CLA 2018

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

CLA Conference Series

`cla.inf.upol.cz`

Department of Computer Science, Palacký University Olomouc, Czech Republic

ISBN 978–80–244–5328–6

Department of Computer Science, Palacký University Olomouc, Czech Republic

The Fourteenth International Conference on Concept Lattices and Their Applications

CLA 2018

Olomouc, Czech Republic
June 12–14, 2018

Edited by

Dmitry I. Ignatov
Lhouari Nourine

CLA 2018
Editors © Dmitry I. Ignatov, Lhouari Nourine, 2018
Publisher © Palacký University Olomouc, 2018

Page count:	xii+291
Impression:	70
Edition:	1st
First published:	2018

Organization

CLA 2018 was organized by the Department of Computer Science, Palacký University Olomouc.

Steering Committee

Radim Belohlavek	Palacký University Olomouc, Czech Republic
Sadok Ben Yahia	Faculté des Sciences de Tunis, Tunisia
Jean Diatta	Université de la Réunion, France
Peter Eklund	IT University of Copenhagen, Denmark
Sergei O. Kuznetsov	National Research University Higher School of Economics, Moscow, Russia
Engelbert Mephu Nguifo	LIMOS, CNRS UMR 6158, 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

Dmitry I. Ignatov	National Research University Higher School of Economics, Moscow, Russia
Lhouari Nourine	LIMOS, University Blaise Pascal, Clermont-Ferrand, France

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
Jaume Baixeries	Universitat Politècnica de Catalunya, Catalonia
Radim Belohlavek	Palacký University Olomouc, Czech Republic
Karell Bertet	Laboratory L3I, University of La Rochelle, France
Franois Brucker	Ecole Centrale Marseille, France
Ana Burusco	Universidad Publica De Navarra, Spain
Aleksey Buzmakov	National Research University Higher School of Economics, Perm, Russia
Peggy Cellier	IRISA/INSA Rennes, France

Pablo Cordero	Universidad de Málaga, France
Jean Diatta	Université de la Réunion, France
Stephan Doerfel	University of Kassel, Germany
Xavier Dolques	ICube, Université de Strasbourg/ENGEES, Strasbourg, France
Florent Domenach	Akita International University, Japan
Vincent Duquenne	CNRS, Paris, France
Sebastien Ferre	Universite de Rennes 1, France
Alain Gély	LITA, Université Lorraine, Metz, France
Robert Godin	Université du Québec Montréal, Canada
Dmitry I. Ignatov	National Research University Higher School of Economics, Russia
Mehdi Kaytoue	LIRIS – INSA de Lyon, France
Stanislav Krajci	P. J. Safarik University, Slovak Republic
Francesco Kriegel	Technische Universität Dresden, Germany
Jan Konecny	Palacký University Olomouc, Czech Republic
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, Strasbourg, France
Rokia Missaoui	Université du Québec en Outaouais, LARIM, Canada
Jesús Medina Moreno	University of Cádiz, Spain
Engelbert Mephu Nguifo	LIMOS – Blaise Pascal University CNRS, France
Amedeo Napoli	LORIA Nancy, France
Sergei Obiedkov	National Research University Higher School of Economics, Russia
Manuel Ojeda-Aciego	Dept. of Applied Mathematics, University of Malaga, Spain
Petr Osička	Palacký University Olomouc, Czech Republic
Jan Outrata	Palacký University Olomouc, Czech Republic
Jean-Marc Petit	Université de Lyon, INSA Lyon, France
Uta Priss	Ostfalia University, Germany
Olivier Raynaud	LIMOS – Blaise Pascal University - CNRS, France
Sándor Radeleczki	Department of Analysis, University of Miskolc, Hungary
Franois Rioult	GREYC CNRS UMR6072 – Université de Caen, France
Camille Roth	CNRS, France
Sebastian Rudolph	Technische Universität Dresden, Germany
Christian Sacarea	Babes-Bolyai University, Romania

Barış Sertkaya	Frankfurt University of Applied Sciences, Germany
Gerd Stumme	University of Kassel, Germany
Laszlo Szathmary	University of Debrecen, Hungary
Andreja Tepavcevic	University of Novi Sad, Serbia
Petko Valtchev	Université du Québec À Montréal, Canada
Francisco J. Valverde Albacete	Universidad Carlos III de Madrid, Spain

Additional Reviewers

Inma P. Cabrera	University de Málaga, Spain
Bogdan Chornomaz	Karazin Kharkiv National University, Ukraine
Tom Hanika	Kassel University, Germany
Marie Le Guilly	LIRIS, Lyon, France
Cristina Nica	University of Strasbourg, France

Organization Committee

Radim Belohlavek	Palacký University Olomouc, Czech Republic
Eduard Bartl	Palacký University Olomouc, Czech Republic
Ivo Fridrich	Palacký University Olomouc, Czech Republic
Radek Janostik	Palacký University Olomouc, Czech Republic
Martin Kauer	Palacký University Olomouc, Czech Republic
Alena Kolovratnikova	Palacký University Olomouc, Czech Republic
Jan Konecny	Palacký University Olomouc, Czech Republic
Petr Krajca	Palacký University Olomouc, Czech Republic
Michal Krupka	Palacký University Olomouc, Czech Republic
Petr Osicka	Palacký University Olomouc, Czech Republic
Jan Outrata	Palacký University Olomouc, Czech Republic
Martin Trnecka	Palacký University Olomouc, Czech Republic
Tomas Urbanec	Palacký University Olomouc, Czech Republic

Sponsoring Institution

Faculty
of Science

Palacký University
Olomouc

esf european social fund in the czech republic

EUROPEAN UNION

MŠMT MINISTRY OF EDUCATION, YOUTH AND SPORTS

OP Education for Competitiveness 2007–13

INVESTMENTS IN EDUCATION DEVELOPMENT

DAMOL

DATA ANALYSIS AND MODELING LAB

The conference was organized by Data Analysis and Modeling Laboratory (DAMOL) that was supported by the ECOP (Education for Competitiveness Operational Programme) project no. CZ.1.07/2.3.00/20.0059, which was co-financed by the European Social Fund and the state budget of the Czech Republic during 2011–2014.

Table of Contents

Preface

Invited Contributions

Regular Papers

Papers with Short Presentation

Preface

CLA 2018 was held in Olomouc, Czech Republic, and it was organized by Dept. of Computer Science, Palacký University Olomouc from 12 until 14 June 2018. The CLA conference is an international conference dedicated to Formal Concept Analysis (FCA) and areas closely related to FCA such as data mining, information retrieval, knowledge management, data and knowledge engineering, logic, algebra and lattice theory. CLA provides a forum for researchers, practitioners, and students.

The program of the conference included four keynote talks given by the following distinguished researchers: Edouard Machery (Center for Philosophy of Science at the University of Pittsburgh, USA), Pauli Miettinen (Max-Planck-Institut für Informatik, Saarbrücken, Germany), Lakhdar Sais (CRIL "Centre de Recherche en Informatique de Lens", France) and Filip Železný (Czech Technical University in Prague).

This volume includes the selected papers and the abstracts of the invited talks. This year, 41 papers were submitted, from which 17 papers were accepted as regular papers and 6 as short presentations.

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. 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.

We also thank the Easychair conference system as it made easier most of our administration tasks related to paper submission, selection, and reviewing. Last but not least we thank Jan Outrata, who assembled the proceedings from the LaTeX files of the individual contributions that we gave him.

June 2018

Dmitry I. Ignatov
Lhouari Nourine
Program Chairs of CLA 2018

Relational Machine Learning

Filip Železný

Czech Technical University in Prague, Czech Republic

Abstract. I will explain the main concepts of relational machine learning, or more precisely, those parts of it employing logic as the knowledge-representation formalism. The talk will not cover other relational approaches such as graph-mining. I will follow what I consider the three main stages of the field's historical development. First, I will visit the roots of relational learning lying in the area of inductive logic programming. Here, one learns logical theories from examples, formalizing the problem as search in a clause subsumption lattice. A newer stream of research called statistical relational learning extended the logical underpinnings with probabilistic inference. I will illustrate this with an example of a logical graphical probabilistic model. Most recently, relational learning has received a new impetus from the current revival of (deep) neural networks. I will exemplify some promising crossovers of the two fields, including the paradigm of Lifted Relational Neural Networks conceived in my lab.

Psychological Theories of Concepts

Edouard Machery

University of Pittsburgh, USA

Abstract. In this talk, I will review classic and more recent theories of concepts, including prototype, example, and theory theories of concepts. We will also look at the modeling of concepts in psychology by means of causal bayes network and generative, hierarchical bayesian models.

Towards Cross-Fertilization between Data Mining and Constraints

Lakhdar Sais

Cril, France

Abstract. In this talk, we overview our contributions to data mining and more generally to the cross-fertilization between data mining, constraint programming and propositional satisfiability. We will focus on three contributions. First, we show how propositional satisfiability (SAT) can be used to model and solve problems in data mining. As an illustration, we present a SAT-based declarative approach for itemset, association rules and sequences mining. Then, we present an original use of data mining techniques to compress Boolean formulas. Finally, we discuss how symmetries widely investigated in Constraint Programming (CP) and Propositional Satisfiability (SAT) can be extended to deal with data mining problems.

Boolean Tensor Factorizations – and Beyond

Pauli Miettinen

Max-Planck-Institut, Germany

Abstract. Boolean matrix factorization (BMF) has become a popular method in data mining, with applications ranging from bioinformatics to lifted inference and multi-label classification. Tensor factorizations (over the standard algebra) have gained increasing interest in data analysis community in the recent years, and they have been applied to network analysis, dynamic networks, and to simplify deep neural networks, among others. Boolean tensor factorization (BTF) – a natural combination of BMF and tensors – can be seen as a generalization of BMF, where instead of a single binary relation (i.e. a matrix), we factorize a higher-arity relation (or a collection of binary relations over the same entities). In this talk I will cover what will happen when we merge ideas from standard tensor factorizations with Boolean algebra, discussing the computational complexity, possible algorithmic ideas, and potential applications. I will also cover some hybrid approaches that merge continuous and Boolean decompositions.

K-Chains Problem and Why it Matters for Extremal Contexts

Bogdan Chornomaz

markyz.karabas@gmail.com

Abstract. Here we discuss a problem of arranging k linear orders on n elements to maximize the number of sets that can be obtained as intersections of their initial intervals. We argue that this problem can shed light on a hard problem of characterizing formal contexts of bounded VC dimension, extremal with respect to the number of their objects *and* attributes. To tackle this problem we introduce limit objects, which capture their asymptotics, and propose, for all k, a tentative optimal solution. We prove that, under additional hypothesis of symmetry, it is indeed optimal for $k = 3$.

1 Introduction

As it was shown by Alexandre Albano and the author [2, 3], the growth of Vapnik-Chervonekis (VC) dimension is, in essence, the only reason for the exponential growth of formal concept lattices. Any formal context of bounded VC-dimension k has its lattice bounded in size by a polynomial in the number of its join-irreducible elements n, specifically $|L| \leq f(n,k)$, where

$$f(n,k) := \sum_{i=0}^{k-1} \binom{n}{i}.$$

This bound itself can be traced back to a well-known lemma of Sauer and Shelah [9, 10]. Here and further $\mathbf{n}$ denotes the standard n-element set $\{1,\ldots,n\}$.

Lemma 1 (Sauer-Shelah) *If $\mathcal{A}$ is a family of subsets of $\mathbf{n}$ and $|\mathcal{A}| > f(n,k)$, then $\mathcal{A}$ shatters some k-set.*

Apart from FCA perspective, this problem can be formulated in purely lattice-theoretical terms by putting a doubly founded (or, less generally, a finite) lattice into correspondence with its *standard context.* With this identification, which we will use throughout the paper, objects and attributes become join-irreducible and meet-irreducible elements, and the VC-dimension of a lattice L is defined as the largest integer k for which a boolean lattice on k elements $\mathfrak{B}(k)$ can be order-embedded into L, see [3, Lemma 1]. Lattices on n objects of VC-dimension at most k with $f(n,k)$ elements are called $(n, k+1)$*-extremal.* We use $k+1$ instead of k to emphasize that these lattices do not allow an embedding of $\mathfrak{B}(k+1)$, or, alternatively, do not shatter any $(k+1)$-set.

© Dmitry I. Ignatov, Lhouari Nourine (Eds.): CLA 2018, pp. 9–23,
ISBN 978–80–244–5328–6, Department of Computer Science, Palacký University
Olomouc, 2018.

A notable feature of $(n, k+1)$-extremal lattices is that they can be completely characterized, in a feasible fashion, through the doubling construction [2, 3], or through their canonical bases [1]. One crucial disadvantage of this approach, however, is that while the number of objects n for a lattice is bounded, there is no estimate or restriction on the number of its attributes m. However, from the duality principle, attributes and objects are interchangeable, so instead we might be interested in maximizing a lattice with respect to $\max(m, n)$, $m + n$, $\alpha n + \beta m$ or $n \cdot m$, where the latter is a good estimate for the size of the formal context in matrix form. An adequate theory of extremal contexts is thus calling for resolution of problems of a kind:

Problem 1 *Characterize concept lattices of formal contexts (G, M, I) of VC-dimension at most k, maximal in size with respect to $\sigma(|G|, |M|)$, where $\sigma(n, m) = n + m$, or $\sigma(n, m) = n \cdot m$, or $\sigma(n, m) = \alpha n + \beta m$.*

An example of a more nuanced conjecture could be as follows:

Conjecture 1 *Maximal in size concept lattice of a formal context (G, M, I) of VC-dimension at most k, such that $|G| + |M| = 2n$, and such that k divides n, is the Cartesian product of k chains of length $n/k - 1$ each:*

$$L = \bigtimes_{k} C\left(\frac{n}{k}\right),$$

where $C(l)$ is an l-element chain. The size of L is $(n/k)^k$.

At the moment, these problems seem too hard to approach. A first easy step could be to construct an $(n, k + 1)$-extremal lattice, which, at the same time, would minimize the number of attributes. Such lattices will be called $(n, k + 1)$-doubly extremal. For $k = 1$ the construction is trivial, and further on we will show that for $k = 2$ doubly extremal lattice is exactly the interval lattice. For $k \geq 3$, however, things start getting bleak. We thus resort to an even simpler problem, which will be the central object of this investigation, and, as we hope, can provide a key insight towards constructing doubly extremal lattices. We now state this problem formally, and postpone the discussion of how it is connected with doubly extremal lattices till Section 2.

Definition 1 (Configuration) *For a fixed k, let a* (discrete) k-configuration $\mathcal{C} = \{\preceq_i \mid i = 1, \ldots, k\}$ *be a set of k linear orderings of $\mathbf{n}$. We say that $\mathcal{C}$* generates *a* feasible *set $Y \subseteq \mathbf{n}$, if Y can be obtained as an intersection of initial intervals of $\leq_i$, that is,*

$$Y = \bigcap_{i=1}^{k} (x_i]_i$$

for some $x_i \in \mathbf{n}$, where $(v]_i = \{u \in \mathbf{n} \mid u \preceq_i v\}$. As a matter of convenience, the empty set is considered to be non-feasible.

Problem 2 (k-chains problem) *Describe a k-configuration $\mathcal{C}$ that generates the maximal number of sets, and the number $|\mathcal{C}|$ of its feasible sets. In particular, what is the asymptotic behavior of $|\mathcal{C}|$ when n approaches infinity, that is, what is the value of the limit*

$$\lim_{n\to\infty} \sup_{\mathcal{C}} \frac{|\mathcal{C}|}{\binom{n}{k}} = \lim_{n\to\infty} \sup_{\mathcal{C}} \frac{k! \cdot |\mathcal{C}|}{n^k}, \tag{1}$$

and which families of configurations correspond to this limit.

As it turns out, as long as we are interested in asymptotics, it is natural to consider continuous objects called *limit configurations*, introduced in Section 3, which enable us to present configuration families with specific asymptotics as a single object.

Problem 2, however, is still too hard to solve in its full extent. In Section 4 we present a tentative optimal family of k-configurations and its continuous counterpart. This object satisfies several sufficient conditions for optimality, which, due to the lack of space, were not included in the paper. However, in Section 5 we prove that, under additional condition of symmetry, the configuration for $k = 3$ is indeed optimal. We also note that the general machinery of the proof holds for arbitrary k. The only part that is specific to $k = 3$ is purely combinatorial Lemma 6. This lemma can be formulated for arbitrary k, but we were unable to handle the general case.

2 Doubly Extremal Lattices and the *K*-Chains Problem

The starting point for the estimation of the number of meet-irreducible elements (attributes) in extremal lattices is the following lemma.

Lemma 2 *Any $(n + k, k + 1)$-extremal lattice L has at least $k(n + 1)$ meet-irreducible elements, arranged in k disjoint chains of length $n + 1$ each. Every such chain contains exactly one element of rank i, for $i \in k - 1, \dots n + k - 1$.*

Proof. We proved this lemma in another paper [5, Theorem 3]. The proof uses the technique of *extremal decompositions*, which was developed in that paper, and is rather involved, so we have no possibility to reproduce it here.

We call the chains of meet-irreducible elements from Lemma 2 *the principal chains*. Figure 1 gives an illustration of this construction. It is also trivial [5, Lemma 7] that, for $k = 2$, the interval lattices are $(n, k + 1)$-extremal with no other attributes than those, provided by Lemma 2. Thus:

Corollary 3 *The interval lattices are $(n, 3)$-doubly extremal.*

For larger k, however, Lemma 2 does not describe all meet-irreducible elements. The technique of extremal decompositions from [5] can be used to prove that, for example, any $(6, 4)$-extremal lattice has at least 3 meet-irreducible elements apart from the principal chains.

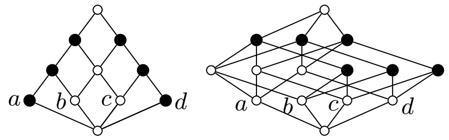


Fig. 1. The principal chains of meet-irreducible elements for $(4,3)$- and $(4,4)$-extremal lattices.

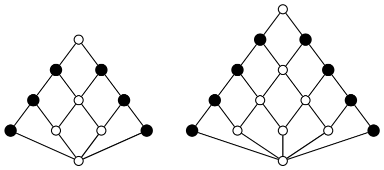


Fig. 2. The interval lattices are $(n,3)$-doubly extremal.

Instead of looking for additional meet-irreducible elements, we can reverse the question and ask how the principal chains should be constructed in order to maximize the number of elements they *generate*, that is, of elements of the lattice that can be constructed from them by intersections. Answering this question may be quite helpful for the construction of doubly extremal lattices, because of the following plausible conjecture:

Conjecture 2 *In a doubly extremal lattice the principal chains are optimal in the sense of generating the (asymptotically) maximal possible number of elements.*

Notice that every principal chain corresponds to a k*-almost ordering* of the set of its objects, where k*-almost ordering* of a set X is a partial order on X, in which all elements are comparable, except for $k-1$ smallest elements, which are incomparable with each other. For the $(4,3)$- and $(4,4)$-extremal lattices from Figure 1, the corresponding orderings are $a \leq b \leq c \leq d$ and $d \leq c \leq b \leq a$ for the former, and $b,c \leq a \leq d$; $b,d \leq a \leq c$ and $c,d \leq a \leq b$ for the latter. There is a one-to-one correspondence between the elements of the principal chains and the initial intervals of these orderings.

Thus, in order to estimate the size of the fragment of an extremal lattice generated by the principal chains, we have to find a family of k almost orderings which is is optimal, in a sense that it generates the (asymptotically) maximal number of sets as intersections of its initial intervals. This, however, is literally Problem 2, but with almost orderings instead of orderings. But it is not a problem, as switching between almost orderings and orderings does not change the asymptotics as n growth to infinity.

Apart from this, it can be easily shown that for a fixed configuration, the family of its feasible sets, together with the empty set, forms a convex geometry. It is known, however, that the convex geometries can be considered a natural generalization of the (n,k)-extremal lattices [4]. Studying the k-chains problem can thus be treated as studying extremal contexts with specific constraints on the structure of their objects and attributes.

3 Asymptotics and a Limit Object

Let us take a k-configuration $\mathcal{C}$. We say that an ordered k-tuple $(m_1,\ldots,m_k)$, $m_i \in \mathbf{n}$, is *feasible* and *corresponds* to X, if

1. $X = \bigcap_i (m_i]_i$ is a nonempty (and thus feasible) set,
2. m_i is maximal in X with respect to $\preceq_i$, for all $i = 1, \dots, k$.

Notice that for every feasible set X and every ordering $\preceq_i$ there is always an element $m_i \in X$, maximal with respect to $\preceq_i$, the k-tuple of these elements is feasible, and it is a unique feasible tuple that corresponds to X. Thus, feasible sets and feasible tuples are in one-to-one correspondence. On the other hand, while we can associate with an arbitrary k-tuple $a = (a_1, \dots, a_m)$ a feasible (or empty) set $A = \bigcap_i (a_i]_i$, in general, a will not be feasible for A, even for nonempty A, as the following example shows:

Example 1 *Let $k = 2$ and $n = 3$, $1 \preceq_1 2 \preceq_1 3$ and $1 \preceq_2 3 \preceq_2 2$. Then for the configuration $\mathcal{C} = \{\preceq_1, \preceq_2\}$ there are four feasible sets: $\{1,2,3\}$, $\{1,2\}$, $\{1,3\}$ and $\{1\}$; and their feasible tuples are $(3,2)$, $(2,2)$, $(3,3)$ and $(1,1)$. The tuple $(2,3)$ is not feasible, because although it corresponds to the feasible set $\{1\} = (2]_1 \cap (3]_2$, elements 2 and 3 do not lie in $\{1\}$, and thus can not be maximal in it with respect to any ordering.*

By putting feasible sets and feasible tuples into correspondence, we conclude that there are at most k^n feasible sets. But this estimate is way too crude, as the following statement holds:

Statement 1 *For a k-configuration $\mathcal{C}$ and a feasible tuple $(m_1, \dots, m_k)$, such that all m_i are different, a tuple $(m_{\sigma(1)}, m_{\sigma(2)}, \dots, m_{\sigma(k)})$ is not feasible for any nontrivial permutation σ.*

Proof. Let us take a nontrivial σ and fix j such that $\sigma(j) \neq j$. Then $m_{\sigma(j)} \prec_j m_j$, and thus $m_j \notin Y = \bigcap_{i=1}^k (m_{\sigma(i)}]_i$. But then m_j cannot be in a feasible tuple, corresponding to Y, a contradiction.

Thus, there can be at most $\binom{n}{k} \approx n^k/k!$ *feasible sets* with distinct components, where a *feasible set* is a k-set in $\mathbf{n}$, for which there is a corresponding feasible tuple. As for the tuples with repeating elements, their number will be asymptotically negligible comparing to n^k, so we can disregard them. This clarification also explains the choice of the denominator in the limit in (1).

When analyzing a k-configuration, or rather a family of configurations $\mathcal{C}_n$, parametrized with n, we will be interested in *the volume vol of* $\mathcal{C}_n$:

$$vol(\mathcal{C}_n) = \lim_{n\to\infty} \frac{|\mathcal{C}_n|}{\binom{n}{k}} = \lim_{n\to\infty} \frac{k! \cdot |\mathcal{C}_n|}{n^k} \leq 1. \tag{2}$$

As long as we are concentrating on the asymptotics, it will be convenient for us to define a notion of a limit object, which approximates the behavior of the sequence of configurations as n goes to infinity. Good example of such limit objects are *graphons* [7] for dense graphs, or *flag algebras* [8] for set families with prohibited configurations.

Our definition of a *limit configuration* exploits the fact that for a given k-configuration $\mathcal{C}$, every element from the base set $\mathbf{n}$ naturally corresponds to a tuple in $\mathbf{n}^k$, whose coordinates represent the relative position of the element in corresponding chains. Further on, by *measure* on $[0,1]^k$ we understand a measure on the σ-algebra of Borel sets. The set $[0,1]^k$ is equipped with a pack of *projections* $\pi_j \colon [0,1]^k \to [0,1]$, $\pi_j(x) = x_k$, where $x = (x_1, \dots, x_k)$. The Lebesgue measure of a set B is denoted by $|B|$.

Definition 2 (Limit configuration) Limit k-configuration μ *is a measure on* $[0,1]^k$*, such that for every measurable set* $B \subseteq [0,1]$*, every* $j = 1, \dots k$*, and every projection* $\pi_j \colon [0,1]^k \to [0,1]$*, it holds:* $|B| = \mu\left(\pi_j^{-1}[B]\right)$*.*

Usually we deal with a measure given in a form of a measurable weight function w *on a* 1*-dimensional manifold* $\mathcal{M} \subseteq [0,1]^k$*, defined as a line integral* $\mu(X) = \int_{x \in X} w(x) ds(x)$*, for any measurable* $X \subseteq \mathcal{M}$*. In this case we denote the configuration as* $(\mathcal{M}, w)$*.*

For a limit configuration μ, the axes, with their natural order, represent the chains; the measure μ represents relative positions of elements in the chains; and the restriction on projections reflects the fact that the elements are uniformly distributed along each chain.

A k-tuple $(x_1, \dots, x_k)$, $x_i \in [0,1]^k$ (the tuple itself is thus in $[0,1]^{k^2}$), is *feasible*, if $\pi_i(x_i) \le \pi_i(x_j)$, for all i, j. Here, for convenience, we suppose that the order on the axes is reversed, so that the top of the chains corresponds to the origin of coordinates. We denote the set of feasible tuples by $\mathcal{F} \subseteq [0,1]^{k^2}$, denoted $\mathcal{F}(\mathcal{M}) = \mathcal{F} \cap \mathcal{M}^k$ when the configuration takes form $(\mathcal{M}, w)$. The volume $vol(\mu)$ is thus defined as

$$vol(\mu) = k! \cdot \int_{(x_1, \dots, x_k) \in \mathcal{F}} \prod_{i=1}^{k} d\mu(x_i) = k! \cdot \mu^k(\mathcal{F}), \tag{3}$$

where μ^k is a measure on $[0,1]^{k^2}$, obtained as a product of k copies of μ.

It is possible, and easy, to show that for every discrete configuration it is possible to construct a limit object, so that, for large n, their volumes would be arbitrary close. And, on the contrary, for every continuous configuration it is possible to construct a family of discrete configurations, which approximate it with respect to volume. The proofs are omitted due to space restrictions.

We conclude the section with a couple of examples of limit configurations.

Example 2 (Discrete configuration) *For a given* k*-configuration* $\mathcal{C}$ *on* $\mathbf{n}$*, let us define a limit* k*-configuration* $\mu_{\mathcal{C}}$ *in the following way. Let*

$$P \subseteq [0,1]^k = \bigcup_{i \in \mathbf{n}} \bigtimes_{l=1,\dots,k} [o_l(i)/n - 1/n, o_l(i)/n],$$

where $o_l(i)$ *is a position of* i *with respect to* $\preceq_l$ *in decreasing order, that is, if* $a \preceq_l b \preceq_l c$*, then* $o_l(a) = 3$*,* $o_l(b) = 2$ *and* $o_l(c) = 1$*. We then take* $\mu_{\mathcal{C}}$ *to be a measure, uniformly distributed over* P*, that is*

$$\mu_{\mathcal{C}}(X) = |X \cap P| / |P|.$$

This construction gives a "continuous version" of the discrete configuration $\mathcal{C}$, $vol(C) \leq vol(\mu_C)$, and for reasonable $\mathcal{C}$, the difference between the volumes is small for large n.

Example 3 (Random configuration) *Let us take a k-configuration C on $\mathbf{n}$ by choosing $\preceq_i$ to be a random ordering of n, taken independently for all i. It is easy to see that for a k-tuple $(x_1, \ldots, x_k)$ with distinct elements, the probability of being feasible is $1/k^k$ (x_1 is the smallest with respect to $\preceq_1$ with probability $1/k$, etc.). Thus,*

$$\mathbb{E}\ vol(C) = \frac{k!}{k^k}.$$

A limit configuration, corresponding to this construction, is simply the Lebesgue measure $\mu(X) = |X|$, and $vol(\mu) = \mathbb{E}\ vol(C)$.

4 Tentative Solution

Author's intuition prompts, and the rest of the paper will be devoted to substantiate this claim, that the following k-configuration can be asymptotically optimal for the k-chains problem, at least for $k = 2$ and 3:

Definition 3 (Tentative optimal configuration) *Let us fix k and $n = k \cdot m$, and let us split $\mathbf{n}$ into k disjoint bunches of m elements each: $a_1, \ldots a_m$; $b_1, \ldots b_m; \ldots; z_1, \ldots, z_m$, where $a, b, \ldots, z$ is a symbolic representation of these k bunches. Then the asymptotically optimal k-configuration is $\mathcal{O}_{k,n} = \{\preceq_a, \preceq_b, \ldots, \preceq_z\}$, where*

$$\begin{aligned}
&\preceq_a = a_1 \geq \cdots \geq a_m \geq b_m \geq \cdots \geq z_m \geq b_{m-1} \geq \cdots \geq z_{m-1} \geq b_1 \geq \cdots \geq z_1;\\
&\preceq_b = b_1 \geq \cdots \geq b_m \geq a_m \geq \cdots \geq z_m \geq a_{m-1} \geq \cdots \geq z_{m-1} \geq a_1 \geq \cdots \geq z_1;\\
&\cdots\\
&\preceq_z = z_1 \geq \cdots \geq z_m \geq a_m \geq \cdots \geq y_m \geq a_{m-1} \geq \cdots \geq y_{m-1} \geq a_1 \geq \cdots \geq y_1.
\end{aligned}$$

Figure 3 below illustrates this construction.

It can be the case that the tentative optimal construction can be optimized further, for example by swapping a_i and b_i in $\preceq_c$. These modifications, however, are asymptotically negligible, and we thus refrain from trying them for the sake of simplicity. Note also, that for $k = 2$ orderings $\preceq_a$ and $\preceq_b$ will be inverse to each other, and it is easy to see that the corresponding lattice, as expected, will be the interval lattice on n elements, that is, $(n, 3)$-doubly extremal lattice. The corresponding limit configuration is defined as follows:

Definition 4 (Optimal limit configuration) *The limit configuration $\mathcal{O}_{k,\infty} = (\mathcal{M}, w)$, which corresponds to the tentative optimal configuration family $\mathcal{O}_{k,n}$, is defined as*

$$\mathcal{M} \subseteq [0,1]^k = \bigcup_{i=1,\ldots,k} [\overline{c}, \overline{e_i}],$$

where $\overline{c}, \overline{e_i} \in [0,1]^k$, $\overline{c} = (1/k, \dots, 1/k)$, $e_{i,j} = 0, i \neq j$, $e_{i,i} = 1$, and $[\overline{c}, \overline{e_i}]$ is a closed line segment. The weight function is $w(x) = 1/kL$, where L is the length of the line segment $[\overline{c}, \overline{e_i}]$.

Figure 4 below depicts the optimal limit configuration for $k = 3$. In order to calculate the volume of the optimal configuration, and to understand better the arrangement of its feasible sets, we now will describe these sets explicitly. The proof of the following proposition is by meticulous examination of the feasibility conditions for the given configuration, and will be omitted due to the lack of space.

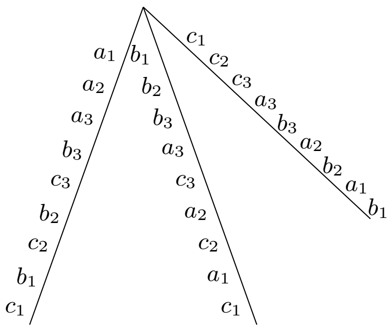


Fig. 3. The tentative optimal configuration $O_{3,9}$.

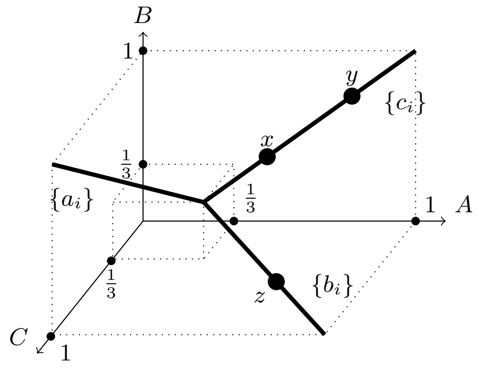


Fig. 4. The optimal limit configuration $\mathcal{O}_{3,\infty}$. The triple $\{x, y, z\}$ is a feasible set, as long as $\pi_A(x) \leq \pi_A(z)$.

Proposition 4 (Feasible sets of the optimal limit configuration.) *A set $\{x_1, \dots, x_k\} \subseteq \mathcal{O}_{k,\infty}$ is feasible iff one of the following mutually exclusive conditions hold:*

1. *all $\{x_i\}$ lie on different line segments, $x_i \in [\overline{c}, \overline{e_i}]$. The corresponding feasible tuple is $(x_1, x_2, \dots, x_k)$;*
2. *or all $\{x_i\}$, except for two of them, x_p and x_q, lie on different line segments: $x_i \in [\overline{c}, \overline{e_i}]$, $i \neq p, q$. Elements x_p and x_q lie on one of the remaining segments $[\overline{c}, \overline{e_p}]$, and for x_q it holds:*

$$\pi_q(x_q) \leq \pi_q(x_i),$$

for all $i = 1, \dots, k$. The corresponding feasible tuple is $(x_1, x_2, \dots, x_k)$.

Figure 4 shows a feasible configuration corresponding to the second case of the above proposition. This, together with volume formula (3), enables us to easily calculate the volume of the optimal solution:

Proposition 5 (Volume of the optimal limit configuration)

$$vol(\mathcal{O}_{k,\infty}) = \frac{k!}{k^{k-1}}.$$

Proof.

$$\begin{aligned} vol(\mathcal{O}_{k,\infty}) &= k! \left(\frac{1}{k}\right)^k \cdot \left[1 + k(k-1)\int_0^1 (1-x)^{k-1} dx\right] \\ &= \frac{k!}{k^k}[1 + k(k-1)/k] = \frac{k \cdot k!}{k^k} = \frac{k!}{k^{k-1}}. \end{aligned}$$

For two and three chains we thus get: $vol(\mathcal{O}_{2,\infty}) = 1$ and $vol(\mathcal{O}_{3,\infty}) = \frac{2}{3}$.

5 Symmetry

An important feature which holds for $\mathcal{O}_{k,\infty}$, is that it is *symmetric* in the following sense:

Definition 5 (Symmetry) *We say that a limit k-configuration (or simply a measure) μ is* symmetric *if $\mu(\rho_\sigma[X]) = \mu(X)$, for every permutation σ on $\mathbf{k}$ and every measurable $X \subseteq [0,1]^k$, where $\rho_\sigma \colon [0,1]^k \to [0,1]^k$ is a coordinate permutation function: $\rho_\sigma(x_1, \ldots, x_k) = (x_{\sigma(1)}, \ldots, x_{\sigma(k)})$.*

In symmetric configurations all chains look alike, and it is reasonable to suppose that the optimal configuration would be symmetric. In this section we prove that, assuming symmetry, our tentative solution for $k = 3$ is the best possible. We start with the following simple combinatorial statement:

Lemma 6 *Let (A, B, C) be a subdivision of the set $\mathbf{9}$ into three nonintersecting subsets of size three each, and let a_1, a_2, a_3; b_1, b_2, b_3 and c_1, c_2 and c_3 be enumerations of A, B and C correspondingly. We say that such triple of enumerations is* feasible *if $a_1 < b_1, c_1$, $b_2 < a_2, c_2$ and $c_3 < a_1, b_1$. Then, for a fixed subdivision, the maximal number of feasible triples is 24.*

Note. This lemma can be reformulated for larger, or even for arbitrary k, and an optimal upper bound can then be used for an upper bound for the symmetric case for arbitrary dimension. The solution strategy which we undertook there can not, however, be easily scaled, so finding such bound may prove problematic.

Proof. In order to be able to compare subdivisions, let us introduce the following notations. For a subdivision $S = (A, B, C)$, we denote the number of feasible triples of enumerations by $\mathbf{n}(S)$. For subdivisions $S = (A, B, C)$ and $S' = (A', B', C')$ we introduce the *shift* operation $[S \to S']$, which translates the enumerations of S into the enumerations of S', so that the relative order of elements inside every set remains the same. An example of the shift is given on Figure 5 below. Every shift is one-to-one and onto, and the inverse of $[S \to S']$

is $[S' \to S]$. For a subdivision S', we say that S is *dominated* by S', denoted $S \ll S'$, if for every feasible enumeration triple α of S, the triple $[S \to' S]\alpha$ is feasible for S'. Trivially, we might look for an optimal subdivision only among those, which are not dominated by any other.

Now, let us proceed with finding an optimal configuration, and note that the problem is symmetric, that is, for a feasible triple $(a_1, a_2, a_3; b_1, b_2, b_3; c_1, c_2, c_3)$ for the subdivision (A, B, C), the triple $(b_2, b_1, b_3; a_2, a_1, a_3; c_2, c_1, c_3)$ will be feasible for the subdivision (B, A, C), and so on. Here we changed the order of each enumeration in the same way as we changed the order of sets in the subdivision. Thus, without restricting generality, we can assume that the element 1 lies in A.

As 1 is the smallest element in $\mathbf{9}$, we can see that the only element from the enumeration that can be 1 is a_1: if, to the contrary, we take, say, $a_2 = 1$, then the constraint $b_2 < a_2$ will not hold. Now, we claim that an optimal position for A is $\{1, 8, 9\}$. Indeed, for a subdivision $S = (A, B, C)$, let $S' = (A', B', C')$ be the subdivision, obtained from S' by shifting the second and the third elements of A to the right. It is easy to see that in this case the subdivision S' dominates S, see Figure 5 for the illustration.

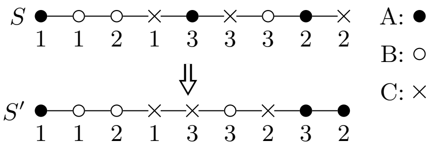


Fig. 5. Shifting the last two elements of A to the right. Numbers show the enumerations of A, B and C. Note that the unfeasible enumeration triple becomes feasible and that the second subdivision dominates the first one.

Now, in order for a subdivision to be optimal, we only need to optimally subdivide the set $\{2, \dots, 7\}$ into B and C. As before, without losing generality, we may assume that the smallest element, that is 2, lies in B. Note that assigning $b_3 = 2$ breaks the constraint $c_3 < b_3$, but it is, in principle, possible for a feasible enumeration to have $b_1 = 2$ or $b_2 = 2$, so we can not apply the same simple argument as we did for the optimal position of A.

However, there are not so many ways to make such subdivision: in fact, there are ten, so we may check them manually. In order to simplify it even further, we note that all subdivisions with $2, 3 \in B$ are dominated by $B = \{2, 3, 7\}$ and $C = \{4, 5, 6\}$, and the subdivisions with $2 \in B$ and $6, 7 \in C$ are dominated by $B = \{3, 4, 5\}$ and $C = \{2, 6, 7\}$. Other five we check manually, and obtain that there are several choices for an optimal subdivision $S^* = (B, C)$, with $\mathbf{n}(S^*) = 12$. The situation is subsumed on Figure 6.

Now, an optimal subdivision for $S = (A, B, C)$ is obtained by combining the optimal position of A with one of the optimal subdivisions for $S^* = (B, C)$. In this case, $\mathbf{n}(S) = 24$, finishing the proof of the lemma. See Figure 7 for example.

For this part we introduce additional definitions for measures on $[0, 1]^k$. The *total size* of a measure μ is just $\mu([0, 1]^k)$. Thus, if μ is a k-configuration,

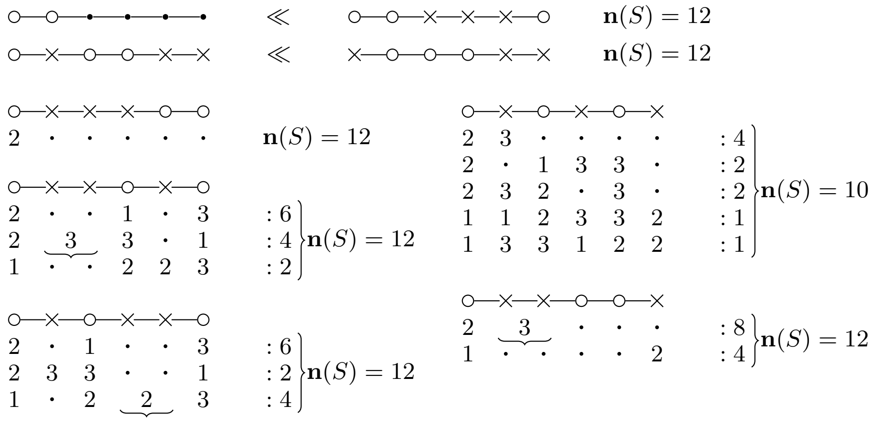


Fig. 6. Case study for an optimal subdivision of the set $\{2, \ldots, 7\}$ into B and C. Each case is a subdivision, with a list of the corresponding feasible enumerations, together with the total number of those enumerations.

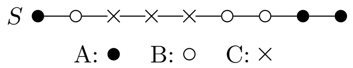


Fig. 7. An optimal subdivision $S = (A, B, C)$ with $\mathbf{n}(S) = 12$. This subdivision is also the one which is obtained from the limit object $\mathcal{O}_{3,\infty}$, see Theorem 9 for the explanation.

then its total size is 1. A measure μ is diagonal-free if $\mu(\mathcal{D}) = 0$, where $\mathcal{D} = \{x \in [0,1]^k \mid x_i = x_j, \text{ for some } i \neq j\}$. And μ is *continuous on projections* if $\mu\left(\pi_i^{-1}(X)\right) = 0$, for every i and every $X \subseteq [0,1]$ such that $|X| = 0$. Again, it is trivial to see that any k-configuration is continuous on projections. The volume $vol(\mu)$ is defined by the same formula (3) as for the k-configurations.

Lemma 7 *For a symmetric continuous on projections diagonal-free measure μ on $[0,1]^3$ of total size 1, it holds*

$$vol(\mu) \leq \frac{3!}{3^{3-1}} = \frac{2}{3}.$$

Proof. Let us fix such μ. The proof strategy is to show that for every feasible tuple on μ, only a specific fraction of tuples, obtained from it by permutations, may be feasible.

By (3), we evaluate the volume of μ as

$$vol(\mu) = 3! \int_{(x,y,z) \in \mathcal{F}_{\mathcal{D}}} d\mu(x)\mu(y)\mu(z),$$

where $\mathcal{F}_{\mathcal{D}} = \mathcal{F} \backslash \mathcal{D}^3$ is a diagonal-free version of $\mathcal{F}$. Note, that the elements of $\mathcal{F}_{\mathcal{D}}$ can have coinciding coordinates. Indeed, due to exclusion of the diagonal, for an element $(x, y, z) \in \mathcal{F}_{\mathcal{D}}$ it holds that $x_1 \neq x_2 \neq x_3$, but it may hold that, for example, $x_1 = y_1$. But finite size of μ and its continuousness on projections ensure that the volume of these points is 0. For example:

$$\begin{aligned}\int_{\substack{(x,y,z)\in\mathcal{F}_{\mathcal{D}}\\ x_1=y_1}} d\mu(x)\mu(y)\mu(z) &\leq \int_{\substack{(x,y,z)\in[0,1]^9\\ x_1=y_1}} d\mu(x)\mu(y)\mu(z)\\ &= \int_{x\in[0,1]^3}\left(\int_{y\in\pi_1^{-1}(x_1)} d\mu(y)\right) d\mu(x) = 0.\end{aligned}$$

Thus, we may restrict $\mathcal{F}_{\mathcal{D}}$ even further, to the set $\mathcal{F}^*_{\mathcal{D}}$ of points with all distinct coordinates. Now, with every x, y and z we associate one of 9! orderings $o(x,y,z)$ of their coordinates, represented as formal letters $x_1,\dots,z_3$. For example:

$$\begin{cases} x = (0.1, 0.5, 0.8)\\ y = (0.3, 0.6, 0.7)\\ z = (0.4, 0.9, 0.2)\end{cases} \Rightarrow o(x,y,z) = (x_1, z_3, y_1, z_1, x_3, y_2, y_3, x_2, z_2).$$

Also, the triple (x,y,z) is feasible if and only if $o(x,y,z)$ is feasible, that is, if the inequalities $x_1 \leq y_1, z_1$; $y_2 \leq x_2, z_2$ and $z_3 \leq x_3, y_3$ hold in o. Note that we can, in a straightforward fashion, represent o as a subdivision of $\mathbf{9}$ into sets X, Y and Z together with three relative enumerations e_x, e_y and e_z correspondingly. In the example above:

$$\begin{aligned}o(x,y,z) &= (x_1, z_3, y_1, z_1, x_3, y_2, y_3, x_2, z_2)\\ &= (xzyzxyyxz, 132, 123, 312).\end{aligned}$$

Now, if we apply permutations σ_x, σ_y and σ_z to the coordinates of x, y and z, we get

$$\begin{aligned}o(x,y,z) &= \big((X,Y,Z), e_x, e_y, e_z\big),\\ o(\sigma_x(x), \sigma_y(y), \sigma_z(z)) &= \big((X,Y,Z), \sigma_x(e_x), \sigma_y(e_y), \sigma_z(e_z)\big).\end{aligned}$$

So,

$$\begin{aligned}vol(\mu) &= 3!\int_{(x,y,z)\in\mathcal{F}^*_{\mathcal{D}}} d\mu(x)\mu(y)\mu(z) = 3!\int_{o(x,y,z)\in\mathcal{F}_o} d\mu(x)\mu(y)\mu(z)\\ &= \frac{3!}{3!^3}\sum_{\sigma_x,\sigma_y,\sigma_z}\int_{o(\sigma_x(x),\sigma_y(y),\sigma_z(z))\in\mathcal{F}_o} d\mu(\sigma_x(x))\mu(\sigma_y(y))\mu(\sigma_z(z))\\ &= \frac{3!}{3!^3}\sum_{\sigma_x,\sigma_y,\sigma_z}\int_{\left(X,Y,Z,\sigma_x(e_x),\sigma_y(e_y),\sigma_z(e_z)\right)\in\mathcal{F}_o} d\mu(x)\mu(y)\mu(z)\\ &= \frac{3!}{3!^3}\sum_{(X,Y,Z)}\mathbf{n}(X,Y,Z)\int_{(x,y,z)\in[X,Y,Z]} d\mu(x)\mu(y)\mu(z)\end{aligned}$$

$$\leq \frac{3! \cdot 24}{3!^3} \sum_{(X,Y,Z)} \int_{(x,y,z)\in[X,Y,Z]} d\mu(x)\mu(y)\mu(z)$$
$$= \frac{2}{3} \int_{(x,y,z)\in[0,1]^9} d\mu(x)\mu(y)\mu(z) = \frac{2}{3}.$$

where $\mathcal{F}_o$ is a set of feasible orderings of $\mathbf{9}$, $\mathbf{n}(X,Y,Z)$ is a number of feasible enumerations for a subdivision (X,Y,Z), and $[X,Y,Z]$ is a subset in $[0,1]^9$, for which the coordinates correspond to a subdivision (X,Y,Z). Here we used an estimation $\mathbf{n}(X,Y,Z) \leq 24$ obtained in Lemma 6. Note that this bound is exact, and it is reached by the measure that is concentrated in the areas, for which $\mathbf{n}(X,Y,Z)$ is maximal an equals 24.

Lemma 8 *For a symmetric k-configuration μ there is a family $\{\mu_a\}_{a\in(1,\infty)}$ of symmetric continuous on projections diagonal-free measures on $[0,1]^k$ of total size 1, such that $\lim_{a\to 1} vol(\mu_a) = vol(\mu)$.*

Proof. Due to the lack of space, we prove this lemma only for $k = 2$. The similar, but more elaborated proof can be carried over for arbitrary k.

For a fixed $\alpha \in (1,\infty)$ we split $[0,1]^2$ into five nonintersecting parts:

$$\mathcal{L}_L = \{(x,y) \mid y = \alpha x, x > 0\}, \qquad \mathcal{L}_U = \{(x,y) \mid x = \alpha y, y > 0\},$$
$$\mathcal{C}_L = \{(x,y) \mid y < \alpha x, x > 0\}, \qquad \mathcal{C}_U = \{(x,y) \mid x < \alpha y, y > 0\},$$
$$\mathcal{Z} = [0,1]^2 \setminus (\mathcal{L}_L \cup \mathcal{L}_U \cup \mathcal{C}_L \cup \mathcal{C}_U).$$

We define the mapping $\cdot^*\colon (\mathcal{L}_L \cup \mathcal{L}_U \cup \mathcal{C}_L \cup \mathcal{C}_U) \mapsto [0,1]^2$ as:

$$(u,v)^* = \begin{cases} (u, \alpha v), & (u,v) \in \mathcal{L}_L \cup \mathcal{C}_L, \\ (\alpha u, v), & (u,v) \in \mathcal{L}_U \cup \mathcal{C}_U. \end{cases}$$

Note that $\cdot^*$ is one-to-one on $\mathcal{C}_L \cup \mathcal{C}_U$, and $(\mathcal{L}_L \cup \mathcal{L}_U)^* = \mathcal{D}$. Now, we define μ_α as

$$\mu_\alpha(X) = \mu\big(X \cap (\mathcal{C}_L \cup \mathcal{C}_U)\big)^* + \frac{1}{2}\mu\big(X \cap \mathcal{L}_L\big)^* + \frac{1}{2}\mu\big(X \cap \mathcal{L}_U\big)^*,$$

for every measurable $X \subseteq [0,1]^2$. Informally speaking, we construct μ_α by shrinking the triangle below the diagonal along y, the triangle above the diagonal along x, and splitting in half the measure concentrated along the diagonal. The construction is illustrated on Figure 8.

It is trivial to check that μ_α is symmetric, diagonal-free, continuous on projections and has total size 1. Notice also that $\mu(Y) = \mu_\alpha([\cdot^*]^{-1}Y)$, for every measurable Y. The only thing we need to check is that the volumes of μ_a converge towards $vol(\mu)$.

$$vol(\mu_\alpha) = 2\int_{(x,y)\in\mathcal{F}\setminus\mathcal{Z}^2} d\mu_\alpha(x)\mu_\alpha(y)$$

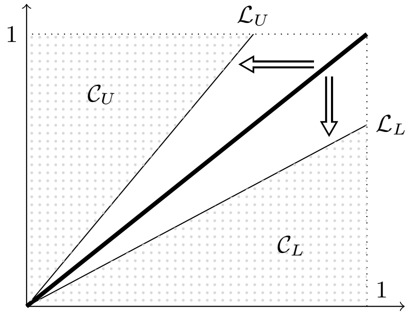


Fig. 8. Construction of μ_α.

$$
\begin{aligned}
&= 2\int_{(x,y)\in\mathcal{F}^+} d\mu_\alpha(x)\mu_\alpha(y) - 2\int_{(x,y)\in\mathcal{F}^+\backslash\mathcal{F}} d\mu_\alpha(x)\mu_\alpha(y)\\
&= 2\int_{(x,y)\in\mathcal{F}^+} d\mu(x^*)\mu_\alpha(y^*) - 2\int_{(x,y)\in\mathcal{F}^+\backslash\mathcal{F}} d\mu_\alpha(x)\mu_\alpha(y)\\
&= 2\int_{(x,y)\in(\mathcal{F}\backslash\mathcal{Z}^2)^*} d\mu(x)\mu_\alpha(y) - 2\int_{(x,y)\in\mathcal{F}^+\backslash\mathcal{F}} d\mu_\alpha(x)\mu_\alpha(y).
\end{aligned}
$$

where $\mathcal{F}\subseteq\mathcal{F}^+ = [\cdot^*]^{-1}(\mathcal{F}\backslash\mathcal{Z}^2)^*$. Then

$$
\begin{aligned}
|vol(\mu_\alpha) - vol(\mu)| \le 2&\int_{(x,y)\in(\mathcal{F}\backslash\mathcal{Z}^2)^*\Delta\mathcal{F}} d\mu(x)\mu(y)\\
&+ 2\int_{(x,y)\in\left(\mathcal{F}^+\backslash\mathcal{F}\right)^*} d\mu(x)\mu(y),
\end{aligned}
$$

where Δ denotes the symmetric difference. We estimate two summands separately. If $(x,y)\in(\mathcal{F}\backslash\mathcal{Z}^2)^*\Delta\mathcal{F}$ then something like $x_1\le y_1\le\alpha x_1$ holds (perhaps along different coordinate, perhaps with x and y swapped). Then

$$
\begin{aligned}
\int_{(x,y)\in(\mathcal{F}\backslash\mathcal{Z}^2)^*\Delta\mathcal{F}} d\mu(x)\mu(y) &\le C\int_{\{(x,y)\ \mid\ x_1\le y_1\le\alpha x_1\}} d\mu(x)\mu(y)\\
&\le C\int_x\left(\int_{y\in[x_1,\alpha+x_1]} d\mu(y)\right) d\mu(x) \le C\big|[x_1,\alpha+x_1]\big| = C\alpha.
\end{aligned}
$$

for some constant C, which depends only on k. For the second estimate let us consider what it means for (x,y) to lie in $\mathcal{F}^+\backslash\mathcal{F}$. First of all, as $\cdot^*$ is one-to-one on $\mathcal{C}_L\cup\mathcal{C}_U$, then either x or y (or both) lie in $\mathcal{L}_L\cup\mathcal{L}_U$. Say, $x\in\mathcal{L}_L$, which means that $x_1=\alpha x_2$. Then $(x,y)\notin\mathcal{F}$ but $(x',y)\in\mathcal{F}$, where $x'=(x_2,x_1)=(x_2,\alpha x_2)$. Yet again, something like $x_2\le y_2\le\alpha x_2$ holds (perhaps along different coordinate, perhaps with x and y swapped). After applying $\cdot^*$, to change from $\mathcal{F}^+\backslash\mathcal{F}$ to $(\mathcal{F}^+\backslash\mathcal{F})^*$, these restriction can only change by α. So, just like for the previous summand, we infer

$$
\int_{(x,y)\in\left(\mathcal{F}^+\backslash\mathcal{F}\right)^*} d\mu(x)\mu(y) \le D\alpha^2.
$$

for some constant D, which depends only on k. The combination of these two estimates finishes the proof.

Theorem 9 (Optimality under symmetry assumption) *For a symmetric limit* 3*-configuration* μ*, it holds:* $vol(\mu) \leq 3!/3^{3-1} = 2/3$*, that is,* $vol(\mu) \leq vol(\mathcal{O}_{3,\infty})$*. Thus, the configuration* $\mathcal{O}_{3,\infty}$ *is optimal symmetric* 3*-configuration.*

Proof. By Lemma 7, this bound holds for arbitrary symmetric continuous on projections diagonal-free measure η of total size 1, and by Lemma 8, every symmetric k-configuration μ can be approximated (in volume) by such measures with arbitrary precision.

Acknowledgment

The author is grateful to Alexandre Luiz Junqueira Hadura Albano for reading the initial draft and making a sanity check, and to anonymous reviewers, for their detailed remarks, which, hopefully, have made this paper more readable.

References

1. Albano, A.: The implication logic of (n, k)-extremal lattices. In: Bertet, K., Borchmann, D., Cellier, P., Ferré, S. (eds). Formal Concept Analysis. ICFCA 2017. Lecture Notes in Computer Science, vol, 10308, 39–55. Springer (2017)
2. Albano, A., Chornomaz, B.: Why concept lattices are large: Extremal theory for the number of minimal generators and formal concepts. In Proceedings of the Twelfth International Conference on Concept Lattices and Their Applications (CLA 2015). 73–86 (2015)
3. Albano, A., Chornomaz, B.: Why concept lattices are large: Extremal theory for generators, concepts and VC-dimension. International Journal of General Systems **46**, 440–457 (2017)
4. Chornomaz, B.: Convex geometries are extremal for generalized Sauer-Shelah bound. Accepted to The Electronic Journal of Combinatorics (2018) https:/hal.archives-ouvertes.fr/hal-01358594
5. Chornomaz, B.: Lower bound on the number of meet-irreducible elements in extremal lattices. Visnyk of V.N.Karazin Kharkiv National University. Ser. Mathematics, Applied Mathematics and Mechanics **86**, 26–48 (2017). https:/hal.archives-ouvertes.fr/hal-01773276
6. Ganter, B., Wille, R.: Formal Concept Analysis: Mathematical Foundations. Springer, Berlin-Heidelberg, (1999)
7. Lovász, L.: Large Networks and Graph Limits. American Mathematical Society. Providence, RI, (2012)
8. Razborov, A.: Flag algebras. Journal of Symbolic Logic **72**(4), 1239-1282 (2007)
9. Sauer, N.: On the density of families of sets. Journal of Combinatorial Theory, Series A **13**(1), 145 – 147 (1972)
10. Shelah, S.: A combinatorial problem: stability and order for models and theories in infinitary languages. Pacific J. Math. **41**(1), 247–261 (1972)

A First Study on What MDL Can Do for FCA

Tatiana Makhalova[1,2], Sergei O. Kuznetsov[1], and Amedeo Napoli[2]

[1] National Research University Higher School of Economics,
3 Kochnovsky Proezd, Moscow, Russia
[2] LORIA, (CNRS – Inria – U. of Lorraine), BP 239
Vandœuvre-lès-Nancy, France
tpmakhalova@hse.ru, skuznetsov@hse.ru, amedeo.napoil@loria.fr

Abstract. Formal Concept Analysis can be considered as a classification engine able to build classes of objects with a description or concepts and to organize these concepts within a concept lattice. The concept lattice can be navigated for selecting significant concepts. Then the problem of finding significant concepts among the potential exponential number of concepts arises. Some measures exist that can be used for focusing on interesting concepts such as support, stability, and other. MDL (minimum description length) is also a good candidate that was rarely used in FCA by now for such objective. In this paper we argue that MDL can give FCA practitioners a good measure for selecting significant and representative concepts.

1 Introduction

Formal concept analysis (FCA) plays an important role in Data Mining and Machine Learning. Concept lattices support mainly unsupervised settings, improving tasks such as building taxonomies and ontologies, computing implications and association rules, clustering and solving classification tasks. These tasks in practice are coupled with the problem of exponential explosion of the number of formal concepts.

By now, to tackle this issue, a lot of different approaches have been proposed, including data pre- and postprocessing, background knowledge incorporation, computing approximate concepts (see [9] for an overview). As a result, one expects to get a small set of interesting, meaningful, non-redundant concepts [3].

In this paper, we focus on the characterization of such a small set of concepts. Instead of using an interesting measure in postprocessing step, we propose to rely on the minimum description length (MDL) principle [7], which allows one to select small sets of diverse and interpretable concepts. Providing the best lossless compression of the data, the MDL optimal sets of patterns (itemsets) automatically provides a balance between the quality of fit of the data and the complexity of the model without any user-defined parameters to be set [1].

In this paper we propose a first study on the application of the MDL principle in FCA settings. To the best of our knowledge, this is one of the first papers to study the effective use of MDL in the framework of FCA.

© Dmitry I. Ignatov, Lhouari Nourine (Eds.): CLA 2018, pp. 25–36,
ISBN 978–80–244–5328–6, Department of Computer Science, Palacký University
Olomouc, 2018.

The rest of the paper has the following structure. In Section 2 we remind the basic definition used in FCA and discuss how FCA can be used to solve classification problem. Section 3 introduces the MDL principle. In Section 4 we discuss what MDL can bring to FCA. Section 5 gives conclusion and directions for future work.

2 Preliminaries

In this section we recall the main notions that are used in the paper. We study attribute sets of formal concepts, which have several alternative names, such as itemset, intent, or pattern. We discuss the application of FCA in unsupervised and supervised settings.

2.1 FCA. Basic Notions

Here we briefly recall FCA terminology [5]. A formal context is a triple (G, M, I), where $G = \{g_1, g_2, ..., g_n\}$ is called a set objects, $M = \{m_1, m_2, \dots, m_k\}$ is called a set attributes and $I \subseteq G \times M$ is a relation called incidence relation, i.e. $(g, m) \in I$ if the object g has the attribute 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. An object g is said to contain a pattern (set of items) $B \subseteq M$ if $B \subseteq g'$. The double application of $(\cdot)'$ is a closure operator, i.e. $(\cdot)''$ is extensive, idempotent and monotone. Sets $A \subseteq G$, $B \subseteq M$, such that $A = A''$ and $B = B''$, are said to be closed.

A (formal) concept is a pair (A, B), where $A \subseteq G$, $B \subseteq M$ and $A' = B$, $B' = A$. A is called the (formal) extent and B is called the (formal) intent of the concept (A, B). A partial order $\leqslant$ is defined on the set of concepts 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).

The number of formal concepts grows exponentially w.r.t. the size of a formal context, i.e. the number of objects in G and attributes in M. Thus, it becomes almost impossible to analyze and interpret the whole set of generated concepts. *Pattern discovery* techniques are designed to solve this problem. The goal of pattern discovery within the framework of FCA is to find a non-redundant set of concepts that are interesting w.r.t. specified constrains/interestingness criterion. The criterion can be applied to both intent and extent, whereas pattern discovery in general is related solely to the itemset assessment.

Example. Let us consider the toy example given in Table 1. We will use either D_1 or D_2 to compute classifiers and the remaining objects will be used to assess the quality of the classifiers. The set of attributes M includes columns $m_1, \dots, m_9$.

Table 1. An example of dataset.

data partitions		objects		m_1 : **4 legs**	m_2 : **hairs**	m_3 : **change size**	m_4 : **cold-resistant**	m_5 :**do not release** CO_2	m_6 : **black-white**	m_7 : **yellow-brown**	m_8 : **green**	m_9 : **gray**	**target** w: **animal**
D_2	D_1	g_1	dog	×	×	×			×				+
		g_2	cat	×	×	×				×			+
		g_3	frog	×		×					×		+
		g_4	car				×					×	−
		g_5	ball			×	×	×	×				−
		g_6	chair	×			×	×			×		−
		g_7	fur coat		×		×	×				×	−
		g_8	sunflower			×				×			−
		g_9	fish			×	×					×	+
		g_{10}	leopard	×	×	×				×			+
		g_{11}	table	×			×	×			×		−

The additional attribute *“target”*, i.e., class labels, is not taken into account under unsupervised settings.

