of edges, and $*$ as concatenation map as well as $id : \{\varepsilon\} \to M$, $\varepsilon \mapsto \varepsilon$.



Example 4. Any preordered set $\mathbb{P} := (P, R)$ can be interpreted as action network which has (P, R, ρ) as underlying network, where $\rho : R \to V \times V, (p, q) \mapsto (p, q)$, and $* : R^{\langle 2\rangle} \to R, ((p, t), (t, q)) \mapsto (p, q)$ as concatenation map and furthermore $id : P \to R, p \mapsto (p, p)$ as passivity map.

Construction 1 (convolution algebra)**.** Let $\mathfrak{R} := (R, +, \cdot, 0, 1, \sum)$ be a complete semiring and let $\mathbb{G} := (\mathcal{G}, *, id)$ be an action network with underlying network $\mathcal{G} := (V, E, \varrho)$; then the *convolution algebra* of $\mathbb{G}$ over $\mathfrak{R}$ is given by

$$\mathfrak{R}[\mathbb{G}] := (R^E, +, *, \mathbf{O}, \mathbf{I}, \sum),$$

where for every index set I for all $i \in I$ and $u_i, u, w \in R^E$ as well as $e \in E$ the following hold:

$$\begin{aligned}(u + w)e &:= ue + we\\(\sum_{i\in I} u_i)e &:= \sum_{i\in I}(u_i e)\\(u * w)e &:= \sum_{(c,d)\in Split_{\mathbb{G}}(e)} uc \cdot wd\end{aligned}$$

with $Split_{\mathbb{G}}(e) := \{(c, d) \in E^{\langle 2\rangle} | c * d = e\}$ and

$$\begin{aligned}&\mathbf{O} : E \to R, e \mapsto 0\\&\mathbf{I} : E \to R, e \mapsto \begin{cases}1 & \text{for } e \in idV\\0 & \text{otherwise}\end{cases}\end{aligned}$$

Remark 7. If u and w are as above then the product $u * w$ will be called the *convolution* of u with w.

Remark 8. In case $\mathbb{M} := (M, *, \varepsilon)$ is a monoid then $\mathfrak{R}[\mathbb{M}]$ is defined as $\mathfrak{R}[\mathbb{G}]$ where $\mathbb{G}$ is the action network associated with $\mathbb{M}$ and $\mathfrak{R}[\mathbb{M}]$.

6 Combining Biresiduation and Biadditivity

The following fact will help us to combine biresiduation and biadditivity, which will be relevant for fuzzy FCA and also for mathematical morphology:

Proposition 5. *Let* $\mathfrak{L} := (\mathbb{L}_1, \mathbb{L}_2, \mathbb{L})$ *be a triple of complete lattices and let* $\otimes : L_1 \times L_2 \to L$ *be a map. Then* $\otimes$ *is biresiduated w.r.t.* $\mathfrak{L}$ *if and only if* $\otimes$ *is biadditive w.r.t.* $\mathcal{A}(\mathfrak{L})$*.*

Theorem 1. Let $\mathfrak{L} := (\mathbb{L}_1, \mathbb{L}_2, \mathbb{L})$ be a triple of complete lattices and let $\otimes$ be a biresiduation w.r.t. $\mathfrak{L}$. Then for sets G, R, and M the following holds:

1. $\boxtimes$ is a biresiduation w.r.t. $(L_1^G, L_2^M, L^{G\times M})$.
   Hence, for all $u \in L_1^G$ and $w \in L_2^M$ and $\alpha \in L^{G\times M}$ we have
   $$u \leq (\alpha \overset{\boxtimes}{\leftarrow} w) \iff (u \boxtimes w) \leq \alpha \iff w \leq (u \overset{\boxtimes}{\rightarrow} \alpha).$$
   Also,
   $$(\alpha \overset{\boxtimes}{\leftarrow} w)(g) = inf_{L_1}\{\alpha(g,m) \overset{\otimes}{\leftarrow} w(m) \mid m \in M\}$$
   for all $g \in G$ and
   $$(u \overset{\boxtimes}{\rightarrow} \alpha)(m) = inf_{L_2}\{u(g) \overset{\otimes}{\rightarrow} \alpha(g,m) \mid g \in G\}$$
   for all $m \in M$.
2. $*$ is a biresiduation w.r.t. $(L_1^{G\times H}, L_2^{H\times M}, L^{G\times M})$.
   Hence, for all $\beta \in L_1^{G\times H}$ and $\eta \in L_2^{H\times M}$ and $\alpha \in L^{G\times M}$ we have
   $$\beta \leq \alpha \overset{*}{\leftarrow} \eta \iff \beta * \eta \leq \alpha \iff \eta \leq \beta \overset{*}{\rightarrow} \alpha.$$
   Also,
   $$(\alpha \overset{*}{\leftarrow} \eta)(g,h) = inf_{L_1}\{\alpha(g,m) \overset{\otimes}{\leftarrow} \eta(h,m) \mid m \in M\}$$
   for all $(g,h) \in G \times H$ and
   $$(\beta \overset{*}{\rightarrow} \alpha)(h,m) = inf_{L_2}\{\beta(g,h) \overset{\otimes}{\rightarrow} \alpha(g,m) \mid g \in G\}$$
   for all $(h,m) \in H \times M$.

3. Let $\alpha \in L^{G\times M}$ and let (β_0, η_0) be a decomposition of $\mathcal{K} = (*, \alpha)$. Then there exists a conceptual decomposition (β, η) of $\mathcal{K}$ with $\beta_0 \leq \beta$ and $\eta_0 \leq \eta$. The corresponding sum-decomposition $(\beta(\cdot, h), \eta(h, \cdot))_{h\in H}$ of $\mathcal{K}_0 := (\boxtimes, \alpha)$ consists of abstract concepts of $\mathcal{K}_0$. Such a sum-decomposition is called *conceptual sum-decomposition.*
   Conversely, if $(x_h, y_h)_{h\in H}$ is a sum-decomposition of $\mathcal{K}_0$ then there exists a conceptual sum-decomposition $(u_h, w_h)_{h\in H}$ of $\mathcal{K}_0$ with $x_h \leq u_h$ and $y_h \leq w_h$. The corresponding decomposition (β, η) of $\mathcal{K}$ defined via
$$\beta : G \times H \to L_1, (g, h) \mapsto u_h(g)$$
   and
$$\eta : H \times M \to L_2, (h, m) \mapsto w_h(m)$$
   is a conceptual decomposition of $\mathcal{K}$. If (β, η) is a conceptual decomposition of $\mathcal{K}$ then, by definition, $\beta = \alpha \leftarrow \eta$ and $\eta = \beta \rightarrow \alpha$.

Proof. Part 1 follows from Propositions 5 and 2. Part 2 follows from Propositions 5 and 3. Part 3 follows from Proposition 5 and 4 together with Corollary. □

The above theorem extends Theorem 6 from [16]. Note that Part 3 of the above theorem employs a well-known fact from linear algebra: matrix multiplication can be rewritten as summing over the dyadic products of the respective column and row vectors.

Remark 9. Referring to remark 6, the last theorem is connected with fuzzy formal concept analysis in the following way: Let $\mathfrak{R} := (R, +, \otimes, 0, 1, \Sigma)$ be a join complete semiring. Then for sets G, M and $\alpha \in R^{G\times M}$, the triple (G, M, α) will be regarded as *fuzzy context* over $\mathfrak{R}$ having the same concept lattice as the abstract context $(\boxtimes, \alpha)$ w.r.t. $(\mathbb{L}^G, \mathbb{L}^M, \mathbb{L}^{G\times M})$ for $\mathbb{L} := \mathbb{L}(\mathfrak{R}_{add})$. Also the decomposition discussed in the third part of the above theorem applies to this situation.

If we restrict ourselves to the situation of M being a singleton, Theorem 1.1 yields for all $r \in R$ and $u, v \in R^G$
$$u \otimes r \leq v \iff r \leq (u \overset{\otimes}{\to} v),$$
that is, $u \to v$ can be interpreted as the degree of u being a subset of v.

Theorem 2. For every complete semiring $\mathfrak{R} := (R, +, \otimes, 0, 1, \sum)$ and every action network $\mathbb{G} := (\mathcal{G}, *, id)$ with $\mathcal{G} := (V, E, \varrho)$ the convolution algebra $\mathfrak{R}[\mathbb{G}]$ is a complete semiring.
In case $\mathfrak{R}$ is join complete then so is $\mathfrak{R}[\mathbb{G}]$; consequently $(\mathbb{L}(R^E, +, \mathbf{O}, \sum), *, \mathbf{I})$ forms a residuated complete lattice, and for all $u, w \in R^E$ and all $d \in E$ it follows
$$(u \overset{*}{\to} w)d = \inf\{uc \overset{\otimes}{\to} w(c * d) \mid c \in E : (c, d) \in E^{\langle 2\rangle}\}.$$

Proof. Straightforward calculation yields that $\mathfrak{R}[\mathbb{G}]$ is a complete semiring, which is join complete if $\mathfrak{R}$ is join complete.
In the latter, it remains to show that for all $u, v, w \in R^E$ the following holds:

$$
\begin{aligned}
&\forall d \in E : vd \leq (u \overset{*}{\to} w)d \\
\Leftrightarrow& v \leq (u \overset{*}{\to} w) \\
\Leftrightarrow& (u * v) \leq w \\
\Leftrightarrow& \forall e \in E : \sum_{(c,d)\in Split(e)} uc \otimes vd \leq we \\
\Leftrightarrow& \forall e \in E,\ \forall (c,d) \in Split(e) : uc \otimes vd \leq we \\
\Leftrightarrow& \forall (c,d) \in E^{\langle 2 \rangle} : uc \otimes vd \leq w(c * d) \\
\Leftrightarrow& \forall (c,d) \in E^{\langle 2 \rangle} : vd \leq uc \overset{\otimes}{\to} w(c * d) \\
\Leftrightarrow& \forall d \in E : vd \leq \inf\{uc \overset{\otimes}{\to} w(c * d) \mid c \in E : (c,d) \in E^{\langle 2 \rangle}\}
\end{aligned}
$$

Indeed, the above equivalences immediately imply

$$(u \overset{\otimes}{\to} w)d = \inf\{uc \overset{\otimes}{\to} w(c * d) \mid c \in E : (c,d) \in E^{\langle 2 \rangle}\} \text{ for all } d \in E.$$

□

Construction 2 (dilation and erosion)**.** Let $\mathfrak{R} := (R, +, \otimes, 0, 1, \sum)$ be a join complete semiring and $\mathbb{G} := (\mathcal{G}, *, id)$ be an action network with $\mathcal{G} := (V, E, \varrho)$. Then an element $\nu \in R^E$ will be considered as *structuring element* of $\mathcal{G}$ over $\mathfrak{R}$, and the map

$$\delta_\nu : R^E \to R^E, \mu \mapsto (\nu * \mu)$$

is called the *dilation* via the structuring element ν and the map

$$\varepsilon_\nu : R^E \to R^E, \mu \mapsto (\nu \overset{*}{\to} \mu)$$

is the *erosion* via the structuring element ν. In this setting, $\mu \in R^E$ often plays the role of an image, the dilation of which via the structuring element ν is given by

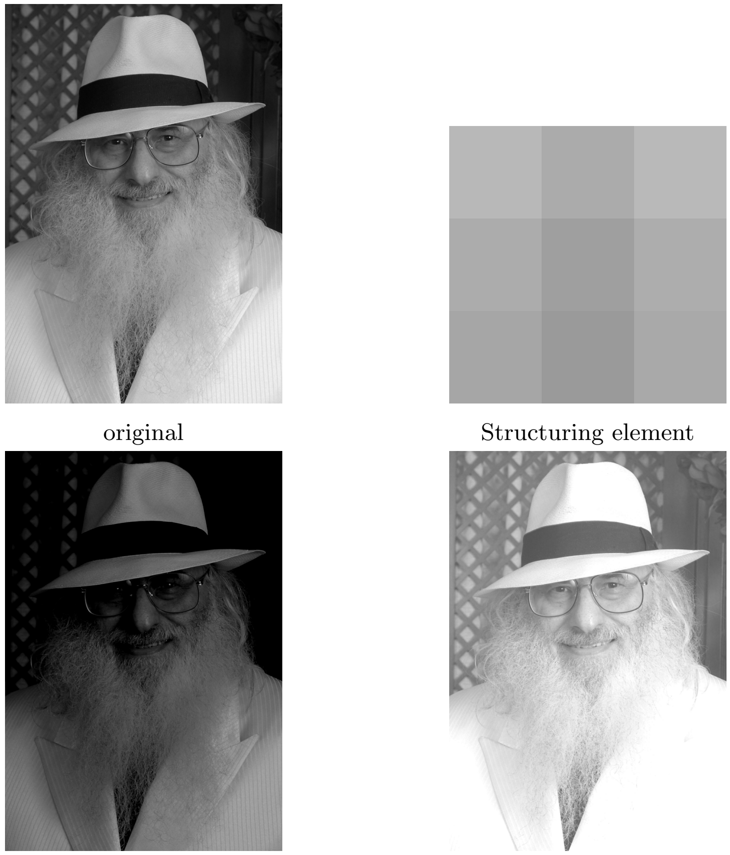
$$(\delta_\nu)\mu = (\nu * \mu).$$

Similarly, the erosion via ν of the image μ is given by

$$(\varepsilon_\nu)\mu = (\nu \overset{*}{\to} \mu).$$

Remark 10. Here, the pair $(\delta_\nu, \varepsilon_\nu)$ is an adjunction on $\mathbb{L}(R^E, +, \mathbf{O}, \sum)$.

A significant application of mathematical morphology is photo editing. Here we visualize dilation and erosion by using convolution algebras induced via the t-norms introduced previously.



original

Structuring element

dilation

erosion

Fig. 1: Łukasiewicz t-norm

dilation

erosion

Fig. 2: Gödel t-norm

Remark 11 (fuzzy FCA)**.** Referring to construction 2, let $\mathfrak{R} := (R, +, \otimes, 0, 1, \sum)$ be a given join complete semiring and $\mathbb{G} := (\mathcal{G}, *, id)$ be an action network with $\mathcal{G} := (V, E, \varrho)$. In this situation, an element $\eta \in R^E$ will be considered as input image and an element $\nu \in R^E$ will be regarded as structuring element. Then $\mu := (\nu \stackrel{*}{\to} \eta)$ is the erosion of η via ν. From the viewpoint of fuzzy FCA, μ is the derivation of ν in the abstract context $\mathcal{K} := (*, \eta)$ w.r.t. $(\mathbb{L}^E, \mathbb{L}^E, \mathbb{L}^E)$ for $\mathbb{L} := \mathbb{L}(\mathfrak{R}_{add})$.

7 Integration of Isabelle Bloch's Approach

Isabelle Bloch has been one of the key scientists in developing mathematical morphology and its fuzzifications during the past 20 years (for example we mention [11,17,18,19] and [20,12,13,21]). In her recent papers on the topic, she considered as underlying structure a so-called *space*, which is given by a commutative group $\mathcal{S} := (S, +, O)$.

Construction 3 (dilation and erosion after Bloch - cf. [11])**.**
Let $\mathcal{S} := (S, +, O)$ be a commutative group and $\otimes : [0,1] \times [0,1] \to [0,1]$ a left continuous t-norm. Then Isabelle Bloch defines the *dilation* via a structuring element $\nu : S \to [0,1]$ as

$$\delta_\nu : [0,1]^S \to [0,1]^S, \quad \mu \mapsto \delta_\nu \mu$$

where

$$(\delta_\nu \mu)x := \sup\{\nu(x - y) \otimes \mu y \mid y \in S\} \text{ for all } x \in S.$$

Similarly, the *erosion* via ν is defined as the map

$$\varepsilon_\nu : [0,1]^S \to [0,1]^S, \quad \mu \mapsto \varepsilon_\nu \mu$$

where

$$(\varepsilon_\nu \mu)x := \inf\{\nu(y - x) \stackrel{\otimes}{\to} \mu y \mid y \in S\} \text{ for all } x \in S.$$

Here $\delta_\nu \mu$ is the dilation of the image μ via the structuring element ν, and $\varepsilon_\nu \mu$ is the erosion of the image μ via the structuring element ν. Indeed, Bloch proves that the pair $(\delta_\nu, \varepsilon_\nu)$ forms an adjunction on $([0,1], \leq)$.

Remark 12. By theorem 2 it follows immediately that $(\delta_\nu, \varepsilon_\nu)$, as introduced by Bloch, forms an adjunction on $([0,1], \leq)$: Indeed, theorem 2 implies for the complete semiring $\mathfrak{R} := ([0,1], \vee, \otimes, 0, 1, \sup)$ that the pair $(\delta_\nu, \varepsilon_\nu)$ forms an adjunction on $\mathfrak{R}[\mathcal{S}]$, that is, on $\mathbb{L}(\mathfrak{R}[\mathcal{S}])$.

In our final section we will discuss hedges and their role for information reduction.

8 Hedges

Hedges were introduced as parameters controlling the size of a fuzzy concept lattice [7]. In [22], fuzzy concept lattices with hedges are shown to be essentially generalized concept lattices in Krajci's sense. Also, in our framework, they have a natural place. We find that they are intimately connected with the substructures of join-complete semirings. A hedge $^*: L \to L$ is defined as a kernel operator on $\mathbb{L}$ (i.e. $a^* \leq b \iff a^* \leq b^*$) such that it preserves the 1 ($1^* = 1$) and weakly preserves implications ($(a \overset{\otimes}{\to} b)^* \leq a^* \overset{\otimes}{\to} b^*$). Note that this definition of a hedge obviously generalizes the notion introduced in [7] (monotonicity for Belohlavek's notion is shown in [22], Lemma 1). The next theorem shows that the theory of hedges can also be formulated in a linear algebraic language within the framework of join-complete semirings.

Theorem 3. Let $\mathcal{L} = (\mathbb{L}, \otimes, \varepsilon)$ be a residuated complete lattice. Then its hedges are in one-to-one correspondence with the substructures of the join-complete semiring $\mathfrak{R}(\mathcal{L})$.

Proof. Since hedges are kernel operators they are in one-to-one correspondence with their kernel systems (a hedge * is mapped to its image set L^*).
Let * be a hedge on $\mathcal{L}$. We will show that its image set L^* forms a substructure (join-complete sub-semiring) of $\mathfrak{R}(\mathcal{L})$. We know already that L^* is closed under joins, that $1^* = 1 \in L^*$, and that $0^* = 0 \in L^*$. To show that L^* is closed under $\otimes$ we will verify that $a \otimes b = (a \otimes b)^*$ holds for all $a, b \in L^*$; it suffices to prove $a \otimes b \leq (a \otimes b)^*$:

$$\begin{aligned}
& a \otimes b \leq a \otimes b \\
\iff \quad & b \leq (a \overset{\otimes}{\to} (a \otimes b)) \\
\implies \quad & b^* \leq (a \overset{\otimes}{\to} (a \otimes b))^* \\
\implies \quad & b^* \leq (a^* \overset{\otimes}{\to} (a \otimes b)^*) \\
\iff \quad & a^* \otimes b^* \leq (a \otimes b)^*
\end{aligned}$$

Since $a^* = a$ and $b^* = b$ we conclude the argument.
Let $\mathcal{K}$ be a substructure of a given join-complete semiring $\mathfrak{R}(\mathcal{L})$. Since $\mathcal{K}$ is closed under arbitrary joins it forms a kernel system in $\mathbb{L}$ inducing its kernel operator *. Thus, $\mathcal{K} = L^*$ and $1 \in L^*$. It remains to be shown that implications are weakly preserved by *:

$$\begin{aligned}
& (a \overset{\otimes}{\to} b) \leq (a \overset{\otimes}{\to} b) \\
\iff \quad & a \otimes (a \overset{\otimes}{\to} b) \leq b \\
\implies \quad & a^* \otimes (a \overset{\otimes}{\to} b)^* \leq b
\end{aligned}$$

Since $\mathcal{K}$ is a substructure of $\mathfrak{R}(\mathcal{L})$ the element $a^* \otimes (a \overset{\otimes}{\to} b)^*$ is a kernel and we get $a^* \otimes (a \overset{\otimes}{\to} b)^* \leq b^*$, which implies $(a \overset{\otimes}{\to} b)^* \leq (a^* \overset{\otimes}{\to} b^*)$. □

Remark 13 (construction of hedges via substructures)**.** Let $\mathbb{G} := (\mathcal{G}, *, id)$ be an action network with $\mathcal{G} := (V, E, \varrho)$. We say that a subset D of E forms a *substructure* of $\mathbb{G}$ if $idV \subseteq D$ and $c * d \in D$ holds for all $(c, d) \in D^{\langle 2 \rangle}$. If $\mathfrak{R} := (R, +, \otimes, 0, 1, \sum)$ is a complete semiring and D forms a substructure of $\mathbb{G}$, then the set

$$U := \{u \in S^E \mid ue = 0 \text{ for all } e \in E \setminus D\}$$

forms a substructure of the complete semiring $\mathfrak{R}[\mathbb{G}]$. In case $\mathfrak{R}$ is a join complete semiring, the hedge associated with U is given by

$$^* : R^E \to R^E, u \mapsto u_D$$

where $u_D e := ue$ for all $e \in D$ and $u_D e := 0$ for all $e \in E \setminus D$.

One application of the above for mathematical morphology is when the action network is given by $\mathbb{Z}^2_{add}$ and the substructure is $D := 2\mathbb{Z}^2$, where the join complete semiring $\mathfrak{R}$ is induced by a t-norm.

Finally we present a generalization of the concept of hedges.

Proposition 6. *Let* $\otimes : P_1 \times P_2 \to P$ *be a biresiduation and let* $(^{*1}, ^{*2}, ^{*})$ *be a triple of kernel operators. The following are equivalent*

1. $\forall p_1 \in P_1, \forall p \in P : (p_1 \overset{\otimes}{\to} p)^{*2} \leq p_1^{*1} \overset{\otimes}{\to} p^*$
2. $\forall p_1 \in P_1, \forall p_2 \in P_2 : p_1^{*1} \otimes p_2^{*2} \leq (p_1 \otimes p_2)^*$
3. $P_1^{*1} \otimes P_2^{*2} \subseteq P^*$

A star system *is a triple of kernel operators where one of the above conditions holds.*

Proof. " $1 \Rightarrow 2$":

$$\begin{aligned} & p_1 \otimes p_2 \leq p_1 \otimes p_2 \\ \iff & p_2 \leq (p_1 \overset{\otimes}{\to} (p_1 \otimes p_2)) \\ \implies & p_2^{*2} \leq (p_1 \overset{\otimes}{\to} (p_1 \otimes p_2))^{*2} \\ \implies & p_2^{*2} \leq (p_1^{*1} \overset{\otimes}{\to} (p_1 \otimes p_2)^*) \\ \implies & p_1^{*1} \otimes p_2^{*2} \leq (p_1 \otimes p_2)^* \end{aligned}$$

" $2 \Rightarrow 1$":

$$\begin{aligned} & (p_1 \overset{\otimes}{\to} p) \leq (p_1 \overset{\otimes}{\to} p) \\ \iff & (p_1 \otimes (p_1 \overset{\otimes}{\to} p)) \leq p \\ \implies & (p_1 \otimes (p_1 \overset{\otimes}{\to} p))^* \leq p^* \\ \implies & (p_1^{*1} \otimes (p_1 \overset{\otimes}{\to} p)^{*2}) \leq p^* \\ \iff & (p_1 \overset{\otimes}{\to} p)^{*2} \leq (p_1^{*1} \overset{\otimes}{\to} p^*) \end{aligned}$$

" $2 \Rightarrow 3$": By 2, $p_1^{*1} \otimes p_2^{*2} \leq (p_1^{*1} \otimes p_2^{*2})^*$ and since * is a kernel operator we have equality and therefore $p_1^{*1} \otimes p_2^{*2} \in P^*$.
" $3 \Rightarrow 2$": Since $p_1^{*1} \otimes p_2^{*2} \leq p_1 \otimes p_2$ and $p_1^{*1} \otimes p_2^{*2} \in P^*$ we have $p_1^{*1} \otimes p_2^{*2} \leq (p_1 \otimes p_2)^*$. □

The previous proposition gives rise to the following application.

Proposition 7. *Let* $\mathfrak{L} := (\mathbb{L}_1, \mathbb{L}_2, \mathbb{L})$, $\mathfrak{P} := (\mathbb{P}_1, \mathbb{P}_2, \mathbb{P})$ *be triples of complete lattices and let* $\otimes_{\mathfrak{L}} : L_1 \times L_2 \to L$ *and* $\otimes_{\mathfrak{P}} : P_1 \times P_2 \to P$ *be biresiduations. Then for every morphism* $\Phi := (\varphi_1, \varphi_2, \varphi)$ *from* $(\mathcal{A}(\mathcal{L}), \otimes_{\mathfrak{L}})$ *to* $(\mathcal{A}(\mathfrak{P}), \otimes_{\mathfrak{P}})$ *the triple* $(\varphi_1 \circ \varphi_1^+, \varphi_2 \circ \varphi_2^+, \varphi \circ \varphi^+)$ *forms a star system w.r.t.* $(\mathfrak{P}, \otimes_{\mathfrak{P}})$.

Proof. It suffices to verify property 3 from the previous proposition:

$$(\varphi_1 \circ \varphi_1^+)P_1 \otimes_{\mathcal{P}} (\varphi_2 \circ \varphi_2^+)P_2 = \varphi(\varphi_1^+ P_1 \otimes_{\mathcal{L}} \varphi_2^+ P_2) \subseteq \varphi L = (\varphi \circ \varphi^+)P.$$

□

As a consequence of our considerations we receive the following extension of Theorem 3.

Theorem 4. Let $\otimes$ be a biresiduation on a triple of complete lattices $\mathfrak{L} := (\mathbb{L}_1, \mathbb{L}_2, \mathbb{L})$. Then the star systems w.r.t. $(\mathfrak{L}, \otimes)$ are in one-to-one correspondence with the substructures of $(\mathcal{A}(\mathfrak{L}), \otimes)$.

9 Conclusion

Some background information: Our paper is indeed based on our 2012 paper "A Macroscopic Approach to FCA and its Various Fuzzifications" but goes far beyond it. Who carefully reads our present paper will realize that notions have been refined and adjusted to our situation. Roughly said, biresiduation generalizes residuated posets while biadditivity generalizes complete semirings. What we mainly need is a specialization where these two concepts meet, that is, residuated complete lattices (algebraic logic point of view) which correspond to join complete semirings (linear algebra point of view). The importance of the latter point is for constructions.

Looking into Isabelle Bloch's constructions for mathematical morphology, we found out that there is a general framework in linear algebra over complete semirings which is outlined in section 5, named "Construction of Convolution Algebras". These turn out to be again complete semirings. So the bijective correspondence between residuated complete lattices and join complete semirings can be lifted from a "coordinate level" to a "space level". That is what Bloch essentially does in a special situation (without putting this into a general framework) and we derive from the general construction of convolution algebras over join complete semirings (and a subsequent paradigm shift into residuated complete lattices).

References

1. Ganter, B., Wille, R.: Formal Concept Analysis: Mathematical Foundations. Springer, Berlin/Heidelberg (1999)

2. Krajci, S.: The basic theorem on generalized concept lattice. In Snásel, V., Belohlávek, R., eds.: CLA. Volume 110 of CEUR Workshop Proceedings., CEUR-WS.org (2004)
3. Medina, J., Ojeda-Aciego, M., Ruiz-Calviño, J.: Formal concept analysis via multi-adjoint concept lattices. Fuzzy Sets and Systems **160**(2) (2009) 130–144
4. Medina, J., Ojeda-Aciego, M.: On the representation theorem of multi-adjoint concept lattices. In Carvalho, J.P., Dubois, D., Kaymak, U., da Costa Sousa, J.M., eds.: IFSA/EUSFLAT Conf. (2009) 1091–1095
5. Davey, B.A., Priestley, H.A.: Introduction to lattices and order. Cambridge University Press, Cambridge (1990)
6. Belohlávek, R., Vychodil, V.: Factor analysis of incidence data via novel decomposition of matrices. In Ferré, S., Rudolph, S., eds.: ICFCA. Volume 5548 of Lecture Notes in Computer Science., Springer (2009) 83–97
7. Belohlávek, R., Vychodil, V.: Reducing the size of fuzzy concept lattices by hedges. In: FUZZ-IEEE 2005. The IEEE International Conference on Fuzzy Systems, 22-25 May, Reno, NV, USA. (2005) 663–668
8. 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. Volume 6743 of LNCS., Springer (2011) 19–26
9. Belohlávek, R., Vychodil, V.: What is a fuzzy concept lattice? In Bělohlávek, R., Snáŝel, V., eds.: Proc. CLA 2005. Volume 162 of CEUR WS., Palacký University in Olomouc, VŠB – Technical University of Ostrava (2005) 34–45
10. Blyth, T.: Lattices and Ordered Algebraic Structures. Universitext, Springer (2005)
11. Bloch, I.: Duality vs. adjunction for fuzzy mathematical morphology and general form of fuzzy erosions and dilations. Volume 160. Elsevier (2009) 1858–1867
12. Bloch, I.: Fuzzy sets and mathematical morphology. Wiley Online Library (2013) 155–176
13. Bloch, I., Maître, H.: Fuzzy mathematical morphology. Volume 10. Springer (1994) 55–84
14. Kaiser, T.B., Schmidt, S.E.: A macroscopic approach to fca and its various fuzzifications. In Domenach, F.e.a., ed.: Formal Concept Analysis, Proceedings of ICFCA 2012. Volume 7278 of LNAI., Springer (2012) 140–147
15. Droste, M., Kuich, W., Vogler, H.: Handbook of Weighted Automata. Monographs in Theoretical Computer Science, Springer (2009)
16. Belohlávek, R.: Optimal decompositions of matrices with entries from residuated lattices. Journal of Logic and Computation (2011)
17. Bloch, I.: Bipolar fuzzy mathematical morphology for spatial reasoning. In: Mathematical Morphology and Its Application to Signal and Image Processing. Springer (2009) 24–34
18. Bloch, I., Maître, H.: Ensembles flous et morphologie mathématique. (1992)
19. Bloch, I., Preteux, F., Boulanger, F., Soussaline, F.: A meta-model for segmentation problems in mathematical morphology. In de Graaf, C., Viergever, M., eds.: Information Processing in Medical Imaging. Springer US (1988) 45–64
20. Bloch, I.: Lattices of fuzzy sets and bipolar fuzzy sets, and mathematical morphology. Volume 181. (2011) 2002 – 2015
21. Bloch, I., Maître, H.: Constructing a fuzzy mathematical morphology: alternative ways. In: Fuzzy Systems, 1993., Second IEEE Int. Conf., IEEE (1993) 1303–1308
22. Krajci, S.: Every concept lattice with hedges is isomorphic to some generalized concept lattice. In Snásel, V., Belohlávek, R., eds.: CLA 2005 proceedings, ISBN 80–248–0863–3. (2005) 1–9

Continuous Target Variable Prediction with Augmented Interval Pattern Structures: Lazy Algorithm

Alexey Masyutin and Sergei O. Kuznetsov

National Research University Higher School of Economics
Moscow, Russia
alexey.masyutin@gmail.com,skuznetsov@hse.ru

Abstract. Pattern structures are known to provide a tool for predictive modeling and classification. However, in order to generate classification rules concept lattice should be built. This procedure may take much time and resources. In previous work it was shown that it is possible to escape the problem with so-called lazy associative classification algorithm. It does not require lattice construction and it is applicable to classification problems such as credit scoring. In this paper we adjust this method to the case of continuous target variable, i.e. regression problem, and apply it to recovery rates forecasting. We perform parameters tuning, assess the accuracy of the algorithm based on the bank data and compare it to the models adopted in the bank system and other benchmarks.

1 Introduction

Banks and financial institutions take and mitigate credit risk on daily basis. Credit risk commonly has the biggest contribution to the bank losses compared to other types of risks such as market, operational and liquidity risks [10]. The key to successful risk-management is to adequately assess the possibility of credit losses and potential amount of the loans that is going to be recovered in case of default. The problem of accurate risk assessment is not only important for an individual bank, but it is also crucial for the banking system as a whole. The problem is so vital that banking industry is strictly regulated by central banks and Basel supervising committee, which even pose certain requirements for predictive models that are used by banks [17]. Some predictive models, so-called "black-box" models provide good results that are hard to interprete. So, the major 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]. At the same time, the more accurate the model is, the less capital charge the bank is going to have. So, banks prefer accurate models that provide interpretable decision-making. Therefore, FCA-based algorithms seem to be helpful since they rely on concepts that have obvious interpretation. The intent of a concept can be interpreted as a set of rules that is supported by the

© Marianne Huchard, Sergei O. Kuznetsov (Eds.): CLA 2016, pp. 273–284,
ISBN 978-5-600-01454-1, National Research University Higher School of Economics, 2016.

extent of the concept. In previous work it was shown that FCA-based interval pattern structures methods are applicable to credit scoring which represents the classification problem with binary target variable [14,16]. Classifying credit applicants into good and potentially delinquent clients is the first part of credit risk assessment. The second part is to estimate *recovery rate* in case of default, i.e. the proportion of the loan that is going to be collected by the bank [10]. As far as recovery rates prediction is concerned, it implies continuous target variable. In this paper, we will adopt the lazy classification algorithm based on interval pattern structures to the case of continuous target variable, i.e. we will introduce modified lazy *regression* algorithm (MLRA). The paper is structured as follows: Section 2 provides basic formal concept analysis definitions. Section 3 describes the architecture of MLRA and its parameters. Section 4 describes the data used for algorithm accuracy evaluation and comparison with benchmarks such as random forests. Section 5 concludes the paper. Finally, we attach a pseudo-code for the algorithm in Appendix.

2 Main Definitions

First, we recall some standard definitions related to FCA, 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 $\rightarrow$ 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) := {δ(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* $(\cdot)^\diamond$:

$$A^\diamond = \bigsqcap_{g \in A} \delta(g) \qquad \text{for } A \in G,$$

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

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]. 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 [14]:

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

The concept-based learning model for standard object-attribute representation (i.e., formal contexts) is naturally extended to pattern structures, when we have a binary target attribute, i.e. a set of positive examples G_+ and a set of negative examples G_- [16].

However, what should we do when the target attribute is not a class label but a continuous variable? For that case we augment the definition of interval pattern structure by equipping it with additional feature h.

Augmented interval pattern structures

Let us define an *augmented interval pattern structure* as a quadruple (G, $\underline{D}$,δ, h), where the description d consists of two elements d_x and d_y (d_y is an interval for target attribute $y \in \mathbb{R}$ and d_x is a vector of intervals for explanatory attributes x which are supposed to predict the target attribute y), $\delta : G \to D$ and $h \in H$, where H is a family of density distribution functions for target attribute y, i.e. $\int_{-\infty}^{+\infty} h(s)ds = 1$. We will also use notation δ_x and δ_y to distinguish between descriptions containing explanatory attributes and target attribute correspondingly. The meet operation definition is left unchanged.

Suppose, we have an arbitrary set of objects $A_0 \subseteq G$, i.e. $A_0 = \{g_1, g_2, ..., g_J\}$, $\delta(g_j) = \{\delta_x, \delta_y\} = \{[x_{1j}; x_{1j}], ..., [x_{Mj}; x_{Mj}], [y_j; y_j]\}$, for $j = 1, ..., J$, where M is number of explanatory attributes. Then we define the derivation operator in the following way

$$A_0^\diamond = (d_0, h_0)$$

where $d_0 = \{d_{x0}, d_{y0}\}$, and $d_{x0} = \delta_x(g_1) \sqcap ... \sqcap \delta_x(g_J)$ and target attribute description $d_{y0} = \delta_y(g_1) \sqcap ... \sqcap \delta_y(g_J)$ which is in fact a single interval $[y_{min}, y_{max}]$ and $h_0 : d_{y0} \to [0; 1]$. The h_0 is in effect a target attribute density distribution function based on observations of A_0, which we describe below. Let $\tau_0, ..., \tau_K$ be a partition of d_{y0} and $\tau_0 = y_{min}, \tau_K = y_{max}$ and $\Delta\tau_i = \frac{y_{max} - y_{min}}{K} = \tau_i - \tau_{i-1}, i = 1, ..., K$. Then:

$$h([\tau_{i-1}, \tau_i)) = \frac{|\{g \in A | [\tau_{i-1}, \tau_i) \sqsubseteq \delta_y(g)\}|}{|A|}, \forall i = 1, ..., K$$

Thus, h is a density function of target attribute y values of objects in A. The derivation operator on descriptions returns the set of objects with description subsuming the description d_{x0} whatever target description d_{y0} and density function h are:

$$A_0^{\diamond\diamond} = (d_0, h_0)^\diamond \overset{\text{def}}{=} d_{x0}^\diamond = A_1$$

where $A_1 = \{g \subseteq G | d_{x0} \sqsubseteq \delta_x(g)\}$. Finally, $A_1^\diamond = (d_1, h_1)$. Note, that $d_1 = \{d_{x0}, d_{y1}\}$, i.e. only target attribute description d_y is updated, so does h density function, while the explanatory variables description d_{x0} remains the same.

In order to approach target attribute prediction problem it will be useful to define α*-weak premises* with *allowed dropout*. An h-augmented interval pattern $d \in D$ is called an α-weak premise with allowed ω-dropout iff:

$$1 - \frac{|\{g \in A | d_y^{min} - \omega(m - d_y^{min}) \leq \delta_y(g) \leq d_y^{max} + \omega(d_y^{max} - m)\}|}{|A|} \leq \alpha$$

where $d = (d_x, d_y)$, d_y is a single interval $[d_y^{min}; d_y^{max}]$ for target attribute y $A = d_x^{\diamond}$, and m is a median of density function h which reflects the distribution of target attribute within the interval d_y based on objects from A. Parameter α controls the frequency of hypothesis falsifications and parameter ω controls the magnitude of falsification, i.e. how dramatically it is falsificated. In our case the magnitude is evaluated as the times the $\delta_y(g) - d_y^{max}$ is larger than $d_y^{max} - m$ if $\delta_y(g) > d_y^{max}$ or the times the $\delta_y(g) - d_y^{min}$ is larger than $m - d_y^{min}$ if $\delta_y(g) < d_y^{min}$ Note, that in case when $\omega = 0$ we apply the strictest criterion to consider a hypothesis as falsificated:

$$1 - \frac{|\{g \in A | d_y^{min} \leq \delta_y(g) \leq d_y^{max}\}|}{|A|} \leq \alpha \Leftrightarrow 1 - \frac{|\{g \in A | d_y \sqsubseteq \delta_y(g)\}|}{|A|} \leq \alpha \Leftrightarrow$$

$$\Leftrightarrow \frac{|\{g \in A | d_y \not\sqsubseteq \delta_y(g)\}|}{|A|} \leq \alpha$$

3 Lazy predictive algorithm with continuous target attribute

Assume we have a set of objects G and numerical context with a set of explanatory attributes $x_1, ..., x_M$ and target attribute y. In contrast to classification problem the context is not divided into positive and negative examples as soon as y take numerical values. Now, suppose we receive a test object g_t with observable attributes x, but with unknown value of target attribute y. Is there a way to predict y using interval pattern structures approach? Indeed, there is, and we are going to describe it below and compare the accuracy results with some benchmarks.