Filtering concepts based on their extent and/or intent belongs to class of unsupervised problems, since any supplemental information is unavailable. In the next subsection we consider the problem of concept selection in supervised settings.

2.2 FCA under Supervised Settings: Concept-based Classifiers

In supervised settings along with the objects and their descriptions an additional attribute w is given. It specifies the class of an object. We denote the set of its values by ε.

We shall confine ourselves to two-class classification and study the simplest case, where each object belongs to a single class. In the defined settings $\varepsilon = \{+, -\}$. We consider the case where a set of objects $G = G_+ \cup G_-$ is divided into 2 disjoint subsets, i.e., a set of positive examples G_+, negative examples G_-. In practice, a set G_τ of unlabeled examples appears as well. The objects are described by attributes from M and the target attribute w is defined as follows: $gIw = \text{“}+\text{”}$ for $g \in G_+$ and $gIw = \text{“}-\text{”}$ for $g \in G_-$. The objects from $G_+ \cup G_-$ compose *training and test sets*, which are used to generate concepts and to estimate quality of classifiers, respectively.

It is assumed that there exists an unknown function that maps each object $g \in G$ (or its description $g' \subseteq M$) to an element in ε. The goal is to *reconstruct as accurately as possible the unknown function* using an observable subset of labeled objects. To do that, one builds classifiers, which can be constructed by means of FCA as follows.

For each concept (A, B) a class label $e \in \varepsilon$ is defined by majority of labeled objects in the extent, i.e. $class((A, B)) = \arg\max_{e \in \varepsilon} |A_e|/|A|$, where $A_e = G_e \cap A$. To classify an unlabeled object g w.r.t. a pair (B, e) we set the following *classification principle*:

$$(B, e)(g) = \begin{cases} e, & \text{if } B \subseteq g' \\ \emptyset, & \text{otherwise.} \end{cases} \qquad (1)$$

Table 2. The values of quality measures (Formulas 2-5) for classifiers $\{$*"cold-resistant"*$\}^-$ and $\{$*"4 legs", "change size"*$\}^+$ from the running example in Table 1. The attribute sets of the classifiers are intents of formal concepts computed on D_1. The set $\{g_8, g_9, g_{10}, g_{11}\}$ is used to assess classifiers.

classifier/ measure	$\{$*"cold-resistant"*$\}^-$	$\{$*"4 legs", "change size"*$\}^+$
prec	$\lvert\{g_{11}\}\rvert/\lvert\{g_9, g_{11}\}\rvert = 1/2$	$\lvert\{g_{10}\}\rvert/\lvert\{g_{10}\}\rvert = 1$
recall	$\lvert\{g_{11}\}\rvert/\lvert\{g_8, g_{11}\}\rvert = 1/2$	$\lvert\{g_{10}\}\rvert/\lvert\{g_9, g_{10}\}\rvert = 1/2$
sup	$\lvert\{g_9, g_{11}\}\rvert/\lvert G^*\rvert = 1/2$	$\lvert\{g_{10}\}\rvert/\lvert G^*\rvert = 1/4$
acc	$\lvert\{g_{10}, g_{11}\}\rvert/\lvert G^*\rvert = 1/2$	$\lvert\{g_{10}\}\rvert/\lvert G^*\rvert = 1/4$

According to Formula 1, we get a non-empty response e from (B, e) if an object description g' contains attribute set B. To simplify notation we will write $\hat{B}^e$ instead of (B, e). In general, B could be any itemset, not necessarily closed.

The details on classification problem in terms of FCA can be found in [6, 8].

To identify the best classifier a test set $G^* \subseteq G_+ \cup G_-$ is used, we will write G^*_+ and G^*_- for subsets of positive and negative objects in the test set G^*, i.e., for $G^* \cap G_+$ and $G^* \cap G_-$, respectively. In our paper we estimate classifiers using the measures listed below and provide small examples of their usage (see Table 2). Precision measures how many correct answers are given by the classifier:

$$\text{prec}(\hat{B}^e, G^*) = \left|\left\{g \mid \hat{B}^e(g) = e, g \in G^*_e\right\}\right| / \left|\left\{g \mid \hat{B}^e(g) = e, g \in G^*\right\}\right|. \quad (2)$$

Recall measures how many objects from a target class are characterized by the classifier (i.e. whether a classifier is specific or general):

$$\text{recall}(\hat{B}^e, G^*) = \left|\left\{g \mid \hat{B}^e(g) = e, g \in G^*_e\right\}\right| / |G^*_e|. \quad (3)$$

Support measures how many objects can be classified (correctly or not):

$$\text{sup}(\hat{B}^e, G^*) = \left|\left\{g \mid \hat{B}^e(g) = e, g \in G^*\right\}\right| / |G^*|. \quad (4)$$

The accuracy takes into account examples from the remaining classes $\varepsilon \setminus \{e\}$ unclassified by $\hat{B}^e$:

$$\text{acc}(\hat{B}^e, G^*) = \frac{\left|\left\{g \mid \hat{B}^e(g) = e, g \in G^*_e\right\} \cup \left\{g \mid \hat{B}^e(g) = \emptyset, g \in G^*_c, c \in \varepsilon \setminus \{e\}\right\}\right|}{|G^*|}. \quad (5)$$

However, other measures can also be examined [2], e.g., F1 score, AUC, etc.

In the next section we consider an ensemble of classifiers based on single concept-based classifiers defined in Formula 1.

2.3 Concept-based Classifiers

A set of the classifiers $\mathcal{S} = \left\{\hat{B}^e_j\right\}_{j \in J}$, $e \in \varepsilon$ with a rule for aggregation of their responses constitute an *ensemble of classifiers.* We call an ensemble *concept-based classifiers* (CBC) if the itemsets B are intents of formal concepts. As an aggregation rule the following principles might be chosen: class of the majority of labels, classification if responses are the same class labels, the highest priority class from the set of responses.The best CBCs are those that ensure high accuracy and have a small size of $\mathcal{S}$.

Example. Let us turn back to the running example from Table 1. The supplementary information on class labels (the target attribute w) is given in column *"target"*. We consider the following set of classifiers:

$$\mathcal{S} = \{(\{\textit{"cold-resistant"}\}, -),\ (\{\textit{"4 legs"},\ \textit{"change size"}\}, +)\}.$$

As a rule for aggregation of responses we use the priority principle, let us suppose that "–" class has higher priority than "+", thus we can apply classifiers from "–" class and, if the object remains unclassified, we try to classify it with "+" class classifiers.

CBC works for g_{10} as follows. A classifiers with the highest priority ("–" class) are firstly applied. Since $\{\textit{"cold-resistant"}\} \not\subseteq g_{10}'$, we get an empty response and turn to the classifiers of lower priority ("+" class). $\{\textit{"4 legs", "change size"}\} \subseteq g_{10}'$, we get "+" response and classify g_{10} as an member of "+" class. The object g_{11} is classified as "–", since $\{\textit{"cold-resistant"}\} \subseteq g_{11}'$. g_8 remains unclassified, since we get empty responses from all classifiers in $\mathcal{S}$, g_9 is misclassified by $(\{\textit{"cold-resistant"}\}, -)$.

Thus, a formal concept is a well-interpreted, quite intuitive and handy tool for describing subsets of objects both un- and supervised settings. However, as it was mentioned earlier, the huge number of generated concepts hampers interpretation of the results as well as its practical application.

In the next section we provide an approach that can be used to select a small set of diverse concepts.

3 Minimal Description Length Principle

3.1 Describing Data with MDL . Unsupervised Settings

The MDL principle in the context of pattern mining is formulated as follows: the best set of patterns is the set that best compresses the database [10].

The main element of this approach is a code table (CT), which is composed of "some" itemsets with their length. The best code table minimizes the total length in bits $L(D, CT) = L(D \mid CT) + L(CT \mid D)$, where $L(D \mid CT)$ is the length of the dataset encoded with the code table CT and $L(CT \mid D)$ is the length of the code table CT computed w.r.t. D. To encode an object g in a dataset one needs to select a subset of disjoint itemsets that cover all attributes of g. By $u(B) = |\{t \in D \mid B \in cover(t)\}|$ we denote the usage of itemset B in dataset D, i.e., how many times B is used to cover objects in D, where $U = \sum_{B \in CT} u(B)$ is the total usage of itemsets. The principles of building code tables will be discussed further.

To define the length of an itemset we use an optimal prefix code given by Shannon entropy, i.e., $l(B) = -logPr(B)$, where probability is defined as follows:

$Pr(B) = u(B)/U$. Thus, the itemsets with higher frequency have smaller code lengths. The details on itemset encoding and memory that is needed to store itemsets of code table are out of scope of this paper. Since we are interested in the length of the description rather than in the encoding itself (materialized codes), we consider simplified version of $L(CT, D)$ where only terms that characterize "specificity" of itemsets for the dataset are taken into account:

$$L(D \mid CT) = \sum_{g \in D} \sum_{B \in cover(g)} l(B) = - \sum_{B \in CT} u(B) \log \frac{u(B)}{U},$$

$$L(CT \mid D) = \sum_{B \in CT} code(B) + l(B),$$

where $code(B)$ is the length in bits to store itemset B in a code table.

Principles of computing a code table. A code table is computed in an incremental manner: starting from a set of single-attribute patterns, i.e. $\{\{m\} \mid m \in M\}$. The optimization procedure is based on adding a new itemset to the code table, correcting the information about usage of the other itemsets in the code table, recomputing the itemset lengths and re-encoding the data. At each iteration a new pattern can be added or not to the code table.

A set of candidates might be composed of any kind of patterns: arbitrary itemsets, closed ones, δ-itemsets, etc. The items in a set of candidates are sorted w.r.t. chosen interestingness measures, in particular, in Krimp [10] patterns are ordered by the length of itemset (its size of attribute set) and frequency.

Note that the problem of computing optimal code table implies exhaustive search for the best combination of patterns, and in practice some heuristics are used.

Example. Let us consider the principle of computing a CT using the running example. The iterative process of updating a CT for class "animal" and recomputing the cover of D_2 is described in Table 3. As candidates we use the set of concepts sorted by area, i.e., frequency × length, the ordered list of candidates with their areas is given in columns "Candidate set, area". Columns "CT" correspond to the CT at each iteration. The CT contains itemsets and the number of times each itemset is used to cover objects. The covering is given in columns "Data with covering". At the beginning it consists of single-attribute itemsets. The first object $m_1 m_2 m_3 m_6$ is covered by four single-attribute itemsets. The first candidate is $m_1 m_2 m_3$ with an area equal to 6. At the next step this candidate is used to cover attributes in the dataset covered by single-attribute itemsets. With the chosen candidate, the covering for two objects is changed (compare the first two objects in columns "Data with covering" in "Step 0" and "Step 1"). Step by step a new itemset with the maximal area is added to the CT. If a new CT compresses the data better then the old one, the itemset is accepted to the CT, otherwise, it is removed from both the CT and the candidate set. If the CT is changed, the usage of itemsets in the CT and area for candidates are recomputed.

Table 3. An iterative procedure of computing a code table for class "animal" and the cover of D_2. The names "i" and "u" stand for an itemset and its usage in covering

Step 0				**Step 1**			
CT		**Data with**	**Candidate**	**CT**		**Data with**	**Candidate**
i	**u**	**covering**	**set, area**	**i**	**u**	**covering**	**set, area**
m_3	4	$(m_1)(m_2)(m_3)(m_6)$	$m_1m_2m_3, 6$	$m_1m_2m_3$	2	$(m_1m_2m_3)(m_6)$	$m_1m_3m_8, 3$
m_1	3	$(m_1)(m_2)(m_3)(m_7)$	$m_1m_3, 6$	m_3	2	$(m_1m_2m_3)(m_7)$	$m_3m_4m_9, 3$
m_2	2	$(m_1)(m_3)(m_8)$	$m_1m_2m_3m_6, 4$	m_1, m_4	1	$(m_1)(m_3)(m_8)$	$m_1m_3, 2$
m_4	1	$(m_3)(m_4)(m_9)$	$m_1m_2m_3m_7, 4$	m_6-m_9	1	$(m_3)(m_4)(m_9)$	
m_6-m_9	1		$m_1m_3m_8, 3$	m_2, m_5	0		
m_5	0		$m_3m_4m_9, 3$				
Step 2				**Step 3**			
CT		**Data with**	**Candidate**	**CT**		**Data with**	**Candidate**
i	**u**	**covering**	**set, area**	**i**	**u**	**covering**	**set, area**
$m_1m_2m_3$	2	$(m_1m_2m_3)(m_6)$	$m_3m_4m_9, 3$	$m_1m_2m_3$	2	$(m_1m_2m_3)(m_6)$	
$m_1m_3m_8$	1	$(m_1m_2m_3)(m_7)$		$m_1m_3m_8$	1	$(m_1m_2m_3)(m_7)$	
m_3, m_4	1	$(m_1m_3m_8)$		$m_3m_4m_9$	1	$(m_1m_3m_8)$	
m_6, m_7, m_9	1	$(m_3)(m_4)(m_9)$		m_6, m_7	1	$(m_3m_4m_9)$	
m_1, m_2, m_5, m_8	0			m_1-m_5	0		
				m_8-m_9	0		

3.2 MDL under Supervised Settings

Being purely unsupervised, the MDL principle can be adapted for usage in supervised settings. The idea is to find a compressed representation for objects using code tables of each target class separately. Classes have their own code tables. A code table consists of typical patterns and their lengths (the more typical patterns have shorter lengths). To classify a new object, its set of attributes is covered by itemsets from code tables of each class. Then, the encoding lengths for each class are computed and the class that corresponds to the minimal encoding length is assigned to the object. The length reflects typicality of an object for a particular class (code table).

Example. Consider classification with code tables "CT$_1$" and "CT$_2$" from Table 4 that have been computed on sets D_1 and D_2, respectively. The details on the computing of the code tables is out of the scope of this paper (see the Krimp algorithm [10]). Each column"CT$_i$" contains code tables CT_A and CT_{NA} for "animal" and "not animal" classes, respectively. Let us consider how an object g_9 is classified with the code tables from "CT$_1$". The main steps of the covering of g_9' are given in Table 5. To find a covering we use a greedy strategy, i.e., we start from the first itemset in a code table and then stop iterating over itemsets when all attributes of the object are covered. To cover g_9' with CT_{NA} we take the first itemset $m_1m_4m_5m_8$ (see Step 1 in Table 5), it does not cover g_9', at the next step we take the second itemset m_4m_9, it covers a subset of g_9' and m_3 remains uncovered (see Step 2). The iterations over itemsets from CT_{NA} continues until all the attributes of g_9 will be covered.

Classification with code tables from Table 4 are given in Table 6. The objects are covered by itemsets from tables of "animal" and "not animal" as it is described in Table 5). The class where the object has the shortest length is assigned to this object.

In this section we considered how the MDL principle is used to select patterns (itemsets) in un- and supervised settings. In the next section we study how MDL works in the FCA framework.

Table 4. Code tables $CT_{1,A}$ and $CT_{1,NA}$, $CT_{2,A}$ and $CT_{2,NA}$ computed on datasets D_1 and D_2, respectively. The lengths of itemsets are given with their relative size. A shorter itemset is more typical, i.e., more often used to cover the data on which they had been computed.

CT$_1$						**CT**$_1$					
code table "animal", CT_A			**code table "not animal"** CT_{NA}			**code table "animal"** CT_A			**code table "not animal"** CT_{NA}		
itemsets	**usage**	**length in CT**	**itemsets**	**usage**	**length in CT**	**itemsets**	**usage**	**length in CT**	**itemsets**	**usage**	**length in CT**
						$m_1m_2m_3$	2	█			
$m_1m_2m_3$	1	██	$m_1m_4m_5m_8$	1	██	$m_3m_4m_9$	1	██	$m_1m_4m_5m_8$	1	██
$m_1m_3m_8$	1	██	m_4m_9	2	█	$m_1m_3m_8$	1	██	m_4m_9	2	█
m_1	0	█████	m_1	0	█████	m_1	0	█████	m_1	0	█████
m_2	0	█████	m_2	1	██	m_2	0	█████	m_2	1	██
m_3	0	█████	m_3	1	██	m_3	0	█████	m_3	2	█
m_4	0	█████	m_4	1	██	m_4	0	█████	m_4	1	██
m_5	0	█████	m_5	2	█	m_5	0	█████	m_5	2	█
m_6	1	██	m_6	1	██	m_6	1	██	m_6	1	██
m_7	1	██	m_7	0	█████	m_7	1	██	m_7	1	██
m_8	0	█████	m_8	0	█████	m_8	0	█████	m_8	0	█████
m_9	0	█████	m_9	0	█████	m_9	0	█████	m_9	0	█████

Table 5. The steps of the covering process of object g_9 by itemsets from the code tables of classes "animal" and "not animal", $CT_{1,A}$ and $CT_{1,NA}$, respectively. To cover g_9' with $CT_{1,A}$ we try to use every itemset from the top, i.e., $m_1m_2m_3$, $m_1m_3m_8$, m_1, etc. We stop when all attributes are covered. The covering procedure for $CT_{1,A}$ and $CT_{1,NA}$ stops after m_9 and m_3, respectively, is being considered.

Covering with CT$_1$ for "animals"			Covering with CT$_1$ for "not animals"		
Step	**Used itemset (an attempt to cover)**	**Remaining attributes in g_9' to cover**	**Step**	**Used itemset (an attempt to cover)**	**Remaining attributes in g_9' to cover**
0	-	$m_3m_4m_9$	0		$m_3m_4m_9$
1	$m_1m_2m_3$	$m_3m_4m_9$	1	$m_1m_4m_5m_8$	$m_3m_4m_9$
2	$m_1m_3m_8$	$m_3m_4m_9$	2	m_4m_9	m_3
3	m_1	$m_3m_4m_9$	3	m_1	m_3
4	m_2	$m_3m_4m_9$	4	m_2	m_3
5	m_3	m_4m_9	5	m_3	$\{\emptyset\}$
6	m_4	m_9			
	...				
11	m_9	$\{\emptyset\}$			

4 MDL in FCA: First Steps

To show that MDL can improve the practical application of FCA, in this section we discuss the results of experiments on the embedding of MDL within FCA. We used the discretized datasets from LUCS-KDD repository [4]. We split the data into 10 parts and in each of 10 experiments we use 9 of them as a training set and one as a test set. The average performance is reported in the paper. To compute code tables and to cover objects the Krimp algorithm [10] is used.

4.1 Descriptive Patterns. FCA in Unsupervised Settings

In unsupervised learning (where the target attribute is not given), one is interested in a small number of meaningful patterns. In our experiments we compute the set of closed itemsets and apply the MDL principle to them. For MDL we

Table 6. Classification with code tables (MDL-based approach) from Table 4. The class of the code table where an object has the shortest encoding length is assigned to the object

	Encoding with itemsets	**animal**	**Encoding with itemsets**	**not animal**	**decision**
CT_1	$g_8' = (m_3)(m_7)$	▬,▬	$g_8' = (m_3)(m_7)$	▬,▬	not animal
	$g_9' = (m_3)(m_4)(m_9)$	▬,▬,▬	$g_9' = (m_3)(m_4m_9)$	▬,▬,▬	not animal
	$g_{10}' = (m_1m_2m_3)(m_7)$	▬,▬	$g_{10}' = (m_1)(m_2)(m_3)(m_7)$	▬,▬,▬,▬	animal
	$g_{11}' = (m_1)(m_4)(m_5)(m_8)$	▬,▬,▬,▬	$g_{11}' = (m_1m_4m_5m_8)$	▬	not animal
CT_2	$g_{10}' = (m_1m_2m_3)(m_7)$	▬,▬	$g_{10}' = (m_1)(m_2)(m_3)(m_7)$	▬,▬,▬,▬	animal
	$g_{11}' = (m_1)(m_4)(m_5)(m_8)$	▬,▬,▬,▬	$g_{11}' = (m_1m_4m_5m_8)$	▬	not animal

Table 7. The parameters of sets of formal concepts and their proper MDL-subsets.

dataset	**nmb. of obj.**	**nmb. of attr.**	**nmb. of concepts**		**avg. length of intent**		**dataset**	**nmb. of obj.**	**nmb. of attr.**	**nmb. of concepts**		**avg. length of intent**	
			total	**MDL**	**total**	**MDL**				**total**	**MDL**	**total**	**MDL**
auto	205	135	67 557	57	8.83	**19.26**	horse colic	368	83	173 808	101	6.96	3.92
breast	699	16	642	24	7.36	**9.04**	iris	150	19	107	13	3.08	**3.92**
car	1 728	25	12 617	94	5.12	3.47	led7	3 200	24	1 937	152	4.60	**6.80**
chess	28 056	58	152 753	1 675	4.85	4.32	mushroom	8 124	90	181 945	**211**	15.23	**19.53**
dermatology	366	49	16 324	47	6.98	5.70	nursery	12 960	30	176 536	**392**	6.53	5.56
ecoli	336	29	694	25	5.49	**6.08**	page blocks	5 473	44	715	45	5.79	**10.27**
flare	1 389	38	16 303	106	6.82	**8.64**	pima indians	768	38	1 609	50	4.99	**5.86**
glass	214	46	4 704	50	5.06	4.32	ticTacToe	958	29	42 685	**160**	5.44	4.02
heart	303	50	36 708	54	7.14	5.09	wine	178	68	13 170	52	5.14	3.90
hepatitis	155	52	199 954	44	8.14	5.59	zoo	101	42	4 563	**17**	7.34	**12.24**

sort itemsets in the candidate set by “length” (the cardinality of intent) and “frequency” (the cardinality of extent).

MDL-optimal concepts are concepts whose intents are included in a code table with a non-empty usage. The results of the experiments are given in Table 7. For instance, the “auto” dataset consists of 205 objects and 135 binary attributes. The total number of formal concepts is 67 557, 57 of them are MDL-optimal.

Our experiments show that the selected itemsets might be shorter or longer on average than the itemsets in the whole set of closed concepts (see column “avg. length of intent” in Table 7). However, around 2% (12 % at most) of the concepts are selected with the MDL-principle. Thus, MDL can be considered as a threshold-free alternative for selection of interesting itemsets. It is important to notice that the subset of MDL-optimal itemsets is composed of diverse patterns and expert assumptions on interestingness of concepts can be embedded by ordering candidates w.r.t. particular interestingness measures. Since a greedy strategy is used to make a code table, one gets a set of diverse itemsets that are in agreement with interestingness.

4.2 Classifier Comparison. FCA in Supervised Settings

In this section we study both the accuracy of ensembles of classifiers $\mathcal{S}$ and their basic elements $\hat{B}$. We also compare the accuracy of the ensembles with commonly used classification methods, e.g., random forest, multilayer perceptron and support vector machine.

Comparison of single classifiers. Here we consider formal concepts as single classifiers. We study precision, recall (Formulas 2 and 3, respectively) and precision loss (Formula 6). Formula 6 is also used as a measure of overfitting, i.e., cases where accuracy on a test set is worse than on a training one. If precision loss is close to 1, we call it overfitting, the classifier makes much more errors on a test set than on a training set. If precision loss is close to 0, we get "expected" precision on a test set.

$$\text{prloss}(\hat{B}^e, G_+ \cup G_-) = \text{precision}(\hat{B}^e, G \setminus G^*) - \text{precision}(\hat{B}^e, G^*) \qquad (6)$$

In Figure 1 we show the performance of single concepts in the following 2-dimensional spaces: "precision" and "precision loss" (blue shapes), "recall" and "precision loss" (green shapes). Due to lack of space we provide two typical kinds of distributions using results for "Breast cancer" and "Wine" datasets. Precision loss is shown on the vertical axis, recall and precision are combined in the horizontal axis. The pictures are density plots for the (sub)set of concepts in the chosen dimensions. Intense-colored regions correspond to the regions with high concentration of concepts.

Let us consider the classifiers of "Breast cancer" dataset (the first line in Figure 1). The first figure shows that most concepts have precision loss close to 0, i.e., they have similar precision both on training and test set. Classifiers with precision loss close to 0 are preferable, since they have the "expected" precision even on unobservable data. Let us consider the position on the horizontal axis. The blue shape ("precision" axis) is located close to 1, it means that most concepts have high precision on a training set. A long stretch of the green shape along the axis means that the set of classifiers consists of both specific (having relatively big intents) and general concepts (recall is from 0 to 1).

The second plot corresponds to MDL-optimal classifiers. In both spaces, i.e., "precision" and "precision loss" (blue shapes), "recall" and "precision loss" (green shapes), classifiers are concentrated in two points on the "precision loss" axis, the bottom shape is brighter than the upper one. It means that a lot of classifiers have similar precision on training and test sets, and there are several classifiers that have much smaller precision on a test set (they are overfitting classifiers). Since the blue shapes are located close to 0 on the horizontal axis, we conclude that the classifiers are very precise on a training set. The concentration of the green shapes around 0 on the horizontal axis means that most classifiers have recall close to 0, thus, the classifiers are very specific.

Classifiers of "Wine" dataset demonstrate another typical distribution of the classifiers. We can read the plots as it is done above. Here we discuss the key difference between two kinds of datasets.

In our experiments, the set of classifiers on the whole set of formal concepts was comprised of either mostly one type of classifiers with precision loss close to 0 or two types of classifiers: with precision loss close to 0 and to 1. Thus, the set of concept-classifiers contains either mostly non-overfitting (good) classifiers or non- and overfitting ones. MDL-based subset usually includes both non- and overfitted classifiers, all these classifiers are quite specific (have big intents).

A reasonable question arises: what is the accuracy of ensembles of classifiers built on such different types of concepts? In the next paragraph we examine the accuracy of ensembles of classifiers that are constituted by MLD-optimal subsets.

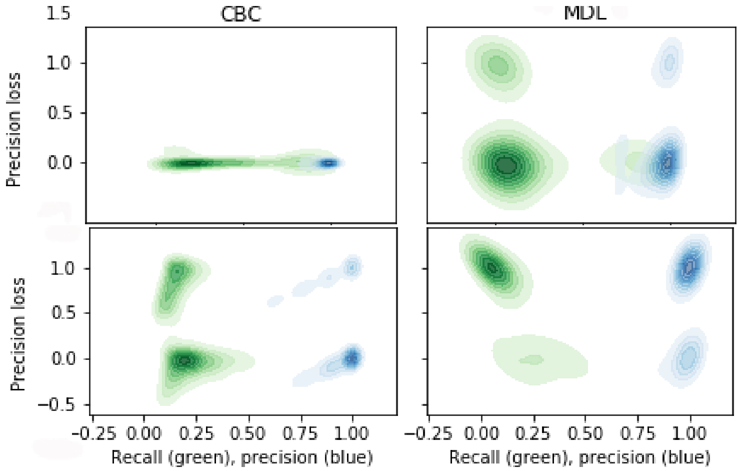


Fig. 1. Precision loss of the whole set of concepts (CBC), MDL-optimal subsets of classifiers for "Breast cancer" (top row) and "Wine" (bottom row) datasets

Comparison of ensembles of classifiers In this section we compare ensembles built on a set of formal concepts ("CBC") and its MDL-optimal subset ("MDL") with commonly used classification methods like random forests (RFs), multilayer perceptrons[3] (MLP) and support vector machines (SVM). We study the average accuracy on a test set, and the results are summarized in Table 8. As it was noticed above, we split our dataset into 10 folds (9:1 for train and test sets). The maximal accuracy over the 10 folds is also reported in the table. We pay our attention to the number of classifiers that constitute an ensemble. The smaller the number of classifiers the faster one can obtain the response and the better this response can be interpreted.

Our experiments show that among itemset-based classifiers (CBC, MDL, RF) MDL-based approach demonstrate quite good performances and have a much smaller set of classifiers. An ensemble with a small number of classifiers performs faster and is better interpretable. Thus, it is easier to classify with MDL-ensembles and understand the obtained results as well.

5 Conclusion

In this paper we have addressed the problem of selecting meaningful, non-redundant sets of formal concepts. We have proposed to use the MDL principle and to show how the expert understanding of interestingness might be incorporated into it.

The MDL principle ensures a good compression even when the set of formal concepts is huge (for example, 24 MDL-optimal among 6 432 concepts for "Breast cancer", 392 among 176 537 concepts for "nursery" dataset).

[3] We select the configuration that ensures the best accuracy among the following ones: (100:50:50),(100:50:25),(50:50:50)

Table 8. Performance of ensembles of classifiers

Dataset name	Classifier	Avg. accuracy	Max. acc.	Avg. rule numb.	Max. rule numb
breast cancer	CBC	0,86 ± 0,04	0,90	355,00 ± 7,10	361
	MDL	**0,94 ± 0,03**	0,99	**27,80 ± 0,98**	30
	RF	0,93 ± 0,04	0,99	30,60 ± 4,48	37
	MLP	**0,94 ± 0,03**	0,99	–	–
	SVM	0,93 ± 0,03	0,99	–	–
ecoli	CBC	0,84 ± 0,05	0,94	394,70 ± 14,47	410
	MDL	0,77 ± 0,10	0,85	**37,30 ± 2,10**	40
	RF	0,77 ± 0,05	0,85	1236,40 ± 532,54	1830
	MLP	0,84 ± 0,03	0,88	–	–
	SVM	**0,86 ± 0,04**	0,94	–	–
iris	CBC	0,93 ± 0,07	1,00	113,20 ± 5,57	119
	MDL	0,94 ± 0,06	1,00	18,50 ± 1,36	20
	RF	0,95 ± 0,07	1,00	169,40 ± 192,45	531
	MLP	0,93 ± 0,07	1,00	–	–
	SVM	0,93 ± 0,07	1,00	–	–

In supervised settings, MDL principle tends to choose the specific classifiers, some of them have a precision loss close to 0.9. However, the MDL principle ensures high classification accuracy.

One interesting direction for future work is to study how some interestingness measures, such as stability, might be embedded into the MDL approach. Another interesting direction is to study connection between the MDL principle and Pareto optimality.

Acknowledgements

The work was supported by the Russian Science Foundation under grant 17-11-01294 and performed at National Research University Higher School of Economics, Moscow, Russia.

References

1. Aggarwal, C.C., Han, J.: Frequent pattern mining. Springer (2014)
2. Baldi, P., Brunak, S., Chauvin, Y., Andersen, C.A., Nielsen, H.: Assessing the accuracy of prediction algorithms for classification: an overview. Bioinformatics **16**(5), 412–424 (2000)
3. Buzmakov, A., Kuznetsov, S.O., Napoli, A.: Fast generation of best interval patterns for nonmonotonic constraints. In: Joint European Conference on Machine Learning and Knowledge Discovery in Databases. pp. 157–172. Springer (2015)
4. Coenen, F.: The lucs-kdd discretised/normalised arm and carm data library (2003), http://www.csc.liv.ac.uk/ frans/KDD/Software/LUCS_KDD_DN
5. Ganter, B., Wille, R.: Formal concept analysis: Logical foundations (1999)
6. Ganter, B., Kuznetsov, S.O.: Formalizing hypotheses with concepts. In: International Conference on Conceptual Structures. pp. 342–356. Springer (2000)
7. Grünwald, P.D.: The minimum description length principle. MIT press (2007)
8. Kuznetsov, S.O.: Machine learning and formal concept analysis. In: Eklund, P. (ed.) Concept Lattices. Springer Berlin Heidelberg, Berlin, Heidelberg (2004)
9. Kuznetsov, S.O., Makhalova, T.: On interestingness measures of formal concepts. Information Sciences **442-443**, 202 – 219 (2018)
10. Vreeken, J., Van Leeuwen, M., Siebes, A.: Krimp: mining itemsets that compress. Data Mining and Knowledge Discovery **23**(1), 169–214 (2011)

Average Size of Implicational Bases

Giacomo Kahn[1] and Alexandre Bazin[2]

[1] LIMOS & Université Clermont Auvergne, France
[2] Le2i - Laboratoire Electronique, Informatique et Image, France
giacomo.kahn@isima.fr, contact@alexandrebazin.com

Abstract Implicational bases are objects of interest in formal concept analysis and its applications. Unfortunately, even the smallest base, the Duquenne-Guigues base, has an exponential size in the worst case. In this paper, we use results on the average number of minimal transversals in random hypergraphs to show that the base of proper premises is, on average, of quasi-polynomial size.

Keywords: Formal Concept Analysis, Implication Base, Average Case Analysis.

1 Introduction

Computing implication bases is a task that has been shown to be costly [6], due to their size and to the enumeration delay. Even the smallest base (the Duquenne-Guigues base) is, in the worst case, exponential in the size of the relation [12]. While the extremal combinatorics of implicational bases is a well studied subject, up to now, the average case has not received a lot of attention.

In this paper, we adapt the results presented in [5] to provide some average-case properties of implicational bases. We consider the base of proper premises and the Duquenne-Guigues base. We bound the average size of the base of proper premises under two statistical models and show that it is, on average, quasi-polynomial. This implies that the size of the Duquenne-Guigues base is on average at most quasi-polynomial. We then give an almost sure lower bound for the number of proper premises.

The paper is organized as follows: in section 2 we introduce the definitions and notations that we use in the remainder of the paper. Section 3 contains the main results of this work. In section 4, we discuss randomly generated contexts and the models that are used in this paper. We then conclude and discuss future works.

2 Definitions and Notations

In this section, we provide the definitions and results that will be used in this paper. Most of the FCA definitions can be found in [10]. From now on, we will omit the brackets in the notation for sets when no confusion is induced by this simplification.

© Dmitry I. Ignatov, Lhouari Nourine (Eds.): CLA 2018, pp. 37–45,
ISBN 978–80–244–5328–6, Department of Computer Science, Palacký University
Olomouc, 2018.

2.1 Formal Concept Analysis

A *formal context* is a triple $\mathcal{C} = (\mathcal{O}, \mathcal{A}, \mathcal{R})$ in which $\mathcal{O}$ and $\mathcal{A}$ are finite sets of objects and attributes and $\mathcal{R} \subseteq \mathcal{O} \times \mathcal{A}$ is a binary relation between them. A pair $(o, a) \in \mathcal{R}$ is read "object o has attribute a". Formal contexts can naturally be represented by cross tables, where a cross in the cell (o, a) means that $(o, a) \in \mathcal{R}$.

Table 1. Toy context $\mathcal{C}$.

	a_1	a_2	a_3	a_4	a_5
o_1	×	×			
o_2		×		×	×
o_3		×	×	×	
o_4			×		×
o_5				×	×

Table 1 shows a toy context with 5 objects and 5 attributes. It will serve as a running example throughout this paper.

Let O be a set of objects and A a set of attributes, we denote by O' the set of all attributes that are shared by all objects of O and A' the set of all objects that have all the attributes of A. More formally, $O' = \{a \in \mathcal{A} \mid \forall o \in O, (o, a) \in \mathcal{R}\}$ and $A' = \{o \in \mathcal{O} \mid \forall a \in A, (o, a) \in \mathcal{R}\}$.

The composition of those two operators, denoted $\cdot''$, forms a closure operator. A set $X = X''$ is said to be closed. A pair (O, A) with $O \subseteq \mathcal{O}$, $A \subseteq \mathcal{A}$, $A' = O$ and $O' = A$ is called a *(formal) concept* of the (formal) context $\mathcal{C}$. In this case, we also have that $A'' = A$ and $O'' = O$.

The set of all the concepts of a context, ordered by inclusion on either their sets of attributes or objects forms a complete lattice. Additionally, every complete lattice is isomorphic to the one formed by the concepts of a particular context.

Definition 1. *An* implication *(between attributes) is a pair of sets $X, Y \subseteq \mathcal{A}$. It is noted $X \rightarrow Y$.*

Definition 2. *An implication $X \rightarrow Y$ is said to* hold *in a context $\mathcal{C}$ if and only if $X' \subseteq Y'$.*

In an implication $X \rightarrow Y$, X is called the premise and Y the conclusion. Many implications are redundant, that is if an implication $a \rightarrow c$ holds, then $ab \rightarrow c$ holds and is redundant. The number of implications that hold can be quite large [12]. It is necessary to focus on the interesting ones.

Definition 3. *An implication set that allows for the derivation of all implications that hold in a context, and only them, through the application of Armstrong's axioms is called an implication base of the context.*

Definition 4 (Duquenne-Guigues Base). *An attribute set P is a pseudo-intent if and only if $P \neq P''$ and $Q'' \subset P$ for every pseudo-intent $Q \subset P$. The set of all the implications $P \to P''$ in which P is a pseudo-intent is called the Duquenne-Guigues Base.*

The Duquenne-Guigues Base, also called *canonical* base, or *stem* base has first been introduced in [11] and is the smallest (cardinality-wise) of all the bases. Here, we denote this base as Σ_{stem}. The complexity of enumerating the elements of this base is studied in [6].

Base of Proper Premises While the Duquenne-Guigues Base is the smallest base, the *base of proper premises*, or *Canonical Direct Base*, noted here Σ_{Proper}, is the smallest base for which the logical closure can be computed with a single pass. The Canonical Direct Base was initially known under five independent definitions, shown to be equivalent by Bertet and Montjardet in [2].

For a set X of attributes, let $X^{\bullet}$ be the set of attributes that are contained in X'' but not in the closure of any proper subset of X, that is

$$X^{\bullet} = X'' \setminus \left(X \cup \bigcup_{S \subset X} S'' \right).$$

X is called a *proper premise* for attribute a if $X^{\bullet}$ is not empty and $a \in X^{\bullet}$.

2.2 Hypergraphs and Transversals

Let V be a set of vertices. A hypergraph $\mathcal{H}$ is a subset of the powerset 2^V. Each $E \in \mathcal{H}$ is called an (hyper)edge of the hypergraph. A set $S \subseteq V$ is called a hypergraph transversal of $\mathcal{H}$ if it intersects every edge of $\mathcal{H}$, that is $S \cap E \neq \emptyset, \forall E \in \mathcal{H}$. A set $S \subseteq V$ is called a minimal hypergraph transversal of $\mathcal{H}$ if S is a transversal of $\mathcal{H}$ and S is minimal with respect to the subset inclusion among all the hypergraph transversals of $\mathcal{H}$. The set of all minimal hypergraph transversals of $\mathcal{H}$ forms a hypergraph, that we denote $Tr(\mathcal{H})$ and that is called the transversal hypergraph.

2.3 Proper Premises as Hypergraph Transversals

In this section, we introduce a definition of the base of proper premises based on hypergraph transversals.

Proposition 1 (from [10]). *$P \subseteq \mathcal{A}$ is a premise of $a \in \mathcal{A}$ if and only if $(\mathcal{A} \setminus o') \cap P \neq \emptyset$ holds for all $o \in \mathcal{O}$ such that $(o, a) \notin \mathcal{R}$. P is a proper premise for a if and only if P is minimal with respect to subset inclusion for this property.*

Proposition 23 from [10] uses $o \swarrow a$ instead of $(o, a) \notin \mathcal{R}$. It is a stronger condition that involves a maximality condition that is not necessary here.

The set of proper premises of an attribute is equivalent to the minimal transversals of a hypergraph induced from the context with the following proposition:

Proposition 2 (From [17]). *P is a premise of a if and only if P is a hypergraph transversal of $\mathcal{H}_a$ where*

$$\mathcal{H}_a = \{\mathcal{A} \setminus o' | o \in \mathcal{O}, (o, a) \notin \mathcal{R}\}$$

The set of all proper premises of a is exactly the transversal hypergraph $Tr(\mathcal{H}_a)$.

To illustrate this link, we show the computation of the proper premises for some attributes of Context 1. We compute the hypergraph $\mathcal{H}_a$ for a_1, a_2 and a_5. Let's begin with attribute a_1. We have to compute $\mathcal{H}_{a_1} = \{\mathcal{A} \backslash o' \, | o \in \mathcal{O}, (o, a_1) \notin \mathcal{R}\}$ and $Tr(\mathcal{H}_{a_1})$. In $\mathcal{C}$, there is no cross for a_1 in the rows o_2, o_3, o_4 and o_5. We have :

$$\mathcal{H}_{a_1} = \{\{a_1, a_3\}, \{a_1, a_5\}, \{a_1, a_2, a_3\}, \{a_1, a_2, a_4\}\}$$

and

$$Tr(\mathcal{H}_{a_1}) = \{\{a_1\}, \{a_2, a_3, a_5\}, \{a_3, a_4, a_5\}$$

We have the premises for a_1, which give implications $a_2a_3a_5 \rightarrow a_1$ and $a_3a_4a_5 \rightarrow a_1$. $\{a_1\}$ is also a transversal of $\mathcal{H}_{a_1}$ but can be omitted here, since $a \rightarrow a$ is always true.

In the same way, we compute the hypergraph and its transversal hypergraph for the other attributes. For example,

$$\mathcal{H}_{a_2} = \{\{a_1, a_2, a_3\}, \{a_1, a_2, a_4\}\} \text{ and } Tr(\mathcal{H}_{a_2}) = \{\{a_1\}, \{a_2\}, \{a_3, a_4\}\}$$

$$\mathcal{H}_{a_5} = \{\{a_1, a_5\}, \{a_3, a_4, a_5\}\} \text{ and } Tr(\mathcal{H}_{a_5}) = \{\{a_5\}, \{a_1, a_3\}, \{a_1, a_4\}\}$$

The set of all proper premises of a_i is exactly the transversal hypergraph $Tr(\mathcal{H}_{a_i})$, $\forall i \in \{1, \ldots, 5\}$, to which we remove the trivial transversals (a_i is always a transversal for $\mathcal{H}_{a_i}$). The base of proper premises for context $\mathcal{C}$ is the union of the proper premises for each attributes:

$$\Sigma_{Proper}(\mathcal{C}) = \bigcup_{a \in \mathcal{A}} Tr(\mathcal{H}_a) \setminus \{a\}$$

3 Average Size of an Implication Base

In [17], Distel and Borchmann provided expected numbers of proper premises and concept intents. Their approach, like the one in [5], uses the Erdős-Rényi model [8] to generate random hypergraphs. However, in [17], the probability for each vertex to appear in a hyperedge is a fixed 0.5 (by definition of the model) whereas the approach presented in [5] consider this probability as a variable of the problem and is thus more general.

3.1 Single Parameter Model

In the following, we assume all sets to be finite, and that $|\mathcal{O}|$ is polynomial in $|\mathcal{A}|$. We call p the probability that an object o has an attribute a. An object having an attribute is independent from other attributes and objects. We denote by $q = 1 - p$ the probability that $(o, a) \notin \mathcal{R}$. The probability of an attribute that is not a appearing in a hyperedge of $\mathcal{H}_a$ is also q.

The hypergraphs that we consider in the following are sub-hypergraphs constructed from $\mathcal{H}_a$ by removing a and removing all the hyperedges that contained only a. The transversal hypergraph of a hypergraph constructed in this way is exactly $Tr(\mathcal{H}_a) \setminus \{a\}$. This allows us to consider the transversal hypergraph without adding a as a premise for a. The average number of hyperedges of this hypergraph is $m = |\mathcal{O}| \times q \times (1 - p^{|\mathcal{A}|-1})$. Indeed, there is one hyperedge for each object o for which $(o, a) \notin \mathcal{R}$ and there exists an attribute a_2 such that $(o, a_2) \notin \mathcal{R}$ (otherwise the edge would be empty and, as such, removed). We note n the number of vertices of $\mathcal{H}_a \setminus \{a\}$. At most all attributes appear in $\mathcal{H}_a \setminus \{a\}$, except a, so $n \leq |\mathcal{A}| - 1$.

Proposition 3 (Reformulated from [5]). *In a random hypergraph with m edges and n vertices, with $m = \beta n^{\alpha}, \beta > 0$ and $\alpha > 0$ and a probability p that a vertex appears in an edge, there exists a positive constant c such that the average number of minimal transversals is*

$$O\left(n^{d(\alpha) log_{\frac{1}{q}} m + c \ln \ln m}\right)$$

with $q = 1 - p$, $d(\alpha) = 1$ if $\alpha \leq 1$ and $d(\alpha) = \frac{(\alpha+1)^2}{4\alpha}$ otherwise.

Proposition 3 bounds the average number of minimal transversals in a hypergraph where the number of edges is polynomial in the number of vertices. In [5], the authors also prove that this quantity is quasi-polynomial.

From Prop. 3 we can deduce the following property for the number of proper premises for an attribute.

Proposition 4. *In a random context with $|\mathcal{A}|$ attributes, $|\mathcal{O}|$ objects and probability p that $(o, a) \in \mathcal{R}$, the number of proper premises for an attribute is on average:*

$$O\left((|\mathcal{A}| - 1)^{\left(d(\alpha) log_{\frac{1}{p}}\left(|\mathcal{O}| \times (q \times (1 - p^{|\mathcal{A}|-1}))\right) + c \ln \ln\left(|\mathcal{O}| \times (q \times (1 - p^{|\mathcal{A}|-1}))\right)\right)}\right)$$

and is quasi-polynomial in the number of objects.

Proposition 4 states that the number of proper premises of an attribute is on average quasi-polynomial in the number of objects in a context where the number of objects is polynomial in the number of attributes.

As attributes can share proper premises, $|\Sigma_{Proper}|$ is on average less than $|\mathcal{A}| \times O\left((|\mathcal{A}|-1)^{\left(d(\alpha)log_{\frac{1}{p}}\left(|\mathcal{O}|\times q\times(1-p^{|\mathcal{A}|-1}))\right)+c\ln\ln\left(|\mathcal{O}|\times q\times(1-p^{|\mathcal{A}|-1}))\right)\right)}\right)$

Since $|\Sigma_{stem}| \leq |\Sigma_{Proper}|$, Prop. 4 immediately yields the following corollary:

Corollary 1. *The average number of pseudo-intents in a context where the number of objects is polynomial in the number of attributes is less than or equal to:*

$$|\mathcal{A}| \times O\left((|\mathcal{A}|-1)^{\left(d(\alpha)log_{\frac{1}{p}}\left(|\mathcal{O}|\times q\times(1-p^{|\mathcal{A}|-1})\right)+c\ln\ln\left(|\mathcal{O}|\times q\times(1-p^{|\mathcal{A}|-1})\right)\right)}\right)$$

Corollary 1 states that in a context where the number of object is polynomial in the number of attributes, the number of pseudo-intents is on average at most quasi-polynomial.

3.2 Almost Sure Lower Bound on the Number of Proper Premises

An almost sure lower bound is a bound that is true with probability close to 1. In [5], the authors give an almost sure lower bound for the number of minimal transversals.

Proposition 5 (Reformulated from [5]). *In a random hypergraph with m edges and n vertices, and a probability p that a vertex appears in an edge, the number of minimal transversals is almost surely greater than*

$$\mathcal{L}_{MT} = n^{log_{\frac{1}{q}} m + O(\ln\ln m)}$$

Proposition 5 states that in a random context with probability p that a given object has a given attribute, one can expect at least $\mathcal{L}_{MT}$ proper premises for each attribute.

Proposition 6. *In a random context with $|\mathcal{A}|$ attributes, $|\mathcal{O}|$ objects and probability q that a couple $(o,a) \notin \mathcal{R}$, the size of Σ_{Proper} is almost surely greater than*

$$|\mathcal{A}| \times (|\mathcal{A}|-1)^{\left(log_{\frac{1}{p}}\left(|\mathcal{O}|\times q\times(1-p^{|\mathcal{A}|-1})\right)+O(\ln\ln\left(|\mathcal{O}|\times q\times(1-p^{|\mathcal{A}|-1})\right))\right)}$$

As Prop 6 states a lower bound on the number of proper premises, no bound on the number of pseudo-intents can be obtained this way.

3.3 Multi-parametric Model

In this section we consider a multi-parametric model that fits real life data better. In this model, each attribute j has a probability p_j of appearing in the description of a given object. All the attributes are not equiprobable.

We consider a context with m objects and n attributes. The set of attributes is partitioned into 3 subsets:

- The set U contains the attributes that appear in a lot of objects' descriptions (ubiquitous attributes). For all attributes $u \in U$ we have $q_u = 1 - p_u < \frac{x}{m}$ with x a fixed constant.
- The set R represents rare events, i.e. attributes that rarely appear. For all attributes $r \in R$, we have that $p_r = 1 - \frac{1}{\ln n}$ tends to 0.
- The set $F = \mathcal{A} \setminus (U \cup R)$ of other attributes.

Proposition 7 (Reformulated from theorem 3 [5]). *In the multi-parametric model, we have:*

- *If* $|F \cup R| = O(\ln |\mathcal{A}|)$*, then the size of the base of proper premises is on average at most polynomial.*
- *If* $|R| = O((\ln |\mathcal{A}|)^c)$*, then the size of the base of proper premises is on average at most quasi-polynomial.*
- *If* $|R| = \Theta(|\mathcal{A}|)$*, then the size of the base of proper premises is on average at most exponential on* $|R|$*.*

Proposition 7 states that when most of the attributes are common (that is, are in the set U), $|\Sigma_{Proper}|$ is on average at most polynomial. When there is a logarithmic quantity of rare attributes (attributes in R), $|\Sigma_{Proper}|$ is on average at most quasi-polynomial (in the number of objects). When most of the attributes are rare events, $|\Sigma_{Proper}|$ is on average at most exponential.

As in the single parameter model, Prop. 7 also yields the same bounds on the number of pseudo-intents.

4 Discussion on Randomly Generated Contexts

The topic of randomly generated contexts is important in FCA, most notably when used to compare performances of algorithms. Since [13], a few experimental studies have been made. In [4], the authors investigate the Stegosaurus phenomenon that arises when generating random contexts, where the number of pseudo-intents is correlated with the number of concepts [3].

As an answer to the Stegosaurus phenomenon raised by experiments on random contexts, in [9], the author discusses how to randomly and uniformly generate closure systems on 7 elements.

In [16], the authors introduce a tool to generate less biased random contexts, avoiding repetition while maintaining a given density, for any number of elements. However this tool doesn't ensure uniformity.

The partition of attributes induced by the multi-parametric model allows for a structure that is close to the structure of real life datasets [5]. However, we can't conclude theoretically on whether this model avoids the stegosaurus phenomenon discussed in [4]. This issue would be worth further theoretical and experimental investigation.

5 Conclusion

In this paper, we used results on average-case combinatorics on hypergraphs to bound the average size of the base of proper premises. Those results concerns only the proper premises, and can't be applied on the average number of pseudo-intents. However, as the Duquenne-Guigues base is, by definition, smaller than the base of proper premises, the average size of the base of proper premises can serve as an average bound for the number of pseudo-intents.

This approach does not give indications on the number of concepts. However, there exists some works on this subject [1, 7, 15].

As the average number of concepts is known [7, 15], and this paper gives some insight on the average size of some implicational bases, future works can be focused on the average number of pseudo-intents. It would also be interesting to study the average number of n-dimensional concepts or implications, with $n \geq 3$ [14, 18].

Acknowledgments

This research was partially supported by the European Union's "*Fonds Européen de Développement Régional (FEDER)*" program.

References

1. Albano, A., Chornomaz, B.: Why concept lattices are large: extremal theory for generators, concepts, and vc-dimension. Int. J. General Systems **46**(5), 440–457 (2017). https://doi.org/10.1080/03081079.2017.1354798, https://doi.org/10.1080/03081079.2017.1354798
2. Bertet, K., Monjardet, B.: The Multiple Facets of the Canonical Direct Unit Implicational Basis. Theor. Comput. Sci. **411**(22-24), 2155–2166 (2010). https://doi.org/10.1016/j.tcs.2009.12.021, https://doi.org/10.1016/j.tcs.2009.12.021
3. Borchmann, D.: Decomposing finite closure operators by attribute exploration. ICFCA 2011, Supplementary Proceedings (2011)
4. Borchmann, D., Hanika, T.: Some experimental results on randomly generating formal contexts. In: Proceedings of the Thirteenth International Conference on Concept Lattices and Their Applications, Moscow, Russia, July 18-22, 2016. pp. 57–69 (2016), http://ceur-ws.org/Vol-1624/paper5.pdf
5. David, J., Lhote, L., Mary, A., Rioult, F.: An Average Study of Hypergraphs and their Minimal Transversals. Theor. Comput. Sci. **596**, 124–141 (2015). https://doi.org/10.1016/j.tcs.2015.06.052, https://doi.org/10.1016/j.tcs.2015.06.052

6. Distel, F., Sertkaya, B.: On the complexity of enumerating pseudo-intents. Discrete Applied Mathematics **159**(6), 450–466 (2011). https://doi.org/10.1016/j.dam.2010.12.004, https://doi.org/10.1016/j.dam.2010.12.004
7. Emilion, R., Lévy, G.: Size of random galois lattices and number of closed frequent itemsets. Discrete Applied Mathematics **157**(13), 2945–2957 (2009). https://doi.org/10.1016/j.dam.2009.02.025, https://doi.org/10.1016/j.dam.2009.02.025
8. Erdős, P., Rényi, A.: On the evolution of random graphs. Publ. Math. Inst. Hung. Acad. Sci **5**(1), 17–60 (1960)
9. Ganter, B.: Random extents and random closure systems. In: Proceedings of The Eighth International Conference on Concept Lattices and Their Applications, Nancy, France, October 17-20, 2011. pp. 309–318 (2011), http://ceur-ws.org/Vol-959/paper21.pdf
10. Ganter, B., Wille, R.: Formal Concept Analysis - Mathematical Foundations. Springer (1999)
11. Guigues, J. L., D.V.: Familles Minimales d'Implications Informatives Résultant d'un Tableau de Données Binaires. Mathématiques et Sciences Humaines **95**, 5–18 (1986), http://eudml.org/doc/94331
12. Kuznetsov, S.O.: On the Intractability of Computing the Duquenne-Guigues Base. J. UCS **10**(8), 927–933 (2004). https://doi.org/10.3217/jucs-010-08-0927, https://doi.org/10.3217/jucs-010-08-0927
13. Kuznetsov, S.O., Obiedkov, S.A.: Comparing performance of algorithms for generating concept lattices. J. Exp. Theor. Artif. Intell. **14**(2-3), 189–216 (2002). https://doi.org/10.1080/09528130210164170, https://doi.org/10.1080/09528130210164170
14. 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, California, USA, August 14-18, 1995, Proceedings. pp. 32–43 (1995). https://doi.org/10.1007/3-540-60161-9-27, https://doi.org/10.1007/3-540-60161-9-27
15. Lhote, L., Rioult, F., Soulet, A.: Average number of frequent (closed) patterns in bernouilli and markovian databases. In: Proceedings of the 5th IEEE International Conference on Data Mining (ICDM 2005), 27-30 November 2005, Houston, Texas, USA. pp. 713–716 (2005). https://doi.org/10.1109/ICDM.2005.31, https://doi.org/10.1109/ICDM.2005.31
16. 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: IEEE International Conference on Systems, Man, and Cybernetics, Manchester, SMC 2013, United Kingdom, October 13-16, 2013. pp. 3464–3470 (2013). https://doi.org/10.1109/SMC.2013.591, https://doi.org/10.1109/SMC.2013.591
17. Ryssel, U., Distel, F., Borchmann, D.: Fast Algorithms for Implication Bases and Attribute Exploration Using Proper Premises. Ann. Math. Artif. Intell. **70**(1-2), 25–53 (2014). https://doi.org/10.1007/s10472-013-9355-9, https://doi.org/10.1007/s10472-013-9355-9
18. Voutsadakis, G.: Polyadic Concept Analysis. Order **19**(3), 295–304 (2002). https://doi.org/10.1023/A:1021252203599, http://dx.doi.org/10.1023/A:1021252203599

Mining Correlated Association Rules from Multi-Relational Data using Interval Patterns

Hirohisa Seki and Masahiro Nagao*

Dept. of Computer Science, Nagoya Institute of Technology
Gokiso-cho, Showa-ku, Nagoya 466-8555, Japan
seki@nitech.ac.jp, m.nagao.738@nitech.jp

Abstract. In this paper, we consider the problem of mining numerical association rules (ARs) from a multi-relational database (MRDB). More specifically, we examine the effectiveness of numerical ARs with *interval patterns (IPs)* proposed by Kaytoue *et al.* in FCA (Formal Concept Analysis), and show that the *MinIntChange* algorithm by Kaytoue *et al.* can be readily extended to mine correlated interval-based ARs with the maximal significance in terms of the χ^2 measure, by incorporating into the algorithm a pruning technique by Morishita *et al.* Moreover, since the search space for computing closed IPs becomes larger as the number of numerical attributes increases, we utilize *Super CWC*, an off the shelf feature selection algorithm to reduce the number of attributes to use. Our approach is experimentally evaluated and compared with the conventional methods such as a discretization-based approach or an optimization-based approach.

1 Introduction

Numerical data arise prevalently in databases, including business and scientific databases. Handling numerical (or quantitative) data in data mining has attracted much attention since the work on mining quantitative association rules by Srikant and Agrawal [16]. Conventionally, *data discretization* is commonly used to handle numerical data; for a quantitative attribute which can have continuous values, it reduces the number of values by dividing the range of the attribute into intervals. The other approaches to handling numerical data have also been proposed, including a statistical distribution-based approach and an optimization-based approach (see the survey in [14]).

Kaytoue *et al.* [8] proposed an FCA-based approach to handling quantitative attributes, and introduced the notions of *closed interval patterns (CIPs)* as well as *generators*. The notion of IPs is an instance of the general framework of *pattern structures* studied by Ganter and Kuznetsov [5]. Recently, some methods have been proposed to use CIPs for mining association rules [6,13]. In particular, the approach in [13] handles multi-relational data mining (MRDM); it uses CIPs for mining *relational* quantitative association rules of the form $A \rightarrow C$, where

* Current Affiliation: Fujitsu Limited, Japan.

© Dmitry I. Ignatov, Lhouari Nourine (Eds.): CLA 2018, pp. 47–58,
ISBN 978–80–244–5328–6, Department of Computer Science, Palacký University
Olomouc, 2018.

A and C are relational patterns (i.e., logical conjunctions), and they consist of categorical attributes and quantitative ones from a given multi-relational data. In both work, although closed interval patterns allow us to represent intervals concisely, the numbers of generated patterns (i.e., rules) are still large, which makes the computation expensive and imposes a significant burden on the user's understanding.

In this paper, we study the problem of mining optimal relational association rules that have the maximum χ^2 value between the assumption and the conclusion of the rule. To find such rules, we use the original *MinIntChange* algorithm by Kaytoue *et al.*, and incorporate into it a pruning technique by Morishita *et al.* [11]. Moreover, since the search space for computing closed IPs becomes larger as the number of numerical attributes increases, we utilize *Super CWC* [15], an off the shelf feature selection algorithm to reduce the number of attributes to use. We give some experimental results, which show the effectiveness of the proposed method.

The organization of the rest of this paper is as follows. We first summarize some basic notations and definitions of relational association rule mining and interval patterns in Sect. 2. We then explain our approach to mining quantitative association rules from multi-relational data in Sect. 3, and show some experimental results in Sect. 4. Finally, we give a summary of this work in Sect. 5.

2 Relational Association Rules with Quantitative Attributes

2.1 Relational Pattern Mining and Interval Patterns

We use some basic notions of MRDM in [3]. To represent data and patterns, we use a class of first-order logical formulas. An *atom* is an expression of the form $p(t_1, \ldots . t_n)$, where p is a *predicate* and each t_i is a *term* (i.e., a constant or a variable). A substitution $\theta = \{X_1/t_1, \ldots, X_n/t_n\}$ is an assignment of terms to variables. The result of applying a substitution θ to a formula (i.e., an atom or a conjunction in this case) F is the formula $F\theta$, where all occurrences of variables V_i have been simultaneously replaced by the corresponding terms t_i in θ. The set of variables occurring in a formula F is denoted by $Var(F)$. A *pattern* is expressed as a conjunction $l_1 \wedge \cdots \wedge l_n$ of atoms, denoted simply by $l_1, \ldots, l_n$.

A database DB is a set of ground atoms. For a pattern C, let $answerset(C; DB)$ be the set of substitutions θ such that $C\theta$ is logically entailed by a database DB, denoted by $DB \models C\theta$.

In MRDM, we often specify one of the predicates as a *key* (e.g., [2,1]), which determines the entities of interest and what is to be counted. The key (target) is thus to be present in all patterns considered. Given a database DB and a conjunction C containing a key atom $key(X)$, the *support* (or *frequency*) of C, denoted by $supp(C)$, is defined to be the number of different keys that answer C divided by the total number of keys. C is said to be *frequent*, if $supp(C)$ is no less than some user defined threshold *minsup*.

$\mathit{customerId}(c_1)$.	$\mathit{marriedTo}(c_1, c_2)$.	$\mathit{income}(c_1, 200)$.
$\mathit{customerId}(c_2)$.	$\mathit{marriedTo}(c_2, c_1)$.	$\mathit{income}(c_2, 120)$.
$\mathit{customerId}(c_3)$.	$\mathit{marriedTo}(c_3, c_4)$.	$\mathit{income}(c_3, 50)$.
...		
$\mathit{age}(c_1, 30)$.	$\mathit{bigSpender}(c_1)$.	
$\mathit{age}(c_2, 25)$.	$\mathit{bigSpender}(c_2)$.	
$\mathit{age}(c_3, 55)$.		
...		

Fig. 1. An Example Database *DB* with *customerId* as a key: adapted from [3].

An association rule we consider in this paper is an existentially quantified implication of the form: $A \to C$, where A (resp., C) is a conjunction of the form: $a_1, \ldots, a_m$ (resp., a single atom) ($m \geq 1$). We call A (C) the *antecedent* (*conclusion*) of the rule, respectively. The *support* of a (relational) association rule is defined as the support of $A \wedge C$, while the *confidence* of an association rule is defined as the support of C divided by the support of the antecedent A. Following [7], we call a rule *strong*, if it satisfies both a minimum support threshold (*minsup*) and a minimum confidence (*minconf*).

Example 1. Consider a toy example of a multi-relational database *DB* in Fig. 1, which is adapted and simplified from [3]. Predicate *customerId* is assumed to be a key. Let P be a pattern of the form: $\mathit{customerId}(X)$, $\mathit{age}(X, Q_1)$, $\mathit{marriedTo}(X, Y)$, $\mathit{income}(Y, Q_2)$, whose meaning is obvious. Then, $\mathit{answerset}(P; DB)$ contains substitutions $\{X/c_1, Q_1/30, Y/c_2, Q_2/120\}$ and $\{X/c_2, Q_1/25, Y/c_1, Q_2/200\}$, for example.

The following rule is an example of association rules:

$$\mathit{customerId}(X), \mathit{age}(X, Q_1), \mathit{marriedTo}(X, Y), \mathit{income}(Y, Q_2) \to \mathit{bigSpender}(X). \tag{1}$$

In the above, Q_1 and Q_2 are quantitative attributes, while the others are considered to be categorical ones. □

We call a variable corresponding to a quantitative (resp., categorical) attribute a *quantitative* (resp., *categorical*) *variable.* We also call a variable occurring in a key predicate a *key variable*.

Relational Association Rules with Interval Patterns We use interval patterns to specify constraints on quantitative variables in an association rule. In the aforementioned association rule (1), for example, we consider the following association rule with constraints consisting of interval patterns:

$$\begin{gathered}\mathit{customerId}(X), \mathit{age}(X, Q_1), \mathit{marriedTo}(X, Y), \mathit{income}(Y, Q_2), \\ \langle Q_1, Q_2 \rangle \in \langle [l_1, u_1], [l_2, u_2] \rangle \to \mathit{bigSpender}(X).\end{gathered} \tag{2}$$

where l_i (u_i) is a value of the domain of the attribute Q_i ($i = 1, 2$), respectively.

Formally, let A be a conjunction such that A contains quantitative variables $Q_1, \ldots, Q_k$ ($k \geq 1$). Then, we call an expression $\boldsymbol{c}$ of the form "$\langle Q_1, \ldots, Q_k \rangle \in \langle I_1, \ldots, I_k \rangle$" an *interval constraint of* A, where I_i ($1 \leq i \leq k$) is an interval pattern for Q_i.

Let θ be a substitution for $\mathit{Var}(A)$ and $I_i = [e_i, f_i]$ for some e_i and f_i. Then, $\mathit{DB} \models (A, \boldsymbol{c})\theta$ iff $\mathit{DB} \models A\theta$ and $Q_i\theta \in [e_i, f_i]$ for $i = 1, \ldots, k$.

For simplicity, we write simply "A, I" instead of $A, \langle Q_1, \ldots, Q_k \rangle \in \langle I_1, \ldots, I_k \rangle$, where $I = \langle I_1, \ldots, I_k \rangle$ and we call I an *interval pattern* of A.

For a conjunction A which has no categorical variables except a key variable, we can define a *closed pattern* in the same way as [8]; let $S = \{\theta_1, \cdots, \theta_n\}$ ($n \geq 0$) be a set of substitutions for variables in A and let $Q_1, \ldots, Q_k$ be quantitative variables in A. Then, we define a mapping $\delta(\cdot)$ as follows: for a substitution $\theta \in S$ such that $\theta \supseteq \{Q_1/a_1, \ldots, Q_k/a_k\}$, $\delta(\theta) = \langle [a_1, a_1], \ldots, [a_k, a_k] \rangle$. Namely, the mapping δ maps θ into an k-dimensional interval pattern $\langle [a_1, a_1], \ldots, [a_k, a_k] \rangle$.

Definition 1 (closed pattern).

For a conjunction A such that A has no categorical variables except a key variable, let $S = \{\theta_1, \ldots, \theta_n\}$ ($n \geq 0$) be a set of substitutions for variables in A, and I an interval pattern of A. We consider the following two operators $(\cdot)^\square$:

$I^\square = \{\theta \mid \theta \in \mathit{answerset}(A, I; \mathit{DB})\}$,

$S^\square = \langle \delta(\theta_1) \sqcap \cdots \sqcap \delta(\theta_n) \rangle$.[1].

Let I and J be an interval pattern of A. Then, I and J are *equivalent* if $I^\square = J^\square$ and we write it by $I \equiv J$. We call I *closed* if there does not exist any other interval pattern J such that $I \equiv J$ and $I \sqsubseteq J$. □

2.2 Correlation Measures

Since the framework using the support/confidence only generates too many rules, we usually use another measure to find "interesting" ones among the generated rules. χ^2-value is such a measure to find correlated rules; it is defined as a normalized derivation of observation from expectation. Given a contingency table in Table 1, where given m and n are both assumed to be constants, χ^2 values are determined by x and y, and we thus denote it by $\chi^2(x, y)$. The following property of χ^2-value is shown by Morishita *et al.* [11].

Lemma 1 (Morishita *et al.*). *[11]* Let $r(I_0)$ be a rule with an interval pattern I_0, and let a (b, resp.) be the number of (positive) tuples that satisfy the antecedent of $r(I_0)$. Let $r(I)$ be a rule with an interval pattern I, and let p (q, resp.) be the number of (positive) tuples that satisfy the antecedent of $r(I)$ such that $0 \leq p \leq a$, $0 \leq q \leq b$, $q \leq p$ and $(p - q) \leq (a - b)$. Then, we have

$$\chi^2(p, q) \leq \max\{\chi^2(b, b), \chi^2(a - b, 0)\}. \tag{3}$$

□

[1] For $I_1 = \langle [a_i, b_i] \rangle_{i \in \{1,;k\}}$ and $I_2 = \langle [e_i, f_i] \rangle_{i \in \{1,;k\}}$, $\sqcap$ is the infimum operator defined by $I_1 \sqcap I_2 = \langle [\min(a_i, e_i), \max(b_i, f_i)] \rangle_{i \in \{1,\ldots,k\}}$, and $I_2 \sqsubseteq I_1 \iff [e_i, f_i] \subseteq [a_i, b_i], \forall i \in \{1, \ldots, k\}$.

The right-hand side of (3) gives an upper bound of $\chi^2(p, q)$, and we thus denote it by $ub(r(I))$.

Table 1. Contingency Table for Rule r=$A \rightarrow C$.

	C is true	C is false	Sum$_{row}$
A is true	$sup(r)$=y	$x - y$	$sup(A)$=x
A is false	$m - y$	$n - x - (m - y)$	$sup(\neg A)$= $n - x$
Sum$_{col}$	$sup(C)$=m	$sup(\neg C)$= $n - m$	n

QuantMiner [14], a GA-based algorithm, searches rules with high *fitness function* rules. The fitness function Fitness($\cdot$) is an evaluation measure for a rule, and it is based on the *Gain* measure proposed in [4]: $Gain(A \rightarrow C) = supp(A \wedge C) -$ min_conf $\cdot$ $supp(A)$. The *Gain* value is a measure giving a trade-off between support and confidence. Using x and y in Table 1, we write $Gain(A \rightarrow B) = G(x, y)$, and $G(x, y)$ is also a convex function.

3 Mining Quantitative ARs with IPs from a MRDB

Algorithm 1 shows the outline of our algorithm for mining correlated ARs with interval patterns from a MRDB.

Given a MRDB DB, the user first specifies a *rule template* of the form: $A \rightarrow C$; it specifies conjunctions occurring in the left-hand side and the right-hand side, and the right-hand side contains a single target atom C. The user also specifies values of categorical variables occurring in A and C. In case the values of the categorical variables in the rule template are not given, its possible values will be computed in the algorithm so that each of the categorical variable is instantiated to some value in its domain.

Next, we compute the answersets of A and $A \wedge C$, and we make the *initial association rule* r_{init} of the form: $A, I^{\perp} \rightarrow C$, where $I^{\perp}$ is the *minimal* interval constraint of A. In the aforementioned rule (2), for example, $I^{\perp} = \langle [l_1, u_1], [l_2, u_2] \rangle$, where l_i (u_i) $(i = 1, 2)$ is the minimum (maximum) value of the domain of the attribute Q_i, respectively.

If $r_{init}$is infrequent (i.e., its support $supp(A \wedge C)$ is less than *minsupp*), then exit. Otherwise, we compute the set $\mathcal{R}$ of strong rules with the best χ^2 value on DB, by calling a function MIC$^{+p(\chi^2)}(r_{init}, 0)$.

Algorithm 1: Correlated AR Mining from a MRDB

```
   input  : a MRDB $DB$, $minsupp$, $minconf$.
   output: a set $\mathcal{R}$ of rules $r_b$ with the best $\chi^2$ value on $DB$.
 1 A rule template of the form: $A \rightarrow C$ is specified by the user;  // initial
     step
 2 Compute answersets $answerset(A; DB)$ and $answerset(A \wedge C; DB)$;
     // mining step for categorical attributes
 3 Make an initial association rule $r_{init}$ of the form: $A, I^{\perp} \rightarrow C$;
 4 if $A \wedge C$ is infrequent then  return;
 5 Initialize $\mathcal{R} \leftarrow \emptyset$; $\tau \leftarrow -\infty$, and compute correlated ARs by calling
     $\mathrm{MIC}^{+p(\chi^2)}(r_{init}, 0)$ ;
 6 return $\mathcal{R}$

 7 Function $\mathrm{MIC}^{+p(\chi^2)}$(a rule $r(I)$: $A, I \rightarrow C$, an integer $j$) : a set $\mathcal{R}$ of
     rules with the best $\chi^2$ value on Database $DB$ is
 8     $\mathcal{A} \leftarrow \{\mu_{i,\alpha} \mid \mu_{i,\alpha}$ is applicable to $I$ for some $i \geq j,\ \alpha \in \{l, r\}\}$;
 9     foreach $\mu_{i,\alpha} \in \mathcal{A}$ do
10         $I' \leftarrow \mu_{i,\alpha}(I)$;
11         if $sup(r(I')) < minsup$ or $ub(r(I')) < \tau$ then  continue
12         $I_1 \leftarrow I'^{\square\square}$;
13         if $I_1$ fails the canonicity test then  continue
14         $\tau_1 \leftarrow \chi^2$ value of $r(I_1)$;
15         if $\tau_1 > \tau$ and $conf(r(I')) \geq minconf$ then  $\tau \leftarrow \tau_1$;  $\mathcal{R} \leftarrow \{r(I_1)\}$
16         else if $\tau_1 = \tau$ and $conf(r(I')) \geq minconf$ then  $\mathcal{R} \leftarrow \mathcal{R} \cup \{r(I_1)\}$;
17         call $\mathrm{MIC}^{+p(\chi^2)}(r(I_1), i)$;
18     end
19 end
```

The function $\mathrm{MIC}^{+p(\chi^2)}(r(I), j)$ is essentially the same as the MinIntChange algorithm by Kaytoue *et al.* [8]; the enumeration of closed IPs is done in the same way as the original MinIntChange. Namely, the algorithm generates its direct subsumers whose supports are strictly lower than its support. New interval patterns are generated by applying *minimal changes* to a given interval pattern (line 10). Since a closed interval pattern may be generated several times, we employ the *canonicity test* due to CloseByOne [10] (line 13).

Definition 2 (minimal change). [8]

Let I be an interval pattern of a conjunction A, where $I = \langle [a_1, b_1], \ldots, [a_k, b_k] \rangle$ for some $k \geq 1$.

A *right minimal change* $\mu_{i,r}(I)$ $(1 \leq i \leq k)$ is defined as I', where I' is I with its i-th interval replaced by $[a_i, v]$ such that $v = \max\{x \in V_i \mid x < b_i\}$ and V_i is the set of values which the quantitative variable Q_i will take in a given database. A *left minimal change* $\mu_{i,l}(I)$ is defined dually.

A right minimal change $\mu_{i,r}$ is *applicable* to I if the resulting interval $[a_i, v]$ does not collapse, i.e., $v - a_i > 0$. An *applicable* left minimal change $\mu_{i,l}(I)$ is defined dually. □

MIC$^{+p(\chi^2)}$ incorporates into the original MinIntChange a pruning mechanism (line 11) based on Lemma 1 and a mechanism storing rules with currently best χ^2 value (line 15–16). We then have the following properties:

Theorem 1 (Correctness of Algorithm 1). Let *minsup* be a given minimum support and *minconf* a given minimum confidence. Let DB be a given database. Then,

[**Soundness**] All output rules in $\mathcal{R}$ of Algorithm 1 give the best χ^2 value on DB, and satisfy both *minsup* and *minconf*.

[**Completeness**] Let $r(I)$ be an association rule of the form: $A, I \rightarrow C$ for some interval I. Then, if r gives the best χ^2 value on DB and satisfies *minsup* and *minconf*, then Algorithm 1 outputs a rule r_1 in $\mathcal{R}$ of the form: $A, I_1 \rightarrow C$ such that $I_1 = I^{\square\square}$.

Proof. Since the soundness is rather obvious, we omit its proof.

For the completeness, we have from the assumption and the completeness of MinIntChange that there exists a sequence s of minimal changes from the root to $I_1 = (I)^{\square\square}$ such that I_1 passes the canonicity test, i.e., it is not pruned in line 13. Furthermore, since the closure operator $I'^{\square\square}$ (line 12) does not change its support, the computation corresponding to s is not pranced in line 11, either. Therefore, we have that $r(I_1) \in \mathcal{R}$. □

We note that Algorithm 1 is generic in a sense that it works for another correlation measure, m, by replacing MIC$^{+p(\chi^2)}$ by MIC$^{+p(m)}$, provided that the correlation measure m has a property such as convexity so that it allows us to compute an upper bound $ub(r(I))$ (line 11). Such correlation measures include *information gain*, *gini index* and *Gain*, to mention a few.

Although the proposed pruning method makes the IP search space smaller, the search space becomes larger, when a given database has many quantitative attributes. To handle such cases, we utilize *Super CWC* [15], an off the shelf feature selection algorithm to reduce the number of attributes to use. We will show some experimental results in the following section.

4 Experimental Results

We show in Table 2 some datasets used in our experiments; one is from the UCI Machine Learning [2] and the others are from the CTU Prague Relational Learning Repository [12][3]. We also show in Table 3 rule templates for those datasets used to compute correlated association rules, where Q_i $(i = 1, 2)$ are quantitative variables, while the other variables are categorical ones.

Table 2. Example Databases: † from the UCI Machine Learning Repository and the others from the CTU Prague Repository [12]. #Relations: the number of tables in the database. #Instances: the number of rows in the target table. Size: size in MB.

Database	#Relations	#Instances	Size (MB)	Domain
Mutagenesis	3	188	0.9	Medicine
Financial	8	682	94.1	Finance
Mondial	33	454	3.3	Geography
Heart†	1	270	0.016	Medicine

Table 3. Rule Templates for the Datasets in Table 2.

Database	Rule Template
Mutagenesis	$mol\_Id(X), ind\_1(A, C), eLumo(X, Q_1), logP(X, Q_2) \to active(X)$
Financial	$loan(X), amount(X, Q_1), duration(X, 60), avg\_salary(X, Y, Q_2) \to status(X, C)$
Mondial	$country(X), continent(X, Europe), agri(X, Q_1), serv(X, Q_2) \to christian(X)$
heart	$id(X), sc(X, Q_1), max\_hra(X, Q_2), cp\_t(X, T) \to disease(X, C)$

We have implemented our proposed method by using Java 8 on a PC with an Intel Core i7 processor running at 2.30GHz, 8GB of main memory, working under Windows 7 (64 bit). We have performed the following experiments varying the thresholds *min_sup* at fixed $min\_conf = 0.6$.

Effects of the Pruning Method To see the effects of the pruning method, we present some results for the two datasets (mutagenesis and financial) in Figure 2. The figures (left) show the numbers of strong rules generated in computing correlated rules for the rule templates in Table 3, where those categorical attributes in each rule template take some values in their domains. The figures (right) show the corresponding execution times in milliseconds.

We have observed that the pruning method based on the branch-and-bound heuristics enables us to generate much less rules compared with the naive approach (i.e., without pruning). The execution time of both cases are also reduced accordingly.

Effects of the Number of Quantitative Attributes Next, to see the effects of the number of quantitative variables in mining correlated rules, we consider two more rule templates from the rule template, R_2, for the Mondial dataset in Table 3, by varying the number of quantitative variables from 2 to 4; namely, one is a rule template R_3 obtained by adding $industry(X, Q_3)$ to the antecedent of R_2, while the other rule template, R_4, is obtained similarly by adding $inflation(X, Q_4)$ to the antecedent of R_3.

[2] http://archive.ics.uci.edu/ml/datasets/statlog+(heart).

[3] https://relational.fit.cvut.cz/.

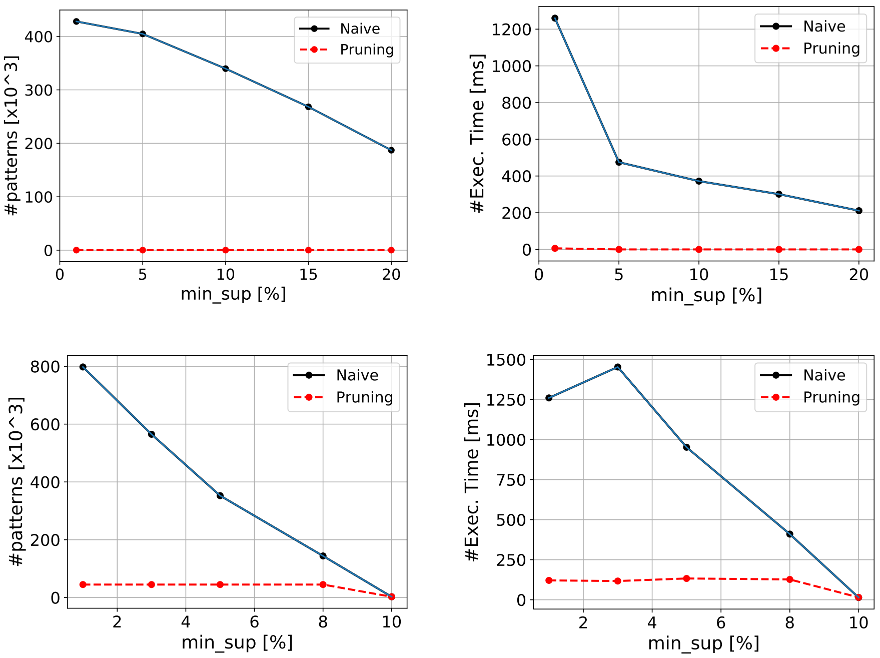


Fig. 2. #(Generated Strong Rules) and Execution Time for Computing Correlated Rules for the Mutagenesis (above) and the Financial (below) Dataset. Rule templates in Table 3 are used.

The number of generated strong rules and the execution time of computing the association rules are shown in Fig. 3. The number of different values in the domain of each quantitative variable Q_i are shown in Table 4. The numbers of possible interval patternss made from the values of Q_1, Q_2, Q_3 and Q_4 thus could become very large. However, the figure shows that the numbers of generated rules using the pruning method only moderately increase. The pruning method thus works well also in this case.

Table 4. Some Statistics of the Mondial Dataset in Table 2. #Values: # of different values in dom(Q_i) ($1 \leq i \leq 4$).

	Q_1 (agri.)	Q_2 (service)	Q_3 (industry)	Q_4 (inflation)
#Values	27	29	30	29

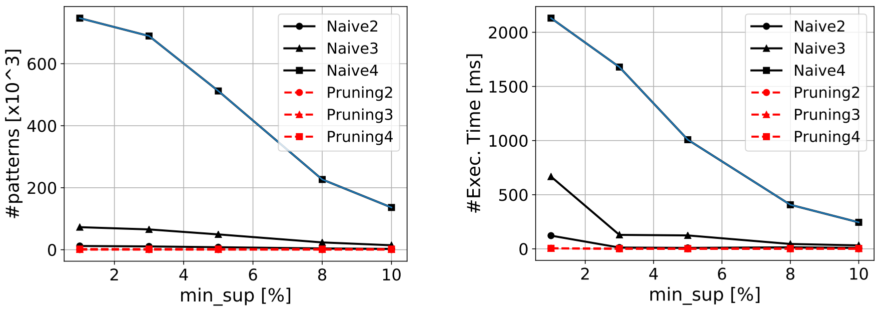


Fig. 3. #(Generated Strong Rules) and Execution Time for Computing Correlated Rules for the Mondial Dataset. *N-i* (resp., *P-i*): the naive (resp., pruning) method for rule R_i with i quantitative attributes ($i = 2, 3, 4$).

The heart dataset in Table 2 contains 13 attributes; among them, 6 attributes take numerical values. In this case, the MinIntChange algorithm generates a large number of CIPs. In fact, we could not obtain outputs of our algorithm within a reasonable time when applying it directly to the complete dataset. We alleviate this problem by first choosing some of the numerical attributes from the dataset by using *Super CWC* mentioned in Sect. 3, and then applying our algorithm to this reduced dataset. Figure 4 shows the numbers of strong rules generated and the execution time in computing correlated rules for an initial association rule.

Table 5 shows some rules with the best χ^2 values obtained by our algorithm as well as the other approaches, i.e., *QuantMiner* [14], an optimization-based approach, and CAIM [9], a data discretization approach. We notice that the rule obtained by our algorithm has the best χ^2 value in this case.

Table 5. Some Obtained Rules: An Example of the Heart Dataset. The Initial Association Rule: $id(X), sc(X, Q_1), max\_hra(X, Q_2), cp\_t(X, 4), \langle Q_1, Q_2\rangle \in \langle I_1, I_2\rangle \rightarrow disease(X, 2)$. $minsupp = 0.1$, $minconf = 0.6$.

	Interval Patterns $\langle I_1, I_2\rangle$	$(supp, conf)$	χ^2 value
Closed IP	$\langle [164.0, 409.0], [71.0, 177.0]\rangle$	(0.33, 0.75)	**81.7**
QuantMiner	$\langle [234.0, 326.0], [122.0, 147.0]\rangle$	(0.09, 0.85)	20.2
CAIM	$\langle [126.0, 407.0], [71.0, 195.0]\rangle$	(0.33, 0.70)	66.0

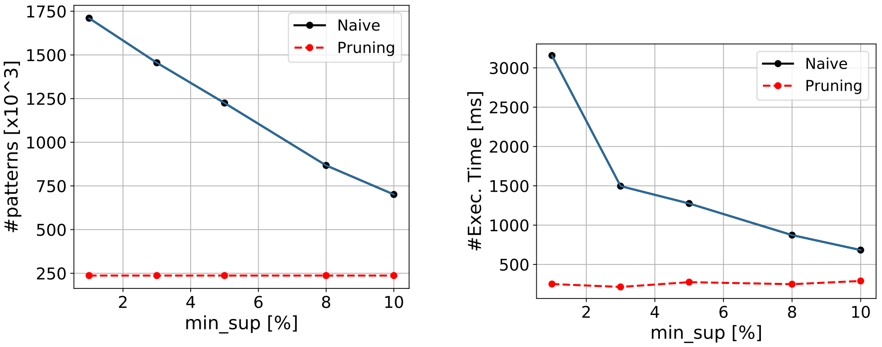


Fig. 4. #(Generated Strong Rules) and Execution Time for Computing Correlated Rules for the Heart Dataset with Initial Association Rule: $id(X)$, $sc(X, Q_1)$, $max\_hra(X, Q_2)$, $cp\_t(X, 4), \langle Q_1, Q_2 \rangle \in \langle I_1, I_2 \rangle \rightarrow disease(X, 2)$. $minsupp = 0.1$, $minconf = 0.6$.

5 Concluding Remarks

In this paper, we have considered the problem of mining relational association rules, especially focusing on the use of closed interval patterns (CIPs) for finding correlated rules with the best χ^2 values. Since the number of mined CIPs increases as the number of attributes and values in the domain of each attribute increases, we have examined the effectiveness of the original MinIntChange algorithm [8] with the pruning technique by Morishita *et al.* on the problem. We have also examined the effectiveness of the use of a feature selection algorithm, *Super CWC*, to reduce the search space of IPs.

Most of the work in the field of MRDM have handled numerical data by using data discretization. To the best of our knowledge, there has been no approach which uses closed interval patterns for mining correlated association rules in multi-relational data.

For future work, we will examine another correlation measure m in $\mathrm{MIC}^{+p(m)}$, since our algorithm is generic and will work for another measure. Since the search space for CIPs is still large in general and the computation of CIPs is costly, we will need some method for reducing the computational time and space to a manageable size.

Acknowledgment The authors would like to thank anonymous reviewers for their useful comments on the previous version of the paper. This work was partially supported by JSPS Grant-in-Aid for Scientific Research (C) 15K00305 and 18K11432.

References

1. De Raedt, L., Ramon, J.: Condensed representations for Inductive Logic Programming. In: Proc. KR'04, pp. 438–446 (2004)
2. Dehaspe, L.: Frequent pattern discovery in first-order logic. Ph.D. thesis, Dept. Computer Science, Katholieke Universiteit Leuven (1998)
3. Dzeroski, S.: Multi-Relational Data Mining: An Introduction. SIGKDD Explorations Newsletter 5(1), 1–16 (2003)
4. Fukuda, T., Morimoto, Y., Morishita, S., Tokuyama, T.: Data mining using two-dimensional optimized association rules: Scheme, algorithms, and visualization. In: Proc. of ACM SIGMOD '96. pp. 13–23 (1996)
5. Ganter, B., Kuznetsov, S.: Pattern Structures and Their Projections. In: ICCS-01, LNCS, vol. 2120, pp. 129–142. Springer (2001)
6. Guyet, T., Quiniou, R., Masson, V.: Mining relevant interval rules. In: Supplementary Proceedings of ICFCA. pp. 79–82 (2017)
7. Han, J., Kamber, M.: Data Mining: Concepts and Techniques. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA (2006)
8. Kaytoue, M., Kuznetsov, S.O., Napoli, A.: Revisiting numerical pattern mining with formal concept analysis. In: International Conference on Artificial Intelligence (IJCAI 2011). pp. 1342–1347 (2011)
9. Kurgan, L.A., Cios, K.J.: CAIM discretization algorithm. IEEE Transactions on Knowledge and Data Engineering 16(2), 145–153 (Feb 2004)
10. Kuznetsov, S.O., Obiedkov, S.A.: Comparing performance of algorithms for generating concept lattices. J. Exp. Theor. Artif. Intell. 14(2-3), 189–216 (2002)
11. Morishita, S., Nakaya, A.: Parallel branch-and-bound graph search for correlated association rules. In: Proc. of ACM SIGKDD Workshop on Large-Scale Parallel KDD Systems. pp. 265–276 (1999)
12. Motl, J., Schulte, O.: The CTU Prague Relational Learning Repository. ArXiv e-prints (Nov 2015)
13. Nagao, M., Seki, H.: On mining quantitative association rules from multi-relational data with FCA. In: Proc. IEEE IWCIA2016. pp. 81–86 (2016)
14. Salleb-Aouissi, A., Vrain, C., Nortet, C.: QuantMiner: A genetic algorithm for mining quantitative association rules. In: IJCAI. pp. 1035–1040 (2007)
15. Shin, K., Kuboyama, T., Hashimoto, T., Shepard, D.: Super-cwc and super-lcc: Super fast feature selection algorithms. In: proceedings of Big Data. pp. 61–67 (2015)
16. Srikant, R., Agrawal, R.: Mining quantitative association rules in large relational tables. In: Proc. of ACM SIGMOD '96. pp. 1–12 (1996)

k-Partite Graphs as Contexts

Alexandre Bazin and Aurélie Bertaux

Le2i, Université Bourgogne Franche-Comté, Dijon, France

Abstract In formal concept analysis, 2-dimensional formal contexts are bipartite graphs. In this work, we generalise the notions of context and concept to graphs that are not bipartite. We then study the complexity of the enumeration and identify the structure of the set of such concepts.

1 Introduction

Formal concept analysis (FCA) is a mathematical framework centered on the notions of formal context (data) and formal concept (significant patterns). Most of the simpler real-life data sets take the form of formal contexts and the interesting patterns are often variations on the theme of formal concepts, making FCA well-suited for applications in any field that deals with data [3,10,6,12]. However, it has its limitations. With the increasing complexity of data, FCA requires extensions and generalisations such as fuzzy or multi-dimensional approaches [2,1,7,13].

Formal contexts in their basic form are binary tables – i.e. bipartite graphs for which a bipartition into independent sets is given. One of the most important generalizations of FCA, Polyadic Concept Analysis (PCA) [13], deals with the same notions of context and concept when said context is an n-uniform[1] n-partite[2] hypergraph – modeling the majority of multidimensional data sets. In PCA, again, an n-partition of the hypergraph is given. This trend can be found in all variants of FCA : the number of dimensions is the size of the data tuples.

We believe that it would be interesting, ultimately, to generalise FCA to n-partite hypergraphs that are not n-uniform in order to create new opportunities of applications involving exotic data. In this work, as a first step toward this goal, we focus on the case of n-partitioned graphs (2-uniform hypergraphs) with $n > 2$. We define the corresponding "concepts", briefly study the complexity of their enumeration and show that they form a complete n-lattice, implying that known algorithms can be used to compute them.

[1] i.e. hypergraph such that all its hyperedges have size n

[2] i.e. the set of graph vertices is decomposed into n disjoint sets such that no two graph vertices within the same set are adjacent

© Dmitry I. Ignatov, Lhouari Nourine (Eds.): CLA 2018, pp. 59–67,
ISBN 978–80–244–5328–6, Department of Computer Science, Palacký University
Olomouc, 2018.

2 Basics

This section briefly presents the basic notions in formal concept analysis and polyadic concept analysis. For a deeper look into the 2-dimensional case, we refer the reader to [5].

2.1 Binary Formal Concept Analysis

Definition 1 *A* (formal) context *is a triple* (S_1, S_2, R) *in which* S_1 *and* S_2 *are sets of what is commonly referred to as* objects *and* attributes *and* R *is a binary relation between objects and attributes representing the fact that an object is* described *by an attribute.*

A formal context is usually represented by a crosstable.

R	a	b	c	d	e
1	×	×			
2		×	×	×	
3		×		×	×
4			×		×
5				×	×

Figure 1. A formal context $(\{1, 2, 3, 4, 5\}, \{a, b, c, d, e\}, R)$

Definition 2 *Let* $\mathcal{C} = (S_1, S_2, R)$ *be a context. A* (formal) concept *of* $\mathcal{C}$ *is a pair* $(E \subseteq S_1, I \subseteq S_2)$ *such that* $E \times I \subseteq R$ *and both* E *and* I *are maximal for this property.*

In other words, a concept is a maximal rectangle full of crosses up to permutation of objects or attributes, also called in graph theory: a full bipartite subgraph or a biclique.

In our Fig. 1 example, $(1, ab)$ and $(23, bd)$ are concepts.

The set of concepts can be ordered by the inclusion relation on both objects and attributes and then forms a complete lattice (i.e. graph of concepts). Every complete lattice is isomorphic to the concept lattice of some context [5].

2.2 Multidimensional Formal Concept Analysis

The notions of formal contexts and concepts have been extensively studied and are successfully used in various fields such as data mining, data analysis, information retrieval, source code error correction, machine learning and for building taxonomies and ontologies [9]. The multidimensional generalization of FCA, polyadic concept analysis [13], has received comparatively less attention but is a promising theoretical as well as applicative field. Let us present here the basics.

Definition 3 *An* n-context *is a tuple* $(S_1, \ldots, S_n, R)$ *in which* S_i, $i \in \{1, \ldots, n\}$, *is a set called a* dimension *and* $R \subseteq \prod_{i \in \{1,\ldots,n\}} S_i$ *is an* n*-ary relation.*

An n-context can be represented by an n-dimensional crosstable.

	a	b	c	a	b	c	a	b	c
1	×	×		×			×		
2	×			×			×	×	
3	×			×		×			×
	α			β			γ		

Figure 2. A 3-context $(\{1, 2, 3\}, \{a, b, c\}, \{\alpha, \beta, \gamma\}, R)$

Definition 4 *Let* $\mathcal{C} = (S_1, \ldots, S_n, R)$ *be an* n*-context. An* n*-concept of* $\mathcal{C}$ *is an* n*-tuple* $(T_1, \ldots, T_n)$ *such that* $T_i \subseteq S_i$, $\prod_{i \in \{1,\ldots,n\}} T_i \subseteq R$ *and there is no* $d \in \{1, \ldots, n\}$ *and* $k \in S_d \setminus T_d$ *such that* $(T_1, \ldots, T_d \cup \{k\}, \ldots, T_n)$ *respects this property.*

In other words, an n-concept is a maximal n-dimensional box full of crosses in $\mathcal{C}$ up to permutations inside dimensions.

In our Fig. 2 example, $(\{1, 2, 3\}, \{a\}, \{\alpha, \beta\})$ and $(\{2\}, \{a, b\}, \{\gamma\})$ are 3-concepts.

The set of all the n-concepts in an n-context, together with the n quasi-orders induced by the inclusion relation on the subsets of each dimension, forms an n-lattice and each complete n-lattice is isomorphic to the concept lattice of an n-context, as stated in the basic theorem of polyadic concept analysis [13].

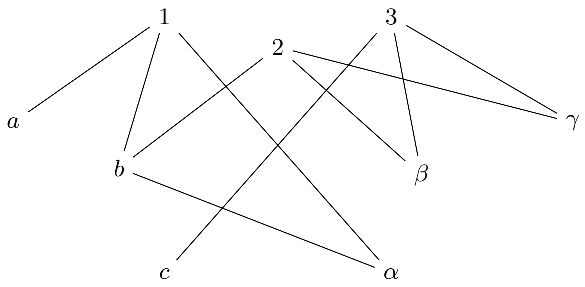


Figure 3. Graph that will be used as running example.

2.3 Graphs

A graph is a pair $G = (V, E)$ in which V is a set of elements called *vertices* and $E \subseteq V^2$ a set of *edges*.

A set $X \subseteq V$ of vertices is a *clique* if there is an edge between any two of its elements. A clique is *maximal* if it is not contained in another clique. An *independent set* is a set of vertices that does not contain any edge. An independent set is *maximal* if it is not contained in any independent set. A *vertex cover* is a set of vertices that contains at least one vertex from every edge. A vertex cover is *minimal* if it does not contain any vertex cover. A (maximal) independent set in a graph G is a (maximal) clique in the complementary graph $\overline{G}$ and reciprocally. The complement of a (maximal) independent set is a (minimal) vertex cover and reciprocally.

We will use $\mathcal{M}(G)$ to denote the set of maximal cliques in a graph G.

A graph $G = (V, E)$ is *k-partite* iff V can be partitioned into k independent sets.

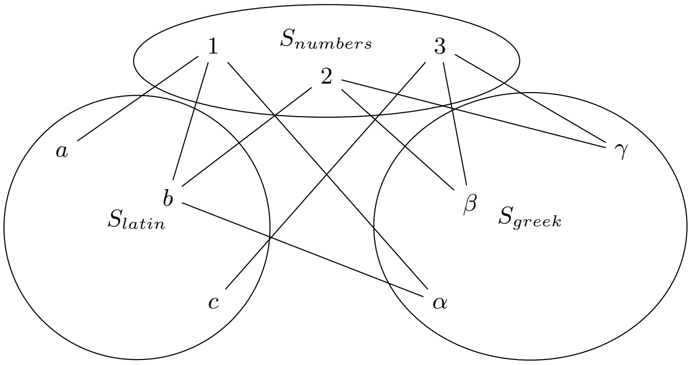


Figure 4. Partition of our example graph into three independent sets $S_{numbers}$, S_{latin} and S_{greek}.

A *complete k-partite graph* is a k-partite graph such that there is an edge between every pair of vertices that do not belong to the same independent set.

In our running example, the subgraphs induced by the vertices sets $\{1, b, \alpha\}$ and $\{1, a, b\}$ are, respectively, complete tripartite and bipartite graphs.

Bidimensional formal contexts (S_1, S_2, R) are bipartite graphs $(S_1 \cup S_2, R)$ for which a bipartition is given. In graph terminology, 2-concepts are thus maximal complete bipartite subgraphs of the context.

3 k-Partite Graphs as Contexts

FCA offers tools to find and manipulate patterns in bipartite graphs. What happens to these patterns and tools when the input graph is not bipartite ?

3.1 Defining the Concepts

Let us start by defining the objects we are looking for. The central patterns in FCA are concepts : maximal complete bipartite subgraphs of the context. When the context is k-partite, a natural generalisation can then be expressed as follows.

Definition 5 *Let $G = (V, E)$ be a graph and $S = (S_1, \ldots, S_k)$ a partition of V into k independent sets. Let $\{j_1, \ldots, j_m\} \subseteq \{1, \ldots, k\}$. An m-$_2$*concept *of (S, E) is a tuple $C = (C_{j_1}, \ldots, C_{j_m})$, $C_{j_x} \neq \emptyset$, $C_{j_x} \subseteq S_{j_x}$, such that $\bigcup_{x \in \{1,\ldots,m\}} C_{j_x}$ induces a maximal complete m-partite subgraph of G and there is no $(C_{j_1}, \ldots, C_{j_m}, C_{j_{m+1}})$ with this property.*

In "m-$_2$concept", the m means that we consider an m-partite graph as "concept" (m dimensions are involved in the pattern) and the $_2$ means the pattern is found in a 2-uniform graph. We have chosen to define them as m-tuples instead of k-tuples with $m \leq k$ in order to avoid having to consider the $m - k$ empty components and confusion with k-concepts from PCA.

We will now suppose, for the remainder of this paper, that our running example is partitioned as in Fig. 4. In this case, $(1, b, \alpha)$ is a 3-$_2$concept and $(1, ab)$ and $(23, \beta\gamma)$ are 2-$_2$concepts. The tuple $(3, c, \beta\gamma)$ is not a 3-$_2$concept because the induced subgraph is complete bipartite, not complete tripartite[3]. The tuple $(1, \alpha)$ is not a 2-$_2$concept because $(1, b, \alpha)$ is a 3-$_2$concept.

When the graph is bipartite **and the partition provided is binary**, the 2-$_2$concepts are the formal concepts with non-empty intents and extents. It is important to note that S_i, $i \in \{1, \ldots, k\}$, is a complete 1-partite subgraph – though (S_i) is not necessarily a 1-$_2$concept.

We will use $\mathcal{T}((S, E))$ to denote the set of m-$_2$concepts, $1 < m \leq |S|$, of a k-partite graph (V, E) together with a partition S of V into k independent sets.

[3] Two sets are considered {3} and {c,$\beta\gamma$} without relations between {c} and {$\beta\gamma$}

Proposition 1 *Let (V,E) be a graph and $S=(S_1,\dots,S_k)$ a partition of V into k independent sets.*

$$\mathcal{T}((S,E)) = \mathcal{M}((V,E\cup X))$$

with $X=\bigcup_{i\in\{1,\dots,k\}}\binom{S_i}{2}$

Proof. In $G=(V,E\bigcup_{i\in\{1,\dots,k\}}\binom{S_i}{2})$, we have that $\forall i\in\{1,\dots,k\}$, S_i is a clique. Let $C=(C_{j_1},\dots,C_{j_m})$ with $C_{j_i}\subseteq S_{j_i}$ be such that $\bigcup_{i\in\{1,\dots,m\}}C_{j_i}$ is a maximal clique in G. By definition, any two vertices $x\in C_{j_a}$ and $y\in C_{j_b}$, $a\neq b$ are neighbours in G. As such, they are neighbours in (V,E) too. Clearly, that makes C an m-partite complete subgraph of (V,E). The maximality property holds from one graph to the other so C is an m-$_2$concept of (V,E).

Let $C=(C_{j_1},\dots,C_{j_m})$ be an m-$_2$concept of (V,E). By definition, any two vertices $x\in C_{j_a}$ and $y\in C_{j_b}$, $a\neq b$ are neighbours in (V,E). As such, they are neighbours in G. As, $\forall i\in\{1,\dots,k\}$, S_i is a clique, $\bigcup_{i\in\{1,\dots,m\}}C_{j_i}$ is a clique in G. The maximality property once again holds from one graph to the other so $\bigcup_{i\in\{1,\dots,m\}}C_{j_i}$ is a maximal clique in G. □

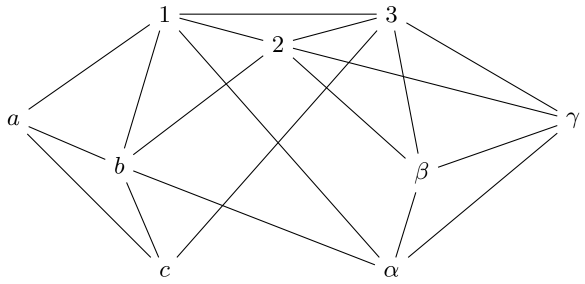


Figure 5. Our example graph with its partitions made into cliques.

This proposition states that m-$_2$concepts are maximal cliques in a graph that can be constructed in polynomial time from the context. This implies that $\mathcal{T}((S,E))$ can be computed from (S,E) in output-polynomial time [11].

3.2 Structuring the Concepts

We now have to characterise the structure of the set $\mathcal{T}((S,E))$. We will show that it forms a k-lattice when put together with the appropriate quasi-orders. The best way to do this is to show that $\mathcal{T}((S,E))$ is isomorphic to the concept k-lattice of a k-context.

Let $\mathcal{K}((S,E))=(S_1\cup\{s_1\},\dots,S_k\cup\{s_k\},R)$ be a k-context such that $s_i\notin S_i$ and

$$(x_1,\dots,x_k)\in R \iff \forall x_i\neq s_i, x_j\neq s_j, \exists e\in E \text{ such that } x_i,x_j\in e$$

Note that, potentially, $x_i = x_j$. In the context $\mathcal{K}((S, E))$ each cross corresponds to a clique of the graph (V,E), including 1-element ones, with the elements s_i representing the fact that a clique does not intersect the set S_i. Figure 6 illustrates the 3-context corresponding to our running example..

Clearly, if $(X_1, \ldots, X_k)$ is a k-concept of $\mathcal{K}((S, E))$, then $\forall i \in \{1, \ldots, m\}$, $s_i \in X_i$.

	a	b	c	s_2	a	b	c	s_2	a	b	c	s_2	a	b	c	s_2
1		×		×									×	×		×
2								×				×		×		×
3								×				×			×	×
s_1		×		×				×				×	×	×	×	×
	α				β				γ				s_3			

Figure 6. The 3-context $(\{1, 2, 3, s_1\}, \{a, b, c, s_2\}, \{\alpha, \beta, \gamma, s_3\}, R)$ corresponding to our running example.

Theorem 1. *Let (V, E) be a graph and S a k-partition of (V, E) into k independent sets. The set of m-$_2$concepts of (S, E), together with the k quasi-orders induced by the inclusion relation on each independent set, forms a k-lattice.*

Proof. Let $(X_1, \ldots, X_k)$ be a k-concept of $\mathcal{K}((S, E)) = (S_1 \cup \{s_1\}, \ldots, S_k \cup \{s_k\}, R)$. By definition, $\prod_{i \in \{1,\ldots,k\}} (X_i \setminus \{s_i\}) \subseteq R$. From the construction of $\mathcal{K}((S, E))$, we get that $\forall x_i \in X_i \setminus \{s_i\}, x_j \in X_j \setminus \{s_j\}$, $\exists e \in E$ such that $x_i, x_j \in e$. This means that the tuple $(X_{j_1} \setminus \{s_{j_1}\}, \ldots, X_{j_m} \setminus \{s_{j_m}\})$, such that the different $X_{j_i} \setminus \{s_{j_i}\}$ are the non-empty components of $(X_1 \setminus \{s_1\}, \ldots, X_k \setminus \{s_k\})$, is an m-$_2$concept of (S, E).

Let $(C_{j_1}, \ldots, C_{j_m})$ be an m-$_2$concept of (S, E). By definition, $\forall A \in \prod_{i \in \{1,\ldots,m\}} C_{j_i}$, $\forall x, y \in A$, $\exists e \in E$ such that $x, y \in e$. As such, the tuple $(X_1, \ldots, X_k)$ such that

$$X_i = \begin{cases} C_i \cup \{s_i\} & \text{if } i \in \{j_1, \ldots, j_m\} \\ \{s_i\} & \text{otherwise} \end{cases}$$

is a k-concept of $\mathcal{K}((S, E))$. □

This implies that algorithms [4,8] for computing n-concepts can be used to compute m-$_2$concepts.

In Fig. 6, the 3-concepts are

$$\begin{array}{cc}
(1s_1, bs_2, \alpha s_3) & (23s_1, s_2, \beta\gamma s_3) \\
(1s_1, abs_2, s_3) & (12s_1, bs_2, s_3) \\
(3s_1, cs_2, s_3) & (123s_1, s_2, s_3) \\
(s_1, abcs_2, s_3) & (s_1, s_2, \alpha\beta\gamma s_3)
\end{array}$$

which yield the m-${}_2$concepts of our running example once the s_i and empty sets are removed.

4 Conclusion

In this paper, we have extended the notions of formal context and concept to graphs that are not bipartitioned in order to allow the handling of a different kind of data. We have shown that, given a k-partition of the graph into independent sets, the set of such m-${}_2$concepts forms a k-lattice. This allows the use of any k-lattice algorithm to compute m-${}_2$concepts.

The next step would be to generalise the notion of n-concept to hypergraphs that are not n-partite n-uniform. This, however, is not as straightforward as m-${}_2$concepts. Indeed, the k-lattice structure of m-${}_2$concepts comes from the fact that a clique with n vertices can freely be converted into 2^n hyperedges (the subsets of vertices). Converting an edge (a, b) into two singletons (a) and (b) does not add complexity. However, converting an hyperedge (a, b, c) into a triangle (a, b), (b, c), (a, c) can potentially create new triangles that do not correspond to existing hyperedges of size 3.

References

1. Belohlavek, R.: What is a fuzzy concept lattice? II. Rough Sets, Fuzzy Sets, Data Mining and Granular Computing pp. 19–26 (2011)
2. Burusco Juandeaburre, A., Fuentes-González, R.: The study of the L-fuzzy concept lattice. Mathware & soft computing. 1994 Vol. 1 Núm. 3 p. 209-218 (1994)
3. Carbonnel, J., Huchard, M., Miralles, A., Nebut, C.: Feature model composition assisted by formal concept analysis. In: ENASE: Evaluation of Novel Approaches to Software Engineering. pp. 27–37. SciTePress (2017)
4. Cerf, L., Besson, J., Robardet, C., Boulicaut, J.F.: Data-peeler: Constraint-based closed pattern mining in n-ary relations. In: proceedings of the 2008 SIAM International conference on Data Mining. pp. 37–48. SIAM (2008)
5. Ganter, B., Wille, R.: Formal Concept Analysis: Mathematical Foundations. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 1st edn. (1997)
6. Kaytoue, M., Kuznetsov, S.O., Napoli, A., Duplessis, S.: Mining gene expression data with pattern structures in formal concept analysis. Information Sciences **181**(10), 1989–2001 (2011)
7. Lehmann, F., Wille, R.: A triadic approach to formal concept analysis. Conceptual structures: applications, implementation and theory pp. 32–43 (1995)
8. Makhalova, T., Nourine, L.: An incremental algorithm for computing n-concepts. In: Formal Concept Analysis for Knowledge Discovery. Proceedings of International Workshop on Formal Concept Analysis for Knowledge Discovery (FCA4KD 2017), Moscow, Russia, June 1, 2017. CEUR-WS. org (2017)
9. Škopljanac Mačina, F., Blašković, B.: Formal concept analysis – overview and applications. Procedia Engineering **69**, 1258 – 1267 (2014). https://doi.org/https://doi.org/10.1016/j.proeng.2014.03.117, 24th DAAAM International Symposium on Intelligent Manufacturing and Automation, 2013

10. Snelting, G.: Software reengineering based on concept lattices. In: Software Maintenance and Reengineering, 2000. Proceedings of the Fourth European. pp. 3–10. IEEE (2000)
11. Tsukiyama, S., Ide, M., Ariyoshi, H., Shirakawa, I.: A new algorithm for generating all the maximal independent sets. SIAM Journal on Computing **6**(3), 505–517 (1977)
12. Valtchev, P., Missaoui, R., Godin, R.: Formal concept analysis for knowledge discovery and data mining: The new challenges. In: ICFCA. vol. 2961, pp. 352–371. Springer (2004)
13. Voutsadakis, G.: Polyadic concept analysis. Order **19**(3), 295–304 (2002)

How Hierarchies of Concept Graphs Can Facilitate the Interpretation of RCA Lattices?[⋆]

Sébastien Ferré and Peggy Cellier

Univ Rennes, CNRS, INSA, IRISA
Campus de Beaulieu, 35042 Rennes cedex, France
ferre@irisa.fr, peggy.cellier@irisa.fr

Abstract. Relational Concept Analysis (RCA) has been introduced in order to allow concept analysis on multi-relational data. It significantly widens the field of application of Formal Concept Analysis (FCA), and it produces richer concept intents that are similar to concept definitions in Description Logics (DL). However, reading and interpreting RCA concept lattices is notoriously difficult. Nica *et al* have proposed to represent RCA intents by cpo-patterns in the special case of sequence structures. We propose an equivalent representation of a family of RCA concept lattices in the form of a hierarchy of concept graphs. Each concept belongs to one concept graph, and each concept graph exhibits the relationships between several concepts. A concept graph is generally transversal to several lattices, and therefore highlights the relationships between different types of objects. We show the benefits of our approach on several use cases from the RCA litterature.

Keywords: Formal Concept Analysis, Relational Concept Analysis, Data Mining, Concept Graph

1 Introduction

Many domains produce multi-relational data. For example, in the health domain, one can have patients taking drugs, drugs giving some symptoms and interacting with other drugs, doctors taking care of patients and prescribing drugs to patients, etc. In order to extract knowledge from that kind of data, many data mining techniques, such as Formal Concept Analysis (FCA) [8], require the flattening of multi-relational data but this results in loss of structural information, and a more difficult interpretation of discovered patterns. It is therefore desirable to have direct methods for multi-relational mining [2]. Several generalizations of FCA have been proposed to handle relational data: Power Context Families [10], Relational and Logical Concept Analysis [7], Relational Concept Analysis (RCA) [11], and Graph-FCA [5]. RCA has so far been the most frequently used approach with applications in health [11,9] or model driven engineering [4].

[⋆] This research is supported by ANR project PEGASE (ANR-16-CE23-0011-08).

© Dmitry I. Ignatov, Lhouari Nourine (Eds.): CLA 2018, pp. 69–80,
ISBN 978–80–244–5328–6, Department of Computer Science, Palacký University
Olomouc, 2018.

An important issue for the effective use of RCA is the interpretation of its outputs. Indeed, RCA produces not one but several concept lattices, and the intent of each concept may depend on the intent of other relationally-related concepts, recursively. To the best of our knowledge, there has been only one proposal to automatically extract and graphically represent the relational patterns that are buried into concept intents: Nica's cpo-patterns [9]. However, it has several restrictions. First, it is defined only for sequential data. Second, it generates cpo-patterns only for the concepts of one chosen lattice. Third, it generates a cpo-pattern for each concept of the chosen lattice, missing potential factorizations between patterns and thus interesting information between patterns.

In this paper, we propose a novel and generic graphical representation of RCA outputs that emphasizes the relational patterns. We call it *hierarchy of concept graphs*. It has the following good properties. First, it makes no assumption on the context family, and can therefore handle all kinds of graph structures, not only sequences. Second, it is at the same time a *complete* and *non-redundant* representation of the family of concept lattices, and does not require to choose one concept lattice as starting point. Third, it offers a better balance in the display between generalization ordering (lattice edges), and relationships (relational attributes). Fourth, it clusters concepts into *concept graphs*, and hence produce a coarser-grained representation. Fifth, it can be efficiently computed from the concept lattices, in linear time.

Section 2 discusses related work. Section 3 shortly recalls the main definitions of RCA. Section 4 introduces our representation of RCA outputs as a hierarchy of concept graphs, illustrates it on a reference example of RCA, and discusses its properties. Section 5 evaluates our approach on a few use cases, and discusses the impact of representation choices. Section 6 concludes and draws perspectives.

2 Related Work

Several generalizations of FCA have been proposed to handle relational data. Power Context Families [10] has a formal context for each relation arity, i.e. a context of objects, a context of couples of objects, a context of triples of objects, etc. A concept lattice is computed for each context, independently of other contexts. The resulting concepts are used as a vocabulary of types and relations to build *concept graphs* that are similar to Conceptual Graphs [12,1]. Relational and Logical Concept Analysis [7] takes as input a power context family limited to unary and binary relations but extended to complex logical descriptions. It generates a single concept lattice where concept intents combine both unary and binary descriptors, and where the labeling of the concept lattice is extended with relationships between concepts. Relational Concept Analysis (RCA) [11] takes as input a power context family limited to unary and binary relations. In practice, the unary context is split in several unary contexts, one for each type of object. RCA generates a concept lattice for each type of object, where concept intents are sets of classical attributes and relational attributes. The latter represent relationships to other concepts in the concept lattice family.

Graph-FCA [5] takes as input a power context family without restriction on arities. It generates a set of graph patterns where each node represents a unary concept, each pair of nodes represents a binary concept, etc. For each concept arity, the set of all concepts forms a concept lattice.

The above shows that there are two kinds of representations of the results: concept lattices and concept graphs. They complement each other: concept lattices emphasizes the generalization ordering between concepts, while concept graphs emphasize the relationship patterns between objects in data. In RCA, the native representation is made of concept lattices, and the relationship patterns are only indirectly accessible through relational attributes. Recently, Nica *et al* [9] have proposed a solution to combine concept lattices with graph patterns. However it is not a general solution for RCA because of the limitations already discussed in the introduction.

3 Relational Concept Analysis (RCA)

We here recall the definitions of *context family* and *lattice family* in RCA. We therefore focus on the input and output of RCA, and we ignore the methodology and algorithms that are used to compute the concept lattices from the context family. Indeed we are here concerned with the graphical representation of RCA lattices rather than on their computation. A detailed presentation of RCA is available in previous papers, in particular [11]. The input data of RCA is called a *relational context family* (RCF). In words, it is a collection of formal contexts, one for each kind of objects, together with a collection of binary relations going from the objects of one context to the objects of the same or another context.

Definition 1. *A* Relational Context Family *(RCF) is a pair* $(\boldsymbol{K}, \boldsymbol{R})$ *where:*

- $\boldsymbol{K} = \{\mathcal{K}_i\}_{i=1..n}$ *is a set of contexts* $\mathcal{K}_i = (O_i, A_i, I_i)$*, and*
- $\boldsymbol{R} = \{r_k\}_{k=1..m}$ *is a set of relations* r_k *where* $r_k \subseteq dom(r_k) \times ran(r_k)$*, and* $dom(r_k), ran(r_k) \in \{O_i\}_{i=1..n}$ *are respectively the* domain *and* range *of* r_k*.*

As a running example, we reuse the RCF defined in [11] about pharmacovigilance of AIDS patients and drugs: $(\{\mathcal{K}_p, \mathcal{K}_d\}, \{takes, itb, iw\})$. Context $\mathcal{K}_p$ describes 4 patients in terms of age, gender, and observed Adverse Drug Reactions (ADR) (14 attributes). Context $\mathcal{K}_d$ describes 6 drugs in terms of active molecule, and expected ADR (16 attributes). Relation *takes* relates patients to the drugs they have taken. Relation *itb* ("is taken by") is the inverse of *takes*. Relation *iw* ("interacts with") relates couples of drugs that interact with each other (it is symmetric).

Given an RCF, the output of RCA is a collection of concept lattices, one for each context of the RCF. The relations of the RCF are taken into account by repeatedly applying a mechanism of relational scaling on each context and its concept lattice, until convergence is reached. This leads to the introduction of *relational attributes* that express relational constraints, and contribute to the formation of concepts (see [11] for more details).

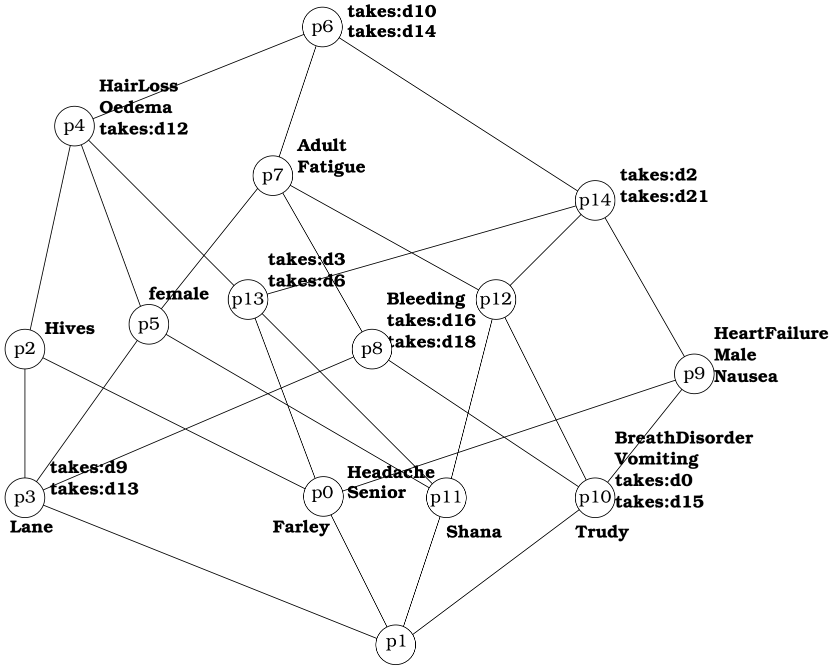


Fig. 1. The relational concept lattice of patients $\mathcal{L}_p$ (reduced labelling)

Definition 2. *Let* $(\boldsymbol{K}, \boldsymbol{R})$ *be a RCF. The* Relational Concept Lattice Family *(RCLF) is a set of concept lattices* $\boldsymbol{L} = \{\mathcal{L}_i\}_{i=1..n}$*, one for each context* $\mathcal{K}_i$*. Each concept* c_i *in* $\mathcal{L}_i$ *is a pair* (X, Y) *where:*

- $X \subseteq O_i$ *is the extent of the concept, and*
- Y *is the intent of the concept, and contains attributes in* A_i *and* relational attributes *in the form* $\rho\ r : c_j$ *where* $\rho \in \{\exists, \forall\exists, ...\}$ *is a scaling operator,* $r \in \boldsymbol{R}$*,* $dom(r) = O_i$*,* $ran(r) = O_j$*, and* $c_j \in \mathcal{L}_j$*.*

In this paper, we only consider existential scaling ($\rho = \exists$) even though our approach is applicable to other scaling operators. Figure 1 shows the concept lattice $\mathcal{L}_p$ of patients, and Figure 2 the concept lattice $\mathcal{L}_d$ of drugs. Both are represented with reduced labelling, i.e. each object/attribute appears only once. Object labels are placed below the concept, while attribute labels are placed above and on the right of the concept. Each relational attribute $\exists r : c_j$ is displayed as $r : c_j$.

The reading and interpretation of RCA lattices is notoriously difficult. The main reason is probably that reading the intent of a concept requires not only to traverse the lattice upward, as in FCA, but also to follow relationships to other concepts through the relational attributes. For example, concept $p7$ groups the adult patients who have fatigue, and take a drug in concept $d10$ and a drug in concept $d14$. Concept $d10$ groups the drugs for which diarrhea is expected, and

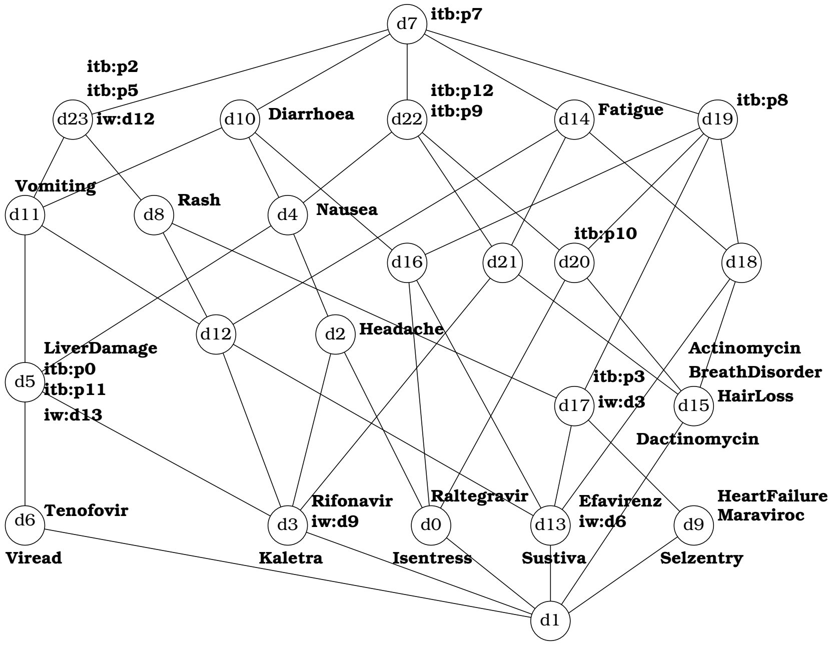


Fig. 2. The relational concept lattice of drugs $\mathcal{L}_d$ (reduced labelling)

which are taken by patients in concept $p7$. Loops in the exploration of the intent, like in that example, lead to circular definitions of concepts, and contribute to the difficulty of interpretation.