The first stage of algorithm is mining α-weak premises with allowed ω-dropout, the second is to perform prediction for test object g_t based on the mined premises. Let us start by choosing *subsample size* parameter which is the number of objects being randomly extracted from G. Then we specify α and ω parameters that control for "anti-support" in terms of both frequency and magnitude. Upon randomly extracting some objects $A_0 = \{g_1, ..., g_K\}$ we compute following pattern $d_0 = \delta(g_1) \sqcap ... \sqcap \delta(g_K) \sqcap \delta(g_t)$ and density distribution function h_0 for target attribute values. If d_0 is an α - weak premise with allowed ω-dropout then it is added to the set of premises that will be used for prediction later. Together with the pattern it is necessary to store the density function h. But which of h_0, h_1 or other we have to use?

Here we introduce another parameter of the algorithm which is called "*capped*". Capped is a boolean value, and if true then the range for target attribute d_{y1} in $d_0^{\diamond\diamond}$ is truncated to d_{y0} and corresponding density function is h_1 calculated on the truncated set of target values. If capped parameter is false, then we add d_{y1} and calculate the density function based on all target values that fell into d_{y1} based on objects from $d_0^\diamond$. The whole procedure is repeated many times and the *number of iterations* parameter controls for that.

Having finished with premises mining, we move on to the next stage which is building up a prediction for target attribute based on mined premises. In our case, the resulting prediction was defined by mixture of distributions from all premises. In practice all target attribute values stored within premises were put together to form a final distribution. Finally, we tried both an average and a median of that distribution as the prediction for target attribute. Such approach takes into accout different *support* of the premises as soon as premises with greater number of objects will contribute more.

However, one can argue that premises are different in sense of *anti-support* and *deviation* in target attribute values. Indeed, we would put more weight to the prediction based on premises with narrow range of target attribute values and the ones with less falsifying examples from set G. Therefore, we added target values to the final distributions with different weights, thus both weighted average and weighted median were used as forecast.

We introduced two boolean parameters which controlled the weightening schemes. The first parameter is *account for anti-support* and the second is *penalty for high deviation*. When account for anti-support parameter is true, then the target values $\delta_y(g)$ of objects $g \in A$ with the premise d are given weight according to the anti-support of that premise:

$$w_a = \frac{|\{g \in A | d_y^{min} - \omega(m - d_y^{min}) \leq \delta_y(g) \leq d_y^{max} + \omega(d_y^{max} - m)\}|}{|A|}$$

When penalty for high deviation is true, then the weight is decreased with the higher deviation in the target attribute values:

$$w_{pen} = \frac{1}{\sigma(\delta_y(g))}$$

where $\sigma(\delta_y(g))$ is standard deviation of target attribute values. If the parameters values are false then the weigths are equal to one. The final weight for the target attribute value of the object g, which will be contributed to aggregate distribution used for prediction, is defined as product of the two weights:

$$w(g) = w_a \cdot w_{pen}$$

Finally, suppose that P is a set of mined α-weak premises with allowed ω-dropout. The prediction for target attribute y of a test object g_t can be based on weighted average:

$$\widehat{\delta_y(g_t)} = \frac{\sum_{p \in P} \sum_{g \in A_p} \delta_y(g) \cdot w(g)}{\sum_{p \in P} \sum_{g \in A_p} w(g)}$$

or on the weighted median:

$$\widehat{\delta_y(g_t)} = \underset{g \in \cup_p A_p}{\text{median}}(\bigcup_{p \in P} \bigcup_{g \in A_p} (\delta_y(g), w(g))$$

In case where P is an empty set, the prediction is average or median of all target attribute values in G, i.e. the prediction is based on "naive" model.

4 Data and experiments

The data we used for the computation represent a pool of delinquent corporate clients loans, which were expected to be restructured. The process of restructuring is started at the early stage when the client shows the first signs of insolvency. At that very moment a bank chooses either to execute *default strategy*, when the court processes are launched and any disposable collateral is displayed for sale, or to execute *restructuring strategy*, when the funding conditions are being revisited usually resulting in a longer credit period. In case of corporate clients banks usually do not want to go to extremes right from the start as soon as court launch and collateral sales imply costs and spending time resources. Also, the bank would prefer to maintain relations with the client if financial distress is temporary. So, the decision whether to launch default strategy or not is based to the greater extent on the recovery expectations. This makes the problem of recovery prediction crucial for banking decision making. Recovery rate is a number between zero and one which reflects the share of the current exposure which the client is going to payback on some time horizon. If recovery rate expectation is at high level, the bank would prefer restructuring and court launch otherwise.

In this paper we use financial data from balance sheets and profit and loss statements of 612 corporate clients of a top-10 Russian bank. Among others factors we used assets-to-liabilities ratio, debt-to-equity ratio, earnings before taxes and interest payments, return on assets etc, resuting. These clients were assessed at the time of early insolvency signals and the resulting recovery rate was collected.

The data was randomly divided into two parts with 70% of observations in one part and 30% in the other. The bigger part was used as a context with known target attribute for the lazy algorithm and 30% was used as a test set to evaluate predictions and their accuracy. The same data partition was used to run random forests with different tunings with 70% part used as a training set and the other as test set. For random forests there were three parameters tuned by grid search which are *minimum nodesize*, *number of trees* and *number of feasible variables.*

The accuracy of predictions were evaluated in terms of mean absolute deviation (MAD):

$$MAD = \frac{\sum_{i=1}^{N} |y_i - \hat{y}_i|}{N}$$

where y_i is a target attribute (recovery rate) for i-th client in the test set and $\hat{y}_i$ is prediction.

The random forests were run with following parameters grid: minimum node-size ranging from 30 to 100 with increment 10, number of trees ranging took values 10, 30, 50 and 100, and number of feasible variables from ranging from 5 to 45 with increment of 5.

As far as lazy algorithm is concerned, we tuned seven parameters, four of them were continuous and three were boolean. *Subsample size* took following values: 0.01, 0.02, 0.03, 0.04, 0.05, 0.1. *Number of iterations*: 100, 500, 1000, 2000. *Alpha threshold*: 0, 0.05, 0.01, 0.015, 0.02. *Allowed dropout*: 0, 0.1, 0.5, 1, 1.5.

For each combination of parameters we calculated MAD for the test set and in fact that produced metadata for the analysis. Effectively we obtained MAD distributions, which at the first step helped us to choose in favour of forecast based on weighted median forecast rather than weighted average as soon as MAD distributions for the latter took dramatically higher values which are, of course, undesirable.

When building new algorithm one has some intuition about it mechanism and we performed regression analysis of algorithm accuracy versus parameters values to check that intuition. Also, the analysis was important to determine better parameters tuning and explain variation in accuracy of the predictions. The results of regression are presented below:

Table 1. Regression analysis for dependency between MAD and algorithm parameters

Coefficients	Estimate	Std.Error	t	p-value
(Intercept)	0,3288	0,0006	519,4	0,0000
Subsample size	0,0155	0,0031	4,940	0,0000
Number of iterations	-0,0004	0,0000	-18,05	0,0000
Alpha-threshold	-0,0457	0,0270	-1,695	0,0903
Allowed dropout	-0,0011	0,0004	-2,975	0,0030
Capped	-0,0022	0,0004	-5,401	0,0000
Account for anti-support	0,0002	0,0004	0,624	0,5329
Penalty for high deviation	0,0010	0,0004	2,433	0,0150

We see that increasing number of iterations, allowing dropouts and using capped improve algorithm performance as soon as the coefficients are negative and significant: overall error of prediction decreases as those factors increase. Surprisingly, adjusting account for anti-support and penalty for high deviation parameters do not show significant improvement in accuracy. Also, we expected that there are some non-linear dependencies between MAD and parameter values as soon as, intuitively, there has to be an optimal subsample size of randomly extracted objects. Therefore, we support the regression output with one-factor scatter plots with average MAD across all other iterations versus each parameter:

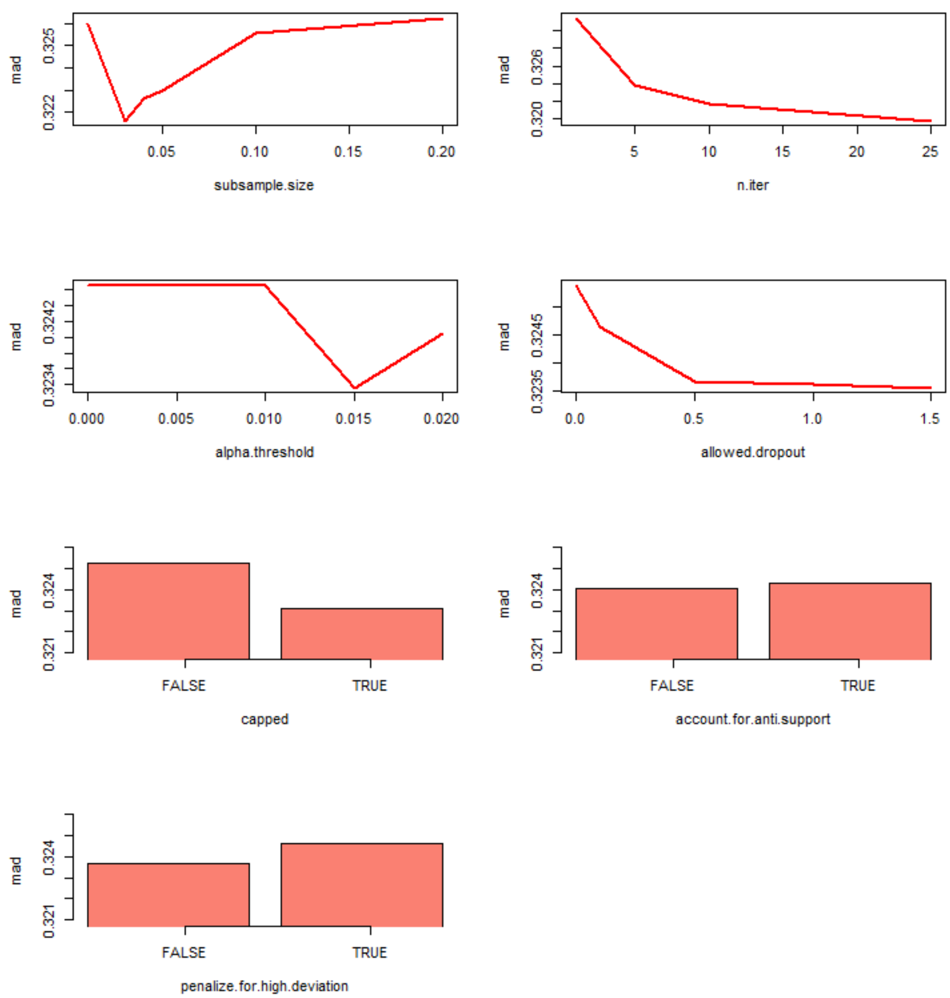


Fig. 1. Single-factor analysis of average MAD versus parameter value: continuous and boolean parameters

As expected, there is a local minimum for the subsample size being extracted from G. It is quite natural 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 by objects with target attribute value out of the premise description target range.

According to performed grid search the range with the lowest MAD (0.247 - 0.290) on the test sample is achieved in following parameter area: alpha-threshold

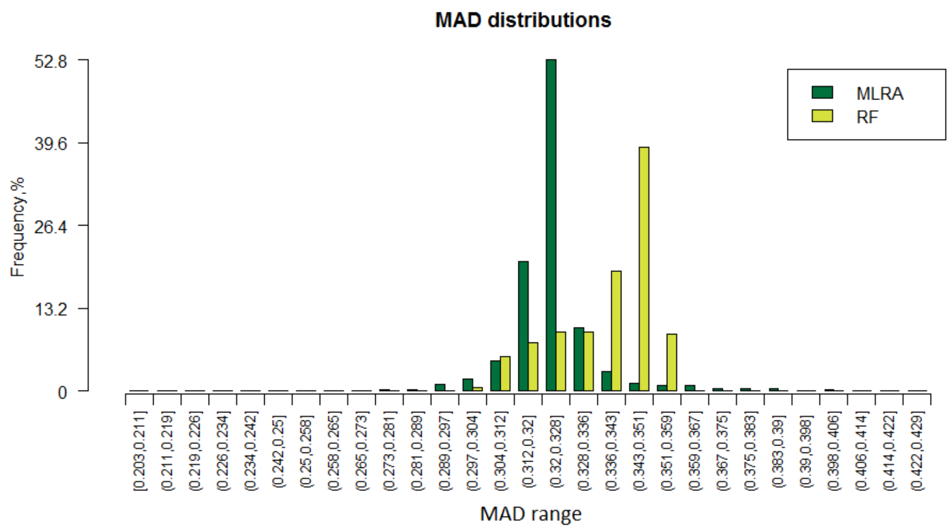


Fig. 2. MAD distribution shows that lazy algorithm allows one to obtain prediction error relatively lower than the one with random forest tunings

= 1.5%, number of iterations = 10, subsample size = 1%, allowed dropout = 0.1. The result was compared to benchmarks represented by random forest tunings.

5 Conclusion

Formal concept analysis offers attractive instruments to extract knowledge from data as soon as intents of concepts can be considered as associative rules. FCA-based algorithms are suitable for predictive modeling in areas where model interpretation clarity is of great priority. However, in previous work only classification problems were considered, while continuous target attribute prediction, i.e. regression problem, was out of focus. In this paper, we adjusted the lazy algorithm [3,16], so that it can perform continuous predictions. The adjustment required a new definition of an augmented interval pattern structure. In effect, the adjusted algorithm mines the premises (with target attribute expected distribution) that are relevant to test object and then prediction is performed based on the target attribute distribution, e.g. based on the median of the distribution.

We applied the algorithm to delinquent corporate clients loans in order to predict the recovery rate for each loan. The data we used comes from the pilot project with one of the top-10 banks in Russia. Mean absolute deviation was chosen as accuracy metric of the algorithm. We performed simple grid search by running the algorithm with different parameter values and chose the tuning with the lowest value of the metric. The classification accuracy of the algorithm was compared to some benchamrks represented by random forests, as soon as their

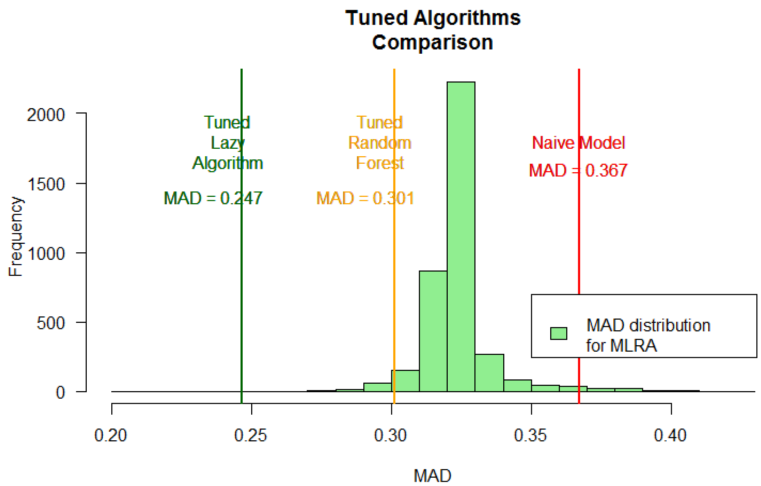


Fig. 3. MAD distribution of the lazy algorithm versus best tuning for random forest and naive model MAD

predictions are based on combination of simple rules, too. The proposed modified lazy regression algorithm showed comparable quality in the greater number of runs and in certain parameters area it outperformed random forests. However, it has to be mentioned that the number of parameters is greater in our algorithm what, in effect, results in greater algorithm complexity and greater degrees of freedom. As an area for further research, one can consider keeping the density function h not only for target attribute in premises, but also make use of those density functions for explanatory attributes as well. It can be expected, that if the premises are mined not only based on allowed dropout and alpha-threshold parameters, but also based on some properties of attributes distribution, then the premises will be more relevant for the test objects and will produce more accurate predictions for target attribute.

References

1. B. Ganter and S.O. 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. B. Ganter, R. Wille: Formal Concept Analysis: Mathematical Foundations. Springer, Berlin, 1999.
3. S.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., vol. 8251 of Lecture Notes in Computer Science, pp. 30–39, Springer, 2013.

4. L.Thomas, D. Edelman, J.Crook: Credit Scoring and Its Applications, Monographs on Mathematical Modeling and Computation, SIAM: Philadelphia, pp. 107–117, 2002.
5. N. Siddiqi: Credit Risk Scorecards: Developing and Implementing Intelligent Credit Scoring, WILEY,ISBN: 978-0-471-75451-0, 2005.
6. B. Baesens, T.V. 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.
7. A. Ghodselahi: A Hybrid Support Vector Machine Ensemble Model for Credit Scoring, International Journal of Computer Applications (0975 – 8887), Volume 17–No.5, 2011.
8. L. Yu, S.Wang and K.K. Lai: 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, 2009.
9. T.V. Gestel, B. Baesens, J.A. Suykens, Poel V.D.V., B.: Bayesian kernel based classification for financial distress detection. European journal of operational research. vol. 172. pp. 979-1003, 2006.
10. P. Ravi 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, pp. 1-28, 2007
11. S.O. Kuznetsov and M.V. Samokhin: Learning closed sets of labeled graphs for chemical applications., in ILP, Stefan Kramer and Bernhard Pfahringer, Eds., vol. 3625 of Lecture Notes in Computer Science, pp. 190– 208, Springer, 2005
12. SAS Institute Inc., Developing Credit Scorecards Using Credit Scoring for SAS® Enterprise Miner™ 12.1, Cary, NC: SAS Institute Inc., 2012.
13. R.R. Hocking: The Analysis and Selection of Variables in Linear Regression, Biometrics, 32, 1976.
14. M. Kaytoue, S.O. Kuznetsov, A. Napoli, and S. Duplessis: Mining gene expression data with pattern structures in formal concept analysis, Information Sciences, vol. 181, no. 10, pp. 1989–2001, 2011.
15. A. Veloso and W. Meira, Jr.: Demand-Driven Associative Classification., Springer, 2011.
16. A. Masyutin, Y. Kashnitsky, S.O. Kuznetsov: Lazy Classification with Interval Pattern Structures: Application to Credit Scoring, In: Proc. 4th International Workshop "What can FCA do for Artificial Intelligence?" (FCA4AI 2015), pp. 43-54, 2015.
17. E.I. Altman, A. Resti and A. Sironi: The link between default and recovery rates: effects on the procyclicality of regulatory capital ratios", BIS Working Papers No 113, Monetary and Economic Department, 2002.

Appendix

Algorithm 1 Lazy Regression by Sub-Samples with Continuous Target Attribute

---

Input: $\{G\}$ – numerical context with explanatory variables x and single target attribute y.
M – number of explanatory 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 context G which represents the objects which falsify the premise (parameter).
g_t – test object.
Output: $\widehat{\delta_y(g_t)}$ – prediction that is produced by the voting rule.
P – a set of premises, i.e. associative rules produced for the test object g_t. P can be empty.

for *iter* from 1 to *num.iter* **do**
  A_0=random.sample(G,size=$sub.smpl \cdot |G|$) — mine α - weak premises with ω-allowed dropout.
  $d_0 = \delta_x(g_1) \sqcap ... \sqcap \delta_x(g_s) \sqcap \delta_x(g_t),\ g_s \in A_0 \forall s$
  Compute empirical density function h_0 for d_{0y}.
  $A_1 = d_0^{\diamond}$
  if $1 - \frac{|\{g \in A_1 | d_0 y^{min} - \omega(m - d_0 y^{min}) \leq \delta_y(g) \leq d_0 y^{max} + \omega(d_0 y^{max} - m)\}|}{|A_1|} \leq \alpha$ **then**
    Update empirical density function h_0 to h_1 based on new values of target attribute in A_1.
    Add (d_0, h_1) to the set P of α - weak premises with ω-allowed dropout.
  else
    Do nothing
  end if
end for
Define weighting scheme w_a, w_{pen}.
Calculate the median for mixture of distribution functions h_p based on d_{py}, $\forall p \in P$.

$$\widehat{\delta_y(g_t)} = \frac{\sum_{p \in P} \sum_{g \in A_p} \delta_y(g) \cdot w(g)}{\sum_{p \in P} \sum_{g \in A_p} w(g)}$$

If P is empty, then calculate the median for target attributes of all $g \in G$ (naive prediction).

Unified External Data Access Implementation in Formal Concept Analysis Research Toolbox

Alexey A. Neznanov, Andrey A. Parinov

National Research University Higher School of Economics,
20 Myasnitskaya Ulitsa, Moscow, 101000, Russia
ANeznanov@hse.ru, AParinov@hse.ru

Abstract. Formal Concept Analysis (FCA) provides mathematical models, methods and algorithms for data analysis. However, by now there is no easily available program system, which would provide data analyst with unified, intelligible and transparent access to various external data sources with large amount of heterogeneous data for subsequent FCA-based knowledge discovery. The lack of such tools complicates spreading FCA methods among big data analysts and miners of unstructured data. In this paper, we describe advances and new functionality in external data querying and preprocessing subsystems of Formal Concept Analysis Research Toolbox (FCART), which helps processing data of different types in a unified way.

Keywords: Formal Concept Analysis, Knowledge Extraction, Data Mining, Text Mining, Social Network Mining, Software.

1 Introduction

By now, mathematical models of Formal Concept Analysis (FCA) [1] are widely used for solving various problems of Knowledge Discovery and Artificial Intelligence [2,3]. Some systems use FCA ideas implicitly, by processing closed sets of attributes or objects. In this paper we will concentrate on explicit implementation of FCA methods as part of analyst's workflow in a software system. Three main problems here can be stated as follows.

1. How to generate suitable input data for FCA-based methods?
2. How to keep initial data properties and metadata while analyzing object-attribute representation by FCA-based methods?
3. How to combat high computational complexity of FCA-based methods in the context of an integral analyst's workflow?

Around the middle of the last decade, there were several successful implementations for transforming a relatively small formal context into a line diagram and computing implications and association rules. In [4] we have discussed well-known FCA-based tools, like ConExp [5], Conexp-clj [6], Galicia [7], Tockit [8],

© Marianne Huchard, Sergei O. Kuznetsov (Eds.): CLA 2016, pp. 285–296, ISBN 978-5-600-01454-1, National Research University Higher School of Economics, 2016.

ToscanaJ [9], Lattice Miner [10], OpenFCA [11], Coron [12,13], Cubist [14]. Most of the reviewed software tools are local applications that require initial data in the form of binary or many-valued context in one of the common formats (CSV, CXT or other). Thus, such programs can not be used on the stage of data gathering and preprocessing, but we should include input formats of those programs in the list of supported formats for future integration.

Formal Concept Analysis Research Toolbox (FCART) [15] supports iterative methodology of data mining and knowledge discovery. One of the goals of developing FCART is to create a system for handy analysis of heterogeneous data gathered from external data sources, e.g. SQL databases, NoSql databases and Social Network Services. FCART was successfully applied to analyzing data in medicine, criminalistics, sociology, and trend detection [3, 15].

In previous papers, we have described the system architecture, main workflow and stages of data extraction from various external sources. Here we would like to describe recent progress in the distributed version [16] of FCART and its Intermediate Data Storage (IDS) subsystem. This progress is mainly related to new functionality in data preprocessing.

2 Problems description

Data analysis is highly dependent on preprocessing, i.e., transformation from the source data format to the target data format, in which data are processed. An important functionality of any data analysis system is to support analyst in preprocessing transformations, making them transparent and easy.

2.1 A gap between FCA analytical artifacts and external data

From an analyst point of view, there is a gap between FCA analytical artifacts workflow and data and the legacy data. Fig. 1 illustrates this gap between “analyzable” and “external” data. It should be emphasized that it is not a gap between concrete data formats or access protocols, it is the gap in ways of thinking and knowledge representation.

The four main questions of object-attribute-value (or object-attribute) representation of data are trivial: 1) What are objects? 2) What are attributes? 3) How do we gather values of attributes? 4) How do we interpret values of attributes?

However, such questions bring into being a great many technological questions. For now, we can observe specific data preprocessing techniques of concrete data analysis projects. Can we propose fully unified approach? In general, the answer is *no*. However, we can try to adapt some common techniques for most popular classes of initial data formats and external data sources. On the one hand, we can see appearance of such terms as “Data Tidying” [17] for some “human readable” variants of ETL (Extract-Transform-Load) processes. On the other hand, there are continuous development of such monster software as Oracle Data Integrator Enterprise Edition [18] or less monstrous Microsoft PowerBI [19].

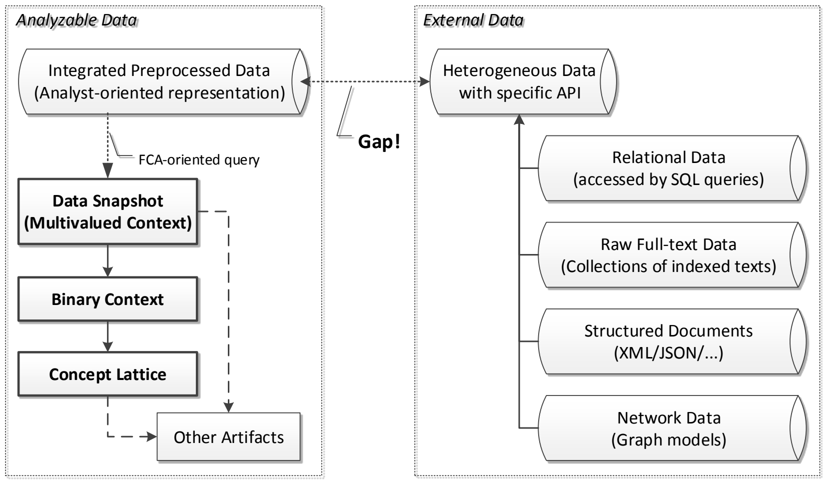


Fig. 1. The gap between "analyzable" and "external" data

FCA-based analytics tools impose additional requirements because of:

1. Basic FCA algorithms have very high computational complexity.
2. Big concept lattices are not suitable for interactive processing and visualizing.

There were many attempts to adapt FCA-based methods for complex tasks. For example, building Iceberg Lattices [20, 21, 22], visualizing other fragments of lattices, using incremental lattice construction algorithms.

2.2 A few comments about methodology

The core of the FCART supports knowledge discovery techniques, based on Formal Concept Analysis, clustering, multimodal clustering, pattern structures and others. From the analyst point of view, basic *FCA workflow* in FCART has four stages. On each stage, a user has the ability to import/export every artifact or add it to a report.

1. The filling Intermediate Data Storage (IDS) of FCART from various external SQL, XML/JSON and other data sources (querying external source is described by an *External Data Query Description* – EDQD). EDQD can be constructed by some visual External Data Browser (see later).
2. The loading a data snapshot from the IDS into an analytic session (a snapshot is described by a *Snapshot Profile*). A data snapshot is a data table with annotated structured and text attributes (a many-valued context) loaded in the system by accessing IDS.
3. The transforming a snapshot to a binary context (a transformation is described by a *Scaling Query*).
4. The building and visualizing formal concept lattice and other artifacts based on the binary context within an analytic session.

Later in this paper we will discuss mainly the first stage and using EDQDs. Hadley Wickham in [17] wrote: “there has been little research on how to make data cleaning as easy and effective as possible”. The second and the third stages with example of Snapshot Profile construction were initially described in [23].

2.3 FCART architecture and the role of the IDS

The current distributed version of FCART consists of the following four parts:

1. *FCART AuthServer* for authentication and authorization, as well as integration of algorithmic and storage resources.
2. *FCART Intermediate Data Storage* (IDS) for storage and preprocessing (initial converting, indexing of text fields, etc.) of big datasets.
3. *FCART Thick Client* (Client) for interactive data processing and visualization in integrated graphical multi-document user interface.
4. *FCART Web-based solvers* (Web-Solvers) for implementing independent resource-intensive computations.

IDS plays important role in effectiveness of whole data analysis process because all data from external data storages, session data and intermediate analytic artifacts saves in IDS. All interaction between user and external data storages goes through the IDS. All interactions between Client, Web-Solvers and IDS go through a RESTful Web-API. The http-request to the IDS web-service constructed from two parts: prefix part and command part. Prefix part contains domain name and local path (e.g. http://zeus2.hse.ru:8444/). The command part describes what IDS has to do and represents some function of the Web-API. Using web-service commands, FCART client can query data from external data storages in uniform and efficient way.

Early we already have implemented populating IDS from external data sources, but now we extend the set of providers and improve data providers’ EDQDs.

3 Worlds of data and data representation in IDS

Readers may have noticed that a simplest case of legacy data for object-attribute representation is *relational data* that meet the well-known conditions of E. Codd [24]. In this case we have virtually multivalued context. In the current state of Internet development we should distinguish at least the following types of data sources:

1. Relational data sources (directly queried by SQL).
2. NoSQL document collections (queried by XQuery or similar query languages).
3. Text collections with full-text index (queried by special full-text queries).
4. Social Network Services (with plenty of different access APIs).

3.1 Data integration problems and FCART Intermediate Data Storage

Documents are kept in many data formats (only ISO standards describe more then 400 formats, for example see [25]). After open data revolution [26] and Web infrastructure integration in Internet [27], most popular formats for information interchange are Comma Separate Values (CSV) [28], Extensible Markup Language (XML) [29] and JavaScript Object Notation (JSON) [30]. Extensible Markup Language (XML) is a markup language that defines a set of rules for encoding documents in a format that is both human-readable and machine-readable. The main goal of XML is to store metainformation with information itself. Hundreds of document formats using XML syntax have been developed, including RSS, Atom, SOAP, and XHTML. XML-based formats have become the default for many office-productivity tools, including Microsoft Office (Office Open XML), OpenOffice.org and LibreOffice (OpenDocument), and Apple's iWork. XML has also been employed as the base language for communication protocols, such as XMPP.

XML and its extensions have regularly been criticized for verbosity and complexity. JSON is lightweight alternative which focus on representing (serializing) programming language level objects with complex data structures rather than documents, which may contain both highly structured and relatively unstructured content. JSON is an open standard format that uses human-readable text to transmit data objects consisting of attribute–value pairs.

Traditional relational databases are not convenient for fast processing of big amounts of unstructured textual datasets with metadata. Document-oriented databases operating with documents in XML or JSON format are successfully used for storing, retrieving and managing big amounts of textual data in last decade. Both FCART IDS and FCART Client can handle XML and JSON documents as input format. XML format is complex and relatively hard to process at the same time. JSON format is more easy to use and lightweight. FCART uses JSON internally as a main format for data serialization and intercomponent communication.

3.2 Main terms and terminology problems

Preliminary problem of the discussed concepts is a terminological one. Table 1 illustrates the difference in approaches to defining terms for basic data-related concepts in SQL Servers (as stated in the SQL ISO Standard [31]), full-text indexing systems (as stated in the Elasticsearch reference [32]) and document-oriented NoSQL storages (as stated in the MongoDB reference [33]). One can look at term “Index” as a good example of polysemantic word. Graph-oriented databases use absolutely different terms for atomic elements (vertices, nodes, link, edges, arcs) and data structures, that reflect incidence, adjacency neighbourhood, etc.

In IDS we use data representation in form of “Databases” with hierarchical structure of “Collections” of JSON “Documents”. Each of the Documents may contains heterogeneous “Fields”. Each Collection can possess metadata, which describes structure of Documents and data types of Fields using JSON Schema [34].

It is very powerful approach, which gives an ability to validate Documents with compound data types.

Table 1. Real cases of different terms usage in popular data storages

	SQL Server	Elasticsearch	Mongo DB
1	Database (non normative)	**Index**	Database
2	Scheme	Mapping	--
3	Table	Type	Collection
4	**Index**	--	**Index**
5	Record/Row (Tuple)	Document (JSON)	Document (BSON)
6	Field/Column (Attribute)	Field	Field
7	Primary key	Document Id	`_id` field
8	Shard	Shard	Shard

4 External Data Queries in IDS

Extracting data is complicated by the fact that any Internet data source may have its own API. For example, we consider Social Network Services as a data source. That is why one needs mechanism to describe how data should be extracted, preprocessed and stored in IDS. For unified data access we developed External Data Query Description (EDQD) language. Each EDQD is a JSON formatted document. It aims to unify access to different data sources. By using EDQD FCART represents data from various data sources as a IDS Collection. There is no way to create a single query with fixed fields that would be with various data sources because each data source has its own set of functions, its own API. However, we developed the most common EDQD types and field set for mentioned above data sources types (Fig. 2).

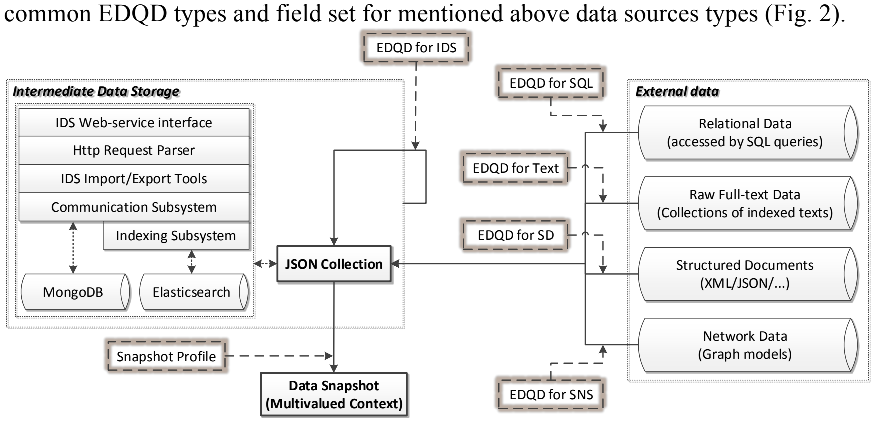


Fig. 2. The architecture of IDS and EDQDs for different data sources

Each field is a JSON object. An EDQD query includes next fields:

- ID – this field describes a unique identifier GUID.
- TYPE – this field describes type of data source. For now, FCART supports next types: “FS Folder”, “FS File”, “TSQL”, “REST”, “SOAP”, ”Facebook”, etc. Also EDQD with type “IDS” can refer to documents which are already stored in IDS.
- URI [27] – this field describes path to the data source. It is optional.
- CS – this field describes connection string [35]. It is optional.
- QUERY – this field describes query to the data source.
- TARGET – this field describes target data storage. For now, it’s can be set to the values – URI file path, “IDS”.
- TRANSFORMATION – this field describes type of source field, connection between source field and fields of target data storage; and all transformation such as scaling, indexing, etc.

This are the common fields for all EDQD queries. Below we have described specific for data sources EDQD queries.

Creating EDQD is a complex task, which needs visual tool for development. For now, External Data Browsers have been prepared to help user constructing EDQD for local JSON/XML files, unstructured text files and SQL data sources. Other types of EDQD can be created via direct JSON editing.

4.1 Query to a SQL data source

EDQD for a SQL data source is the most straightforward. For now, IDS supports connection to the Microsoft SQL 2014 (and its earlier versions) and Postgres 9.5.2 (and its earlier versions). EDQD for SQL has the following fields:

- “ID” – GUID (Globally Unique Identifier).
- “TYPE” – DBMS Type. Can be “TSQL” or “PS”.
- “URI” – This field is empty for that EDQD query type.
- “CS” – Connection String.
- “QUERY” – TSQL or PL-SQL query.
- “TRANSFORMATION” – For now it describes mapping a column name to a target field path in JSON document.

Example of EDQD for query to the instance of Microsoft SQL 2014:

```
{ “ID”: {6F9619FF-8B86-D011-B42D-00CF4FC964FF},
    ”TYPE”:”sql-server-2014”,
    “URI”:””,
    “CS”:”DataSource=190.190.200.100,1433;
Server=myServerName\myInstance;    Ininial    Catalog=myDataBase;    User
ID=myUsername; Password=myPassword;”,
    “QUERY”:”SELECT  column_name  FROM  table1  INNER  JOIN  table2  ON
table1.column_name=table2.column_name;”
    “TRANSFORMATION”: { “field”:{
      “name”: “name”,
      “target_field”:”user_name”}
}
```

4.2 Query to unstructured text files

EDQD for Text (a collection of files with unstructured text) provides ability to extract and transform data from unstructured texts. To analyze data from unstructured data file we need to create an inverted index. Using inverted index reduces searching time for every text word.

The inverted index is a central component of indexing search engine. A goal of a search engine implementation is to optimize the speed of the query: find the documents where word X occurs. Once a forward index is developed, which stores lists of words per document, it is next inverted to develop an inverted index. Querying the forward index would require sequential iteration through each document and each word to verify a matching document. The time, memory, and processing resources to perform such a query are not always technically realistic. Instead of listing the words per document in the forward index, the inverted index data structure is developed which lists the documents per word [36].

To create inverted index, we use full-text search engine. For now, there are many full-text search engines, which provides rapid search, complicated query language and REST interface. Solr [37] and Elasticsearch [32] are the most powerful and popular search engines for now. In the previous paper [38] we described detailed comparison of Solr and Elasticsearch as basis for implementing full-text manipulating part of IDS. In the paper we showed speed advantage of Elasticsearch in situation of indexing and inserting data at the same time. It's important to search text data because unstructured text is often a part of other data types, e.g., structured documents (CSV, JSON, XML), documents extracted from social network services (user information, posts).

Initial sets of automatically extracted keywords may be very big. We can have additional instruments for such sparse contexts with many uniform attributes like sorting and searching attributes (Fig. 3) or analyzing attributes usage statistics. But more proper way to generate initial context is using adjustable query.

Example of EDQD for a query to a folder with text files:

```
{ "ID": {6F9619FF-8B86-D011-B42D-00CF4FC964FF},
  "TYPE":"FS folder"
  "URI":"file://localhost/c|/source/"
  "CS":""
  "QUERY":""
  "TARGET":" file://localhost/c|/target"
  "TRANSFORMATION": {
    "field": {
    "name": ""
    "target_field":"body",
    "type":"text",
    "indexing": True}
}
```

Field "Transformation" is the most interesting part of EDQD for a local text-files folder. "Name" refers to a field of result IDS document is affected. "Target_field" describes name in IDS document. "Type" describes type of the source field. The value of the EDQD field "Type" determines operations and transformations which are applicable to a document field. By now, FCART supports an indexing operation on the "text" type. By default, the value of "Indexing" field is False.

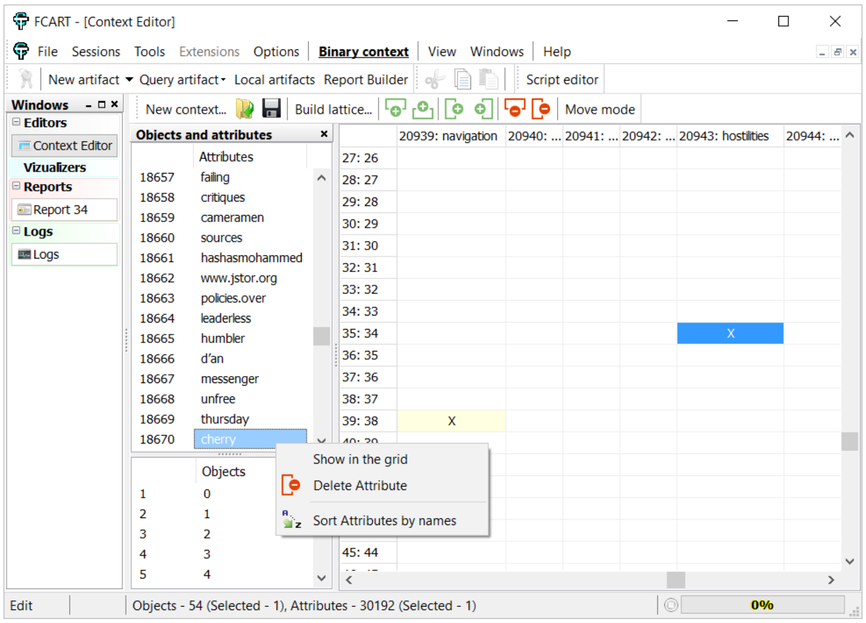


Fig. 3. Example of sparse context for all keywords in 54 documents – 30192 attributes!