4 Hierarchies of Concept Graphs

Our objective is to facilitate the reading of the intent of RCA concepts, in order to facilitate their interpretation. The first idea is to display the different lattices side-by-side, and to materialize each relational attribute $r : c_2$ on a concept c_1 as a *relational edge*, i.e. a labeled and directed edge $c_1 \xrightarrow{r} c_2$. However, the graphical representation becomes denser and even less readable; and its structure is dominated by the lattice structures at the cost of elongated relational edges. A better balance between lattice edges and relational edges is desirable. The second idea is to identify *relational structures* as subsets of interrelated concepts from different lattices, and to use them as building blocks in the graphical representation. We propose to define those relational structures as the Strongly Connected Components (SCC) [3] of the *dependency graph* between concepts. The intuition behind that dependency graph is that the intent of a concept depends on its ancestors in the lattice, and also on the target concepts of relational attributes.

Dependency Graph. We first define the notion of concept dependency and then the dependency graph of a RCLF.

Definition 3. *Let $\boldsymbol{L}$ be a RCLF and c_1, c_2 be two concepts in $\boldsymbol{L}$. Concept c_1 depends on concept c_2, denoted by $c_1 \rightarrow c_2$, if:*

- *c_2 is a parent concept of c_1 ($c_1 \leq c_2$),*
- *or c_1 is labeled by a relational attribute $\rho\, r : c_2$.*

Definition 4. *Let $\boldsymbol{L}$ be a RCLF. The* dependency graph *of $\boldsymbol{L}$ is the directed graph $G_{\boldsymbol{L}} = (V, E)$ where:*

- *V is the set of all concepts of all lattices in $\boldsymbol{L}$ except bottom concepts,*
- *and $E = \{(c_1, c_2) \mid c_1, c_2 \in V$ and $c_1 \rightarrow c_2\}$.*

SCCs as Concept Graphs. From there, a SCC of $G_{\boldsymbol{L}}$ is a maximal set of concepts (possibly from several lattices) where each concept has a dependency path to all other concepts in the SCC for the definition of its intent. SCCs are used to define *concept graphs*, which are the building blocks of our graphical representation.

Definition 5. *A* concept graph *is the subgraph of the lattice family (enriched with relational edges) that is induced by a SCC of $G_{\boldsymbol{L}}$. It therefore mixes concepts from several lattices, and both lattice edges and relational edges.*

In Figure 3, each rounded box contains a concept graph (G1-G6). Nodes are concepts from the two RCA lattices in Figures 1 and 2 (same id, same labels). Relational edges (arrows) replace the relational attributes to graphically represent the relational dependencies. Lattice edges that cross concept graph boundaries are displayed as dotted lines to keep the graph light, and to emphasize the concept graphs over the global lattice structures. It is notable in this example that no relational edge crosses concept graph boundaries. This is because in the context family each relation either has an inverse relation (e.g., *takes* and *itb*) or is a symmetric relation (e.g., *iw*).

Furthermore, it is known that the SCCs of a graph form a directed acyclic graph, where $SCC_1 \rightarrow SCC_2$ if any concept in SCC_1 depends on any concept in SCC_2. The concept graphs of a RCLF can therefore be organized into a *hierarchy of concept graphs*. For example, concept graph G2 is a child of concept graph G1 in Figure 3 because several concepts in G2 have lattice edges (dotted lines) to concepts in G1: e.g., from p4 to p6, or from d12 to d14. Those hierarchical relationships can be seen as a complex version of the lattice edges, combining several lattice edges across different lattices.

Interpretation. We here give a short interpretation of the hierarchy of concept graphs in Figure 3. G1 represents the most general pattern between patients and drugs. It shows that all patients take drugs expected to give fatigue and diarrhea, and that all drugs are taken by an adult with fatigue. G2-4 are specializations of G1. For example, G4 specializes G1 to patients with bleeding. G2 specializes G1 to patients with hairloss and oedema, which all take drugs giving vomiting and rash, which are taken by a patient with hives, and by a female patient. G5 and G6 represent patterns that are specific to individual patients and drugs,

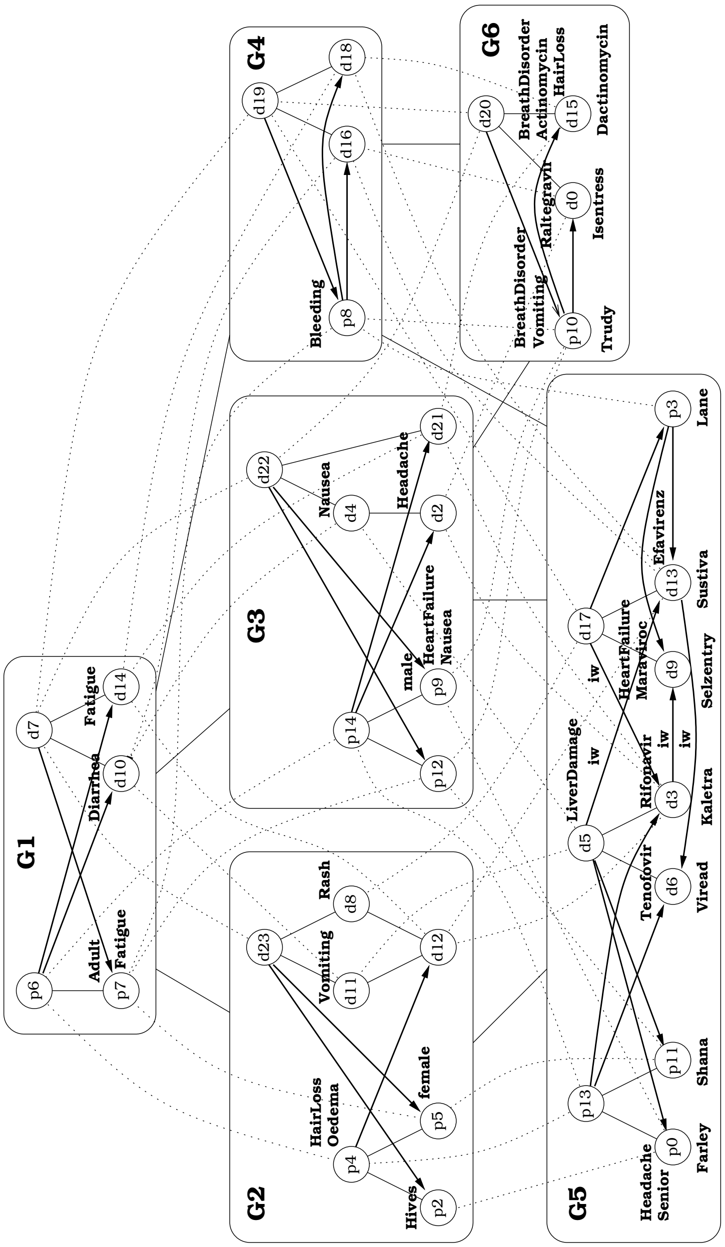


Fig. 3. Hierarchy of concept graphs: arrows represent relational attributes (*takes* from patients to drugs, *itb* from drugs to patients, *iw* between drugs), dotted lines represent lattice edges across concept graphs

and are therefore less interesting from a data-mining perspective. G5 shows that some patients (p13) take drugs (d6 and d3) that cause liver dammage (d5) and are in interaction with drug Sustiva (d13) which is taken by patient Lane (p3).

Discussion. The hierarchy of concept graphs has a number of good theoretical properties. First, it is a *complete* representation because it keeps all concepts and edges from the concept lattices. It is also parcimonious in that it does not duplicate any concept, edge, or label. Second, it is more *readable* because it displays relation attributes as relation edges, and because its layout offers a better balance between lattice edges and relational edges. Moreover, when the many concepts are clustered in a small number of concept graphs, the RCLF can be read at a higher level of granularity. Third, it is *efficient* to compute because the SCCs can be extracted in time linear with the size of $G_{\boldsymbol{L}}$, and hence in the cumulated size of lattices in $\boldsymbol{L}$.

5 Use Cases

In this section we present two use cases in order to compare graph concepts with results of Graph-FCA and cpo-patterns from [9]. The first one describes the royal family. The second use case is about flu patients and medical examinations.

5.1 Royal Family: Genealogical Data

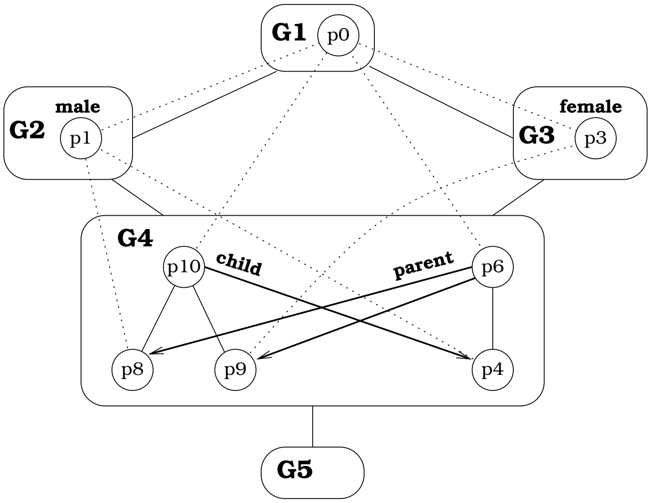


Fig. 4. Hierarchy of concept graphs about the royal family (G5 not detailed)

The first use case is the one used for Graph-FCA [6]. It describes a subset of the British royal family: Charles, Diana, William, Harry, Kate, George, and Charlotte. The power context family uses two attributes (*male* and *female*), and two relations (*parent* and its inverse *child*). Note that the relations have the same object type as domain and range: people. RCA produces one lattice containing 19 concepts. Figure 4 shows the hierarchy of concept graphs obtained from that lattice with our approach. Concepts are clustered in 5 concept graphs. G5 is not detailed because it contains very specific concepts, and hence does not bring new knowledge. The hierarchy of concept graphs enables to reach the following interpretations for each concept:

p0 people	p8 male parents (fathers)
p1 male people (men)	p9 female parents (mothers)
p3 female people (women)	p6 people with a father and mother (children)
p10 people with a child (parents)	p4 male children (sons)

The concept of “daughters” has a single instance (Charlotte), and is found in G5, a specialization of G4. The relation from p10 (parents) to p4 (sons) shows that, in the context, every parent has a son but not necessarily a daughter. The relations froms p6 to p8 and p9 shows that, in the context, every child who has a known father also has a known mother, and reciprocally. Concept graph G4 exhibits the relational pattern of a nuclear family, relating children to their father and mother as parents, with the specificity in this context that all parents have a son.

Comparing those results to Graph-FCA, it is interesting to note that the graph patterns of Graph-FCA are equivalent to the RCA concept graphs, up to a few representation changes. Graph-FCA patterns only represent relational edges, not lattice edges. The generalization ordering between concepts and patterns is therefore not explicitly represented. In Graph-FCA the use of inverse relations is implicit so that the *child* relation is redundant with the *parent* relation. Note that Graph-FCA also defines n-ary concepts such as “couple” or “sibling”. The equivalence with Graph-FCA on this example must not be generalized. In fact, it does not hold on the running example about patients and drugs.

5.2 Medical Histories and Comparison to cpo-Patterns

The second use case is the one used for cpo-Patterns [9]. It describes flu patients through their symptoms, their viral tests and their medical examinations. The specificity of that dataset is the sequentiality of the data. For instance for patient $p1$ we know that a viral test on 28/09 is preceded by a medical examination on 26/09 which is also preceded by another medical examination on 25/09. The power context family uses 6 symptom attributes (COUGHmoderate, FEVERmoderate, ?moderate, COUGHhigh, FEVERhigh, ?high) and two relations: RME-ipb-ME (sequential relation between medical examinations), RVT-ipb-ME (sequential relation from viral tests to medical examinations). It describes five viral tests and ten medical examinations. RCA produces two lattices. The viral test lattice contains 12 concepts and the medical examination lattice contains 18 concepts.

Figure 5 shows the hierarchy of concept graphs obtained from those lattices with our approach. We note that it is a special case, indeed each concept is a

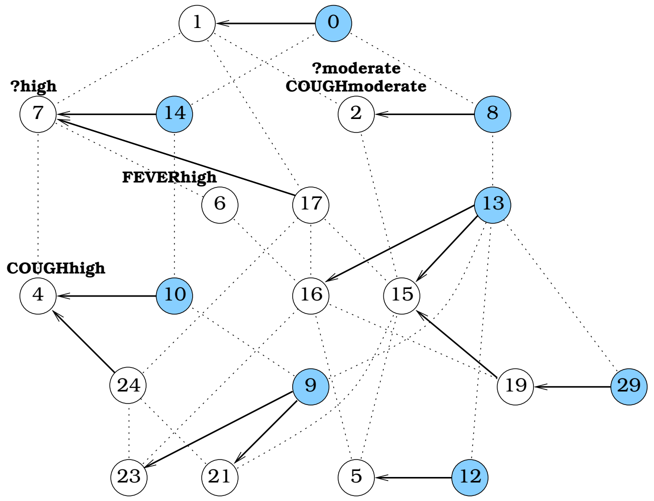


Fig. 5. Hierarchy of concept graphs about medical histories. Each concept is a concept graph on its own. Greyed concepts are about viral tests, others are about medical examinations. Only concepts with support greater than one are shown.

concept graph by its own. It is due to the fact that the relation ”is-preceded-by” has no inverse relation, and forms no cycle because of its sequential nature. It is thus impossible to find more than one concept in a strongly connected component. The greyed concepts are the concepts from the viral test lattice, other concepts come from the medical examination lattice. In the graph only concepts with support greater than one are shown. We can read in the hierarchy that all viral tests (top concept 0) are preceded by a medical examination (top concept 1). That relational pattern has two specialisations. The first one where the symptom during the examination is moderate cough (concepts 2 and 8). The second one where there is a high symptom (concepts 7 and 14). We can also note that parts of sequential patterns are shared by several concepts. For instance, concept 24 and 10 are preceded by a medical examination with high cough (concept 4). In fact, concept 9 specializes concept 10 by inserting between the viral test and the high cough (concept 4) two additional medical examinations (concepts 21 and 23), which are themselves specialisations of concept 24. It highlights the overlaps between sequential patterns.

We have also conducted experiments when considering the inverse relation of ”is-preceded-by” (ipb), i.e. ”is-followed-by” (ifb). The power context family is thus extended with two relations: RME-ifb-ME and RME-ifb-VT. RCA still produces two lattices but the hierarchy of concept graphs is different. Indeed, eleven concept graphs are extracted. Each of them contains several concepts and only one viral test concept. Figure 6 shows an excerpt of the hierarchy of concept graphs with those inverse relations. For the sake of readability and compactness, we modified the representation of concept graphs in two ways: (a) only the most

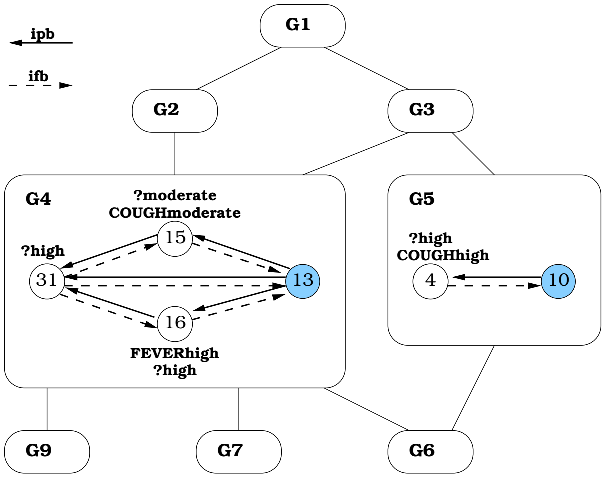


Fig. 6. Hierarchy of concept graphs about medical histories with inverse relation *ifb*. Only concept graphs whose concepts have their support greater than one are shown.

specific concepts of a concept graph are kept, and (b) the full intent of those concepts is shown, instead of the reduced intent, so that each concept graph can be read in isolation. Eight concept graphs among the eleven are shown and only two of them (G4 and G5) are detailled in the figure. Concept graph G4 can be read as ”a viral test (concept 13) preceded by a medical examination with moderate cough (concept 15) and a medical examination with high fever (concept 16) and both of them are preceded by a medical examination with a high symptom (concept 31)”.

The interesting result is that the eleven concept graphs match exactly the eleven cpo-patterns extracted by [9]. However, there are some differences in the display of the patterns. Indeed, in order to compute the strongly connected components, the inverse relations have to be added and they appear in the result. For instance between concepts 13 and 15 there are two arrows, one in each direction, because the relation is-followed-by (ifb) is the inverse relation of is-preceded-by (ipb). In the same vein, ifb and ipb are transitive relations, and thus some arrows are redundant. For example, the arrows between concepts 13 and 31 can be deduced from the paths through concepts 15, and 16 by transitivity. In [9], the representation was specialized for sequential data, and so that kind of redanduncies were avoided. On the contrary, our approach is general and allows to take into account any kind of relations without any assumption on them. In order to avoid those redundancies the description language of power context families should be modified in order to add a way to specify relation properties (e.g., is transitive, is symmetric, has an inverse), and then it should be taken into account when computing and displaying the concept graphs. The same can be said for the redundancy on attributes. Indeed, when looking at the intent of concept 15, we can note the redundancy between ”?moderate” and

"COUGHmoderate". It is due to the conceptual scaling used on the symptom attributes. By taking into account the hierarchy between attributes, the display of the concept graphs can be simplified without loosing information.

6 Conclusion and Perspectives

We have proposed a novel and general representation of RCA concept lattices, called *hierarchy of concept graphs*, in order to facilitate their interpretation. The key idea is to exhibit relational patterns by having a better balance in the display between lattice edges and relational edges. Each concept graph clusters a set of concepts (from different lattices) whose intents are mutually dependent, and exhibits a relational pattern. Concept graphs are organized into a hierarchy so that generalization ordering between concepts is lifted to concept graphs. As future work, we plan to study the impact of relation properties (e.g., inverse, transitivity) on the trade-off between the number of concept graphs and the size of each concept graph. It will also be necessary to develop tools for the dynamic visualization of large hierarchies of concept graphs, *à la* Conexp[1].

References

1. Chein, M., Mugnier, M.L.: Graph-based knowledge representation: computational foundations of conceptual graphs. Advanced Information and Knowledge Processing, Springer (2008)
2. Džeroski, S.: Relational Data Mining, pp. 887–911. Springer US, Boston, MA (2010), https://doi.org/10.1007/978-0-387-09823-4_46
3. Even, S.: Graph algorithms. Cambridge University Press (2011)
4. Falleri, J.R., Arévalo, G., Huchard, M., Nebut, C.: Use of model driven engineering in building generic fca/rca tools. In: CLA. vol. 7, pp. 225–236 (2007)
5. Ferré, S.: A proposal for extending formal concept analysis to knowledge graphs. In: Baixeries, J., Sacarea, C., Ojeda-Aciego, M. (eds.) Int. Conf. Formal Concept Analysis (ICFCA). pp. 271–286. LNCS 9113, Springer (2015)
6. Ferré, S., Cellier, P.: Graph-FCA in practice. In: Haemmerlé, O., et al. (eds.) Int. Conf. Conceptual Structures (ICCS). pp. 107–121. LNCS 9717, Springer (2016)
7. Ferré, S., Ridoux, O., Sigonneau, B.: Arbitrary relations in formal concept analysis and logical information systems. In: ICCS (2005)
8. Ganter, B., Wille, R.: Formal Concept Analysis: Mathematical Foundations. Springer-Verlag New York. (1999)
9. Nica, C., Braud, A., Dolques, X., Huchard, M., Le Ber, F.: Extracting hierarchies of closed partially-ordered patterns using relational concept analysis. In: Int. Conf. Conceptual Structures. pp. 17–30. Springer (2016)
10. Prediger, S.: Simple concept graphs: A logic approach. In: Int. Conf. Conceptual Structures. pp. 225–239. LNCS 1453 (Aug 1998)
11. 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)
12. Sowa, J.: Conceptual structures. Information processing in man and machine. Addison-Wesley, Reading, US (1984)

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

Cover Problems, Dimensions And The Tensor Product Of Complete Lattices

Christian Jäkel[1] and Stefan E. Schmidt[2]
Technische Universität Dresden, 01062, Dresden, Germany
[1]christian.jaekel(at)tu-dresden.de
[2]midt1(at)msn.com

Abstract. In this article, we analyze different dimensional concepts of complete (ortho)lattices and their tensor products. The determination of these dimensions can be translated to certain set cover problems and the cardinal product of the complementary underlying formal contexts. To treat this cover problems in a unified manner, we take a more universal approach via the general set cover problem and its product. This yields a sufficient condition for the multiplicativity of various lattice dimensions with respect to the tensor product of complete lattices.

Keywords: formal concept analysis, tolerance relation, cardinal product, tensor product, order dimension, rectangle cover, square cover, block cover.

1 Motivation

The *order* 2*-dimension* of a complete lattice $\mathbb{L} := (L, \leq)$, $\dim_2(\mathbb{L})$, is the smallest n such that an order embedding, that is an order preserving and reflecting map, from $\mathbb{L}$ to the powerset of an n-element set $\mathfrak{P}(\underline{n})$ exists. This can be seen as a measure of $\mathbb{L}$'s "complexity" with respect to set representations.

For two complete lattices $\mathbb{L}_1$ and $\mathbb{L}_2$ with $\dim_2(\mathbb{L}_1) = m$ and $\dim_2(\mathbb{L}_2) = n$ the tensor product, $\mathbb{L}_1 \otimes \mathbb{L}_2$, admits an order embedding to $\mathfrak{P}(\underline{mn})$[1]. Hence, $\dim_2(\mathbb{L}_1 \otimes \mathbb{L}_2)$ is less or equal to $\dim_2(\mathbb{L}_1)\dim_2(\mathbb{L}_2)$. Equality would be an analogue to vector spaces where $\dim(\mathbb{V}_1 \otimes \mathbb{V}_2) = \dim(\mathbb{V}_1)\dim(\mathbb{V}_2)$ holds. In the case of complete lattices, this is also a question whether the "complexity" of $\mathbb{L}_1 \otimes \mathbb{L}_2$ always grows in the same way as the one of $\mathfrak{P}(\underline{mn}) \cong \mathfrak{P}(\underline{m}) \otimes \mathfrak{P}(\underline{n})$ does. But, it turns out that the tensor product is generally not multiplicative with respect to the order 2-dimension.

In this paper we will analyze a sufficient condition when multiplicativity holds. This will be achieved by studying a set cover problem which is equivalent to the determination of the order 2-dimension. In Section 4 we will take a more universal approach and present a general result about set cover problems and their product. This will be applied to different notions of dimension in formal concept analysis (Section 2) and dimensional concepts of tolerance spaces (Section 3). The latter one have interpretations in graph theory, which we will elaborate on too.

[1] This fact is a consequence of Theorem 2, as we will show in Section 5.

© Dmitry I. Ignatov, Lhouari Nourine (Eds.): CLA 2018, pp. 81–92,
ISBN 978–80–244–5328–6, Department of Computer Science, Palacký University
Olomouc, 2018.

2 Basics Of Formal Concept Analysis

In this section, we will provide the facts from formal concept analysis that we will use in the sequel. If not mentioned otherwise, all results can be found in [7].

A *formal context* is a triple $\mathbb{K} = (G, M, I)$, where the *incidence* $I \subseteq G \times M$ is a binary relation between finite sets. For $A \subseteq G$ and $B \subseteq M$, we define two derivation operators:

$$A^I := \{m \in M | \forall a \in A : (a, m) \in I\} = \bigcap_{a \in A} \{a\}^I,$$
$$B_I := \{g \in G | \forall b \in B : (g, b) \in I\} = \bigcap_{b \in B} \{b\}_I.$$

If $A^I = B$ and $B_I = A$, the pair (A, B) is called a *formal concept* and the cartesian product $A \times B$ is a *maximal rectangle* of $\mathbb{K}$. The set of all formal concepts of $\mathbb{K}$ is denoted by $\mathfrak{B}(\mathbb{K})$ and defines the *concept lattice* $\underline{\mathfrak{B}}(\mathbb{K})$, via the order $(A_1, B_1) \leq (A_2, B_2) :\Longleftrightarrow A_1 \subseteq A_2$. The *complementary context* is defined as $\mathbb{K}^c = (G, M, I^c) := (G, M, (G \times M) - I)$. Furthermore, two special formal concepts of importance are the *object concepts* $\gamma(g) := (\{g\}_I^I, \{g\}^I)$ and *attribute concepts* $\mu(m) = (\{m\}_I, \{m\}_I^I)$. It holds that:

$$gIm \Longleftrightarrow \gamma(g) \leq \mu(m). \tag{1}$$

For two contexts $\mathbb{K}_1 = (G_1, M_1, I_1)$ and $\mathbb{K}_2 = (G_2, M_2, I_2)$, we use notation from [4] and define the *direct product* $\mathbb{K}_1 \check{\times} \mathbb{K}_2 := (G_1 \times G_2, M_1 \times M_2, I_1 \check{\times} I_2\ )$,

$$((g, h), (m, n)) \in I_1 \check{\times} I_2 \ :\Longleftrightarrow \ (g, m) \in I_1 \text{ or } (h, n) \in I_2$$

and the *cardinal product* $\mathbb{K}_1 \hat{\times} \mathbb{K}_2 := (G_1 \times G_2, M_1 \times M_2, I_1 \hat{\times} I_2\ )$,

$$((g, h), (m, n)) \in I_1 \hat{\times} I_2 \ :\Longleftrightarrow \ (g, m) \in I_1 \text{ and } (h, n) \in I_2.$$

It follows that the direct and cardinal product fulfill De Morgan laws:

$$(\mathbb{K}_1 \check{\times} \mathbb{K}_2)^c = \mathbb{K}_1^c \hat{\times} \mathbb{K}_2^c \quad \text{ and } \quad (\mathbb{K}_1 \hat{\times} \mathbb{K}_2)^c = \mathbb{K}_1^c \check{\times} \mathbb{K}_2^c. \tag{2}$$

For two complete lattices $\mathbb{L}_1$ and $\mathbb{L}_2$ the *tensor product* $\mathbb{L}_1 \otimes \mathbb{L}_2$ is the concept lattice $\underline{\mathfrak{B}}(\mathbb{L}_1 \check{\times} \mathbb{L}_2)$, where $\mathbb{L}_1$ and $\mathbb{L}_2$ are considered as formal contexts with respect to their order relations. The concept lattice of the direct product is isomorphic to the tensor product of the factors concept lattices.

$$\underline{\mathfrak{B}}(\mathbb{K}_1 \check{\times} \mathbb{K}_2) \cong \underline{\mathfrak{B}}(\mathbb{K}_1) \otimes \underline{\mathfrak{B}}(\mathbb{K}_2). \tag{3}$$

Next, we will treat the dimension theory of formal concepts. A *Ferrers relation* is a relation $F \subseteq G \times M$ such that $(g, m), (h, n) \in F$ implies $(g, n) \in F$ or $(h, m) \in F$. The definition states that F can be brought into a stair-shaped form by permuting the rows and columns. This is equivalent to $\underline{\mathfrak{B}}(G, M, F)$ being a chain. The *length* l of F is defined as $l(F) := \#\mathfrak{B}(G, M, F) - 1$ and F is *k-step* if $k = \#\{\, \{g\}^F \mid g \in G\}$. Furthermore, it holds that the complement F^c of a k-step Ferrers relation is a Ferrers relation of length k.

Let $\mathrm{Ferr}{\upharpoonleft}(\mathbb{K})$ denote the set of all Ferrers relations contained in I and $\mathrm{Ferr}{\downharpoonleft}(\mathbb{K})$ the set of all Ferrers relations containing I. We define the *Ferrers cover number*[2] and the *Ferrers dimension* of $\mathbb{K}$:

$$\mathrm{fc}(\mathbb{K}) := \min\{\#\mathcal{F} \mid \mathcal{F} \subseteq \mathrm{Ferr}{\upharpoonleft}(\mathbb{K}),\ I = \bigcup_{F \in \mathcal{F}} F\},$$
$$\mathrm{fdim}(\mathbb{K}) := \min\{\#\mathcal{F} \mid \mathcal{F} \subseteq \mathrm{Ferr}{\downharpoonleft}(\mathbb{K}),\ I = \bigcap_{F \in \mathcal{F}} F\}.$$

Analogously, $\mathrm{Ferr}{\upharpoonleft}_k(\mathbb{K})$ denotes the set of all at most k-step Ferrers relations contained in I and $\mathrm{Ferr}{\downharpoonleft}_k(\mathbb{K})$ the set of all Ferrers relations with length less than k containing I. We define the the *Ferrers k-cover number*[2] and the *Ferrers k-dimension*:

$$\mathrm{fc}_k(\mathbb{K}) := \min\{\#\mathcal{F} \mid \mathcal{F} \subseteq \mathrm{Ferr}{\upharpoonleft}_k(\mathbb{K}),\ I = \bigcup_{F \in \mathcal{F}} F\},$$
$$\mathrm{fdim}_k(\mathbb{K}) := \min\{\#\mathcal{F} \mid \mathcal{F} \subseteq \mathrm{Ferr}{\downharpoonleft}_k(\mathbb{K}),\ I = \bigcap_{F \in \mathcal{F}} F\}.$$

Especially, we want to highlight fc_1, the *rectangle cover number*[3]:

$$\mathrm{rc}(\mathbb{K}) := \min\{\#\mathcal{F} \mid \mathcal{F} \subseteq \mathfrak{B}(\mathbb{K}),\ I = \bigcup_{(A,B) \in \mathcal{F}} A \times B\}.$$

Let $\mathcal{A}_I$ be the adjacency matrix of the incidence relation I. In [1] it is implicitly shown that $\mathrm{rc}(\mathbb{K}) = \mathrm{r_B}(\mathcal{A}_I)$, where $\mathrm{r_B}$ denotes the *Boolean rank*[4].

Lastly, we will state the dimension theory of complete lattices and relate it to the above defined dimensions of formal contexts. The *order dimension* of a complete lattice $\mathbb{L}$, $\dim(\mathbb{L})$, is the least number of chains, such that $\mathbb{L}$ can be order embedded in their product. If we restrict the cardinality of these chains to be at

[2] This term is introduced by us, although the concept itself is described in [7].

[3] In the context of Formal Concept Analysis, this term is introduced by us. With respect to Boolean matrices it has already been used in [10].

[4] The *Boolean rank*, $\mathrm{r_B}$, of an $n \times m$ Boolean matrix C is the least integer k such that Boolean $m \times k$ and $k \times n$ matrices with $C = A \circ B$ exist. This definition is equivalent to the fact that C is the sum of k rank one matrices (see [13]).

most k, we get the *k-dimension* of $\mathbb{L}$, denoted by $\dim_k(\mathbb{L})$. Of special interest will be the *2-dimension.* That is because the n-fold direct product of chains of cardinality 2 is isomorphic to the powerset lattice of the n-element set $\underline{n}$. It holds that $\mathrm{fc}(\mathbb{K}) = \mathrm{fdim}(\mathbb{K}^c) = \dim(\underline{\mathfrak{B}}(\mathbb{K}^c))$, $\mathrm{fc}_{k-1}(\mathbb{K}) = \mathrm{fdim}_k(\mathbb{K}^c) = \dim_k(\underline{\mathfrak{B}}(\mathbb{K}^c))$ and $\mathrm{r_B}(\mathcal{A}_I) = \mathrm{rc}(\mathbb{K}) = \mathrm{fdim}_2(\mathbb{K}^c) = \dim_2(\underline{\mathfrak{B}}(\mathbb{K}^c))$.

In the next section, we will need a proposition from [6] and we will also make use of some definitions from its proof.

Proposition 1. *[6, Hilfssatz 32] There exists an order embedding from $\underline{\mathfrak{B}}(\mathbb{K})$ into a complete lattice $\mathbb{L}$ if and only if there exist mappings $\alpha : G \to \mathbb{L}$ and $\beta : M \to \mathbb{L}$ with*

$$gIm \iff \alpha(g) \leq \beta(m).$$

Proof. *"$\Rightarrow$": The required order embedding is given through $\varphi(A, B) := \bigvee_{g \in A} \alpha(g)$. "$\Leftarrow$": Define $\alpha := \varphi \circ \gamma$ and $\beta := \varphi \circ \mu$.*

3 Tolerance Spaces, Dimension And Graph Theory

A *tolerance relation* or simply a *tolerance* is a reflexive and symmetric binary relation τ on a non-empty finite set V. The pair $(V, \tau) =: \mathbb{T}$ is called *tolerance space* and is a special case of a formal context. An introduction to tolerance spaces together with applications can be found in [8] and [9].

For a tolerance τ on V, a non-empty subset $S \subseteq V$ is called *τ-preblock* if $S \times S$ is contained in τ. A maximal τ-preblock with respect to set inclusion is called *τ-block.* In other words, this means that a τ-block $S \subseteq V$ defines a non-enlargeable *square* $S \times S \subseteq \tau$. The set of all τ-blocks is denoted by $\mathrm{Bl}(\mathbb{T})$. Analogously, the set of all maximal squares of $\mathbb{T}$ is denoted by $\mathrm{Sq}(\mathbb{T})$. This set determines the tolerance τ, that is $\tau = \bigcup \mathrm{Sq}(\mathbb{T})$. But often not all squares are necessary to cover τ. This motivates the definition of the *square cover number*[5], $\mathrm{sc}(\mathbb{T})$, of a tolerance space $\mathbb{T}$, as the minimal number of maximal squares necessary to cover τ:

$$\mathrm{sc}(\mathbb{T}) := \min\{\#\mathcal{S} \mid \mathcal{S} \subseteq \mathrm{Sq}(\mathbb{T}),\ \tau = \bigcup \mathcal{S}\}.$$

Another covering problem of tolerance spaces is the *block cover number*[6]:

$$\mathrm{bc}(\mathbb{T}) := \min\{\#\mathcal{B} \mid \mathcal{B} \subseteq \mathrm{Bl}(\mathbb{T}),\ V = \bigcup \mathcal{B}\}.$$

Similarly to the Ferrers cover numbers and rectangle cover number of general formal contexts, we can relate these cover numbers of tolerance spaces to an

[5] This term is introduced by us and is a logical consequence of the term "rectangle cover number".

[6] This term is introduced by us.

intersection problem and a dimension of the complements concept lattice.

We start with the square cover number and notice that the complement of a square is a symmetric Ferrers relation of length 1. Hence, we define the *symmetric Ferrers* 2*-dimension* of $\mathbb{T}^c$, denoted by sfdim_2, as the smallest number of symmetric Ferrers relations of length 1 whose intersection is equal to τ^c.

The concept lattice[7] $\underline{\mathfrak{B}}(\mathbb{T}^c)$ was characterized in [12] as a *complete ortholattice*. This is a complete bounded lattice $\mathbb{L} = (L, \leq, c)$ with an involutory antiautomorphism c, such that for all $x \in L$ it holds that $x \wedge x^c = 0$ and $x \vee x^c = 1$. An abstract *orthogonality relation* $\perp$ is defined through $x \perp y :\Longleftrightarrow x \leq y^c$. In the special case of concept lattices the *orthocomplement* is given via $(A, B)^c := (B, A)$.

Definition 1. *An* orthoembedding*, between two complete ortholattices* $\mathbb{L}_1 = (L_1, \leq, c_1)$ *and* $\mathbb{L}_2 = (L_2, \leq, c_2)$*, is an order embedding* $\varphi : \mathbb{L}_1 \to \mathbb{L}_2$ *which additionally preserves orthogonality (*$x \perp y \Longrightarrow \varphi(x) \perp \varphi(y)$*), such that there exists an order preserving map* $\psi : \mathbb{L}_2 \to \mathbb{L}_1$ *satisfying for all* $x \in L_1$*:*

1. $\psi(\varphi(x)) = x$,
2. $\psi(\varphi(x)^{c_2}) = x^{c_1}$.

Remark 1. The map φ is a section from $\mathbb{L}_1$ to $\mathbb{L}_2$, such that φ and the *dual* of φ given by

$$\varphi^d : \mathbb{L}_1 \to \mathbb{L}_2, \;\; x \mapsto (\varphi(x^{c_1}))^{c_2}$$

have ψ as a common retraction.

Fig. 1. The maps φ and $\varphi^d := c_2 \circ \phi \circ c_1$, and their common retraction ψ.

$$\begin{tikzcd}
\mathbb{L}_1 \arrow[d, tail, "\varphi"'] & \mathbb{L}_1 \arrow[l, tail, two heads, "c_1"'] \\
\mathbb{L}_2 \arrow[r, tail, two heads, "c_2"'] & \mathbb{L}_2 \arrow[u, two heads, "\psi"']
\end{tikzcd}$$

The next proposition provides an equivalent condition for the existence of an orthoembedding from $\underline{\mathfrak{B}}(\mathbb{T}^c)$ to a Boolean algebra (in other words a distributive ortholattice) with the additional property that $x \perp y \Longleftrightarrow x \wedge y = 0$.

[7] The concept lattice was treated under the name *neighborhood ortholattice*.

Proposition 2. *There exists an orthoembedding φ from $\underline{\mathfrak{B}}(\mathbb{T}^c)$ into a Boolean algebra $\mathbb{L} = (L, \leq, c)$, with the special property that for all $x, y \in L$ it holds that $x \perp y \Longleftrightarrow x \wedge y = 0$, if and only if there exists a mapping $\alpha : V \to \mathbb{L}$ with*

$$u \tau^c v \Longleftrightarrow \alpha(u) \perp \alpha(v).$$

Proof. "⇒": We define $\alpha := \varphi \circ \gamma$ and use Equation 1 to conclude that:

$$\begin{aligned} u\,\tau^c v &\Longleftrightarrow \gamma(u) \leq \mu(v) = \gamma(v)^c \\ &\Longleftrightarrow \gamma(u) \perp \gamma(v) \\ &\Longrightarrow \varphi \circ \gamma(u) \perp \varphi \circ \gamma(v). \end{aligned} \qquad \begin{aligned} u\,\tau^c v &\Longleftrightarrow \gamma(u) \leq \gamma(v)^c \\ &\Longleftarrow \psi(\varphi \circ \gamma(u)) \leq \psi((\varphi \circ \gamma(v))^c) \\ &\Longleftarrow \varphi \circ \gamma(u) \leq (\varphi \circ \gamma(v))^c \\ &\Longleftrightarrow \varphi \circ \gamma(u) \perp \varphi \circ \gamma(v). \end{aligned}$$

"⇐": We define $\varphi(A, B) := \bigvee_{a \in A} \alpha(a)$. It follows from Proposition 1 that φ is an order embedding. Let (A, B) and (C, D) be formal concepts of $\underline{\mathfrak{B}}(\mathbb{T}^c)$. We show that $(A, B) \perp (C, D) \Longrightarrow \varphi(A, B) \perp \varphi(C, D)$. From the definition of φ and the complement in $\underline{\mathfrak{B}}(\mathbb{T}^c)$ it follows that:

$$(A, B) \perp (C, D) \Longleftrightarrow (A, B) \leq (D, C) \Longleftrightarrow \bigvee_{a \in A} \alpha(a) \leq \bigvee_{d \in D} \alpha(d), \tag{4}$$

$$\varphi(A, B) \perp \varphi(C, D) \Longleftrightarrow \bigvee_{a \in A} \alpha(a) \wedge \bigvee_{c \in C} \alpha(c) = 0. \tag{5}$$

In the next step, we show that 4 implies 5 by "connecting" the ends.
"(4)⇒(5)": Since, (C, D) is a formal concept it holds that $\alpha(d) \leq \alpha(c)^c$ for all $d \in D$ and all $c \in C$.

$$(4) \Rightarrow \bigvee_{a \in A} \alpha(a) \leq \bigvee_{c \in C} \alpha(c)^c \Rightarrow \bigvee_{a \in A} \alpha(a) = \bigvee_{a \in A} \alpha(a) \wedge \bigvee_{c \in C} \alpha(c)^c.$$

Finally, we take the meet with $\bigvee_{c \in C} \alpha(c)$ and use distributivity.

$$\bigvee_{a \in A} \alpha(a) \wedge \bigvee_{c \in C} \alpha(c) = (\bigvee_{a \in A} \alpha(a) \wedge \bigvee_{c \in C} \alpha(c)^c) \wedge \bigvee_{c \in C} \alpha(c) = 0.$$

Furthermore, we define the desired map $\psi : \mathbb{L} \to \underline{\mathfrak{B}}(\mathbb{T}^c)$ through $\psi(x) := (\{v \in V \mid \alpha(v) \leq x\}, \{v \in V \mid x \leq \alpha(v)^c\})$ and show that $\psi(x)$ is a formal concept of $\underline{\mathfrak{B}}(\mathbb{T}^c)$, as well as that ψ satisfies Property 1 and 2 from Definition 1. The proof is similar to some parts of the proof from the Basic Theorem on Concept Lattices (see [7]). Also note that ψ is order preserving.

The definition of formal concepts states that $A^{\tau^c} = B$ and $B_{\tau^c} = A$ must hold in order for $(A, B) \in \mathfrak{B}(\mathbb{T}^c)$. We only show the second condition as the first one can be shown in the same way.

$$\begin{aligned} u \in \{v \in V \mid \alpha(v) \leq x\} &\iff \alpha(u) \leq x \\ &\iff \alpha(u) \leq \alpha(w)^c \text{ for all } w \in \{v \in V \mid x \leq \alpha(v)^c\} \\ &\iff u\,\tau^c w \text{ for all } w \in \{v \in V \mid x \leq \alpha(v)^c\} \\ &\iff u \in \{v \in V \mid x \leq \alpha(v)^c\}_{\tau^c}. \end{aligned}$$

Next, we show that the second component from $\psi(\varphi(A, B))$ equals B. The first component must then be equal to A, due to the fact shown above.

$$\begin{aligned} \{v \in V \mid \bigvee_{a \in A} \alpha(a) \leq \alpha(v)^c\} &= \{v \in V \mid \alpha(a) \leq \alpha(v)^c \text{ for all } a \in A\} \\ &= \{v \in V \mid a\,\tau^c v \text{ for all } a \in A\} \\ &= A^{\tau^c} = B. \end{aligned}$$

Lastly, we show that the first component from $\psi(\varphi(A, B)^c)$ equals B. The second component must then be equal to A, due to the fact shown above. Hence, we can conclude that $\psi(\varphi(A, B)^c) = \psi(\bigwedge_{a \in A} \alpha(a)^c) = (B, A) = (A, B)^c$.

$$\begin{aligned} \{v \in V \mid \alpha(v) \leq \bigwedge_{a \in A} \alpha(a)^c\} &= \{v \in V \mid \alpha(v) \leq \alpha(a)^c \text{ for all } a \in A\} \\ &= \{v \in V \mid v\,\tau^c a \text{ for all } a \in A\} \\ &= A^{\tau^c} = B. \end{aligned}$$

Definition 2. *The* orthodimension *of a complete ortholattice* $\mathbb{L} = (L, \leq, c)$, *denoted by* $\dim_\perp(\mathbb{L})$, *is the smallest* n *such that there exists an orthoembedding from* $\mathbb{L}$ *to* $\mathfrak{P}(\underline{n})$.

Theorem 1. *For a tolerance space* $\mathbb{T}$ *its holds that* $\mathrm{sc}(\mathbb{T}) = \dim_\perp(\underline{\mathfrak{B}}(\mathbb{T}^c))$.

Proof. If $\mathrm{sc}(\mathbb{T}) = n$, there exists a minimal square cover $\{S_1 \times S_1, \ldots, S_n \times S_n\}$. It follows from [5] that this is equivalent to the existence of a minimal *set representation* of $\mathbb{T}$, which is a map $\alpha : V \to \mathfrak{P}(\underline{\mathrm{n}})$, such that we have $u\tau v \iff \alpha(a) \cap \alpha(v) \neq \emptyset$. From the minimal square cover, this map can be defined via:

$$\alpha : v \mapsto \{i \mid v \in S_i\}.$$

Consequently, α provides a minimal *complementary set representation* for $\mathbb{T}^c$, that is $u\,\tau^c v \iff \alpha(a) \cap \alpha(v) = \emptyset$. Since it holds that $\alpha(a) \cap \alpha(v) = \emptyset \iff \alpha(a) \subseteq \alpha(v)^c \iff \alpha(a) \perp \alpha(v)$, Proposition 2 yields that $\dim_\perp(\underline{\mathfrak{B}}(\mathbb{T}^c)) = n$. As all stated implications are equivalences, it follows that $\mathrm{sc}(\mathbb{T}) = \dim_\perp(\underline{\mathfrak{B}}(\mathbb{T}^c))$.

We have shown that analogue to general formal contexts, in the special case of tolerance spaces, it holds that:

$$\mathrm{sc}(\mathbb{T}) = \mathrm{sfdim}_2(\mathbb{T}^c) = \dim_\perp(\underline{\mathfrak{B}}(\mathbb{T}^c)).$$

Next, we will treat the block cover number. For this, we will use some tools from graph theory which we introduce in the following.

The *underlying graph*, $\mathbb{G}_\mathbb{T}$, of a tolerance space $\mathbb{T} = (V, \tau)$ is defined through the same relation but with all diagonal elements removed. Analogously, to the block and square cover number of $\mathbb{T}$, we can define the *vertex clique cover number*, $\theta_v(\mathbb{G}_\mathbb{T}) = \mathrm{bc}(\mathbb{T})$, and the *edge clique cover number*, $\theta_e(\mathbb{G}_\mathbb{T}) = \mathrm{sc}(\mathbb{T})$ (see [3]). The vertex clique cover number is equal to the *chromatic number* of the complementary graph (see [14]), $\chi(\mathbb{G}^c_\mathbb{T})$[8]. Here the complement is taken in the sense of graph theory, which always yields an irreflexive relation. On the other hand, for tolerance spaces we consider full complements as defined in Section 2. This yields $\mathrm{bc}(\mathbb{T}) = \chi(\mathbb{T}^c)$, since τ^c is irreflexive and symmetric.

We saw that $\underline{\mathfrak{B}}(\mathbb{T}^c)$ is a complete ortholattice. An *orthomap* between complete ortholattices (see [12]) preserves order and orthogonality, and maps only the bottom element of the domain lattice to the bottom element of the codomain lattice.

Definition 3. *We define the* chromatic dimension, $\mathrm{cdim}(\mathbb{L})$*, of a complete ortholattice* $\mathbb{L} = (L, \leq, c)$*, to be the minimal* n *such that an orthomap to the powerset lattice* $\mathfrak{P}(\underline{n})$ *exists.*

The nomenclature is justified by the following proposition.

Proposition 3. *For a graph* $\mathbb{G} = (V, E)$ *with an irreflexive and symmetric relation* $E \subseteq V \times V$*, it holds that* $\chi(\mathbb{G}) = \mathrm{cdim}(\underline{\mathfrak{B}}(\mathbb{G}))$*.*

Proof. The chromatic number of $\mathbb{G}$ is n if and only if n is minimal with the property that there exists a graph homomorphism to K_n, the complete graph with n vertices. In [12] it is shown that this is equivalent to the existence of an orthomap from $\underline{\mathfrak{B}}(\mathbb{G})$ to $\mathfrak{P}(\underline{\mathrm{n}}) \cong \underline{\mathfrak{B}}(K_n)$. Consequently, it holds that $\mathrm{cdim}(\underline{\mathfrak{B}}(\mathbb{G})) = n$.

The calculation of the chromatic number is a minimization problem, but not an intersection problem. In order to define an intersection problem, we notice that the complement B^c for a block $B \in \mathrm{Bl}(\mathbb{T})$ is an *independent set* of vertices with respect to τ^c. Hence, we can define an intersection problem with respect to $\mathbb{T}^c$. This yields the *independence dimension* of $\mathbb{T}^c$, denoted by $\mathrm{idim}(\mathbb{T}^c)$, to be the smallest number of independent sets whose intersection is empty.

$$\mathrm{bc}(\mathbb{T}) = \mathrm{idim}(\mathbb{T}^c) = \chi(\mathbb{T}^c) = \mathrm{cdim}(\underline{\mathfrak{B}}(\mathbb{T}^c)).$$

[8] The chromatic number of a graph is the minimal n such that a graph homomorphism, which is an edge preserving vertex map, to the complete graph with n vertices exists.

4 Set Cover Problems And Their Product

A *set cover system* is a triple $\mathbb{S} := (U, X, \mathcal{S})$, with *universe* U, a subset $X \subseteq U$ and $\mathcal{S} \subseteq \mathfrak{P}(X)$. The *cover number* and *isolation number* of $\mathbb{S}$ are defined as:

$$\begin{aligned} \mathrm{c}(\mathbb{S}) &:= \min\{\#\tilde{\mathcal{S}} \mid \tilde{\mathcal{S}} \subseteq \mathcal{S},\, X = \bigcup \tilde{\mathcal{S}}\}, \\ \mathrm{i}(\mathbb{S}) &:= \max\{\#\tilde{X} \mid \tilde{X} \subseteq X, \forall\, S \in \mathcal{S} : \#(\tilde{X} \cap S) \leq 1\}. \end{aligned}$$

The isolation number is the maximal cardinality of an *isolated set* $\tilde{X}$ from $\mathbb{S}$, which means that $\tilde{X}$ is maximal with respect to the property that any pair of its elements is not contained in the same $S \in \mathcal{S}$. Consequently, the isolation number is a lower bound for the cover number.

Remark 2. Both optimization problems can be described as an integer linear program. For this purpose, we define the representation matrix $A \in \{0,1\}^{X \times \mathcal{S}}$ of $\mathbb{S}$ through $A(i, S) = 1 :\Leftrightarrow i \in S$.

$$\begin{array}{llll} \text{minimize:} & \displaystyle\sum_{S \in \mathcal{S}} x_S & \text{maximize:} & \displaystyle\sum_{i \in X} y_i \\ \text{subject to:} & A\boldsymbol{x} \geq \mathbb{1} & \text{subject to:} & A^T \boldsymbol{y} \leq \mathbb{1} \\ & x_S \in \{0,1\}. & & y_i \in \{0,1\}. \end{array}$$

Thus, the inequality $\mathrm{i}(\mathbb{S}) \leq \mathrm{c}(\mathbb{S})$ also follows from the weak duality theorem of optimization and the difference $\mathrm{c}(\mathbb{S}) - \mathrm{i}(\mathbb{S})$ is the *duality gap*.

A *set intersection system* is a triple $\mathbb{S} := (U, X, \mathcal{S})$, with *universe* U, a subset $X \subseteq U$ and $\mathcal{S} \subseteq \mathfrak{P}(U)$, such that $X \subseteq S$ for all $S \in \mathcal{S}$. The *intersection number* is defined as:

$$\mathrm{int}(\mathbb{S}) := \min\{\#\tilde{\mathcal{S}} \mid \tilde{\mathcal{S}} \subseteq \mathcal{S},\, X = \bigcap \tilde{\mathcal{S}}\}.$$

It follows that the complement of a set cover system, defined through $\mathbb{S}^c := (U, X^c, \{S^c \mid S \in \mathcal{S}\})$[9], is a set intersection system with $\mathrm{c}(\mathbb{S}) = \mathrm{int}(\mathbb{S}^c)$.

The product of two set cover systems $\mathbb{S}_1 = (U_1, X_1, \mathcal{S}_1)$ and $\mathbb{S}_2 = (U_2, X_2, \mathcal{S}_2)$ is defined as $\mathbb{S}_1 \times \mathbb{S}_2 := (U_1 \times U_2, X_1 \times X_2, \mathcal{S}_1 \times \mathcal{S}_2)$. This definition yields to the next theorem which is a generalization of Theorem 3.2 from [13].

Theorem 2. *For the the product of set cover systems $\mathbb{S}_1$ and $\mathbb{S}_2$, it holds that:*

$$\begin{aligned} \max(\mathrm{i}(\mathbb{S}_1)\,\mathrm{c}(\mathbb{S}_2), \mathrm{c}(\mathbb{S}_1)\,\mathrm{i}(\mathbb{S}_2)) \leq \mathrm{c}(\mathbb{S}_1 \times \mathbb{S}_2) \leq \mathrm{c}(\mathbb{S}_1)\,\mathrm{c}(\mathbb{S}_2) \\ \mathrm{i}(\mathbb{S}_1)\,\mathrm{i}(\mathbb{S}_2) \leq \mathrm{i}(\mathbb{S}_1 \times \mathbb{S}_2) \leq \min(\mathrm{i}(\mathbb{S}_1)\,\mathrm{c}(\mathbb{S}_2), \mathrm{c}(\mathbb{S}_1)\,\mathrm{i}(\mathbb{S}_2)). \end{aligned}$$

[9] The complements are defined with respect to U.

Proof. We prove the first inequality and the second one follows from duality. For the upper bound, let $\tilde{\mathcal{S}}_1$ and $\tilde{\mathcal{S}}_2$ be minimal covers from $\mathbb{S}_1$ and $\mathbb{S}_2$ respectively. It is easy to see that $\tilde{\mathcal{S}}_1 \times \tilde{\mathcal{S}}_2$ is a cover from $\mathbb{S}_1 \times \mathbb{S}_2$.

For the lower bound, let $\tilde{\mathcal{S}}$ be a minimal cover from $\mathbb{S}_1 \times \mathbb{S}_2$ and $\tilde{X}_1$ a maximal isolated set from $\mathbb{S}_1$. We define for $i \in \tilde{X}_1$ the set $\tilde{\mathcal{S}}_i := \{(S,T) \mid (S,T) \in \tilde{\mathcal{S}},\, i \in S\}$. Since $\tilde{X}_1$ is an isolated set, it follows that for different $i, j \in \tilde{X}_1$, the sets $\tilde{\mathcal{S}}_i$ and $\tilde{\mathcal{S}}_j$ are disjoint. A similar argument implies that every $\tilde{\mathcal{S}}_i$ induces a cover from $\mathbb{S}_2$ and hence $\#\tilde{\mathcal{S}}_i \geq \mathrm{c}(\mathbb{S}_2)$. These facts yield:

$$\#\tilde{\mathcal{S}} \geq \#(\biguplus_{i \in \tilde{X}_1} \tilde{\mathcal{S}}_i) \geq \#\tilde{X}_1\, \mathrm{c}(\mathbb{S}_2) = \mathrm{i}(\mathbb{S}_1)\, \mathrm{c}(\mathbb{S}_2).$$

The other lower bound's component can be deduced in the same way.

5 Dimension Of The Tensor Product Of Complete Lattices

With Theorem 2, it is easy to provide a sufficient condition when dim, $\dim_k$, $\dim_2$, $\dim_\perp$ and cdim are multiplicative with respect to the tensor product. Since every complete lattice is isomorphic to a concept lattice ([7]), we can shift this problem to the multiplicativity of the associated intersection number with respect to the direct product (Equation 3). Furthermore, due to Equation 2, we transform this problem to the multiplicativity of the corresponding cover number with respect to the cardinal product. All we have to do is to define suitable set cover systems and show that there product expresses the cardinal product of the underlying formal contexts.

In the sense of Section 4, we define the *Ferrers isolation number* fi, the *k-Ferrers isolation number* fi_k, the *rectangle isolation number* ri, the *square isolation number* si and the *block isolation number* bi.

Fig. 2. This table gives an overview about the set cover problems introduced above. Note that $\mathfrak{B}(\mathbb{K})$ actually denotes a set of formal concepts, but here we identify it with the set of all rectangles.

set cover system	cover n.	isolation n.	intersection n.	lattice dim.
$(G \times M, I, \mathrm{Ferr}\!\upharpoonright\!(\mathbb{K}))$	$\mathrm{fc}(\mathbb{K})$	$\mathrm{fi}(\mathbb{K})$	$\mathrm{fdim}(\mathbb{K}^c)$	$\dim(\underline{\mathfrak{B}}(\mathbb{K}^c))$
$(G \times M, I, \mathrm{Ferr}\!\upharpoonright_{k-1}(\mathbb{K}))$	$\mathrm{fc}_{k-1}(\mathbb{K})$	$\mathrm{fi}_{k-1}(\mathbb{K})$	$\mathrm{fdim}_k(\mathbb{K}^c)$	$\dim_k(\underline{\mathfrak{B}}(\mathbb{K}^c))$
$(G \times M, I, \mathfrak{B}(\mathbb{K}))$	$\mathrm{rc}(\mathbb{K})$	$\mathrm{ri}(\mathbb{K})$	$\mathrm{fdim}_2(\mathbb{K}^c)$	$\dim_2(\underline{\mathfrak{B}}(\mathbb{K}^c))$
$(V \times V, \tau, \mathrm{Sq}(\mathbb{T}))$	$\mathrm{sc}(\mathbb{T})$	$\mathrm{si}(\mathbb{T})$	$\mathrm{sfdim}_2(\mathbb{T}^c)$	$\dim_\perp(\underline{\mathfrak{B}}(\mathbb{T}^c))$
$(V, V, \mathrm{Bl}(\mathbb{T}))$	$\mathrm{bc}(\mathbb{T})$	$\mathrm{bi}(\mathbb{T})$	$\mathrm{idim}(\mathbb{T}^c)$	$\mathrm{cdim}(\underline{\mathfrak{B}}(\mathbb{T}^c))$

For the cardinal product of formal contexts, it holds that $(A,B) \in \mathfrak{B}(\mathbb{K}_1 \hat{\times} \mathbb{K}_2)$ if and only if there exists $(A_1,B_1) \in \mathfrak{B}(\mathbb{K}_1)$ and $(A_2,B_2) \in \mathfrak{B}(\mathbb{K}_2)$, such that $A = A_1 \times A_2$ and $B = B_1 \times B_2$ (see [11]). Only the formal concepts with $A \neq \emptyset$ and $B \neq \emptyset$ are of importance for the covering problems of the cardinal product, since they correspond to maximal rectangles. Hence, these formal concepts (A,B) can be uniquely identified with the pair $((A_1,B_1),(A_2,B_2))$. Comparing this with the definition of the product of set cover systems, we see the desired correspondence to the cardinal product of the respective formal contexts. The same holds for the Ferrers relations. Also note that squares are a special case of rectangles and that blocks are derived from maximal squares. These fact yield the following theorem.

Theorem 3. *We consider the set cover problems of Figure 2 and their products. If for one of the factors, it holds that the isolation number is equal to the cover number, then the respective lattice dimension is multiplicative with respect to the tensor product of the corresponding complete lattices.*

Remark 3. We introduce the *strong product* of two simple graphs $\mathbb{G}_1 = (V_1, E_1)$ and $\mathbb{G}_2 = (V_2, E_2)$, defined as $\mathbb{G}_1 \boxtimes \mathbb{G}_2 := (V_1 \times V_2, \tilde{E})$, with $(u_1,u_2)\tilde{E}(v_1,v_2) :\Longleftrightarrow$ $(u_1 E_1 v_1$ and $u_2 = v_2)$ or $(u_1 = v_1$ and $u_2 E_2 v_2)$ or $(u_1 E_1 v_1$ and $u_2 E_2 v_2)$.

The *reflexive closure* of a graph $\mathbb{G} = (V,E)$ is defined as $\mathbb{G}^{\text{ref}} := (V, E^{\text{ref}})$, where E^{ref} is the reflexive closure of the symmetric relation E. Hence, $\mathbb{G}^{\text{ref}}$ is a tolerance space. It holds that $(\mathbb{G}_1 \boxtimes \mathbb{G}_2)^{\text{ref}} = \mathbb{G}_1^{\text{ref}} \hat{\times} \mathbb{G}_2^{\text{ref}}$. Consequently, we have that $\theta_e(\mathbb{G}_1 \boxtimes \mathbb{G}_2) = \text{sc}(\mathbb{G}_1^{\text{ref}} \hat{\times} \mathbb{G}_2^{\text{ref}})$. In [2] the multiplicativity of the edge clique cover number with respect to the strong product was studied and similar results as with the square cover number of the cardinal product of tolerance spaces have been obtained. That is why this setting would provide another example of a set cover system and its product.

In [2] an example such that $\theta_e(\mathbb{G} \boxtimes \mathbb{G}) < \theta_e(\mathbb{G})\theta_e(\mathbb{G})$ was provided for $\mathbb{G}$ being the join of a 5-cycle with two isolated vertices. We did a computational experiment to find the smallest tolerance space (in terms of the number of vertices) for which the square isolation number is strictly smaller than the square cover number. Thereby, we found out that the smallest tolerance space has 7 vertices and is the reflexive closure of the graph $\mathbb{G}$ described above.

6 Conclusion

We showed that the three fold relationship between cover problem (Ferrers cover number), intersection problem (Ferrers dimension) and lattice dimension (order dimension) also applies to tolerance spaces. That is the equality of the square cover number, the symmetric Ferrers 2-dimension and the orthodimension. Surprisingly, this theme is also present in the case of the block cover number and the chromatic dimension.

Additionally, we highlighted how the cover problems with respect to tolerance spaces have a strong graph theoretic flavor, *i.e.*, interpretations in terms of the chromatic number or the relationship to the strong product of graphs.

Our initial question, about the multiplicativity of the lattice dimension with respect to the tensor product, could, in all investigated examples, be translated to a cover problem of the cardinal product of the related formal contexts complements. This fundamental principle lead to the abstraction to the general set cover problem, which provides a unified setting to treat these various cover problems related to formal contexts. Especially, the question about the multiplicativity of the dimension of the cardinal product could be dealt with in a unified way. This in turn lead to a sufficient condition for the multiplicativity of the lattice dimensions with respect to the tensor product of complete lattices.

The introduced isolation numbers have to our knowledge not been present in the theory of formal concept analysis. It is an open problem to find a purely lattice theoretical interpretation of these isolation numbers.

Acknowledgments. Finally, we want thank the anonymous referees for their effort and valuable suggestions.

References

1. Belohlavek, R., Vychodil, V.: Discovery of optimal factors in binary data via a novel method of matrix decomposition. JCSS **76**, 3–20 (2010)
2. Choudum, S.A., Parthasarathy, K.R.: The edge clique cover number of product graphs. Jour. Math. Phy. Sci. **10**, 255–261 (1976)
3. Choudum, S.A., Parthasarathy, K.R., Ravindra, G.: Line-clique cover number of a graph. Indian Nat. Sci. Acad. Proc. **41**, 289–293 (1975)
4. Deiters, K., Erné, M.: Sums, products and negations of contexts and complete lattices. Algebra Universalis **60**, 469–496 (2009)
5. Erdös, P., Goodman, A.W., Pósa, L.: The representation of graphs by set intersections. Canadian J. of Math. **18**, 106–112 (1966)
6. Ganter, B.: Diskrete Mathematik: Geordnete Mengen. Springer Spektrum (2013)
7. Ganter, B., Wille, R.: Formal Concept Analysis: Mathematical Foundations. Springer-Verlag New York, Inc. (1997)
8. Peters, J.F., Wasilewski, P.: Tolerance spaces: Origins, theoretical aspects and applications. Information Sciences **195**, 211–225 (2012)
9. Pogonowski, J.: Introduction to Graph Theory. University Press, Institute of Linguistics, Adam Mickiewicz University (1981)
10. Pourmoradnasseri, M., Theis, D.O.: The rectangle covering number of random boolean matrices. The Electronic J. of Combinatorics **24** (2017)
11. Reuter, K.: On the dimension of the cartesian product of relations and orders. Order **6**, 277–293 (1989)
12. Walker, J.W.: From graphs to ortholattices and equivariant maps. Journal of Combinatorial Theory **B 35**, 171–192 (1983)
13. Watts, V.L.: Covers and partitions of graphs by complete bipartite subgraphs. Ph.D. thesis, Queen's University (Canada) (2001)
14. West, D.B.: Tolerance spaces with applications to linguistics. Prentice Hall (2000)

Finding Concepts with Unexpected Multi-Label Objects Based on Shared Subspace Method

HongJie Zhai, Makoto Haraguchi, Yoshiaki Okubo, and HeYu Liu

Graduate School of Information Science and Technology
Hokkaido University
N-14 W-9, Sapporo 060-0814, Japan
mh@ist.hokudai.ac.jp

Abstract. We discuss a method of retrieving *unexpected* objects for a given query, where each data object is represented as a feature vector and assigned a multi-label as well. Given an object-feature matrix X_1 and an object-label matrix X_2, we try to *simultaneously* factorize X_1 and X_2 as $X_1 \approx BV$ and $X_2 \approx SW$ by means of *Nonnegative Shared Subspace Method*, where the basis S is a part (subspace) of the basis B. With the help of the shared subspace, thus, we can predict a multi-label for a query feature-vector with unknown labels. Our unexpected object for the query is defined as an object which is similar to the query in the feature space, but is dissimilar in the label space. In order to obtain unexpected objects from several viewpoints of similarity, we formalize our retrieval task as a problem of finding formal concepts satisfying a constraint w.r.t. the unexpectedness. We present an efficient depth-first branch-and-bound algorithm for extracting our target concepts.

Keywords: formal concept, shared subspace method, nonnegative matrix factorization, unexpectedness of objects, multi-labels, recommendation

1 Introduction

Information Retrieval (IR) is a fundamental task in our daily life. In popular keyword-based IR, since it is not easy to get desirable data objects by providing query keywords just once, we *iteratively* input queries until we can meet satisfiable results. Particularly, in *Associative Search* [9], at each step we repeatedly input a query, our query is shifted to its sibling concept [10]. As the results, we often find an interesting search result which is *surprising* or *unexpected* for us but still keeps a certain degree of relevance to our initial query. The authors consider that such an aspect of associative search is strongly desirable especially for *recommendation-oriented IR systems*. This paper discusses a recommendation-oriented method for finding interesting objects for a given query, especially taking an *unexpectedness* of objects for the query into account.

A notion of unexpectedness in recommender systems has been discussed in [14]. In the framework, the unexpectedness of an item (object) is evaluated

© Dmitry I. Ignatov, Lhouari Nourine (Eds.): CLA 2018, pp. 93–104,
ISBN 978–80–244–5328–6, Department of Computer Science, Palacký University
Olomouc, 2018.

based on a distance between the item and those a user already knows in some sense. Another related notions, *novelty* and *serendipity*, have also been investigated in [16]. An object is said to be novel for a user if he/she does not know it. For example, it can be evaluated based on user's history of recommendations. On the other hand, since the notion of serendipity is emotional and difficult to be defined, it has been discussed in terms of *diversity* which is based on dissimilarity among objects [16]. It is noted here that those notions previously proposed are *subjectively* defined because we need some kind of *user-dependent* information.

In contrast with them, we propose an *objective* unexpectedness of objects. A data object is usually represented as a vector in a primary feature space. Then, the notions of novelty and serendipity have been formalized with the help of additional information specific to particular users. Nowadays, however, several kinds of additional information are also *commonly* available. In case of movie objects, for example, each movie would be primarily represented as a vector of feature terms extracted from their plots. In addition, each movie is often assigned some *genre labels* by commercial companies or many SNS users. Since those secondary features provide us valuable information about movies, they would make our IR systems more flexible and useful for a wide range of users.

In our framework, as such commonly-available additional features, we assume each object is assigned some *labels* (as a multi-label) beforehand. That is, our data objects are given by two data matrices, X_1 and X_2, each of which represents an *object-feature* relation and *object-label* relation, respectively. Then, we propose our notion of unexpectedness *with respect to label-information of objects*.

More concretely speaking, in our recommendation-oriented IR, a query q is given as a feature vector and supposed to have no label. As a reasonable guess, we often consider that if an object x is similar to q in the feature space, q would also have a multi-label similar to that of x. Conversely, if we observe (by any means) their multi-labels are far from each other, we would find some unexpectedness of x for q because they have distant multi-labels even though their features are similar. Based on the idea of unexpectedness, we formalize our IR task as a problem of detecting *formal concepts* [12] each of which contains some unexpected objects in the extent. By finding those formal concepts, we can obtain our search results from various viewpoints of similarity among the query and objects.

The point in our IR is to predict a multi-label of the query represented as a feature-vector. For the task, our object-feature matrix X_1 and object-label matrix X_2 are *simultaneously* factorized as $X_1 \approx BV$ and $X_2 \approx SW$ by *Non-negative Shared Subspace Method* [1], where the basis S is a part (subspace) of the basis B. In a word, such a shared subspace associates the label-information with the feature-information of the original matrices. With the shared subspace, we can predict a multi-label for the query feature-vector with unknown labels.

To predict a multi-label of a given object, a method of *multi-label classification* has already been proposed in [8]. In the method, we need to obtain a subspace and its orthogonal basis for the original feature space which approximately reflects similarity among multi-labels assigned to the objects by solving an eigenvalue problem. However, such an orthogonal basis yields negative components in the subspace which complicate interpretation of our search results. Moreover, orthogonality is not necessarily required in prediction purpose. As its simplified and efficient version, a prediction method has also been discussed in [17], where a subspace is just defined as a *real line*. However, the prediction is not so reliable as we will see later. This is the reason why we prefer nonnegative factorization in our framework.

In our recommendation-oriented IR, for a given query, we try to find formal concepts whose extents contains some unexpected objects for the query. We first create a formal context consisting of only objects similar (relevant) to the query in the feature space with a standard *Nonnegative Matrix Factorization* [2]. Then, we try to extract concepts with unexpected ones in the context. Since we often have a huge number of concepts, we evaluate a concept with its extent E by the average distance between each object in E and the query in the label-subspace, and try to extract concepts with the top-N largest evaluation values. We present a depth-first algorithm for finding those top-N concepts, where a simple branch-and-bound pruning based on the evaluation function is available. Our experimental result for a movie dataset shows our system can actually detect an interesting concept of movies whose plots are similar to a given query but some of them have genre-labels which are far from predicted genres of the query.

From the viewpoint of *Formal Concept Analysis* (*FCA*) [12], our study in this paper is closely related to several interesting topics. In order to reduce the size of formal context preserving important information, methods with non-negative matrix factorizations have been investigated in [3, 5]. Although our method is also based on such a factorization technique, the main purpose is not only to reduce the context but also to associate label information with the reduced context.

A smaller lattice with a reduced number of concepts is desirable as a practical requirement in FCA. Computing a subset of possible concepts, e.g., in [6], is a useful approach for that purpose. Interestingness measures of concepts can meaningfully restrict our targets to be extracted and several representatives are surveyed in [4]. Although our method also proposes a kind of interestingness based on unexpectedness of objects, we emphasize that it is a query-specific one.

2 Approximating Data Matrices Based on Nonnegative Shared Subspace Method

In this section, we discuss how to *simultaneously* approximate a pair of data matrices representing different informations of the same (common) objects. The approximation is based on *Nonnegative Shared Subspace Method* [1].

2.1 Approximating Object-Feature Matrix Reflecting Multi-Label Information

Let $X_1 = \begin{pmatrix} f_1 & \cdots & f_N \end{pmatrix}$ be an $M \times N$ *object-feature* matrix and $X_2 = \begin{pmatrix} \ell_1 & \cdots & \ell_L \end{pmatrix}$ an $M \times L$ *object-label* matrix, where each object is represented as a row-vector.

As will be discussed later, in our experimentation, movies are regarded as objects and terms (word) in their plots as features. In addition, each movie is assigned a set of genre-labels as its multi-label.

Since each object is often represented as a high-dimensional feature vector, it would be required to compress the matrix X_1. Moreover, although the number of possible labels would be less than that of features, the matrix X_2 also tends to be sparse because each object has only a few labels in general. We, therefore, try to compress both X_1 and X_2 by means of *Nonnegative Matrix Factorization* [2].

More formally speaking, X_1 and X_2 are approximated as follows:

$$X_1 \approx \left(\tilde{f}_1 \ \cdots \ \tilde{f}_K\right) V, \quad \text{where } V = (v_{ij})_{ij},\ f_j \approx \sum_{i=1}^{K} v_{ij}\tilde{f}_i$$

$$X_2 \approx \left(\tilde{\ell}_1 \ \cdots \ \tilde{\ell}_{K_L}\right) W_L, \quad \text{where } W_L = (w^L_{ij})_{ij},\ \ell_j \approx \sum_{i=1}^{K_L} w^L_{ij}\tilde{\ell}_i.$$

It is noted here that $\left(\tilde{f}_1 \ \cdots \ \tilde{f}_K\right)$ is a compressed representation of X_1 and $\left(\tilde{\ell}_1 \ \cdots \ \tilde{\ell}_{K_L}\right)$ that of X_2. As has been stated previously, we especially try to use the latter matrix $\left(\tilde{\ell}_1 \ \cdots \ \tilde{\ell}_{K_L}\right)$ as a part of the former $\left(\tilde{f}_1 \ \cdots \ \tilde{f}_K\right)$ in order to associate the label-information with the feature-information in our approximation process. That is, assuming a $(K_F = K - K_L) \times N$ coefficient matrix V_F and a $K_L \times N$ coefficient matrix V_L, we try to perform approximations such that

$$X_1 \approx \left(\tilde{f}_1 \ \cdots \ \tilde{f}_{K_F}\ \tilde{\ell}_1 \ \cdots \ \tilde{\ell}_{K_L}\right) \begin{pmatrix} V_F \\ V_L \end{pmatrix} = \left(\tilde{f}_1 \ \cdots \ \tilde{f}_{K_F}\right) V_F + \left(\tilde{\ell}_1 \ \cdots \ \tilde{\ell}_{K_L}\right) V_L,$$

$$X_2 \approx \left(\tilde{\ell}_1 \ \cdots \ \tilde{\ell}_{K_L}\right) W_L.$$

Note that the original column-vector f_i of X_1 is approximated by a linear combination of basis vectors $\tilde{f}_j$ in $F = (\tilde{f}_1 \ \cdots \ \tilde{f}_{K_F})$ and $\tilde{\ell}_j$ in $L = (\tilde{\ell}_1 \ \cdots \ \tilde{\ell}_{K_L})$ which are respectively *unaffected* and *affected* by label-compression.

In order to obtain a certain degree of quality in the approximation process, we have to care a balance between feature and label-compressions. Following [1], we take into account *Frobenius Norm* of the original matrices X_1 and X_2 and try to solve the following optimization (minimization) problem:

$$\min \frac{\left\| X_1 - \left(F|L\right) \begin{pmatrix} V_F \\ V_L \end{pmatrix} \right\|_F^2}{\|X_1\|_F^2} + \frac{\|X_2 - LW_L\|_F^2}{\|X_2\|_F^2},$$

where $\|X_1\|_F$ and $\|X_2\|_F$ can be treated as constants.

Based on a similar discussion to the standard formulation of NMF [2], we can obtain a set of multiplicative update rules for the optimization problem [1]. With element-wise expressions and $\lambda = \|X_1\|_F^2 / \|X_2\|_F^2$, we have

$$(L)_{ij} \leftarrow (L)_{ij} \times (S)_{ij}, \tag{1}$$

where $(S)_{ij}$ is given by

$$\frac{1}{(S)_{ij}} = \frac{(LV_LV_L^T + FV_FV_L^T)_{ij}}{(X_1V_L^T + \lambda X_2W_L^T)_{ij}} + \lambda \frac{(LW_LW_L^T)_{ij}}{(X_1V_L^T + \lambda X_2W_L^T)_{ij}} \tag{2}$$

for L defining the shared subspace. And, for $V = \begin{pmatrix} V_F \\ V_L \end{pmatrix}$, W_L and F, we have

$$(V)_{ij} \leftarrow (V)_{ij} \frac{(\big(F|L\big)^T X_1)_{ij}}{(\big(F|L\big)^T \big(F|L\big) V)_{ij}}, \tag{3}$$

$$(W_L)_{ij} \leftarrow (W_L)_{ij} \frac{(L^TX_2)_{ij}}{(L^TLW_L)_{ij}} \qquad \text{and} \tag{4}$$

$$(F)_{ij} \leftarrow (F)_{ij} \frac{(X_1V_F^T)_{ij}}{(LV_LV_F^T + FV_FV_F^T)_{ij}}. \tag{5}$$

2.2 Predicting Unknown Labels of Query

Based on the matrix factorization discussed above, we can predict a multi-label of a given query. We assume our query q with unknown-label is given as just an N-dimensional (feature) vector, that is, $q = (q_i)_{1 \leq i \leq N}$. A prediction about labels of q can be performed by computing a coefficient vector for the basis vectors $\tilde{\ell}_j$ reflecting label-information of objects in the factorization process. More precisely speaking, the query can be represented as

$$q = \sum_{i=1}^{N} q_i f_i, \qquad \text{where} \;\; f_i \approx \sum_{j=1}^{K_F} v_{ji}^F \tilde{f}_j + \sum_{j=1}^{K_L} v_{ji}^L \tilde{\ell}_j.$$

Then we have

$$q \approx \sum_{j=1}^{K_F} \left(\sum_{i=1}^{N} q_i v_{ji}^F \right) \tilde{f}_j + \sum_{j=1}^{K_L} \left(\sum_{i=1}^{N} q_i v_{ji}^L \right) \tilde{\ell}_j.$$

Thus, the (K_L-dimensional) coefficient vector for $\tilde{\ell}_j$ can be given by V_Lq.

After the approximation, each object x is represented by its corresponding row-vector v_x^T in the compressed matrix $L = \big(\tilde{\ell}_1 \;\cdots\; \tilde{\ell}_{K_L} \big)$ reflecting their original label-information. Therefore, a distance between v_x and V_Lq provides us a hint about which labels the query would have. If the vectors are close enough, q seems to have labels similar to those of x. In other words, we can evaluate a *farness/closeness* between labels of x and q by defining an adequate distance function for those vectors. In the following discussion, we assume a distance function, $dist_L$, based on *cosine similarity* between vectors. That is, for an object x and a query q, it is defined as $dist_L(x, q) = 1 - \frac{v_x^T V_L q}{||v_x||\; ||V_L q||}$.

2.3 Evaluating Similarity among Features of Objects

As is similar to the case of labels, we can evaluate similarity among features of objects based on the compressed matrix $(F|L)$. However, since the matrix is affected by not only the feature compression but also the label compression, it would not be adequate for evaluating only similarity of features. For our evaluation of feature similarity, therefore, we try to approximate the original matrix X_1 into an $M \times K_T$ matrix H_{X_1} with the standard NMF such that $X_1 \approx H_{X_1} W_F$, where each object x after the compression is given as its corresponding row-vector h_x^T in H_{X_1}. Therefore, we can evaluate similarity between features of x and q by computing a distance between h_x and $W_F q$, denoted by $dist_F(x, q)$.

3 Extracting Formal Concepts with Unexpected Multi-Label Objects for Query

Towards a recommendation-oriented information retrieval, we present in this section our method for finding formal concepts whose extents include some unexpected objects for a given query. The reason why we try to detect formal concepts is that the extent of a concept can be explicitly interpreted by its intent. That is, the intent provides us a clear explanation why those objects are grouped together. By extracting various concepts, we can therefore obtain interesting object clusters from multiple viewpoints.

3.1 Unexpected Objects Based on Predicted Multi-Label of Query

We first present our notion of unexpectedness of objects for a given query. Especially, we propose here an objective definition for the notion.

As has been discussed, we can *implicitly* predict a multi-label of a given query q with unknown-label. More precisely, we can measure a farness/closeness between labels of an object x and the query. In addition, we can evaluate similarity of features between x and q. Suppose here that we find both x and q have similar or relevant features. In such a case, it would be plausible that we expectedly consider they also have similar/relevant multi-labels. However, if we observe their labels are far from each other, we seem to find some unexpectedness of x for q because they have distant multi-labels even though their features are similar.

With the distance functions, $dist_F$ in the feature-subspace and $dist_L$ in the label-subspace, we can formalize this kind of unexpectedness of objects for a given query q with unknown-label.

Definition 1. (Unexpected Object for Query)
For an object x, if x satisfies the following two constraints, then x is said to be *unexpected* for the query q:

Relevance/Similarity of Features : x is relevant/similar to q in the feature-subspace, that is, $dist_F(x, q) \leq \delta_F$, and

Farness of Multi-Labels : x and q are distant from each other in the label-subspace, that is, $dist_L(x, q) \geq \delta_L$,

where δ_F (> 0) and δ_L (> 0) are user-defined parameters. ■

3.2 Extracting Formal Concepts with Unexpected Objects

Constructing Formal Context for Query : Suppose the original object-feature matrix X_1 is approximated as $X_1 \approx H_{X_1} W_F$ with the standard NMF, where H_{X_1} is regarded as a compressed representation of X_1. It is recalled that the i-th object x_i in X_1 is given as the i-th row-vector $v_i^T = (v_{i1} \ \cdots \ v_{iK_T})$ in H_{X_1}, where the j-th element v_{ij} is the value of the feature f_j for x_i.

In order to extract formal concepts with unexpected objects for a given query q, we first define a formal context $\mathcal{C}_q$ consisting of the objects relevant to q. Formally speaking, with the parameter δ_F, the set of relevant objects is defined as $\mathcal{O}_q = \{x_i \mid x_i \in \mathcal{O},\ dist_F(x_i, q) \leq \delta_F\}$. The set of features (attributes), $\mathcal{F}_q$, is simply defined as $\mathcal{F}_q = \{f_1, \ldots, f_{K_T}\}$. Moreover, introducing a parameter θ as a threshold for feature values, we define a binary relation $\mathcal{R}_q \subseteq \mathcal{O}_q \times \mathcal{F}_q$ as

$$\mathcal{R}_q = \{(x_i, f_j) \mid x_i \in \mathcal{O}_q, v_i^T = (v_{i1} \ \ldots \ v_{iK_T}) \text{ in } H_{X_1} \ \text{ and } \ v_{ij} \geq \theta\}.$$

Thus, our formal context is given by $\mathcal{C}_q = \langle \mathcal{O}_q, \mathcal{F}_q, \mathcal{R}_q \rangle$.

Evaluating Formal Concepts : It is obvious that for each formal concept in the context $\mathcal{C}_q$, its extent consists of only objects relevant to q. The extent, however, does not always have unexpected ones. For a purpose of recommendation, since it would be desirable to involve some unexpected objects, we need to evaluate formal concepts in $\mathcal{C}_q$ taking the farness of multi-labels into account.

Let us consider a concept in $\mathcal{C}_q$ with its extent E. We evaluate the concept by the average distance between each object in E and the query in the label-subspace. That is, our evaluation function, *eval*, is defined as $eval(E) = \frac{\sum_{x \in E} dist_L(x,q)}{|E|}$.

Although our formal context consists of only objects relevant to the query, we could have many concepts in some cases. We, therefore, try to obtain concepts with the top-N largest evaluation values.

Interpreting Intents of Formal Concepts : Our formal context is constructed from the matrix H_{X_1} which is a compressed representation of the original object-feature matrix X_1 approximated as $X_1 \approx H_{X_1} W_F$ with the standard NMF. That is, the intent of a concept in the context is given as a set of *compressed* features interpretable in terms of original features. The relationship among compressed and original features is represented as the matrix W_F. Each compressed feature is given as a row-vector in W_F in which larger components can be regarded as relevant original features. When we, therefore, interpret the intent of a concept, it is preferable to show relevant original features as well as each compressed one in the intent. As a simple way of dealing with that, assuming a (small) integer k, we can present original features corresponding to the k largest components for each compressed feature.

3.3 Algorithm for Extracting Top-N Formal Concepts

We present our algorithm for extracting target formal concepts.

```
[Input]   $\mathcal{C}_q = (\mathcal{O}_q, \mathcal{F}_q, \mathcal{R}_q)$ : a formal context obtained for a query $q$
          $N$ : a positive integer for top-$N$
[Output]  $\mathcal{FC}$ : the set of formal concepts with the top-$N$ largest evaluation values
--------------------------------------------------------------------------------------------------
procedure Main($\mathcal{C}_q$, $N$) :
  $\mathcal{FC} = \emptyset$ ;
  $\alpha = 0.0$ ; // the $N$-th (tentative) evaluation value of concepts in $\mathcal{FC}$
  Fix a total order $\prec$ on $\mathcal{O}_q$ such that for any $x_i, x_j \in \mathcal{O}_q$, $x_i \prec x_j$ if $dist_L(x_i, q) \leq dist_L(x_j, q)$ ;
  while $\mathcal{O}_q \neq \emptyset$ do
    begin
      $x = head(\mathcal{O}_q)$ ;
      $\mathcal{O}_q = (\mathcal{O}_q \setminus \{x\})$ ; // removing $x$ from $\mathcal{O}_q$
      FCFind($\{x\}$, $\emptyset$, $\mathcal{O}_q$) ; // $\mathcal{O}_q$ as candidate objects
    end
  return $\mathcal{FC}$ ;

--------------------------------------------------------------------------------------------------
procedure FCFind($P$, $PrevExt$, $Cand$) :
  $FC = (Ext = P'', P')$ ; // computing FC
  if $\exists x \in (Ext \setminus PrevExt)$ such that $x \prec tail(P)$ then
    return; // found to be duplicate formal concept
  endif
  Update $\mathcal{FC}$ adequately so that it keeps concepts with top-$N$ largest evaluation values found so far ;
  $\alpha$ = the $N$-th (tentative) evaluation value of concepts in $\mathcal{FC}$;
  while $Cand \neq \emptyset$ do
    begin
      $x = head(Cand)$ ;
      $Cand = (Cand \setminus \{x\})$ ; // removing $x$ from $Cand$
      $NewCand = (Cand \setminus Ext)$ ; // new candidate objects
      if $NewCand = \emptyset$ then continue ;
      if $eval(P \cup \{x\} \cup NewCand) < \alpha$ then continue ; // branch-and-bound pruning
      FCFind($P \cup \{x\}$, $Ext$, $NewCand$) ;
    end
```

Fig. 1. Algorithm for Finding Top-N Formal Concepts with Unexpected Objects

Let $\mathcal{C}_q = \langle \mathcal{O}_q, \mathcal{F}_q, \mathcal{R}_q \rangle$ be a formal context constructed for a given query q. As the basic strategy, generating extents of concepts, we explore object sets along the *set enumeration tree*, rooted by $\emptyset$, based on a total ordering $\prec$ on $\mathcal{O}_q$. More concretely speaking, we expand a set of objects P into $Px = P \cup \{x\}$ by adding an object x succeeding to $tail(P)$ and then compute $((Px)'', (Px)')$ to obtain a formal concept, where we refer to the last (tail) element of P as $tail(P)$. Such an object x to be added is called a *candidate* and is selected from $cand(P) = \{x \mid x \in (\mathcal{O}_q \setminus P''),\ tail(P) \prec x\}$. Initializing P with $\emptyset$, we recursively iterate this process in depth-first manner until no P can be expanded.

Our target concepts must have the top-N largest evaluation values. Let us assume the objects x_i in $\mathcal{O}_q$ are sorted in *ascending order* of $dist_L(x_i, q)$. Based on the ordering, along our depth-first expansion process of concepts, the evaluation values of obtained concepts increase *monotonically*. It should be noted here that for a set of objects $P \subseteq \mathcal{O}_q$, the extent E of each concept obtained by expanding P is a subset of $P'' \cup cand(P)$. Due to the monotonicity of evaluation values, therefore, $eval(P'' \cup cand(P))$ gives an upper bound we can observe in our expansion process from P. This means that if we find $eval(P'' \cup cand(P))$ is less than the tentative N-th largest evaluation value of concepts found so far, there is no need to expand P because we never meet any target concept by the expansion. Thus, a branch-and-bound pruning is available in our search process.

A pseudo-code of the algorithm is presented in Figure 1. The head element of a set S is referred to as $head(S)$. Although we skip details due to space limitation, the code incorporates a mechanism for avoiding duplicate generations of the same concept, as **if** statement at the beginning of **procedure** FCFind.

4 Experimental Result

In this section, we present our experimental result with our system.

We have prepared a movie dataset consisting of 17,000 movies with their plots and genres. Our dataset has been created from *CMU Movie Summary Corpus* [13] [1]. After preprocessing, the plot of each movie is represented as a boolean vector of 6,251 feature terms with medium frequency. That is, our movie-plot matrix X_P has the dimension of $17,000 \times 6,251$. Moreover, each movie is assigned some of 364 genre-labels as its multi-label. Then, our movie-label matrix X_L is given as a boolean matrix with the dimension of $17,000 \times 364$. Applying Nonnegative Shared Subspace Method, we have compressed X_P into a $17,000 \times 500$ matrix $(F|L)$, where dimensions of F and L are $17,000 \times 450$ and $17,000 \times 50$, respectively, and L is also a compressed matrix of X_L.

In addition, as candidates of our queries, we have also prepared 783 movies with only their plots, hiding their genres. Thus, our system is given a 6,251-dimensional boolean vector as a query obtained from those candidates.

Example of Extracted Formal Concept for "*Shark Bait* (2006)"
For a query vector obtained from the plot of a candidate movie "*Shark Bait*", we present here a formal concept actually detected by our system.

"*Shark Bait*" is a computer animated family movie released in 2006. The story is about Pi, the main fish character, his relatives and friends while fighting against a mean tiger shark terrorizing Pi's reef community.

For the query vector (plot), an example of formal concept our system detected is shown in Figure 2.

Similarity of Movie Plots : The extent of the concept consists of 5 movies all of which are concerned with marine animals. For example, "*Jaws*" is a very famous movie about a great white shark wrecking havoc in a beach resort. "*Open Water*" is a suspense movie based on a real story about a couple of scuba divers left behind in the shark-infested sea due to an accident. Moreover, "*Free Willy*", a family-oriented adventure movie, is about the friendship between a street child and a killer whale in a local amusement park. As the intent shows, all of them are commonly associated with 5 features compressed with NMF each of which is related to several relevant feature terms used in the original plots. Actually, we can observe a certain degree of similarity among the plots of the movies in the extent. It is, furthermore, easy to see that they are also similar to the query plot.

[1] `http://www.cs.cmu.edu/ark/personas/`

Extent (movie : genres as multi-labels)	
Jaws (1975)	: Thriller
Open Water (2003)	: Thriller, Natural Horror Films, Horror, Indie, Drama, Suspense, Disaster
Jaws 3-D (1983)	: Thriller, Natural Horror Films, Horror, Action/Adventure, Mystery, Suspense, Action
Free Willy (1993)	: Adventure, Children's/Family, Animal Picture, Drama, Family Film, Family-Oriented Adventure
Help! I'm a Fish (2000)	: Science Fiction, Adventure, Animation, Children's, Fantasy, Drama, Family Film

Intent (compressed feature : related original feature (words))	
F1	: tank, fuel, leak, gasoline, oxygen
F2	: surface, beneath, descend, underwater, flood
F3	: ocean, Pacific, liner, Ocean, drift
F4	: swim, swimming, underwater, fish, dive
F5	: owe, loan, borrow, shark, fund

Fig. 2. Example of Extracted Formal Concept

Farness of Multi-Labels : As is shown in the figure, the movies in the extent have various multi-labels. More precisely speaking, the movies are listed in descending order of distance between the (implicitly) predicted multi-label of the query and that of each movie. That is, upper movies are expected to have multi-labels further from that of the query as unexpected ones. On the other hand, multi-labels of lower movies would be similar to that of the query. According to our problem setting, although the correct multi-label of the query was intensionally hidden, its actual genre labels are "Family Film" and "Animation". It is easy to see that the lowest movie, "*Help! I'm a Fish*", is categorized into genres very similar to the actual genres of the query. In other words, the multi-label of the query can reasonably be predicted with the help of Nonnegative Shared Subspace Method. As the result, based on our evaluation function for formal concepts, we can find some movies in the extent with unexpected (further) genre labels, like "*Jaws*" and "*Open Water*". Inspired by such a concept, users could newly try and enjoy some movies with those unexpected genres. Thus, our method has an ability to stimulate us to try unexperienced movies based on unexpectedness of multi-labels.

Quality of Multi-Label Prediction : Quality of multi-label prediction would be important in our method because our unexpectedness is based on the prediction. We here observe quality of prediction by comparing two object rankings, one is based on predicted multi-labels of the query and the other based on its correct multi-label (hidden in our retrieval process). More concretely speaking, for the former, each movie in the dataset is ranked in ascending order of similarity between its multi-label and the predicted label of the query. The obtained ranking is referred to as R_{pred}. Similarly, we also rank each movie in ascending order of similarity between its multi-label and the correct label of the query. We refer to the ranking as $R_{correct}$. Then we can compute the Spearman's rank correlation coefficient between R_{pred} and $R_{correct}$. If we can observe a certain degree of correlation, we can expect that our prediction would be reasonable.

For the above example, we actually have the coefficient of 0.35 showing a weak positive correlation. As has been mentioned previously, a prediction method has been discussed in [17], where prediction is performed based on a subspace defined as a real line. For comparison, the correlation coefficient between the ranking according to the previous method and $R_{correct}$ is 0.03 showing little correlation. Although the value of 0.35 seems to be a little bit small, it can be increased to 0.84 when we focus on Top-10% ranked $(1,700)$ objects in the ranking. It is noted here that our main purpose of prediction is just to identify objects whose multi-labels are far from that of the query. In this sense, precise lower ranks in R_{pred} are not matter. Therefore, we consider that the prediction of multi-labels in our method can work well for our purpose.

5 Concluding Remark

We discussed our method of finding interesting formal concepts with unexpected objects for a given query with no label. We defined our unexpected object for the query as object which is similar to the query in the feature space, but is dissimilar in the label space. In order to predict a multi-label of the query, the original object-feature and object-label matrices are simultaneously factorized by means of Nonnegative Shared Subspace Method, where the obtained subspace associates the label-information with the feature-information of the original matrices. Our retrieval task was formalized as a problem of enumerating every formal concept with Top-N evaluation values w.r.t. unexpectedness.

At the moment, we still leave quantitative evaluation of our method. As the unexpectedness in [14] has been quantitatively evaluated for another movie dataset according to [15], we can attempt a similar comparison. In addition, it would be worth verifying actual usefulness of our system through user trials.

We also need to investigate a prediction method of multi-labels. In our current framework, although the basis of label space is assumed to be a part of the basis of feature space. In order to further improve quality of multi-label prediction, it would be required to carefully distinguish two classes of labels, ones which can be well explained in terms of features and the others. By a correlation analysis for labels and features, we can define an adequate regularization term in the objective function for our matrix factorization. This kind of analysis is also very important in the framework of *feature association* which identifies reasonable similarities among features in different data matrices [11].

Moreover, a recommendation system based on query-specific FCA-based biclustering has been proposed in [7]. We need to clarify relationship between the method and ours.

References

1. Gupta, S.K., Phung, D., Adams, B., Tran, T. and Venkatesh, S.: Nonnegative Shared Subspace Learning and Its Application to Social Media Retrieval, Proc.

of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining - KDD'10, pp. 1169 – 1178 (2010).

2. Lee, D. D. and Seung, H. S.: Algorithms for Non-negative Matrix Factorization, Proc. of NIPS 2000, pp. 556 – 562 (2000).
3. Kumar, C. A., Dias, S. M. and Vieira, N. J.: Knowledge Reduction in Formal Contexts Using Non-negative Matrix Factorization, Mathematics and Computers in Simulation, 109(C), pp. 46 – 63, Elsevier (2015).
4. Kuznetsov, S. O. and Makhalova, T: On Interestingness Measures of Formal Concepts, Information Sciences, 442(C), pp. 202 – 219, Elsevier (2018).
5. Akhmatnurov, M. and Ignatov, D. I.: Context-Aware Recommender System Based on Boolean Matrix Fatorisation, Proc. of the 12th International Conference on Concept Lattices and Their Applications - CLA'15, pp. 99 – 110 (2015).
6. Stumme, G., Taouil, R., Bastide, Y., Pasquier, N. and Lakhal, L.: Computing Iceberg Concept Lattices with TITANIC, Data & Knowledge Engineering, 42(2), pp. 189 – 222, Elsevier (2002).
7. Alqadah, F, Reddy, C. K., Hu, J. and Alqadah, H. F.: Biclustering Neighborhood-Based Collaborative Filtering Method for Top-N Recommender Systems, Knowledge Information Systems, 44(2), pp. 475 – 491, Springer (2015).
8. Sun, L., Ji, S. and Ye, J.: Hypergraph Spectral Learning for Multi-Label Classification, Proc. of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining - KDD'08, pp. 668 – 676 (2008).
9. A. Takano, Y. Niwa, S. Nishioka, M, Iwayama, T. Hisamitsu, O. Imaichi, and H, Sakurai: Information Access Based on Associative Calculation, Proc. of SOFSEM 2000: Theory and Practice of Informatics, LNCS 1963, pp. 187 – 201 (2000).
10. H. Zhai, M. Haraguchi, Y. Okubo, K. Hashimoto and S. Hirokawa: Shifting Concepts to Their Associative Concepts via Bridges, Proc. of the 9th International Conference on Machine Learning and Data Mining in Pattern Recognition, LNAI 7988, 586 – 600 (2013).
11. H. Zhai and M. Haraguchi: A Linear Algebraic Inference for Feature Association, Proc. of the 12th Int'l Conf. on Knowledge, Information and Creativity Support Systems, IEEE Conference Publishing Services, pp. 102 – 107 (2017).
12. Ganter, B. and Wille, R.: Formal Concept Analysis - Mathematical Foundations, 284 pages, Springer (1999).
13. Bamman, D., O'Connor, B. and Smith, N. A.: Learning Latent Personas of Film Characters, Proc. of the 51st Annual Meeting of the Association for Computational Linguistics, pp. 352 – 361 (2013).
14. Adamopoulos, P. and Tuzhilin, A.: On Unexpectedness in Recommender Systems: Or How to Better Expect the Unexpected, ACM Transactions on Intelligent Systems and Technology, 5(4), Article No. 54 (2015).
15. Murakami, T., Mori, K. and Orihara, R: Metrics for Evaluating the Serendipity of Recommendation Lists, New Frontiers in Artificial Intelligence, JSAI 2007 Conference and Workshops, Revised Selected Papers, LNAI-4914, pp. 40 – 46 (2008).
16. Herlocker, J. L., Konstan, J. A., Terveen, L. G. and Riedl, J. T.: Evaluating Collaborative Filtering Recommender Systems, ACM Transactions on Information Systems, 22(1), pp. 5 – 53 (2004).
17. Okubo, Y., Haraguchi, M. and Liu, H.: Finding Interesting Formal Concepts with Unexpected Objects with respect to Multi-Labels, Proc. of the 6th Asian Conference on Information Systems - ACIS 2017, pp. 98 – 105 (2017).

Steps Towards Achieving Distributivity in Formal Concept Analysis

Alain Gély[1], Miguel Couceiro[2], and Amedeo Napoli[2]

[1] Université de Lorraine, CNRS, LORIA, F-57000 Metz, France
[2] Université de Lorraine, CNRS, Inria, LORIA, F-54000 Nancy, France
{alain.gely,miguel.couceiro,amedeo.napoli}@loria.fr

Abstract. In this paper we study distributive lattices in the framework of Formal Concept Analysis (FCA). The main motivation comes from phylogeny where biological derivations and parsimonious trees can be represented as median graphs. There exists a close connection between distributive lattices and median graphs. Moreover, FCA provides efficient algorithms to build concept lattices. However, a concept lattice is not necessarily distributive and thus it is not necessarily a median graph.
In this paper we investigate possible ways of transforming a concept lattice into a distributive one, by making use Birkhoff's representation of distributive lattices. We detail the operation that transforms a reduced context into a context of a distributive lattice. This allows us to reuse the FCA algorithmic machinery to build and to visualize distributive concept lattices, and then to study the associated median graphs.

1 Context and Motivations

Formal Concept Analysis (FCA) has proved to be an effective tool in data analysis and knowledge discovery in several application domains [10,14]. Concept lattices provide a valuable support for several tasks, such as classification, information retrieval and pattern recognition. Besides lattices, trees and their extensions [4,5,13] are used in biology, notably, in phylogeny, for modeling inter-species filiations. In this domain, one of the main problems is to find evolution trees for representing existing species from accessible DNA fragments. When several trees are leading to the same inter-species filiations, the preferred ones are the most "parsimonious", i.e. the number of modifications such as mutations for example, is minimal for the considered species. However, several possible parsimonious trees may exist. Such a situation arises with inverse or parallel mutations, e.g., when a gene goes back to a previous state or the same mutation appears for two non-linked species. This asks for a generic representation of such a family of trees.

Bandelt [2,3] proposes the notion of *median graph* to overcome this issue, since he noticed that a median graph is capable of encoding all parsimonious trees. A median graph is a connected graph such that for any three vertices a, b, c, there is exactly one vertex x which lies on a shortest path between each pair of vertices in $\{a, b, c\}$. Alternatively, median graphs can be thought of as a

© Dmitry I. Ignatov, Lhouari Nourine (Eds.): CLA 2018, pp. 105–116,
ISBN 978–80–244–5328–6, Department of Computer Science, Palacký University
Olomouc, 2018.

generalization distributive lattices [8,15]. However, the extraction of such structures directly from data remained unaddressed.

Uta Priss [19,20] made a first attempt to use algorithmic machinery of FCA and the links between distributive lattices and median graphs, to analyze phylogenetic trees. However, not every concept lattice is distributive, and thus FCA alone does not necessarily outputs median graphs. In [20] Uta Priss sketches an algorithm to convert any lattice into a median graph. The key step is to transform any lattice into a distributive lattice.

In this article, we propose an algorithm supporting such a transformation that minimizes the changes introduced to the original lattice. Using the context of an initial concept lattice as input, the algorithm outputs the context of a distributive lattice, without necessarily building the lattice. Our approach relies on Birkhoff's representation of distributive lattices [6,7]. Moreover, we illustrate our approach with a generic example that reveals the difficulties of transforming of a concept lattice into a distributive lattice. We do not settle this issue entirely, but we propose major steps and an approach towards its solution.

The paper is organized as follows. In Sections 2 and 3 we recall the basic background and notation as well as some key results on distributive lattices. The transformation algorithm is presented in Section 4 and we discuss the strengths and limitations of our approach in Section 5.

2 Definitions and Notations

In this section we recall basic notions and notation needed throughout the paper. We will mainly adopt the formalism of [14], and we refer the reader to [11,12] for further background.

2.1 Partially Ordered Sets, Lattices and Homomorphisms

A *partially ordered set* (or *poset* for short) is a pair $(P, \leqslant)$ where P is a set and $\leqslant$ is a *partial order* on P, that is, a reflexive, antisymmetric and transitive binary relation on P. A poset $(P', \leqslant')$ is a *subposet* of $(P, \leqslant)$ if $P' \subseteq P$ and $\leqslant' \subseteq \leqslant$. For a subset $X \subseteq P$, let $\downarrow X = \{y \in P : y \leqslant x \text{ for some } x \in X\}$ and $\uparrow X = \{y \in P : x \leqslant y \text{ for some } x \in X\}$. If $X := \{x\}$, we use the notation $\downarrow x$ and $\uparrow x$ instead of $\downarrow \{x\}$ and $\uparrow \{x\}$, respectively. In this paper, we will only consider finite posets $(P, \leqslant)$ and, when there is no danger of ambiguity, we will refer to them by their underlying universes P.

A set $X \subseteq P$ is a (*poset*) *ideal* (resp. *filter*) if $X = \downarrow X$ (resp. $X = \uparrow X$). If $X = \downarrow x$ (resp. $X = \uparrow x$) for some $x \in P$, then X is said to be a *principal* ideal (resp. filter) of P. For $x, y \in P$, the greatest element of $\downarrow x \cap \downarrow y$ (resp. least element $\uparrow x \cap \uparrow y$) if it exists, is called the *infimum* (resp. *supremum*) of x and y. A *lattice* is a poset $(L, \leqslant)$ such that the infimum and the supremum of any pair $x, y \in L$ exist, and they are denoted respectively by $x \wedge y$ and $x \vee y$. A subset $X \subseteq L$ is a *sublattice* of L if for every $x, y \in X$ we have that $x \wedge y, x \vee y \in X$. As for posets, we will only consider finite lattices $(L, \leqslant)$ and we will refer to them by their underlying universes L.

In this finite setting, posets and lattices can be represented and clearly visualized by their Hasse-diagrams [12]. Also, the notions of infimum and supremum naturally extend from pairs to any subset of elements of a given lattice L. In this way, the notions of $\wedge$- and $\vee$-irreducible elements (that constitute the building blocks of lattices) can be defined as follows. For $x \in L$, let $x^* = \bigwedge(\uparrow x \backslash \{x\})$ and $x_* = \bigvee(\downarrow x \backslash \{x\})$. Then $x \in L$ is said to be a $\wedge$-*irreducible* element of L if $x \neq x^*$. Dually, x is said to be a $\vee$-*irreducible* element of L if $x \neq x_*$. We will denote the set of $\wedge$-irreducible elements and $\vee$-irreducible elements of L by $M(L)$ and $J(L)$, respectively. Observe that both $M(L)$ and $J(L)$ are posets when ordered by $\leqslant$.

We now recall the notions of poset and lattice homomorphisms.
Let $(P, \leqslant)$ and $(P', \leqslant')$ be two posets. A mapping $f\colon P \to P'$ is said to be a (*poset*) *homomorphism* if $x \leqslant y$ implies $f(x) \leqslant' f(y)$. In addition, if $f\colon P \to P'$ is injective (one-to-one), then it is called a (*poset*) *embedding.* If it is a bijection and an embedding such that, for every $x', y' \in P'$, $x' \leqslant' y'$ implies $f^{-1}(x') \leqslant f^{-1}(y')$, then it is called a (*poset*) *isomorphism.*

In the case of lattices, the notions of homomorphism, embedding and isomorphism become more stringent. Let $(L, \wedge, \vee)$ and $(L', \wedge', \vee')$ be two lattices. A mapping $f\colon L \to L'$ is said to be a (*lattice*) *homomorphism* if $f(x \wedge y) = f(x) \wedge' f(y)$ and $f(x \vee y) = f(x) \vee' f(y)$. In addition, if $f\colon L \to L'$ is injective, then it is called a (*lattice*) *embedding.* If it is a bijection and it is an embedding such that f^{-1} is also an embedding, then it is called a (*lattice*) *isomorphism.* When it is clear from the context, we will drop "(poset)" and "(lattice)" and simply refer to homomorphism, embedding and isomorphism.

It is noteworthy that the image $f(L)$ of a homomorphism $f\colon L \to L'$ is a sublattice of L', and that two isomorphic lattices have the same Hasse diagram. In particular, two lattices L and L' are isomorphic if and only if both (1) $J(L)$ and $J(L')$, and (2) $M(L)$ and $M(L')$ are isomorphic. In the case of distributive lattices, Birkhoff [7] showed that (1) suffice to guarantee that L and L' are isomorphic ($(J, \leqslant)$ and $(M, \leqslant)$ are isomorphic). The latter result is key ingredient in *Birkhoff's representation of distributive lattices* that we will discuss in Section 3, and that we will use in Section 4 to devise an algorithm to modify any finite lattice into an "optimal" distributive lattice containing it.

2.2 Formal Concept Analysis

Reduced Contexts, Concepts and Concept Lattices. We denote by $C = (O, A, I)$ a formal context where O is a set of objects, A a set of attributes and I an incidence relation between objects and attributes. In phylogenetic data, objects are usually species, attributes are mutations, and $(o, a) \in I$ –or oIa– when mutation a is spotted in species o.

Definition 1 (Galois connections). *For a set $X \subseteq O$, $Y \subseteq A$ we define:*

$$X' = \{y \in A \mid xIy \textit{ for all } x \in X\}$$
$$Y' = \{x \in O \mid xIy \textit{ for all } y \in Y\}$$

Then a formal concept is a pair (X, Y), where $X \subseteq O$, $Y \subseteq A$ and $X' = Y$ and $Y' = X$. X is the extent and Y is the intent of the concept. The set of all formal concepts ordered by inclusion of the extents –dually the intents– denoted by $\leqslant$ generates the concept lattice of the context $C = (O, A, I)$.

For $o \in O$, $\gamma o = (o'', o')$ denotes the concept introducing object o. For $a \in A$, $\mu a = (a', a'')$ denotes the concept introducing attribute a.

A *clarified context* is a context such that $x' = y'$ implies $x = y$ for any element of O and any element of A. Moreover, a clarified context is *reduced* iff it contains:

- no vertex $x \in O$ such that $x' = X'$ with $X \subseteq O$, $x \notin X$
- no vertex $x \in A$ such that $x' = X'$ with $X \subseteq A$, $x \notin X$

The reduced context is also called a *standard context.* Note that the standard context of lattice L is such that $O = J(L)$ and $A = M(L)$.

Arrow Relations.

Definition 2. *Let us consider a context (O, A, I), an object $o \in O$ and an attribute $a \in A$, then:*

- *$o \nearrow a$ iff $(o, a) \notin I$ and if $a' \subseteq x'$, $a' \neq x'$ then $(o, x) \in I$*
- *$o \swarrow a$ iff $(o, a) \notin I$ and if $o' \subseteq x'$, $o' \neq x'$ then $(x, a) \in I$*
- *$o \swarrow\!\!\!\!\nearrow a$ iff $o \nearrow a$ and $o \swarrow a$*

Stated differently, $o \swarrow a$ iff o' is maximal among all object intents which do not contain a. It can be shown that:

$$o \swarrow a \quad \Leftrightarrow \quad \gamma o \in J(L) \text{ and } \gamma o \wedge \mu a = (\gamma o)_*(\text{with } x_* = \bigvee(\downarrow x \backslash \{x\}))$$
$$o \nearrow a \quad \Leftrightarrow \quad \mu a \in M(L) \text{ and } \gamma o \vee \mu a = (\mu a)^*(\text{with } x^* = \bigwedge(\uparrow x \backslash \{x\}))$$

Arrow relations are related to irreducible elements in $J(L)$ and $M(L)$. In the following, we only consider arrow relations in reduced contexts.

An alternative equivalent definition of arrow relations is the following:

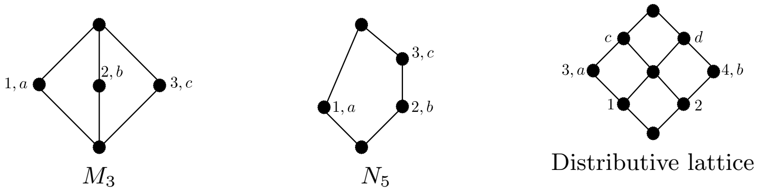
Definition 3. *Let L be a lattice, $j \in J(L)$ and $m \in M(L)$, then:*

- *$j \nearrow m$ iff $\mu m \in max(L \backslash \uparrow \gamma j)$ where $max(.)$ denotes the maximal elements.*
- *$j \swarrow m$ iff $\gamma j \in min(L \backslash \downarrow \mu m)$ where $min(.)$ denotes the minimal elements.*
- *$j \swarrow\!\!\!\!\nearrow m$ iff $j \nearrow m$ and $j \swarrow m$.*

$C = (J, M, I, \swarrow, \nearrow)$ is the reduced context with arrow relations. It can be represented by a table with (irreducible) objects in lines, (irreducible) attributes in columns, and in cell (j, m) (intersection of row j and column m):

- × if $j \leqslant m$ where $\leqslant$ is the partial ordering in the concept lattice,
- $\swarrow$ if $j \swarrow m$,
- $\nearrow$ if $j \nearrow m$,
- ⤢ if $j \swarrow m$ and $j \nearrow m$,
- otherwise an empty cell.

Fig. 1 shows three examples of reduced contexts with arrow relations $C = (J, M, I, \swarrow, \nearrow)$ and the corresponding concept lattices. The two first lattices on the left are respectively named M_3 and N_5 and they are the smallest non-distributive lattices. The third lattice on the right is a distributive lattice.



M_3

N_5

Distributive lattice

	a	b	c
1	×	⤢	⤢
2	⤢	×	⤢
3	⤢	⤢	×

	a	b	c
1	×	↙	⤢
2	⤢	×	×
3	↗	⤢	×

	a	b	c	d
1	×	⤢	×	×
2	⤢	×	×	×
3	×		×	⤢
4		×	⤢	×

Fig. 1. Three lattices and their reduced contexts with arrow relations.

3 Distributive Lattices and Their Representation

A lattice is *distributive* if $\wedge$ and $\vee$ are distributive one with respect to the over. Formally, a lattice L is distributive if for every $x, y, z \in L$, we have that one (or, equivalently, both) of the following identities holds:

$$(i)\ x \vee (y \wedge z) = (x \vee y) \wedge (x \vee z), \quad (ii)\ x \wedge (y \vee z) = (x \wedge y) \vee (x \wedge z).$$

Distributive lattices appear naturally in any classification task or as computation and semantic models; see, e.g., [11,12,16,17]. This is partially due to the fact that *any distributive lattice can be thought of as a sublattice of a power-set lattice, i.e., the set $\mathcal{P}(X)$ of subsets of a given set X*. This result is a corollary to Birkhoff's representation of distributive lattices that we will further discuss in Subsection 3.2.

3.1 Characterization of Distributive Lattice

The distributive property of lattices has been equivalently described in several ways. We recall a few useful characterizations that we will use in the following sections of the paper.

Theorem 1. *A lattice L is distributive if and only if one (or, equivalently, all) of the following conditions hold:*

1. $(x \wedge y) \vee (y \wedge z) \vee (z \wedge x) = (x \vee y) \wedge (y \vee z) \wedge (z \vee x)$;
2. *L does not contain neither N_5 nor M_3 as sublattice;*
3. *the reduced context of L with arrow relations contains exactly one double-arrow $\swarrow\!\!\!\nearrow$ in each row and in each column, and no other arrows.*

The first characterization establishes a correspondence between distributive lattices and median algebras. Indeed, a median algebra is a structure (M, m) where M is a nonempty set and $m : M^3 \to M$ is an operation, called *median operation*, that satisfies the following conditions $m(a, a, b) = a$ and $m(m(a, b, c), d, e) = m(a, m(b, c, d), m(b, c, e))$, for every $a, b, c, d, e \in M$. It is not difficult to see that if L is distributive, then $m(a, b, c) = (a \wedge b) \vee (b \wedge c) \vee (c \wedge a)$ is a median operation. The connection to *median graphs* was established by Avann [1] who showed that every median graph is the Hasse diagram of a median algebra (thought of as a semilattice). For further background see, e.g., [2].

The second characterization describes distributive lattices in terms of two forbbiden structures, namely, M_3 and N_5 (see Fig. 1) that are, up to isomorphism, the smallest non distributive lattices. The third characterization is given in terms of formal contexts and it is also illustrated in Fig. 1: neither M_3 nor N_5 are distributive since

- for M_3, there are two double arrows by row and column;
- for N_5, there is one double arrow by row and column, but additional simple arrows.

3.2 Distributive Lattices and Ideal Lattices

Let $(P, \leqslant)$ be a poset and consider the set $\mathcal{O}(P)$ of ideals of P, i.e.,

$$\mathcal{O}(P) = \{\bigcup_{x \in X} \downarrow x \mid X \subseteq P\}.$$

It is well-known that for every poset P, the set $\mathcal{O}(P)$ ordered by inclusion is a distributive lattice, called *ideal lattice* of P, and that two posets P and P' are isomorphic if and only if $\mathcal{O}(P)$ and $\mathcal{O}(P')$ are isomorphic as lattices. Furthermore, the poset of $\vee$-irreducible elements of $\mathcal{O}(P)$ is

$$J(\mathcal{O}(P)) = \{\downarrow x \mid x \in P\}$$

and it is (order) isomorphic to P.

Dually, we saw in Subsection 2.1 that for any lattice L the set $J(L)$ of $\vee$-irreducible elements of L is a poset ordered by inclusion, and that if two lattices L and L' are isomorphic, then $J(L)$ and $J(L')$ are also isomorphic (as posets). Moreover, for any lattice L the set $\mathcal{O}(J(L))$ of ordered ideals of $J(L)$ is a distributive lattice that contains an isomorphic copy of L as a subposet. In particular, if L is isomorphic to $\mathcal{O}(J(L))$, then L must be distributive. The representation theorem of Birkhoff [6] states that the converse is true.

Birkhoff's Representation Theorem 1 *Let L be a (finite) distributive lattice and $J(L)$. Then the mapping $x \rightarrow \downarrow x \cap J(L)$ is a (lattice) isomorphism from L to $\mathcal{O}(J(L))$.*

As immediate consequences we have that every (finite) distributive lattice can be thought of as a sublattice of a powerset lattice or, equivalently, as a lattice of ideals of a poset. Figure 2 illustrates the latter assertion: on the left is a poset P, and on the right is the lattice of ideals of P. For an arbitrary lattice L, not necessarily distributive, there may be several lattices such that their poset of $\vee$-irreducible elements are isomorphic but only one of them is a distributive lattice [9,18]. Our goal is to make use of the previous results to present an algorithmic approach that receives a lattice L as input, and outputs an "optimal" distributive lattice L_d such that $(J(L), \leqslant)$ is isomorphic to$(J(L_d), \leqslant_d)$. Here, by "optimal" it should be understood "with the least number of modifications" (notably, insertions).

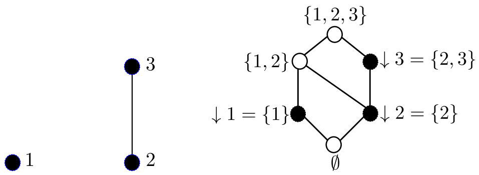


Fig. 2. Illustration of Birkhoff's Representation Theorem.

4 Proposal for Building a Distributive Lattice

From any lattice L, we want to obtain a distributive one L_d. Moreover, we want L_d to be "similar" to L. In this work, L_d is considered as similar to L if the posets of $\vee$-irreducible elements of L_d and of L are isomorphic. In this case, L can be embedded in L_d (L_d is a $\vee$-completion of L).

With this definition of "similar" (which can dually be applied to $\wedge$-irreducible elements), we can use Birkhoff Representation Theorem to compute L_d or its reduced context.

The main idea of algorithm 1 is to compute the context of L_d from the reduced context of L as input. Our approach relies on the underlying poset $(J, \leqslant)$ which is used to compute M_d.

Property 1. Algorithm 1 outputs the reduced context of the ideal lattice of $J(L)$.

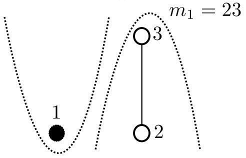
Proof. By construction, there is only one double-arrow by row and by column, and no other arrows. It follows that C_d is the context of a distributive lattice As discussed in section 2.1, this lattice is the ideal poset of $(J(L), \leqslant)$. It follows that $(J(L), \leqslant)$ and $(J(L_d), \leqslant_d)$ are isomorphic. □

Algorithm 1: Construction of context of distributive lattice.

```
Data: Reduced context $C(J, M, I)$
Result: Reduced context $C_d(J, M_d, I_d)$ of $(\mathcal{O}(J), \subseteq, \cap, \cup)$
$M_d \leftarrow \varnothing$
$I_d \leftarrow \varnothing$
foreach $j \in J$ do
    $\uparrow j \leftarrow \varnothing$
    foreach $i \in J$ do
        if $j' \subseteq i'$ then $\uparrow j \leftarrow \uparrow j \cup i$
    $M_d \leftarrow M_d \cup m_j$ // add a $\wedge$-irreducible element $m_j$ such that $j \swarrow\!\!\!\!\nearrow m_j$
    $X \leftarrow J \backslash \uparrow j$ // elements of poset $J$ which are not greater than $j$
    foreach $x \in X$ do
        $I_d \leftarrow I_d \cup (x, m_j)$
```

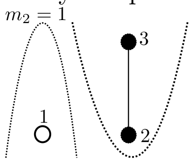
To illustrate the algorithm, we use N_5 context as input. At the beginning of the algorithm, the context C_d has $|J|$ rows but zero columns. Each step of the external loop computes m_j, a new $\wedge$-irreducible element of C_d such that $j \swarrow\!\!\!\!\nearrow m_j$.

Step 1. Computation of m_1 using $J \backslash \uparrow 1$. The algorithm computes the $\vee$-representation of m_1, the $\wedge$-irreducible element such that $1 \swarrow\!\!\!\!\nearrow m_1$. At the end of this step of the loop, C_d has a unique column which correspond to m_1.



	m_1
1	
2	×
3	×

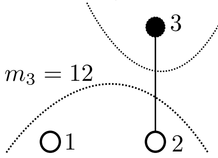
Step 2. Computation of m_2 using $J \backslash \uparrow 2$. The algorithm computes the $\vee$-representation of m_2, the $\wedge$-irreducible element such that $2 \swarrow\!\!\!\!\nearrow m_2$. At the end of this step of the loop, C_d has two columns which correspond to m_1 and to a newly computed element m_2.



	m_1	m_2
1		×
2	×	
3	×	

Step 3. Computation of m_3 using $J \backslash \uparrow 3$. The algorithm computes the $\vee$-representation of m_3, the $\wedge$-irreducible element such that $3 \swarrow\!\!\!\!\nearrow m_3$. At the end of this step of the loop, C_d has three columns which correspond to m_1, m_2 and

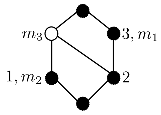
to a newly computed element m_3.



	m_1	m_2	m_3
1		×	×
2	×		×
3	×		

The whole context C_d for L_d is now computed. By construction, the only arrow relations are double arrows between j and m_j Below, L_d is drawn with black circles for concepts which were present in L and white circles for new concepts.

	m_1	m_2	m_3
1	⤢	×	×
2	×	⤢	×
3	×		⤢



5 Discussion and Conclusion

Motivated by the work of Priss [20] on the use of FCA on phylogenetic problems, we have proposed an algorithmic approach to compute the reduced context of a distributive lattice L_d from the reduced context of any lattice L, that ensures an order embedding from L into L_d that preserves $\wedge$. So, L_d can be considerated "not too far" from L and thus suitable for applications in phylogeny. In the remainder of this final section, we discuss some features of this algorithm.

First, we discuss an interpretation of the behavior of the algorithm for phylogenetic data. The algorithm computes $\wedge$-irreducible elements of L_d without any consideration for $\wedge$-irreducible elements of L but, as discussed in Subsection 3.2, this is not a problem. Now, for real data, two cases may appear:

1. $\mu m_j \in L$: in this case, we can use the initial label m of the object (this label may represent a particular gene mutation);
2. $\mu m_j \notin L$: this case suggests a gene mutation that is not spotted in the data, but that is necessary to provide a parsimonious tree.

Similarly, it is possible that $m \in M(L)$ but $m \notin M(L_d)$ but in any case, μm exists in L and L_d. This is the case when a mutation m is regarded as the infimum of other mutations.

Second, we propose an algorithm to build the context of a distributive lattice from any context. However, it is only a partial solution to the problem considered in [20]:

> "an algorithm for converting a concept lattice [into a median graph] consists of omitting the bottom node and then checking every principal filter for distributivity and turning it into a distributive lattice if it is not already one."

In the following, we discuss the whole process presented in [20]. We have proposed an algorithmic approach to "turning it into a distributive lattice if it is not already one". However, there is still some work to do as the suggestion in [20] does provide suitable solutions. This is illustrated by the example given in Figure 3.

	a	b	c	d	e
1	×	×		×	
2		×	×		×
3	×			×	
4			×		×
5				×	
6					×

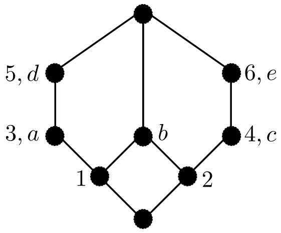


Fig. 3. Problematic context and associated concept lattice

Indeed, if we were to follow the steps suggested by Priss [20] on this example, the procedure would not provide a correct solution (i.e., a distributive lattice for principal filters). Consider a local approach on ↑ 1 and ↑ 2. The first step is to compute the reduced context of ↑ 1 (since the example is symmetric for 1 and 2, we only give details for 1). The reduced context C^1 of $L^1 =\uparrow 1$ can be built from $C(J, M, I)$ by first observing that $1' = \{a, b, d\}$, which entails the following context:

	a	b	d
1	×	×	×
2		×	
3	×		×
4			
5			×
6			

and that reduces to:

	a	b	d
2		×	
3	×		×
5			×

Algorithm 1 then returns the context C_d^1 of a distributive lattice; similarly, Algorithm 1 returns context C_d^2 of $L^2 =\uparrow 2$.