4.3 Query to an XML, JSON or CSV data file

EDQD for a collection of XML [29], JSON [30] or CSV [28] data files is similar to EDQD for the text file. In case of XML one should specify path to the source document field in the EDQD field “source_path” according to the XPath standard [39]. In case of JSON one should use the JSONPath draft [40]. In the case of CSV one should use fragment identifier according RFC-7111 [41].

Example of EDQD for a query to a folder with collection of JSON data files:

```
{ "ID":
  "TYPE": "JSON Folder"
  "URI": "file://localhost/c|/source/"
  "CS ": ""
  "QUERY": ""
  "TARGET": "file://localhost/c|/target"
  "TRANSFORMATION": {
    "field": {
    "name": "body"
    "source_path": "/store/book/title"
    "target_field": "/body" }
}
```

4.4 EDQD query to a web-service

EDQD for web-service interface provides ability to extract data from web-service. In the current version FCART supports REST and SOAP interfaces. EDQD for web-service has the following fields:

- "ID" – GUID (Globally Unique Identifier).
- "TYPE" – Web-service type. Can be "REST" or "SOAP".
- "URI" – URI of web-service.
- "CS" - This field is empty.
- "QUERY" – JSON document which contains query.
- "TRANSFORMATION" – JSON document which describes field mapping.

Example of EDQD query to an Elasticsearch REST interface:

```
{ "ID": {6F9619FF-8B86-D011-B42D-00CF4FC964FF},
  "TYPE":"REST",
  "URI":"http://elasticsearch:1234/index_name/mapping_name/",
  "CS": "",
  "QUERY": "{
    "query": {
    "bool": {
    "must": [
      {"match":{"address": "mill" }},
      {"match":{"address": "lane" }}
    ] }}}",
  "TRANSFORMATION": "{
    "field":{
    "target_field": "body",
    "indexing": True}"
}
```

REST interfaces can iterate set of elements, which are returned by query. Query field contains JSON document written on Elasticsearch query language (https://www.elastic.co/guide/en/elasticsearch/reference/current/index.html).

Transformation field contains JSON document, which describes target field and preprocessing operation. The current version of FCART supports only indexing operation.

4.5 EDQD query to a Social Network Service

Social networks services have special API types. FCART processes the most common part of social networks services analysis. EDQD represents a user profile of Social Network Service as a hierarchical JSON document that has next fields:

```
{"user":{
     "id":".."
     "path": ".."
     "user_info": ".."
     "friend:[…]
     }
"post":{
     "time":
```

```
    "body: ".."
    "title" ".."
    "tags":[ "..",…, ".."]}
}
```


Besides full-text queries, analyst can query neighborhood of a person, e.g. friends or colleagues. In the current version FCART server supports connection to Livejournal, Twitter, and Facebook. Example of extracting posts from third neighborhood layer of the person:


```
{ "ID": {6F9619FF-8B86-D011-B42D-00CF4FC964FF},
  "TYPE": "facebook"
  "URI": "https://www.facebook.com/someuser/"
  "CS ": ""
  "QUERY": {
    "path": "friend/post/body",
    "layer": "3",
    "BEGIN": "2005-08-09T18:31:42-03",
    "END":"",
    "COUNT":100 },
  "TARGET":"IDS"
  "TRANSFORMATION": {
    "field":{
    "type":"post",
    "target_field":"body",
    "indexing": True }
}
```


5 Discussion and future work

In this paper, main problems of external data access in FCA-based analytics software were addressed and some real cases were examined while implementing new functionality in the FCART system. The demo version of FCART client is available at https://cs.hse.ru/en/ai/issa/proj_fcart and the test version of the IDS Web-service is available at http://zeus2.hse.ru:8444.

For FCA-based data analysis fundamental requirements for software are as follows:

1. The ability to merge heterogeneous data sources in a query to external data.
2. The ability to cache frequent queries.
3. The automatic populating of query metadata.
4. The support of many formats of local data files to communicate with other software tools easily.
5. The support of apriori prescribed constraints on FCA algorithms and visualization schemes.
6. The availability of common and special "quick and dirty" methods of query result visualization with low computational complexity.

When prototyping clinical decision support system components, we have realized the importance of having local and web-based versions of the preprocessing tools. So unification of external data access tools is the first step in satisfying informal analysts' wishes. We also understand importance of other subsystems, including efficient data transformation algorithms, dashboards, etc. However, without unified and reproducible access to initial data no one can build real data analysis workflow.

Improved mechanisms of query data work faster, more intelligible and provide necessary information to data analyst. The next steps in our development process are adding new External Data Browsers, increasing efficiency of EDQD processing and standardizing new API for running Web-Solvers inside IDS instead of Client.

Acknowledgments This work was carried out by the authors within the project "Mining Data with Complex Structure and Semantic Technologies" supported by the Basic Research Program of the National Research University Higher School of Economics.

References

1. Ganter, B., Wille, R. Formal Concept Analysis: Mathematical Foundations, Springer, 1999.
2. Poelmans, J., Kuznetsov, S.O., Ignatov, D.I., Dedene, G. Formal Concept Analysis in knowledge processing: A survey on models and techniques. Expert Systems with Applications, 40(16), 2013 pp. 6601-6623.
3. Poelmans, J., Ignatov, D.I., Kuznetsov, S.O., Dedene, G. Formal Concept Analysis in knowledge processing: A survey on applications. Expert Systems with Applications, 40(16), 2013, pp. 6538-6560.
4. Neznanov, A.A., Parinov, A.A. About Universality and Flexibility of FCA-based Software Tools. Proceedings of the 3rd International Workshop "What can FCA do for Artificial Intelligence?", 2014, pp. 59-65.
5. Yevtushenko, S.A. System of data analysis "Concept Explorer" (In Russian). 7th National Conference on Artificial Intelligence (KII-2000), Russia, 2000, pp. 127-134.
6. Conexp-clj (http://daniel.kxpq.de/math/conexp-clj)
7. Valtchev, P., Grosser, D., Roume, C. Hacene, M.R. GALICIA: an open platform for lattices, in Using Conceptual Structures // Contributions to the 11th International Conference on Conceptual Structures (ICCS'03), 2003, pp. 241-254.
8. Tockit: Framework for Conceptual Knowledge Processing (http://www.tockit.org)
9. Becker, P., Hereth, J., Stumme, G. ToscanaJ: An Open Source Tool for Qualitative Data Analysis. Workshop FCAKDD of the 15th European Conference on Artificial Intelligence (ECAI-2002). Lyon, France, 2002.
10. Lahcen, B., Kwuida, L. Lattice Miner: A Tool for Concept Lattice Construction and Exploration. Suplementary Proceeding of International Conference on Formal Concept Analysis (ICFCA'10), 2010.
11. Borza, P.V., Sabou, O., Sacarea, C. OpenFCA, an open source formal concept analysis toolbox. IEEE International Conference on Automation Quality and Testing Robotics (AQTR), 2010, pp. 1-5.
12. Szathmary, L. The Coron Data Mining Platform (http://coron.loria.fr)
13. Szathmary, L., Napoli, A., Kuznetsov, S.O. ZART: A Multifunctional Itemset Mining Algorithm. 5th International Conference on Concept Lattices and Their Applications (CLA'07), pp.26-37.
14. Cubist Project (http://www.cubist-project.eu)

15. Neznanov, A., Ilvovsky, D., Kuznetsov, S. FCART: A New FCA-based System for Data Analysis and Knowledge Discovery. Contributions to the 11th International Conference on Formal Concept Analysis, Dresden, 2013, pp. 31-44.
16. Neznanov, A.A., Parinov, A.A. Distributed architecture of data analysis system based on formal concept analysis approach. Studies in Computational Intelligence, 616, 2016, pp. 265-271.
17. Wickham, H. Tidy Data. Journal of Statistical Software, 59 (10), 2014, pp. 1-23.
18. Oracle Data Integrator Enterprise Edition (http://www.oracle.com/us/products/middleware/data-integration/enterprise-edition/overview/index.html)
19. Microsoft PowerBI (http://powerbi.microsoft.com)
20. Rouane, M.H., Nehme, K., Valtchev, P., Godin, R. On-line maintenance of iceberg concept lattices. ICCS-2004, 2004.
21. Szathmary, L., Valtchev, P., Napoli, A., Godin, R., Boc, A., Makarenkov, V. Fast Mining of Iceberg Lattices: A Modular Approach Using Generators. CLA-2011, 2011, pp. 191-206.
22. Smith, D.T. A Formal Concept Analysis Approach to Data Mining: The QuICL Algorithm for Fast Iceberg Lattice Construction. Computer and Inf. Science, 7(1), 2014, pp. 10-32.
23. Neznanov, A.A., Kuznetsov, S.O. Information Retrieval and Knowledge Discovery with FCART. Proceedings of the Workshop Formal Concept Analysis Meets Information Retrieval (FCAIR 2013), CEUR-977, 2013, pp. 74-82.
24. Codd, E. F. Further Normalization of the Data Base Relational Model. Data Base Systems: Courant Computer Science Symposia Series 6, Prentice-Hall, 1972, pp. 33-64.
25. ISO Standards catalogue 35.040: Character sets and information coding (http://iso.org/iso/products/standards/catalogue_ics_browse.htm?ICS1=35&ICS2=040)
26. Open Data Watch (http://opendatawatch.com)
27. RFC-3986: Uniform Resource Identifier (URI): Generic Syntax (https://tools.ietf.org/html/rfc3986)
28. RFC-4180: Common Format and MIME Type for Comma-Separated Values (CSV) Files (http://tools.ietf.org/html/rfc4180)The Connection Strings Reference (http://www.connectionstrings.com)
29. Extensible Markup Language (XML) 1.0 (Fifth Edition), W3C Recommendation, 2008 (https://www.w3.org/TR/REC-xml)
30. Standard ECMA-404: The JSON Data Interchange Format, 2013 (http://www.ecma-international.org/publications/standards/Ecma-404.htm)
31. ISO/IEC 9075-2:2011 Information technology - Database languages - SQL - Part 2: Foundation (https://www.iso.org/obp/ui/#iso:std:iso-iec:9075:-2:ed-4:v1:en)
32. Elasticsearch Reference (https://www.elastic.co/guide/en/elasticsearch/reference)
33. MongoDB Reference Glossary (https://docs.mongodb.org/manual/reference/glossary)
34. The home of JSON Schema (http://json-schema.org)
35. The Connection Strings Reference (http://www.connectionstrings.com)
36. The inverted index description (https://en.wikipedia.org/wiki/Inverted_index)
37. Apache Solr (http://lucene.apache.org/solr)
38. Neznanov, A.A. Parinov, A.A. Full-text Search in Intermediate Data Storage of FCART. RuZA2015 Workshop, 2015. (http://ceur-ws.org/Vol-1552/paper7.pdf)
39. XPath Syntax (http://www.w3schools.com/xsl/xpath_syntax.asp)
40. JSONPath (http://goessner.net/articles/JsonPath)
41. RFC-7111: URI Fragment Identifiers for the text/csv Media Type (https://tools.ietf.org/html/rfc7111)

Exploring Temporal Data Using Relational Concept Analysis: An Application to Hydroecology

Cristina Nica[1], Agnès Braud[1], Xavier Dolques[1]
Marianne Huchard[2], Florence Le Ber[1]

[1] ICube, University of Strasbourg, CNRS, ENGEES
firstname.lastname@engees.unistra.fr,agnes.braud@unistra.fr
http://icube-sdc.unistra.fr
[2] LIRMM, University of Montpellier, CNRS
huchard@lirmm.fr
https://www.lirmm.fr

Abstract. This paper presents an approach for mining temporal data, based on Relational Concept Analysis (RCA), that has been developed for a real world application. Our data are sequential samples of biological and physico-chemical parameters taken from watercourses. Our aim is to reveal meaningful relations between the two types of parameters. To this end, we propose a comprehensive temporal data mining process starting by using RCA on an ad hoc temporal data model. The results of RCA are converted into closed partially ordered patterns to provide experts with a synthetic representation of the information contained in the lattice family. Patterns can also be filtered with various measures, exploiting the notion of temporal objects. The process is assessed through some quantitative statistics and qualitative interpretations resulting from experiments carried out on hydroecological datasets.

1 Introduction

Exploring temporal datasets is a major challenge in current research and various methods have therefore been proposed since the 90's [1]. It is worth pointing out that temporal data are relational, so that relational methods [6] can be useful to respect their relational structure, e.g. [9]. In particular, Relational Concept Analysis (RCA, [16]) allows to classify relational data and provides hierarchical results which facilitates the analysis step.

Based on these properties, we propose to use RCA for exploring sequential datasets from the hydroecological domain. These datasets were collected during the Fresqueau project[3] that focused on methods for assessing the quality of watercourses. The collected data represent biological (Bio) and physico-chemical (PhC) samples taken at fixed points (river sites) and repeated in time. Both parameters are used by the experts to determine the quality of watercourses.

[3] http://engees-fresqueau.unistra.fr/presentation.php?lang=en

© Marianne Huchard, Sergei O. Kuznetsov (Eds.): CLA 2016, pp. 299–310,
ISBN 978-5-600-01454-1, National Research University Higher School of Economics, 2016.

Therefore, a global assessment of the temporal relationship between PhC and Bio parameters is needed. To this end, preprocessings of the raw sequential data allow to build a qualitative temporal model that can be used to apply RCA on these data. The RCA result is a family of lattices that can be navigated by the users. The users can select relevant navigation paths through the lattices (starting from concepts in a main lattice) by applying measures of interest based on the concept extents, that can be linked to geographical information in our application. Furthermore, in order to help their analysis and to synthetize the results, we propose to transform those concepts within closed partially ordered patterns (cpo-patterns, [5]), i.e. directed acyclic graphs where vertices are labelled with information extracted from the concepts out of the family of lattices. Since concepts can be more or less general or specific, the extracted patterns can be classified within three types, according to the number of vertices that are labelled with general information. Then the users can choose to select and to navigate general or specific paths in the lattices.

The paper is structured as follows. Section 2 presents basic definitions and related work. Section 3 describes the hydroecological data and their preprocessing while the RCA process is detailed in Section 4. Section 5 introduces some measures of interest dealing with the temporal dimension of obtained concepts. Section 6 presents cpo-patterns in order to help the analysis. Section 7 describes and discusses the experimental results carried out on Fresqueau datasets. Section 8 concludes and gives a few perspectives of this work.

2 Basics and Related Work

Relational Concept Analysis (RCA, [16]) extends Formal Concept Analysis (FCA [11]) to classify sets of objects described by attributes and relations, thus allowing to discover knowledge patterns and implication rules in relational datasets. RCA applies iteratively FCA on a Relational Context Family (RCF) that is constituted of a set $\mathcal{K}$ of object-attribute contexts and a set $\mathcal{R}$ of object-object contexts. $\mathcal{K}$ contains n object-attribute formal contexts $K_i = (G_i, M_i, I_i), i \in \{1, ..., n\}$. $\mathcal{R}$ contains m object-object relational contexts $R_j = (G_k, G_l, r_j), j \in \{1, ..., m\}$, where G_k, called the domain of the relation, and G_l, called the range of the relation, are respectively the sets of objects of K_k and K_l, and $r_j \subseteq G_k \times G_l, k, l \in \{1, ..., n\}$. At each step, object-attribute contexts are extended with relational attributes taking the syntactic form $qr_j(C)$, where q is a quantifier, r_j is a relation and $C = (X, Y)$ is a concept where X is a subset of objects from the range of r_j. This paper uses the *existential* quantifier: $\exists r_j(C)$ is an attribute of $o \in G_k$ if $r_j(o) \cap X \neq \emptyset$. RCA process consists in applying FCA first on each object-attribute context of an RCF, and then iteratively on each object-attribute context extended by the relational attributes created using the concepts from the previous step. The RCA result is obtained when the family of lattices of two consecutive steps are isomorphic and the contexts are unchanged.

RCA has been applied to various data, e.g. for software model analysis and re-engineering [2]. To our knowledge, this is the first time that RCA is used to explore sequential datasets. There are, however, various related FCA approaches. [18] introduced Temporal Concept Analysis where objects are characterized with a date and a state (i.e. a set of attributes). Data are merged into a single context, and the resulting concept lattice is analysed thanks to the date element in the concepts, so that temporal relations between concepts are actually revealed by the analyst. This approach has been used to analyse sequential data about crime suspects [15]. In our RCA approach, the temporal relation between dates is considered as an object-object relation and it links concepts from several lattices. In [8], sequential datasets are processed without involving any partial order. In [5], closed subsequences are mined and then grouped in a lattice similar to a concept lattice. In [4], sequential data are mapped onto pattern structures whose projections are used to build a pattern concept lattice. The authors combine the stability of concepts and the projections of pattern structures in order to select relevant patterns.

Besides, there exist various methods to explore qualitative sequential data. Indeed, sequential pattern mining is an active research area, in relation to the exponential growth of temporal and spatio-temporal databases. Sequential patterns have been introduced by [1] and used for different purposes. Such an approach has been developed within the Fresqueau project and focused on closed po-patterns, which were selected through various measures [7]. Indeed, selecting relevant results is a main challenge for all approaches dealing with large datasets. In FCA, the most used measures for selecting relevant concepts are stability [13], probability and separation [12]. Unfortunately, these measures are not able to take into account the specific structure of concepts built on temporal objects. We thus propose to use specific measures, as detailed in Section 5.

3 Context and Data Preprocessing

In the Fresqueau project, the analysed data cover various compartments such as physico-chemistry, hydrobiology, hydromorphology and land use (as described in [3]). Here, we try to tackle the following issue by means of RCA: *Can experts explain values of biological parameters from PhC values occuring in past months and thus improve the global assessment of the quality of watercourse ecosystems?*

To answer this question we should mention that the quality of watercourses is determined by the Bio parameters (e.g. Standardised Global Biological Index (IBGN), Biological Index of Diatoms (IBD) and Fish Biotic Index (IPR)). Hence, the objects of interest from our work are the Bio samples and we want to assess, over a period of time, the impact of PhC macro-parameters (e.g. Nitrogen (AZOT), Phosphor (PHOS) and Particulate Matter (PAES)) on Bio ones.

Table 1(a) illustrates a small raw sequential dataset of Bio and PhC samples taken from a *site* (e.g. S1) corresponding to a river segment. A set of sites constitutes a *geographical area*. A data sequence is a chronologically ordered set of PhC samples with a Bio one at the end, all taken from the same site. This raw sequential dataset shows measurements made only for IBGN Bio parameter and for four PhC parameters namely Ammonium (NH_4^+), Kjeldahl Nitrogen

Table 1: Small example of raw and corresponding preprocessed sequential dataset.

(a) Raw Sequential Data

Site	**Date**	NH_4^+	NKJ	NO_2^-	PO_4^{3-}	**IBGN**
S1	08/05	-	-	-	-	10
	06/05	0.004	-	0.012	0.035	-
	09/04	-	-	-	-	8
	08/04	-	1.414	-	-	-
	01/04	0.043	0.146	0.421	-	-

(b) Preprocessed Sequential Data

Site	**Date**	**AZOT**	**PHOS**	**IBGN**
S1	08/05	-	-	Yellow
	06/05	Blue	Green	-
	09/04	-	-	Orange
	08/04	Green	-	-

(NKJ), Nitrite (NO_2^-) and Orthophosphate (PO_4^{3-}). For instance, 0.043 mg/l of NH_4^+ is measured on 01/04, i.e. January 2004, for the site $S1$. An IBGN score of 8/20 is measured on September 2004 for the same site.

The raw sequential dataset contains only numerical values. For mining such data, we transform them by applying discretization and selection processes based on domain knowledge. The discretization aims at converting numerical values into qualitative ones. To this end, we use qualitative values for Bio and PhC parameters that are provided by the SEQ-Eau[4] standard. Both types of parameters have five qualitative values, namely *very good*, *good*, *medium*, *bad* and *very bad* represented respectively by the colors *blue*, *green*, *yellow*, *orange* and *red*. In addition, SEQ-Eau standard groups PhC parameters into macro-parameters. For example, NH_4^+, NKJ and NO_2^- are grouped into AZOT macro-parameter. The selection process considers only relevant data by defining some constraints based on expert advice. For instance, the only analysed PhC samples are those taken within *4 months* before a Bio parameter, from the same site.

Table 1(b) shows the preprocessed sequential dataset ready to be mined using RCA. This sequential dataset is obtained by applying the discretization and selection processes to the raw sequential dataset illustrated in Tab. 1(a). It is worth pointing out that the preprocessed sequential dataset is significantly small compared to the raw one thanks to the macro-parameters and the limited analysed period of time.

4 Temporal Relational Analysis

The sequential dataset is structured following the schema depicted in Fig. 1. The four rectangles represent the four sets of objects we manipulate: Bio samples, PhC samples, Bio parameters and PhC parameters. The links between Bio/PhC samples and PhC samples are defined by the temporal binary relation *is preceded by* (denoted by ipb). This temporal relation associates one sample to another one if the first sample is preceded in time by the second one, on the same site. There

[4] `http://rhin-meuse.eaufrance.fr/IMG/pdf/grilles-seq-eau-v2.pdf`

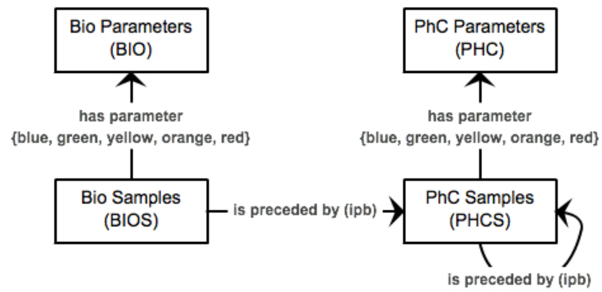


Fig. 1: The modelling of the hydroecological sequential dataset.

is no temporal binary relation between Bio samples since in this work we evaluate the impact of physico-chemistry on biology. The Bio/PhC samples are described only by the qualitative relations *has parameter blue/green/yellow/orange/red* that link the Bio/PhC samples with the measured Bio/PhC parameters. For instance, *has parameter green* links the PhC samples taken from S1 on 08/04 (Tab. 1(b)) with AZOT PhC parameter.

Following the temporal data model illustrated in Fig. 1, we build the RCF depicted in Tab. 2 for a small hydroecological sequential dataset. The tables KPHC (PhC parameters), KBIOS (Bio samples) and KPHCS (PhC samples) represent object-attribute contexts. There is no object-attribute context for Bio parameters because each dataset is restricted to one value of one parameter (here IBGN red). KBIOS and KPHCS have no column since the samples are only described using the qualitative relations. The tables RPHCS-ipb-PHCS, RBIOS-ipb-PHCS , RbPHC and RgPHC represent object-object contexts. In these object-object contexts, a row is an object from the domain of the relation, a column is an object from the range of the relation and a cross indicates a link between two objects. For example, RPHCS-ipb-PHCS defines the temporal relations (ipb) between PhC samples and has KPHCS both as domain and range. RbPHC defines the qualitative relations between PhC samples and PhC parameters that have the blue (b) qualitative value.

Figure 2 represents the family of concept lattices obtained by applying RCA on the RCF illustrated in Tab. 2. There are three lattices, one for each formal context: $\mathcal{L}_{\texttt{KPHCS}}$ (PhC samples, Fig. 2(a)), $\mathcal{L}_{\texttt{KPHC}}$ (PhC parameters, Fig. 2(b)) and $\mathcal{L}_{\texttt{KBIOS}}$ (Bio samples, Fig. 2(c)). Each concept is represented by a box structured from top to bottom as follows: concept name, simplified intent and simplified extent. As said before, we have used the existential quantifier to build relational attributes. For instance, the intent of C_KPHCS_2 from concept $\mathcal{L}_{\texttt{KPHCS}}$ contains the relational attribute ∃RgPHC(C_KPHC_1) inherited from concept C_KPHCS_5. This relational attribute is common to all PhC samples that measure a green PHOS parameter, which represents the extent of concept C_KPHC_1 shown in Fig. 2(b).

Table 2: RCF composed of object-attribute contexts: KPHC, KBIOS and KPHCS; temporal object-object contexts: RBIOS-ipb-PHCS and RPHCS-ipb-PHCS; qualitative object-object contexts: RbPHC and RgPHC.

object-attribute contexts

KPHC	AZOT	PHOS
AZOT	×	
PHOS		×

KBIOS
S1_20/01
S1_28/12
S2_30/02

KPHCS
S1_17/01
S1_10/01
S1_25/12
S2_28/02
S2_20/02

object-object contexts

RBIOS-ipb-PHCS	S1_17/01	S1_10/01	S1_25/12	S2_28/02	S2_20/02
S1_20/01	×	×			
S1_28/12			×		
S2_30/02				×	×

RPHCS-ipb-PHCS	S1_17/01	S1_10/01	S1_25/12	S2_28/02	S2_20/02
S1_17/01		×			
S1_10/01					
S1_25/12					
S2_28/02					×
S2_20/02					

RbPHC	AZOT	PHOS
S1_17/01		
S1_10/01	×	
S1_25/12	×	
S2_28/02		
S2_20/02	×	

RgPHC	AZOT	PHOS
S1_17/01	×	×
S1_10/01		
S1_25/12		×
S2_28/02	×	×
S2_20/02		×

The navigation amongst the lattices shown in Fig. 2 follows the concepts used to build relational attributes. For example, the aforementioned relational attribute ∃RgPHC(C_KPHC_1) allows us to navigate from concept C_KPHCS_2 out of $\mathcal{L}_{\texttt{KPHCS}}$ to concept C_KPHC_1 out of $\mathcal{L}_{\texttt{KPHC}}$.

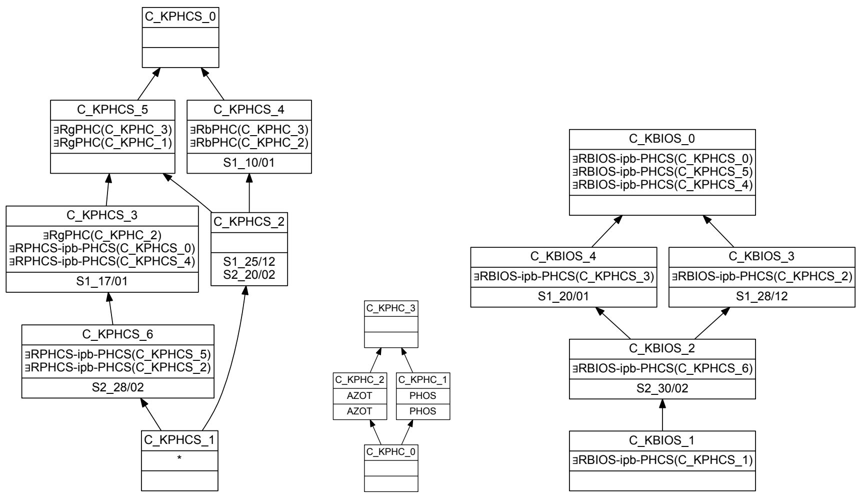
5 Measures of Interest for Temporal Concepts

To analyse the results of the RCA process, experts start from a *main lattice*, here the lattice $\mathcal{L}_{\texttt{KBIOS}}$, and navigate through the relational attributes linking concepts of different lattices. Besides, since RCA process can produce a large number of interrelated concepts, depending on the dataset volume and characteristics, some interestingness measures are required to select relevant concepts from where to start the navigation.

Such measures should take into account the specificity of concepts built on temporal objects, whereas well-known measures (e.g. concept stability) fit basic concepts. For example, Fig. 3 depicts two concept extents where the temporal objects are the Bio samples. Both concepts – that we call temporal concepts – have the same number of Bio samples and they cover the same geographical area. If two Bio samples are deleted, following the idea of stability measure, one of the site $S2$ and one of $S3$, then both concepts still have the same number of Bio samples but they cover different river sites.

To overcome this limitation, we introduce below an approach based on the distribution of temporal concept extents. The main idea in our method states that a concept is relevant if it is frequent and related to many sites where Bio samples are evenly distributed amongst these sites. Accordingly, we try to find temporal concepts whose intents represent universally available regularities in the studied geographical area. In our example, both concepts have the same frequency (7 samples), but the distribution is different: Concept_1 is more relevant than Concept_2.

Let (X, Y) be a formal concept of the main lattice, then its extent X is a set of temporal objects – or pairs – $(Object, Date)$. If the value of $Object$ is not identical for all the pairs, then the pairs can be grouped into categories by objects. We accordingly define $\bar{X}$ which represents the set of distinct objects



(a) $\mathcal{L}_{\texttt{KPHCS}}$ (b) $\mathcal{L}_{\texttt{KPHC}}$ (c) $\mathcal{L}_{\texttt{KBIOS}}$

Fig. 2: The family of concept lattices obtained by applying RCA on the RCF given in Tab.2. The $*$ symbol represents all the relational attributes of `KPHCS`.

from X pairs: $\bar{X} = \{o \in O | \exists t \in T, (o,t) \in X\}$, where O is the object set and T the set of dates.

Definition 1 (Absolute Frequency (ϕ_o)). *Let $C = (X, Y)$ be a temporal concept and o an object of $\bar{X}$. The absolute frequency of o in C, denoted ϕ_o, is equal to the number of distinct pairs of X where o occurs. $\bar{X}_\phi = \{(o, \phi_o) | o \in \bar{X}\}$.*

In our example (Fig. 3), $\bar{X}_1 = \bar{X}_2 = \{S1, S2, S3\}$. `Concept_1` has $\bar{X}_{1\phi} = \{(S1,3), (S2,3), (S3,1)\}$ and `Concept_2` has $\bar{X}_{2\phi} = \{(S1,5), (S2,1), (S3,1)\}$.

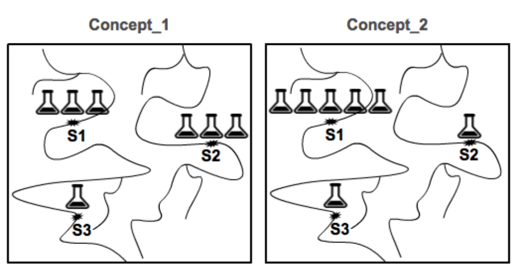


Fig. 3: Bio samples distribution by sites for two concept extents.

Definition 2 (Support and Richness (ρ)). *The support of a concept* (X, Y) *corresponds to the number of pairs* $(Object, Date)$ *out of* X*. Its richness, represented by* ρ*, is defined as the cardinality of* $\bar{X}$*.*

Definition 3 (Distribution index (IQV)). *The distribution of a concept* (X, Y) *describes the number of times each object out of* $\bar{X}$ *occurs in* X *and it is measured by the Index of Qualitative Variation (IQV, [10]). IQV is based on the ratio of observed differences in* $\bar{X}_\phi$ *to the total number of possible differences within* $\bar{X}_\phi$ $(\rho > 1)$*.*

$$IQV = \frac{\rho\left(|X|^2 - \sum_{i=1}^{\rho} {\phi_{o_i}}^2\right)}{|X|^2(\rho - 1)} \tag{1}$$

If $\rho = 1$*,* $IQV = 0$*.*

Our choice of IQV stems from the observation that the objects of $\bar{X}$ do not have an intrinsic ordering. Thus, measuring their distribution using the IQV [10] seems interesting. The IQV ranges from 0 to 1. When all pairs of X contain the same object, there is no diversity and the IQV is 0. In contrast, when there are different objects and all pairs of $\bar{X}_\phi$ have equal ϕ_o, there is even distribution and the IQV is 1.

Returning to our example (Fig. 3), both concepts have support $|X_1| = |X_2| = 7$ and richness $\rho_1 = \rho_2 = 3$. For `Concept_1` the distribution is $IQV_1 = \frac{3\left[7^2 - \left(3^2+3^2+1^2\right)\right]}{7^2(3-1)} = 0.91$ and for `Concept_2` $IQV_2 = 0.67$. Hence, `Concept_1` is computed as more relevant than `Concept_2` since its objects (Bio samples) are better distributed amongst the sites.

6 CPO-patterns for Helping Expert Analysis

Since our aim is to facilitate the analysis work, we propose, in addition to the selection of relevant concepts, to convert those concepts into cpo-patterns. Indeed cpo-patterns are structures with a graphical representation easy to read and understand (e.g. Fig. 4). The expert can choose a cpo-pattern that highlights interesting, surprising knowledge, and deepen the analysis by exploring the area in the lattice surrounding the corresponding concept. Thus, starting from the family of lattices built using RCA, we extract cpo-patterns following the approach proposed in [14]. It is worth pointing out that there is a cpo-pattern for each concept out of the lattice corresponding to the objects of interest for the study, i.e. $\mathcal{L}_{\texttt{KBIOS}}$ in our work.

Formally, let $\mathcal{I} = \{I_1, I_2, ..., I_m\}$ be a set of *items*. An *itemset* IS is a non empty, unordered, set of items, $IS = (I_{j_1}...I_{j_k})$ where $I_{j_i} \in \mathcal{I}$. Let $\mathcal{IS}$ be the set of all itemsets built from $\mathcal{I}$. A *sequence* S is a non empty ordered list of itemsets, $S = \langle IS_1 IS_2 ... IS_p\rangle$ where $IS_j \in \mathcal{IS}$. The sequence S is a *subsequence* of another sequence $S' = \langle IS'_1 IS'_2 ... IS'_q\rangle$, denoted as $S \preceq_s S'$, if $p \leq q$ and if there are integers $j_1 < j_2 < ... < j_k < ... < j_p$ such that $IS_1 \subseteq IS'_{j_1}, IS_2 \subseteq IS'_{j_2}, ..., IS_p \subseteq IS'_{j_p}$. Sequential patterns have been defined by [1] as frequent

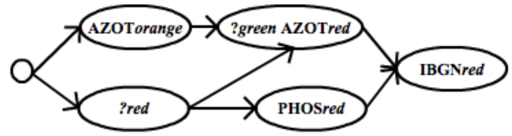


Fig. 4: Hybrid cpo-pattern: each vertex corresponds to a set of parameter values, edges represent the temporal relation, e.g. $\texttt{IBGN}_{red}$ is preceded by $\texttt{PHOS}_{red}$ that is preceded by $\texttt{?}_{red}$; this notation means that a PhC parameter with a red quality has been measured.

subsequences found in a sequence database. A *po-pattern* is a directed acyclic graph $G = (\mathcal{V}, \mathcal{E}, l)$. $\mathcal{V}$ is the set of vertices, $\mathcal{E}$ is a set of directed edges such that $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$, and l is a labelling function mapping each vertex to an itemset. A partial order can be defined on G as follows: for all $\{u, v\} \in \mathcal{V}^2$, $u < v$ if there is a directed path from u to v. However, if there is no directed path from u to v, these elements are not comparable. Each path of the graph is a sequential pattern as defined before. The set of paths in G is denoted by $\mathcal{P}_G$. A po-pattern is associated to the set of sequences $\mathcal{S}_G$ that contains all paths of $\mathcal{P}_G$. Furthermore, let G and G' be two po-patterns with $\mathcal{P}_G$ and $\mathcal{P}_{G'}$ their sets of paths. G is a sub po-pattern of G', denoted by $G \preceq_g G'$, if $\forall M \in \mathcal{P}_G, \exists M' \in \mathcal{P}'_G$ such that $M \preceq_s M'$. A po-pattern G is closed, denoted *cpo-pattern*, if there exists no po-pattern G' such that $G \prec_g G'$ with $\mathcal{S}_G = \mathcal{S}_{G'}$.

As described in [14], thanks to the hierarchical structure of the RCA results, more or less accurate cpo-patterns are extracted. Based on their accuracy, three types of cpo-patterns could be defined: abstract, hybrid and concrete. Firstly, the *abstract* cpo-pattern represents an imprecise common trend of the analysed data. Secondly, the *hybrid* one, depicted in Fig. 4, corresponds to a more or less accurate common trend of the analysed data. Finally, the *concrete* cpo-pattern designates an accurate common trend of the analysed data.

7 Experiments and Discussion

The experiments are carried out on a MacBook Pro with a 2.9 GHz Intel Core i7, 8GB DDR3 RAM running OS X 10.9.5. RCA is applied using the RCAExplore[5] tool. For the extraction and selection of cpo-patterns we have developed an algorithm in Java 8 based on Java Collections Framework and Lambda Expressions.

Three sequential datasets (each dataset concerns only one Bio parameter having the *yellow* quality) from the Fresqueau project are analysed: IBD_{yellow}, IPR_{yellow} and $IBGN_{yellow}$. These datasets are interesting since the *yellow* quality of watercourses represents a median area between good ecological status and bad ecological status of watercourses. Other quality values have also been analysed but are not presented here. The objective is to extract more or less accurate

[5] http://dolques.free.fr/rcaexplore

Table 3: The results of mining the Fresqueau datasets. `Bio` and `PhC Samples` are the number of analysed samples; `Output` is the number of concepts from the main lattice ($\mathcal{L}_{\texttt{KBIOS}}$) and the lattice of PhC samples ($\mathcal{L}_{\texttt{KPHCS}}$); `CPO-patterns` is the number of the extracted cpo-patterns; `Execution Time` in seconds.

Datasets				RCA		Extraction			Execution Time
Index	Quality	Samples		Output		CPO-patterns			RCA & Extraction
		Bio	PhC	$\mathcal{L}_{\texttt{KBIOS}}$	$\mathcal{L}_{\texttt{KPHCS}}$	Concrete	Abstract	Hybrid	
IPR	yellow	80	194	35699	39605	433	3388	31877	593
IBD				32146	20947	503	1444	30198	115
IBGN				9414	11580	305	815	8293	32

cpo-patterns representing frequent PhC trends of watercourses common in many sites. To this end, the datasets are preprocessed and temporally modelled as described in Sections 3 and 4. The temporal relational analysis relies on the IceBerg algorithm [17], which result is a concept lattice of frequent closed itemsets. A 10% threshold is used only for the input of Bio samples (it corresponds to the lattice of Bio samples that covers the objects of interest from our work). The choice of this value allows us to focus on the cpo-patterns that describe many sites.

Table 3 shows some quantitative statistics regarding the temporal relational analysis and the extraction of cpo-patterns. The results in `Output` column show that the number of extracted concepts for the IBGN dataset is about 3 times smaller than the number of extracted concepts for the IPR and IBD datasets. This reveals greater heterogeneity in IPR and IBD datasets in contrast with IBGN. Consequently, cpo-patterns linking PhC and IBGN Bio parameters represent more examples and will provide more reliable forecasts of the *yellow* quality of watercourses.

The `CPO-patterns` columns represent the different types of extracted cpo-patterns and illustrate their quite large number that has to be reduced. To this end, we select relevant cpo-patterns based on the support, richness and distribution of the associated concepts (see Section 5). Figure 5 shows three scatter-plots (for the three sets of extracted concrete cpo-patterns in Tab. 3) of the *distribution index (IQV)* with respect to the *support*. The diameter of the circles is proportional to the *richness*. The user can first explore a few selected cpo-patterns based on high thresholds for these measures. Then he/she can follow the cpo-pattern hierarchy to deepen the analysis, as described below, or select more cpo-patterns based on lower thresholds. For example, by defining two thresholds $\theta_{IQV} = 0.98$ and $\theta_{Support} = 25$, the top-6 (IBGN), the top-26 (IBD) and the top-30 (IPR) best distributed and most frequent cpo-patterns are selected. Focusing on IBD, if the thresholds are e.g. $\theta_{IQV} = 0.98$ and $\theta_{Support} = 20$, 52 cpo-patterns are selected. These cpo-patterns cover various numbers of sampling sites, and thus more or less extensive geographical areas. To select greater or smaller areas, the cpo-patterns are ranked by analysing the diameter of the circles.

The qualitative interpretation of the extracted cpo-patterns was performed by an hydroecologist. In Fig. 6 is an interesting excerpt from the main lattice of $IBGN_{yellow}$ dataset. This group of cpo-patterns is subsumed by the abstract cpo-pattern of C_KBIOS_868 (support = 28) that represents the less accurate common trend: *often before yellow IBGN are sampled simultaneously a green PhC parameter and another yellow PhC parameter*. Figure 6 also emphasizes the well-known correspondence between MOOX (organic matter pollutions) quality classes and IBGN ones: a *yellow* MOOX appears in the *yellow* IBGN cpo-pattern, which is associated to C_KBIOS_595. The concepts C_KBIOS_720, C_KBIOS_550 and C_KBIOS_400 highlight the impact of phosphorus pollution (PHOS) on macro-invertebrates (IBGN) that is a lesser-known fact.

Moreover, in Fig. 6 two benefits of exploring sequential data by means of RCA are observed. The first one is the generalisation order regarding the structure of the extracted cpo-patterns. For example, the structure of C_KBIOS_400 cpo-pattern is more specific than the structure of its ancestor cpo-patterns, i.e. there exist a projection from its ancestor cpo-patterns into C_KBIOS_400 cpo-pattern. The second benefit is the generalisation of items. For instance, the C_KBIOS_550 cpo-pattern reveals the rule $\{\mathtt{PAES}_{green}, \mathtt{PHOS}_{yellow}\} \rightarrow \{\mathtt{IBGN}_{yellow}\}$ that is a specialisation of the rule revealed by the C_KBIOS_720 cpo-pattern, that is $\{?_{green}, \mathtt{PHOS}_{yellow}\} \rightarrow \{\mathtt{IBGN}_{yellow}\}$. These properties are useful for the expert who can navigate from specific to general patterns or vice versa.

8 Conclusion

We have introduced an original approach for exploring temporal data using RCA. Given a hydroecological dataset, where data represent Bio or PhC samples measured at a given time in a certain site, we find hierarchies of more or less general cpo-patterns that summarize the impact of PhC parameters on Bio ones. A comprehensive process for mining sequential datasets has been proposed: 1) preprocessing of the raw data based on domain knowledge, 2) relational analysis of the preprocessed data based on an original temporal data model, 3) selection of temporal concepts using the distribution, the richness and the support measures, and 4) extraction of cpo-patterns by navigating amongst temporal concepts (step

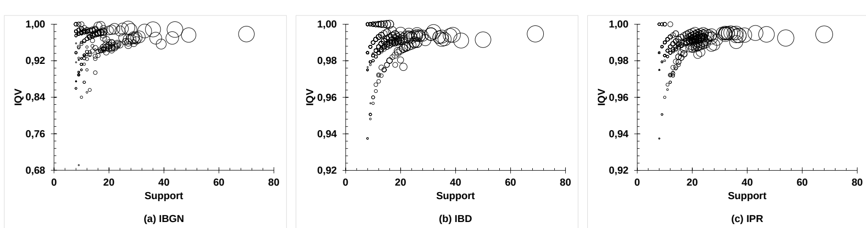


Fig. 5: Concrete cpo-patterns by distribution index, support and richness of the associated concepts.

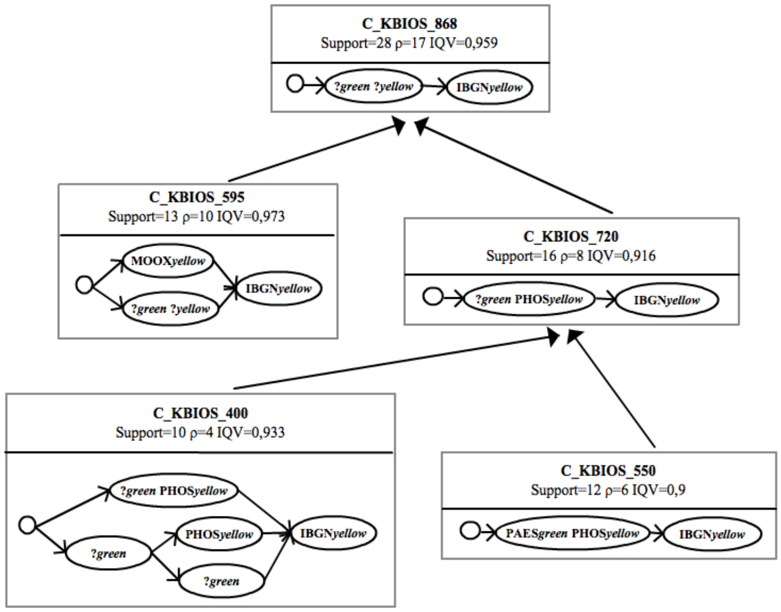


Fig. 6: Excerpt from a hierarchy of cpo-patterns (IBGN *yellow*).

detailed in [14]). Our method has been applied to sequential datasets from the Fresqueau project.

The main benefits of our approach are as follows. Using RCA produces hierarchical concepts, while cpo-patterns synthetize complex navigation paths, both facilitating the expert analysis. Furthermore, the proposed measures on temporal concepts are useful to select relevant information in our application.

In the future, we plan to apply our approach on other relational datasets. This will require to deeply investigate the behaviour of our measures and maybe to find other methods for selecting the extracted cpo-patterns.

References

1. Agrawal, R., Srikant, R.: Mining sequential patterns. In: Int. Conference on Data Engineering. pp. 3–14 (1995)
2. Arévalo, G., Falleri, J.R., Huchard, M., Nebut, C.: Building abstractions in class models: Formal concept analysis in a model-driven approach. In: MoDELS 2006. pp. 513–527 (2006)
3. Bimonte, S., Boulil, K., Braud, A., Bringay, S., Cernesson, F., Dolques, X., Fabrègue, M., Grac, C., Lalande, N., Le Ber, F., Teisseire, M.: Un système décisionnel pour l'analyse de la qualité des eaux de rivières. Ingénierie des Systèmes d'Information 20(3), 143–167 (2015)

4. 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. International Journal of General Systems 45, 135–159 (2016)
5. Casas-Garriga, G.: Summarizing sequential data with closed partial orders. In: 2005 SIAM Int. Conference on Data Mining. pp. 380–391 (2005)
6. Džeroski, S.: Relational data mining. In: Maimon, O., Rokach, L. (eds.) Data Mining and Knowledge Discovery Handbook, pp. 869–898. Springer (2005)
7. Fabrègue, M., Braud, A., Bringay, S., Grac, C., Le Ber, F., Levet, D., Teisseire, M.: Discriminant temporal patterns for linking physico-chemistry and biology in hydro-ecosystem assessment. Ecological Informatics 24, 210–221 (2014)
8. Ferré, S.: The efficient computation of complete and concise substring scales with suffix trees. In: Formal Concept Analysis, pp. 98–113. Springer (2007)
9. Ferreira, C.A., Gama, J., Costa, V.S.: Exploring multi-relational temporal databases with a propositional sequence miner. Progress in AI 4(1-2), 11–20 (2015)
10. Frankfort-Nachmias, C., Leon-Guerrero, A.: Social Statistics for a Diverse Society, chap. Measures of Variability. SAGE Publications (2010)
11. Ganter, B., Wille, R.: Formal Concept Analysis: Mathematical Foundations. Springer (1999)
12. Klimushkin, M., Obiedkov, S., Roth, C.: Approaches to the selection of relevant concepts in the case of noisy data. In: Formal Concept Analysis, pp. 255–266. Springer (2010)
13. Kuznetsov, S.O.: On stability of a formal concept. Annals of Mathematics and Artificial Intelligence 49(1-4), 101–115 (2007)
14. Nica, C., Braud, A., Dolques, X., Huchard, M., Le Ber, F.: Extracting Hierarchies of Closed Partially-Ordered Patterns using Relational Concept Analysis. In: International Conference on Conceptual Structures, ICCS'2016, Annecy, France. pp. 1–14. Springer (2016)
15. Poelmans, J., Elzinga, P., Viaene, S., Dedene, G.: A Method based on Temporal Concept Analysis for Detecting and Profiling Human Trafficking Suspects. In: Artificial Intelligence and Applications, AIA 2010, Innsbruck, Austria. pp. 1–9 (2010)
16. Rouane-Hacene, M., Huchard, M., Napoli, A., Valtchev, P.: Relational concept analysis: Mining concept lattices from multi-relational data. Annals of Mathematics and Artificial Intelligence 67(1), 81–108 (2013)
17. Stumme, G.: Efficient data mining based on formal concept analysis. In: Database and Expert Systems Applications, pp. 534–546. Springer (2002)
18. Wolff, K.E.: Temporal Concept Analysis. In: ICCS-01 Workshop on Concept Lattice for KDD, 9th Int. Conference on Conceptual Structures. pp. 91–107 (2001)

Data Retrieval and Noise Reduction by Fuzzy Associative Memories

Irina Perfilieva, Marek Vajgl

University of Ostrava,
Institute for Research and Applications of Fuzzy Modeling,
Centre of Excellence IT4Innovations,
30. dubna 22, 701 03 Ostrava 1,
Czech Republic
E-mails: Irina.Perfilieva,Marek.Vajgl@osu.cz

Abstract. A novel theoretical background of fuzzy associative memories (FAM) is proposed. A framework of formal concept analysis is used for a new working theory of FAM. Two principal activities of FAM are formalized : data retrieval and noise reduction. It is shown that the problem of data retrieval is connected with solvability and eigen sets of a certain system of fuzzy relation equations. The differentiation of FAM models according to their ability to reduce noise is defined. It is shown how the choice of formal context determines a type of noise that can be reduced by the corresponding retrieval mechanism. Finally, we propose a fast algorithm of data retrieval.

1 Introduction

One of the first publications devoted to *fuzzy associative memories* (FAM) has been made by Kosko - [6]. The FAM has been characterized as a single-layer feedforward neural net performing a nonlinear matrix-vector multiplication. This approach was later extended with the purpose to increase the storage capacity (e.g. [4]). Significant progress was achieved by introduction of learning implication rules [3, 5], that afterwards led to *implicative fuzzy associative memory* (IFAM) with *implicative fuzzy learning*. A justification of validity of a certain IFAM model was discussed in [14] where the characterization of one type of suppressed noise - eroded - was proposed.

In the current contribution, we use the framework of formal concept analysis [17] and propose a working theory of FAM. Let us remark that the language and technique of formal concept analysis is used in other theories, e.g., fuzzy property-oriented concept lattices, as well as in applications such as modeling and processing of incomplete knowledge in information systems [1, 2].

We formalize two principal activities of FAM: data retrieval and noise reduction. We use the proposed formalism and show that the problem of data retrieval is connected with solvability and eigen sets of a certain system of fuzzy relation equations. We differentiate FAM models according to their ability to

© Marianne Huchard, Sergei O. Kuznetsov (Eds.): CLA 2016, pp. 313–324,
ISBN 978-5-600-01454-1, National Research University Higher School of Economics, 2016.

reduce noise and show how the choice of formal context determines a type of noise that can be reduced by the corresponding retrieval mechanism.

From the technical point of view, we extend the theory of FAM by using a general algebraic structure instead of a specific one (Łukasiewicz algebra in [14]) and by enlarging the set of autoregressive fuzzy associative memory models (AFAM). We propose a formal characterization of AFAM models and show the way of various modifications. We analyze the retrieval mechanism of AFAM and its ability to remove noise. We show that the larger is the amount of noise, the greater should be the fuzzy relation that models retrieval with noise reduction. Further, we show how the type of removable noise depends on which type of AFAM models is applied. Finally, we construct a fast algorithm of data retrieval and give illustration of the noise reduction.

2 Preliminaries

2.1 Implicative fuzzy associative memory

In this Section, we discuss underlying assumptions related to autoregressive fuzzy associative memories (AFAM) using denotation in [14]. We formalize the retrieval mechanism and the problem of noise reduction.

Let us propose the following formalization of AFAM. A database

$$D = \{\mathbf{x}^1, \dots, \mathbf{x}^p\},$$

of objects (images, patterns, signals, texts, etc.) is represented by normal fuzzy sets such that every $\mathbf{x}^k$, $k = 1, \dots, p$, is a map $\mathbf{x}^k : X \to [0, 1]$, where $X = \{u_1, \dots, u_n\}$ is a universe. The problem is to find a model of D together with a retrieval mechanism such that every element $\mathbf{x}^k \in D$ can be successfully retrieved even from its noisy version.

According to [14], a *model of AFAM for database D* can be identified with the 3-tuple $(W, \theta, \boxdot)$, consisting of a fuzzy relation $W : X \times X \to [0, 1]$, a bias vector $\theta \in [0, 1]^n$, and a set-relation composition $\boxdot : [0, 1]^{n\times n} \times [0, 1]^n \times [0, 1]^n \to [0, 1]^n$, such that for all $\mathbf{x}^k \in D$,

$$\mathbf{x}^k = W \boxdot \mathbf{x}^k \vee \theta,\ k = 1, \dots, p. \tag{1}$$

We say that (1) represents a *retrieval mechanism in AFAM* and that this mechanism *reduces noise*, if

$$\mathbf{x}^k = W \boxdot \widetilde{\mathbf{x}}^k \vee \theta, k = 1, \dots, p, \tag{2}$$

where $\widetilde{\mathbf{x}}^k$ is a noisy version of $\mathbf{x}^k$.

In [14], an implicative model of AFAM has been proposed. The model uses Łukasiewicz algebra of operations on $[0, 1]$, so that fuzzy relation W is expressed in the implicative form

$$W(u_i, u_j) = \bigwedge_{k=1}^{p} (\mathbf{x}^k(u_i) \to \mathbf{x}^k(u_j)), \tag{3}$$

and other constituents are as follows:

$$\theta_i = \bigwedge_{k=1}^{p} \mathbf{x}^k(u_i),\ i = 1, \dots, n,$$

$\boxdot$ is the sup $-\otimes$-composition.

For a given input $\mathbf{x}$, AFAM returns the output in accordance with (1) and the assignment given above, so that

$$W \boxdot \mathbf{x} \vee \theta = \bigvee_{i=1}^{n} (W(u_i, u_j) \otimes \mathbf{x}(u_i)) \vee \theta.$$

The proposed in [14] AFAM is able to reduce an eroded noise, i.e. if $\mathbf{x}$ is less than some database element, say $\mathbf{x}^k$, then the retrieved output is close to $\mathbf{x}^k$ as well.

2.2 Algebraic background

In this Section, we step aside from the terminology of associative memories and introduce the algebraic background of what will be proposed as a new model of AFAM.

Let $\mathcal{L} = \langle L, \vee, \wedge, *, \rightarrow, 0, 1\rangle$ be a fixed, complete, integral, residuated, commutative l-monoid (*a complete residuated lattice*). We remind the main characteristics of this structure: $\langle L, \vee, \wedge, 0, 1\rangle$ is a complete bounded lattice, $\langle L, *, \rightarrow, 1\rangle$ is a residuated, commutative monoid.

Let X be a non-empty set, L^X a class of (L-valued) *fuzzy sets* on X and $L^{X\times X}$ a class of (L-valued) *fuzzy relations* on X. Fuzzy sets and fuzzy relations are identified with their membership functions, i.e. elements from L^X and $L^{X\times X}$, respectively. A fuzzy set A is *normal* if there exists $x_A \in X$ such that $A(x_A) = 1$. The (ordinary) set $Core(A) = \{x \in X \mid A(x) = 1\}$ is the *core* of the normal fuzzy set A. Fuzzy sets $A \in L^X$ and $B \in L^X$ are *equal* ($A = B$), if for all $x \in X$, $A(x) = B(x)$. A fuzzy set $A \in L^X$ is *less than or equal* to a fuzzy set $B \in L^X$ ($A \leq B$), if for all $x \in X$, $A(x) \leq B(x)$.

The lattice operations $\vee$ and $\wedge$ induce the union and intersection of fuzzy sets, respectively. The binary operations $*$ and $\rightarrow$ of $\mathcal{L}$ are used for set-relation compositions of the types sup $-*$ or inf $- \rightarrow$ that are usually denoted by $\circ$ and $\triangleright$ where

$$(R \circ A)(y) = \bigvee_{x \in X} (R(x, y) * A(x)),$$

and

$$(R \triangleright A)(y) = \bigwedge_{x \in X} (R(x, y) \rightarrow A(x)).$$

We say that compositions $\circ$ and $\triangleright$ are *skew adjoint*, which means that for every $A, B \in L^X$, $R \in L^{X\times X}$, the following holds:

$$R \circ A \leq B \leftrightarrow A \leq R^{op} \triangleright B,$$

where $R^{op}(x,y) = R(y,x)$.

Let us remind that the $\circ$ composition was introduced by L. Zadeh [18] in the form $\max - \min$.

3 Fuzzy Preorders and Their Eigen Sets

In this Section, we recall basic facts about fuzzy preorder relations. Then we characterize eigen sets of fuzzy preorder relations.

Our interest to fuzzy preorder relations is connected with the analysis of the expression in (3) – a representation of the AFAM model. This is the representation of the so called Valverde (fuzzy) preorder [16].

3.1 Fuzzy preorders

A binary fuzzy relation on X is a $*$-*fuzzy preorder* of X, if it is reflexive and $*$-transitive. The fuzzy preorder $Q^* \in L^{X\times X}$, where

$$Q^*(x,y) = \bigwedge_{i\in I}(A_i(x) \to A_i(y)), \tag{4}$$

is *generated* by an arbitrary family of fuzzy sets $(A_i)_{i\in I}$ of X.

Remark 1. The fuzzy preorder Q^* is often called Valverde order determined by the family of fuzzy sets $(A_i)_{i\in I}$ of X (see [16] for details).

If Q is a fuzzy preorder on X, then Q^{op} is a fuzzy preorder on X as well.

3.2 Eigen sets of fuzzy preorders

In this Section, we show that fuzzy preorder Q^* (given by (4)) generated by fuzzy sets $(A_i)_{i\in I} \subseteq L^X$, is the greatest solution to the system of fuzzy relation equations

$$W \circ A_i = A_i,\ i \in I, \tag{5}$$

where W denotes an unknown fuzzy relation. At the same time, $(Q^*)^{op}$ is the greatest solution to the system of fuzzy relation equations

$$W \triangleright A_i = A_i,\ i \in I. \tag{6}$$

Moreover, we show that there exists a binary preorder that gives a solution to (5) and (after the "transposition") to (6).

Proposition 1. *Let $(A_i)_{i\in I} \subseteq L^X$, be a family of fuzzy sets of X and a fuzzy preorder Q^* be generated by this family in the sense of (4). Then Q^* is the greatest solution to the system of fuzzy relation equations (5) and $(Q^*)^{op}$ is the greatest solution to the system of fuzzy relation equations (6).*

Proof. It is obvious that both systems are solvable - this is because the identity (fuzzy) relation is a solution to (5) and (6). This fact implies that fuzzy relation Q^* is the greatest solution to the system (5). Let us prove the second claim.

The following chain of equivalences can be easily obtained from the first claim:

$$(\forall y)\left(\bigvee_{x\in X}(Q^*(x,y)*A_i(x))\le A_i(y)\right)\Leftrightarrow(\forall y)(\forall x)(Q^*(x,y)*A_i(x)\le A_i(y))\Leftrightarrow$$

$$(\forall x)(\forall y)(A_i(x)\le Q^*(x,y)\to A_i(y))\Leftrightarrow(\forall x)\left(A_i(x)\le\bigwedge_{y\in X}(Q^*(x,y)\to A_i(y))\right).$$

By reflexivity of Q^*,

$$\bigwedge_{y\in X}(Q^*(x,y)\to A_i(y))\le A_i(x).$$

Therefore,

$$(Q^*)^{op}\rhd A_i=A_i.$$

Corollary 1. *Let the assumptions of Proposition 1 be fulfilled. Then fuzzy sets A_i, $i\in I$, are eigen fuzzy sets of the relation Q^* ($(Q^*)^{op}$) with respect to composition $\circ$ ($\rhd$).*

By Proposition 1, fuzzy relation Q^* ($(Q^*)^{op}$) is the greatest solution of the system (5) (similarly, fuzzy relation $(Q^*)^{op}$ is the greatest solution of the system (6)). Let us show that there are smaller fuzzy relations that solve the system (5) (resp. the system (6)). Moreover, these smaller relations are ordinary (binary) preorders on X.

Proposition 2. *Let $(A_i)_{i\in I}\subseteq L^X$ be a family of fuzzy sets of X and fuzzy preorder Q^* be generated by this family in the sense of (4). Let $\Delta:L\to L$ be the following unary operation on L:*

$$\Delta(a)=\begin{cases}1, & \text{if } a=1,\\ 0, & \text{otherwise.}\end{cases}$$

Then $\Delta(Q^*)$ is a solution to the system (5) and $(\Delta(Q^*))^{op}$ is a solution to the system (6). Moreover, $\Delta(Q^*)$ is an ordinary (binary) preorder on X.

Proof. At first, we prove that $\Delta(Q^*)$ is a solution to (5). This fact easily follows from the following two inequalities:

$$\Delta(Q^*)\circ A_i\le Q^*\circ A_i=A_i,\ i\in I,$$
$$\Delta(Q^*)\circ A_i\ge A_i.$$

The first inequality is due to $\Delta(Q^*)\le Q^*$. The second inequality follows from reflexivity of $\Delta(Q^*)$.

At second, we show that $\Delta(Q^*)$ is a preorder relation on X. The property of reflexivity is inherited from Q^*.To prove transitivity we choose $t, u, v \in X$ and consider the case $\Delta(Q^*)(t,u) = 1$ and $\Delta(Q^*)(u,v) = 1$. It is easy to see that

$$\Delta(Q^*)(t,u) * \Delta(Q^*)(u,v) = Q^*(t,u) * Q^*(u,v) \leq Q^*(t,v).$$

We refer to $\Delta(Q^*)$ as to a binary "skeleton" of Q^*.

4 New AFAM Models

In this Section, we put a bridge between the theory, presented in Section 3, and the theory of autoregressive fuzzy associative memories (AFAM), presented in Section 2. We propose a new concept of adjoint AFAM models that share a common fuzzy preorder relation. We characterize types of noise that can be reduced by retrieval in corresponding AFAM models.

4.1 Adjoined AFAM Models

Let us choose and fix complete residuated lattice with the support $L = [0,1]$ and database $D = \{\mathbf{x}^1, \ldots, \mathbf{x}^p\}$ of 2D $[0,1]$-valued (gray scaled) images. We assume that the images in D are represented by n-dimensional vectors, so that each vector is a sequence of image rows. We identify every image with a fuzzy set on $\bar{n} = \{1, 2, \ldots, n\}$, so that $\mathbf{x}^k \in [0,1]^n$, $k = 1, \ldots, p$. We additionally assume that all fuzzy sets in D are normal.

Definition 1. *We say that a pair $(W, \boxdot)$, where $W \in [0,1]^{n \times n}$ is a fuzzy relation and $\boxdot : [0,1]^{n\times n} \times [0,1]^n \to [0,1]^n$ is a set-relation composition, is an AFAM model of database D, if for all $\mathbf{x}^k \in D$,*

$$\mathbf{x}^k = W \boxdot \mathbf{x}^k,\ k = 1, \ldots, p. \tag{7}$$

We say that two AFAM models $(W, \boxdot)$ and $(W, \widetilde{\boxdot})$ of D are adjoint, if there exists a complete residuated lattice $\mathcal{L}$ such that $\boxdot$ is of the $\sup - *$ type, $\widetilde{\boxdot}$ is of the $\inf - \to$ type and $\boxdot$ and $\widetilde{\boxdot}$ are skew-adjoint.

By (7) and the terminology of AFAM, any element from D is successfully retrieved from its sample in a corresponding AFAM model. On the other hand, according to the terminology of set-relation compositions, the same equation (7) characterizes any element from D as an eigen set of the fuzzy relation with respect to a certain composition - both are constituents of the corresponding AFAM model.

Example 1. Let us choose a complete residuated lattice $\mathcal{L}$ on $[0,1]$ and give two examples of adjoint AFAM models. Following (4), we construct the fuzzy preorder relation Q^*, such that

$$Q^*(i,j) = \bigwedge_{k=1}^{p} (\mathbf{x}^k(i) \to \mathbf{x}^k(j)), \tag{8}$$

and take its binary skeleton $\Delta(Q^*)$. By Proposition 1, the two pairs $(Q^*, \circ)$ and $((Q^*)^{op}, \triangleright)$ are examples of adjoint AFAM models. By Proposition 2, the two pairs $(\Delta(Q^*), \circ)$ and $(\Delta(Q^*)^{op}, \triangleright)$ are examples of adjoint AFAM models too.

4.2 AFAM and Noise Reduction

Let us characterize types of noise that can be removed/reduced by the retrieval mechanisms of adjoint AFAM models.

Definition 2. *Let* $(W, \boxdot)$ *be a model of AFAM with respect to database* D*,* $\tilde{\mathbf{x}} \in [0,1]^n$*, and* $\tilde{\mathbf{x}} \notin D$*. We say that* $\tilde{\mathbf{x}}$ *is a noisy version of some element* $\mathbf{x} \in D$*, that can be removed by the retrieval in* $(W, \boxdot)$*, if*

$$\mathbf{x} = W \boxdot \tilde{\mathbf{x}}. \tag{9}$$

Proposition 3. *Let* D *be a database,* $\mathcal{L}$ *a complete residuated lattice on* $[0,1]$*, and* $(W, \circ)$*,* $(W, \triangleright)$ *adjoint AFAM models. Let moreover, fuzzy relation* W *be reflexive. Then*

- *(i) if* $(W, \circ)$ *removes a noisy version* $\tilde{\mathbf{x}}$ *of the element* $\mathbf{x} \in D$*, then* $\tilde{\mathbf{x}}$ *is an eroded version of* $\mathbf{x}$*, i.e.* $\tilde{\mathbf{x}} \le \mathbf{x}$*;*
- *(ii) if* $(W, \triangleright)$ *removes a noisy version* $\tilde{\mathbf{x}}$ *of the element* $\mathbf{x} \in D$*, then* $\tilde{\mathbf{x}}$ *is a dilated version of* $\mathbf{x}$*, i.e.* $\tilde{\mathbf{x}} \ge \mathbf{x}$*.*

Proof. We give the proof of the case *(i)*. Assume that $\mathbf{x} = (x_1, \ldots x_n)$, $\tilde{\mathbf{x}} = (\tilde{x}_1, \ldots \tilde{x}_n)$ and $W \circ \tilde{\mathbf{x}} = \mathbf{x}$. Then for any $j = 1, \ldots, n$,

$$x_j = \bigvee_{i=1}^{n} (W(i,j) * \tilde{x}_i) \ge W(j,j) * \tilde{x}_j) = \tilde{x}_j.$$

Therefore, $\tilde{\mathbf{x}} \le \mathbf{x}$.

5 AFAM and Formal Concept Analysis

In this section, we explain the relationship between the proposed theory of AFAM and the theory of formal concept analysis (FCA) [17]. For this purpose, we adapt the terminology of FCA to the proposed above analysis of fuzzy associative memories.

We choose and fix a finite set X and a complete residuated lattice $\mathbf{L}$. The following *formal context* $\mathcal{K} = (\mathcal{A}, \mathcal{R}, \mathcal{I}_{\boxdot})$ where $\mathcal{A} \subseteq L^X$ is a set of objects, $\mathcal{R} \subseteq L^{X \times X}$ is a set of attributes, and $\mathcal{I}_{\boxdot}$ is an incidence relation on $\mathcal{A} \times \mathcal{R}$, is proposed. We say that an object (dataset) A possesses an attribute R, if the latter is an AFAM model of A via the composition $\boxdot$. Equivalently and in accordance with Definition 1, $\mathcal{I}_{\boxdot}(A, R) = 1$ if and only if $A \in \mathcal{A}$ is an eigen fuzzy set of $R \in \mathcal{R}$ with respect to the composition $\boxdot$.

Let us choose and fix a formal context $\mathcal{K}$ with the given above specification. A couple $(\mathbf{A}, \mathbf{W})$ is a *formal concept* of $\mathcal{K}$, if $\mathbf{A} \subseteq \mathcal{A}$ is a dataset, $\mathbf{W} \subseteq \mathcal{R}$ is a set of fuzzy relations such that $A \in \mathbf{A}$ if and only if A is an eigen fuzzy set of any $W \in \mathbf{W}$ with respect to the composition $\boxdot$, and vise versa, $W \in \mathbf{W}$ if and only if $(W, \boxdot)$ is an AFAM model of $\mathbf{A}$.

Using both languages, we say that every element in dataset $\mathbf{A}$ has every attribute in $\mathbf{W}$, i.e. can be successfully retrieved from its sample using any AFAM model $(W, \boxdot)$ where $W \in \mathbf{W}$ and composition $\boxdot$ is clear from the context $\mathcal{K}$.

Below, we analyze, how the problems of data retrieval and noise reduction can be formalized in terms of context analysis. We formulate two assertions and characterize formal concepts where datasets as formal concept objects are connected with retrieval models as formal concept attributes. The proofs of both below given statements can be obtained from Propositions 1,2.

Proposition 4. *Let X be a set, $\mathcal{L}$ a complete residuated lattice, $\mathcal{K} = (L^X, L^{X\times X}, \mathcal{I}_\circ)$ a formal context and $\mathbf{A} \subseteq L^X$ a dataset of objects. Then the smallest concept with objects from $\mathbf{A}$ includes as attributes all fuzzy relations $R \in L^{X\times X}$ such that $\Delta(Q^*) \le R \le Q^*$ where Q^* and $\Delta(Q^*)$ are specified in Propositions 1,2.*

Proposition 5. *Let X be a set, $\mathcal{L}$ a complete residuated lattice, $\mathcal{K} = (L^X, L^{X\times X}, \mathcal{I}_\triangleright)$ be a formal context and $\mathbf{A} \subseteq L^X$ a dataset of elements. Then the smallest concept with elements from $\mathbf{A}$ includes as attributes all fuzzy relations $R \in L^{X\times X}$ such that $\Delta(Q^*)^{op} \le R \le (Q^*)^{op}$ where $(Q^*)^{op}$ and $\Delta(Q^*)^{op}$ are specified in Propositions 1,2.*

The difference between Propositions 4 and 5 is in the choice of formal context. The latter determines a type of composition that connects an element from a dataset with the corresponding model.

By Definition 2, a noisy and ideal elements from $\mathbf{A}$ are connected by equation (9). The following relationship is an easy consequence of (9): the larger is an amount of noise, the greater should be a fuzzy relation that models retrieval with noise reduction and vise versa. This fact is illustrated in Fig. 2 where we demonstrate two results of noise reduction: the one is based on the model $(Q^*, \circ)$ and the other one - on the model $(\Delta(Q^*), \circ)$. It easily observed that model $(Q^*, \circ)$ reduces a larger amount of noise than the other model $(\Delta(Q^*), \circ)$. This is because fuzzy relation Q^* is greater than fuzzy relation $\Delta(Q^*)$.

Let us construct a formal concept of $\mathcal{K}$, suitable for characterization of an AFAM model with the ability of noise reduction. For this purpose we differentiate AFAM models according to "degrees of fuzziness" of their fuzzy relations. The latter will be defined as

$$\delta(W) = \sum_{x,y \in X} W(x, y),$$

where $W \in L^{X\times X}$. It is easy to see that for two fuzzy relations Q^* (given by (4)) and its binary skeleton $\Delta(Q^*)$, the following inequality holds: $\delta(\Delta(Q^*)) \le \delta(Q^*)$. A formal concept $(\mathbf{A}, \mathbf{W}_D)$ of $\mathcal{K}$ with the ability of noise reduction is a couple $(\mathbf{A}, \mathbf{W}_D)$, where $A \in \mathbf{A}$, if and only if A is an eigen fuzzy set of any $W \in \mathbf{W}_D$ such that $\delta(W) \ge D$, and vise versa, $W \in \mathbf{W}_D$, if and only if $\delta(W) \ge D$ and $(W, \boxdot)$ is an AFAM model of $\mathbf{A}$. The value of D regulates the amount of noise and should be less or equal than $\delta(Q^*)$.

6 Illustration

The aim of this Section is to give illustrations to the theoretical results of this paper. We use gray scaled images with the range $[0, 1]$, where 0 (1) represents the black (white) color. We choose two different datasets of images, both were artificially created from open access databases. These sets contain 2D images of 40×30 and 120×90 pixels, respectively. All images are represented by corresponding fuzzy sets with values in $[0, 1]$.

6.1 Algorithms of data retrieval

We tested the two AFAM models $(Q^*, \circ)$ and $(\Delta(Q^*), \circ)$, given in Example 1. The first experiment is to verify data retrieval. Both models successfully passed the verification. For the model that is based on the relation $\Delta(Q^*)$, we elaborated a fast algorithm of data retrieval that uses the fact that $\Delta(Q^*)$ is actually a binary relation.

We compare average execution time of the two corresponding algorithms and conclude that the algorithm based on $\Delta(Q^*)$ is up to thirty time faster than that based on Q^*, see Table 1.

Image size	Run-time 1	Run-time 2
40 x 30	5.40	0.195
120 x 90	435.80	13.79

Table 1. Run-time (in seconds) of the two algorithms of data retrieval that are based on models $(Q^*, \circ)$ (left column) and $(\Delta(Q^*), \circ)$ (right column).

6.2 Reduction of Noise

The second experiment is to analyze the influence of different retrieval AFAM models $(Q^*, \circ)$ and $(\Delta(Q^*), \circ)$ on eroded noise. For this purpose, we added 70% pepper noise to the original image in Fig. 1. The algorithm of data retrieval, based on the model $(Q^*, \circ)$, was more efficient than that, based on the model $(\Delta(Q^*), \circ)$, see Fig. 2.

7 Conclusion

A new theory of autoregressive fuzzy associative memories (AFAM) has been proposed. It extends [14] by using general algebraic structures and new types of autoregressive fuzzy associative memory models. We showed how the proposed theory is connected with systems of fuzzy relation equations and eigen sets of their solutions. We proposed a new concept of adjoint AFAM models that share

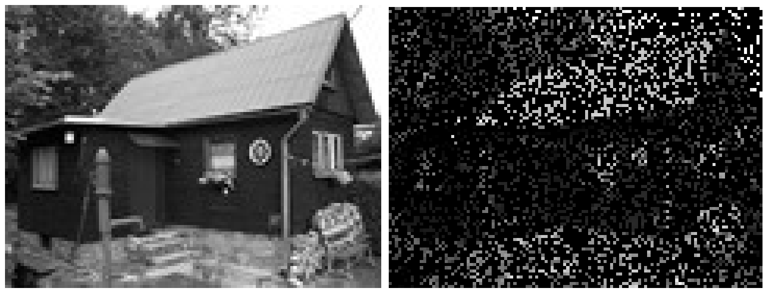


Fig. 1. Original image without noise (left) and with 70 % pepper noise (right).

Fig. 2. Noise reduction based on the model $(Q^*, \circ)$ (left) and on the model $(\Delta(Q^*), \circ)$ (right).

a common fuzzy preorder relation. We characterized two types of noise that can be reduced by retrieval in corresponding AFAM models. Two problems have been discussed: data retrieval and noise reduction.

The relationship between the AFAM data retrieval and the formal concept analysis has been analyzed. We proposed a new working theory of FAM and formalized it in the language of formal concept analysis. We characterized two principal activities of FAM: data retrieval and noise reduction. We used the proposed formalism and showed that the problem of data retrieval is connected with solvability and eigen sets of a certain system of fuzzy relation equations. We differentiated FAM models according to their ability to reduce noise and showed how the choice of formal context determines a type of noise that can be reduced by the corresponding retrieval mechanism.

Finally, we proposed a fast algorithm of data retrieval that is based on an AFAM model with a binary fuzzy preorder.

Acknowledgment This paper was supported by the NPUII project IT4I XS with the number LQ1602.

References

1. Alcalde, C., Burusco, A., Díaz-Moreno, J. C., Fuentes-González, R., Medina, J.: Fuzzy property-oriented concept lattices in morphological image and signal processing. Lect. Notes in Computer Science, 7903, 246–253 (2013)
2. Díaz, J.C., Madrid, N., Medina, J., Ojeda-Aciego, M.: New Links between Mathematical Morphology and Fuzzy Property-Oriented Con- cept Lattices. The 2014 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE 2014), pp. 599?603, Beijing (China) (2014)
3. Cheng, Q., Fan, Z.-T.: The stability problem for fuzzy bidirectional associative memories. Fuzzy Sets Syst. 132, 83–90, (2002)
4. Chung, F.-L., Lee, T.: On fuzzy associative memory with multiple-rule storage capacity. IEEE Trans. Fuzzy Syst. 4, 375–384 (1996)
5. Junbo, F., Fan, J., Yan, S.: A learning rule for fuzzy associative memories. Proc. IEEE Int. Conf. Neural Networks, IEEE World Congr. Computational Intelligence 7, 4273-4277 (1994)
6. Kosko, B.: Neural Networks and Fuzzy Systems: A Dynamical Systems Approach to Machine Intelligence. Prentice-Hall (1992)
7. Perfilieva, I.: Fuzzy relation equations in semilinear spaces. Proc. Conf. IPMU'2010 Dortmund, 545 – 552 (2008)
8. Perfilieva, I.: Systems of fuzzy relation equations in a space with fuzzy preorder. Proc. of IFSA-EUSFLAT 2009 Joint Conf., Lisbon, 1601–1605 (2009)
9. Perfilieva, I.: Finitary solvability conditions for systems of fuzzy relation equations. Inf. Sci. 234, 29–43 (2013)
10. Perfilieva, I., Vajgl, M.: Autoassociative Fuzzy Implicative Memory on the Platform of Fuzzy Preorder. Proc. of IFSA-EUSFLAT 2015, Atlantis Press, 2015.
11. Ruspini, E. H.: A new approach to clustering. Inf. and Cont. 15, 22–32 (1969)
12. Sanchez, E.: Resolution of Eigen fuzzy sets equations. Fuzzy Sets and Syst. 1, 69-74 (1978)

13. Steinbuch, K.: Die lernmatrix. Kybernetik 1, 36-45 (1961)
14. Sussner, P. , Valle, M.-E.: Implicative Fuzzy Associative Memories. IEEE Trans. Fuzzy Systems 14, 793–807 (2006)
15. Taylor, W.: Eletrical simulation on some nervous system functional activities. Inf. Theory 15, 166-177 (2004)
16. Valverde, L.: On the structure of F-indistinguishability operators. Fuzzy Sets and Syst. 17, 313-328 (1985)
17. Wille, R.: Formal Concept Analysis as Mathematical Theory of Concepts and Concept Hierarchies. In: B. Ganter et al. Formal Concept Analysis. Foundations and Applications, Springer (2005)
18. Zadeh, L. A.: The concept of a linguistic variable and its application to approximate reasoning I, II, III. Inf. Sci. 8-9, 199–257, 301–357, 43–80 (1975)

Textual Information Extraction in Document Images Guided by a Concept Lattice

Cynthia Pitou[1 2] and Jean Diatta[1]

[1] EA2525-LIM, Saint-Denis de La Réunion, F-97490, France
[2] Groupe Austral Assistance, 16 rue Albert Lougnon, 97490 Sainte Clotilde, France
cpitou@gaa.fr jean.diatta@univ-reunion.fr

Abstract. Text Information Extraction in images is concerned with extracting the relevant text data from a collection of document images. It consists in localizing (determining the location) and recognizing (transforming into plain text) text contained in document images. In this work we present a textual information extraction model consisting in a set of prototype regions along with pathways for browsing through these prototype regions. The proposed model is constructed in four steps: (1) produce synthetic invoice data containing the textual information of interest, along with their spatial positions; (2) partition the produced data; (3) derive the prototype regions from the obtained partition clusters; (4) build the concept lattice of a formal context derived from the prototype regions. Experimental results, on a corpus of 1000 real-world scanned invoices show that the proposed model improves significantly the extraction rate of an Optical Character Recognition (OCR) engine.