C_d^1	m_2	m_3	m_5
2		×	×
3	×		×
5	×		

C_d^2	m_1	m_4	m_6
1		×	×
4	×		×
6	×		

Moreover, in the whole lattice, every $m \in M^1$ is greater than 1 and every $m \in M^2$ is greater than 2. Hence we obtain the left context for the whole lattice and the reduced context on the right:

	m_2	m_3	m_5	m_1	m_4	m_6
2		×	×	×	×	×
3	×		×			
5	×					
1	×	×	×		×	×
4				×		×
6				×		

	m_2	m_5	m_1	m_6
2		×	×	×
3	×	×		
5	×			
1	×	×		×
4			×	×
6			×	

The resulting lattice is presented in Figure 4.a; not every principal filter is distributive. The problem comes from the fact that the modified parts of the lattice belong to intersection of $\uparrow 1$ and $\uparrow 2$. The new added elements in a filter may belong to other filters, and this may "break" the consistency achieved in the other filters.

Now we applied this procedure in parallel for $\uparrow 1$ and $\uparrow 2$, and someone could argue that it should be iterated filter by filter until a fixed point is reached. Nevertheless, an optimal solution cannot be found through the general procedure suggested by Priss [20], since all filters must be considered simultaneously. In the present case, there exists an optimal solution with only one new concept. This solution is given in Figure 4.b

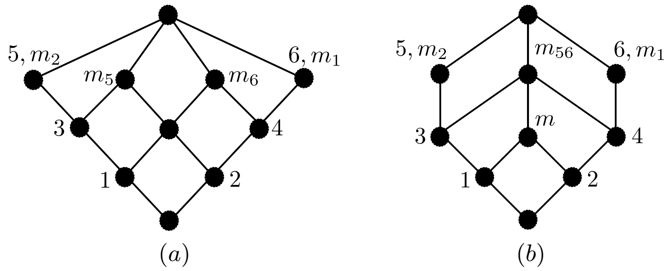


Fig. 4. (a) Solution obtained after a local approach and (b) optimal solution.

The difficulty of simultaneously considering all the filters should be studied and solved to deal with phylogenetic data. This entails to the two following open problems.

Problem 1 (Lattice version). Given a lattice L, propose an efficient algorithm to output a lattice L_d such that:

- for each atom x of L_d, $\uparrow x$ is a distributive lattice,
- there is an order embedding from L to L_d, and
- $|L_d| - |L|$ is minimal.

Problem 2 (Context version). Given the reduced context of a lattice L, propose an efficient algorithm to output the reduced context of a lattice L_d such that:

- for each atom x of L_d, $\uparrow x$ is a distributive lattice,

- there is an order embedding from L to L_d, and
- $|L_d| - |L|$ is minimal.

We are currently working on these two variations of the problem. The objective is to establish an operational bridge between FCA (concept lattices) and distributive lattices to allow the use of FCA algorithms in phylogeny.

References

1. Avann, S.P.: Median algebras. Proceedings of the American Mathematical Society **12**, 407–414 (1961)
2. Bandelt, H.J., Hedlíková, J.: Median algebras. Discrete mathematics **45**(1), 1–30 (1983)
3. Bandelt, H.J., Forster, P., Röhl, A.: Median-joining networks for inferring intraspecific phylogenies. Molecular biology and evolution **16**(1), 37–48 (1999)
4. Bertrand, P., Diatta, J.: Multilevel clustering models and interval convexities. Discrete Applied Mathematics **222**, 54–66 (2017)
5. Bertrand, P., Janowitz, M.F.: Pyramids and weak hierarchies in the ordinal model for clustering. Discrete Applied Mathematics **122**(1), 55–81 (2002)
6. Birkhoff, G.: On the combination of subalgebras. In: Mathematical Proceedings of the Cambridge Philosophical Society. vol. 29, pp. 441–464. Cambridge University Press (1933)
7. Birkhoff, G.: Lattice theory. In: Colloquium Publications, vol. 25. Amer. Math. Soc., 3. edn. (1967)
8. Birkhoff, G., Kiss, S.A.: A ternary operation in distributive lattices. Journal of Symbolic Logic **13**(1), 50–51 (1948)
9. Bordalo, G.H., Monjardet, B.: The lattice of strict completions of a poset. Electronic Notes in Discrete Mathematics **5**, 38–41 (2000)
10. Carpineto, C., Romano, G.: Concept Data Analysis: Theory and Applications. John Wiley & Sons (2004)
11. Caspard, N., Leclerc, B., Monjardet, B.: Ensembles Ordonnés Finis: Concepts, Résultats et Usages, vol. 60. Springer (2007)
12. Davey, B.A., Priestley, H.A.: Introduction to Lattices and Order. Cambridge university press (2002)
13. Diatta, J.: Galois weak hierarchies: Theoretical and computational issues. In: UK Workshop on Computational Intelligence. p. 3 (2005)
14. Ganter, B., Wille, R.: Formal Concept Analysis: Mathematical Foundations. Springer (1999)
15. Grätzer, G.: General Lattice Theory (Second Edition). Birkhäuser (1996)
16. Hopcroft, J.E., Motwani, R., Rotwani, Ullman, J.D.: Introduction to Automata Theory, Languages and Computability. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 2nd edn. (2000)
17. Mattern, F.: Virtual time and global states of distributed systems. Parallel and Distributed Algorithms **1**(23), 215–226 (1989)
18. Nation, J., Pogel, A.: The lattice of completions of an ordered set. Order **14**(1), 1–7 (1997)
19. Priss, U.: Concept lattices and median networks. In: CLA. pp. 351–354 (2012)
20. Priss, U.: Representing median networks with concept lattices. In: ICCS. pp. 311–321. Springer (2013)

A Data Analysis Application of Formal Independence Analysis[⋆]

Francisco J. Valverde-Albacete[1] [⋆⋆], Carmen Peláez-Moreno[1], Inma P. Cabrera[2], Pablo Cordero[2], and Manuel Ojeda-Aciego[2]

[1] Depto. Teoría de Señal y Comunicaciones, Univ. Carlos III de Madrid, Madrid, Spain, fva@tsc.uc3m.es carmen@tsc.uc3m.es

[2] Dpt. Matemática Aplicada, Univ. de Málaga, Málaga, Spain
ipcabrera@uma.es pcordero@uma.es aciego@uma.es

Abstract. In this paper we present a new technique for the analysis of data tables by means of Formal Independence Analysis (FIA). This is an analogue of Formal Concept Analysis for the study of independence relations in data, instead of hierarchical relations. A FIA of a context produces, when possible, its block diagonalization by detecting pairs of sets of objects and attributes that are not mutually incident, or *tomoi*, that partition the context. In this paper we combine this technique with the exploration of contexts with entries in a semifield to find independent sets in contingency matrices. Specifically, we apply it to a number of confusion matrices issued from cognitive experiments to find evidences for the hypothesis of perceptual channels.

1 Introduction and Motivation

In this paper we derive a technique for data analysis from the recently introduced Formal Independence Analysis, (FIA) [11]. This is an analysis technique for formal contexts based on the description of certain pairs of subsets of objects and attributes called *tomoi*, e.g. *divisions*, which are unrelated through the incidence. We set out to demonstrate how these *tomoi* allow us to dissect the structure and information of certain matrices.

Independent Perceptual Channels. Miller and Nicely [4] posited that for certain human perceptual tasks—e.g. consonant perception—the underlying structure of confusion matrices provide evidence of the existence of *perceptual channels* associated with specific *perceptual features*. This work is aimed at providing a technique to make such channels evident with the goals and techniques of Lattice Theory.

[⋆] FJVA and CPM have been partially supported by the Spanish Government-MinECo projects TEC2014-53390-P and TEC2017-84395-P. IPC and PC have been partially supported by the Spanish Government-MinECo project TIN2017-89023-P; MOA has been partially supported by the Spanish Government-MinECo project TIN2015-70266-C2-P-1.

[⋆⋆] Corresponding author.

© Dmitry I. Ignatov, Lhouari Nourine (Eds.): CLA 2018, pp. 117–128,
ISBN 978–80–244–5328–6, Department of Computer Science, Palacký University
Olomouc, 2018.

Specifically, consider the confusion matrix C_{ij} describing the results of an iterated classification experiment "when presented with stimulus i, and the (human) classifier answered response j." If the hypothesis of independent channels were true, we would expect this confusion matrix to be reordered by specific permutations of its rows and columns into a block diagonal form, more specifically, a *squared* block diagonal form. In this block-diagonal form, each block would describe the confusions *within* a perceptual channel, while confusions *outside* the channel would not be observed.

Reading Guide. In this paper we will use the recently developed FIA (Section 2.1) to obtain a block-diagonal form for confusion matrices, that leads to the independent virtual channel hypothesis of Miller and Nicely. This result actually stems from the consideration of a disjoint union of subcontexts decomposition technique already available from [2] that we relate to the notion of tomos and boolean tomoi lattice (Section 2.2). Our main results are the theoretical technique (Section 3.1) and the actual analyses carried out in the Miller and Nicely data (Section 3.2). We also provide a Discussion, a look into Further Work and some Conclusions.

2 Methods

2.1 Formal Independence Analysis

FIA was defined to complement the analysis of the information in formal contexts carried out by FCA, originally in terms of the hierarchical relation of formal concepts in terms of the inclusion between extents and intents. Instead, FIA targets the *relation of independence* between sets of objects and attributes [11], therefore called *tomoi*[3] The objects in the "extent" of a formal tomoi have no relation with the attributes of the "intent" of the tomoi.

Theorem 1 (Basic theorem of formal independence analysis).

1. *The context analysis phase: Given a formal context* (G, M, I)*,*
   (a) *The operators* $\cdot^{\sim} : 2^G \to 2^M$ *and* $\cdot_{\sim} : 2^M \to 2^G$

$$\alpha^{\sim} = M \smallsetminus \bigcup_{g \in \alpha} I(g, \cdot) = \{m \in M \mid g \not I m \text{ for all } g \in \alpha\} \tag{1}$$

$$\beta_{\sim} = G \smallsetminus \bigcup_{m \in \beta} I(\cdot, m) = \{g \in G \mid g \not I m \text{ for all } m \in \beta\} \tag{2}$$

form a right-Galois connection $(\cdot^{\sim}, \cdot_{\sim}) : (2^G, \subseteq) \leftharpoonup\!\!\rightharpoonup (2^M, \subseteq)$ *whose* formal *tomoi are the pairs* (α, β) *such that* $\alpha^{\sim} = \beta$ *and* $\alpha = \beta_{\sim}$.

[3] From the Greek "tomos-tomoi", *division.*

(b) *The set of formal tomoi* $\mathfrak{A}(G,M,I)$ *with the relation*

$$(\alpha_1,\beta_1) \leq (\alpha_2,\beta_2) \;\; \textit{iff} \;\; \alpha_1 \supseteq \alpha_2 \;\; \textit{iff} \;\; \beta_1 \subseteq \beta_2$$

is a complete lattice, which is called the tomoi lattice of (G,M,I) *and denoted* $\underline{\mathfrak{A}}(G,M,I)$*, where infima and suprema are given by:*

$$\bigwedge_{t\in T}(\alpha_t,\beta_t) = \left(\bigcup_{t\in T}\alpha_t, \Big(\bigcap_{t\in T}\beta_t\Big)_{\sim}^{\;\sim}\right) \quad \bigvee_{t\in T}(\alpha_t,\beta_t) = \left(\Big(\bigcap_{t\in T}\alpha_t\Big)^{\sim}_{\;\sim}, \bigcup_{t\in T}\beta_t\right)$$

(c) *The mappings* $\overline{\gamma} : G \to \underline{\mathfrak{A}}(G,M,I)$ *and* $\overline{\mu} : M \to \underline{\mathfrak{A}}(G,M,I)$

$$g \mapsto \overline{\gamma}(g) = (\{g\}^{\sim}{}_{\sim}, \{g\}^{\sim}) \qquad m \mapsto \overline{\mu}(m) = (\{m\}_{\sim}, \{m\}_{\sim}{}^{\sim})$$

are such that $\overline{\gamma}(G)$ *is infimum-dense in* $\underline{\mathfrak{A}}(G,M,I)$ *,* $\overline{\mu}(M)$ *is supremum-dense in* $\underline{\mathfrak{A}}(G,M,I)$*.*

2. *The context synthesis phase: Given a complete lattice* $\mathbb{L} = \langle L, \leq\rangle$
   (a) $\mathbb{L}$ *is isomorphic to*[4] $\underline{\mathfrak{A}}(G,M,I)$ *if and only if there are mappings* $\overline{\gamma} : G \to L$ *and* $\overline{\mu} : M \to L$ *such that*
   - $\overline{\gamma}(G)$ *is infimum-dense in* $\mathbb{L}$ *,* $\overline{\mu}(M)$ *is supremum-dense in* $\mathbb{L}$*, and*
   - $g\ I\ m$ *is equivalent to* $\overline{\gamma}(g) \not\geq \overline{\mu}(m)$ *for all* $g \in G$ *and all* $m \in M$*.*

   (b) *In particular,* $\mathbb{L} \cong \underline{\mathfrak{A}}(L,L,\not\geq)$ *and, if* L *is finite,* $\mathbb{L} \cong \underline{\mathfrak{A}}(M(\mathbb{L}),J(\mathbb{L}),\not\geq)$ *where* $M(\mathbb{L})$ *and* $J(\mathbb{L})$ *are the sets of meet- and join-irreducibles, respectively, of* $\mathbb{L}$*.*

It is already known that the lattices of formal tomoi and concepts are deeply related [13,7]. Recall that the contrary context to any (G,M,I) is the context (M,G,I^{cd}), where the incidence has been transposed and inverted.

Proposition 1. *The formal lattice of the contrary formal context is isomorphic to the tomoi lattice:*

$$\underline{\mathfrak{A}}(G,M,I) \cong \underline{\mathfrak{B}}(M,G,I^{\mathrm{cd}})$$

2.2 Disjoint Context Sum and Adjoined Lattices

To set this scenario in a Formal Concept Analysis setting, recall from [2, Definition 30] that the *disjoint sum of two contexts* $\mathbb{K}_1 = (G_1,M_1,I_1)$ and $\mathbb{K}_2 = (G_2,M_2,I_2)$, with disjoint object and attribute sets is the context $\mathbb{K}_1 \dot{\cup} \mathbb{K}_2 = (G_1 \cup G_2, M_1 \cup M_2, I_1 \cup I_2)$, and that the concept lattice of the total context is the *horizontal sum of the two concept lattices*, that is, a union of the two lattices which only overlap in the top and bottom elements

$$\mathbb{K} = \mathbb{K}_1 \dot{\cup} \mathbb{K}_2 \iff \underline{\mathfrak{B}}(\mathbb{K}_1 \dot{\cup} \mathbb{K}_2) = \underline{\mathfrak{B}}(\mathbb{K}_1) \dot{\cup} \underline{\mathfrak{B}}(\mathbb{K}_2).$$

[4] Read *can be built as.*

This can be straightforwardly generalized to a finite number n of lattices,

$$\mathbb{K} = \dot{\bigcup}_{i=1}^{n} \mathbb{K}_i \iff \underline{\mathfrak{B}}\left(\dot{\bigcup}_{i=1}^{n} \mathbb{K}_i\right) = \dot{\bigcup}_{i=1}^{n} \underline{\mathfrak{B}}(\mathbb{K}_i). \tag{3}$$

This is what we call in this paper an *(explicit) block diagonal form* for the context, which results in a concept lattice of *adjoined sublattices*.

For this latter generalization, notice that each extent of $\mathbb{K}$, except for the extent $G = \cup_i G_i$, is entirely contained in one of the sets G_i, and concept-lattice dually for intents. So it makes sense to say that *two non-extreme concepts are orthogonal is they belong to different adjoined sublattices*[5].

The relationship between tomoi lattices and block decompositions is provided by the following proposition.

Proposition 2. *If* $\underline{\mathfrak{A}}(G, M, I) \cong 2^n$ *then the context* (G, M, I) *has an explicit block diagonal form.*

Proof (Sketch). By Theorem 1 (item 2.b) the context (G, M, I) can be transformed into another one whose object-concepts are the meet-irreducible elements and whose attribute-concepts are the join-irreducible elements and, hence, because of the isomorphism with 2^n, they are the co-atoms and the atoms, respectively. Moreover, they are complementary pairs of one object-tomos and one attribute-tomos.

As consequence, it is possible to reorganize the tabular expression of (G, M, I) in such a way that we obtain a block diagonal form. □

3 Results

3.1 Theoretical Analysis.

The purpose of proving the existence of independent channels for different percepts can be achieved by reducing a confusion matrix to a block diagonal form. But, confusion matrices are *not* binary incidences and may not be subject to a simple process of block diagonalization. Instead, we may look for an *approximate block-diagonal block*, that retains the main structure of the confusions.

We can motivate this approximation in the following way:

- A perfect classifier would obtain a diagonal matrix of counts. This has been proven in terms of information-theoretic arguments in [9], for instance.
- But in most cases what we can hope for is a diagonally dominant matrix, that is not even symmetrical. For instance, the heatmap of the symmetrized confusion matrix for the M&N data for −6dB, to the left of Fig. 1, shows such a shape. Even its symmetrical part of C_S has a corresponding structure that is far from being block-diagonal, e.g. center of Fig. 1.

[5] The basis for this definition is, of course, the embedding of extents and intents as vectors in semimodules over an idempotent semifield which allows us to define a dot product between extents, resp. intents. [10]. Note that in idempotent semimodules, which are *zero-sum free*, null dot-products can only occur for vectors of disjoint support, and this is precisely the case at hand.

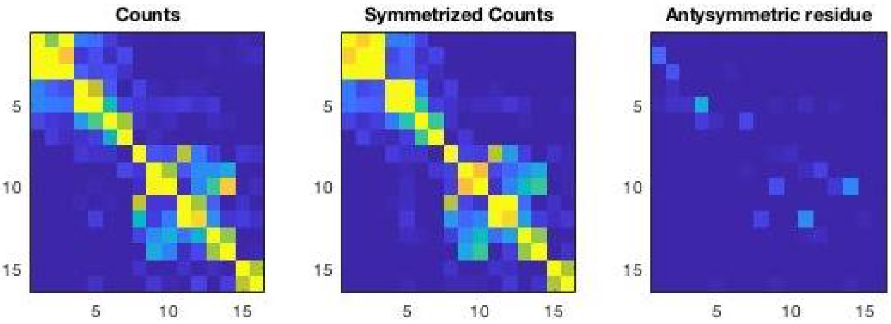


Fig. 1: (Color online) Heatmaps of the count confusion matrix in M&N for a SNR of -6dB. Left: count matrix; C center: symmetrized count matrix C_S; right: antisymmetrical residue C_A.

– Using *structural analysis* from an adequately transformed matrix $M = f(C_A)$ we could use the paradigm of *Landscapes-of-Knowledge (LoK)* [14] extended to multi-valued contexts [8] to explore the sequence of boolean incidences $I(\varphi)_{ij} = M_{ij} \lesseqgtr \varphi$ where φ ranges in the values of the original matrix:
  - Choosing $I(\varphi)_{ij} = M_{ij} \geq \varphi$ uses the *min-plus structural analysis*, while
  - Choosing $I(\varphi)_{ij} = M_{ij} \leq \varphi$ uses the *max-plus structural analysis.*

– The criterion for finding a “correct” value for φ is to ensure that the $I(\varphi)$ has a tomoi lattice that is boolean. A proxy criterion for this is to select and inspect only those φ whose number of formal tomoi is a power of 2. Note that after obtaining the appropriate φ by Proposition 2 we would have the block-decomposition.

In the following section, we check the feasibility of this scheme on the Miller and Nicely data.

3.2 FIA Exploration of Confusion Matrices

Data Description. In this paper we will use the data from the Miller and Nicely study to show examples of phenomena and test the proposed data analysis procedures. These are the confusion data of a consonant perception task, and we will refer to it as the *M&N data*. Specifically they are six different confusion matrices of 16 entries for different Signal-to-Noise Ratios (SNR) in dB of $\{-18, -12, -6, 0, 6, 12\}$ obtained in a (human) speech recognition task for the consonants listed in Table 1. The stimuli where balanced, but the responses may be unbalanced due to non-symmetrical confusion effects.

Data Preprocessing. Due to the symmetry inherent in the confusion task, since the category of the responses was the same as that of the stimuli, we extracted the symmetric component of each confusion matrix. This was done by obtaining from each matrix C its symmetric component $C_S = (C + C^{\mathrm{t}})/2$.

Table 1: Ordering of the consonants used in the confusion matrices analyzed (from [4]).

symbol	p , t , k , f , th , s , sh , b , d , g , v , dh , z , zh , m , n
phone	/p/, /t/, /k/, /f/, /θ /, /s/, /ʃ/, /b/, /d/, /g/, /v/, /ð/, /z/, /zh/, /m/, /n/

The antisymmetric component $C_A = (C - C^{\mathrm{t}})/2$ can then be interpreted as a residue. For the M&N confusion matrix at -6dB these two components can be seen in Fig. 1.

The data were preprocessed to obtain both the Pointwise Mutual Information (MI) and the Weighted Pointwise Mutual information (WPMI) as shown in Fig. 2. Although prior work suggested that WPMI lends itself to more clear analyses, for the purpose of finding independent blocks in the matrix, we found—on using both types of data preprocessing—MI to retain more details about confusions that define the blocks, e.g. between elements that share (unknown) features motivating the confusion, for instance the voiceless fricatives /s/ vs. /ʃ/.

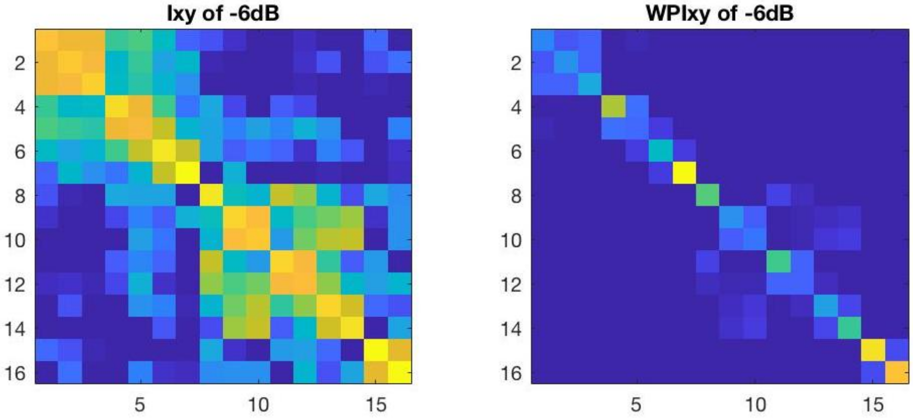
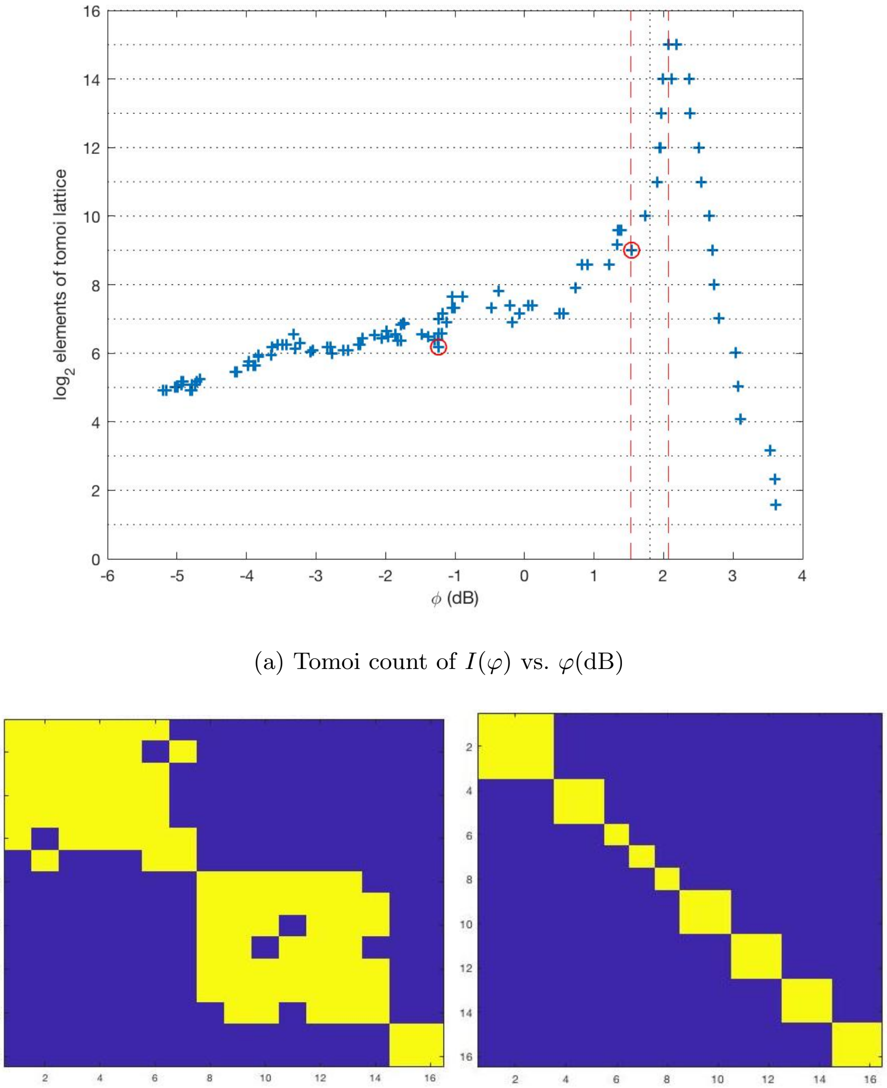


Fig. 2: (Color online) Heatmaps of the confusion matrix in M&N for a SNR of -6dB for different preprocessing. Left: pointwise mutual information. Right: weighted pointwise mutual information.

Data Analysis. We carried out min-plus exploratory analysis in the MI-transformed confusion matrix above by thresholding for each φ in increasing order and generated a sequence of $K = 105$ (binary) formal contexts $\mathbb{K}(\varphi_k) = (G, M, I(\varphi_k)), k \in [1, \ldots K]$.

For each of these contexts, we calculated the number of formal tomoi for each thresholded $I(\varphi)$ by actually working out the formal concepts of the contrary context $\mathbb{K}(\varphi)^{\mathrm{cd}} = (M, G, \not{I}^{\mathrm{t}})$. We do not explore at $\varphi = -\infty$ which entails a trivial full-incidence and a count of one tomoi.

The graph of these counts in base-2 logarithm, shown in Fig. 3.a, allows us to define three regions:



(a) Tomoi count of $I(\varphi)$ vs. φ(dB)

(b) $I(\varphi)$ at $\varphi = -1.299010$

(c) $I(\varphi)$ at $\varphi = 1.539160$

Fig. 3: (Color online) Number of formal tomoi vs φ for $I(\varphi)$ and heatmaps for two highlighted φ.

- *An **initial segment** where the threshold is too lax and we see essentially few blocks and a number of "noise" tomoi, where our assumption, viz. that there are virtual channels, does not hold.*
  In the example being analyzed, this is the range $(-5.2, 1.53)$, to the left of the leftmost vertical line in Fig. 3.a . To ascertain the shapes of the thresholded we present an instance for where $\varphi \approx -1.23$ and $|\mathbb{K}(\varphi)| = 30$ focused on by the leftmost circle. In the heatmap of Fig. 3.b we can see and inkling of three different blocks, but since they are not complete, a number of "noisy" tomoi appear, making the tomoi lattice drift away from $\mathbb{2}^3$. Figure 4.a shows this non-boolean tomoi lattice whose incidence is that of Fig. 3.b
- *A **middle segment** where we start seeing many blocks, and consequently the number of tomoi $|\underline{\mathfrak{A}}(G, M, I(\varphi))|$ falls exactly into one of the powers of 2, where our assumption holds.*
  In the example, this is $\varphi \in [1.53, 2.07]$ between the vertical lines in Fig. 3.a comprising the ramp where the cardinalities range from 2^9 to 2^{14} tomoi. This is the case, for instance, of $\varphi \approx -1.53, |\mathbb{K}(\varphi)| = 2^9$, signaled as the rightmost red circle. We can see the 9-block incidence in Fig. 3.c, while Fig. 4.b shows the boolean lattice $\mathbb{K}(\varphi) \cong \mathbb{2}^9$. For reference, the (average) mutual information for this matrix, $MI_{-6dB} = 1.80$ falls within this range, and would generate the tomoi lattice isomorphic to $\mathbb{2}^{10}$.
- *A **final segment** where the threshold is too stringent and we no longer see a block diagonal form.*
  This is the least interesting zone for us. In the example it appears as a descending slope in the range $\varphi \in (2.07, 3.61)$ of Fig. 3.a.

We checked whether this behavior was analogous for all confusion matrices by analyzing the rest of the matrices at different SNR. The following are the main trends of analysis:

- We could only obtain boolean tomoi lattices considering all stimuli for those confusion matrices with SNR of $\{18, 12, 6, 0, -6\}$. The matrices at SNR $\in \{-12, -18\}$ were too noisy and some elements in the diagonal were less stable than elements off the diagonal, hence they disappeared on early exploration.
- In all of the instances where in some range of MI values the exploration procedure obtained boolean tomoi lattices, the average MI for the whole matrix, that is in the standard definition of mutual information, actually belonged in the range where the hypothesis held. Most of the times, this MI was close to the value for values of φ that obtained the boolean tomoi lattice of highest cardinality.
- The highest SNR in the confusion matrix being analyzed, the higher number of blocks in $I(\varphi)$. This is congruent with the supposition that high SNR situations allow us to distinguish individual phones better and it is therefore more difficult to obtain evidence of the perceptual channels through confusions.

Extracting Perceptual Channels. The tomoi provide the basis for obtaining the perceptual channels on top of boolean tomoi lattices, since for every object-tomoi, a *meet-irreducible*, its complement is an attribute tomoi, hence a *join-irreducible*. By the properties of complementary tomoi, the crossed extents and intents, define the blocks in the block diagonalization.

To see this, consider Table 2 of object-tomoi *extents* and their complementary tomoi intents—the attribute tomoi—to be used to build the block-diagonal form of (3). We see how, modulo a permutation, they constitute a refinement of the perceptual channels that Miller and Nicely proposed [4].

Table 2: Paired table of meet- and join-irreducibles of $\mathbb{K}(\varphi)$ in Fig. 4.b .

Object and attribute subsets	object-tomoi extent	complementary attribute-tomoi intent
(G_1, M_1)	$\{p, t, k\}$	$\{p, t, k\}$
(G_2, M_2)	$\{f, \theta\}$	$\{f, \theta\}$
(G_3, M_3)	$\{s\}$	$\{s\}$
(G_4, M_4)	$\{\int\}$	$\{\int\}$
(G_5, M_5)	$\{b\}$	$\{b\}$
(G_6, M_6)	$\{d, g\}$	$\{d, g\}$
(G_7, M_7)	$\{v, ð\}$	$\{v, ð\}$
(G_8, M_8)	$\{z, zh\}$	$\{z, zh\}$
(G_9, M_9)	$\{m, n\}$	$\{m, n\}$

Discussion and Further Work. Note that the problem we address in this paper was already approached in [12], but not solved satisfactorily, and we believe FIA provides a principled approach to the study of independent blocks within matrices.

In fact, FIA seems to detect much finer perceptual channels than the original paper suggests, perhaps because of the granularity of the perceptual features used there (see below). In order to obtain a rougher partition of phones to support Miller and Nicely's hypothesis, we have tried to analyze a balanced mixture of all the confusion matrices. But FIA has proven too strong for this unrealistic type of noise: the absence of confusions at 12dB dominates the behavior of the mixture, and the confusions from those behaviors at -18dB and -12dB are lost. Recall that it is precisely from the confusions where we obtain the evidence for the perceptual channels, so clearly a more nuanced approach to such mixture would be needed.

Although we have provided a data-induced procedure to obtain perceptual channels from confusion matrices, this is only a first step in actually obtaining the *experimental channels*. In particular, we have not investigated justifying those channels in terms of perceptual characteristics. While Miller and Nicely proposed an encoding of phones based on traditional *categorical* phonetic features, modern studies favor the consideration of *numeric* features. In our opinion this necessarily entails considering idempotent semimodule models of such spaces [6] and would lead to higher-value hypotheses. This is left for future work.

By no means is ours the only attempt at block-diagonalizing matrices over idempotent semifields. In fact, such process is important for the calculation of the Moore-Penrose inverse of a matrix over an idempotent semifield [5]. The main difference with out work is that we are trying to *approximate* the block-diagonal form in the presence of an *implicit noise*.

Yet a more general version of the problem is that of Cell Formation (CF) in Group Technology, in the field of Manufacturing [1], because it involves the block diagonalization of rectangular matrices. FIA is not restricted to squared matrices, but our application and interpretation indeed are because of the fact that confusion matrices are usually square. CF therefore opens up as an open research and application avenue for FIA.

Finally, further work is necessary to ascertain the relationship of lattices of formal tomoi to lattices of formal concepts, as well as to find out whether these are the only information lenses available for formal contexts, or how to measure the "quality" of the tomoi, in an effort similar to that shown for triadic analysis and triclustering in [3]. Our next main aim, though, is to incorporate these techniques in the over-arching exploratory data analysis framework first laid out in full in [8].

4 Conclusions

We have introduced a new technique to analyze data tables based on the newly proposed Formal Independence Analysis. The purpose of the technique is to obtain *tomoi*,—pairs of sets of objects and attributes unrelated through an incident relation— and their complements in the lattice of tomoi, which define as many partitions of the sets of objects and attributes. These tomoi will then be used to define a block-diagonal form for the incidence.

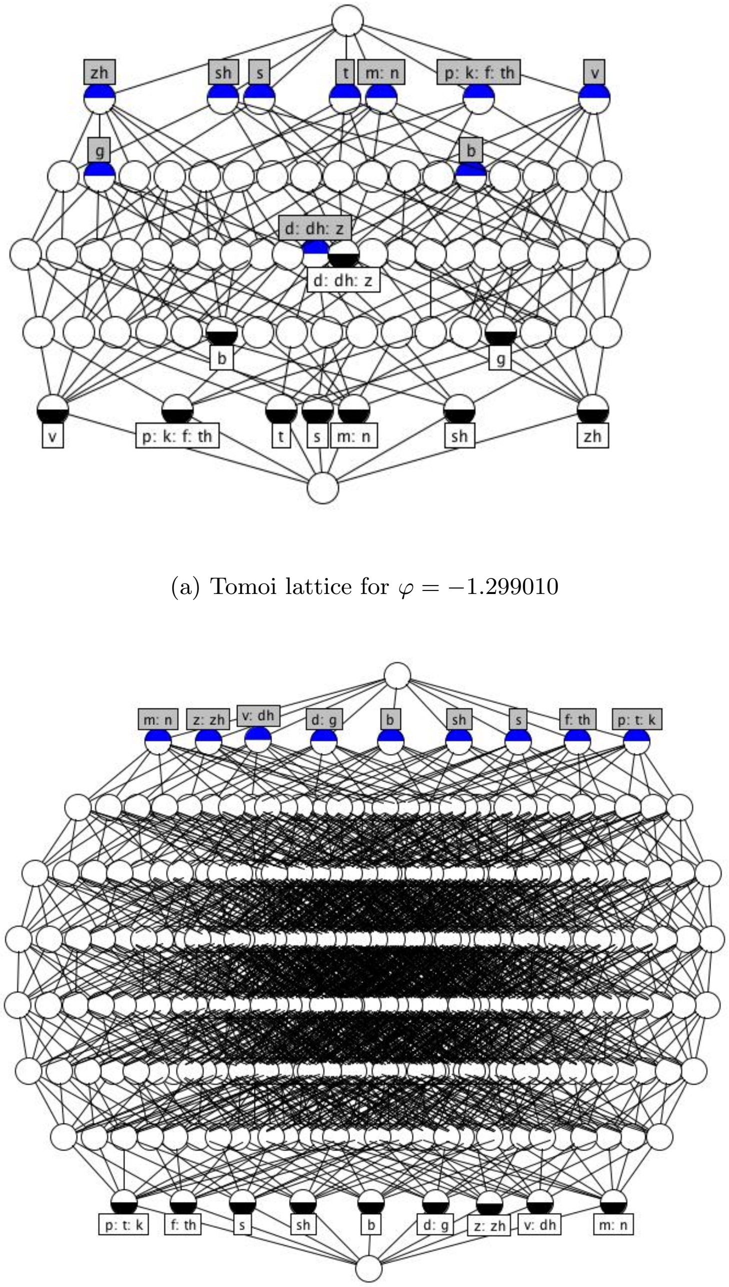
We apply the technique to the diagonally-dominant incidences of confusion matrices. By a process of exploration we select special thresholds that obtain boolean tomoi lattices. In these lattices we obtain the meet-irreducible object-tomoi and their complements, the join-irreducible attribute-tomoi, that define diagonal blocks on the looked-for incidence.

These diagonal blocks can be interpreted as virtual channels that transmit different types of information in the spirit of some classical perceptual experiments, e.g. Miller and Nicely's.

References

1. Belohlavek, R., Kulkarni, N., Vychodil, V.: A Novel Approach to Cell Formation. In: Formal Concept Analysis, ICFCA, pp. 210–223. Springer-Verlag (2009)
2. Ganter, B., Wille, R.: Formal Concept Analysis: Mathematical Foundations. Springer, Berlin, Heidelberg (1999)
3. 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), 271–302 (Apr 2015)

4. Miller, G.A., Nicely, P.E.: An analysis of perceptual confusions among some English consonants. The Journal of the Acoustic Society of America **27**(2), 338–352 (Oct 1955)
5. Pati, S.: Moore–Penrose Inverse of Matrices on Idempotent Semirings. SIAM J. Matrix Anal. Appl **22**(2), 617–626 (electronic) (May 2000)
6. Peláez-Moreno, C., García-Moral, A.I., Valverde-Albacete, F.J.: Analyzing phonetic confusions using Formal Concept Analysis. Journal of the Acoustical Society of America **128**(3), 1377–1390 (Sep 2010)
7. Reuter, K.: The jump number and the lattice of maximal antichains. Discrete Mathematics **88**(2-3), 289–307 (1991)
8. Valverde-Albacete, F.J., González-Calabozo, J.M., Peñas, A., Peláez-Moreno, C.: Supporting scientific knowledge discovery with extended, generalized formal concept analysis. Expert Systems with Applications **44**, 198 – 216 (2016). https://doi.org/http://dx.doi.org/10.1016/j.eswa.2015.09.022, `http://www.sciencedirect.com/science/article/pii/S0957417415006442`
9. Valverde-Albacete, F.J., Peláez-Moreno, C.: 100% classification accuracy considered harmful: the normalized information transfer factor explains the accuracy paradox. PLOS ONE pp. 1–10 (january 2014). https://doi.org/10.1371/journal.pone.0084217
10. Valverde-Albacete, F.J., Peláez-Moreno, C.: The linear algebra in formal concept analysis over idempotent semifields. In: Formal Concept Analysis, pp. 97–113. No. 9113 in LNAI, Springer Berling Heidelberg, Heidelberg (Apr 2015)
11. Valverde-Albacete, F.J., Peláez-Moreno, C., Cabrera, I.P., Cordero, P., Ojeda-Aciego, M.: Formal independence analysis. In: Medina, J., et al. (eds.) Proceedings of IPMU 2018. vol. 853 of CCIS., pp. 1–12 (2018). https://doi.org/10.1007/978-3-319-91473-2_51
12. Valverde-Albacete, F.J., Peláez-Moreno, C., Peñas, A.: On concept lattices as information channels. In: Bertet, K., Rudolph, S. (eds.) Concept Lattices and Their Applications (CLA 2014). pp. 119–131. Institute of Computer Science, Pavol Josef Safarik, University in Kosice (October 2014), iSBN 978–80–8152–159–1
13. Wille, R.: Finite distributive lattices as concept lattices. Atti Inc. Logica Mathematica **2**, 635–648 (1985)
14. Wille, R.: Conceptual landscapes of knowledge: a pragmatic paradigm for knowledge processing. In: Mineau, G., Fall, A. (eds.) Proceedings of the Second International Symposium on Knowledge Retrieval, Use and Storage for Efficiency. pp. 2–13. Vancouver (August 11–13 1997)



(a) Tomoi lattice for $\varphi = -1.299010$

(b) Tomoi lattice for $\varphi = 1.539160$

Fig. 4: (Color online) Tomoi lattices for the two incidences of Fig. 3

Towards Simplification Logic for Graded Attribute Implications with General Semantics

Pablo Cordero*, Manuel Enciso*, Angel Mora*, and Vilem Vychodil‡

*Universidad de Málaga, Andalucía Tech, Spain
e-mail: {pcordero,enciso}@uma.es, amora@ctima.uma.es
‡Dept. Computer Science, Palacky University Olomouc, Czechia
e-mail: vilem.vychodil@upol.cz

Abstract. We present variant of simplification logic for reasoning with if-then dependencies that arise in formal concept analysis of data with graded attributes. The dependencies and the proposed logic are parameterized by systems of isotone Galois connections which allows us to handle a large family of possible interpretations of data dependencies. We describe semantics of the rules, axiomatic system of the logic, and prove its soundness and completeness.

Keywords: Closure operator, lattice theory, fuzzy logic, implication

1 Introduction and Problem Setting

In this paper, we contribute to the area of inference systems that emerge in formal concept analysis [10] of data with graded attributes. By a *graded attribute*, sometimes called a *fuzzy attribute,* we mean an attribute that may apply to an object to degrees. Needless to say, there are basically two options to treat such attributes: Either by binary scaling and exploiting the existing methods in FCA or by providing a suitable formalization of structures of degrees and developing FCA considering such structures in order to include "graded attributes" as fundamental notions. In this paper, we use the "approach by generalization" and explore general inference systems related to graded attribute implications, i.e., if-then formulas describing dependencies between graded attributes.

1.1 Early Approaches

The first approach to FCA that contained results on graded attribute implications was introduced by Silke Polandt in her somewhat unappreciated book [19]. The approach is based on residuated lattices [28] considered as basic structures of truth degrees [11,13] and introduces attribute implications as if-then formulas $A \Rightarrow B$, where A and B are graded collections of attributes (fuzzy sets of attributes), i.e., technically both A and B are maps $A : Y \to L$ and $B : Y \to L$, where Y is a set of attributes and L is a set of utilized degrees. The interpretation of $A \Rightarrow B$ in a given formal context with graded attributes is defined in terms of a *graded subsethood.*

© Dmitry I. Ignatov, Lhouari Nourine (Eds.): CLA 2018, pp. 129–140,
ISBN 978–80–244–5328–6, Department of Computer Science, Palacký University
Olomouc, 2018.

The role of the graded subsethood in [19] as well as in the later approaches is crucial, so let us clarify what we mean by that. The classic subsethood (or inclusion) can be seen as a binary relation on the set of all subsets of a given universe, e.g., the set of all attributes. When one thinks of a subsethood in presence of graded attributes, it appears almost immediately that there seem to be multiple reasonable choices for that. For instance, for maps A and B as above, we might say that "B is fully included in A" whenever $B(y) \leq A(y)$ for each attribute $y \in Y$, where $\leq$ is a partial order on the set of all degrees in L. As such, the full inclusion is a classic binary relation on the set of all graded sets in the universe Y, i.e., for each A, B, either B is fully included in A or not.

However straightforward, the full inclusion may be regarded as not natural by some because it does not reflect closeness of degrees. For instance, when $B(y) \leq A(y)$ for all y except for some $z \in Y$ for which we have $B(z) = 0.63$ and $A(y) = 0.62$ (when a real unit interval is used as the scale of truth degrees), then B is not fully included in A, however, most observers would regard B to be almost fully a subset of A. This issue can be resolved by introducing a graded subsethood. While there are many approaches to define a graded subsethood, [19] and later works use the one introduced by Goguen [11] which is based on residuated implication. Using the notation of [2], a degree $\mathrm{S}(A, M)$ to which A is a subset of M (see [2, p. 82]) is defined by

$$\mathrm{S}(A, M) = \textstyle\bigwedge_{y \in Y}\bigl(A(y) \rightarrow M(y)\bigr), \tag{1}$$

where $A\colon Y \rightarrow L$, $M\colon Y \rightarrow L$, $\rightarrow$ is residuum (a truth function of graded/fuzzy implication), i.e., $\mathrm{S}(A, M)$ is the infimum of degrees $A(y) \rightarrow M(y)$ for all $y \in Y$. Since $\mathrm{S}(A, M)$ is a general degree from L, a high degree $\mathrm{S}(A, M)$ can naturally be interpreted so that "A is almost included in M." Interestingly, the two notions of subsethood are related in the following sense: "A is fully included in M" if and only iff $\mathrm{S}(A, M) = 1$ (with 1 being the highest degree in L).

Using the graded subsethood (and the notation of [2,13]), the initial approach to attribute implications [19] defined a degree to which $A \Rightarrow B$ holds for an object $x \in X$ by

$$||A \Rightarrow B||_{I_x} = \mathrm{S}(A, I_x) \rightarrow \mathrm{S}(B, I_x), \tag{2}$$

where $I_x : Y \rightarrow L$ represents graded attributes of the object $x \in X$ (i.e., $I_x(y)$ is a "degree to which object x has the attribute y.") One can immediately see that (2) is indeed a proper generalization of the classic notion of $A \Rightarrow B$ (where $A, B \subseteq Y$) being true in $I_x \subseteq Y$. By a straightforward extension of the notion, we can introduce a degree $||A \Rightarrow B||_{\langle X,Y,I\rangle}$ to which a graded attribute implication $A \Rightarrow B$ is true in a context $\langle X, Y, I\rangle$ with graded attributes:

$$||A \Rightarrow B||_{\langle X,Y,I\rangle} = \textstyle\bigwedge_{x \in X} ||A \Rightarrow B||_{I_x}.$$

In this setting, Polandt [19] investigated several important areas including characterization of completeness in data and similarity issues.

1.2 Approaches Using Hedges

The approach by Pollandt is sound but it turned out it lacks a certain level of generality that can be found in the approach using hedges which initally started by [3], cf. also [6,7] for comprehensive description. The approach uses a linguistic hedge [29] as an additional parameter that influences the interpretation of graded attribute implications and related notions from FCA [5]. Technically, instead of considering (2), one considers $||A \Rightarrow B||^{*}_{I_x}$ defined by

$$||A \Rightarrow B||^{*}_{I_x} = \mathrm{S}(A, I_x)^{*} \rightarrow \mathrm{S}(B, I_x), \tag{3}$$

where * is an idempotent truth-stressing linguistic hedge [14]. By letting * being the identity map on L, (3) collapses into the Pollandt-style definition (2). The interesting point about this particular general approach is that for other choices of hedges, we obtain other interesting interpretations of graded dependencies. For instance, when * is the so-called globalization [22], then (3) becomes

$$||A \Rightarrow B||^{*}_{I_x} = \begin{cases} 1, & \text{if } \mathrm{S}(A, I_x) < 1, \\ \mathrm{S}(B, I_x), & \text{otherwise.} \end{cases} \tag{4}$$

In particular, $||A \Rightarrow B||^{*}_{I_x} = 1$ iff $\mathrm{S}(A, I_x) = 1$ (i.e., A is fully contained in I_x) implies $\mathrm{S}(B, I_x) = 1$ (i.e., B is fully contained in I_x). In general, (2) and (4) are different and coincide only if the scale of degrees is a two-valued Boolean algebra. Therefore, the approach by hedges can be seen as a generalization that encompasses interpretation of graded if-then rules based on both the graded and full inclusions and these two borderline cases result by different choices of hedges. This is an important aspect from users' point of view.

From the theoretical point of view, the generalization by hedges brought new insights into the properties of several important notions depending on the choice of a hedge. For instance, minimal bases and pseudo-intents in the general setting [3,7] have almost the exact same characterization as in the classic case [12] when the hedge is globalization which it is not the case for general hedges where several incomparable systems of pseudo-intents may exist for a single dataset, see [24,25] for details.

1.3 Parameterizations by Isotone Galois Connections

The present paper is closely related to general methods of parameterizing the semantics of graded attribute implications proposed in [26]. Such parameterizations subsume the paramaterizations by hedges as well as other non-trivial alternative semantics of attribute implications. It may be motivated by two fundamental observations on properties of graded attribute implications paramaterized by hedges [6,7]. First, we have [6, Theorem 3]

$$||A \Rightarrow B||^{*}_{I_x} = \bigvee\bigl\{c \in L;\, ||A \Rightarrow c{\otimes}B||^{*}_{I_x} = 1\bigr\}, \tag{5}$$

where $c{\otimes}B$ denotes a map from Y to L such that $(c{\otimes}B)(y) = c \otimes B(y)$, where $\otimes$ is the multiplication adjoint to $\rightarrow$ appearing in (1). Put in words, (5) shows that the degrees to which graded attribute implications are true can be expressed just by focusing on implications that are fully true, i.e., true to degree 1.

Second, we have $||A \Rightarrow B||^{*}_{I_x} = 1$ (i.e., $A \Rightarrow B$ is fully true in I_x) iff for any truth degree $c \in L$, it holds that [6, Lemma 2]

$$\mathrm{S}(c^{*}{\otimes}A, I_x) = 1 \text{ implies } \mathrm{S}(c^{*}{\otimes}B, I_x) = 1. \tag{6}$$

By a slight abuse of notation and denoting the full inclusion by $\subseteq$, the previous condition can be restated as follows:

$$c^{*}{\otimes}A \subseteq I_x \text{ implies } c^{*}{\otimes}B \subseteq I_x. \tag{7}$$

Therefore, one may introduce a general interpretation of graded attribute implications $A \Rightarrow B$ by defining $A \Rightarrow B$ true in I_x whenever the following condition holds: For any $\boldsymbol{f} \in S$, it holds that $\boldsymbol{f}(A) \subseteq I_x$ implies $\boldsymbol{f}(B) \subseteq I_x$. In order to obtain a formalization which is sufficiently strong, [26] shows that it is sufficient to consider S as a set of (lower) adjoints of isotone Galois connections that is closed under composition. Using this formalism, [26] shows a standard agenda of attribute implications, including a complete Armstrong-style [1] axiomatization and characterization of completeness in data.

The parameterizations by systems of isotone Galois connections can be used not only in case of graded attribute implications but for other formalisms for reasoning with if-then rules. For instance, attribute implications developed in context of linear temporal logic [23] fall in this category as well. The properties of this family of parameterizations and related closure structures are studied in [27].

1.4 Our Contribution

As an alternative to the well-known Armstrong inference system [1] which is not very suitable for automated reasoning, [9] proposed a simplification logic and novel algorithms for if-then rules based on simplification equivalence. Further results derived from this work include automated methods based directly on the simplification logic [17,18,8,16,15]. The simplification logic was later introduced for graded attribute implications parameterized by hedges in [4].

In this paper, we outline a general simplification logic for graded if-then rules whose semantics is parameterized by systems of isotone Galois connections. In Section 2, we present the underlying algebraic structures that are involved in the simplification logic as well as the parameterizations. We emphasize that, we utilize the co-residuated lattices in order to have a reasonable truth-function of logical difference upon which the simplification logic is based. The role of the classic multiplications and residua in the ordinary residuated lattices is substituted by the general parameterizations. In Section 3, we outline the logic including the semantics of its formulas, we present an inference system and show its soundness. Furthermore, in Section 4 and Section 5, we present a further properties of the inference system and outline the completeness result.

2 Preliminaries

Throughout this paper we consider, as the structure of degrees, a complete co-residuated lattice, that is, an algebra $\mathbb{L} = \langle L, \leq, \oplus, \ominus, 0, 1\rangle$ satisfying the following conditions:

- $\langle L, \leq, 0, 1\rangle$ is a complete lattice where 0 is the least element and 1 is the greatest element. As usual, we use the symbols $\vee$ and $\wedge$ to denote suprema (least upper bounds) and infima (greatest lower bounds), respectively.
- $\langle L, \oplus, 0\rangle$ is a commutative monoid.
- The pair $\langle \oplus, \ominus\rangle$ satisfies the following adjointness property:

$$\text{For all } a, b, c \in L, \quad a \leq b \oplus c \text{ if and only if } a \ominus b \leq c. \tag{8}$$

Notice that (8) is equivalent to the following condition:

$$(a \oplus b) \ominus a \leq b \leq a \oplus (b \ominus a), \text{ for all } a, b \in L. \tag{9}$$

Any complete Brouwerian algebra [21,20] (also known as complete co-Heyting algebra) is a complete co-residuated lattice. Thus, as an example of complete co-residuated lattice, one has the unit interval with the operations $\oplus$ and $\ominus$ such that $a \oplus b = \max\{a, b\}$, $a \ominus b = a$ when $b < a$, and $a \ominus b = 0$ otherwise. We also take advantage of the following properties:

$$a \leq b \text{ if and only if } a \ominus b = 0, \tag{10}$$

$$a \ominus 0 = a, \tag{11}$$

$$a \ominus b \leq a \leq a \oplus b, \tag{12}$$

$$b \leq c \text{ implies } a \oplus b \leq a \oplus c,\ b \ominus a \leq c \ominus a \text{ and } a \ominus c \leq a \ominus b, \tag{13}$$

$$a \vee b \leq a \oplus (b \ominus a) \leq a \oplus b, \tag{14}$$

$$a \oplus ((a \oplus b) \ominus c) = a \oplus (b \ominus c), \tag{15}$$

$$a \oplus (b \wedge c) = (a \oplus b) \wedge (a \oplus c). \tag{16}$$

For illustration, we use a running example based on a particular structure of degrees. The structure is shown in the next example.

Example 1. Consider $\mathbb{L} = \langle L, \leq, \oplus, \ominus, 0, 1\rangle$ where $L = \{\frac{i}{10} \mid i \in \mathbb{N}, 0 \leq i \leq 10\}$, the relation $\leq$ is the usual order, and $\oplus$ and $\ominus$ are defined as follows:

$$a \oplus b = \begin{cases} a + b, & \text{if } a + b \leq \frac{1}{2}, \\ \max\{\frac{1}{2}, a, b\}, & \text{otherwise,} \end{cases} \qquad a \ominus b \ = \begin{cases} 0, & \text{if } a \leq b, \\ 1 - b, & \text{if } 0 \leq b < a \leq \frac{1}{2}, \\ \max\{a, b\}, & \text{otherwise.} \end{cases}$$

It is easy to see that $\mathbb{L}$ is a complete co-residuated lattice.

Using $\mathbb{L}$, we use the notion of $\mathbb{L}$-fuzzy sets, i.e., maps from non-empty universe sets to L. The collection of all $\mathbb{L}$-fuzzy sets in universe Y is denoted by L^Y. Also, in the examples we use the usual notation $\{\ldots, {y/}^{A(y)} \ldots\}$ for writing $\mathbb{L}$-fuzzy sets in finite universes.

Operations in $\mathbb{L}$ can be extended pointwise to $\mathbb{L}$-fuzzy sets in the usual way: For $A, B \in L^Y$ the $\mathbb{L}$-fuzzy sets $A \oplus B$ and $A \ominus B$ are defined by $(A \oplus B)(y) = A(y) \oplus B(y)$ and $(A \ominus B)(y) = A(y) \ominus B(y)$ for all $y \in Y$.

The parameterizations [26] we use in out paper are defined in terms of isotone Galois connections in $\langle L^Y, \subseteq\rangle$. Specifically, we consider pairs of self-maps $\langle \boldsymbol{f}, \boldsymbol{g}\rangle$, i.e., $\boldsymbol{f}\colon L^Y \to L^Y$ and $\boldsymbol{g}\colon L^Y \to L^Y$, such that,

$$\text{for all } A, B \in L^Y,\ \boldsymbol{f}(A) \subseteq B \text{ iff } A \subseteq \boldsymbol{g}(B). \tag{17}$$

In this pair, each mapping is uniquely determined by the other, because $\boldsymbol{f}(A) = \bigcap\{B \in L^Y \mid A \subseteq \boldsymbol{g}(B)\}$ and $\boldsymbol{g}(B) = \bigcup\{A \in L^Y \mid \boldsymbol{f}(A) \subseteq B\}$. It is well-known that (17) is equivalent to postulating that both of the following conditions hold:

1. $\boldsymbol{f}$ and $\boldsymbol{g}$ are isotone, i.e., $A \subseteq B$ implies $\boldsymbol{f}(A) \subseteq \boldsymbol{f}(B)$ and $\boldsymbol{g}(A) \subseteq \boldsymbol{g}(B)$ for all $A, B \in L^Y$.
2. $\boldsymbol{g} \circ \boldsymbol{f}$ is inflationary (extensive) and $\boldsymbol{f} \circ \boldsymbol{g}$ is deflationary (intensive), i.e., $A \subseteq \boldsymbol{g}(\boldsymbol{f}(A))$ and $\boldsymbol{f}(\boldsymbol{g}(A)) \subseteq A$ for all $A \in L^Y$.

In fact, $\boldsymbol{g}\circ\boldsymbol{f}$ is a *closure operator* and $\boldsymbol{f}\circ\boldsymbol{g}$ is a *kernel operator* (*interior operator*).

For any isomorphism $\boldsymbol{f}$ in $\langle L^Y, \subseteq\rangle$, the pair $\langle \boldsymbol{f}, \boldsymbol{f}^{-1}\rangle$ is an isotone Galois connection. Thus, the identity mapping $\boldsymbol{I}_Y \colon L^Y \to L^Y$, with $\boldsymbol{I}_Y(A) = A$ for all $A \in L^Y$, provides an isotone Galois connection. Another important example is $\langle \mathbf{0}_Y, \mathbf{1}_Y\rangle$ where $\mathbf{0}_Y(A)(y) = 0$ and $\mathbf{1}_Y(A)(y) = 1$, for any $A \in L^Y$ and $y \in Y$.

In addition, given two isotone Galois connections $\langle \boldsymbol{f}_1, \boldsymbol{g}_1\rangle$ and $\langle \boldsymbol{f}_2, \boldsymbol{g}_2\rangle$, their *composition* $\langle \boldsymbol{f}_1 \circ \boldsymbol{f}_2, \boldsymbol{g}_2 \circ \boldsymbol{g}_1\rangle$ is also an isotone Galois connection.

Definition 1. *A family of isotone Galois connections S in $\langle L^Y, \subseteq\rangle$ is said to be an $\mathbb{L}$-parameterization [26] if it is closed for composition and contains the identity.*

In other words, S is an $\mathbb{L}$-parameterization iff $\mathbb{S} = \big\langle S, \circ, \langle \boldsymbol{I}_Y, \boldsymbol{I}_Y\rangle\big\rangle$ is a monoid.

Example 2. Consider the algebra $\mathbb{L}$ introduced in Example 1, an arbitrary non-empty set Y and, for each $\ell \in L$, an isotone Galois connection $\langle \boldsymbol{f}_\ell, \boldsymbol{g}_\ell\rangle$ in $\langle L^Y, \subseteq\rangle$ defined as follows: for all $A \in L^Y$ and $y \in Y$,

$$\boldsymbol{f}_\ell(A)(y) = \max\{0, A(y) - \ell\} \quad \text{and} \quad \boldsymbol{g}_\ell(A)(y) = \min\{1, A(y) + \ell\}.$$

In particular, $\boldsymbol{f}_1 = \mathbf{0}_Y$, $\boldsymbol{g}_1 = \mathbf{1}_Y$, and $\boldsymbol{f}_0 = \boldsymbol{g}_0 = \boldsymbol{I}_Y$.

The family $S = \{\langle \boldsymbol{f}_{\frac{i}{5}}, \boldsymbol{g}_{\frac{i}{5}}\rangle \mid i \in \mathbb{N}, 0 \leq i \leq 5\}$ is an $\mathbb{L}$-parameterization.

3 Parameterized Simplification Logic

Given a non-empty *alphabet* Y, whose elements are named *attributes*, the set of well-formed formulas of the language is:

$$\mathcal{L}_Y = \{A \Rightarrow B \mid A, B \in L^Y\}.$$

These well-formed formulas will be named *implications* and, in each implication, the first and the second component will be named *premise* and *conclusion* respectively. Finally, the sets of implications $\Sigma \subseteq \mathcal{L}$ will be named *theories.*

We have just introduced the syntax of our logic. In the rest of the section we complete its formal presentation. Thus, first we introduce the semantics of the logic, second we present an axiomatic system and, finally, we show that both the semantic and the syntactic points of views coincide proving the soundness and completeness of the axiomatic system.

3.1 Semantics

Before we define the interpretation of formulas, we introduce S-additive $\mathbb{L}$-fuzzy sets that will play the role of models.

Definition 2. *Let Y be a non-empty set and S be an $\mathbb{L}$-parameterization. An $\mathbb{L}$-fuzzy set $A \in L^Y$ is said to be S-*additive *if $\boldsymbol{f}(B) \subseteq A$ and $\boldsymbol{f}(C) \subseteq A$ imply $\boldsymbol{f}(B \oplus C) \subseteq A$, for all $B, C \in L^Y$ and $\langle \boldsymbol{f}, \boldsymbol{g} \rangle \in S$.*

The following proposition follows directly from Definition 2 and (17).

Proposition 1. *Let Y be a non-empty set and S be an $\mathbb{L}$-parameterization. An $\mathbb{L}$-fuzzy set $A \in L^Y$ is S-additive if and only if $\boldsymbol{g}(A) \oplus \boldsymbol{g}(A) = \boldsymbol{g}(A)$.*

Example 3. Let $\mathbb{L}$ be the algebra introduced in Example 1, S be the $\mathbb{L}$-parameterization introduced in Example 2, and Y be an arbitrary non-empty set. A set $A \in L^Y$ is S-additive if and only if, for all $y \in Y$, $A(y) = 0$ or $A(y) \geq \frac{1}{2}$.