Keywords: textual information extraction, concept lattice, clustering

1 Introduction

Document processing is the transformation of a human understandable data in a computer system understandable format. Document analysis and understanding are the two phases of document processing. Considering a document containing lines, words and graphical objects such as logos, the analysis of such a document consists in extracting and isolating the words, lines and objects and then grouping them into blocks. The subsystem of document understanding builds relationships (to the right, left, above, below) between the blocks. A document processing system must be able to: locate textual information, identify if that information is relevant comparatively to other information contained in the document, extract that information in a computer system understandable format. For the realization of such a system, major difficulties arise from the variability of the documents characteristics, such as: the type (invoice, form, quotation, report, etc.), the layout (font, style, disposition), the language, the typography and the quality of scanning. In the literature, works in pattern recognition [16] and character recognition [28] provide solutions for textual information extraction in a computer system understandable format. Works in automatic natural language

© Marianne Huchard, Sergei O. Kuznetsov (Eds.): CLA 2016, pp. 325–336,
ISBN 978-5-600-01454-1, National Research University Higher School of Economics, 2016.

processing [6] contribute to solving the problem about the detection of relevant information. This paper is concerned with scanned documents, also known as document images. We are particularly interested in locating textual information in invoice images. Invoices are largely used and well regulated documents, but not unified. They contain mandatory information (invoice number, unique identifier of the issuing company, VAT amount, net amount, etc.) which, depending on the issuer, can take various locations in the document. For instance, it seems difficult to identify a trend as to the position of the date and invoice number. However, similarities may occur locally for one or many information. To take an example, the amount is usually positioned at bottom-right in the French and English systems. Recent approaches such as those presented in [3, 4, 9] are specifically concerned with the extraction of information in administrative documents such as invoices. These works have in common the search, within a base, for a document similar to an input document. Each document of this base is assigned a template that lists some attributes (position, type, keywords) to be used in order to locate information contained in similar input documents. Bartoli et al. [3] propose a system of selecting, for an input document, the nearest wrapper based on a distance measure. A wrapper is an object containing information about geometric properties and textual content of elements to extract. Belaid et al. [4] propose a case-based-reasoning approach for invoice processing. Cesarini et al. [9] propose a system to process documents that can be grouped into classes. The system comprises three phases: (1) document analysis, (2) document classification, (3) document understanding.
The present paper is in the framework of region-based textual information localization and extraction [29, 30]. We present a textual information extraction model consisting in a set of prototype regions along with pathways for browsing through these prototype regions. The proposed model is constructed in four steps:

1. produce synthetic invoice data from real-world invoice images containing the textual information of interest, along with their spatial positions;
2. partition the produced data;
3. derive the prototype regions from the obtained partition clusters;
4. derive pathways for browsing through the prototype regions, from the concept lattice of a suitably defined formal context;

The paper is organized as follows. Section 2 is devoted to the construction of prototype regions. The formal context defined using the obtained prototype regions, and the determination of paths from the concept lattice of that formal context are described in Section 3. Section 4 presents our approach for textual information extraction, using the defined paths. Finally, some experimental results are presented in Section 5 and the paper is closed with a conclusion and perspectives.

2 Construction of prototype regions

2.1 Construction of a synthetic data set

The present work is motivated by the request of a company interested in developing its own system enabling to automatically extract some textual information from scanned invoices. The company has provided us with a corpus of 1000 real-world scanned invoices, emitted by 18 service providers whose business is around car towing and auto repair. All the images are one page A4 documents. The whole set of information the company is interested in, comprises: invoice number, invoice date, net amount, VAT amount, customer reference, the type of service provided, the issuer identity. In our study, we consider only the following five information:

- I1: the key word of the service provided: towing or auto repair,
- I2: customer reference: a string of 9-13 characters,
- I3: the plate number of the assisted vehicle,
- I4: the invoice issuer unique identifier: a string of 14 digits,
- I5: the net amount of money requested for the provided service.

Each of the textual information is located in a region delimited by a rectangle defined by the coordinates (x, y) (in pixels) of its top left corner and the coordinates (z, q) of its bottom right corner. In the sequel, by the term region will be meant a rectangular area in an invoice image. Hence, a region may be represented by the four coordinates (x, y, z, q) of its top left and bottom right corners. As the information to be extracted are located in (rectangular) regions we adopt a region-based extraction approach. The regions which the proposed approach is based on are prototypes obtained from the more specific regions containing, each, a single information. Now, the coordinates of the regions containing the needed information are not available for the real-world scanned invoices at hand. To cope with this, we develop a JAVA program, with a graphical interface, enabling to create synthetic invoice data simulating the real-world scanned invoices along with the approximate coordinates of the specific rectangles containing the needed information. For instance, from a real-world scanned invoice an initial synthetic invoice is manually created. This synthetic invoice is a single black and white A4 page. This page will contain a string corresponding to a plate number approximately at the same location as the plate number information appears in the real-world invoice. Additionally, the string will be inserted approximately with the same size and the same font as in the real-world invoice in order to look like it. The string is inserted manually in the initial synthetic invoice as one can do with a text editor. However, many strings contained in the real-world invoice are not reproduced in the synthetic invoice. For instance, the information about the emitter and the receiver (address, phone number, ...) are not reproduced because they are not relevant for the study. Thus, the set of textual information I1 to I5 is placed manually on the synthetic invoice. Then, from the obtained initial synthetic invoice a fixed number of synthetic invoice images may be created automatically. In such synthetic invoice images, the information locations are maintained identically to the initial synthetic invoice but the contained string may vary. Finally, for each distinct emitter of the real-world invoice images corpus, one initial synthetic invoice image is created manually and a fixed number of synthetic invoice images is created automatically from the initial synthetic invoice. A corpus of 1000 synthetic invoice images is thus produced and, for each synthetic invoice, both the textual information and the coordinates of the respective rectangles containing them, are stored in a database. The original distribution per emitter of the real-world invoices is preserved in the synthetic corpus of images. Synthetic data sets can then be generated from this database for closer insight. An example of such data sets is a set of (synthetic invoice) records described by 20 variables representing 5 blocks of 4 coordinates (x, y, z, q), each block being associated with one of the 5 considered information I1 to I5. It should be noted that this possibility to produce a synthetic representation of a real-world scanned invoice is an important step for updating the proposed model, namely when one has to extract information from previously unseen scanned invoice. This point will be discussed later in Section 6.

2.2 Clustering of the synthetic data

As we mentioned in Section 2.1, the regions which our proposed approach is based on are the so-called prototype regions, obtained from the specific regions that contain, each, a single information. More precisely, a prototype region associated with a given information should be a region containing a homogeneous set of specific regions related to various positions of this information in different invoice images. This makes cluster analysis methods, (such as the partitioning ones) good candidates for capturing such homogeneous sets of specific regions. Then, in the next section, the construction of prototype regions from such homogeneous sets of specific regions is explained.
The synthetic data obtained from the previous phase can be partitioned either: (a) in an overall view taking into account all of the 20 variables, or (b) in 5 independent views, each corresponding to one of the 5 information I1 to I5 and taking into account, for each view, the 4 associated variables. The approach in five independent views consists in creating five data sets: D1, D2, D3, D4 and D5. A record in Di is described by the four coordinates of regions containing information Ii. For both approaches we adopted the K-means [23] clustering with Euclidean distance. K-means is a popular, simple and efficient algorithm for cluster analysis. To determine the number of clusters, we conducted, on the one hand, an (agglomerative) ascending hierarchical clustering with Ward criterion (clustering method based on a classical sum-of-squares criterion, producing groups that minimize within-group dispersion) [24] and, on the other hand, executions of K-means for values of k between 2 and 18. Several validity criteria, such as within cluster sum of squares, silhouette and Calinkski-Harabasz [33], provided by the package clusterCrit of R software, were used to determine the optimal number of clusters for each data set. It turns out that considering five independent views leads to better clusters w.r.t. each of the considered validity

criteria. Therefore, we adopt the option consisting in partitioning each of the 5 views. The best values of k obtained for the respective five views are shown in Table 1.

Table 1. Number of clusters for data sets D1, D2, D3, D4 and D5.

	D1: Info I1	D2: Info I2	D3: Info I3	D4: Info I4	D5: Info I5
k	10	10	10	3	10

2.3 Determination of the prototype regions

As we mentioned in the previous section, a prototype region associated with an information should contain a homogeneous set of specific regions related to various positions of this information in different invoice images.
Recall that for each information Ii, the associated data set Di is partitioned into some number of clusters (see Table 1). Then, we associate to each of these clusters, say C, a *prototype region* defined as the smallest rectangle R containing each of the specific rectangles in C. Thus, we obtain 43 prototype regions R_1,...,R_{43}, with the first 10 related to information I1, the next 10 to I2, the next 10 to I3, the next 3 to I4 and the last 10 to I5. Figure 1 shows the prototype regions related to information I4. The next step of the construction of our proposed model is to set up pathways for efficiently browsing through the set of defined prototype regions. Such pathways will be obtained from the concept lattice of a suitably defined formal context.

3 Determination of pathways for browsing through the prototype regions

So far, we indicated how we determine prototype regions containing the textual information to be extracted. So we come to the fourth step in our approach, namely, define pathways for efficiently browsing through the set of these prototype regions. For this, Formal Concept Analysis (FCA) appears very appropriate. Indeed, the pathways we seek to determine may be obtained from the concept lattice of a suitably defined formal context.

3.1 Construction of the concept lattice

Recall that a *formal context* is a triple $\mathbb{K} = (O, A, \mathcal{R})$, where O is a set of objects, A a set of attributes and $\mathcal{R} \subseteq O \times A$ a binary relation from O to A. A *formal concept* of $\mathbb{K}$ is a pair (X, Y) such that $Y = X' = \{a \in A : x\mathcal{R}a \text{ for all } x \in X\}$ and $X = Y' = \{x \in O : x\mathcal{R}a \text{ for all } a \in Y\}$. Note that the double application of

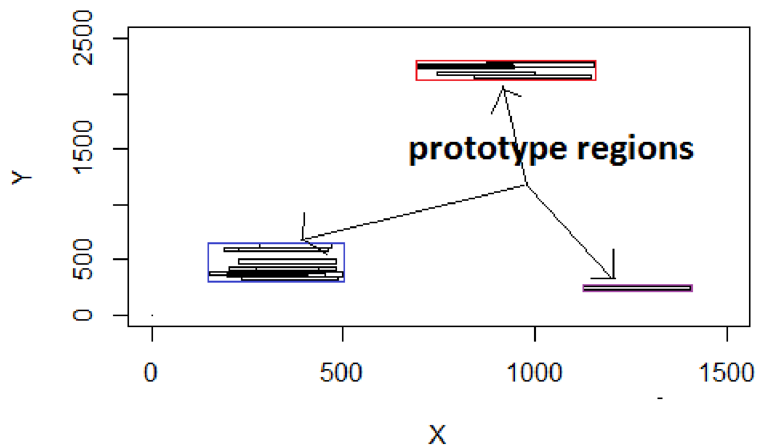


Fig. 1. Prototype regions related to information I4.

the derivation operator $(.)'$ is a closure operator, i.e. $(.)''$ is extensive, idempotent and monotone. Sets $X \subseteq O, Y \subseteq A$, such that $X = X''$ and $Y = Y''$ are said to be closed. The subset $X \subseteq O$ is called the extent of the concept (X, Y) and Y its intent. The concept lattice of the formal context $\mathbb{K}$ [35], also known as the Galois lattice of the binary relation $\mathcal{R}$ [2], is the (complete) lattice $(\mathcal{L}(\mathbb{K}), \leq)$, where $\mathcal{L}(\mathbb{K})$ is the set of formal concepts of $\mathbb{K}$ and $\leq$ the subconcept/superconcept partial order. Thus, a concept lattice contains a minimum (resp. a maximum) element according to the relation $\leq$, called the bottom (resp. the top). In this work, we consider the formal context where the objects are the invoice images and the attributes the predicates $I_i = j$, where Ii, $i = 1, \ldots, 5$ denotes the five textual information mentioned in Section 2.1, and $j = 1, \ldots, 43$ denotes the ID of the 43 prototype regions R_1,...,R_{43}. An invoice $\mathrm{o_n}$ is in relation with a predicate $I_i = j$ if the textual information I_i is located at prototype region R_j in the invoice $\mathrm{o_n}$. A summary of this formal context is shown in Table 2.

3.2 Determination of paths from the concept lattice

Recall that, given a formal context $\mathbb{K} = (O, A, \mathcal{R})$, an association rule is a pair (X, Y), denoted as $X \to Y$, where X and Y are disjoint subsets of A [1]. The set X is called the antecedent of the rule $X \to Y$ and Y its consequent. The support of an association rule $X \to Y$ is the proportion of objects that contain all the attributes in $X \cup Y$, i.e. $\frac{|(X \cup Y)'|}{|O|}$. The confidence of $X \to Y$ is the proportion of objects that contain Y, among those containing X. A (support,confidence)-valid

Table 2. Part of the Formal context of invoices data sets.

	I1=1	I1=2	I1=3	I1=4	I1=5	I1=6	I1=7	I1=8	I1=9	I1=10	...	I4=31	I4=32	I4=33	I5=34	I5=35	I5=36	I5=37	I5=38	I5=39	I5=40	I5=41	I5=42	I5=43
o_1	X										...	X			X									
...											...													
o_{895}										X	...	X				X								
...											...													
o_{1000}							X				...		X								X			

association rule is an association rule whose support and confidence are at least equal to a fixed minimum support threshold and a fixed minimum confidence threshold, respectively. An approximate association rule is an association rule whose confidence is less than 1. When the minimum support threshold is set to 0, the Luxenburger basis of approximate association rules is the set of rules of the form $X \rightarrow Y \setminus X$ where $X = X''$, $Y = Y''$, $X \subset Y$ and there is no Z such that $Z'' = Z$ and $X \subset Z \subset Y$ [21].

The Luxenburger basis can be visualized directly in the line diagram of a concept lattice. Each approximate rule in the Luxenburger basis corresponds exactly to one edge in the line diagram. The line diagram of a lattice contains paths by which one can move from the top concept to the bottom one. The pathways we adopt for browsing through the set of prototype regions are exactly those corresponding to sequences of association rules of the Luxenburger basis, i.e. top-down consecutive edges in the concept lattice. In other words, a *pathway* is a sequence $Y_0 \rightarrow Y_1 \rightarrow \ldots \rightarrow Y_n$, where Y_0 is the intent of the top formal concept and for all $0 \leq i < n$, $Y_i \rightarrow Y_{i+1}$ is an association association rule of the Luxenburger basis. Given a node of the concept lattice, there are as many approximate association rules of the Luxenburger basis whose antecedent is the intent of this node, as are the children nodes of this node in the concept lattice. Between two approximate association rules having the same antecedent, the one with highest support is considered first. For instance, let a pathway p_1: I5=42 $\rightarrow$ {I1=9, I3=25} holds with a support of 4% and a pathway p_2: I5=42 $\rightarrow$ {I2=19, I3=28} holds with a support of 6%. In the aim to extract information I1 to I5 from a candidate invoice image, and supposing that I5=42 is the lattice top node's direct child node which holds the highest support value, prototype region R_{42} is visited first in order to find information I5. Then, using pathway p_2, prototype regions R_{19} and R_{28} are visited for finding information I2 and I3 respectively. When, an information Ii is not found in a prototype region given by pathway p_2, so p_1 may be used to find it. Thus, all approximate association rules given by the Luxenburger basis are used for information I1 to I5 localization and extraction. In systems, such as, CREDO [8] and SearchSleuth [12] the browsing strategy consists in focusing on a concept and its neighbors.

The effectiveness and performance for using this type of strategy in Web search have been demonstrated in [8, 12].

4 Textual information extraction

To extract the textual information of interest, we perform an optical character recognition (OCR) engine on prototype regions, using the pathways determined in the previous step. Recall that a pathway is a sequence $Y_0 \rightarrow Y_1 \rightarrow ... \rightarrow Y_n$, where Y_0 is the intent of the top formal concept and for all $0 \leq i < n$, $Y_i \rightarrow Y_{i+1}$ is an approximate association rule of the Luxenburger basis. It should be noted that each node Y_α in such a sequence represents a set of predicates "Ii=j" indicating that information Ii belongs to prototype region R_j. First, the set of pathways is ordered by descending support value of the intents. Then, in each node given by a pathway, an OCR engine is performed on each prototype region in order to extract the corresponding information Ii. In formal language theory, a regular expression is a sequence of characters that defines a search pattern, mainly for use in pattern matching with strings. For each sought information Ii, a regular expression is built and then used to check whether the extracted string (by the OCR engine) matches with the given information.
In the literature, approaches such as in [34, 18, 19] are based on *concept lattice classifier* and use a concept lattice for a classification task. Such approaches aim to improve the task of character or symbol recognition in images. In [34], the authors developed a recognition system named Navigala and fitted to recognize noisy graphical objects and especially symbols images in technical documents such as architectural plans or electrical diagrams. The authors noted that Navigala is somewhat generic and can be successfully applied to other types of data. In [18], the authors proposed modifications of some classifiers (naive Bayes, nearest neighbor and random forest classifiers) in order to use the modified classifiers as a part of the ABBYY OCR Technologies recognition schema for its performance improving. The authors note that their approach based on random forest can be applied to combine results of concept lattice classifiers.
In this paper, the task of text extraction is done with a free OCR engine named Tesseract OCR *(https://github.com/tesseract-ocr)*. Tesseract OCR was chosen because it is a free software providing a JAVA API. In this paper, we focus on the textual information localization task in administrative document images. Indeed, OCR engine such as ABBYY OCR has a better recognition rate than free OCR such as Tesseract OCR, but both are not able to localize or pick out a given information such as the net amount in invoice images. Their task is just to transform, as efficiently as possible, the text contained in images into plain text. In this work, we propose to combine the proposed localization approach (based on clustering analysis and navigation in a concept lattice) with any OCR engine in order to extract a given information in document images without browsing and recognizing the entire images.

5 Experimental results

We achieved an experiment in order to test the proposed model for textual information extraction in real-world invoice images. The experiment consists in extracting information I1 to I5, in the set of 1000 real-word invoice images (Section2.1). Despite the fact that the approach was trained and tested with good results on the synthetic data, in this section we present test results of the approach on real-world invoice images. Indeed, the corpus of real-world invoice images contains some noise which is not present in the synthetic invoice images. The real-world invoice images may contain colored images such as a logo, shadow areas and handwritten text. Additionally, they may be scanned with poor quality and may present distortion. Thus, the corpus of real-world invoice images seems to us to be quite interesting for testing the proposed textual information localization and extraction model. We performed two types of extraction:

1. from full images: OCR is performed on the entire page images regardless to specific regions;
2. from prototype regions, using the pathways presented in Section 3: OCR is performed only on image sub-regions, using the pathways.

To perform OCR on images, the JAVA library of the free OCR engine named Tesseract in its 3.02 version is used. We considered two measures:

1. the rate of correct information among the total number of sought information (recall),
2. the rate of correct information among the total number of detected information (precision).

On the one hand, a sought information is considered detected, if a string which matches the corresponding regular expression is found. On the other hand, a sought information is considered correctly extracted, if the extracted textual information corresponds exactly to the visual information that should be read in the image. For instance, let 'net amount' be a sought information and assume that the net amount is 107€ in the invoice image. During the process, if the retrieved information is "101€", the net amount will not be considered correctly extracted because the real net amount mentioned in the original invoice image is "107€". The results are presented in Table 3. On the one hand, despite the fact that according to [25], the Tesseract OCR engine has accuracy of 70% for text extraction in gray scale number plate images, we observe that the accuracy of the OCR engine is weak for text extraction in real-world invoice images. On the other hand, these results show that our proposed model improves significantly the performance of the OCR engine. Note that a p-value of 8.799e-05 was obtained for this experiment, which means that the results are significant.

6 Conclusion and perspectives

We presented a (prototype) region-based model for localizing and extracting textual information in document images. Experimental results show that the

Table 3. Results of experiments.

	recall	precision
OCR	29,84 %	37,36 %
OCR + pathways	35,88 %	50,46 %

proposed model improves significantly the correctness of a textual information extraction process based on an OCR engine. This model is constructed in four steps:

1. produce synthetic invoice data from real-world invoice images containing the textual information of interest, along with their spatial positions;
2. partition the produced data;
3. derive the prototype regions from the obtained partition clusters;
4. derive pathways for browsing through the prototype regions, from the concept lattice of a suitably defined formal context;

The Step 1 is important when one has to extract information from a previously unseen invoice image. Indeed, if some information of such an invoice image are not retrieved, then a synthetic representation of the considered invoice can be produced, and this triggers incremental updates of the synthetic data sets, the prototype regions, and the concept lattice. Our future work will focus on these update processes, and compare them with those proposed in [3, 4, 9]. We also plan to develop a classification model, as in [9], that will enable to predict the invoice emitter based on the five textual information I1 to I5 considered in the present paper. This will allow to easier retrieve the other textual information the company is interested in: invoice number, invoice date, tax rate, tax due.

Acknowledgment

This work was partially carried out within the project ClustOverlap supported by Reunion Island Region - grant DIRED 20140704. The authors are also very grateful to the 3 anonymous reviewers for their valuable comments.

References

1. Agrawal, R., Imielinski, T., Swami, A.: Mining association rules between sets of items in large databases. ACM SIGMOD Record, 22.2, pp. 207–216 (1993).
2. Barbut, M., Monjardet, B.: Ordre et classification: algèbre et combinatoire. Hachette (1970).
3. Bartoli, A., Davanzo, G., Medvet, E., Sorio, E.: Semisupervised wrapper choice and generation for print-oriented documents. IEEE Transactions on Knowledge and Data Engineering 26.1, pp. 208–220 (2014).
4. Belaïd, A., D'Andecy, V. P., Hamza, H., Belaïd, Y.: Administrative document analysis and structure. Learning Structure and Schemas from Documents, pp. 51–71 (2011).

5. Birkhoff, G.: Lattice Theory. American Mathematical Society 25.3 (1967).
6. Cambria, E., White, B.: Jumping NLP Curves: A Review of Natural Language Processing Research. IEEE Computational Intelligence Magazine 9.2, pp. 48–57 (2014).
7. Carpineto, C., Michini, C., Nicolussi, R.: A concept lattice-based kernel for SVM text classification. Formal Concept Analysis, pp. 237–250 (2009).
8. Carpineto, C., Romano G.: Expoiting the Potential of Concept Lattices for Information Retrieval with CREDO. Journal of Universal Computer Science, 10.8, pp. 985–1013 (2004).
9. Cesarini, F., Francesconi, E., Gori, M., Soda, G.: Analysis and understanding of multi-class invoices. International Journal on Document Analysis and Recognition 6.2, pp. 102–114 (2003).
10. Charikar, M., Chekuri, C., Feder, T., Rajeev Motwani: Incremental clustering and dynamic information retrieval. In: Proceedings of the twenty-ninth annual ACM symposium on Theory of computing. ACM, pp. 626–635 (1997).
11. Cho, W.C., Richards, D.: Improvement of Precision and Recall for Information Re- trieval in a Narrow Domain: Reuse of Concepts by Formal Concept Analysis. In Proceedings of the 2004 IEEE/WIC/ACM International Conference on Web Intelligence (WI '04), pp. 370–376 (2004).
12. Ducrou, J., Eklund, P. W.: SearchSleuth: The conceptual neighbourhood of an web query. In J. Diatta, P. Eklund, & M. Liquire (Eds.), Proc. CLA 2007, LIRMM & University of Montpellier II (2007).
13. Eisenbarth, T., Koschke, R., Simon, D.: Locating Features in Source Code. IEEE Transactions on software engineering 29.3, pp. 210–224 (2003).
14. Ganter, B., Wille, R.: Contextual attribute logic. Springer Berlin Heidelberg (1999).
15. Ganter, B., Wille, R.: Formal Concept Analysis: Mathematical Foundations. Springer Science & Business Media (2012).
16. Guyon, I., Cawley, G., Dror, G., Saffari, A.: Hands-on Pattern Recognition. Challenges in Machine Learning 1. Isabelle Guyon, Gavin Cawley, Gideon Dror, and Amir Saffari editors (2011).
17. Hyontai, S.: An Effective Sampling Method for Decision Trees Considering Comprehensibility and Accuracy. WSEAS Transactions on Computers 8.4, pp. 631–640 (2009).
18. Itskovich, L., Kuznetsov, S.O.: Machine Learning Methods in Character Recognition. Proc. 13th International Conference on Rough Sets, Fuzzy Sets, Data Mining, and Granular Computing (RSFDGrC 2011), Lecture Notes in Computer Science 6743, pp. 322–329 (2011).
19. Kaytoue, M., Kuznetsov, S.O., Napoli, A., Duplessis, S.: Mining gene expression data with pattern structures in Formal Concept Analysis. Inf. Sci. 181(10): 1989-2001 (2011).
20. Langley, R.: Practical statistics simply explained. Courier Corporation (1971).
21. Kuznetsov, S.O., Makhalova, T.P.: Concept interestingness measures: a comparative study. CLA 2015, pp. 59–72 (2015).
22. Luxenburger, M.: Implications partielles dans un contexte. Mathématiques et Sciences Humaines 113, pp. 35–55 (1991).
23. MacQueen, J. et al.: Some methods for classification and analysis of multivariate observations. Proceedings of the fifth Berkeley symposium on mathematical statistics and probability 1.14, pp. 281–297 (1967).

24. Murtagh, F., Legendre, P.: Wardś hierarchical agglomerative clustering method: Which algorithms implement wardś criterion?. Journal of Classification 31.3, pp. 274–295 (2014).
25. Patel, C., Patel, A., Patel, D.: Optical character recognition by open source OCR tool tesseract: A case study. International Journal of Computer Applications, 55.10, pp. 50–56 (2012).
26. Poelmans, J., Ignatov, D.I., Kuznetsov, S.O., Dedene, G.: Formal Concept Analysis in knowledge processing: A survey on applications. Expert systems with applications 40.16, pp. 6538–6560 (2013).
27. R Core Team: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria. ISBN 3-900051-07-0, (2013).
28. Singh, S.: Optical Character Recognition Techniques: A survey. International Journal of Advanced Research in Computer Engineering & Technology (IJARCET) 4.6, pp. 545–550 (2013).
29. Sumathi, C.P., Santhanam, T., Devi, G.G.: A survey on various approaches of text extraction in images 3.4, pp. 27–42 (2012).
30. Sumathi, C.P., Santhanam, T., Priya, N.: Techniques and challenges of automatic text extraction in complex images: A survey. Journal of Theoretical and Applied Information Technology, 35.2, pp. 225-235 (2012).
31. Taouil, R., Pasquier, N., Bastide, Y., Lakhal, L.: Mining bases for association rules using closed sets. ICDE2000 International Conference, pp. 307 (2000).
32. van De Vel, M.L.J: Theory of convex structures. Elsevier (1993).
33. Vendramin, L., Campello, R., Hruschka, E.: Relative Clustering Validity Criteria: A Comparative Overview. Statistical Analysis and Data Mining 3, pp. 209–235 (2010).
34. Visani, M., Bertet, K., Ogier, J.-M.: NAVIGALA: An original symbol classifier based on navigation through a galois lattice. International Journal of Pattern Recognition and Artificial Intelligence 25.4, pp. 449–473 (2011).
35. Wille, R.: Restructuring lattice theory: an approach based on hierarchies of concepts. Springer Netherlands (1982).

CAISL: Simplification Logic for Conditional Attribute Implications

Estrella Rodríguez-Lorenzo[1], Pablo Cordero[1], Manuel Enciso[1], Rokia Missaoui[2], Ángel Mora[1]

[1]Universidad de Málaga, Andalucía Tech, Spain,
e-mail: {estrellarodlor,amora}@ctima.uma.es, {pcordero,enciso}@uma.es
[2] Université du Québec en Outaouais, Canada,
e-mail: {rokia.missaoui}@uqo.ca

Abstract. In this work, we present a sound and complete axiomatic system for conditional attribute implications (*CAI*s) in Triadic Concept Analysis (TCA). Our approach is strongly based on the Simplification paradigm which offers a more suitable way for automated reasoning than the one based on Armstrong's Axioms. We also present an automated method to prove the derivability of a *CAI* from a set of *CAI*s.

1 Introduction

Implications in FCA represent associations between two attribute sets, denoted by $X \to Y$, and capture an important knowledge hidden in the input data. They also allow an alternate representation of the concept lattice and open the door to their automated management through logic. Such a management is used, for instance, to characterize representations of the whole knowledge by means of the notion of implicational systems. There exist different axiomatic systems in FCA, the first one is called *Armstrong's Axioms* [1], but later, other equivalent logics emerged [4, 5, 9].

The first study on triadic implications has been investigated by Biedermann [2] and then an extended work has been poposed by Ganter and Obiedkov [6]. In addition to a formal definition of implications and their language, we believe that the introduction of a sound and complete inference system is needed to reason about such implications and determine whether a given implication can be derived from an implication basis. Soundness ensures that implications derived by using the axiomatic system are valid in the formal context and completeness guarantees that all valid implications can be derived from the implicational system.

As far as we know, there does not exist an axiomatic system in Triadic Concept Analysis. The main goal of this paper is then to define a new axiomatic system based on Simplification Logic [4] as an alternate view of the inference system recently developed by the authors [12]. This new way also allows an efficient automated reasoning, commonly called the implication problem, to determine if a conditional attribute implication (*CAI*) can be derived from a set of *CAI*s.

Given a set of dependencies Σ and a further dependency σ, the implication problem means that one would like to check whether σ holds in all datasets

© Marianne Huchard, Sergei O. Kuznetsov (Eds.): CLA 2016, pp. 337–348,
ISBN 978-5-600-01454-1, National Research University Higher School of Economics, 2016.

satisfying Σ. This problem occurs in research areas such as database theory and knowledge reasoning, and its solution allows the search for associations in an interactive and exploratory way rather than an exhaustive manner. Using Armstrong's axioms, many polynomial time algorithms for implication problem decision have been defined and the closure of an attribute set has been exploited to solve it.

The remainder of this paper is organized as follows. In Section 2 we provide a background on TCA. Section 3 briefly presents a logic for conditional attribute implications called CAIL [12] while Section 4 describes a new axiomatic system called CAISL that is more suitable for solving the implication problem in the triadic framework. In Section 5 we establish equivalences derived from CAISL between sets of *CAI*sand show how we can syntactically transform and simplify a set of *CAI*s while preserving their semantics in the CAISL context. To check whether a *CAI* holds for a given set of *CAI*s, we propose and illustrate a new procedure in Section 6. Finally, Section 7 summarizes our contribution and presents further work.

2 Triadic Concept Analysis

As a natural extension to Formal Concept Analysis (FCA), theoretical foundations of Triadic Concept Analysis have been investigated by Lehmann and Wille [8] who were inspired by the philosophical framework of Charles S. Peirce [11] of three universal categories. The input is a formal triadic context describing objects in terms of attributes that hold under given conditions and the output is a concept trilattice that allows the generation of triadic association rules, including implications [2, 6, 7, 10].

Definition 1. *A triadic context* $\mathbb{K} = \langle G, M, B, I\rangle$ *consists of three sets: a set of objects (*G*), a set of attributes (*M*) and a set of conditions (*B*) together with a ternary relation* $I \subseteq G \times M \times B$. *A triple* (g, m, b) *in* I *means that object* g *possesses attribute* m *under condition* b.

Figure 1 shows a triadic context $\mathbb{K} := \langle G, M, B, I\rangle$, where $G = \{1, 2, 3, 4, 5\}$ is a set of customers, $M = \{\mathrm{P, N, R, K, S}\}$ a set of suppliers and $B = \{\mathrm{a, b, d, e}\}$ represents a set of products. The ternary relation gives information about the customers and the suppliers from whom they buy products. For instance, Customer 1 buys from Supplier P products a, b and e.

$\mathbb{K}$	P	N	R	K	S
1	abe	abe	ad	ab	a
2	ae	bde	abe	ae	e
3	abe	e	ab	ab	a
4	abe	be	ab	ab	e
5	ae	ae	abe	abd	a

Fig. 1. A triadic context

The derivation operators in Triadic Concept Analysis were introduced in [8]. If X_1, X_2 and X_3 are subsets of G, M and B respectively, then one can get:

$$X_1' = \{(a_j, a_k) \in M \times B \mid (a_i, a_j, a_k) \in I \text{ for all } a_i \in X_1\}.$$

$$(X_2, X_3)' = \{a_i \in G \mid (a_i, a_j, a_k) \in I \text{ for all } (a_j, a_k) \in X_2 \times X_3\}.$$

In a similar way, X_2', $(X_1, X_3)'$, X_3' and $(X_1, X_2)'$ can be defined. As shown in [13], the above family of operators, by setting a subset of objects, attributes or conditions (respectively) yields Galois connections. In this paper, we use the family of Galois connections associated with condition subsets. That is, given $\mathcal{C} \subseteq B$ we consider the Galois connection between the lattices $(2^M, \subseteq)$ and $(2^G, \subseteq)$ as the pair of mappings:

$$\begin{aligned} (-,\mathcal{C})' : 2^G &\longrightarrow 2^M \\ X_1 &\longmapsto (X_1, \mathcal{C})' \end{aligned} \qquad \begin{aligned} (-,\mathcal{C})' : 2^M &\longrightarrow 2^G \\ X_2 &\longmapsto (X_2, \mathcal{C})' \end{aligned}$$

Thus, for each $X_1 \subseteq G$ and $X_2 \subseteq M$, one has $X_2 \subseteq (X_1, \mathcal{C})'$ if and only if $X_1 \subseteq (X_2, \mathcal{C})'$.

In a similar way as in dyadic FCA, the composition of both derivation operators leads to the notion of triadic concept.

Definition 2. *A triadic concept of a triadic context is a triple* (A_1, A_2, A_3) *with* $A_1 \subseteq G$, $A_2 \subseteq M$, $A_3 \subseteq B$ *and* $A_1 \times A_2 \times A_3 \subseteq I$ *such that for* $X_1 \subseteq G, X_2 \subseteq M$, *and* $X_3 \subseteq B$ *with* $X_1 \times X_2 \times X_3 \subseteq I$, *the containments* $A_1 \subseteq X_1, A_2 \subseteq X_2$, *and* $A_3 \subseteq X_3$ *always lead to* $(A_1, A_2, A_3) = (X_1, X_2, X_3)$. *The subsets* A_1, A_2 *and* A_3 *are called the* extent, *the* intent *and the* modus *of the triadic concept* (A_1, A_2, A_3) *respectively.*

There are a few kinds of triadic implications with different semantics. Biedermann [3] defines a triadic implication to be an expression of the form: $(A \to B)_{\mathcal{C}}$ where A and B are attribute sets and $\mathcal{C}$ is a set of conditions. This implication is interpreted as: *If an object has all attributes from* A *under all conditions from* $\mathcal{C}$, *then it also has all attributes from* B *under all conditions from* $\mathcal{C}$. Its formal definition is the following:

Definition 3. *Let* $\mathbb{K} = \langle G, M, B, I \rangle$ *be a triadic context,* $X, Y \subseteq M$ *and* $\mathcal{C} \subseteq B$. *The implication* $(X \to Y)_{\mathcal{C}}$ *holds in the context* $\mathbb{K}$ *iff* $(X, \mathcal{C})' \subseteq (Y, \mathcal{C})'$.

Ganter and Obiedkov [6] consider three kinds of triadic implications. We will describe and make use of the following one which is stronger than Biedermann's expression and has another notation: $X \xrightarrow{\mathcal{C}} Y$, where $X, Y \subseteq M$ and $\mathcal{C} \subseteq B$. Such implication is called *conditional attribute implication* (CAI) and is read as "X implies Y under all conditions in $\mathcal{C}$ or any subset of it".

Definition 4 (Conditional attribute implication). *Let* $\mathbb{K} = \langle G, M, B, I \rangle$ *be a triadic context,* $X, Y \subseteq M$ *and* $\mathcal{C} \subseteq B$. *The implication* $X \xrightarrow{\mathcal{C}} Y$ *holds in the context* $\mathbb{K}$ *when* $(X, \{c\})' \subseteq (Y, \{c\})'$ *for all* $c \in \mathcal{C}$.

Notice that *CAI*s preserve the dyadic implications that hold for each elementary condition in $\mathcal{C}$. The following proposition relates both notions of implications and also shows that Biedermann's definition is weaker than the *CAI* definition.

Proposition 1 ([6]). *Let* $\mathbb{K} = \langle G, M, B, I\rangle$ *be a triadic context,* $X, Y \subseteq M$ *and* $\mathcal{C} \subseteq B$. *Then* $X \xrightarrow{\mathcal{C}} Y$ *holds in* $\mathbb{K}$ *iff* $(X \to Y)_{\mathcal{N}}$ *also holds in* $\mathbb{K}$ *for all* $\mathcal{N} \subseteq \mathcal{C}$.

The following example illustrates the above proposition.

Example 1. Let $\mathbb{K}$ be the triadic formal context given in Figure 1.

i) The *CAI* $N \xrightarrow{ae} P$ holds in $\mathbb{K}$ since the following implications are satisfied: $(N \to P)_a, (N \to P)_e, (N \to P)_{ae}$.
ii) The Biedermann's implication $(N \to P)_{abe}$ is satisfied but the *CAI* $N \xrightarrow{abe} P$ does not hold because, for instance, $(N \to P)_b$ is not satisfied.

Our objective in this paper is to provide inference mechanisms for a set of *CAI*s.

To that end, a sound and complete axiomatic system is needed. As mentioned earlier, we have introduced in [12] a novel logic for computing *CAI*s and reasoning about them. This logic is briefly presented in the following section.

3 CAIL: Conditional Attribute Implication Logic

In this section, we describe CAIL, a logic for reasoning about *CAI*s in the framework of TCA [12]. This logic is presented in a classical style by considering three pillars: the language, the semantics and the inference system.

Language: As it has been outlined, we use the following language: given an attribute set Ω and a set of conditions Γ, the set of well-formed formulas (hereinafter, formulas or implications) is $\mathcal{L}_{\Omega,\Gamma} = \{A \xrightarrow{\mathcal{C}} B \mid A, B \subseteq \Omega, \mathcal{C} \subseteq \Gamma\}$.