Fixed S being an $\mathbb{L}$-parameterization, the models of the logic are defined in terms of S-additive $\mathbb{L}$-sets as follows:

Definition 3. *Let $A \Rightarrow B \in \mathcal{L}_Y$. An S-additive set $M \in L^Y$ is said to be a* model *for $A \Rightarrow B$ if $\boldsymbol{f}(A) \subseteq M$ implies $\boldsymbol{f}(B) \subseteq M$, for all $\langle \boldsymbol{f}, \boldsymbol{g} \rangle \in S$.*

The set of models for $A \Rightarrow B$ is denoted by $\mathcal{M}od(A \Rightarrow B)$. As usual, we say that an S-additive set M is model for a theory $\Sigma \subseteq \mathcal{L}_Y$ if it is model for all the implications $A \Rightarrow B \in \Sigma$, that is, $\mathcal{M}od(\Sigma) = \bigcap_{A \Rightarrow B \in \Sigma} \mathcal{M}od(A \Rightarrow B)$.

As it is usual for graded attribute implications, we can interpret our formulas in $\mathbb{L}$-contexts. An $\mathbb{L}$-context $\mathbf{I} = \langle X, Y, I \rangle$ consists of a non-empty sets X (and Y) of objects (and attributes—as before) and a map $I : X \times Y \to L$. For $x \in X$, we consider $I_x \in L^Y$ such that $I_x(y) = I(x, y)$ for all $y \in Y$. An $\mathbb{L}$-context $\mathbf{I} = \langle X, Y, I \rangle$ is called a model of $A \Rightarrow B$ whenever $\{I_x \mid x \in X\} \subseteq \mathcal{M}od(A \Rightarrow B)$.

Example 4. Consider the algebra $\mathbb{L}$ introduced in Example 1 and the $\mathbb{L}$-parameterization S introduced in Example 2. For the following $\mathbb{L}$-contexts

$\mathbf{I}_1$	y_1	y_2	y_3
x_1	$\frac{3}{5}$	$\frac{3}{5}$	1
x_2	1	1	0
x_3	0	$\frac{4}{5}$	$\frac{4}{5}$

$\mathbf{I}_2$	y_1	y_2	y_3
x_1	$\frac{7}{10}$	$\frac{7}{10}$	1
x_2	1	1	0
x_3	0	$\frac{1}{2}$	$\frac{1}{2}$

we have that $\mathbf{I}_1$ is model for $\{y_1/\frac{9}{10}\} \Rightarrow \{y_2/1\}$. In contrast, $\mathbf{I}_2$ is not model for this implication: $\boldsymbol{f}_{\frac{1}{5}}(\{y_1/\frac{9}{10}\}) = \{y_1/\frac{7}{10}\} \subseteq I_{x_1}$ and $\boldsymbol{f}_{\frac{1}{5}}(\{y_2/1\}) = \{y_1/\frac{4}{5}\} \not\subseteq I_{x_1}$.

Definition 4. *Let $A \Rightarrow B \in \mathcal{L}_Y$ and $\Sigma_1, \Sigma_2 \subseteq \mathcal{L}_Y$.*

- *The implication $A \Rightarrow B$ is said to be* semantically derived *from the theory Σ_1 if $\mathcal{M}od(\Sigma_1) \subseteq \mathcal{M}od(A \Rightarrow B)$. It is denoted by $\Sigma_1 \models A \Rightarrow B$.*
- *Both theories Σ_1 and Σ_2 are said to be* semantically equivalent *if $\mathcal{M}od(\Sigma_1) = \mathcal{M}od(\Sigma_2)$. It is denoted by $\Sigma_1 \equiv \Sigma_2$.*

3.2 Inference System

We look for a syntactic inference system capable of characterizing the semantic entailment $\models$ as defined before. In this subsection, we introduce the inference system and prove its soundness.

Definition 5. *For all $A, B, C, D \in L^Y$ and $\langle \boldsymbol{f}, \boldsymbol{g} \rangle \in S$, the inference system consists of following axiom scheme:*

Reflexivity: Infer $A \Rightarrow A$, (Ref)

together the following inference rules:

Composition: From $A \Rightarrow B$ and $A \Rightarrow C$ infer $A \Rightarrow B \oplus C$, (Comp)

Simplification: From $A \Rightarrow B$ and $C \Rightarrow D$ infer $A \oplus (C \ominus B) \Rightarrow D$, (Simp)

Extension: From $A \Rightarrow B$ infer $\boldsymbol{f}(A) \Rightarrow \boldsymbol{f}(B)$. (Ext)

The notion of syntactic derivation, or inference, is introduced in the standard way.

Definition 6 (Syntactic derivation). *An implication $A \Rightarrow B \in \mathcal{L}_Y$ is said to be syntactically derived or inferred from a theory $\Sigma \subseteq \mathcal{L}_Y$, denoted by $\Sigma \vdash A \Rightarrow B$, if there exists a sequence $\sigma_1, \ldots, \sigma_n \in \mathcal{L}_Y$ such that σ_n is the implication $A \Rightarrow B$ and, for all $1 \leq i \leq n$, one of the following conditions holds:*

- *$\sigma_i \in \Sigma$;*
- *σ_i is an axiom obtained from* (Ref)*;*
- *σ_i is obtained by applying any of the inference rules* (Comp)*,* (Simp)*, or* (Ext) *to formulas in $\{\sigma_j \mid 1 \leq j < i\}$.*

Theorem 1 (Soundness). *For any implication $A \Rightarrow B \in \mathcal{L}_Y$ and any theory $\Sigma \subseteq \mathcal{L}_Y$, it follows that $\Sigma \vdash A \Rightarrow B$ implies $\Sigma \models A \Rightarrow B$.*

Proof. Assume that $\Sigma \vdash A \Rightarrow B$, i.e. there exists a sequence $\sigma_1, \ldots, \sigma_n \in \mathcal{L}_Y$ such that the conditions in Definition 6 hold. We prove that any model $M \in \mathcal{M}od(\Sigma)$ is model for σ_i for all $1 \leq i \leq n$ and, therefore, $M \in \mathcal{M}od(A \Rightarrow B)$.

It is straightforward that, if σ_i is an axiom or belongs to Σ, the set M is a model for σ_i. Assume now that $M \in \mathcal{M}od\{\sigma_j \mid 1 \leq j < i\}$ and prove that M is model for any formula that is obtained by applying (Comp), (Simp) or (Ext).

We only show the proof for (Simp) because the cases of (Comp) and (Ext) are straightforward from the facts that the models are S-additive and S is closed under compositions, respectively.

Consider $U_1 \Rightarrow V_1, U_2 \Rightarrow V_2 \in \{\sigma_j \mid 1 \leq j < i\}$. Since M is model for these implications, we have that $\boldsymbol{f}(U_k) \subseteq M$ implies $\boldsymbol{f}(V_k) \subseteq M$, for all $\langle \boldsymbol{f}, \boldsymbol{g} \rangle \in S$ and $k \in \{1, 2\}$. We must prove that M is model for $U_1 \oplus (U_2 \ominus V_1) \Rightarrow V_2$.

Consider $\langle \boldsymbol{f}, \boldsymbol{g} \rangle \in S$ such that $\boldsymbol{f}(U_1 \oplus (U_2 \ominus V_1)) \subseteq M$. Since $\boldsymbol{f}$ is isotone and $U_1 \subseteq U_1 \oplus (U_2 \ominus V_1)$, we have that $\boldsymbol{f}(U_1) \subseteq M$ and, therefore, $\boldsymbol{f}(V_1) \subseteq M$. Now, from the S-additivity of M, $\boldsymbol{f}(V_1 \oplus U_1 \oplus (U_2 \ominus V_1)) \subseteq M$. From (9), we have $U_2 \subseteq V_1 \oplus (U_2 \ominus V_1) \subseteq V_1 \oplus U_1 \oplus (U_2 \ominus V_1)$ and, therefore, $\boldsymbol{f}(U_2) \subseteq M$. Finally, since M is model for $U_2 \Rightarrow V_2$, we have that $\boldsymbol{f}(V_2) \subseteq M$. □

4 Basic Properties

In this section we show equivalences that are derived from the primitive inference rules and allow us to remove redundant information, i.e., simplify theories. In the following proposition we introduce some derived inference rules.

Proposition 2. *The following rules are derived from the axiomatic system:*

Generalized Reflexivity : $\vdash A \Rightarrow B$ *when* $B \subseteq A$ (`GRef`)

Transitivity : $A \Rightarrow B, B \Rightarrow C \vdash A \Rightarrow C$ (`Tran`)

Generalization : $A \Rightarrow B \vdash C \Rightarrow D$ *when* $A \subseteq C$ *and* $D \subseteq B$ (`Gen`)

Generalized Composition : $A \Rightarrow B, C \Rightarrow D \vdash A \cup C \Rightarrow B \oplus D$ (`GComp`)

Augmentation : $A \Rightarrow B \vdash A \cup C \Rightarrow B \oplus C$ (`Augm`)

Generalized Transitivity : $A \Rightarrow B, B \cup C \Rightarrow D \vdash A \cup C \Rightarrow D$ (`GTran`)

Proof. All (`GRef`)–(`GTran`) can be verified using properties of $\oplus$ and $\ominus$ in $\mathbb{L}$. □

One outstanding characteristic of Simplification logic is that their inference rules induces a set of equivalences, providing a way to design automated prover methods strongly based in the axiomatic system presented in Definition 5. In the following proposition we present these equivalences.

Proposition 3. *The following equivalences hold:*

Decomposition: $\{A \Rightarrow B\} \equiv \{A \Rightarrow B \ominus A\}$ (`DeEq`)

Composition: $\{A \Rightarrow B, A \Rightarrow C\} \equiv \{A \Rightarrow B \oplus C\}$ (`CoEq`)

Simplification: *if* $A \subseteq C, \{A \Rightarrow B, C \Rightarrow D\} \equiv \{A \Rightarrow B, C \ominus B \Rightarrow D \ominus B\}$ (`CoEq`)

Proof. These equivalences, read from left to right, follow directly from (`GRef`), (`Comp`), and (`Simp`). For limitations of space we will prove the opposite direction only for (`DeEq`): In order to see that $\{A \Rightarrow B \ominus A\} \vdash A \Rightarrow B$ holds, observe that

(i) $A \Rightarrow B \ominus A$ .... by Hypothesis.
(ii) $A \Rightarrow A$ .... by (`Ref`).
(iii) $A \Rightarrow A \oplus (B \ominus A)$ .... by (i), (ii) and (`Augm`).
(iv) $A \Rightarrow B$ .... by (iii) and (`Gen`).

In the last step, we have utilized the fact that $B \subseteq A \oplus B \subseteq A \oplus (B \ominus A)$. □

5 Syntactic Closure and Completeness

In this section, we prove the completeness of the axiomatic system in the case of both $\mathbb{L}$ and Y are finite. First, we consider, in this framework, the generalization of the notion of syntactic closure of an $\mathbb{L}$-set.

Theorem 2. *Let* $\Sigma \subseteq \mathcal{L}_Y$ *be a theory. If* L^Y *is finite, the mapping* $\boldsymbol{c}_\Sigma : L^Y \to L^Y$ *defined as follows: for each* $A \in L^Y$,

$$\boldsymbol{c}_\Sigma(A) = \bigcup \{B \in L^Y \mid \Sigma \vdash A \Rightarrow B\}$$

is a closure operator in $\langle L^Y, \subseteq \rangle$. In addition,

$$\Sigma \vdash A \Rightarrow B \textit{ if and only if } B \subseteq \boldsymbol{c}_\Sigma(A) \textit{ for all } A, B \in L^Y.$$

Proof (Sketch). From (`Ref`) and (`Tran`), we easily obtain that $\boldsymbol{c}_\Sigma$ is extensive and isotone. Now, since L^Y is finite, applying (`Comp`) and (`Gen`) a finite number of times we get $\Sigma \vdash A \Rightarrow \boldsymbol{c}_\Sigma(A)$. The rest is obvious. □

Definition 7 (Syntactic closure). *Given $\Sigma \subseteq \mathcal{L}_Y$ and $A \in L^Y$, the set $\boldsymbol{c}_\Sigma(A)$ is called* syntactic closure *of A with respect to Σ.*

Theorem 3. *If L^Y is finite, for any theory $\Sigma \subseteq \mathcal{L}_Y$, we have that*

$$\mathcal{M}od(\Sigma) = \{\boldsymbol{c}_\Sigma(A) \mid A \in L^Y\}.$$

Proof. First, for all $A \in L^Y$, we prove that $\boldsymbol{c}_\Sigma(A)$ is S-additive: given $\langle \boldsymbol{f}, \boldsymbol{g} \rangle \in S$, if $\boldsymbol{f}(B) \subseteq \boldsymbol{c}_\Sigma(A)$ and $\boldsymbol{f}(C) \subseteq \boldsymbol{c}_\Sigma(A)$, from Theorem 2, $\Sigma \vdash A \Rightarrow \boldsymbol{f}(B)$ and $\Sigma \vdash A \Rightarrow \boldsymbol{f}(C)$. The following sequence prove that $\Sigma \vdash A \Rightarrow \boldsymbol{f}(B \oplus C)$ and, therefore, $\boldsymbol{f}(B \oplus C) \subseteq \boldsymbol{c}_\Sigma(A)$.

(i) $A \Rightarrow \boldsymbol{f}(B)$ .... by Hypothesis.
(ii) $A \Rightarrow \boldsymbol{f}(C)$ .... by Hypothesis.
(iii) $A \Rightarrow \boldsymbol{f}(B) \oplus \boldsymbol{f}(C)$ .... by (i), (ii) and (`Comp`).
(iv) $\boldsymbol{f}(B) \oplus \boldsymbol{f}(C) \Rightarrow \boldsymbol{f}(B \cup C)$ .... by (`GRef`).
(v) $A \Rightarrow \boldsymbol{f}(B \cup C)$ .... by (iii), (iv) and (`Tran`).
(vi) $B \Rightarrow B$ .... by (`Ref`).
(vii) $B \cup C \Rightarrow B \oplus C$ .... by (vi) and (`Augm`).
(viii) $\boldsymbol{f}(B \cup C) \Rightarrow \boldsymbol{f}(B \oplus C)$ .... by (vii) and (`Ext`).
(ix) $A \Rightarrow \boldsymbol{f}(B \oplus C)$ .... by (v), (viii) and (`Tran`).

In (iv), we have considered that $\boldsymbol{f}(B \cup C) = \boldsymbol{f}(B) \cup \boldsymbol{f}(C) \subseteq \boldsymbol{f}(B) \oplus \boldsymbol{f}(C)$ because $\langle \boldsymbol{f}, \boldsymbol{g} \rangle$ is an isotone Galois connection and (14) holds.

Second, we prove that $\boldsymbol{c}_\Sigma(A)$ is model for Σ: for all $\langle \boldsymbol{f}, \boldsymbol{g} \rangle \in S$, if $U \Rightarrow V \in \Sigma$ and $\boldsymbol{f}(U) \subseteq \boldsymbol{c}_\Sigma(A)$, then $\Sigma \vdash A \Rightarrow \boldsymbol{f}(U)$ and, by (`Ext`), $\Sigma \vdash \boldsymbol{f}(U) \Rightarrow \boldsymbol{f}(V)$. Therefore, by (`Tran`), $\Sigma \vdash A \Rightarrow \boldsymbol{f}(V)$ and $\boldsymbol{f}(V) \subseteq \boldsymbol{c}_\Sigma(A)$.

Finally, it is straightforward that $\boldsymbol{c}_\Sigma(M) = M$ for any $M \in \mathcal{M}od(\Sigma)$. □

We already have the necessary results to ensure that everything that can be semantically derived can also be syntactically inferred.

Theorem 4 (Completeness). *If L^Y is finite, $\Sigma \models A \Rightarrow B$ implies $\Sigma \vdash A \Rightarrow B$, for any $\Sigma \subseteq \mathcal{L}_Y$ and $A \Rightarrow B \in \mathcal{L}_Y$.*

Proof. If $\Sigma \not\vdash A \Rightarrow B$, then, from Theorem 3, $\boldsymbol{c}_\Sigma(A) \in \mathcal{M}od(\Sigma)$ but, from Theorem 2, $\boldsymbol{c}_\Sigma(A) \notin \mathcal{M}od(A \Rightarrow B)$. Therefore, $\Sigma \not\models A \Rightarrow B$. □

Returning to the graded attribute implications parameterized by hedges, it can be easily seen that our inference system and the complete logic presented in our paper generalizes the simplification logic for (FASL) from [4]. Indeed, one may put $\oplus = \vee$ and let $\ominus$ be the adjoint operation satisfying (8). Furthermore, given a hedge *, one can consider an $\mathbb{L}$-parameterization S which consists of all $\langle \boldsymbol{f}_{c^*\otimes}, \boldsymbol{g}_{c^*\to}\rangle$ where $(\boldsymbol{f}_{c^*\otimes}(A))(y) = c^* \otimes A(y)$ and $(\boldsymbol{g}_{c^*\to}(A))(y) = c^* \to A(y)$ for any $A \in L^Y$, $c \in L$, and $y \in Y$. In this setting, our inference system coincides with the inference system of FASL. In particular, the *rule of multiplication* (from $A \Rightarrow B$ infer $c^*{\otimes}A \Rightarrow c^*{\otimes}B$), cf. also [6,26], coincides with (Ext).

6 Conclusions

In this work, we have proposed a parameterized simplification logic for reasoning with graded implications in formal concept analysis. To achieve this goal, we have used systems of isotone Galois connections to handle a large family of possible interpretations in data dependencies. As it is usual, the logic was described in terms of a formal language, the semantics, and the axiomatic system. We proved its soundness and completeness. We showed how FASL proposed in [4] is a particular case of the parameterized simplification logic proposed in the present paper. In addition, different logics can be seen as particular cases of the general setting established here. Future research will focus on efficient algorithms based on the proposed logic.

Acknowledgment

Supported by Grants TIN2014-59471-P and TIN2017-89023-P of the Science and Innovation Ministry of Spain, co-funded by the European Regional Development Fund (ERDF). V. Vychodil was also supported the ECOP (Education for Competitiveness Operational Programme) project no. CZ.1.07/2.3.00/20.0059, which was co-financed by the European Social Fund and the state budget of the Czech Republic during 2011–2014.

References

1. Armstrong, W.W.: Dependency structures of data base relationships. In: Rosenfeld, J.L., Freeman, H. (eds.) Information Processing 74: Proceedings of IFIP Congress. pp. 580–583. North Holland, Amsterdam (1974)
2. Belohlavek, R.: Fuzzy Relational Systems: Foundations and Principles. Kluwer Academic Publishers, Norwell, MA, USA (2002)
3. Belohlavek, R., Chlupova, M., Vychodil, V.: Implications from data with fuzzy attributes. In: Proc. 2004 IEEE International Conference on Advances in I ntelligent Systems—Theory and Applications. University of Canberra, Centre de Recherche Public Henri Tudor (2004), 5 pages
4. Belohlavek, R., Cordero, P., Enciso, M., Mora, A., Vychodil, V.: Automated prover for attribute dependencies in data with grades. International Journal of Approximate Reasoning 70, 51–67 (2016)
5. Belohlavek, R., Vychodil, V.: Formal concept analysis and linguistic hedges. International Journal of General Systems 41(5), 503–532 (2012)
6. Belohlavek, R., Vychodil, V.: Attribute dependencies for data with grades I. International Journal of General Systems 45(7–8), 864–888 (2016)
7. Belohlavek, R., Vychodil, V.: Attribute dependencies for data with grades II. International Journal of General Systems 46(1), 66–92 (2017)

8. Cordero, P., Enciso, M., Mora, A., Ojeda-Aciego, M.: Computing minimal generators from implications: a logic-guided approach. In: Proc. of Concept Lattices and Applications, CLA 2012. pp. 187–198 (2012)
9. Cordero, P., Enciso, M.M., Mora, A., de Guzmán, I.P.I., Mora, Á., Pérez de Guzman, I.: SLFD Logic: Elimination of Data Redundancy in Knowledge Representation 2527, 141–150 (2002)
10. Ganter, B., Wille, R.: Formal Concept Analysis: Mathematical Foundations. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 1st edn. (1997)
11. Goguen, J.A.: The logic of inexact concepts. Synthese 19, 325–373 (1969)
12. Guigues, J.L., Duquenne, V.: Familles minimales d'implications informatives resultant d'un tableau de données binaires. Math. Sci. Humaines 95, 5–18 (1986)
13. Hájek, P.: Metamathematics of Fuzzy Logic. Kluwer Academic Publishers, Dordrecht, The Netherlands (1998)
14. Hájek, P.: On very true. Fuzzy Sets and Systems 124(3), 329–333 (2001)
15. Lorenzo, E.R., Adaricheva, K.V., Cordero, P., Enciso, M., Mora, A.: From an Implicational System to its Corresponding D-basis. In: Proceedings of the Twelfth International Conference on Concept Lattices and Their Applications, Clermont-Ferrand, France, October 13-16, 2015. pp. 217–228 (2015)
16. Lorenzo, E.R., Bertet, K., Cordero, P., Enciso, M., Mora, A.: The direct-optimal basis via reductions. In: Proceedings of the Eleventh International Conference on Concept Lattices and Their Applications, Košice, Slovakia, October 7-10, 2014. pp. 145–156 (2014)
17. Mora, A., Cordero, P., Enciso, M., Fortes, I., Aguilera, G.: Closure via functional dependence simplification. International Journal of Computer Mathematics 89(4), 510–526 (2012)
18. Mora, Á., Enciso, M., Cordero, P., de Guzmán, I.P.P.: An Efficient Preprocessing Transformation for Functional Dependencies Sets Based on the Substitution Paradigm. Lecture Notes in Computer Science, vol. 3040, pp. 136–146 (2003)
19. Pollandt, S.: Fuzzy-Begriffe: Formale Begriffsanalyse unscharfer Daten. Springer (1997)
20. Rauszer, C.: Semi-boolean algebras and their applications to intuitionistic logic with dual operations. Fundamenta Mathematicae 83(3), 219–249 (1974)
21. Stone, M.H.: Topological representations of distributive lattices and brouwerian logics. Časopis pro pěstování matematiky a fysiky 67(1), 1–25 (1938)
22. Takeuti, G., Titani, S.: Globalization of intuitionistic set theory. Annals of Pure and Applied Logic 33, 195–211 (1987)
23. Triska, J., Vychodil, V.: Logic of temporal attribute implications. Annals of Mathematics and Artificial Intelligence 79(4), 307–335 (2017)
24. Vychodil, V.: On minimal sets of graded attribute implications. Information Sciences 294, 478–488 (2015)
25. Vychodil, V.: Computing sets of graded attribute implications with witnessed nonredundancy. Information Sciences 351, 90–100 (2016)
26. Vychodil, V.: Parameterizing the semantics of fuzzy attribute implications by systems of isotone Galois connections. IEEE Trans. on Fuzzy Systems 24, 645–660 (2016)
27. Vychodil, V.: Closure structures parameterized by systems of isotone Galois connections. International Journal of Approximate Reasoning 91, 1–21 (2017)
28. Ward, M., Dilworth, R.P.: Residuated lattices. Trans. Amer. Math. Soc. 45, 335–354 (1939)
29. Zadeh, L.A.: A fuzzy-set-theoretic interpretation of linguistic hedges. Journal of Cybernetics 2(3), 4–34 (1972)

A Similarity Measure to Generalize Attributes

Rostand S. Kuitché[1], Romuald E. A. Temgoua[2], and Léonard Kwuida[3]

[1] Université de Yaoundé I, Département des Mathematiques,
BP 812 Yaoundé, Cameroun
kuitcher@yahoo.com

[2] Université de Yaoundé I, École Normale Supérieure,
BP 47 Yaoundé, Cameroun
retemgoua@gmail.com

[3] Bern University of Applied Sciences,
Brückenstrasse 73, 3005 Bern, Suisse
leonard.kwuida@bfh.ch

Abstract. Formal Concept Analysis (FCA) plays a crucial role in various domains, especially in qualitative data analysis. Here knowledge are extracted from an information system in form of clusters (forming a concept lattice) or in form of rules (implications basis). The number of extracted pieces of information can grow very fast. To control the number of cluster, one possibility is to put some attributes together to get a new attribute called a generalized attribute. However, generalizing does not always lead to the expected results: the number of concepts can even exponentially increase after generalizing two attributes [7,8]. A natural question is whether there is a similarity measure, (possibly cheap and fast to compute), that is compatible with generalizing attributes: i.e. if m_1, m_2 are **more similar** than m_3, m_4, then putting m_1, m_2 together should not lead to more concepts as putting m_3, m_4 together. This paper is an attempt to answer this question.

Keywords: Formal Concept Analysis; Generalizing Attributes; Similarity Measures.

1 Introduction

In Formal Concept Analysis (FCA), a **formal context** is a binary relation (G, M, I) that models an elementary information system, whereby G is the set of objects, M the set of attributes and $I \subseteq G \times M$ the incidence relation. To extract knowledge from such an elementary information system, one possibility is to get clusters of objects and/or attributes by grouping together those sharing the same characteristics. These pairs, called **concepts**, were formalized by Rudolf Wille [16]. For $A \subseteq G$ and $B \subseteq M$ we set

$$A' = \{m \in M \mid g \,\mathrm{I}\, m \text{ for all } g \in A\} \text{ and}$$
$$B' = \{g \in M \mid g \,\mathrm{I}\, m \text{ for all } m \in B\}.$$

© Dmitry I. Ignatov, Lhouari Nourine (Eds.): CLA 2018, pp. 141–152,
ISBN 978–80–244–5328–6, Department of Computer Science, Palacký University
Olomouc, 2018.

A **concept** is a pair (A, B) such that $A' = B$ and $B' = A$. A is called **extent** and B **intent** of the concept (A, B). The set of concepts of a context $\mathbb{K} := (G, M, \mathrm{I})$ is ordered by the relation $(A, B) \leq (C, D) :\iff A \subseteq C$, and forms a lattice, denoted by $\mathfrak{B}(\mathbb{K})$ and called **concept lattice** of $\mathbb{K}$. To control the size of concept lattices, many methods have been suggested: decomposition [18,19,17], iceberg lattices [14] α-Galois lattices [15], fault tolerant patterns [3], closure or kernel operators and/or approximation [6]. In [7] the authors consider putting together some attributes to get a generalized attribute. Doing this one has to decide when an object satisfies a (new) generalized attribute. They discuss several scenarios among which the following, called $\exists$-generalization:

> an object $g \in G$ satisfies a generalized attribute $s \subseteq M$ if g satisfies at least one of the attributes in s. i.e. $s' = \bigcup\{m' \mid m \in s\}$.

In the rest of this contribution, we will simply say **generalization** to mean $\exists$-generalization. By generalizing (i.e putting together some attributes) we reduce the number of attributes and hope to also reduce the size of the concept lattice. Unfortunately this is not always the case. In [8] the authors provide some examples where the size increases exponentially after generalizing two attributes and also give the maximal increase.

In [1,5], the authors discuss similarity measures on concepts, and even on lattices. For our purpose, we need a measure of similarity on attributes such that if m_1, m_2 are more similar than m_3, m_4, then generalizing m_1, m_2 should not lead to more concepts as generalizing m_3, m_4. We say that such a similarity measure is **compatible with the generalization**. Given a set M of attributes, a **similarity measure** on M is defined as a function $S : M \times M \to \mathbb{R}$ such that for all m_1, m_2 in M,

(i) $S(m_1, m_2) \geq 0$, **positivity**
(ii) $S(m_1, m_2) = S(m_2, m_1)$ **symmetry**
(iii) $S(m_1, m_1) \geq S(m_1, m_2)$ **maximality**

If in addition $S(m_1, m_2) \leq 1$, we say that S is **normalized**. Similarity measures aim at quantifying to which extent two attributes resemble each other. Getting a similarity measure compatible with the generalization will be a valuable tool in preprocessing and will warn the data analyst on possible lost or gain when generalizing.

The rest of the paper is organized as follows: In Section 2, we investigate the existing similarity measures that we found in the literature. In Section 3, we give a new similarity measure that characterize the pairs of attributes which can increase the size of the concept lattice after generalizing. Section 4 exposes an example on lexicographic data and Section 5 concludes the paper.

2 Test of Existing Similarity Measures in ∃-Generalization

Similarity and dissimilarity measures play a key role in pattern analysis problems such as classification, clustering, etc. Ever since *Pearson* proposed a coefficient of correlation in 1896, numerous similarity measures and distance have been proposed in various fields. These measures can be grouped into tree main types, depending of the data on which they are used:

Correlation coefficients: They are often used in data to compare variables with qualitative characters subdivided in more than two states.

Distance similarity coefficients: They are generally used in data with pure quantitative variables. In most cases, for quantitative data, the similarity between two taxa is expressed as a function of their distance in a dimensional space whose coordinates are the characters.

Coefficients of association: They are often used in data with presence-absence characters or in data with individuals having qualitative characters subdivided into two states.

There are two subsets of coefficients of association: those that only depend on characteristics present in at least one of the taxa compared, but are independent of the attributes absent in both taxa (denoted by type 1), and those that also take into account the attributes absent in both taxa (denoted by type 2). Those measures use

- a as the number of cases where the two variables occur together in a sample,
- d as the number of cases where none of the two attributes occur in a sample,
- b as the number of cases in which only the first variable occur, and
- c as the number of cases where only the second variable occur.

One of the most important similarity measure of type 1 is the **Jaccard measure** $\left(\frac{a}{a+b+c}\right)$, proposed in order to classify ecological species. Also in the ecological field, the **Dice coefficient of association** $\left(\frac{2a}{2a+b+c}\right)$ aims at quantifying the extent to which two different species are associated in a biotope, the **Sorensen coefficient of association** $\left(\frac{4a}{4a+b+c}\right)$ and the **Anderberg coefficient of association** $\left(\frac{8a}{8a+b+c}\right)$ are of the same type. The **Sneath and Sokal 2** similarity coefficient $\left(\frac{\frac{1}{2}a}{\frac{1}{2}a+b+c}\right)$, put in place in order to compare organisms in numerical taxonomy, the **Kulczynski similarity** measure $\left(\frac{1}{2}(\frac{a}{a+b}+\frac{a}{a+c})\right)$ and the **Ochiai similarity** measure $(\frac{a}{\sqrt{(a+b)(a+c)}})$ are also from this first type.

The most used similarity coefficient of the second type is the **Sokal and Michener** coefficient of association $\left(\frac{a+d}{a+d+b+c}\right)$, also called the **simple matching coefficient**, put in place to express the similarity between two species of bees. Moreover, the **Rogers and Tanimoto similarity measure** $(\frac{\frac{1}{2}(a+d)}{\frac{1}{2}(a+d)+b+c})$ whose aim was to compare species of plants in the ecological field, the **Sokal and Sneath 1** similarity coefficient $(\frac{2(a+d)}{2(a+d)+b+c})$ was defined to make comparison in numerical taxonomy and the **Russels and Rao** similarity measure $(\frac{a}{a+d+b+c})$ put in place with the aim of showing resemblance between species of *anopheline* *larvae,* are included in this type. Same are the **Yule and Kendall similarity coefficients** $\left(\frac{ad}{ad+bc}\right)$, often used in the statistical field. Some of the above similarity measures can be found in [5].

Regarding the definitions of the above kinds of similarity measures, only the coefficients of association suitable to formal contexts, since formal contexts are data with presence-absence characters. We will investigate the impact of these coefficients of association on a special pair of attributes in some formal contexts. The objective is to show that these similarity measures are not helpful in finding whether their generalization increases the size of the lattice or not.

Our first example is an arbitrary formal context (G, M, I) containing two attributes $x, y \in M$ such that $x' \subseteq y'$ and $|x' \cap y'| = 1$. Then $|x' \setminus y'| = 0$ and the generalization of the attributes x and y does not increase the size of the lattice. Choosing $|y' \setminus x'| = 20$ and $|G \setminus (x' \cup y')| = 1$ yields $a = |x' \cap y'| = 1$, $b = |x' \setminus y'| = 0$, $c = |y' \setminus x'| = 20$ and $d = |G \setminus (x' \cup y')| = 1$. For the coefficient of association of type 1 with Jaccard (Jc), Dice (Di), Sorensen (So), Anderberg (An), Sneath and Sokal 2 (SS_2), Kulczynski (Ku) and Orchiai (Orch), and the coefficient of association of type 2 with Sokal and Michener (SM), Rogers and Tanimoto (RT), Sneath and Sokal 1 (SS_1) and Russel and Rao (RR), we get the table below for $s(x, y)$:

Jc	Di	So	An	SS_2	Ku	Orch	SM	RT	SS1	RR
0,05	0,09	0,17	0,29	0,02	0,52	0,22	0,09	0,05	0,17	0,05

The table above shows that with almost all these measures, the similarity measured between the attributes x and y is very low, despite the fact that their generalization does not increase the size of the lattice.

Our second example is the formal context $\mathbb{K}_6 := (S_6 \cup \{g_1\}, S_6 \cup \{m_1, m_2\}, \mathrm{I})$ below, with $S_6 = \{1, 2, 3, 4, 5, 6\}$.

$\mathbb{K}_6$	1	2	3	4	5	6	m_1	m_2
1		×	×	×	×	×	×	
2	×		×	×	×	×	×	×
3	×	×		×	×	×	×	×
4	×	×	×		×	×	×	×
5	×	×	×	×		×	×	×
6	×	×	×	×	×			×
g_1	×	×	×	×	×	×		

We observe that $|m_1' \cap m_2'| = 4$, $|m_1' \backslash m_2'| = 1$ and $|m_2' \backslash m_1'| = 1$. Putting together the attributes m_1 and m_2 by a $\exists$-generalization increases the size of the lattice by 16. The following table shows the measures of type 1 and type 2 between the attribute m_1 and any other attribute i. All the similarity measures of the

	Jc	Di	So	An	SS_2	Ku	Orch	SM	RT	SS1	RR
$i \in S_5$	0,57	0,80	0,89	0,94	0,50	0,80	0,80	0,71	0,56	0,83	0,57
$i = 6$	0,83	0,91	0,95	0,97	0,71	0,92	0,91	0,75	0,75	0,92	0,71
$i = m_2$	0,67	0,80	0,89	0,94	0,50	0,80	0,80	0,71	0,56	0,83	0,57

two types show that the attribute m_1 is more similar to m_2 than to any other attribute $i \in S_6$ (apart from $i = 6$); But putting m_1 and m_2 together increases the size of the lattice. We can conclude that these similarity measures are not compatible with the ∃-generalization. We are actually looking for a measure on attributes that will flag pairs of attributes as **less similar** when putting these together increases the size of the concept lattice.

3 A Similarity Measure Compatible with ∃-Generalization

In this section we define a similarity measure on attributes which is compatible with the existential generalization. This generalization means that from an attribute reduced context $\mathbb{K} := (G, M, \mathrm{I})$, two attributes a, b are removed and replaced with an attribute s defined by $s' = a' \cup b'$. We set $M_0 := M \setminus \{a, b\}$ and

$$\begin{aligned} \mathbb{K}_{00} &:= (G, M_0, \mathrm{I} \cap (G \times M_0)), && \text{(removing } a, b \text{ from } \mathbb{K}) \\ \mathbb{K}_{0s} &:= (G, M_0 \,\dot{\cup} \{s\}, I_0^s), && \text{(adding } s \text{ to } \mathbb{K}_{00}) \end{aligned}$$

where $I_0^s := (\mathrm{I} \cap (G \times M_0)) \cup \{(g, s) \mid g \,\mathrm{I}\, b \text{ or } g \,\mathrm{I}\, a\}$. Furthermore we denote the set of extents of $\mathbb{K}_{00}$ by $\mathrm{Ext}(\mathbb{K}_{00})$. We also set

$$\begin{aligned} \mathcal{H}(a) &:= \{A \cap a' \mid A \in \mathrm{Ext}(\mathbb{K}_{00}) \text{ and } A \cap a' \notin \mathrm{Ext}(\mathbb{K}_{00})\}, \\ \mathcal{H}(b) &:= \{A \cap b' \mid A \in \mathrm{Ext}(\mathbb{K}_{00}) \text{ and } A \cap b' \notin \mathrm{Ext}(\mathbb{K}_{00})\}, \\ \mathcal{H}(a \cup b) &:= \{A \cap (a' \cup b') \mid A \in \mathrm{Ext}(\mathbb{K}_{00}) \text{ and } A \cap (a' \cup b') \notin \mathrm{Ext}(\mathbb{K}_{00})\}, \\ \mathcal{H}(a \cap b) &:= \{A \cap (a' \cap b') \mid A \in \mathrm{Ext}(\mathbb{K}_{00}) \text{ and } A \cap (a' \cap b') \notin \mathrm{Ext}(\mathbb{K}_{00})\}. \end{aligned}$$

We will often write $h(x)$ for $|\mathcal{H}(x)|$, for any $x \in \{a, b, a \cap b, a \cup b\}$. Before we start the construction, let us recall the following result partly proved in [8]:

Theorem 1. *Let $\mathbb{K} := (G, M, \mathrm{I})$ be an attribute reduced context with $|G| \geq 3$ and $|M| > 3$. Let a and b be two attributes such that their existential generalization $s = a \cup b$ increases the size of the concept lattice. Then*

a) $|\mathfrak{B}(\mathbb{K})| = |\mathfrak{B}(\mathbb{K}_{00})| + |\mathcal{H}(a, b)|$, with $|\mathcal{H}(a, b)| = |\mathcal{H}(a) \cup \mathcal{H}(b) \cup \mathcal{H}(a \cap b)|$.
b) The increase is $|\mathcal{H}(a \cup b)| - |\mathcal{H}(a, b)| \leq 2^{|a'|+|b'|} - 2^{|a'|} - 2^{|b'|} + 1$.

Proof. Let $\mathbb{K} := (G, M, \mathrm{I})$ be such context and a, b two attributes of $\mathbb{K}$. One proceeds to the ∃-generalization of attributes a and b.

a) We set $\mathbb{K}^a = (G, M \setminus \{b\}, \mathrm{I})$. It holds:

$$|\mathfrak{B}(\mathbb{K})| = |\mathfrak{B}(\mathbb{K}^a)| + h^*(b) = |\mathfrak{B}(\mathbb{K}_{00})| + h(a) + h^*(b)$$

where $h^*(b) = |\{B\cap b';\ B\in \mathrm{Ext}(\mathbb{K}^a),\ B\cap b' \notin \mathrm{Ext}(\mathbb{K}^a)\}|$. Our aim is to express $h^*(b)$ as a function of $h(b)$ and $h(a\cap b)$. According to [8], $\mathrm{Ext}(\mathbb{K}^a) = \mathrm{Ext}(\mathbb{K}_{00})\cup \mathcal{H}(a)$. Hence,

$$\begin{aligned}\mathcal{H}^*(b) &= \{B\cap b' \mid B\in \mathrm{Ext}(\mathbb{K}^a), B\cap b' \notin \mathrm{Ext}(\mathbb{K}^a)\}\\ &= \{B\cap b' \mid B\in \mathrm{Ext}(\mathbb{K}_{00}) \text{ and } B\cap b' \notin \mathrm{Ext}(\mathbb{K}^a)\}\\ &\qquad \cup \{B\cap b' \mid B\in \mathcal{H}(a) \text{ and } B\cap b' \notin \mathrm{Ext}(\mathbb{K}^a)\}\end{aligned}$$

Replacing $\mathrm{Ext}(\mathbb{K}^a)$ by $\mathrm{Ext}(\mathbb{K}_{00})\cup\mathcal{H}(a)$, we get

$$\{B\cap b' \mid B\in \mathrm{Ext}(\mathbb{K}_{00}) \text{ and } B\cap b' \notin \mathrm{Ext}(\mathbb{K}^a)\} = \mathcal{H}(b)\setminus\mathcal{H}(a) \quad \text{ and}$$

$$\{B\cap b' \mid B\in \mathcal{H}(a) \text{ and } B\cap b' \notin \mathrm{Ext}(\mathbb{K}^a)\} = \mathcal{H}(a\cap b)\setminus(\mathcal{H}(b)\cup\mathcal{H}(a)).$$

$$\begin{aligned}\text{Thus,}\quad h^*(b) &= h(b)+h(a\cap b) - |\mathcal{H}(a)\cap\mathcal{H}(b)| + |\mathcal{H}(a\cap b)\cap\mathcal{H}(a)\cap\mathcal{H}(b)|\\ &\quad - |\mathcal{H}(a\cap b)\cap\mathcal{H}(a)| - |\mathcal{H}(a\cap b)\cap\mathcal{H}(b)|.\end{aligned}$$

Hence,

$$\begin{aligned}|\mathfrak{B}(\mathbb{K})| &= |\mathfrak{B}(\mathbb{K}_{00})| + |\mathcal{H}(a)| + |\mathcal{H}(b)| + |\mathcal{H}(a\cap b)| + |\mathcal{H}(a\cap b)\cap\mathcal{H}(a)\cap\mathcal{H}(b)|\\ &\quad - |\mathcal{H}(a)\cap\mathcal{H}(b)| - |\mathcal{H}(a\cap b)\cap\mathcal{H}(a)| - |\mathcal{H}(a\cap b)\cap\mathcal{H}(b)|\\ &= |\mathfrak{B}(\mathbb{K}_{00})| + |\mathcal{H}(a)\cup\mathcal{H}(b)\cup\mathcal{H}(a\cap b)|.\end{aligned}$$

b) Although b) was proved in [8], we can now get it from a). To maximize the increase $a'\cap b'$ should be $\emptyset$; i.e. $|\mathcal{H}(a\cap b)|\in\{0,1\}$.

- If $|\mathcal{H}(a\cap b)| = 0$, then

$$\begin{aligned}|\mathfrak{B}(\mathbb{K})| &= |\mathfrak{B}(\mathbb{K}_{00})| + |\mathcal{H}(a)\cup\mathcal{H}(b)\cup\mathcal{H}(a\cap b)|\\ &= |\mathfrak{B}(\mathbb{K}_{00})| + |\mathcal{H}(a)| + |\mathcal{H}(b)|.\end{aligned}$$

- If $|\mathcal{H}(a\cap b)| = 1$, then we consider two subcases:
  – The only element of $\mathcal{H}(a\cap b)$ is not in $\mathcal{H}(a)\cup\mathcal{H}(b)$. Then,

$$\begin{aligned}|\mathcal{H}(a)\cap\mathcal{H}(b)| &= |\mathcal{H}(a\cap b)\cap\mathcal{H}(a)\cap\mathcal{H}(b)|\\ &= |\mathcal{H}(a\cap b)\cap\mathcal{H}(a)| = |\mathcal{H}(a\cap b)\cap\mathcal{H}(b)| = 0\end{aligned}$$

  and $|\mathfrak{B}(\mathbb{K})| = |\mathfrak{B}(\mathbb{K}_{00})| + |\mathcal{H}(a)| + |\mathcal{H}(b)| + |\mathcal{H}(a\cap b)|$.
  – The only element of $\mathcal{H}(a\cap b)$ is either in $\mathcal{H}(a)$ or $\mathcal{H}(b)$. Then

$$|\mathcal{H}(a\cap b)| + |\mathcal{H}(a\cap b)\cap\mathcal{H}(a)\cap\mathcal{H}(b)| - |\mathcal{H}(a\cap b)\cap\mathcal{H}(a)| - |\mathcal{H}(a\cap b)\cap\mathcal{H}(b)|$$

  is equal to zero and $|\mathcal{H}(a)\cap\mathcal{H}(b)|\in\{0,1\}$. Thus

$$|\mathfrak{B}(\mathbb{K})| = |\mathfrak{B}(\mathbb{K}_{00})| + |\mathcal{H}(a)| + |\mathcal{H}(b)| + 1 - |\mathcal{H}(a)\cap\mathcal{H}(b)|.$$

In all these subcases, considering that $|\mathfrak{B}(\mathbb{K}_{0s})| = |\mathfrak{B}(\mathbb{K}_{00})| + |\mathcal{H}(a \cup b)|$, the increase after the generalization is

$$\begin{aligned} |\mathfrak{B}(\mathbb{K}_{0s})| - |\mathfrak{B}(\mathbb{K})| &= |\mathcal{H}(a \cup b)| - |\mathcal{H}(a,b)| \\ &\leq 2^{|a'|+|b'|} - 2^{|a'|} - 2^{|b'|} + (d_1 + d_2 - d_0) \\ &\leq 2^{|a'|+|b'|} - 2^{|a'|} - 2^{|b'|} + 1, \text{ since } d_1 + d_2 - d_0 \leq 0, \end{aligned}$$

with $d_1 = |\{A \subseteq a' \mid A \in \text{Ext}(\mathbb{K}_{00})\}|$, $d_2 = |\{A \subseteq b' \mid A \in \text{Ext}(\mathbb{K}_{00})\}|$ and $d_0 = |\{A \subseteq a' \cup b' \mid A \in \text{Ext}(\mathbb{K}_{00})\}|$. □

Now, we define the following gain function:

$$\begin{aligned} \psi : M \times M &\longrightarrow \mathbb{Z} \\ (a,b) &\longmapsto \psi(a,b) = |\mathcal{H}(a \cup b)| - |\mathcal{H}(a,b)| \end{aligned}$$

Note that $\mathcal{H}(a \cup b) = \mathcal{H}(b \cup a)$, and $\mathcal{H}(a,b) = \mathcal{H}(b,a)$ because the order of adding the attributes a and b does not matter. Therefore $\psi(a,b) = \psi(b,a)$. By definition, $\psi(a,a) = 0$. Further, we define the map δ as followed:

$$\begin{aligned} \delta : M \times M &\longrightarrow \mathbb{R} \\ (a,b) &\longmapsto \begin{cases} 1 & \text{if } \psi(a,b) \leq 0 \\ 0 & \text{else} \end{cases} \end{aligned}$$

Since $\mathbb{K}$ is a finite context, there is a pair of attributes a_0, b_0 in M such that

$$|a_0'| + |b_0'| = \max_{a,b \in M} (|a'| + |b'|).$$

We set $n_0 = 2^{|a_0'|+|b_0'|} - 2^{|a_0'|} - 2^{|b_0'|} + 1$. Then $n_0 \geq 2^{|a'|+|b'|} - 2^{|a'|} - 2^{|b'|} + 1$ for all pairs $\{a,b\} \subseteq M$. With the function δ, we construct the following map:

$$\begin{aligned} S_{\text{gen}} : M \times M &\longrightarrow \mathbb{R} \\ (a,b) &\longmapsto S_{\text{gen}}(a,b) = \tfrac{1+\delta(a,b)}{2} - \tfrac{|\psi(a,b)|}{2n_0} \end{aligned}$$

where $|\psi(a,b)|$ is the absolute value of $\psi(a,b)$. That leads to the following results.

Proposition 1. *Let (G,M,I) be a reduced context with $|G| \geq 3$ and $|M| > 3$. Then S_{gen} is a normalized similarity measure on M.*

Proof. Let a, b two attributes of (G,M,I). Since $|\psi(a,b)| \leq n_0$ we can easily check that $0 \leq S_{\text{gen}}(a,b) = S_{\text{gen}}(b,a) \leq S_{\text{gen}}(a,a) = 1$ holds. □

S_{gen} also has the following properties:

Proposition 2. *Let (G,M,I) be a reduced context with $|G| \geq 3$ and $|M| > 3$. Let $a, b, c, d \in M$. It holds:*

a) $S_{gen}(a,b) \geq \frac{1}{2}$ *if and only if* $\psi(a,b) \leq 0$.

b) If $\psi(a,b) \leq 0 < \psi(d,c)$ *then* $S_{gen}(d,c) < S_{gen}(a,b)$.
c) If $0 < \psi(a,b) \leq \psi(d,c)$ *then* $S_{gen}(d,c) \leq S_{gen}(a,b)$.
d) If $\psi(a,b) \leq \psi(d,c) \leq 0$ *then* $S_{gen}(a,b) \leq S_{gen}(d,c)$.

Proof. Let $\mathbb{K} = (G, M, I)$ be such a context and $a, b, c, d \in M$.

a) If $\psi(a,b) \leq 0$ then $\delta(a,b) = 1$ and

$$S_{\text{gen}}(a,b) = \frac{1+\delta(a,b)}{2} - \frac{|\psi(a,b)|}{2n_0} = \frac{1}{2}\left(2 + \frac{\psi(a,b)}{n_0}\right) \geq \frac{1}{2}.$$

Now, $S_{\text{gen}}(a,b) \geq \frac{1}{2}$ implies $\frac{1+\delta(a,b)}{2} - \frac{|\psi(a,b)|}{2n_0} \geq \frac{1}{2}$ and $|\psi(a,b)| \leq n_0\delta(a,b)$. If $\delta(a,b) = 0$ then $|\psi(a,b)| = 0$. If $\delta(a,b) = 1$ then $\psi(a,b) \leq 0$ by definition of δ. Hence, $S_{\text{gen}}(a,b) \geq \frac{1}{2}$ if and only if $\psi(a,b) \leq 0$.

b) If $\psi(a,b) \leq 0 < \psi(d,c)$ then $S_{\text{gen}}(d,c) < \frac{1}{2} \leq S_{\text{gen}}(a,b)$.

c) If $0 < \psi(a,b) \leq \psi(d,c)$ then $\delta(a,b) = \delta(d,c) = 0$, and

$$S_{\text{gen}}(d,c) = \frac{1}{2} - \frac{\psi(d,c)}{2n_0} \leq \frac{1}{2} - \frac{\psi(a,b)}{2n_0} = S_{\text{gen}}(a,b).$$

d) If $\psi(a,b) \leq \psi(d,c) \leq 0$ then $\delta(a,b) = \delta(d,c) = 1$, and

$$S_{\text{gen}}(a,b) = 1 + \frac{\psi(a,b)}{2n_0} \leq 1 + \frac{\psi(d,c)}{2n_0} = S_{\text{gen}}(d,c).$$

□

Proposition 3. *Let* (G, M, I) *be a reduced context and* $a, b \in M$. *The following assertions are equivalent:*

(i) $\delta(a,b) = 1$.
(ii) $\psi(a,b) \leq 0$.
(iii) $S_{gen}(a,b) \geq \frac{1}{2}$.
(iv) *A* $\exists$*-generalization of* a *and* b *does not increase the size of the concept lattice.*

Proof. (i) $\iff$ (ii) follows from the definition of δ. (ii) $\iff$ (iii) is Proposition 2 a). (ii) $\iff$ (iv) follows from the fact that $\psi(a,b) = |\mathcal{H}(a \cup b)| - |\mathcal{H}(a,b)|$ is actually the difference $|\mathfrak{B}(G, M \cup \{s\} \setminus \{a,b\}, I)| - |\mathfrak{B}(G,M,I)|$ between the number of concepts before and after generalizing a, b to s with $s' = a' \cup b'$.

Therefore, generalizing two attributes a, b in a reduced context (G, M, I) increases the size of the lattice if and only if $S_{\text{gen}}(a,b) < \frac{1}{2}$. The threshold $\frac{1}{2}$ is just a consequence of the way S_{gen} has been defined.

To test our results we have designed a naive algorithm (see Algorithm 1) that computes S_{gen} on all pairs of attributes a, b of $\mathbb{K}$. If the set of attributes M is considered as a vector, then for any attribute $a \in M$, we set T(a) the set of all attributes coming before a in M. The complexity of our algorithm is given by

$$\sum_{a \in M} (1 + \sum_{b \in M \setminus T(a)} ((q(a,b) + 4)[4(q(a,b) + 1) + 4] + 3),$$

which is equal to

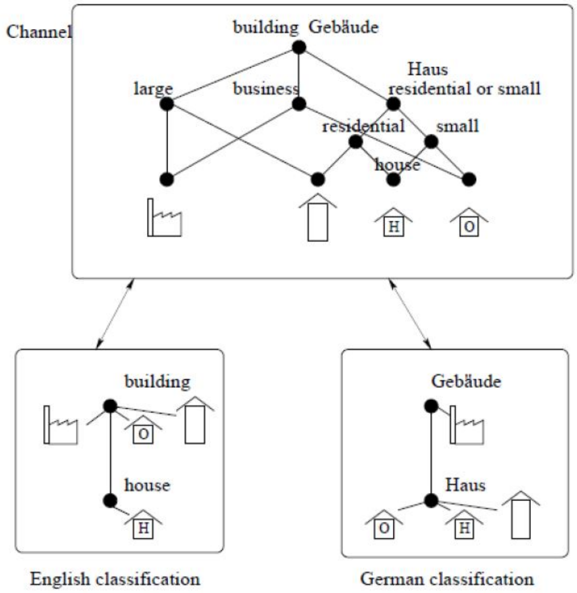
$$|M| + \sum_{a\in M}\sum_{b\in M\setminus T(a)} (4q^2(a,b) + 24q(a,b) + 35), \qquad \text{with } q(a,b) = |\,\mathrm{Ext}(\mathbb{K}_{00})|.$$

Algorithm 1: Computing a similarity measure

```
Data: An attribute reduced context $(G, M, I)$
Result: $\psi$ and $S_{\mathrm{gen}}$ on $M \times M$
Choose $x, y$ in $M$, $x \neq y$ with $|x'| + |y'|$ maximal;
$n_0 \leftarrow 2^{|x'|+|y'|} - 2^{|x'|} - 2^{|y'|} + 1$;
$T \leftarrow \emptyset$;
foreach $a$ in $M$ do
    $T \leftarrow T \cup \{a\}$;
    foreach $b$ in $M \setminus T$ do
        $\mathrm{Ext}_0 \leftarrow \mathrm{Ext}(G, M \setminus \{a,b\}, I)$;
        foreach $x$ in $\{a, b, a \cup b, a \cap b\}$ do $\mathcal{H}(x) \leftarrow \emptyset$;
        foreach $A$ in $\mathrm{Ext}_0$ do
            foreach $x$ in $\{a, b, a \cup b, a \cap b\}$ do
                if $A \cap x' \notin \mathrm{Ext}_0$ then $\mathcal{H}(x) \leftarrow \mathcal{H}(x) \cup \{A \cap x'\}$;
            end
        end
    end
    $\psi(a,b) \leftarrow |\mathcal{H}(a \cup b)| - |\mathcal{H}(a) \cup \mathcal{H}(b) \cup \mathcal{H}(a \cap b)|$;    $\psi(b,a) \leftarrow \psi(a,b)$;
    if $\psi(a,b) \leq 0$ then
        $\delta(a,b) \leftarrow 1$
    else
        $\delta(a,b) \leftarrow 0$
    end
    $S_{\mathrm{gen}(a,b)} \leftarrow \frac{1 + \delta(a,b)}{2} - \frac{|\psi(a,b)|}{2n_0}$
end
```

4 An Example from Lexicographic Data

Formal Concept Analysis has been applied to compare lexical databases. In [11] Uta Priss proposes an example in where the information channel is "*building*". With respect to this, the main difference between English and German is that in English, the word "house" only refers to small residential buildings whereas in German even small office buildings and large residential buildings can be called "Haus", and only factories would normally not be called "Haus". Moreover, "building" in English refers to either a factory, an office or even a big residential house. But only a factory can be called "Gebäude" in German. She presented in the figure below the information channel of the word "building" in the sense of Barwise and Seligman [2] in both English and German.



With the above information channel we can construct a formal context as follows: The objects are different kinds of buildings: small house ("h"), office ("o"), factory ("f") and large residential house ("l"). The attributes are different names of these objects in both languages: English and German. These are "building", "house", "Haus", "Gebäude", "large building" (short: "large"), "business building" (short: "business"), "residential house" (short: "residential"), and "small house" (short: "small"). Thus $G = \{$h, o, f, l$\}$ and $M = \{$"building", "house", "Haus", "Gebäude", "large", "business", "residential", "small"$\}$. In the following, a set of objects will be denoted as a concatenation of those objects. For example we will write ho or oh for the set $\{$h, o$\}$. The English and German classifications of the word "building" are then presented in the following formal context:

	building	house	Haus	Gebäude	large	business	residential	small
factory	×			×	×	×		
office	×		×			×		×
house		×	×				×	×
large	×		×		×		×	

For this formal context, $n_0 = 2^{3+3} - 2^3 - 2^3 + 1 = 49$. Let consider the attributes $a :=$ house and $b :=$ Gebäude. Then $a' \cup b' = \{f, h\}$ and $a' \cap b' = \emptyset$. We have

$$\text{Ext}(\mathbb{K}_{00}) = \{fohl, fol, ohl, fo, fl, ol, oh, hl, f, o, h, l, \emptyset\}, \text{ and}$$

$\mathcal{H}(a) = \mathcal{H}(b) = \mathcal{H}(a \cap b) = \emptyset$ and $\mathcal{H}(a \cup b) = \{fohl\}$. Therefore, $\psi(a, b) = 1$ and $S_{\text{gen}}(a, b) = \frac{1}{2} - \frac{1}{98} \approx 0.49$. Using our algorithm, we compute $\psi(a, b)$ and

$S_{\text{gen}}(a, b)$ for all pairs $a, b \in M$. The table below show $\psi(a, b)$ below the diagonal, and $S_{\text{gen}}(a, b)$ on the rest.

	building	house	Haus	Gebäude	large	business	residential	small
building	1.00	0.98	0.97	1.00	0.99	0.98	0.97	0.97
house	−2	1.00	1.00	0.49	0.49	0.49	1.00	1.00
Haus	−3	0	1.00	0.98	0.97	0.97	0.99	0.99
Gebäude	0	1	−2	1.00	1.00	1.00	0.49	0.49
large	−1	1	−3	0	1.00	0.98	0.49	0.97
business	−2	1	−3	0	−2	1.00	0.98	0.49
residential	−3	0	−1	1	1	−2	1.00	0.98
small	−3	0	−1	1	−3	1	−2	1.00

From the above table, the attributes ”house” and ”Gebäude” are less similar. It reflects the fact that these words ”Gebäude” (in German) and ”house” (in English) do not have the same meaning. It is also the case for the attributes ”house” and ”business buildings” as well as ”Gebäude” and ”residential building”. Hence, putting together each of the above pairs of attributes will increase the size of the lattice. On the contrary, the attributes ”large” and ”Haus”, ”building” and ”Haus” are more similar through S_{gen}. It is because the word ”Haus” which designates a house, a business office or simply large building in German, often coincides with the words ”building” or ”large building” in English. For these pairs, the existential generalization will not increase the size of the lattice.

5 Conclusion

We have constructed a similarity measure compatible with the change in the size of the lattice after a generalization of a pair of attributes in a formal context. That measure should send a warning when grouping two attributes. Also, it enables us to characterize contexts where generalizing two attributes increases the size of the concept lattice. Our next step is to look at the implication between generalized attributes. We suspect that the number of implications decreases if the number of concepts increases.

References

1. Alqadah, F., Bhatnagar, R.: Similarity Measures in Formal Concept Analysis. AMAI – Springer (2009)
2. Barwise, J., Seligman, J.: Information Flow: the logic of distributed systems. Cambridge University Press (1997)
3. Besson, J., Pensa, R. G., Robardet, C., Boulicaut, J.: Constraint-Based Mining of Fault-Tolerant Patterns from Boolean Data. KDID 55–71 (2005)
4. Dice, L. R.: Measures of the Amount of Ecologic Association Between Species. esa. Promoting the Science of Ecology Vol. 26, No 3, 297–302 (1945)

5. Domenach, F.: Similarity Measures of concept lattices In: Lausen B., Krolak-Schwerdt S., Böhmer M. (eds) Data Science, Learning by Latent Structures, and Knowledge Discovery. Studies in Classification, Data Analysis, and Knowledge Organization. Springer, Berlin, Heidelberg
6. Kwuida, L.: On concept lattice approximation. In Osamu Watanabe (Ed.) Foundations of Theoretical Computer Science: For New Computational View. RIMS No. 1599, Proceedings of the LA Symposium (2008), January 28–30, Kyoto, Japan, 42–50 (2008)
7. Kwuida, L., Missaoui, R., Balamane A., Vaillancourt, J.: Generalized pattern extraction from concept lattices. AMAI – Springer 151–168 (2014)
8. Kwuida, L., Kuitché, R. S., Temgoua, E. R. A.: On the Size of ∃-Generalized Concepts. ArXiv:1709.08060.
9. Ganter, B., Wille, R.: Formal Concept Analysis. Mathematical Foundations. Springer (1999)
10. Jaccard, P.: Nouvelle recherche sur la distribution florale. Bulletin de la Société Vaudoise des Sciences Naturelles (1908)
11. Priss, U.: Linguistic Applications of Formal Concept Analysis. Formal Concept Analysis, 149–160 (2005)
12. Rogers, D. J., Tanimoto, T. T.: A Computer Program for classifying plants. Springer (1960)
13. Sneath, P. H. A.: The Application of Computers to Taxonomy J. gen Microbiol. 17, 201–226 (1957)
14. Stumme, G., Taouil, R., Bastide, Y., Pasquier, N., Lakhal, L.: Computing iceberg concept lattices with Titanic. Data Knowl. Eng., 42(2): 189–222 (2002)
15. Ventos, V., Soldano, H., Lamadon, T.: Alpha Galois Lattices. ICDM, 555–558 (2004)
16. Wille, R.: Restructuring lattice theory: an approach based on hierarchies of concepts. In I. Rival (Ed.) Ordered Sets. Reidel, 445–470 (1982)
17. Wille, R.: Lattices in data analysis: how to draw them with a computer. In I. Rival (Ed.) Algorithms and Order. Kluwer, 33–58 (1989)
18. Wille, R.: Tensorial decomposition of concept lattices. Order 2, 81–95 (1985)
19. Wille, R.: Subdirect product construction of concept lattices. Discrete Mathematics 63, 305–313 (1987)

Order-embedded Complete Lattices

Bernhard Ganter

TU Dresden, Germany

Abstract. We study complete lattices which are contained in other complete lattices as suborders, but not necessarily as subsemilattices. We develop a representation of such lattices by means of implications, and show how to navigate them using a modification of the standard Next closure algorithm. Our approach is inspired by early work of Shmuely [8] and Crapo [1].