In the sequel we use X, Y, Z, W to mean subsets of attributes ($X, Y, Z, W \subseteq \Omega$) and $\mathcal{C}, \mathcal{C}_1, \mathcal{C}_2$ for subsets of conditions ($\mathcal{C}, \mathcal{C}_1, \mathcal{C}_2 \subseteq \Gamma$). For the sake of readability of formulas, we omit the brackets and commas (e.g. abc denotes the set $\{a, b, c\}$) and, as usual, the union is denoted by set juxtaposition (e.g. XY denotes $X \cup Y$).

Semantics: Based on Definition 4, the semantics is introduced by means of the notions of interpretation and model. From a language $\mathcal{L}_{\Omega,\Gamma}$, an interpretation is a triadic context $\mathbb{K} = \langle G, M, B, I\rangle$ such that $M = \Omega$ and $B = \Gamma$. A model for a formula $X \xrightarrow{\mathcal{C}} Y \in \mathcal{L}_{\Omega,\Gamma}$ is an interpretation that satisfies $X \xrightarrow{\mathcal{C}} Y$ in $\mathbb{K}$. In this case, we write $\mathbb{K} \models X \xrightarrow{\mathcal{C}} Y$.

As usual, for $\Sigma \subseteq \mathcal{L}_{\Omega,\Gamma}$, an interpretation $\mathbb{K}$ is a model for Σ (briefly, $\mathbb{K} \models \Sigma$) if $\mathbb{K} \models X \xrightarrow{\mathcal{C}} Y$ for each $X \xrightarrow{\mathcal{C}} Y \in \Sigma$. Similarly, $\Sigma \models X \xrightarrow{\mathcal{C}} Y$ states that $X \xrightarrow{\mathcal{C}} Y$ is a semantic consequence of Σ, i.e. every model for Σ is also a model for $X \xrightarrow{\mathcal{C}} Y$.

Syntactic inference: The syntactic derivation in CAIL is denoted by the symbol $\vdash_{\mathscr{C}}$ and covers two axiom schemes and four inference rules.

Definition 5. *The CAIL axiomatic system consists of the following rules:*

`[Non-constraint]` $\vdash_{\mathscr{C}} \emptyset \xrightarrow{\emptyset} \Omega$.
`[Inclusion]` $\vdash_{\mathscr{C}} XY \xrightarrow{\Gamma} X$.
`[Augmentation]` $X \xrightarrow{\mathcal{C}} Y \vdash_{\mathscr{C}} XZ \xrightarrow{\mathcal{C}} YZ$.
`[Transitivity]` $\{X \xrightarrow{\mathcal{C}_1} Y, Y \xrightarrow{\mathcal{C}_2} Z\} \vdash_{\mathscr{C}} X \xrightarrow{\mathcal{C}_1 \cap \mathcal{C}_2} Z$.
`[Conditional Decomposition]` $X \xrightarrow{\mathcal{C}_1\mathcal{C}_2} Y \vdash_{\mathscr{C}} X \xrightarrow{\mathcal{C}_1} Y$.
`[Conditional Composition]` $\{X \xrightarrow{\mathcal{C}_1} Y, Z \xrightarrow{\mathcal{C}_2} W\} \vdash_{\mathscr{C}} XZ \xrightarrow{\mathcal{C}_1\mathcal{C}_2} Y \cap W$.

The *derivation* notion is introduced as usual: For a given set $\Sigma \subseteq \mathcal{L}_{\Omega,\Gamma}$ and $\varphi \in \mathcal{L}_{\Omega,\Gamma}$, we state that φ is derived (or inferred) from Σ by using the CAIL axiomatic system, denoted by $\Sigma \vdash_{\mathscr{C}} \varphi$, if there exists a chain of formulas $\varphi_1, \ldots, \varphi_n \in \mathcal{L}_{\Omega,\Gamma}$ such that $\varphi_n = \varphi$ and, for all $1 \leq i \leq n$, φ_i is either an axiom, an implication in Σ or is obtained by applying the CAIL inference rules to formulas in $\{\varphi_j \mid 1 \leq j < i\}$.

Soundness and completeness: In [12], we prove that every model of Σ is a model of $X \xrightarrow{\mathcal{C}} Y$ iff such implication can be derived syntactically from Σ using the CAIL axiomatic system, i.e.

$$\Sigma \models X \xrightarrow{\mathcal{C}} Y \quad \text{if and only if} \quad \Sigma \vdash_{\mathscr{C}} X \xrightarrow{\mathcal{C}} Y$$

4 CAISL: Simplification Logic for *CAI*s

Once the preliminary results have been introduced, we now present a new axiomatic system which is more suitable for automated reasoning. We will use the same language and semantics provided in the previous section but give a novel equivalent axiomatic system based on simplification paradigm [4]. For this axiomatic system, the symbol $\vdash_{\mathscr{S}}$ denotes the syntactic derivation.

Definition 6. *The* CAISL *axiomatic system has two axiom schemes:*

`[Non-constraint]` $\vdash_{\mathscr{S}} \emptyset \xrightarrow{\emptyset} \Omega$.
`[Reflexivity]` $\vdash_{\mathscr{S}} X \xrightarrow{\Gamma} X$.

and four inference rules:

`[Decomposition]` $\{X \xrightarrow{\mathcal{C}_1\mathcal{C}_2} YZ\} \vdash_{\mathscr{S}} X \xrightarrow{\mathcal{C}_1} Y$.
`[Composition]` $\{X \xrightarrow{\mathcal{C}_1} Y, Z \xrightarrow{\mathcal{C}_2} W\} \vdash_{\mathscr{S}} XZ \xrightarrow{\mathcal{C}_1 \cap \mathcal{C}_2} YW$.
`[Conditional Composition]` $\{X \xrightarrow{\mathcal{C}_1} Y, Z \xrightarrow{\mathcal{C}_2} W\} \vdash_{\mathscr{S}} XZ \xrightarrow{\mathcal{C}_1\mathcal{C}_2} Y \cap W$.
`[Simplification]` *If* $X \cap Y = \emptyset$*,*

$$\{X \xrightarrow{\mathcal{C}_1} Y, XZ \xrightarrow{\mathcal{C}_2} W\} \vdash_{\mathscr{S}} XZ \smallsetminus Y \xrightarrow{\mathcal{C}_1 \cap \mathcal{C}_2} W \smallsetminus Y.$$

The two axiom schemes in CAISL have the following interpretations respectively: (1) all attributes hold for all objects under a void condition, and (2) X always implies itself under all conditions.

The key statement is that both axiomatic systems are equivalent as the following theorem proves. However, as we will show below, CAISL is more appropriate for developing automated methods to reason about implications.

Theorem 1 (Equivalence between CAIL and CAISL). *For any* $\Sigma \subseteq \mathcal{L}_{\Omega,\Gamma}$ *and* $X \xrightarrow{\mathcal{C}} Y \in \mathcal{L}_{\Omega,\Gamma}$*, one has*

$$\Sigma \vdash_{\mathscr{S}} X \xrightarrow{\mathcal{C}} Y \quad \text{if and only if} \quad \Sigma \vdash_{\mathscr{C}} X \xrightarrow{\mathcal{C}} Y$$

Proof. To prove the equivalence between both logics, we will show that the inference rules of CAISL can be derived from those in CAIL and vice versa.

i) Inference rules derived from CAIL

`[Reflexivity]:`

1. $X \xrightarrow{\Gamma} X$ .... Inclusion.

`[Decomposition]:`

1. $X \xrightarrow{\mathcal{C}_1\mathcal{C}_2} YZ$ ....Hypothesis.
2. $YZ \xrightarrow{\Gamma} Y$ .... Inclusion.
3. $X \xrightarrow{\mathcal{C}_1\mathcal{C}_2} Y$ ....1, 2 Trans.
4. $X \xrightarrow{\mathcal{C}_1} Y$ .....3 Cond. Decomp.

`[Composition]:`

1. $X \xrightarrow{\mathcal{C}_1} Y$ ....Hypothesis.
2. $Z \xrightarrow{\mathcal{C}_2} W$ ....Hypothesis.
3. $XZ \xrightarrow{\mathcal{C}_1} YZ$...........1 Augm.
4. $YZ \xrightarrow{\mathcal{C}_2} YW$..........2 Augm.
5. $XZ \xrightarrow{\mathcal{C}_1\cap\mathcal{C}_2} YW$.....3, 4 Trans.

`[Simplification]:`

1. $X \xrightarrow{\mathcal{C}_1} Y$ ....Hypothesis.
2. $XZ \xrightarrow{\mathcal{C}_2} W$ ....Hypothesis.
3. $XZ\smallsetminus Y \xrightarrow{\Gamma} X$ ....Inclusion.
4. $W \xrightarrow{\Gamma} W\smallsetminus Y$ ....Inclusion.
5. $XZ \xrightarrow{\mathcal{C}_2} W\smallsetminus Y$ .....2, 4 Trans.
6. $XZ\smallsetminus Y \xrightarrow{\mathcal{C}_1} Y$ ......3, 1 Trans.
7. $XZ\smallsetminus Y \xrightarrow{\mathcal{C}_1} XYZ$ ....6 Augm.
8. $XYZ \xrightarrow{\mathcal{C}_2} WY$ ....5 Augm.
9. $XZ\smallsetminus Y \xrightarrow{\mathcal{C}_1\cap\mathcal{C}_2} WY$ 7, 8 Trans.
10. $XZ\smallsetminus Y \xrightarrow{\mathcal{C}_1\cap\mathcal{C}_2} W\smallsetminus Y$. 9 Decomp.

ii) Inference rules derived from CAISL

`[Inclusion]:`

1. $XY \xrightarrow{\Gamma} XY$ .... Reflexivity.
2. $XY \xrightarrow{\Gamma} Y$ ....1 Decomp.

`[Augmentation]:`

1. $X \xrightarrow{\mathcal{C}} Y$ ....Hypothesis.
2. $Z \xrightarrow{\Gamma} Z$.............Reflexivity.
3. $XZ \xrightarrow{\mathcal{C}} YZ$ ....1, 2 Comp.

`[Transitivity]:`

1. $X \xrightarrow{\mathcal{C}_1} Y$...........Hypothesis.
2. $Y \xrightarrow{\mathcal{C}_2} Z$........... Hypothesis.
3. $X \xrightarrow{\mathcal{C}_1} Y\smallsetminus X$ ....1 Decomp.
4. $Y \xrightarrow{\mathcal{C}_2} Z\smallsetminus Y$.........2 Decomp.
5. $X \xrightarrow{\Gamma} X$............Reflexivity.
6. $X \xrightarrow{\Gamma} \emptyset$.............5 Decomp.
7. $XY \xrightarrow{\mathcal{C}_2} Z\smallsetminus Y$ ......4, 6 Comp.
8. $Y\smallsetminus X \xrightarrow{\Gamma} Y\smallsetminus X$ ....Reflexivity.
9. $X \xrightarrow{\mathcal{C}_1\cap\mathcal{C}_2} Z\smallsetminus Y$ .....3, 7 Simp.
10. $X \xrightarrow{\mathcal{C}_1\cap\mathcal{C}_2} YZ$ ......1, 9 Comp.
11. $X \xrightarrow{\mathcal{C}_1\cap\mathcal{C}_2} Z$ ....10 Decomp.

`[Conditional Decomposition]:`

1. $X \xrightarrow{\mathcal{C}_1\mathcal{C}_2} Y$.........Hypothesis.
2. $X \xrightarrow{\mathcal{C}_1} Y$............1 Decomp.

□

Since the two axiomatic systems are equivalent, in the sequel we will omit the subscript in the syntactic derivation symbol using simply $\vdash$.

5 CAISL Equivalences

In this section, we introduce several results which constitute the basis of the automated reasoning method that will be introduced in the next section. These results illustrate how we can use CAISL as a framework to syntactically transform and simplify a set of *CAI*s while entirely preserving their semantics. This is the common feature of the family of Simplification Logics.

The notion of equivalence is introduced as usual: two sets of *CAI*s, Σ_1 and Σ_2, are equivalent, denoted by $\Sigma_1 \equiv \Sigma_2$, when their models are the same. Equivalently, $\Sigma_1 \equiv \Sigma_2$ iff $\Sigma_1 \vdash \varphi$ for all $\varphi \in \Sigma_2$, and $\Sigma_2 \vdash \varphi$ for all $\varphi \in \Sigma_1$.

Lemma 1. *The following equivalences hold:*

$$\{X \xrightarrow{\mathcal{C}_1} Y, X \xrightarrow{\mathcal{C}_2} W\} \equiv \{X \xrightarrow{\mathcal{C}_1 \cap \mathcal{C}_2} YW, X \xrightarrow{\mathcal{C}_1 \smallsetminus \mathcal{C}_2} Y, X \xrightarrow{\mathcal{C}_2 \smallsetminus \mathcal{C}_1} W\} \quad (1)$$

$$\{X \xrightarrow{\mathcal{C}_1} Y, XV \xrightarrow{\mathcal{C}_2} W\} \equiv \{X \xrightarrow{\mathcal{C}_1} Y, XV \xrightarrow{\mathcal{C}_2 \smallsetminus \mathcal{C}_1} W, X(V \smallsetminus Y) \xrightarrow{\mathcal{C}_1 \cap \mathcal{C}_2} W \smallsetminus Y\} \quad (2)$$

Proof. For Equivalence (1), first, we prove that $X \xrightarrow{\mathcal{C}_1 \cap \mathcal{C}_2} YW$, $X \xrightarrow{\mathcal{C}_1 \smallsetminus \mathcal{C}_2} Y$, and $X \xrightarrow{\mathcal{C}_2 \smallsetminus \mathcal{C}_1} W$ can be inferred from $\{X \xrightarrow{\mathcal{C}_1} Y, X \xrightarrow{\mathcal{C}_2} W\}$:

- By applying Composition to $X \xrightarrow{\mathcal{C}_1} Y$ and $X \xrightarrow{\mathcal{C}_2} W$, we get $X \xrightarrow{\mathcal{C}_1 \cap \mathcal{C}_2} YW$.
- $X \xrightarrow{\mathcal{C}_1 \smallsetminus \mathcal{C}_2} Y$ and $X \xrightarrow{\mathcal{C}_2 \smallsetminus \mathcal{C}_1} W$ are obtained by Decomposition.

On the other hand, we prove that $X \xrightarrow{\mathcal{C}_1} Y$ and $X \xrightarrow{\mathcal{C}_2} W$ can be inferred from $\{X \xrightarrow{\mathcal{C}_1 \cap \mathcal{C}_2} YW, X \xrightarrow{\mathcal{C}_1 \smallsetminus \mathcal{C}_2} Y, X \xrightarrow{\mathcal{C}_2 \smallsetminus \mathcal{C}_1} W\}$ by applying Conditional Composition.

For Equivalence (2), from $\{X \xrightarrow{\mathcal{C}_1} Y, XV \xrightarrow{\mathcal{C}_2} W\}$, we infer $XV \xrightarrow{\mathcal{C}_2 \smallsetminus \mathcal{C}_1} W$ by applying Decomposition to $XV \xrightarrow{\mathcal{C}_2} W$. In addition, we infer $XV \smallsetminus Y \xrightarrow{\mathcal{C}_1 \cap \mathcal{C}_2} W \smallsetminus Y$ by applying Simplification to $X \xrightarrow{\mathcal{C}_1} Y$ and $XV \xrightarrow{\mathcal{C}_2} W$.

Finally, $\{X \xrightarrow{\mathcal{C}_1} Y, XV \xrightarrow{\mathcal{C}_2 \smallsetminus \mathcal{C}_1} W, XV \smallsetminus Y \xrightarrow{\mathcal{C}_1 \cap \mathcal{C}_2} W \smallsetminus Y\} \vdash XV \xrightarrow{\mathcal{C}_2} W$ is proved. By applying Reflexivity and Decomposition, we get $XV \xrightarrow{\mathcal{C}_1 \cap \mathcal{C}_2} XV \smallsetminus Y$ and, by Transitivity with $XV \smallsetminus Y \xrightarrow{\mathcal{C}_1 \cap \mathcal{C}_2} W \smallsetminus Y$, one has $XV \xrightarrow{\mathcal{C}_1 \cap \mathcal{C}_2} W \smallsetminus Y$. Now, by applying Composition to $X \xrightarrow{\mathcal{C}_1} Y$ and $XV \xrightarrow{\mathcal{C}_1 \cap \mathcal{C}_2} W \smallsetminus Y$, we infer $XV \xrightarrow{\mathcal{C}_1 \cap \mathcal{C}_2} WY$ and, by Decomposition, $XV \xrightarrow{\mathcal{C}_1 \cap \mathcal{C}_2} W$. At last, by applying Conditional Composition to $XV \xrightarrow{\mathcal{C}_1 \cap \mathcal{C}_2} W$ and $XV \xrightarrow{\mathcal{C}_2 \smallsetminus \mathcal{C}_1} W$, we obtain $XV \xrightarrow{\mathcal{C}_2} W$. □

The following theorem highlights a common characteristic of Simplification Logics, which shows that inference rules can be read as equivalences that allow redundancy removal.

Theorem 2. *The following equivalences hold:*

`Axiom Eq.`: $\{X \xrightarrow{\emptyset} Y\} \equiv \{X \xrightarrow{\mathcal{C}} \emptyset\} \equiv \emptyset$

`Decomposition Eq.`: $\{X \xrightarrow{\mathcal{C}} Y\} \equiv \{X \xrightarrow{\mathcal{C}} Y \smallsetminus X\}$

`Composition Eq.`: $\{X \xrightarrow{\mathcal{C}} Y, X \xrightarrow{\mathcal{C}} W\} \equiv \{X \xrightarrow{\mathcal{C}} YW\}$

`Conditional Composition Eq.`: $\{X \xrightarrow{\mathcal{C}_1} Y, X \xrightarrow{\mathcal{C}_2} Y\} \equiv \{X \xrightarrow{\mathcal{C}_1 \mathcal{C}_2} Y\}$

`Simplification Eq.:` *If* $X \cap Y = \emptyset$, *then*

$$\{X \xrightarrow{\mathcal{C}_1\mathcal{C}_2} Y, XV \xrightarrow{\mathcal{C}_2} W\} \equiv \{X \xrightarrow{\mathcal{C}_1\mathcal{C}_2} Y, XV\smallsetminus Y \xrightarrow{\mathcal{C}_2} W\smallsetminus Y\}$$

Proof. The first equivalence is straightforward because both implications are axioms. For the rest of equivalences, the left to right inference is directly obtained by applying the homonymous inference rule. Thus, we prove the right to left inference:

i) $X \xrightarrow{\mathcal{C}} XY$ is inferred by Composition of $X \xrightarrow{\mathcal{C}} Y\smallsetminus X$ and $X \xrightarrow{\mathcal{C}} X$ obtained by reflexivity. Then, by applying Decomposition, one has $X \xrightarrow{\mathcal{C}} Y$.
ii) $X \xrightarrow{\mathcal{C}} Y$ and $X \xrightarrow{\mathcal{C}} W$ are inferred by applying Decomposition to $X \xrightarrow{\mathcal{C}} YW$.
iii) $X \xrightarrow{\mathcal{C}_1} Y$ and $X \xrightarrow{\mathcal{C}_2} Y$ are inferred from $X \xrightarrow{\mathcal{C}_1\mathcal{C}_2} Y$ by applying Decomposition.
iv) It is a consequence of Axiom Equivalence and Equivalence (2) in Lemma 1.

□

This section has been devoted to equivalences in CAISL of a *CAI*s set in order to remove redundancy or, dually, to extend the set. The effect depends on the direction we apply the equivalence. In next section, we are going to use other equivalences where the empty set plays a main role. The Deduction Theorem presented below gives to the empty set such a role. This theorem establishes the necessary and sufficient condition to ensure the derivability of a *CAI* from a set of *CAI*s.

6 Automated reasoning

This section shows the merits of CAISL for the development of automated methods. Specifically, we present a method that checks whether a *CAI* is derived from a set of *CAI*s. The next theorem is the core of our approach in the design of the automated prover.

Theorem 3 (Deduction). *For any* $\Sigma \subseteq \mathcal{L}_{\Omega,\Gamma}$ *and* $X \xrightarrow{\mathcal{C}} Y \in \mathcal{L}_{\Omega,\Gamma}$, *one has*

$$\Sigma \vdash X \xrightarrow{\mathcal{C}} Y \text{ if and only if } \Sigma \cup \{\emptyset \xrightarrow{\mathcal{C}} X\} \vdash \emptyset \xrightarrow{\mathcal{C}} Y$$

Proof. Straightforwardly, we have $\Sigma \vdash X \xrightarrow{\mathcal{C}} Y$ implies $\Sigma \cup \{\emptyset \xrightarrow{\mathcal{C}} X\} \vdash \emptyset \xrightarrow{\mathcal{C}} Y$. Conversely, assuming $\Sigma \cup \{\emptyset \xrightarrow{\mathcal{C}} X\} \vdash \emptyset \xrightarrow{\mathcal{C}} Y$, we have to prove that $\mathbb{K} \models \Sigma$ implies $\mathbb{K} \models X \xrightarrow{\mathcal{C}} Y$ for each model $\mathbb{K}$.

Consider $\mathbb{K} = \langle G, M, B, I\rangle$ as a model of Σ. In order to prove $(X, \{c\})' \subseteq (Y, \{c\})'$ for all $c \in C$ in $\mathbb{K}$, we build the context $\mathbb{K}_1 = \langle G_1, M, B, I_1\rangle$ where $G_1 = (X, \{c\})'$ and $I_1 = I \cap (G_1 \times M \times B)$.

Since $\mathbb{K} \models \Sigma$, we have $\mathbb{K}_1 \models \Sigma \cup \{\emptyset \xrightarrow{\{c\}} X\}$ and therefore, by hypothesis, $\mathbb{K}_1 \models \{\emptyset \xrightarrow{\{c\}} Y\}$. That is, $(Y, \{c\})' \supseteq (\emptyset, \{c\})' = G_1 = (X, \{c\})'$.

If we go back to the original triadic context $\mathbb{K}$, $(X, \{c\})'$ remains unchanged whereas $(Y, \{c\})'$ could grow up. Therefore, in $\mathbb{K}$, one has $(X, \{c\})' \subseteq (Y, \{c\})'$ for all $c \in C$. □

Function CAISL-Prover(Σ,$X \xrightarrow{\mathcal{C}} Y$)

input : A set of implications Σ, and a *CAI* $X \xrightarrow{\mathcal{C}} Y$
output: A boolean answer
begin
  $\Delta_X := X \times \mathcal{C}$
  $\Delta_Y := (Y \times \mathcal{C}) \smallsetminus (X \times \mathcal{C})$
  repeat
    flag:=false
    foreach $U \xrightarrow{\mathcal{C}_1} V \in \Sigma$ *with* $\mathcal{C}_1 \cap \mathcal{C} \neq \emptyset$ **do**
      $\Delta_C := \{c \in \mathcal{C}_1 \cap \mathcal{C} \mid U \times \{c\} \subseteq \Delta_X\}$
      if $\Delta_C \neq \emptyset$ **then** .... Equivalence (4)
        $\Delta_X := \Delta_X \cup (V \times \Delta_C)$
        $\Delta_Y := \Delta_Y \smallsetminus (V \times \Delta_C)$
        $\Sigma := \Sigma \smallsetminus \{U \xrightarrow{\mathcal{C}_1} V\}$
        $\mathcal{C}_1 := \mathcal{C}_1 \smallsetminus \Delta_C$
        if $\mathcal{C}_1 \neq \emptyset$ **then** $\Sigma := \Sigma \cup \{U \xrightarrow{\mathcal{C}_1} V\}$
        flag:=true
      $\Delta_C := \{c \in \mathcal{C}_1 \cap \mathcal{C} \mid V \times \{c\} \subseteq \Delta_X\}$
      if $\Delta_C \neq \emptyset$ **then** .... Equivalence (5)
        if $\Delta_C = \mathcal{C}_1$ **then** $\Sigma := \Sigma \smallsetminus \{U \xrightarrow{\mathcal{C}_1} V\}$
        else $\Sigma := (\Sigma \smallsetminus \{U \xrightarrow{\mathcal{C}_1} V\}) \cup \{U \xrightarrow{\mathcal{C}_1 \smallsetminus \Delta_C} V\}$
  until $(\Delta_Y = \emptyset)$ *or* (flag=false)
  return the boolean value $(\Delta_Y = \emptyset)$

Theorem 3 guides the design of the automated prover. To check that the formula $X \xrightarrow{\mathcal{C}} Y$ is inferred from the set Σ we apply the family of simplification equivalences iteratively - while it is possible - to the set $\Sigma \cup \{\emptyset \xrightarrow{\mathcal{C}} X\}$ looking for $\emptyset \xrightarrow{\mathcal{C}} Y$.

The following proposition revisits Theorem 2 by instantiating the particular case of having the empty premise.

Proposition 2. *The following equivalences hold:*

$$\{\emptyset \xrightarrow{\mathcal{C}_1} X, U \xrightarrow{\mathcal{C}_2} V\} \equiv \{\emptyset \xrightarrow{\mathcal{C}_1} X, U \smallsetminus X \xrightarrow{\mathcal{C}_1 \cap \mathcal{C}_2} V \smallsetminus X, U \xrightarrow{\mathcal{C}_2 \smallsetminus \mathcal{C}_1} V\} \tag{3}$$

$$\{\emptyset \xrightarrow{\mathcal{C}_1} X, U \xrightarrow{\mathcal{C}_2} V\} \equiv \{\emptyset \xrightarrow{\mathcal{C}_1 \cap \mathcal{C}_2} XV, \emptyset \xrightarrow{\mathcal{C}_1 \smallsetminus \mathcal{C}_2} X, U \xrightarrow{\mathcal{C}_2 \smallsetminus \mathcal{C}_1} V\},\ \textit{when } U \subseteq X \tag{4}$$

$$\{\emptyset \xrightarrow{\mathcal{C}_1} X, U \xrightarrow{\mathcal{C}_2} V\} \equiv \{\emptyset \xrightarrow{\mathcal{C}_1} X, U \xrightarrow{\mathcal{C}_2 \smallsetminus \mathcal{C}_1} V\},\ \textit{when } V \subseteq X \tag{5}$$

Proof. Equivalence (3) is a particular case of Equivalence (2). In particular, when $U \subseteq X$, Equivalence (4) is obtained from (3) by applying Conditional Composition and Composition equivalences:

$$\begin{aligned}\{\emptyset \xrightarrow{\mathcal{C}_1} X, U \xrightarrow{\mathcal{C}_2} V\} &\equiv \{\emptyset \xrightarrow{\mathcal{C}_1} X, \emptyset \xrightarrow{\mathcal{C}_1 \cap \mathcal{C}_2} V \smallsetminus X, U \xrightarrow{\mathcal{C}_2 \smallsetminus \mathcal{C}_1} V\} \\ &\equiv \{\emptyset \xrightarrow{\mathcal{C}_1 \smallsetminus \mathcal{C}_2} X, \emptyset \xrightarrow{\mathcal{C}_1 \cap \mathcal{C}_2} X, \emptyset \xrightarrow{\mathcal{C}_1 \cap \mathcal{C}_2} V \smallsetminus X, U \xrightarrow{\mathcal{C}_2 \smallsetminus \mathcal{C}_1} V\} \\ &\equiv \{\emptyset \xrightarrow{\mathcal{C}_1 \smallsetminus \mathcal{C}_2} X, \emptyset \xrightarrow{\mathcal{C}_1 \cap \mathcal{C}_2} XV, U \xrightarrow{\mathcal{C}_2 \smallsetminus \mathcal{C}_1} V\}\end{aligned}$$

Analogously, when $V \subseteq X$, by applying Equivalence (3) and Axiom equivalence, one has Equivalence (5):

$$\begin{aligned}\{\emptyset \xrightarrow{\mathcal{C}_1} X, U \xrightarrow{\mathcal{C}_2} V\} &\equiv \{\emptyset \xrightarrow{\mathcal{C}_1} X, U \smallsetminus X \xrightarrow{\mathcal{C}_1 \cap \mathcal{C}_2} \emptyset, U \xrightarrow{\mathcal{C}_2 \smallsetminus \mathcal{C}_1} V\} \\ &\equiv \{\emptyset \xrightarrow{\mathcal{C}_1} X, U \xrightarrow{\mathcal{C}_2 \smallsetminus \mathcal{C}_1} V\}\end{aligned}$$

□

The equivalences introduced in Proposition 2 constitute the core of the function called `CAISL-Prover`, which acts as an automated prover for CAISL. The prover works by splitting the original formula into its left and right hand sides (see Theorem 2) and, by applying the equivalences, we check whether its right side can be reduced to the empty set. The derivability is proved if and only if such reduction is fulfilled.

Finally, we conclude this section with an illustrative example.

Steep	State
0	$\Delta_X = \{(M,a),(M,b),(M,d),(L,a),(L,b),(L,d)\}$ $\Delta_Y = \{(Q,a),(Q,b),(Q,d)\}$ $\Sigma = \{\mathrm{Q} \xrightarrow{\mathrm{de}} \mathrm{M}, \mathrm{M} \xrightarrow{\mathrm{a}} \mathrm{T}, \mathrm{Q} \xrightarrow{\mathrm{bd}} \mathrm{L}, \mathrm{ML} \xrightarrow{\mathrm{bde}} \mathrm{Q}, \mathrm{T} \xrightarrow{\mathrm{ab}} \mathrm{RL}, \mathrm{R} \xrightarrow{\mathrm{ae}} \mathrm{Q}\}$
1	$\Delta_X = \{(M,a),(M,b),(M,d),(L,a),(L,b),(L,d)\}$ $\Delta_Y = \{(Q,a),(Q,b),(Q,d)\}$ $\Sigma = \{\mathbf{Q} \xrightarrow{\mathbf{\cancel{d}e}} \mathbf{M}, \mathrm{M} \xrightarrow{\mathrm{a}} \mathrm{T}, \mathrm{Q} \xrightarrow{\mathrm{bd}} \mathrm{L}, \mathrm{ML} \xrightarrow{\mathrm{bde}} \mathrm{Q}, \mathrm{T} \xrightarrow{\mathrm{ab}} \mathrm{RL}, \mathrm{R} \xrightarrow{\mathrm{ae}} \mathrm{Q}\}$
2	$\Delta_X = \{(M,a),(M,b),(M,d),(L,a),(L,b),(L,d),(T,a)\}$ $\Delta_Y = \{(Q,a),(Q,b),(Q,d)\}$ $\Sigma = \{\mathrm{Q} \xrightarrow{\mathrm{e}} \mathrm{M}, \cancel{\mathbf{M} \xrightarrow{\mathbf{a}} \mathbf{T}}, \mathrm{Q} \xrightarrow{\mathrm{bd}} \mathrm{L}, \mathrm{ML} \xrightarrow{\mathrm{bde}} \mathrm{Q}, \mathrm{T} \xrightarrow{\mathrm{ab}} \mathrm{RL}, \mathrm{R} \xrightarrow{\mathrm{ae}} \mathrm{Q}\}$
3	$\Delta_X = \{(M,a),(M,b),(M,d),(L,a),(L,b),(L,d),(T,a)\}$ $\Delta_Y = \{(Q,a),(Q,b),(Q,d)\}$ $\Sigma = \{\mathrm{Q} \xrightarrow{\mathrm{e}} \mathrm{M}, \cancel{\mathbf{Q} \xrightarrow{\mathbf{bd}} \mathbf{L}}, \mathrm{ML} \xrightarrow{\mathrm{bde}} \mathrm{Q}, \mathrm{T} \xrightarrow{\mathrm{ab}} \mathrm{RL}, \mathrm{R} \xrightarrow{\mathrm{ae}} \mathrm{Q}\}$
4	$\Delta_X = \{(M,a),(M,b),(M,d),(L,a),(L,b),(L,d),(T,a),(Q,b),(Q,d)\}$ $\Delta_Y = \{(Q,a)\}$ $\Sigma = \{\mathrm{Q} \xrightarrow{\mathrm{e}} \mathrm{M}, \mathbf{ML} \xrightarrow{\mathbf{\cancel{b}\cancel{d}e}} \mathbf{Q}, \mathrm{T} \xrightarrow{\mathrm{ab}} \mathrm{RL}, \mathrm{R} \xrightarrow{\mathrm{ae}} \mathrm{Q}\}$
5	$\Delta_X = \{(M,a),(M,b),(M,d),(L,a),(L,b),(L,d),(T,a),(Q,b),(Q,d),(R,a)\}$ $\Delta_Y = \{(Q,a)\}$ $\Sigma = \{\mathrm{Q} \xrightarrow{\mathrm{e}} \mathrm{M}, \mathrm{ML} \xrightarrow{\mathrm{e}} \mathrm{Q}, \mathbf{T} \xrightarrow{\mathbf{\cancel{a}b}} \mathbf{RL}, \mathrm{R} \xrightarrow{\mathrm{ae}} \mathrm{Q}\}$
6	$\Delta_X = \{(M,a),(M,b),(M,d),(L,a),(L,b),(L,d),(T,a),(Q,b),(Q,d),(R,a),(Q,a)\}$ $\Delta_Y = \emptyset$ $\Sigma = \{\mathrm{Q} \xrightarrow{\mathrm{e}} \mathrm{M}, \mathrm{ML} \xrightarrow{\mathrm{e}} \mathrm{Q}, \mathrm{T} \xrightarrow{\mathrm{b}} \mathrm{R}, \mathbf{R} \xrightarrow{\mathbf{\cancel{a}e}} \mathbf{Q}\}$
Output	Return `TRUE`

Table 1. Illustration of the derivability of $\mathrm{ML} \xrightarrow{\mathrm{abd}} \mathrm{Q}$

Example 2. Let $\Sigma = \{\mathrm{Q} \xrightarrow{\mathrm{de}} \mathrm{M}, \mathrm{M} \xrightarrow{\mathrm{a}} \mathrm{T}, \mathrm{Q} \xrightarrow{\mathrm{bd}} \mathrm{L}, \mathrm{ML} \xrightarrow{\mathrm{bde}} \mathrm{Q}, \mathrm{T} \xrightarrow{\mathrm{ab}} \mathrm{RL}, \mathrm{R} \xrightarrow{\mathrm{ae}} \mathrm{Q}\}$ be a set of *CAI*s. The `CAISL-Prover` function proves that $\mathrm{ML} \xrightarrow{\mathrm{abd}} \mathrm{Q}$ is inferred from Σ (see Table 1) and $\mathrm{T} \xrightarrow{\mathrm{ab}} \mathrm{Q}$ is not inferred from Σ (see Table 2).

Step	State
0	$\Delta_X = \{(T,a),(T,b)\}$ $\Delta_Y = \{(Q,a),(Q,b)\}$ $\Sigma = \{\mathrm{Q} \xrightarrow{\mathrm{de}} \mathrm{M}, \mathrm{M} \xrightarrow{\mathrm{a}} \mathrm{T}, \mathrm{Q} \xrightarrow{\mathrm{bd}} \mathrm{L}, \mathrm{ML} \xrightarrow{\mathrm{bde}} \mathrm{Q}, \mathrm{T} \xrightarrow{\mathrm{ab}} \mathrm{RL}, \mathrm{R} \xrightarrow{\mathrm{ae}} \mathrm{Q}\}$
1	$\Delta_X = \{(T,a),(T,b)\}$ $\Delta_Y = \{(Q,a),(Q,b)\}$ $\Sigma = \{\mathbf{Q} \xrightarrow{\mathbf{de}} \mathbf{M}, \mathrm{M} \xrightarrow{\mathrm{a}} \mathrm{T}, \mathrm{Q} \xrightarrow{\mathrm{bd}} \mathrm{L}, \mathrm{ML} \xrightarrow{\mathrm{bde}} \mathrm{Q}, \mathrm{T} \xrightarrow{\mathrm{ab}} \mathrm{RL}, \mathrm{R} \xrightarrow{\mathrm{ae}} \mathrm{Q}\}$
2	$\Delta_X = \{(T,a),(T,b)\}$ $\Delta_Y = \{(Q,a),(Q,b)\}$ $\Sigma = \{\mathrm{Q} \xrightarrow{\mathrm{de}} \mathrm{M}, \cancel{\mathbf{M} \xrightarrow{\mathbf{a}} \mathbf{T}}, \mathrm{Q} \xrightarrow{\mathrm{bd}} \mathrm{L}, \mathrm{ML} \xrightarrow{\mathrm{bde}} \mathrm{Q}, \mathrm{T} \xrightarrow{\mathrm{ab}} \mathrm{RL}, \mathrm{R} \xrightarrow{\mathrm{ae}} \mathrm{Q}\}$
3	$\Delta_X = \{(T,a),(T,b)\}$ $\Delta_Y = \{(Q,a),(Q,b)\}$ $\Sigma = \{\mathrm{Q} \xrightarrow{\mathrm{de}} \mathrm{M}, \mathbf{Q} \xrightarrow{\mathbf{bd}} \mathbf{L}, \mathrm{ML} \xrightarrow{\mathrm{bde}} \mathrm{Q}, \mathrm{T} \xrightarrow{\mathrm{ab}} \mathrm{RL}, \mathrm{R} \xrightarrow{\mathrm{ae}} \mathrm{Q}\}$
4	$\Delta_X = \{(T,a),(T,b)\}$ $\Delta_Y = \{(Q,a),(Q,b)\}$ $\Sigma = \{\mathrm{Q} \xrightarrow{\mathrm{de}} \mathrm{M}, \mathrm{Q} \xrightarrow{\mathrm{bd}} \mathrm{L}, \mathbf{ML} \xrightarrow{\mathbf{bde}} \mathbf{Q}, \mathrm{T} \xrightarrow{\mathrm{ab}} \mathrm{RL}, \mathrm{R} \xrightarrow{\mathrm{ae}} \mathrm{Q}\}$
5	$\Delta_X = \{(T,a),(T,b),(R,a),(R,b),(L,a),(L,b)\}$ $\Delta_Y = \{(Q,a),(Q,b)\}$ $\Sigma = \{\mathrm{Q} \xrightarrow{\mathrm{de}} \mathrm{M}, \mathrm{Q} \xrightarrow{\mathrm{bd}} \mathrm{L}, \mathrm{ML} \xrightarrow{\mathrm{bde}} \mathrm{Q}, \cancel{\mathbf{T} \xrightarrow{\mathbf{ab}} \mathbf{RL}}, \mathrm{R} \xrightarrow{\mathrm{ae}} \mathrm{Q}\}$
6	$\Delta_X = \{(T,a),(T,b),(R,a),(R,b),(L,a),(L,b),(Q,a)\}$ $\Delta_Y = \{(Q,b))\}$ $\Sigma = \{\mathrm{Q} \xrightarrow{\mathrm{de}} \mathrm{M}, \mathrm{Q} \xrightarrow{\mathrm{bd}} \mathrm{L}, \mathrm{ML} \xrightarrow{\mathrm{bde}} \mathrm{Q}, \mathbf{R} \xrightarrow{\cancel{\mathbf{a}}\mathbf{e}} \mathbf{Q}\}$
	Loop 2
7	$\Delta_X = \{(T,a),(T,b),(R,a),(R,b),(L,a),(L,b),(Q,a)\}$ $\Delta_Y = \{(Q,b))\}$ $\Sigma = \{\mathbf{Q} \xrightarrow{\mathbf{de}} \mathbf{M}, \mathrm{Q} \xrightarrow{\mathrm{bd}} \mathrm{L}, \mathrm{ML} \xrightarrow{\mathrm{bde}} \mathrm{Q}, \mathrm{R} \xrightarrow{\mathrm{e}} \mathrm{Q}\}$
8	$\Delta_X = \{(T,a),(T,b),(R,a),(R,b),(L,a),(L,b),(Q,a)\}$ $\Delta_Y = \{(Q,b))\}$ $\Sigma = \{\mathrm{Q} \xrightarrow{\mathrm{de}} \mathrm{M}, \mathbf{Q} \xrightarrow{\cancel{\mathbf{b}}\mathbf{d}} \mathbf{L}, \mathrm{ML} \xrightarrow{\mathrm{bde}} \mathrm{Q}, \mathrm{R} \xrightarrow{\mathrm{e}} \mathrm{Q}\}$
9	$\Delta_X = \{(T,a),(T,b),(R,a),(R,b),(L,a),(L,b),(Q,a)\}$ $\Delta_Y = \{(Q,b))\}$ $\Sigma = \{\mathrm{Q} \xrightarrow{\mathrm{de}} \mathrm{M}, \mathrm{Q} \xrightarrow{\mathrm{d}} \mathrm{L}, \mathbf{ML} \xrightarrow{\mathbf{bde}} \mathbf{Q}, \mathrm{R} \xrightarrow{\mathrm{e}} \mathrm{Q}\}$
10	$\Delta_X = \{(T,a),(T,b),(R,a),(R,b),(L,a),(L,b),(Q,a)\}$ $\Delta_Y = \{(Q,b))\}$ $\Sigma = \{\mathrm{Q} \xrightarrow{\mathrm{de}} \mathrm{M}, \mathrm{Q} \xrightarrow{\mathrm{d}} \mathrm{L}, \mathrm{ML} \xrightarrow{\mathrm{bde}} \mathrm{Q}, \mathbf{R} \xrightarrow{\mathbf{e}} \mathbf{Q}\}$
Output	Return `FALSE`

Table 2. Illustration of the non derivability of $\mathrm{T} \xrightarrow{\mathrm{ab}} \mathrm{Q}$

7 Conclusion and Future Work

We have proposed a logic named CAISL to deal with conditional attribute implications in Triadic Concept Analysis. Its soundness and completeness have been proved by establishing the equivalence between the axioms and rules of CAISL and those of CAIL - a recently developed solution [12]. This novel approach is strongly based on the Simplification paradigm which is a useful formalism to develop automated methods. In this direction, CAISL has been used to build an automated prover to check the derivability of a CAI from a set of CAIs, which is an important issue in data and knowledge management.

The use of a logic-based approach is a challenging but very interesting issue that has not been explored in the TCA framework yet. Our short-term research activity is to adapt our proposal to other kinds of triadic implications and analyze the interplay between them.

Acknowledgment

This work is supported by regional project TIN2014-59471-P of the Science and Innovation Ministry of Spain, co-funded by the European Regional Development Fund (ERDF) as well as by a discovery grant from the Natural Sciences and Engineering Research Council of Canada (NSERC).

References

1. Armstrong, W.: Dependency structures of data base relationships. Proc. IFIP Congress. North Holland, Amsterdam pp. 580–583 (1974)
2. Biedermann, K.: How triadic diagrams represent conceptual structures. In: ICCS. pp. 304–317 (1997)
3. Biedermann, K.: Powerset trilattices. In: ICCS. pp. 209–224 (1998)
4. Cordero, P., Enciso, M., Mora, A., P. de Guzmn, I.: Slfd logic: Elimination of data redundancy in knowledge representation. In: IBERAMIA. Lecture Notes in Computer Science, vol. 2527, pp. 141–150. Springer Berlin Heidelberg (2002)
5. Fagin, R.: Functional dependencies in a relational database and propositional logic. IBM. Journal of research and development 21 (6), 534–544 (1977)
6. Ganter, B., Obiedkov, S.A.: Implications in triadic formal contexts. In: ICCS. pp. 186–195 (2004)
7. Jäschke, R., Hotho, A., Schmitz, C., Ganter, B., Stumme, G.: Trias - an algorithm for mining iceberg tri-lattices. In: ICDM. pp. 907–911 (2006)
8. Lehmann, F., Wille, R.: A triadic approach to formal concept analysis. In: ICCS. pp. 32–43 (1995)
9. Maier, D.: The theory of Relational Databases. Computer Science Press (1983)
10. Missaoui, R., Kwuida, L.: Mining triadic association rules from ternary relations. In: Formal Concept Analysis - 9th International Conference, ICFCA 2011, Nicosia, Cyprus, May 2-6, 2011. Proceedings. pp. 204–218 (2011)
11. Peirce, C.S.: Collected Papers. Harvard University Press, Cambridge (1931-1935)
12. Rodríguez-Lorenzo, E., Cordero, P., Enciso, M., Missaoui, R., Mora, A.: An axiomatic system for conditional attribute implications in triadic concept analysis. Submitted to International Journal of Intelligent Systems (2016)
13. Wille, R.: The basic theorem of triadic concept analysis. Order 12(2), 149–158 (1995)

Building a Domain Knowledge Model Based on a Concept Lattice Integrating Expert Constraints

My Thao Tang[1], Aleksey Buzmakov[2],
Yannick Toussaint[1], and Amedeo Napoli[1]

[1] LORIA (CNRS - Inria NGE - U. de Lorraine), Vandœuvre-lès-Nancy, France
[2] National Research University Higher School of Economics, Perm, Russia
my-thao.tang@loria.fr,avbuzmakov@hse.ru,
yannick.toussaint@loria.fr,amedeo.napoli@loria.fr

Abstract. Traditionally, FCA can be used as a tool for eliciting from data a class schema in the form of either a set of attribute implications or a concept lattice. However, such a schema does not necessarily fit the point of view of a domain expert for different reasons, e.g. noise, errors or exceptions in the data. For example, in the domain of animals, an expert may expect that the rule "mammal implies do not lay eggs" holds, while this may not be the case if the platypus is among the objects in the formal context. In this paper, we propose to bridge the possible gap between the representation model based on a concept lattice and the representation model of a domain expert. The knowledge of the expert is encoded as a set of attribute dependencies or constraints which is "aligned" with the set of implications provided by the concept lattice, leading to modifications in the original concept lattice. The method can be generalized for generating lattices satisfying constraints based on attribute dependencies and using extensional projections. This method also allows the experts to keep a trace of the changes occurring in the original lattice and the revised version, and to assess how concepts in practice are related to concepts automatically issued from data.

Keywords: Formal Concept Analysis, projection, attribute implication, attribute dependency

1 Introduction

Formal Concept Analysis (FCA) [1] is a classification method which is helpful for the conceptualisation step in building ontologies [2]. FCA elicits from data a class schema in the form of either a set of attributes implications or a concept lattice. The concept lattice can be interpreted as a knowledge model in the form of a concept hierarchy and the logical structures of formal concepts and concept lattices are effective in supporting human reasoning [3]. However, building knowledge bases is a cognitive process and does not obey to strict and formal rules as domain experts may understand the domain in a different way than what is represented in data. Thus, often there exists a gap between the representation

© Marianne Huchard, Sergei O. Kuznetsov (Eds.): CLA 2016, pp. 349–360,
ISBN 978-5-600-01454-1, National Research University Higher School of Economics, 2016.

model based on a concept lattice and the representation model as imagined by a domain expert. In order to bridge this gap, researchers [4–6] have tried to integrate into lattices experts' knowledge in the form of dependencies between attributes. In these approaches, attribute dependencies serve as constraints that lead to more comprehensible structures of formal concepts for domain experts: formal concepts which satisfy the constraints are then provided to experts, and formal concepts which do not satisfy the constraints are disregarded.

Accordingly, in this paper, we introduce a formal method for integrating expert constraints into concept lattices in such a way that we can maintain the lattice structure as this structure is effective in supporting human reasoning [3]. Moreover, instead of providing only concepts that satisfy experts' knowledge, the method allows experts to keep a trace of changes occurring in the original lattice and the final constrained version, and to assess how concepts in practice are related to concepts automatically issued from data.

In this work, we "align" a set of given attribute dependencies with the set of implications provided by the concept lattice, leading to modifications in the original lattice. The method extends the definition of dependencies between single attributes introduced in [5] to the case of dependencies between attribute sets, and allows domain experts to have more possibilities for expressing constraints. We are able to build the constrained lattices without changing data and provide the trace of changes by using extensional projections [7, 8] over lattices. From an original lattice, two different projections produce two different constrained lattices, and thus, the disagreement between the representation model based on a concept lattice and the representation model as imagined by a domain expert is filled with projections.

The paper is organized as follows. Firstly, we introduce some basic notions that provide the foundations of our work. Next, we detail the method for generating constrained lattices, providing the trace of changes by using projections. Finally, we conclude our work and draw some perspectives over the approach.

2 Preliminaries

In the following, we introduce some important definitions that support our work. Firstly, we introduce attribute implications and dependencies and secondly, we describe projections in a concept lattice. For the following definitions, we use the notation of FCA in [1], where the formal context (G, M, I) is composed of a set of objects G, a set of attributes M and an incidence relation set $I \subseteq G \times M$. An example of such formal context is presented in Table 1.

2.1 Attribute Implication and Dependencies

Implications in a formal context represent dependency relations between attributes existing in data. In a nutshell, given an implication $x \rightarrow y$ we can understand that *"all objects having x also have y"*. Attribute implications can be read off directly from a formal context as stated in Definition 1.

Definition 1 (Attribute Implication [1]) *An implication between sets of attributes $X, Y \subseteq M$ in a formal context (G, M, I) is denoted by $X \rightarrow Y$, where every object having all the attributes from X has also all the attributes from Y, i.e. $X' \subseteq Y'$.*

Following Definition 1, attribute implications can be verified in the formal context and in the concept lattice thanks to Propositions 1 and 2.

Proposition 1 ([1]). *An implication $X \rightarrow Y$ between sets of attributes $X, Y \subseteq M$ holds in (G, M, I) iff $Y \subseteq X''$. It then automatically holds in the set of all concept intents in the concept lattice as well.*

Proposition 2 ([1]). *An implication $X \rightarrow Y$ between sets of attributes $X, Y \subseteq M$ holds in a concept lattice iff $X \rightarrow m$, $\forall m \in Y$, where $X \rightarrow m \iff (X', X'') \leq (m', m'')$. (m', m'') is the attribute concept of m.*

Animal	has_two_legs (m1)	lays_eggs (m2)	can_fly (m3)	has_wings (m4)	has_fins (m5)	has_feathers (m6)	has_milk (m7)	has_backbone (m8)	lives_in_ water (m9)
bear (g1)	x						x	x	
carp (g2)		x			x			x	x
chicken (g3)	x	x	x	x		x		x	
crab (g4)		x							x
dolphin (g5)			x		x		x	x	x
honeybee (g6)		x	x	x					
penguin (g7)	x	x		x		x		x	x
wallaby (g8)	x						x	x	

Table 1: Formal context of animals

For example, the implication `m7:has_milk` $\rightarrow$ `m8:has_backbone` holds in the formal context of Table 1. Different from implications, attribute dependencies do not arise from data. They are dependency relations that experts *expect* to exist as attribute implications. In the following, we provide the definition of attribute dependencies based on [5] and [6].

Definition 2 (Attribute Dependency [6]) *An attribute dependency is in the form $x \prec y$, where attribute x is less important than attribute y and the presence of x is not meaningful without the presence of y.*

Definition 3 (Formal Concept Satisfaction & Constrained Posets [5]) *A formal concept (A, B) satisfies an attribute dependency $x \prec y$ between attributes x and y iff whenever $x \in B$ then $y \in B$.*

(1) A concept lattice $\mathcal{L}$ constrained by an attribute dependency $x \prec y$, is the collection of all formal concepts from lattice $\mathcal{L}$ which satisfy $x \prec y$.
(2) A concept lattice $\mathcal{L}$ constrained by a set of attribute dependencies D, is the collection of all formal concepts from lattice $\mathcal{L}$ which satisfy all attribute dependencies in D.

Notice that both collections are partially ordered subsets of the original lattice $\mathcal{L}$ [5]. We will refer to them as *constrained posets of lattice $\mathcal{L}$ w.r.t. dependency $x \prec y$ (or w.r.t. the set of dependencies D).* To illustrate Definitions 2 and 3 above, consider the formal context of animals in Table 1. The corresponding concept lattice is shown in Fig. 1. The constrained poset w.r.t. attribute dependency `can_fly` $\prec$ `has_wings` is the collection of formal concepts from the lattice circled in blue in Fig. 1.

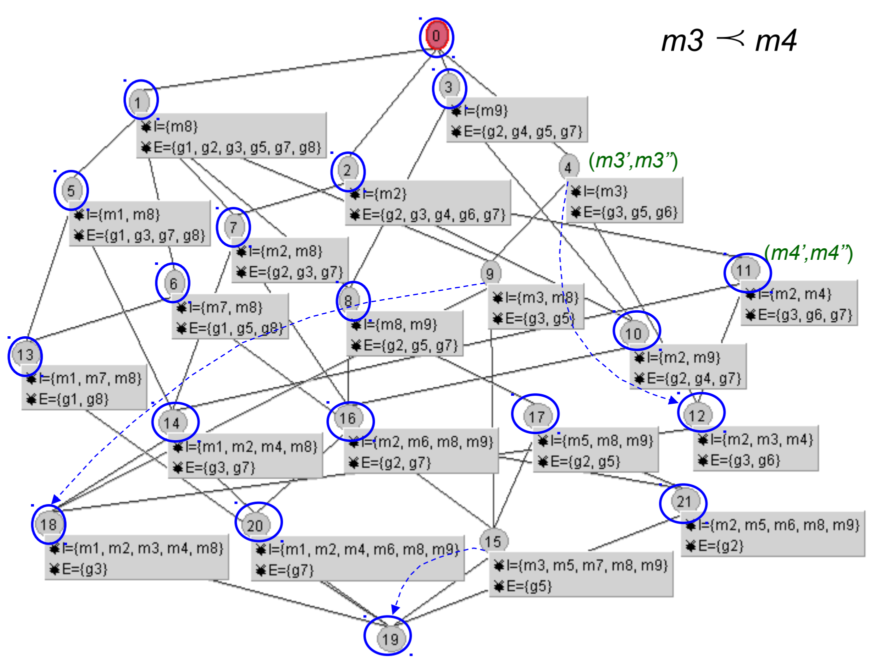


Fig. 1: The lattice constrained by `m3` $\prec$ `m4` and the trace of changes.

2.2 Projections

Projections are mathematical functions that allow mapping extents or intents in a concept lattice into "simpler" extents or intents, called *extensional* or *intensional* projections respectively [8, 9]. Initially, projections are used for simplifying descriptions of concepts in pattern structures [7, 10]. For our purpose, we use extensional projections.

Definition 4 (Extensional Projection [7]) *An extensional projection ψ is defined as a mapping verifying the following properties for each pair (A_1, A_2) of extents of a lattice $\mathcal{L}$: (1) if $A_1 \subseteq A_2$, then $\psi(A_1) \subseteq \psi(A_2)$ (monotone), (2) $\psi(A_1) \subseteq A_1$ (contractive), and (3) $\psi(\psi(A_1)) = \psi(A_1)$ (idempotent).*

Let $\mathcal{L}$ be a lattice, and ψ be an extensional projection of $\mathcal{L}$, then the set of extents E in $\mathcal{L}$ can be divided into two sets: $E = \{e \in E|\psi(e) = e\} \cup \{e \in E|\psi(e) \subset e\}$. The set $\{e \in E|\psi(e) = e\}$ is called the *fixed point* of ψ.

The mapping of a concept in a lattice onto the corresponding concept in the projected lattice can be computed thanks to Proposition 3.

Proposition 3 ([9]). *Let $\mathcal{L}$ be a lattice, and ψ be an extensional projection of $\mathcal{L}$, then a concept (A, B) in $\mathcal{L}$ is projected in the corresponding lattice $\psi(\mathcal{L})$ onto the concept (A_1, B_1) such that $A_1 = \psi(A)$ and $B_1 = A_1'$.*

It is worth noticing that the "projected lattice" which is the set of all projected concepts (or extents) is actually a concept lattice. For our purposes, this adds the benefit that the result of constraining a lattice through a projection preserves the lattice structure. Hereafter, we will refer to the result of constraining the lattice through a projection as a *constrained lattice*.

3 Projections for Generating Constrained Lattices

As previously discussed, in a given formal context, there exists some implications that represent attribute dependencies among attributes. However, implications which can be checked within the concept lattice are usually not in the data. For example, some objects may have missing attribute associations or may have wrongly assigned attributes. In a different scenario, an expert may simply want to observe formal concepts aligned through her particular point-of-view of the domain. Thus, often there exists a *disagreement* between the relations of attributes in the data and the relations of attributes as a domain expert understands them.

3.1 Discussion about Constrained Lattices

In order to bridge the disagreement between the representation model based on a concept lattice and the representation model as imagined by a domain expert, we "align" a set of given attribute dependencies with the set of implications provided by the concept lattice. According to Definitions 1 and 2, if an implication $x \rightarrow y$

holds in a lattice, then that lattice satisfies the attribute dependency $x \prec y$. To build a lattice satisfying a set of attribute dependencies, we look for a lattice that satisfies the corresponding set of attribute implications. In our setting, we want to use a well-founded process based on projections. Thus, we provide a method for projecting the lattice in such a way that the lattice structure is preserved. Moreover, we provide experts with a mapping of concepts in the original lattice onto the corresponding concepts in the revised version to make visible the changes in the lattice. We refer to these mappings as the *trace of changes* occurring in the original lattice and the revised version. We achieve this by using extensional projections over lattices.

We illustrate this scenario where a lattice $\mathcal{L}$ is mapped onto a lattice $\mathcal{L}_1$ which is a revised version w.r.t. the attribute dependency $x \prec y$. Let us call this mapping ς.

We observe the following characteristics of ς: (i) ς reduces the size of the lattice $\mathcal{L}$ because the constrained lattice $\mathcal{L}_1$ do not contain formal concepts in $\mathcal{L}$ which do not satisfy the constraints; (ii) In order to get formal concepts in $\mathcal{L}$ satisfying the constraints, ς replaces the concept extents in that lattice with smaller sets of objects which are still extents. This replacement may result in a loss of information. Hence, the trace of changes should be kept as it may be useful for domain experts to be aware that some concepts in the lattice will be lost some important objects. Indeed, (i) and (ii) are consequences of the fact that ς is an extensional projection. ς is a special case of projections from [7, 8]. This extensional projection does not create new extents, it replaces the concept extents in the lattice with smaller extents.

In the following, we describe how extensional projection is defined in two different cases, namely for a single attribute dependency and for a set of attribute dependencies.

3.2 Projections for Constrained Lattices w.r.t. Dependencies between Attribute Sets

Let us consider the problem of finding an extensional projection $\psi : \mathcal{L} \rightarrow \mathcal{L}_1$, where $\mathcal{L}$ is a concept lattice which does not satisfy the implication $x \rightarrow y$ between attributes $x, y \in M$, $\mathcal{L}_1$ is the projected lattice of $\mathcal{L}$ which satisfies the implication $x \rightarrow y$.

The following propositions state the main properties of ψ.

Proposition 4. *Let $\mathcal{L}$ be a concept lattice which does not satisfy the implication $x \rightarrow y$, then $x' \not\subseteq y' \Longrightarrow x' \cap y' \subset x'$.*

Proposition 5. *Let $\mathcal{L}$ be a concept lattice which does not satisfy the implication $x \rightarrow y$, and ψ be an extensional projection of $\mathcal{L}$ such that the projected lattice satisfies $x \rightarrow y$, then $\psi(x') = x' \cap y'$.*

Proof. see Appendix

Proposition 5 gives us an important property of ψ to observe changes in lattice $\mathcal{L}$: From lattice $\mathcal{L}$ which does not satisfy the implication $x \rightarrow y$, we are going to project x' in lattice $\mathcal{L}$ to $x' \cap y'$ to get the projected lattice that satisfies the implication $x \rightarrow y$. And this follows the properties of projections as $\psi(x') = x' \cap y' \subset x'$.

Given an extensional projection ψ of $\mathcal{L}$ such that $\psi(x') = x' \cap y'$, extents A in $\mathcal{L}$ can be divided into three categories as shown in Fig. 2.

- Category I contains all extents A that are subsumed by $x' \cap y'$, i.e. $A \subseteq (x' \cap y') \subset x'$ ($x' \cap y' \subset x'$ by Proposition 4).
- Category II contains all extents A that are subsumed by x' but not subsumed by $x' \cap y'$, i.e. $A \subseteq x'$, $A \nsubseteq (x' \cap y')$.
- Category III contains extents A that are not in categories I, II, i.e. $A \nsubseteq x'$.

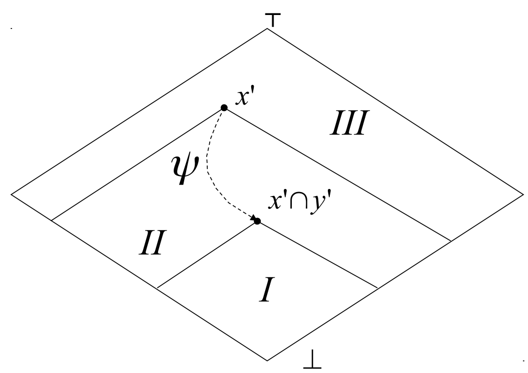


Fig. 2: Three possible categories of extents in lattice $\mathcal{L}$ when $\psi(x') = x' \cap y'$.

Consider an element A in Category I. As the objects in the set $x' \cap y'$ have both attributes x and y, the concept with extent $x' \cap y'$ in $\mathcal{L}$ satisfies the implication $x \rightarrow y$. Because A is subsumed by $x' \cap y'$, the concept with extent A in $\mathcal{L}$ satisfies $x \rightarrow y$ and remains the same in the projected lattice, i.e. $\psi(A) = A$. Category I is a component of the fixed point of the projection ψ.

Consider an element A in Category III. Because the objects in A do not have attribute x, the concept with extent A in $\mathcal{L}$ satisfies the implication $x \rightarrow y$ and remains the same in the projected lattice, i.e. $\psi(A) = A$. Category III is also a component of the fixed point of the projection ψ.

Consider an element A in Category II. We have: 1) $\psi(A) \subseteq A$ by the contractive property of projections; 2) As $A \subseteq x'$, $\psi(A) \subseteq \psi(x')$ by the monotonic property of projections. Moreover, $\psi(x') = x' \cap y'$ by Proposition 5. So, $\psi(A) \subseteq x' \cap y'$. As a result of 1) and 2), $\psi(A) \subseteq A \cap (x' \cap y')$.

In order to have the largest fixed point, we set $\psi(A) = A \cap (x' \cap y')$. $\psi(A) = A \cap (x' \cap y')$ complies with the properties of projections and the concept with extent $\psi(A)$ satisfies $x \to y$ because objects in $\psi(A) = A \cap (x' \cap y')$ have both attributes x and y. This introduces the fact that Category II constrains concepts which are projected in such a way that $\psi(A) = A \cap (x' \cap y')$.

Thus, the projection with the largest fixed point given by $\psi(x') = x' \cap y'$ is:

$$\psi(A) = \begin{cases} A \cap (x' \cap y') & \text{if } A \subseteq x', A \nsubseteq (x' \cap y'), \\ A & \text{otherwise.} \end{cases} \tag{1}$$

This projection gives the projected lattice satisfying the implication $x \to y$.

This projection can be extended to support dependencies between attribute sets of the form $X \prec Y$, where $X, Y \subseteq M$. In such a case, the projection is defined as:

$$\psi(A) = \begin{cases} A \cap (X' \cap Y') & \text{if } A \subseteq X', A \nsubseteq (X' \cap Y'), \\ A & \text{otherwise.} \end{cases} \tag{2}$$

The trace of changes occurring in the original lattice $\mathcal{L}$ and the constrained lattice $\mathcal{L}_1$ can be obtained thanks to Proposition 3.

Example 1. Consider the running example where the expert provides her knowledge in the form of an attribute dependency `m3:can_fly` $\prec$ `m4:has_wings`. According to the data in Table 1: `m3`$'$=\{`g3, g5, g6`\}, `m4`$'$=\{`g3, g6, g7`\}, `m3`$' \cap$ `m4`$'$=\{`g3, g6`\}.

Applying Equation 2, the extensional projection ψ for generating the constrained lattice $\mathcal{L}_1$ from the original lattice $\mathcal{L}$ is:

$$\psi(A) = \begin{cases} A \cap \{\texttt{g3, g6}\} & \text{if } A \subseteq \{\texttt{g3, g5, g6}\}, A \nsubseteq \{\texttt{g3, g6}\}, \\ A & \text{otherwise.} \end{cases} \tag{3}$$

Fig. 1 depicts the constrained lattice and the trace of changes provided by the projection ψ. In Fig. 1, the formal concepts of the constrained lattice are circled blue. The transformations correspond to: $C_4 \to C_{12}$ ({`g3, g5, g6`} $\to$ {`g3, g6`}), $C_9 \to C_{18}$ ({`g3, g5`} $\to$ `g3`), and $C_{15} \to C_{19}$ ($g5 \to \emptyset$). The other transformation {`g5, g6`} $\to g6$ does not apply as neither {`g5, g6`} or $g6$ are extents in lattice $\mathcal{L}$.

An interpretation of the change C_4 in $\mathcal{L}$ to C_{12} in $\mathcal{L}_1$ according to the semantics of the extensional projection is as follows. According to the data, objects `g3:chicken`, `g6:honeybee` are grouped together with object `g5:dolphin` to form concept C_4 whose intent is {`m3:can_fly`}. According to the expert, animals that can fly should also have wings (`can_fly` $\prec$ `has_wings`), `g5:dolphin` should not be grouped together with `g3:chicken` and `g6:honeybee` to form a concept. This is better represented by concept C_{12} whose extent is {`g3:chicken`, `g6:honeybee`} and intent is {`m2:lays_eggs`, `m3:can_fly`, `m4:has_wings`}. By checking this change, we found that the data contain a noisy element: dolphins can fly.

3.3 Projections for Constrained Lattices w.r.t. Sets of Dependencies

A "naive" way to generate a constrained lattice satisfying a set of dependencies consists in generating first a constrained lattice for each dependency, and then getting the final constrained lattice by the intersection of these constrained lattices. A question raised here: is there a "good" way to generate the constrained lattice? In the following, we will show that dependencies should be treated following an order of projections.

Definition 5 ([9, 10]) *Let $\mathcal{L}$ be a concept lattice, and ψ_1, ψ_2 be two extensional projections of $\mathcal{L}$, we say that $\psi_1 \leq \psi_2$, iff there is some projection ψ defined on $\psi_2(\mathcal{L})$ such that for all extent A in $\mathcal{L}$, $\psi_1(A) = \psi \circ \psi_2(A)$. If $\psi_1(\mathcal{L}) \subseteq \psi_2(\mathcal{L})$, then $\psi_1 \leq \psi_2$.*

Definition 5 states that actually projections can be ordered from "general" to "specific". If ψ_1 is a projection over the projected lattice of ψ_2, then we say ψ_1 is more *specific* than ψ_2 or ψ_2 is more *general* than ψ_1.

There is a partial order on projections of a lattice given by Proposition 6. This proposition has been proven in [10].

Proposition 6 ([10]). *Extensional projections of a lattice $\mathcal{L}$ ordered by Definition 5 form a semi-lattice $(\mathcal{F}, \wedge)$, where the semi-lattice operation between $\psi_1, \psi_2 \in \mathcal{F}$ is given by $\psi_1 \wedge \psi_2 = \psi_3$ iff $\psi_3(\mathcal{L}) = \psi_1(\mathcal{L}) \cap \psi_2(\mathcal{L})$.*

The order on projections for generating constrained lattices can be characterized by Proposition 7.

Proposition 7. *Let $\mathcal{L}$ be a concept lattice, ψ_1 be an extensional projection of $\mathcal{L}$ such that the projected lattice satisfies the implication $X_1 \to Y_1$, and ψ_2 be an extensional projection of $\mathcal{L}$ such that the projected lattice satisfies the implication $X_2 \to Y_2$, where $X_1, Y_1, X_2, Y_2 \subseteq M$, we have:*

1) If $X_1 \subseteq X_2$ and $Y_1 \subseteq Y_2$, then $\psi_1 \leq \psi_2$.
2) If $\psi_1 \leq \psi_2$, then the projected lattice given by ψ_1 satisfies $X_1 \to Y_1$ and $X_2 \to Y_2$.
3) There exists an extensional projection ψ_3 of $\mathcal{L}$ such that $\psi_3(\mathcal{L}) = \psi_1(\mathcal{L}) \cap \psi_2(\mathcal{L})$. ψ_3 gives the constrained lattice that satisfies $X_1 \to Y_1$ and $X_2 \to Y_2$.

As a result of Proposition 7, given two dependencies between attribute sets, we can order the projections corresponding to these dependencies as follows. First, if one dependency depends on attribute sets that are included in the attribute sets of the other dependency, then the projection of that dependency is more specific than the projection of the other. Second, if one projection is more specific than the other, then we can use the more specific projection for generating the final constrained lattice . Third, we can use the projection that is the meet of the two projections for generating the final constrained lattice.

Let us now go back to our scenario of generating a constrained lattice that satisfies a set of dependencies. Let $\mathcal{L}$ be a concept lattice, and ψ_i be an extensional projection of $\mathcal{L}$ such that the projected lattice satisfies an implication

$X_i \rightarrow Y_i$, where $X_i, Y_i \subseteq M$. The set of projections ψ_i can be ordered according to the order of projections given by Proposition 7. This order forms a semi-lattice given by Proposition 6. By Proposition 7, the projection that is the meet of the most specific projections in this order gives the final constrained lattice satisfying the set of implications.

According to the order of projections, we have two possible ways of generating constrained lattices. The first way uses the meet of the most specific projections in the order of the set of projections to generate the final constrained lattice. The second way uses all the projections to generate a set of constrained lattices. Applying the first or the second way to generate constrained lattices depends on what experts need. The first way using the meet of the most specific projections to generate the final constrained lattice is more efficient in computation than the second way using all the projections to generate the corresponding constrained lattices. However, by using the meet of the most specific projections to generate the final constrained lattice, the first way only provides the trace of changes between the original lattice and the final constrained lattice. In the case experts need all the traces of changes, we need the second way using all the projections to generate the corresponding constrained lattices.

Example 2. Consider the running example where the expert provides her knowledge in the form of a set of dependencies d_i:

d_1) {m3, m8} $\prec$ {m4},
d_2) m3 $\prec$ m4,
d_3) {m1, m2} $\prec$ {m3},
d_4) m4 $\prec$ m3.

Let ψ_i be the extensional projection for generating the constrained lattice satisfying dependency d_i, applying Equation 2, we have:

$$\texttt{For } d_1 : \psi_1(A) = \begin{cases} A \cap \{\texttt{g3}\} & \text{if } A \subseteq \{\texttt{g3, g5}\}, A \not\subseteq \{\texttt{g3}\}, \\ A & \text{otherwise.} \end{cases} \quad (4)$$

$$\texttt{For } d_2 : \psi_2(A) = \begin{cases} A \cap \{\texttt{g3, g6}\} & \text{if } A \subseteq \{\texttt{g3, g5, g6}\}, A \not\subseteq \{\texttt{g3, g6}\}, \\ A & \text{otherwise.} \end{cases} \quad (5)$$

$$\texttt{For } d_3 : \psi_3(A) = \begin{cases} A \cap \{\texttt{g3}\} & \text{if } A \subseteq \{\texttt{g3, g7}\}, A \not\subseteq \{\texttt{g3}\}, \\ A & \text{otherwise.} \end{cases} \quad (6)$$

$$\texttt{For } d_4 : \psi_4(A) = \begin{cases} A \cap \{\texttt{g3, g6}\} & \text{if } A \subseteq \{\texttt{g3, g6, g7}\}, A \not\subseteq \{\texttt{g3, g6}\}, \\ A & \text{otherwise.} \end{cases} \quad (7)$$

Lattices in Figs. 3, 4, 5 and 6 depict the constrained lattices and the traces of changes provided by the projections ψ_1, ψ_2, ψ_3, ψ_4 respectively. In this set of projections, $\psi_2 \leq \psi_1$ and $\psi_4 \leq \psi_3$. We can see that the constrained lattice $\psi_2(\mathcal{L})$ is included in $\psi_1(\mathcal{L})$ and $\psi_4(\mathcal{L})$ is included in $\psi_3(\mathcal{L})$.

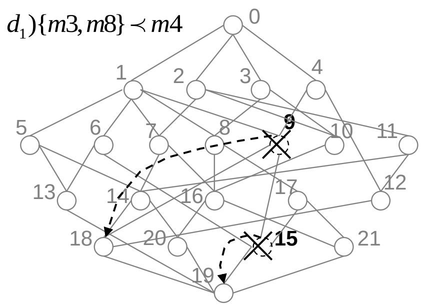


Fig. 3: The constrained lattice for d_1.

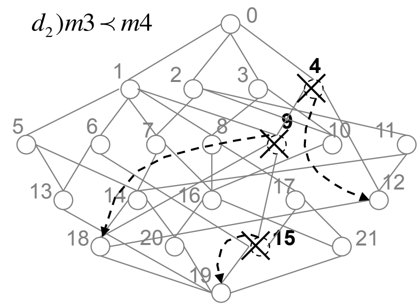


Fig. 4: The constrained lattice for d_2.

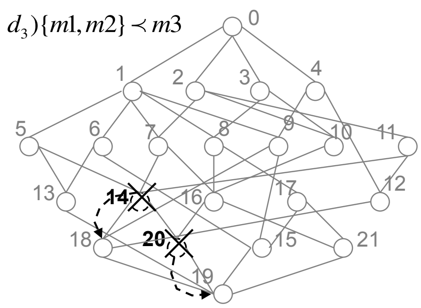


Fig. 5: The constrained lattice for d_3.

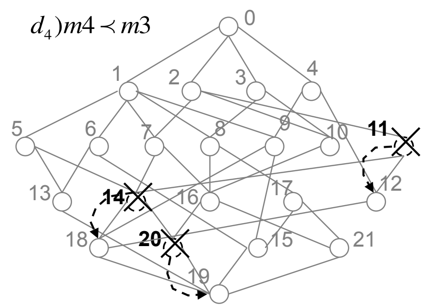


Fig. 6: The constrained lattice for d_4.

Let ψ_5 be an extensional projection that is the meet of the most specific projections: $\psi_5 = \psi_2 \wedge \psi_4$. ψ_5 can be defined such that:

$$\psi_5(A) = \begin{cases} A \cap \{\texttt{g3, g6}\} & \text{if } A \subseteq \{\texttt{g3, g6, g7}\}, A \not\subseteq \{\texttt{g3, g6}\}, \\ A \cap \{\texttt{g3, g6}\} & \text{if } A \subseteq \{\texttt{g3, g5, g6}\}, A \not\subseteq \{\texttt{g3, g6}\}, \\ A & \text{otherwise.} \end{cases} \quad (8)$$

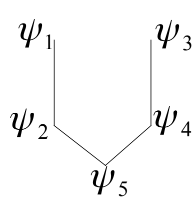


Fig. 7: The semi-lattice of projections.

The set of projections ψ_i forms a semi-lattice as shown in Fig. 7. We have two ways of generating the final constrained lattice. The first way uses the meet ψ_5 of the most specific projections. Lattice in Fig. 8 depicts the final constrained lattice and the trace of changes between the original lattice and the final constrained lattice provided by the projection ψ_5. The second way uses all the projections to have all the traces of changes. According to the semi-lattice of the projections, because $\psi_2 \leq \psi_1$ and $\psi_4 \leq \psi_3$, in order to get all the traces of changes, ψ_1 is applied before ψ_2 and ψ_3 is applied before ψ_4. By contrast, as ψ_2 and ψ_4 are incompatible, it does not matter if ψ_1 and ψ_2 or ψ_3 and ψ_4 are applied first. Thus, the projections can be applied according to either the order ψ_1,ψ_2,ψ_3,ψ_4,ψ_5 or ψ_3,ψ_4.ψ_1,ψ_2,ψ_5. The trace of changes can be either the chain of lattices in Figs. 3, 4, 5, 6, 8 or lattices in Figs. 5, 6, 3, 4, 8. In both cases, experts still can access the changes corresponding to each dependency.

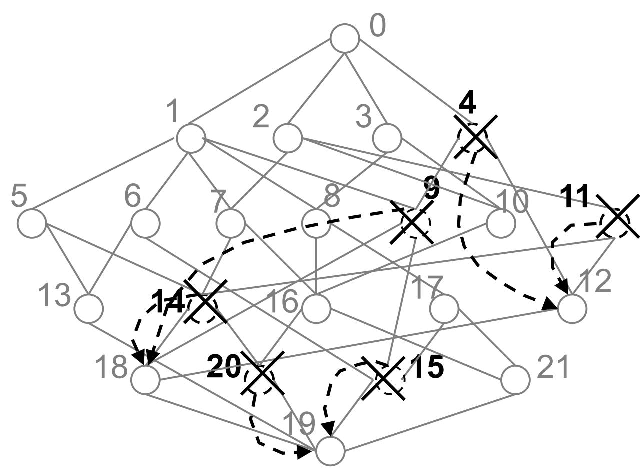


Fig. 8: The final constrained lattice and the trace of changes.

4 Related Work, Discussion and Conclusion

Taking into account expert knowledge in the form of dependencies between attributes in concept lattices has been proposed by several researchers [11, 12, 4, 5]. [11, 12] extended attribute exploration to include background implications, i.e. the implications that experts already know to be valid. In these approaches, they trust the original data and experts have to provide new objects as counterexamples. To deal with a situation such that experts know dependencies between attributes, but they do not know any new objects to provide, the other approaches

[4, 5] were proposed to build a concept hierarchy from a formal context extracted from data and from a hierarchy of attributes provided by domain experts. Two possible ways of adding knowledge to object descriptions were discussed in [4]. One way is to expand the formal context by adding to the description of each object all attributes that are implied by the original attributes. Expanding the context can lead to lose the original data and the formal context may become very large. The other way is to work with an unexpanded formal context by adapting the construction algorithms of lattices to extract formal concepts satisfying dependencies. A similar idea was proposed by [5], in which the authors adapted an incremental algorithm for computing constrained posets. By contrast, we do not expand the formal context nor adapt the construction algorithms of lattices. Our method uses projections to generate constrained lattices instead of constrained posets and to provide the trace of changes.

To conclude, in this paper, we have presented a formal method based on extensional projections for integrating expert knowledge into concept lattices in such a way that the lattice structure and the trace of changes are preserved. The expert knowledge is encoded as a set of attribute dependencies which is aligned with the set of implications provided by the concept lattice. According to the order of projections, the method offers two ways of generating constrained lattices. The first way uses the meet of the most specific projections to generate the final constrained lattice. The second way uses all the projections to generate a set of constrained lattices. The first way is more efficient in computation, but provides only the trace of changes between the original lattice and the final constrained lattice while the second way provides all the traces of changes, but less efficient in computation.

Currently, we are implementing the method for experiments with large datasets. Future work includes defining intensional projections to integrate dependencies between object sets. This is useful for many applications, e.g. when classifying documents, this can be applied to integrate a partial order of documents from experts into concept lattices. Another interesting application could be to complete definitions in data, e.g. the method presented in this paper can be applied to add implications that experts expect to exist.

Appendix

Proof (Proposition 5).

1) As the projected lattice satisfies the implication $x \to y$, $(\psi(y'), \psi(y')') \geq (\psi(x'), \psi(x')')$ (by Proposition 2). So, $\psi(x') \subseteq \psi(y')$.
2) $\psi(y') \subseteq y'$ (by the contractive property of projections).
3) As a result of 1) and 2), we have $\psi(x') \subseteq \psi(y') \subseteq y'$.
4) $\psi(x') \subseteq x'$ (by the contractive property of projections).
5) As a result of 3) and 4), we have $\psi(x') \subseteq x' \cap y'$.
6) For any $m \in M$, m' is the maximal set of objects having m in $\mathcal{L}$. Since ψ is contractive, this is also true in the projected lattice. Thus, $\psi(x')$ is the maximal set of objects having attribute x in the projected lattice $\mathcal{L}_1$.

7) Because the objects in the set $x' \cap y'$ have both attributes x and y, the concept with extent $x' \cap y'$ in $\mathcal{L}$ satisfies $x \to y$ and remains the same in $\mathcal{L}_1$. So, $x' \cap y'$ exists in $\mathcal{L}_1$ and the objects in this set have attribute x.
8) As a result of 6) and 7), we have $x' \cap y' \subseteq \psi(x')$.

As a result of 5) and 8), $\psi(x') = x' \cap y'$.

Proof (Proposition 7).

1) Let $\psi_1(\mathcal{L}), \psi_2(\mathcal{L})$ be the projected lattices given by ψ_1, ψ_2 respectively. As $\psi_1 \leq \psi_2$, according to Definition 5, $\exists\psi$: $\psi_1(\mathcal{L}) = \psi \circ \psi_2(\mathcal{L})$. Hence, $\psi_1(\mathcal{L}) \subseteq \psi_2(\mathcal{L})$. So, the projected lattice given by ψ_1 satisfies $X_1 \to Y_1$ and $X_2 \to Y_2$.
2) In order to give proof that $\psi_1 \leq \psi_2$, we will show that a projection ψ can be defined on $\psi_2(\mathcal{L})$ such that $\psi_1(\mathcal{L}) = \psi \circ \psi_2(\mathcal{L})$:

$$\psi_1(A) = \begin{cases} \psi_2(A) \cap (X_1' \cap X_2') & \text{if } A \not\subseteq X_2', A \not\subseteq X_1' \cap Y_1', A \subseteq X_1', \\ \psi_2(A) & \text{otherwise.} \end{cases} \tag{9}$$

3) This follows from Proposition 6 that the set of projections $\mathcal{F}$ over $\mathcal{L}$ is a semi-lattice, then the meet $\psi_3 = \psi_1 \wedge \psi_2$ must exist.

References

1. Ganter, B., Wille, R.: Formal Concept Analysis, Mathematical Foundations. Springer (1999)
2. Poelmans, J., Ignatov, D., Kuznetsov, S., Dedene, G.: Formal concept analysis in knowledge processing: A survey on applications. Expert Syst. Appl. (2013) 6538–6560
3. Wille, R.: Why can concept lattices support knowledge discovery in databases? Exp. Theor. Artif. Intell. (2002) 81–92
4. Carpineto, C., Romano, G.: Adding knowledge to concept lattices. In: Concept Data Analysis: Theory and Applications. John Wiley & Sons (2004) 64–81
5. Belohlavek, R., Sklenar, V.: Formal concept analysis constrained by attribute-dependency formulas. In: Formal Concept Analysis. Springer (2005) 176–191
6. Messai, N., Devignes, M.D., Napoli, A., Smail-Tabbone, M.: Extending attribute dependencies for lattice-based querying and navigation. In: ICCS. (2008) 189–202
7. Ganter, B., Kuznetsov, S.: Pattern structures and their projections. In: ICCS. Volume 2120 of LNCS. Springer Berlin Heidelberg (2001) 129–142
8. Pernelle, N., Rousset, M.C., Soldano, H., Ventos, V.: Zoom: a nested galois lattices-based system for conceptual clustering. Exp. Theor. Artif. Intell. (2002) 157–187
9. Soldano, H., Ventos, V.: Abstract concept lattices. In: Formal Concept Analysis. Volume 6628 of LNCS. Springer Berlin Heidelberg (2011) 235–250
10. Buzmakov, A., Kuznetsov, S., Napoli, A.: Revisiting pattern structure projections. In: Formal Concept Analysis. Springer International Publishing (2015) 200–215
11. Stumme, G.: Attribute exploration with background implications and exceptions. In: Data Analysis and Information Systems. Springer (1996) 457–469
12. Ganter, B.: Attribute exploration with background knowledge. Theoretical Computer Science (1999) 215 – 233

Scalable Performance of FCbO Update Algorithm on Museum Data

Tim Wray[1], Jan Outrata[2*], and Peter Eklund[1]

[1] IT University of Copenhagen
[2] Dept. Computer Science, Palacký University Olomouc, Czech Republic

Abstract. Formal Concept Analysis – known as a technique for data analysis and visualisation – can also be applied as a means of creating interaction approaches that allow for knowledge discovery within collections of content. These interaction approaches rely on performant algorithms that can generate conceptual neighbourhoods based on a single formal concept, or incrementally compute and update a set of formal concepts given changes to a formal context. Using case studies based on content from museum collections, this paper describes the scalability limitations of existing interaction approaches and presents an implementation and evaluation of the FCbO update algorithm as a means of updating formal concepts from large and dynamically changing museum datasets.

1 Introduction

Formal Concept Analysis is best known as a technique for data analysis, knowledge representation and visualisation. A number of case studies have been developed that also use FCA as a means of creating and visualising the semantic spaces within museum collections – allowing users to visualise, explore and discover new objects within these collections based on their associations and commonalities with other objects. Some of these applications include *Virtual Museum of the Pacific* [1], the *Brooklyn Museum Canvas* [2] and the *A Place for Art* [3] iPad app. These case studies led to the development of a set of web services called the CollectionWeb framework [4, 5]. Their analysis gave rise to new interactions approaches based on FCA that required the use of fast algorithms for computing the upper and lower neighbours of a formal concept, and for computing and updating a set of formal concepts based on incremental changes to their formal contexts. These approaches are described as the *conceptual neighbourhood* approach and *concept layer* approach, respectively. This paper focuses on the implementation and scalability limitations of the *conceptual neighbourhood* approach, along with the *FCbO update* algorithm, its implementation within the *concept layer* approach and its performance evaluation.

* J. Outrata acknowledges support by the grant No. IGA_PrF_2016_027 of the IGA Palacký University Olomouc.

ISBN 978-5-600-01454-1, National Research University Higher School of Economics, 2016.

The case studies are motivated by emerging museological movements that have occurred since the 1970s that recognise the museum's role in collecting, creating and shaping knowledge in which the *context* of an object has become an increasingly important part of its analysis, interpretation and communication. [6–9]. Context can refer to an object's materials, construction, design, ornamentation, provenance, history, environment, connection to people and human society [9, 10]. This focus towards context reflects a shift from a *classical* world-view, where objects were classed in terms of order, hierarchy and taxonomy, to a modern perspective where objects are analysed in terms of links to other objects, people, social and cultural histories [9]. The natural association between these modern perspectives of information and knowledge sharing within museums are in accord with the foundations of Formal Concept Analysis in its ability to augment human thought, communication and interpretation. [11, 12]. This association motivates the research into new design and interaction approaches that emphasise concept generation and discovery within museum collections that rely on fast and efficient algorithms for computing formal concepts and their conceptual neighbours.

2 FCA algorithms: scalability and performance evaluation

2.1 The conceptual neighbourhood approach

In the museum-based case studies reported, FCA is used to provide conceptual structures that can be navigated by a user. The conceptual neighbourhood approach, as reported in [13], offers the ability to view individual concepts and move between neighbouring concepts within a concept lattice. One implementation of this approach is to compute and store a complete concept lattice that can then be traversed by the user. However as is well known, complete concept lattices – while adequate for visualising small datasets – are computationally prohibitive and visually complex on medium to larger datasets typically associated with museum collections that typically contain tens of thousands of objects [14].

The time and space complexities of pre-computing and storing a complete concept lattice can be understood by a discussion of how the approach scales with respect to the size of a formal context. Following an analysis of algorithms that build complete concept lattices, Carpineto and Romano [14] identify their time complexities: the best result being the ConceptsCover algorithm which has a worst-case time complexity of $O(|C||M|(|G| + |M|))$ which is dependent, in part, on the number of formal concepts generated from a formal context. The number of formal concepts $|C|$ generated from a formal context $K := \langle G, M, I \rangle$, can be linear (in the best case) or quadratic (in the worst case) with respect to $|G|$ (the number of objects) or $|M|$ (the number of attributes) within the formal context depending on the number of attributes per object. However, even withstanding the time and space complexities for initially computing and storing concept lattices from a large formal context (which, if the system employed update algorithms to update the concept lattice, would only need to be run once), the worst-case time complexity for updating a pre-computed concept lattice –

i.e., only computing a portion of a concept lattice given changes to a formal context – is quadratic with respect to the number of formal concepts $|C|$; although experimental results [15, 16] (cited in [14]) suggest that in practice, the growth may be linear, rather than quadratic. Despite this, updating and storing a complete concept lattice for conceptual navigation poses major scalability and space concerns for large formal contexts.

CollectionWeb implements an alternate approach that does not require computation of the complete concept lattice and therefore negates the above scalability issues, but still allows the user to navigate between neighbouring formal concepts – via the reduction and inclusion of query attributes. This method, called the *conceptual neighbourhood approach*, was used in *ImageSleuth* [13, 12] and again in the *Virtual Museum of the Pacific* [1]. In both cases interaction follows a partial view of the concept lattices in the form of a single formal concept and its immediate neighbours.

The algorithm used by CollectionWeb for generating conceptual neighbourhoods is the NearestNeighbours algorithm [14], presented in Algorithm 1. The conceptual neighbourhood of a formal concept can be formed by finding both the upper and lower neighbours of a formal concept which can be computed separately. In the description of the algorithm that follows, a formal context is denoted by the triplet $\langle G, M, I\rangle$ with the finite non-empty sets of objects $G = \{0, 1, \ldots, g\}$ and attributes $M = \{0, 1, \ldots, m\}$ and $I \subseteq G \times M$ being an incidence relation with $\langle g, m\rangle \in I$, meaning that object $g \in G$ has attribute $m \in M$. Concept-forming operators defined on I are denoted by $' : 2^G \mapsto 2^M$ and $' : 2^M \mapsto 2^G$ [17].

The worst-case time complexity of Algorithm 1 is $O(|G||M|(|G| + |M|))$, the sum of the time to find its lower neighbours, $O(|G||M|^2)$, and the time to find its upper neighbours, $O(|G|^2|M|)$. Hence, the maximum running time of the algorithm is quadratic with respect to the number of objects or the number of attributes within the formal context – whichever is larger. As implemented in CollectionWeb, the NearestNeighbours algorithm runs dynamically at query time – i.e., everytime a user views a formal concept or moves to an upper or lower neighbour, the new concept and its neighbouring concepts are computed. For *ImageSleuth* [13, 12] and *Virtual Museum of the Pacific* case studies [1] this means that any changes to the underlying formal context – new attributes or objects added or removed from the collection – are immediately reflected in its underlying concept lattice, allowing the collection and the relationships among the objects to dynamically respond to user tagging and curatorial management.

However, the advantage offered by dynamically computing the conceptual neighbourhood – namely in that it negates the need to compute or store a potentially large concept lattice while still offering the ability to dynamically expose sections of it for user interaction – also presents another scalability limitation as the size of the collection grows. Given the dynamic nature of the query and the quadratic time complexity with respect to the number of objects in a collection, the *conceptual neighbourhood* approach becomes less suited for use in larger collections, as the response time for user interaction (in the worst case

Algorithm 1: The NearestNeighbours algorithm used for generating a conceptual neighbourhood for formal concept $\langle X, Y\rangle$ in formal context $\langle G, M, I\rangle$, cf. [14]

Input: Formal concept $\langle X, Y\rangle$ of formal context $\langle G, M, I\rangle$
Output: The set of lower and upper neighbours of $\langle X, Y\rangle$ in the concept lattice of $\langle G, M, I\rangle$

```
// Returns the lower neighbours of ⟨X,Y⟩
lowerNeighbours := ∅;
lNCandidates := ∅;
foreach m ∈ M \ Y do
    X1 := X ∩ {m}';
    Y1 := X1';
    if ⟨X1, Y1⟩ ∉ lNCandidates then
        Add ⟨X1, Y1⟩ to lNCandidates;
        count(⟨X1, Y1⟩) := 1;
    else
        count(⟨X1, Y1⟩) := count(⟨X1, Y1⟩) + 1;
    if (|Y1| − |Y|) = count(⟨X1, Y1⟩) then
        Add ⟨X1, Y1⟩ to lowerNeighbours;

// Returns the upper neighbours of ⟨X,Y⟩
upperNeighbours := ∅;
uNCandidates := ∅;
foreach g ∈ G \ X do
    Y2 := Y ∩ {g}';
    X2 := Y2';
    if ⟨X2, Y2⟩ ∉ uNCandidates then
        Add ⟨X2, Y2⟩ to uNCandidates;
        count(⟨X2, Y2⟩) := 1;
    else
        count(⟨X2, Y2⟩) := count(⟨X2, Y2⟩) + 1;
    if (|X2| − |X|) = count(⟨X2, Y2⟩) then
        Add ⟨X2, Y2⟩ to upperNeighbours;
```

scenario) grows quadratically with respect to the number of objects in the collection. While the approach is well suited for dynamically presenting relatively smaller-sized collections at a specialist or 'exhibition' sized scale, such as the 427 objects present in the *Virtual Museum of the Pacific* or the 80 objects present in *A Place for Art*, the approach remains unsuited for larger collections, such as the the *Brooklyn Museum Canvas* case study with many thousands of objects.

2.2 The concept layer approach

For all other case studies, CollectionWeb constructs and maintains a set of formal concepts from a formal context of collection objects. The set of all formal concepts for the formal context in CollectionWeb is called the *concept layer*. The framework relies on a concept layer in order to efficiently create the required data visualisations and semantic structures so that users can associatively browse, visualise and navigate the the collection.

To create and maintain the concept layer, CollectionWeb relies on an algorithm with a low running time for computing formal concepts from a formal context, and for recomputing formal concepts if any objects or attributes in the formal context changes. Specifically, the algorithm should accommodate changes to a formal context in large museum datasets if a single object (or a relatively small batch of objects) changes, ensuring that it can dynamically update the concept layer for a large museum dataset in real time.

There are many high performance algorithms that compute formal concepts from formal contexts [18–22], along with a recent evaluation study of those algorithms applied to data from the Web [23]. As these algorithms offer high performance batch computation of an entire set of formal concepts from a formal context, they work well for large museum collections that do not change over time. However, this is not a common use case: as part of their curatorial practices, museums continually add or modify objects in their online collections, and some require the data to be kept up-to-date as it changes. For instance, the Brooklyn Museum dataset used for the *Brooklyn Museum Canvas* case study [2], along with other large public facing datasets such as the one provided by the Rijksmuseum [3] – also used in this evaluation – require as part of their terms of use, that all front-facing applications or representation of content must be up-to-date. [4] In these cases, such changes from these data sources should be propogated to these front-facing applications as quickly as possible. In addition, large-scale collaborative tagging efforts such as the *steve.museum* project [24] and the Flickr Commons recognise museum collections as dynamic, rather than static datasets. As discussed further in Section 2.3, the ability to quickly recompute a set of formal concepts given incremental updates to its formal context can lead to real-time interaction and visualisation of museum data-sets. Such scenarios call for an efficient FCA algorithm that can accommodate incremental

[3] `https://www.rijksmuseum.nl/`

[4] `http://www.brooklynmuseum.org/opencollection/api/docs/terms`

changes to a formal context, rather than require the recomputation of the entire set of formal concepts when one or a few of its objects changes.

CollectionWeb employs the FCbO algorithm to initially compute all concepts of a formal context [22] (the algorithm is an improved version of Kuznetsov's Close-by-One algorithm [25, 26]) and, more importantly, a modification of that algorithm called *FCbO update* [27] (earlier version also in [28]) to update formal concepts as objects in the formal context are added, modified or deleted. We briefly present *FCbO update* here for the purposes of self-containment. The presentation uses a scenario where new objects are added to the formal context which results in the algorithm producing new and updated formal concepts.

In the description of the algorithm that follows we use the same notation for formal context and concept-forming operators that were used in Algorithm 1. In addition, new objects to be added to $\langle G, M, I\rangle$ and not present in G are denoted by $G_N = \{g+1, \ldots, g_U\}$ (i.e. $G_N \cap G = \emptyset$), $M_N = \{i, \ldots, k\}$ is the set of attributes shared by at least one of the objects G_N and either present or not present in M (but usually $M_N \subseteq M$) and $N \subseteq G_N \times M_N$ is an incidence relation between G_N and M_N. By the triplet $\langle G_U, M_U, I_U\rangle$ we denote the formal context which results as a union of $\langle G, M, I\rangle$ and $\langle G_N, M_N, N\rangle$, both extended to G_U and M_U, i.e. $G_U = G \cup G_N = \{0, \ldots, g_U\}$, $M_U = M \cup M_N = \{0, \ldots, m_U\}$, $m_U = k$ if $k > m$ and $m_U = m$ otherwise, and $I_U \subseteq G_U \times M_U$ such that $I_U \cap (G \times M) = I$, $I_U \cap (G_N \times M_N) = N$ and $I_U \cap (G \times (M_N \setminus M)) = I_U \cap (G_N \times (M \setminus M_N)) = \emptyset$.

The algorithm is represented by the recursive procedure UpdateFastGenerateFrom, presented in Algorithm 2. The procedure is a modified form of the recursive procedure FastGenerateFrom – the core of the FCbO algorithm as described in [22] (Algorithm 2). The procedure accepts as its arguments a formal concept $\langle X, Y\rangle$ of $\langle G_U, M_U, I_U\rangle$ (an initial formal concept), an attribute $m \in M_N$ (first attribute to be processed) and a set $\{N_m \subseteq M_U \mid m \in M_U\}$ of subsets of attributes M_U, and uses a local variable *queue* as a temporary storage for computed formal concepts and M_m ($m \in M_U$) as sets of attributes which are used in place of N_m for further invocations of the procedure. When the procedure is invoked, it recursively descends, in a combined depth-first and breadth-first search, the space of new and updated formal concepts of $\langle G_U, M_U, I_U\rangle$ resulted by adding new objects G_N described by attributes M_N to $\langle G, M, I\rangle$, beginning with $\langle X, Y\rangle$. For a full description of the procedure, see [27] or [28], recalling that the set $M_{U,j} \subseteq M_U$ in Algorithm 2 is defined by: $M_{U,j} = \{m \in M_U \mid m < j\}$. In order to compute all new and updated formal concepts of $\langle G_U, M_U, I_U\rangle$ which are not formal concepts of $\langle G, M, I\rangle$, each of them exactly once, UpdateFastGenerateFrom shall be invoked with $\langle \emptyset', \emptyset''\rangle$, m being the first attribute in M_N and $\{N_m = \emptyset \mid m \in M\}$ as its initial arguments.

The worst-case time complexity of Algorithm 2 remains the same as of the original FCbO (and CbO) algorithm, $O(|C||M|^2|G|)$, because when adding all objects to the empty formal context it actually performs FCbO.

For updating a set of formal concepts given by incremental object-by-object updates of a formal context, there are a number of other incremental algorithms that can be used for determine a set of formal concepts and, subsequently, for

Algorithm 2: The UpdateFastGenerateFrom($\langle X, Y\rangle, m, \{N_m \mid m \in M_U\}$) algorithm used for computing all new and updated formal concepts of formal context $\langle G_U, M_U, I_U\rangle$, cf. [27]

```
Input: Formal concept ⟨X, Y⟩ of formal context ⟨G_U, M_U, I_U⟩, attribute
       m ∈ M_N (or a number ⩾ m_U) and set {N_m ⊆ M_U | m ∈ M_U} of
       subsets of attributes M_U
Output: The set of all new and updated formal concepts of ⟨G_U, M_U, I_U⟩

// output ⟨X, Y⟩, e.g., print it on screen or store it
if (X ∩ G)′ ≠ Y then
    output ⟨X, Y⟩ as new;
else
    if (X ∩ G) ⊂ X then
        output ⟨X, Y⟩ as updated;
    else
        return
if Y = M_U or m > m_U then
    return
for j from m upto m_U do
    set M_j to N_j;
    // go through attributes from M_N only
    if j ∉ Y and j ∈ M_N and N_j ∩ M_{U,j} ⊆ Y ∩ M_{U,j} then
        set X_1 to X ∩ {j}′;
        set Y_1 to X_1′;
        if Y ∩ M_{U,j} = Y_1 ∩ M_{U,j} then
            put ⟨⟨X_1, Y_1⟩, j + 1⟩ to queue;
        else
            set M_j to Y_1;
while get ⟨⟨X_1, Y_1⟩, j⟩ from queue do
    UpdateFastGenerateFrom(⟨X_1, Y_1⟩, j, {M_m | m ∈ M_U});
return
```

computing the concept lattice, such as [16, 29, 30] along with the algorithms in [14]. AddIntent [30] is considered to be one of the most efficient of these algorithms, however, along with the other algorithms, it requires the complete concept lattice prior to computation. The FCbO update algorithm [27] described above, differentiates itself from other incremental algorithms in that it does not require the concept lattice (nor the set of all formal concepts) as its input. However, the number of concepts computed from datasets we use – even without the complexities of storing a complete concept lattice – is of the order hundreds of thousands (see Figures 1 and 2). In light of this, the FCbO update algorithm not only computes changes based only on a set of objects marked for update, but it also outputs only the new and updated formal concepts, rather than the entire set of formal concepts. This allows for quick execution of the algorithm and ingestion of its results where changes to formal context are relatively minor: strengthening the algorithm's utility in applications where datasets are large but updated frequently and in small increments.

2.3 Performance Evaluation

The algorithm was evaluated on two museum datasets: the first being the Brooklyn Museum collection consisting of 10,000 objects and 8,952 attributes and the second being the Rijksmuseum collection consisting of 100,000 objects and 1,716 attributes. The purpose of the performance evaluation was to determine the total running time and performance benefit of using the FCbO update algorithm to incrementally update a set of formal concepts given changes to a formal context, rather than recomputing its entire set of formal concepts.

Table 1. Running time of computing all formal concepts from a formal context using the FCbO update algorithm, average of 10 iterations

Dataset	No. of attributes	No. of objects	No. of concepts	Avg. running time (ms)
Brooklyn Museum	8,952	10,000	98,547	36,218
Rijksmuseum	1,716	100,000	994,967	68,792

Table 1 shows the running time to compute the entire set of formal concepts from the formal contexts generated from the Brooklyn Museum and Rijksmuseum datasets. For the sake of clarity, a *batch* or *non-update* computation – such as the one demonstrated in the table above – is defined as a computation that computes the entire set of formal concepts from formal context, whereas an *update* formal concept computation is defined as a computation that uses a set of objects to add, remove or update within the formal context as its input and outputs a set of changed concepts. The above figures in Table 1 are used as a benchmark in the evaluation of the performance benefit of the *update*, rather than the *batch* computations of the FCbO algorithm.

An *update* computation can be triggered by three different events: adding new objects to the formal context, removing existing objects from the formal context, or updating the attribute sets of existing objects within the formal context. Given that objects can be *added*, *removed* or *updated* within a museum dataset, these three operations are defined and evaluated separately with respect to the running time of the algorithm. Assuming a full set of formal concepts have already been computed, each operation produces a number of *modified concepts* that refer to the set of formal concepts added, removed or updated as a result of each operation. In addition to the time it takes to perform each operation, the number of *modified concepts* serves as an important indicator of complexity.

The results of a performance evaluation demonstrating *add*, *remove* and *update* operations for the FCbO update algorithm are shown in Fig. 1 for the Brooklyn Museum dataset, and Fig. 2 for the Rijksmuseum dataset. The figures demonstrate how the algorithm scales with each operation for adding, removing or updating 1, 5, 50 or 500 objects to their respective datasets. In each figure, the horizontal axis first groups the number of objects N, which is then further sub-divided into its three operations with respect to the formal context: incrementally compute the set of formal concepts when N objects are *added*, *removed* and *updated* from the formal context. As a way of comparing the running time of the FCbO update algorithm to its batch counterpart, the performance metrics of the update algorithm – its running time and number of modified concepts – are shown along with the total running time and number of formal concepts produced by the non-update algorithm, the dashed line in Figures 1 and 2.

For the smaller Brooklyn Museum collection, the number of *modified concepts* and time taken to compute them is reasonable when adding 5 or 50 objects, with running times far less than the time it takes for the algorithm to recompute the entire set of formal concepts. However, in the larger Rijksmuseum collection – due to the smaller number of attributes and higher context density – removing and updating a larger batch of objects requires the re-computation of a large number of formal concepts where in some cases, Figures 1 and 2, the time taken to update the set of formal concepts is greater than the time to recompute the entire set as a batch operation.

The benefits of an incremental FCbO update algorithm with a low running time with respect to museum curation practices and visitor experiences can be realised with respect to user interactions that lead to dynamically changing contexts. For example, in many online collections such as the Powerhouse Museum Online Collection [5] and the Brooklyn Museum Online Collection [6], visitors can add their own interpretations to the objects by adding their own keywords or 'tags'. These interactions can introduce new perspectives on the works [24] that can potentially reframe the way objects are related to one another [31] in that audiences are invited to shape the context, and subsequently, the knowledge that surrounds the objects. Given that formal concepts can be used to represent contextual knowledge of a domain where museum objects are treated as formal

[5] `http://www.powerhousemuseum.com/collection/database/menu.php`
[6] `https://www.brooklynmuseum.org/opencollection/collections/`

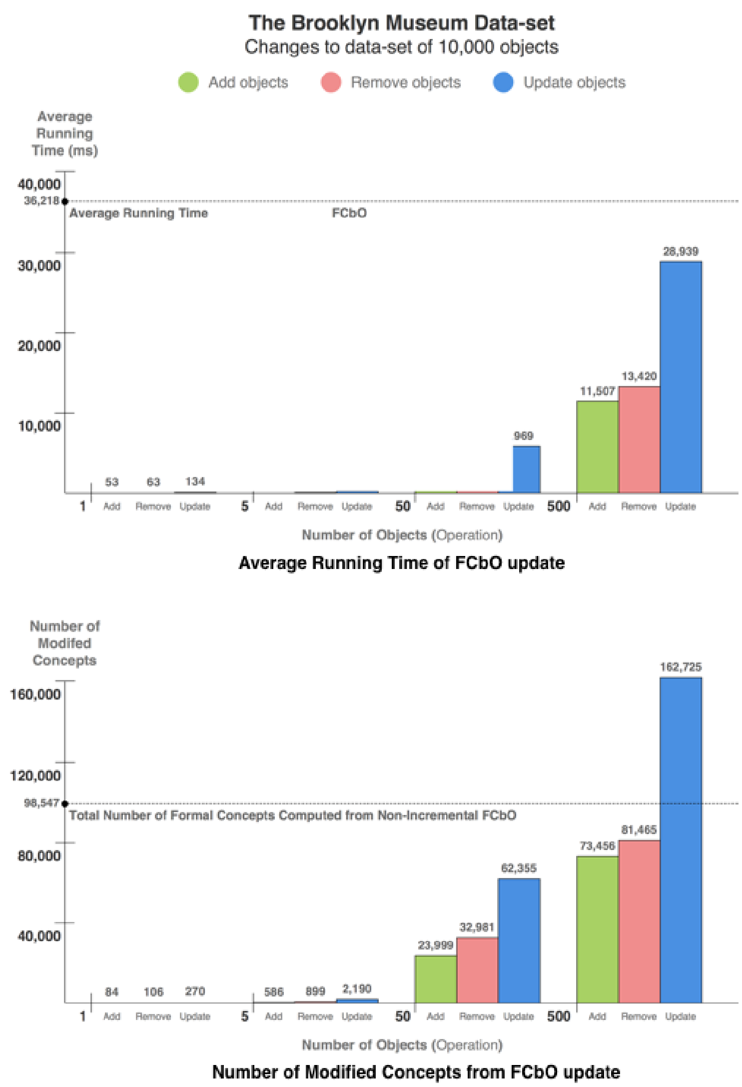


Fig. 1. Average running time and number of modified concepts for adding, removing or updating objects to a formal context and incrementally recomputing the set of formal concepts using the FCbO update algorithm on the Brooklyn Museum dataset. The top graph shows the total running time for each operation for 1, 5, 50 and 500 objects, whereas the bottom graph shows the total number of *modified concepts* for each operation for 1, 5, 50 and 500 objects.

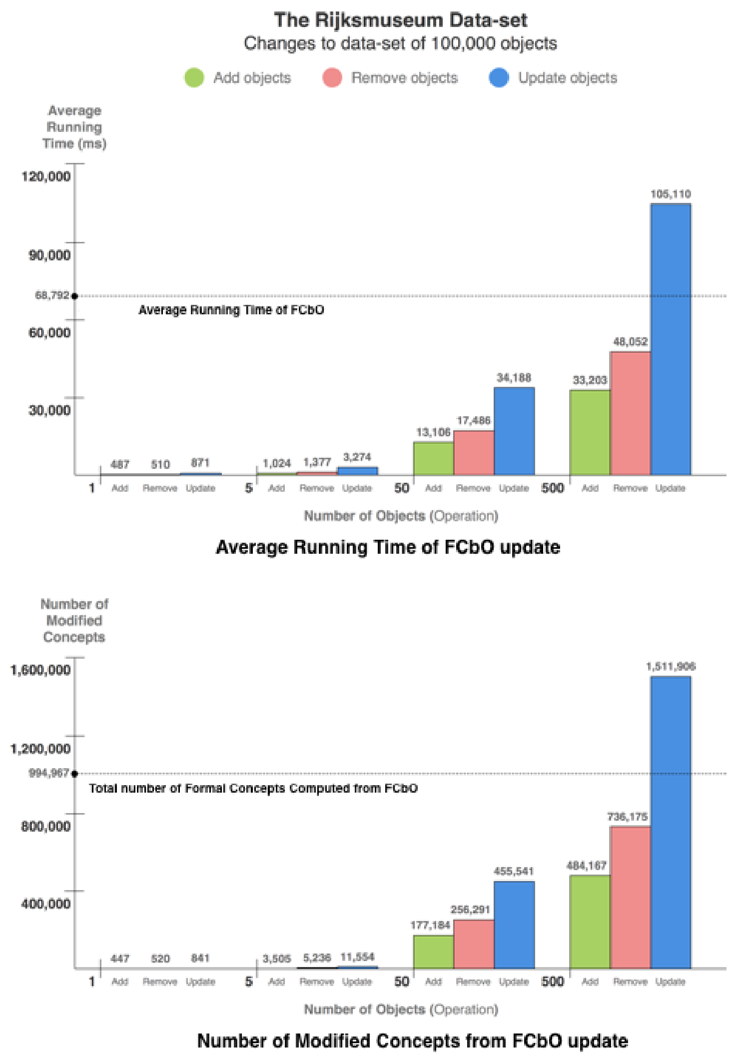


Fig. 2. Average running time and number of modified concepts for adding, removing or updating objects to a formal context and incrementally recomputing the set of formal concepts using the FCbO update algorithm on the Rijksmuseum dataset. The top graph shows the total running time for each operation for 1, 5, 50 and 500 objects, whereas the bottom graph shows the total number of *modified concepts* for each operation for 1, 5, 50 and 500 objects.

objects and tags as formal attributes, user tagging can provide the ability to update representations of knowledge in real-time. Due to the low running time of the FCbO update algorithm on small sets of objects as their input, a user could potentially tag an object and then, through the use of incremental concept computation coupled with data visualisation, immediately realise not only how their tagging enhances the content of the objects, but also shapes the knowledge that surrounds it in relation to other objects.

In many other cases, updates to museum collection data are provided as a batch – i.e., whole groups of objects added or modified as a result of changes to objects within a museum dataset. For example, the Smithsonian Cooper-Hewitt National Design Museum uses GitHub [7] to host their collection data [8] – allowing anyone to access, update and provide updates to the collection. Many other museums provide a timestamp in their object records to indicate when it was last updated, so that data harvesters can collect changes. In other situations it may be more feasible to implement updates to the dataset as a batch rather than as a set of small frequently occurring object updates.

3 Conclusion

Overall, the FCbO update algorithm – as implemented by CollectionWeb to construct and maintain its concept layer – provides a fast way to update formal concepts from large and dynamically changing museum datasets, given that the changes within those datasets are relatively small relative to the size of the formal context. The algorithm provides a scalable way to construct and maintain a concept layer once the initial and potentially time costly computation of the entire set of formal concepts from a formal context is complete. The algorithm is less efficient at adding, removing or updating large changes to the collection where, in such cases, it may be preferential to recompute the entire set of formal concepts.

References

1. Eklund, P., Goodall, P., Wray, T.: Cluster-based Navigation for a Virtual Museum. In: 9th RIAO Conference – Adaptivity, Personalization and Fusion – of Heterogeneous Information, Paris, ACM Press (April 2010)
2. Wray, T., Eklund, P.: Concepts and Collections: A Case Study using Objects from the Brooklyn Museum. In Predoiu, L., Hennicke, S., Nurnberger, A., Mitschick, A., Ross, S., eds.: Proceedings of the 1st International Workshop on Semantic Digital Archives. (2011) 109–120
3. Wray, T., Eklund, P., Kautz, K.: Pathways through Information Landscapes: Alternative Design Criteria for Digital Art Collections. In: ICIS 2013 Proceedings, Milan, Italy (2013)

[7] GitHub is a popular source code management system traditionally used for making available, committing and providing updates to, program source code.

[8] See: `http://www.cooperhewitt.org/collections/data`

4. Eklund, P., Wray, T., Ducrou, J.: Linking Objects and their Stories: An API For Exploring Cultural Heritage Using Formal Concept Analysis. Journal of Emerging Technologies in Web Intelligence **3**(3) (2011) 239–252
5. Eklund, P., Wray, T., Ducrou, J.: Web services and Digital Ecosystem Support using Formal Concept Analysis. In: Proceedings of the International Conference on Management of Emergent Digital EcoSystems. MEDES '09, New York, NY, USA, ACM (2009) 36–245
6. Ross, M.: Interpreting the new museology. Museum and Society **2**(2) (2004) 84–103
7. Styliani, S., Fotis, L., Kostas, K., Petros, P.: Virtual museums, a survey and some issues for consideration. Journal of Cultural Heritage **10**(4) (October 2009) 520–528
8. Skov, M.: The Reinvented Museum: Exploring Information Seeking Behaviour in a Digital Museum Context. PhD thesis, Royal School of Library and Information Science (2009)
9. Hooper-Greenhill, E.: Museums and the Shaping of Knowledge. Routledge (1992)
10. Pearce, S.: Thinking about Things. In Pearce, S., ed.: Interpreting Objects and Collections. Routledge, London (1994)
11. Wille, R.: Formal Concept Analysis as Mathematical Theory of Concepts and Concept Hierarchies. In Ganter, B., Stumme, G., Wille, R., eds.: Formal Concept Analysis. Volume 3626 of Lecture Notes in Computer Science. Springer Berlin / Heidelberg (2005) 47–70
12. Ducrou, J.: Design for conceptual knowledge processing: case studies in applied formal concept analysis. PhD thesis, University of Wollongong (2007)
13. Ducrou, J., Vormbrock, B., Eklund, P.: FCA-based Browsing and Searching of a Collection of Images. In: Proceedings of 14th International Conference on Conceptual Structures. LNAI 4068, Springer (2006) 203–214
14. Carpineto, C., Romano, G.: Concept data analysis: Theory and applications. J. Wiley (2004)
15. Carpineto, C., Romano, G.: A lattice conceptual clustering system and its application to browsing retrieval. Machine Learning **24**(2) (1996) 1–28
16. Godin, R., Missaoui, R., Alaoui, H.: Incremental concept formation algorithms based on Galois (concept) lattices. Computational Intelligence **11**(2) (1995) 246–247
17. Wille, R., Ganter, B.: Formal Concept Analysis: Mathematical Foundations. Springer-Verlag, Berlin (1999)
18. Andrews, S.: In-Close2, a High Performance Formal Concept Miner. In Andrews, S., Polovina, S., Hill, R., Akhgar, B., eds.: Conceptual Structures for Discovering Knowledge. Volume 6828 of Lecture Notes in Computer Science. Springer Berlin Heidelberg (2011) 50–62
19. Krajca, P., Outrata, J., Vychodil, V.: Advances in algorithms based on CbO. In Kryszkiewicz, M., Obiedkov, S., eds.: Proceedings of the 7th International Conference on Concept Lattices and Their Applications, Sevilla, Spain (October 2010) 71–82
20. Krajca, P., Outrata, J., Vychodil, V.: Computing formal concepts by attribute sorting. Fundamenta Informaticae **115**(4) (2012) 395–417
21. Krajca, P., Outrata, J., Vychodil, V.: Parallel algorithm for computing fixpoints of Galois connections. Annals of Mathematics and Artificial Intelligence **59**(2) (2010) 257–272
22. Outrata, J., Vychodil, V.: Fast Algorithm for Computing Fixpoints of Galois Connections Induced by Object-Attribute Relational Data. Information Sciences **185**(1) (2012) 114–127

23. Kirchberg, M., Leonardi, E., Tan, Y.S., Link, S., Ko, R.K.L., Lee, B.S.: Formal Concept Discovery in Semantic Web Data. In: Formal Concept Analysis. Volume 7278 of Lecture Notes in Computer Science. (2012) 164–179
24. Trant, J.: Tagging, Folksonomy and Art Museums: Results of steve.museum's research. Technical report, University of Toronto (2009)
25. Kuznetsov, S.O.: A fast algorithm for computing all intersections of objects in a finite semi-lattice. Nauchno-tekhnicheskaya Informatsiya (1) (1993) 17–20
26. Kuznetsov, S.: Learning of Simple Conceptual Graphs from Positive and Negative Examples. In: PKDD 1999. (1999) 384–391
27. Outrata, J.: A lattice-free concept lattice update algorithm. International Journal of General Systems **45**(2) (2016) 211–231
28. Outrata, J.: A lattice-free concept lattice update algorithm based on ∗CbO. In Ojeda-Aciego, M., Outrata, J., eds.: Proceedings of the 10th International Conference on Concept Lattices and their Applications, La Rochelle, France (2013) 261–274
29. Norris, E.M.: An Algorithm for Computing the Maximal Rectangles in a Binary Relation. Revue Roumaine de Mathématiques Pures et Appliquées **23**(2) (1978) 243–250
30. van der Merwe, D., Obiedkov, S., Kourie, D.: AddIntent: A New Incremental Algorithm for Constructing Concept Lattices. In: Proceedings of the International Conference on Formal Concept Analysis. Volume 2961 of Lecture Notes in Artificial Intelligence. Springer Berlin Heidelberg (2004) 205–206
31. Cairns, S.: Mutualizing Museum Knowledge: Folksonomies and the Changing Shape of Expertise. Curator: The Museum Journal **56**(1) (January 2013) 107–119

Author Index

Editors:	Marianne Huchard, Sergei O. Kuznetsov
Title:	CLA 2016, Proceedings of the Thirteenth International Conference on Concept Lattices and Their Applications
Publisher & Print:	National Research University Higher School of Economics Kochnovskiy proezd, dom 3 Moscow, Russia
Place, year, edition:	Moscow, Russia, 2016, 1st
Page count:	xiv+377
Number of copies:	200
Archived at:	`cla.inf.upol.cz`

ISBN 978-5-600-01454-1