Keywords. Complete lattice, implication, fixed point.

1 Introduction

The interest in *concept lattices* [5] has stimulated the creation of algorithms for generating lattices, and the availability of fast algorithms may conversely have contributed to the popularity of concept lattices. Moreover, concept lattices have easy representations either by a binary relation or by a set of implications, both of which can conveniently be used as input for the algorithms.

Although all complete lattices are isomorphic to concept lattices, they sometimes come in a form for which the above mentioned algorithms are not easy to apply. There are, for example, many families of sets which form complete lattices when ordered by the subset relation $\subseteq$, but are neither closure nor kernel systems. We provide an "implicational" representation for such lattices and modify one of the standard algorithms accordingly.

Throughout the paper, $(L, \leq)$ will be some abstract complete lattice. The reader may assume, without much loss of generality, that $(L, \leq)$ is a powerset lattice $(\mathfrak{P}(M), \subseteq)$. We use the abstract setting because it is more transparent.

2 Monotone Functions

Definition 1 Let $(L, \leq)$ be a complete lattice. A function $\varphi : L \to L$ is **monotone**[1] if $x \leq y$ always implies $\varphi(x) \leq \varphi(y)$. A monotone function is called
idempotent if $\varphi(x) = \varphi(\varphi(x))$ for all $x \in L$,
extensive if $x \leq \varphi(x)$ for all $x \in L$,
contractive[2] if $x \geq \varphi(x)$ for all $x \in L$,
tensive if $\varphi(x) = \varphi(x \wedge \varphi(x))$ for all $x \in L$,
increasing[3] if $\varphi(x) \leq \varphi(\varphi(x))$ for all $x \in L$, and
decreasing if $\varphi(x) \geq \varphi(\varphi(x))$ for all $x \in L$. $\Diamond$

[1] Synonyms are **order-preserving** and **isotone**.

[2] A synonym is **intensive**.

[3] Following Shmuely [8]. Tarski [9] uses "increasing" in the sense of "order-preserving".

© Dmitry I. Ignatov, Lhouari Nourine (Eds.): CLA 2018, pp. 153–165,
ISBN 978–80–244–5328–6, Department of Computer Science, Palacký University
Olomouc, 2018.

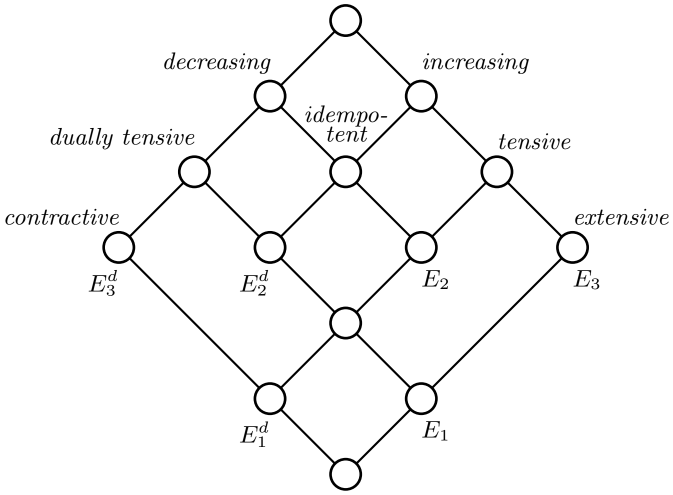


Fig. 1. The result of an attribute exploration [4] for monotone functions, see Proposition 1.

Proposition 1. *Figure 1 shows the logical hierarchy of the properties given in Definition 1. In particular, if $\varphi : L \to L$ is monotone, then the following statements hold (as well as their duals):*

1. *If φ is extensive, then φ is tensive.*
2. *If φ is tensive, then φ is increasing.*
3. *φ is idempotent iff φ is both increasing and decreasing.*
4. *If φ is idempotent and extensive, then φ is dually tensive.*

Moreover, there are examples of monotone functions falsifying other implications, as indicated in the diagram.

Proof. 1) If $x \leq \varphi(x)$, then $x \wedge \varphi(x) = x$ and thus $\varphi(x \wedge \varphi(x)) = \varphi(x)$. 2) From $x \wedge \varphi(x) \leq \varphi(x)$ we infer $\varphi(x) = \varphi(x \wedge \varphi(x)) \leq \varphi(\varphi(x))$. 3) is obvious. 4) If $x \leq \varphi(x)$ then $\varphi(x \vee \varphi(x)) = \varphi(\varphi(x)) = \varphi(x)$.

For the separating examples we use functions of the form $\varphi_{\mathcal{L}}$, to be defined in Proposition 6. $(L, \leq)$ is the powerset lattice of $\{a, b, c\}$ for E_1 and E_2 and of $\{a, b\}$ for E_3. E_1: $\mathcal{L} = \{\{a\} \to \{b, c\}, \{b\} \to \{c\}, \{c\} \to \{b\}\}$, E_2: $\mathcal{L} = \{\{a\} \to \{b\}, \{a, b, c\} \to \{a, b, c\}\}$ (see Example 1 below), E_3: $\mathcal{L} = \{\emptyset \to \{a\}, \{a\} \to \{b\}, \{b\} \to \{b\}\}$. E_1^d, E_2^d, E_3^d are dual to E_1, E_2, E_3. □

Definition 2 If $\varphi : L \to L$ is a mapping and $x \in L$, then we say that x is a **fixed** point of φ, iff $\varphi(x) = x$, and that x is a **closed** point of φ, iff $\varphi(x) \leq x$. ◊

Proposition 2. *Every fixed point is closed. If φ is monotone and increasing, and x is closed, then $\varphi(x)$ is fixed.*

Proof. The first statement is obvious. Suppose that x is closed, i.e., that $x \geq \varphi(x)$. Then $\varphi(x) \geq \varphi(\varphi(x)) \geq \varphi(x)$, since φ is monotone and increasing. We conclude that $\varphi(x) = \varphi(\varphi(x))$ and thus $\varphi(x)$ is fixed.

The proposition may suggest a pairing between fixed and closed elements. But note for example that when φ is the function which maps everything to the least element of $(L, \leq)$, then *every* element of $(L, \leq)$ is closed, but *only the least* element is fixed.

A function that is both idempotent and monotone is called a **closure operator** on $(L, \leq)$ if it is extensive, and is a **kernel operator** if contractive. The set of fixed points of a closure operator is called a **closure system**. It is well known that the closure systems are precisely the $\bigwedge$-subsemilattices. Each complete meet-subsemilattice of a complete lattice is itself a complete lattice, because the join operation can be expressed in terms of the meet operation: the join of a subset S equals the meet of all upper bounds of S. However, this join operation usually is not identical with the join in the original complete lattice. The meet-subsemilattice therefore is a complete lattice, but *not* a complete sublattice in general. In a closure system of sets, for example, the join of two elements is usually not given by their set union, but by the closure of this union. Thus a closure system, ordered by set inclusion, is a complete lattice, but not necessarily a sublattice.

The fixed points of a kernel operator are closed under arbitrary joins and thus form a $\bigvee$-subsemilattice, also called a **kernel system**. Again we get the second operation from the first, so that each kernel system also is a complete lattice.

This shows that closure systems are not the only subsets yielding order-embedded complete lattices. In fact, the following is well known[4]:

Lemma 1. *A subset of a complete lattice $(L, \leq)$, with the induced order, is a complete lattice if and only if it is the image of a monotone and idempotent function $\varphi : L \to L$.*

Proof. Suppose that $\mathcal{F} = \{\varphi(x) \mid x \in L\}$ for some monotone and idempotent function $\varphi : L \to L$. We claim that for any subfamily $\mathcal{S} \subseteq \mathcal{F}$ the element $\varphi(\bigwedge \mathcal{S})$ is the infimum of $\mathcal{S}$ in $\mathcal{F}$. Clearly $\bigwedge \mathcal{S} \leq s$ holds for every $s \in \mathcal{S}$. Since φ is monotone, we get that $\varphi(\bigwedge \mathcal{S}) \leq \varphi(s) = s$ for all $s \in \mathcal{S}$, which shows that $\varphi(\bigwedge \mathcal{S})$ is a lower bound of $\mathcal{S}$. But any lower bound b of $\mathcal{S}$ must satisfy $b \leq s$ for all $s \in \mathcal{S}$ and therefore $b \leq \bigwedge \mathcal{S}$. If $b \in \mathcal{F}$, then $b = \varphi(b) \leq \varphi(\bigwedge \mathcal{S})$, as desired.

For the converse suppose that $\mathcal{F} \subseteq L$ is a complete lattice and define a function $\varphi : L \to L$ by $\varphi(x) := \sup_{\mathcal{F}}\{f \in \mathcal{F} \mid f \leq x\}$ (where $\sup_{\mathcal{F}}$ denotes the supremum in $\mathcal{F}$). This function is clearly idempotent and monotone, and its image is $\mathcal{F}$. □

Lemma 1 adds a kind of converse to the celebrated Knaster-Tarski theorem [6,9], which states that the set of fixed points of any monotone function on a complete lattice is itself a complete lattice:

[4] Crapo [1] cites Duffus and Rival [2], while Shmuely [8] cites older notes by Crapo.

Corollary 1. *A subset $\mathcal{F} \subseteq L$ of a complete lattice $(L, \leq)$, with the induced order, is a complete lattice if and only if $\mathcal{F}$ is the set of fixed points of some monotone function.*

Two details from the proof of Lemma 1 will be used later. We list them as separate propositions:

Proposition 3. *Let φ be an idempotent and monotone function on a complete lattice $(L, \leq)$, and let $\bigvee$, $\bigwedge$ denote the supremum and infimum operation of $(L, \leq)$, respectively.*

In the complete lattice of fixed points of φ, the supremum and infimum of a set $\mathcal{S}$ are given by

$$\varphi(\bigvee \mathcal{S}) \qquad \text{and} \qquad \varphi(\bigwedge \mathcal{S}).$$

The second part of the proof of Lemma 1 is stronger than necessary: the function which was used is not only monotone and idempotent, but has an additional property:

Proposition 4. *The function which was used in the proof of Lemma 1,*

$$x \mapsto \varphi(x) := \sup_{\mathcal{F}}\{f \in \mathcal{F} \mid f \leq x\},$$

is tensive.

Proof. If $f \leq x$ and $f \in \mathcal{F}$, then $f \leq \varphi(x)$ and so $f \leq x \wedge \varphi(x)$. Thus

$$\{f \in \mathcal{F} \mid f \leq x\} \subseteq \{f \in \mathcal{F} \mid f \leq x \wedge \varphi(x)\},$$

which implies that $\varphi(x) \leq \varphi(x \wedge \varphi(x))$. Since φ is monotone, we conclude equality. □

A simple consequence of the Knaster-Tarski result which we will use is

Proposition 5. *If $(L, \leq)$ is a complete lattice, $\varphi : L \to L$ is monotone, and $x \in L$ is an element for which $x \leq \varphi(x)$, then there is a least fixed point of φ that is greater or equal to x.*

Proof. Note that since φ is monotone, the set $\uparrow x := \{y \in L \mid x \leq y\}$ is mapped into itself by φ: when $y \geq x$, then $\varphi(y) \geq \varphi(x) \geq x$. But $\uparrow x$ is a complete lattice as well, to which the Knaster-Tarski result can be applied. So there is a least fixed point of φ in $\uparrow x$. □

Lemma 2. *If $\varphi : L \to L$ is monotone and increasing, then for each $x \in L$ there is a least closed element $\overline{\varphi}(x) \geq x$, and there is a least fixed element $\widehat{\varphi}(x) \geq \varphi(x)$. If φ is tensive, then so is $\widehat{\varphi}$.*

Proof. For the first claim define a function $\rho(x) := x \vee \varphi(x)$. Note that the fixed points of ρ are precisely the closed points of φ. Clearly ρ is monotone and extensive, so by Proposition 5 there is a least fixed point y of ρ which is greater or equal to x.

The second claim follows again from Proposition 5, assuming that the function φ is increasing.

Finally, assume that φ is tensive. By definition, $\widehat{\varphi}(x \wedge \varphi(x))$ is the least fixed point of φ greater or equal to $\varphi(x \wedge \varphi(x))$. But when φ is tensive, the latter equals $\varphi(x)$, and therefore $\widehat{\varphi}(x \wedge \varphi(x)) = \widehat{\varphi}(x)$. But since $\varphi(x) \leq \widehat{\varphi}(x)$, we get $\widehat{\varphi}(x) = \widehat{\varphi}(x \wedge \varphi(x)) \leq \widehat{\varphi}(x \wedge \widehat{\varphi}(x)) \leq \widehat{\varphi}(x)$, which concludes the proof. □

Note that the function $\overline{\varphi}$, defined in Lemma 2, is a closure operator, and that $\widehat{\varphi}$ has the same fixed points as φ.

Lemma 3. *If φ is monotone and increasing, then for all $x \in L$*

$$\widehat{\varphi}(x) = \overline{\varphi}(\varphi(x)).$$

Proof. $\widehat{\varphi}(x)$ is fixed and therefore closed, and contains $\varphi(x)$, thus $\widehat{\varphi}(x) \geq \overline{\varphi}(\varphi(x))$. It remains to show that $\widehat{\varphi}(x) \leq \overline{\varphi}(\varphi(x))$. Proposition 2 yields that $\varphi(\overline{\varphi}(\widehat{\varphi}(x)))$ is fixed and less or equal to $\overline{\varphi}(\varphi(x))$. The proof is complete if we show that this fixed element contains $\varphi(x)$, because that forces it to be equal to $\widehat{\varphi}(x)$ (which is the *least* such fixed point). But from $\varphi(x) \leq \overline{\varphi}(\varphi(x))$ and the fact that φ is increasing and monotone we conclude that $\varphi(x) \leq \varphi(\varphi(x)) \leq \varphi(\overline{\varphi}(\varphi(x)))$. □

3 Implications

There is a simple way of constructing such monotone and increasing functions without reference to an embedded lattice. It relies on *implications*. An **implication** over L is just an ordered pair[5] of elements $x, y \in L$, denoted $x \to y$. We say that a lattice element z **respects** an implication $x \to y$ if $x \not\leq z$ or $y \leq z$.

Proposition 6. *Let $\mathcal{L}$ be a set of implications over L. The function $\varphi_{\mathcal{L}} : L \to L$, defined as*

$$\varphi_{\mathcal{L}}(x) := \bigvee \{a \vee b \mid a \leq x, a \to b \in \mathcal{L}\},$$

is monotone and tensive. Conversely, if $\varphi : L \to L$ is monotone and tensive, then $\varphi = \varphi_{\mathcal{L}}$ for

$$\mathcal{L} = \{x \wedge \varphi(x) \to \varphi(x) \mid x \in L\}.$$

Proof. When $x \leq y$, then $\{a \to b \in \mathcal{L} \mid a \leq x\} \subseteq \{a \to b \in \mathcal{L} \mid a \leq y\}$, and thus $\varphi_{\mathcal{L}}(x) \leq \varphi_{\mathcal{L}}(y)$. So $\varphi_{\mathcal{L}}$ is monotone. For the other claim, note that if $a \to b \in \mathcal{L}$ and $a \leq x$, then $a \leq \varphi_{\mathcal{L}}(x)$ and thus $a \leq x \wedge \varphi_{\mathcal{L}}(x)$. It follows that $\varphi_{\mathcal{L}}(x \wedge \varphi_{\mathcal{L}}(x)) \geq \varphi_{\mathcal{L}}(x)$. Monotonicity of $\varphi_{\mathcal{L}}$ yields equality and concludes the proof that $\varphi_{\mathcal{L}}$ is tensive.

For the converse we claim that we have $\varphi_{\mathcal{L}}(y) = \varphi(y)$ for all $y \in L$. Since $y \wedge \varphi(y) \leq y$ and $y \wedge \varphi(y) \to \varphi(y) \in \mathcal{L}$, we get that $\varphi_{\mathcal{L}}(y) \geq \varphi(y)$. If $x \wedge \varphi(x) \leq y$ holds for some x, then $\varphi(x) = \varphi(x \wedge \varphi(x)) \leq \varphi(y)$ (since φ is monotone and tensive), and therefore $\varphi_{\mathcal{L}}(y) \leq \varphi(y)$. This proves $\varphi_{\mathcal{L}} = \varphi$. □

[5] The notion abstracts that of an *attribute implication* in Formal Concept Analysis. Note that in our approach implication sets are *not* assumed to be closed under the Armstrong rules.

Example 1. The separating example E_2 of Figure 1 was defined in the proof of Proposition 1 as the monotone function $\varphi_{\mathcal{L}}$ on the power set of $\{a, b, c\}$ given by the implication set

$$\mathcal{L} := \{\{a\} \to \{b\}, \{a, b, c\} \to \{a, b, c\}\}.$$

According to the definition in Proposition 6, this function has the following values:

x	$\emptyset$	$\{a\}$	$\{b\}$	$\{c\}$	$\{a,b\}$	$\{a,c\}$	$\{b,c\}$	$\{a,b,c\}$
$\varphi_{\mathcal{L}}(x)$	$\emptyset$	$\{a,b\}$	$\emptyset$	$\emptyset$	$\{a,b\}$	$\{a,b\}$	$\emptyset$	$\{a,b,c\}$

.

It is easy to check that the function is monotone, idempotent, and tensive. But it is not dually tensive, since $\varphi_{\mathcal{L}}(\{a, c\}) = \{a, b\} \neq \varphi_{\mathcal{L}}(\{a, c\} \cup \varphi_{\mathcal{L}}(\{a, c\})) = \varphi_{\mathcal{L}}(\{a, b, c\}) = \{a, b, c\}$.

Following Definition 2 we call an element $x \in L$ **fixed** under a set $\mathcal{L}$ of implications, if $x = \varphi_{\mathcal{L}}(x)$, and **closed**, if $x \geq \varphi_{\mathcal{L}}(x)$. It is easy to see that the latter is equivalent to the standard definition in Formal Concept Analysis, where an element is called *closed* under $\mathcal{L}$ when it respects all implications in $\mathcal{L}$. The corresponding closure operator is often denoted $x \mapsto \mathcal{L}(x)$. Here we write $\overline{\varphi}_{\mathcal{L}}$, as it is suggested by Lemma 2. The following is a corollary to that lemma.

Corollary 2. *Let $\mathcal{L}$ be a set of implications over L. Then the function $\widehat{\varphi}_{\mathcal{L}} : L \to L$, defined by*

$$\widehat{\varphi}_{\mathcal{L}}(x) \text{ is the least fixed point of } \varphi_{\mathcal{L}} \text{ greater or equal to } \varphi_{\mathcal{L}}(x),$$

is idempotent, monotone, and tensive. Moreover,

$$\widehat{\varphi}_{\mathcal{L}}(x) = \overline{\varphi}_{\mathcal{L}}(\varphi_{\mathcal{L}}(x)) \text{ for all } x \in L.$$

A very welcome consequence of this corollary is that $\widehat{\varphi}_{\mathcal{L}}$ can efficiently be computed, for example by the LinClosure algorithm (see Algorithm 15 in [4]).

It is actually possible to give a (kind of) explicit representation of $\widehat{\varphi}_{\mathcal{L}}$ in terms of $\varphi_{\mathcal{L}}$: If $(L, \leq)$ is finite, then

$$\widehat{\varphi}_{\mathcal{L}}(x) := \varphi_{\mathcal{L}}(x) \vee \varphi_{\mathcal{L}}(\varphi_{\mathcal{L}}(x)) \vee \varphi_{\mathcal{L}}(\varphi_{\mathcal{L}}(\varphi_{\mathcal{L}}(x))) \vee \ldots . \qquad (*)$$

Without the finiteness condition it may be necessary to apply $\varphi_{\mathcal{L}}$ "transfinitely often", as the following example shows.

Example 2. Let $L := \mathbb{N} \cup \{\infty_1, \infty_2\}$, where the integers are in the natural order, ∞_1 is greater than all integers and $\infty_1 < \infty_2$. Moreover, let

$$\mathcal{L} := \{0 \to 1,\ 1 \to 2,\ 2 \to 3, \ldots\} \cup \{\infty_1 \to \infty_2\}.$$

For $x := 0$ we get $\varphi_{\mathcal{L}}(x) = 1$, $\varphi_{\mathcal{L}}(\varphi_{\mathcal{L}}(x)) = 2$, and so on. Applying formula $(*)$ yields $\widehat{\varphi}_{\mathcal{L}}(0) = 1 \vee 2 \vee 3 \vee \ldots = \infty_1$. But this is no fixed point, since $\varphi_{\mathcal{L}}(\infty_1) = \infty_2$.

Example 3. Let $M := \{a, b, c, d\}$, $(L, \leq) := (\mathfrak{P}(M), \subseteq)$, and

$$\mathcal{L} := \{\{a\} \to \{a\}, \{b\} \to \{b\}, \{b, c, d\} \to \{b, c, d\}, \{a, b\} \to \{c\}\}.$$

Then, for example, $\widehat{\varphi}_{\mathcal{L}}(\{a, c, d\}) = \{a\}$ and $\widehat{\varphi}_{\mathcal{L}}(\{a, b, d\}) = \{a, b, c\}$. $\widehat{\varphi}_{\mathcal{L}}$ has six fixed points; $\emptyset, \{a\}, \{b\}, \{a, b, c\}, \{b, c, d\}$, and $\{a, b, c, d\}$. These six sets, ordered by $\subseteq$, form a complete lattice which is neither a $\bigcap$- nor a $\bigcup$-subsemilattice of the powerset lattice.

We claim that our construction based on implications is universal in the sense, that every embedded complete lattice is obtained. This is shown in the theorem below.

Theorem 1. *For every monotone function $\varphi : L \to L$ on a complete lattice $(L, \leq)$ there is a set $\mathcal{L}$ of implications over L such that φ and $\widehat{\varphi}_{\mathcal{L}}$ have the same fixed points.*

Proof. The set $\mathcal{F}$ of fixed points of any monotone function is, according to Knaster and Tarski (see Corollary 1), a complete lattice. For each such complete lattice $(\mathcal{F}, \leq)$ we therefore need to find a suitable set of implications. Let $\sup_L$ denote the supremum in $(L, \leq)$ and $\sup_{\mathcal{F}}$ denote the supremum in $(\mathcal{F}, \leq)$. We choose

$$\mathcal{L} := \{\sup_L \mathcal{S} \to \sup_{\mathcal{F}} \mathcal{S} \mid \mathcal{S} \subseteq \mathcal{F}\}$$

and prove that the fixed points of $\widehat{\varphi}_{\mathcal{L}}$ are precisely the elements of $\mathcal{F}$:

First suppose that $e \in \mathcal{F}$. The $\varphi_{\mathcal{L}}(e) = e$ because, first of all, $e \to e \in \mathcal{L}$, and secondly e respects all implications in $\mathcal{L}$: if $\sup_L \mathcal{S} \leq e$ for some set $\mathcal{S} \subseteq \mathcal{F}$, then $e \geq s$ for all $s \in \mathcal{S}$ and, since $(\mathcal{F}, \leq)$ is a complete lattice, $e \geq \sup_{\mathcal{F}} \mathcal{S}$. Conversely, if $e = \widehat{\varphi}_{\mathcal{L}}(e)$ is a fixed point of $\widehat{\varphi}_{\mathcal{L}}$, then let $\mathcal{S} := \{f \in \mathcal{F} \mid f \leq e\}$ be the set of all $\mathcal{F}$-elements below e. Since e respects the implication $\sup_L \mathcal{S} \to \sup_{\mathcal{F}} \mathcal{S}$, we get $\sup_{\mathcal{F}} \mathcal{S} \leq e$. But whenever the premise of an implication in $\mathcal{L}$ is below e, it must be below $\sup_L \mathcal{S}$. Therefore $e = \varphi_{\mathcal{L}}(e) = \sup_{\mathcal{F}} \mathcal{S}$ and thus $e \in \mathcal{F}$. □

As an immediate consequence of Theorem 1 we get

Corollary 3. *The subsets of a complete lattice which are, with the induced order, complete lattices themselves, are precisely the sets of fixed elements under some set of implications.*

$\widehat{\varphi}_{\mathcal{L}}$ is usually not extensive, while the closure operator $\overline{\varphi}_{\mathcal{L}}$ is. That condition can easily be achieved by including $\{x \to x \mid x \in L\}$ into the list of implications (it actually suffices to do this for a join-dense set of elements), so all closure operators are of the form $\widehat{\varphi}_{\mathcal{L}}$ for a suitable set $\mathcal{L}$.

But *kernel operators* can be represented as well: If $\mathcal{F}$ is a kernel system, then $\widehat{\varphi}_{\mathcal{L}}$ is the corresponding kernel operator when $\mathcal{L} := \{f \to f \mid f \in \mathcal{F}\}$. More generally, if $\mathcal{F}$ is an arbitrary family of sets then the so defined function $\widehat{\varphi}_{\mathcal{L}}$ is the kernel operator for the kernel system generated by $\mathcal{F}$.

Is it possible to find, for a given function, a suitable implication set without reference to the embedded lattice of fixed points? The next proposition gives an answer. However, we shall learn from Example 4 that this is not always practical.

Proposition 7. *If $\varphi : L \to L$ is monotone, idempotent, and tensive, then the set*

$$\mathcal{L} := \{x \wedge \varphi(x) \to \varphi(x) \mid x \in L\}$$

is such that $\widehat{\varphi}_{\mathcal{L}} = \varphi$.

Proof. From Proposition 6 we get that $\varphi = \varphi_{\mathcal{L}}$. But when φ is idempotent, then $\varphi(x) \to \varphi(x) \in \mathcal{L}$, which makes $\varphi(x)$ a fixed point of $\varphi_{\mathcal{L}}$, and we conclude $\varphi = \varphi_{\mathcal{L}} = \widehat{\varphi}_{\mathcal{L}}$. □

To summarize: Every embedded complete lattice is the image of some function which is monotone, idempotent and tensive (Proposition 4). These are precisely the functions which can be described by means of implications as in Corollary 2. Implications can easily be found for any given such function (Proposition 7).

4 The Next Fixed Point Algorithm

Many years ago the author suggested a simple algorithm [3] for finding all closed sets of a given closure operator φ on a (finite, linearly ordered) set G. One starts with the closure $A := \varphi(\emptyset)$ of the empty set and then repeats the procedure shown in Figure 2, using the output of each application as the input of the next one, until it returns $\perp$. The algorithm is extremely useful for browsing and navigating in closure systems. And since it is so simple, many variations and generalizations have been invented, see [4].

```
for all $g \in G$ in reverse order do
    if $g \in A$ then $A := A \setminus \{g\}$
    else
        $B := \varphi(A \cup \{g\})$
        if $g$ is the smallest element of $B \setminus A$ then return B
return $\perp$.
```

Fig. 2. The Next closure algorithm, from [4]

It is easy to generalize the algorithm to closure operators on complete lattices, not only powerset lattices. It therefore seems natural to ask if a modification of Next closure can be used for generating all images of any given idempotent, monotone, and tensive function. Unfortunately, the answer is “no”, unless some additional information is provided. Our pessimism is prompted by the following example:

Example 4. Let $A \subseteq L$ be an antichain in a complete lattice $(L, \leq)$, let 0_L and 1_L be the least and the greatest element of $(L, \leq)$, and let f be an element of

A. The function

$$\varphi(x) := \begin{cases} f & \text{if } x = f \\ 1_L & \text{if } a < x \text{ for some } a \in A \\ 0_L & \text{else} \end{cases}$$

is idempotent, monotone, and tensive.

In this example it is tedious to determine the fixed points by repeated invocation of φ. Since the number of antichains may be exponential[6] in the size of L, but nevertheless may be large on average, it seems difficult to find an algorithm which determines the fixed point f reasonably fast. Stronger assumptions are needed.

Proposition 8. *Let $(L, \leq)$ be a complete lattice, let $\varphi : L \to L$ be a monotone and idempotent function and let $G \subseteq L$ be a finite set that is join-dense in the complete lattice formed by the images of φ. Endow G with an arbitrary linear order.*

Compute a sequence of sets, starting with $A := \{g \in G \mid g \leq \varphi(\emptyset)\}$, and then repeatedly invoking the algorithm in Figure 3, always using the previous output as the next input, until $\perp$ is reached. For each set B in this sequence, $\varphi(\bigvee B)$ is a fixed point of φ, and all fixed points occur exactly once.

```
for all $g \in G$ in reverse order do
    if $g \in A$ then $A := A \setminus \{g\}$
    else
        $B := \{h \in G \mid h \leq \varphi(\bigvee(A \cup \{g\}))\}$
        if $g$ is the smallest element of $B \setminus A$ then return B
return $\perp$.
```

Fig. 3. The Next fixed point algorithm

Proof. Each fixed point f of φ is uniquely determined by its projection

$$\Pi(f) := \{g \in G \mid g \leq f\}$$

to the join-dense set G, because it can be obtained as the join of these elements: $f = \varphi(\bigvee(\Pi(f)))$ (recall from Proposition 3 that $S \mapsto \varphi(\bigvee S)$ is the join operation in the fixed point lattice). These projection sets form a closure system on G, for which $F \mapsto \Pi(\varphi(\bigvee F))$ is the closure operator. □

[6] For example, the Dedekind numbers in case that L is a powerset lattice.

	1	2	3	4
a	×	×		
b		×	×	×
c			×	
d				×

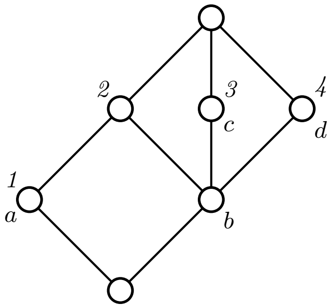


Fig. 4. A formal context and its concept lattice. Of its 66 embedded complete lattices, 20 are complete sublattices.

Example 5. We illustrate our findings by calculating the closed relations of the formal context in Figure 4. Closed relations are subrelations with the property that every formal concept of the subrelation is a formal concept of the original formal context [5]. They are in 1-1-correspondence with the complete sublattices of the concept lattice. The formal context has 20 closed relations, 18 of which are shown in Figure 5. The two missing ones are the empty relation and the full incidence of the formal context itself. Note that these relations (including the trivial ones) are not closed under intersection nor under union. The union of relations R_1 and R_2 is not closed, nor is the intersection of R_3 and R_4 closed. However, when ordered by set inclusion $\subseteq$, these 20 relations form a complete lattice which is isomorphic to the lattice of all complete sublattices. This lattice of closed relations is contained in the lattice $(\mathfrak{P}(\{a,b,c,d\} \times \{1,2,3,4\}), \subseteq)$ of all relations between these two sets as a suborder, but not as a sublattice.

1

			×
			×

2

		×	
		×	

3

	×		
	×		

4

	×	×	×

5

	×	×	×
			×

6

	×	×	×
		×	

7

	×	×	×
		×	
			×

8

	×		
	×	×	×

9

	×		
	×	×	×
			×

10

	×		
	×	×	×
		×	

11

	×		
	×	×	×
		×	
			×

12

×	×		

13

×	×		
			×
			×

14

×	×		
		×	
		×	

15

×	×		
	×		

16

×	×		
	×	×	×

17

×	×		
	×	×	×
		×	

18

×	×		
	×	×	×
			×

Fig. 5. The non-trivial closed relations of the formal context in Figure 4

The closed relations R_1, R_2, R_3, R_4, and R_{12} are of the form $A \times B$ for some nontrivial formal concept (A, B) and represent the complete sublattice with exactly one nontrivial element. These are the join-irreducible closed relations. Together, they form a join-dense set. For reasons that become clear later we reverse the order and work with

$$G := \{R_{12} < R_4 < R_3 < R_2 < R_1\}.$$

The function φ will be given by eight implications, five of which are of the form $X \to X$. Three more are derived from the condition that a sublattice must be closed under join and meet. So $\varphi := \widehat{\varphi}_{\mathcal{L}}$ for

$$\begin{aligned}\mathcal{L} := &\{R_1 \to R_1, R_2 \to R_2, R_3 \to R_3, R_4 \to R_4, R_{12} \to R_{12}\}\\ \cup\ &\{R_1 \cup R_2 \to R_4, R_1 \cup R_3 \to R_4, R_2 \cup R_3 \to R_4\}.\end{aligned}$$

If Algorithm 3 is used for this operator φ and is started with the empty relation, it produces all closed relations in the order of Figure 5, terminating with the full incidence relation of the context in Figure 4.

We give one intermediate step of the algorithm in detail, namely the step from R_2 to R_3:

R_2 contains none of the other relations in G, so the Next fixed point algorithm is invoked with $A := \{R_2\}$. The largest element of G is R_1, which is not in A, so $B := \{h \in G \mid h \leq \varphi(\bigvee(A \cup \{R_1\}))\}$ must be computed. $A \cup \{R_1\} = \{R_1, R_2\}$, and the join $\bigvee$ is the union $\bigcup$ of relations. We obtain $\bigvee(A \cup \{R_1\}) = R_1 \cup R_2$, which is not a closed relation. But $\mathcal{L}$ contains three implications the premise of which is contained in $R_1 \cup R_2$, and we find that $\varphi_{\mathcal{L}}(R_1 \cup R_2) = \varphi(R_1 \cup R_2) = R_1 \cup R_2 \cup R_4 = R_7$ and, since R_7 contains no further elements of G, $B = \{R_1, R_2, R_4\}$. However, R_1 is *not* the smallest element of $B \setminus A$ (the smallest element is R_4), so this iteration step does not return a result. The next iteration has $A = \{R_1\}$ and $g = R_1$, so R_1 is simply removed from A. Then $A = \emptyset$ and $g = R_3$ result in $B = \{R_3\}$, which is returned as the next closed relation.

For this particular example, the set of all 20 sublattices of the lattice in Figure 5 is easily determined by hand. In general, a concept lattice can be much larger than its formal context. Working with the formal context then may be more efficient.

How to find a join-dense set, as it is required in Proposition 8 ? There is an easy answer when the function φ is given by implications.

Proposition 9. *Let $\varphi := \widehat{\varphi}_{\mathcal{L}}$ for some set $\mathcal{L}$ of implications. Then the set*

$$\{\varphi(a) \mid a \to b \in \mathcal{L}\}$$

is join-dense in the lattice of fixed points of φ.

Proof. Any fixed point of $\widehat{\varphi}_{\mathcal{L}}$ by definition also is a fixed point of $\varphi_{\mathcal{L}}$. So if $\widehat{\varphi}_{\mathcal{L}}(f) = f$ then

$$f = \varphi_{\mathcal{L}}(f) = \bigvee\{a \vee b \mid a \leq f, a \to b \in \mathcal{L}\}.$$

But since $a \vee b \leq \varphi(a)$ whenever $a \to b \in \mathcal{L}$, we get that $f = \bigvee\{\varphi(a) \mid a \to b \in \mathcal{L}, a \leq f\}$, which proves the claim. □

Example 6. The set $\mathcal{L}$ in Example 3 consists of four implications, and we get

$$\{\{a\},\{b\},\{b,c,d\},\{a,b,c\}\} = \{\widehat{\varphi}_{\mathcal{L}}(\{a\}), \widehat{\varphi}_{\mathcal{L}}(\{b\}), \widehat{\varphi}_{\mathcal{L}}(\{b,c,d\}), \widehat{\varphi}_{\mathcal{L}}(\{a,b\})\}$$

as a join-dense set according to Proposition 9. However, $\{a,b,c\}$ is not join-irreducible, because the supremum of $\{a\}$ and $\{b\}$ is, using Proposition 3,

$$\widehat{\varphi}_{\mathcal{L}}(\{a\} \vee \{b\}) = \widehat{\varphi}_{\mathcal{L}}(\{a\} \cup \{b\}) = \widehat{\varphi}_{\mathcal{L}}(\{a,b\}) = \{a,b,c\}.$$

5 Discussion

Apart from closure and kernel systems, there are many "lattices of sets", i.e., families of sets which form complete lattices, when ordered by set inclusion. More generally we have studied subsets of arbitrary complete lattices which, endowed with the induced order, are complete lattices themselves. We have shown that each such complete lattice can be described by a set of implications, in a way which is very similar to the standard one in Formal Concept Analysis. The NEXT CLOSURE algorithm can be tweaked to work with this representation, so that we were able to give an algorithm for generating such lattices.

The reader may wonder why we did not use the even more general operator

$$x \mapsto \bigvee\{b \mid a \leq x, a \to b \in \mathcal{L}\}, \qquad x \in L,$$

which also is monotone. But such operators are no longer tensive in general, not even increasing. Actually, it is easy to see that *every* monotone function φ can so be represented (choose $\mathcal{L} := \{a \to \varphi(a) \mid a \in L\}$). Such operators are more difficult to handle, and we see no possibility of using LINCLOSURE here. But the fixed point sets of such functions describe the same as we have treated with tensive functions: all embedded complete lattices.

Much more important is the question if embedded complete lattices have a natural and useful interpretation. The work of Shmuely [8] gives interesting hints. Her *$u-v$-connections* generalize Galois connections and are closely related to what we construct. One might hope that these can be derived from formal contexts with additional, meaningful structure.

Acknowledgements

The author thanks Bogdan Chornomaz for his helpful remarks, and the reviewers for their careful proofreading.

References

1. Crapo, H.: Ordered sets: retracts and connections. Journal of Pure and Applied Algebra **23**(1), 13–28 (1982)
2. Duffus, D., Rival, I.: Retracts of partially ordered sets. Journal of the Australian Mathematical Society **27**(4), 495–506 (1979)
3. Ganter, B.: Two basic algorithms in concept analysis. FB4–Preprint 831, TH Darmstadt (1984), reprinted in [7]
4. Ganter, B., Obiedkov, S.: Conceptual exploration. Springer (2016)
5. Ganter, B., Wille, R.: Formal Concept Analysis: Mathematical Foundations. Springer (1998)
6. Knaster, B.: Un théorème sur les fonctions d'ensembles. Ann. Soc. Polon. Math. **6**, 133–134 (1928)
7. Kwuida, L., Sertkaya, B. (eds.): Formal Concept Analysis. Proceedings ICFCA 2010, Lecture Notes in Computer Science, vol. 5986. Springer (2010)
8. Shmuely, Z.: Increasing and decreasing operators on complete lattices. Journal of Combinatorial Theory, Series A **21**(3), 369–383 (1976)
9. Tarski, A.: A lattice-theoretical fixpoint theorem and its applications. Pacific J. Math. **5**(2), 285–309 (1955)

The 8M Algorithm from Today's Perspective

Radim Belohlavek and Martin Trnecka

Palacky University, Olomouc, Czech Republic,
radim.belohlavek@acm.org, martin.trnecka@gmail.com

Abstract. 8M is an old but nowadays virtually unknown algorithm for Boolean matrix factorization. In this paper, we provide a detailed analysis of 8M. We demonstrate by experiments that even though the algorithm uses a limited insight into the decomposition problem, its performance is reasonably good even from today's perspective. We analyze all the steps involved in 8M, provide a first complete description of 8M, and the relationships of 8M to the main currently available factorization algorithms. It turns out that 8M involves certain interesting concepts, which are not exploited by the current algorithms. We discuss the prospect of these concepts and, furthermore, propose an enhancement of 8M which is based on the current understanding of Boolean matrix factorization and significantly improves the performance of the original 8M.

1 Introduction

1.1 The Goal of this Paper

In the past decade or so, considerable research has been devoted to Boolean matrix factorization (BMF, called also Boolean matrix decomposition). This research has resulted in various new methods of analysis and processing of Boolean data and has also contributed to our understanding of Boolean (binary, yes/no) data as regards foundational aspects. A vast majority of the respective research contributions has been devoted to the design of factorization algorithms, which is also the subject of our paper. To name some of the best-known algorithms (more detailed information about some of these algorithms is provided in the subsequent sections), let us recall Tiling [9], the nowadays classic Asso [13], GreConD [3], Hyper [19], PaNDa [11], GreEss [5], and various modifications of these algorithms and modifications of the factorization problems discussed in the above-mentioned papers, as well as in [4,10,12,14,16,18].

Interestingly, there exists an old BMF algorithm, namely the 8M algorithm, which is virtually unknown in the present research on BMF. This fact is remarkable particularly in view of our experimental evaluations which demonstrate that the 8M algorithm performs reasonably well even from today's perspective. We learned about this algorithm from Hana Řezanková who used it in her various works on comparison of various clustering and factorization methods; see e.g. the references in [2]. Even though the performance of 8M may be partially assessed from those works, the principles of 8M have never been discussed in

© Dmitry I. Ignatov, Lhouari Nourine (Eds.): CLA 2018, pp. 167–178,
ISBN 978–80–244–5328–6, Department of Computer Science, Palacký University
Olomouc, 2018.

the literature. The goal of this paper is threefold. First, we provide a complete description of the 8M algorithm, including its pseudo-code and the description of its principles from today's perspective. Second, we propose an improvement of the 8M algorithm, which turns out to improve its performance reasonably. Third, we utilize one of the principles of 8M to enhance the performance of two standard algorithms. Note at this point that we discussed the 8M algorithm in our yet unpublished paper [6] in which we were solely interested in one particular property of this algorithm which we exploited in [6]; the present description is complete and comprehensive compared to the one presented in [6].

1.2 Basic Notions

The set of all $n \times m$ Boolean matrices shall be denoted $\{0,1\}^{n\times m}$ and the particular matrices by I, J, and the like. An input matrix I shall primarily be interpreted as an object-attribute incidence matrix (hence the symbol I). That is, the entry I_{ij} corresponding to the row i and the column j is either 1 or 0, indicating that the object i does or does not have the attribute j, respectively. The ith row and jth column vector of I is denoted by $I_{i\_}$ and $I_{\_j}$, respectively. In BMF, one generally attempts to find for a given $I \in \{0,1\}^{n\times m}$ matrices $A \in \{0,1\}^{n\times k}$ and $B \in \{0,1\}^{k\times m}$ for which

$$I \text{ (approximately) equals } A \circ B, \tag{1}$$

where $\circ$ is the Boolean matrix product, i.e. $(A \circ B)_{ij} = \max_{l=1}^{k} \min(A_{il}, B_{lj})$. A decomposition of I into $A \circ B$ may be interpreted as a discovery of k factors that exactly or approximately explain the data: Interpreting I, A, and B as object-attribute, object-factor, and factor-attribute matrices, model (1) reads: The object i has the attribute j if and only if there exists factor l such that l applies to i and j is one of the particular manifestations of l. The least k for which an exact decomposition $I = A \circ B$ exists is called the *Boolean rank* (or Schein rank) of I. The approximate equality in (1) is assessed in BMF by means of the metric $E(\cdot,\cdot)$, defined for $C, D \in \{0,1\}^{n\times m}$ by

$$E(C,D) = \textstyle\sum_{i,j=1}^{m,n} |C_{ij} - D_{ij}|. \tag{2}$$

The following particular variants of the BMF problem, relevant to this paper, are considered in the literature.

- *Discrete Basis Problem* (DBP, [13]):
  Given $I \in \{0,1\}^{n\times m}$ and a positive integer k, find $A \in \{0,1\}^{n\times k}$ and $B \in \{0,1\}^{k\times m}$ that minimize $E(I, A \circ B)$.
- *Approximate Factorization Problem* (AFP, [3]):
  Given I and prescribed error $\varepsilon \geq 0$, find $A \in \{0,1\}^{n\times k}$ and $B \in \{0,1\}^{k\times m}$ with k as small as possible such that $E(I, A \circ B) \leq \varepsilon$.

These problems reflect two important views of BMF: DBP emphasizes the importance of the first few (presumably most important) factors; AFP emphasizes the need to account for (and thus to explain) a prescribed portion of data.

In general, the committed error $E(I, A \circ B)$ has two parts, namely

$$E(I, A \circ B) = E_u(I, A \circ B) + E_o(I, A \circ B), \tag{3}$$

where $E_u(I, A \circ B) = |\{\langle i, j\rangle \mid I_{ij} = 1 \text{ and } (A \circ B)_{ij} = 0\}|$ and $E_o(I, A \circ B) = |\{\langle i, j\rangle \mid I_{ij} = 0 \text{ and } (A \circ B)_{ij} = 1\}|$ are the so-called undercovering error and overcovering error, respectively, which view shall be used below.

2 8M Described

2.1 History of 8M

The 8M method is one of the many data analysis methods available in an old and widely used statistical software package known as BMDP. The acronym "BMDP" stands for "Bio-Medical Data Package" (some sources say "BioMeDical Package"). The package was developed primarily for biomedical applications since the 1960s at the University of California in Los Angeles (UCLA) under the leadership of W. J. Dixon.[1] BMDP was originally available for free, later through BMDP Statistical Software, Inc., and then by its subsidiary, Statistical Solutions Ltd. As of 2017, BMDP is no longer available.[2]

BMDP and its methods are described in several editions of manuals, starting with a 1961 manual of BMD, a direct predecessor of BMDP. In our description of 8M, we use the 1992 edition [7], which accompanies release 7 of BMDP. There, 8M is described in chapter "Boolean factor analysis" on pp. 933–945, written by M. R. Mickey, L. Engelman, and P. Mundle, and in appendix B.11 on pp. 1401–1403.

The 8M method has been added to BMDP in the late 1970s: It was not part of the 1979 manual but it is part along with other new methods in the next version, whose revised printing appeared in 1983. According to this edition, 8M is based on research done by the statistician M. Ray Mickey of the UCLA, was designed by Mickey with contributions from Laszlo Engelman, and was programmed by Peter Mundle and Engelman.[3]

2.2 Description of the Method

Even though the description of 8M in [7] is fairly detailed, certain parts are somewhat unclear, both as to the procedural details and the rationale of various steps. As to the procedural details, we therefore examined the step-by-step

[1] The package grew out from an older computer program BIMED, which was developed for biomedical applications, and was first called BMD. Since the implemented methods allowed a parameterized format, the letter "P" was added. Later, "P" was interpreted as standing for "Package."

[2] We crosschecked our implementation against the version of BMDP we purchased in 2015 from Statistical Solutions Ltd.

[3] The references of the BMDP manual include some papers by Mickey but none of them concerns 8M and Boolean factor analysis.

program behavior on various data to figure out the unclear parts until our own implementation yielded the same results as the software which we purchased from Statistical Solutions Ltd. As to the rationale, we provide our explanation of the particular steps of 8M below.

Basic Idea We first describe the basic idea of 8M. The algorithm takes as its input four parameters: an $n \times m$ Boolean matrix I (object-attribute matrix), a number k of desired factors, and two auxiliary parameters, a number *init* of initial factors, and a number *cost* used to refine the factors being computed. The desired output consists of $n \times k$ and $k \times m$ Boolean matrices A (object-factor matrix) and B (factor-attribute matrix).

The algorithm starts by computing *init* initial factors. Then the algorithm iteratively computes new factors until k desired factors are obtained. The way 8M computes the factors is very different from the current BMF algorithms in two respects. First is the very way of generating a new factor. Second is the fact that the previously generated factors are revisited and dropped. The corresponding procedures are described in detail below.

Even though 8M's revisiting of the previously generated factors is done in a straightforward manner, it represents an interesting property. Namely, while the undercovering and overcovering error, E_u and E_o, see (3), seem symmetric, they have a different role in the design of BMF algorithms: Due to the NP-hardness of the various versions of the decomposition problem [17], most of the current factorization algorithms are heuristic approximation algorithms computing the factors one-by-one until a satisfactory factorization is obtained. Now, having computed say k factors, the next computed factor may make the overall error E smaller but its overcover part E_o never decreases (hence the decrease is E is due to a decrease in E_u). Put another way, while committing the E_u error may be repaired by adding further factors, committing the E_o error will never be repaired by adding further factors and must thus be carefully considered. Revisiting and possibly dropping some of the previously generated factors is a natural procedure to cope with this problem as it makes it possible to repair the E_o error. From this perspective it is interesting to note that while the current algorithms producing general factorization, such as Asso or PaNDa, do not use any kind of revisiting, the old 8M already used this idea.

Detailed Description and Pseudocode of 8M *(algorithm 1)* To compute $n \times k$ and $k \times m$ Boolean matrices A and B form the given $n \times m$ Boolean matrix I, the prescribed number *init* of initial factors, the desired number k of factors, and the parameter *cost*, the algorithm 8M (algorithm 1) proceeds as follows. First, *init* initial factors are computed (l. 1) as explained below. Note at this point that by default, $init = k - 2$ but *init* is generally set by the user. The variable f storing the number of the currently computed factors is set accordingly (l. 2). The matrices A and B are then refined (l. 3) by the procedure RefineMatricesAB described below. The algorithm then enters a loop (l. 5–17) whose purpose is to add new factors and remove some of the previously generated ones until the desired number k of factors is reached for the second time or all 1s in I are covered

```
Algorithm 1: 8M
    Input: Boolean $n \times m$ matrix $I$, desired number of factors $k$, number $init$ of
           initial factors, number $cost$
    Output: Boolean matrices $A$ and $B$
 1  $B \leftarrow$ ComputeInitialFactors($init$); $A \leftarrow \mathbf{0}_{n\times init}$
 2  $f \leftarrow init$
 3  RefineMatricesAB($A, B, I, cost$)
 4  $kReached \leftarrow 0$
 5  while $kReached < 2$ or $I \leq A \circ B$ do
 6      foreach $\langle i, j\rangle$ do if $I_{ij} > (A \circ B)_{ij}$ then $\Delta^{+}_{ij} \leftarrow 1$ else $\Delta^{+}_{ij} \leftarrow 0$
 7
 8      add column $j$ of $\Delta^{+}$ with the largest count of 1s as new column to $A$
 9      add row of 0s as new row to $B$ and set entry $j$ of this row to 1
10      $f \leftarrow f + 1$
11      RefineMatricesAB($A, B, I, cost$)
12      if another two new factors were added then
13          remove column $A_{\_(f-2)}$ from $A$ and row $B_{(f-2)\_}$ from $B$
14          $f \leftarrow f - 1$
15          RefineMatricesAB($A, B, I, cost$)
16      end
17      if $f{=}k$ then $kReached \leftarrow kReached + 1$
18  end
19  return $A, B$
```

by $A \circ B$, i.e. $I_{ij} \leq (A \circ B)_{ij}$ for all i, j holds (l. 5). Whenever a factor is added or removed, A and B are refined. Adding and removing factors is performed according to the following scheme. One starts with $f = init$ factors, adds two factors so that $f + 2$ factors are obtained, then removes the factor generated two steps back, i.e. the fth factor, adds another two factors, removes a factor generated two steps back, and so on. Hence, starting with $init = 2$ factors, one successively obtains 2, 3, 4, 3, 4, 5, 4, 5, 6, 5, 6, 7, 6, 7, 8, etc. factors. One stops when the desired number k of factors is obtained the second time. For instance, if $k = 6$ one computes the sequence $2, 3, 4, 3, 4, 5, 4, 5, 6, 5, 6$ of factors and the last six factors are the final factors output by the algorithm (provided the algorithm does not stop due to the second condition in l. 5).

The initial factors are computed by ComputeInitialFactors (algorithm 2) as follows. First, an $m \times m$ matrix C is computed in which $C_{ij} = 1$ iff column i is included in column j in I (i.e. $I_{qi} \leq I_{qj}$ for each q). One then goes through the rows i of C, $i = 1, 2, \ldots$, and adds them as new rows of B until $init$ rows have been added: row i of C is added to B provided there exists j with $C_{ij} = 1$ such that no row previously added to B contains 1 at position j.

Initialization of the factors is a key step in 8M in that the quality of the computed factorization depends on it. Below we propose a new way to initialize. At this point, let us point out an interesting observation. Computing the association matrix in the Asso algorithm is a kind of initialization. In particular,

Algorithm 2: ComputeInitialFactors

```
   Input: $n \times m$ Boolean matrix $I$ and the number of initial factors $init$
   Output: $init \times m$ Boolean matrix $B$
1  $C \leftarrow m \times m$ Boolean matrix with all entries equal to 0
2  foreach $C_{ij}$ do
3      if $I_{\_i} \leq I_{\_j}$ and $|I_{\_i}| > 0$ then
4          $C_{ij} \leftarrow 1$
5      end
6  end
7  remove all duplicate and empty rows from $C$
8  $f \leftarrow 0$
9  foreach row $i \in 1, \ldots, m$ of matrix $C$ do
10     if row $C_{i\_}$ has entry $j$ for which $C_{ij} = 1$ and $C_{kj} = 0$ for all $k < i$ then
11         $f \leftarrow f + 1$
12         add row $C_{i\_}$ as a new row to $B$
13     end
14     if $f = init$ then
15         return $B$
16     end
17 end
```

the vectors of the association matrix serve as the candidate B-parts of factors. Now, it is easy to observe that to select the rows of the association matrix, Asso uses basically the same strategy as 8M, only more general. Where 8M tests inclusion of columns i and j (l. 3 of algorithm 2), Asso tests whether the degree of partial inclusion of column i in column j exceeds a user-specified threshold τ (or whether the confidence of the association rule $\{i\} \Rightarrow \{j\}$ exceeds τ in terms of Asso). Setting $\tau = 1$ would yield the same vectors in the association matrix of Asso as what 8M uses as the initial factors. Even though we do not explore this observation in this paper, is shall be explored further.

Algorithm 3: RefineMatricesAB

```
Input: Boolean matrices $A$, $B$, $I$, number $cost$
repeat
    RefineMatrixA($A, B, I, cost$)
    RefineMatrixB($A, B, I, cost$)
until loop executed 3 times or $A$ and $B$ did not change
```

Refining of A and B by RefineMatricesAB (algorithm 3) consists in performing a cycle until A and B do not change but at most three times, in which A is computed from I, B, and the parameter *cost* by a so-called Boolean regression described in RefineMatrixA (algorithm 4), followed by computing symmetrically B using RefineMatrixB. A new factor is computed in l. 6–8 of 8M by computing first the positive part Δ^+ of the discrepancy matrix $\Delta = I - A \circ B$, one adds to A as new column the column j of Δ^+ containing the largest number of 1s, and adds to B a row of 0s with 1 at position j. For space reasons, we do not describe the meaning of Boolean regression further here; it shall be described in an extended version of this paper.

Algorithm 4: RefineMatrixA

```
Input: Boolean matrices $A$, $B$, $I$ and $cost$
foreach row $i \in \{1, \ldots, n\}$ do
    $y \leftarrow I_{i\_}$; $Z \leftarrow B$; $A_{i\_} \leftarrow \mathbf{0}$
    repeat
        foreach factor $l \in 1, \ldots, f$ do
            $m_l \leftarrow \sum_{j=1}^{m} y_j \cdot Z_{lj} - cost \cdot \sum_{j=1}^{m} (1 - y_j) \cdot Z_{lj}$
        end
        select $p$ for which $m_p = \max_l m_l$
        if $m_p > 0$ then
            $A_{ip} \leftarrow 1$
            foreach $j \in \{1, \ldots, m\}$ do
                if $Z_{pj} = 1$ then
                    $Z_{\_j} \leftarrow \mathbf{0}$; $y_j \leftarrow 0$
                end
            end
        end
    until $m_p > 0$
end
```

3 Experimental Evaluation

3.1 Datasets and Algorithms

Our evaluation involves the real-world datasets Apj [8] (2044 × 1164, density 0.003), DNA [15] (4590 × 392, density 0.015), Emea [8] (3046 × 35, density 0.068), Chess [1] (size 3196 × 76, density 0.487), Firewall 1 [8] (365 × 709, 0.124), Firewall 2 [8] (325 × 590, 0.190), Mushroom [1] (8124 × 119, 0.193), and Paleo[4] (501 × 139, density 0.051) well known and commonly used in the literature on BMF. Note that size refers to the number of objects × number of attributes and that density is the percentage of the entries with 1 of the dataset. Moreover, we used two collections, $X1$ and $X2$, of synthetic datasets. Each collection includes 1000 randomly generated matrices obtained as Boolean products $A \circ B$ of 1000×40 and 40×500 matrices A and B which are randomly generated. The

[4] NOWpublic release 030717, available from `http://www.helsinki.fi/science/now/`.

average densities of datasets included in $X1$ is 0.15. In case of $X2$, the average densities are 0.2. An extended version shall contain more datasets but the present results are representative of the algorithms' behavior.

We used in the experimental evaluation the algorithms: Tiling, Asso, GreConD, Hyper, and PaNDa (see section 1).

3.2 Evaluating 8M and Its Improved Version 8M+

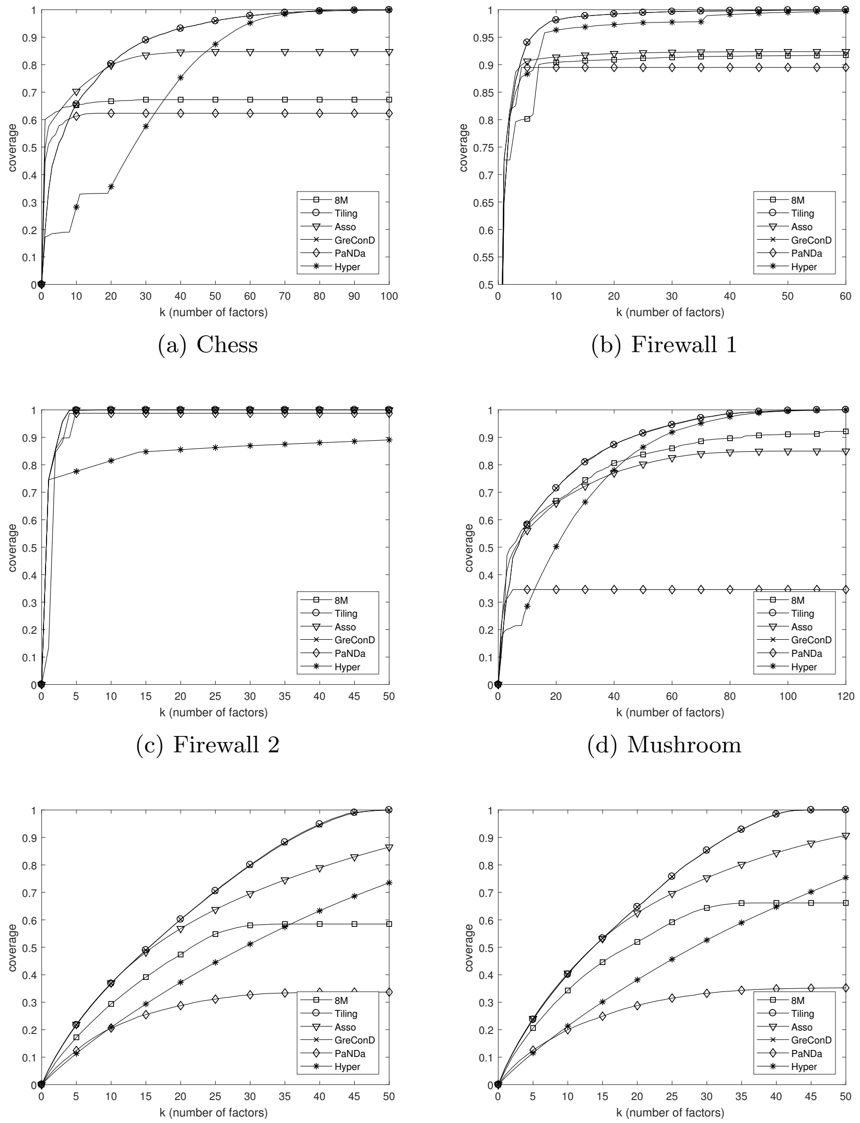
In our evaluation, we use the so-called coverage c of the input data I by the first l computed factors, i.e. the $n \times l$ and $l \times m$ matrices A and B, defined by $c(l) = 1 - E(I, A \circ B)/|I|$, in which $|I|$ is the number of 1s in I. For 8M we used the default recommendation $cost = 1$ and used various values for *init*.

Fig. 1 presents a comparison of the selected current BMF algorithms with the 8M method. The graphs depict the coverage $c(l)$ of the first l factors generated by the algorithms. One may observe that 8M compares fairly well with the current algorithms. It even outperforms PaNDa on all these datasets and on most of those we experimented with. On some data, 8M outperforms Asso and very often it outperforms Hyper in its coverage by the first few factors.

Fig. 2 presents a comparison of the basic 8M algorithm with its enhanced version denoted 8M+, which consists in a simple improvement of the initialization step of 8M. Namely, since the purpose of initialization in 8M is to obtain some reasonably good factors and since the initialization of 8M is rather simplistic, we exploited the very fast strategy of the GreConD algorithm to compute the first *init* factors. These have the additional advantage of committing no overcovering error. One may observe from the graphs that the improvement is significant. Moreover, taking into account Fig. 1, one can see that this improvement makes the new algorithm an interesting rival to the current algorithms.

3.3 Evaluating the Improvement of GreConD Inspired by 8M

It turns out that the idea of revisiting the previously generated factors may easily be implemented in one of the currently best BMF algorithms, GreConD, and yields a significant improvement as regards exact and almost exact factorizations. In our modification of GreConD, we revisit—every time a new factor is generated as in the original GreConD—the previously generated factors. If removal of a factor under consideration would result in an increase in the error E not larger than $p \times |I|$, where p is a parameter, we removed the factor. In Table 1, the columns represent the original GreConD and its modifications for $p = 0, 0.01, \ldots, 0.05$, the rows labeled "$k$" represent the number of factors obtained by the particular algorithm on the given dataset, and the row labeled "c" contains the coverage of the computed factorization. Thus, for instance, when factorizing the Mushroom data, the original GreConD needs 120 factors to obtain exact factorization. Our modification with $p = 0$ requires only 113 factors for exact factorization and only 61 factors for computing a highly accurate factorization, namely with coverage 0.951. Since such behavior is typical, we find it



(a) Chess

(b) Firewall 1

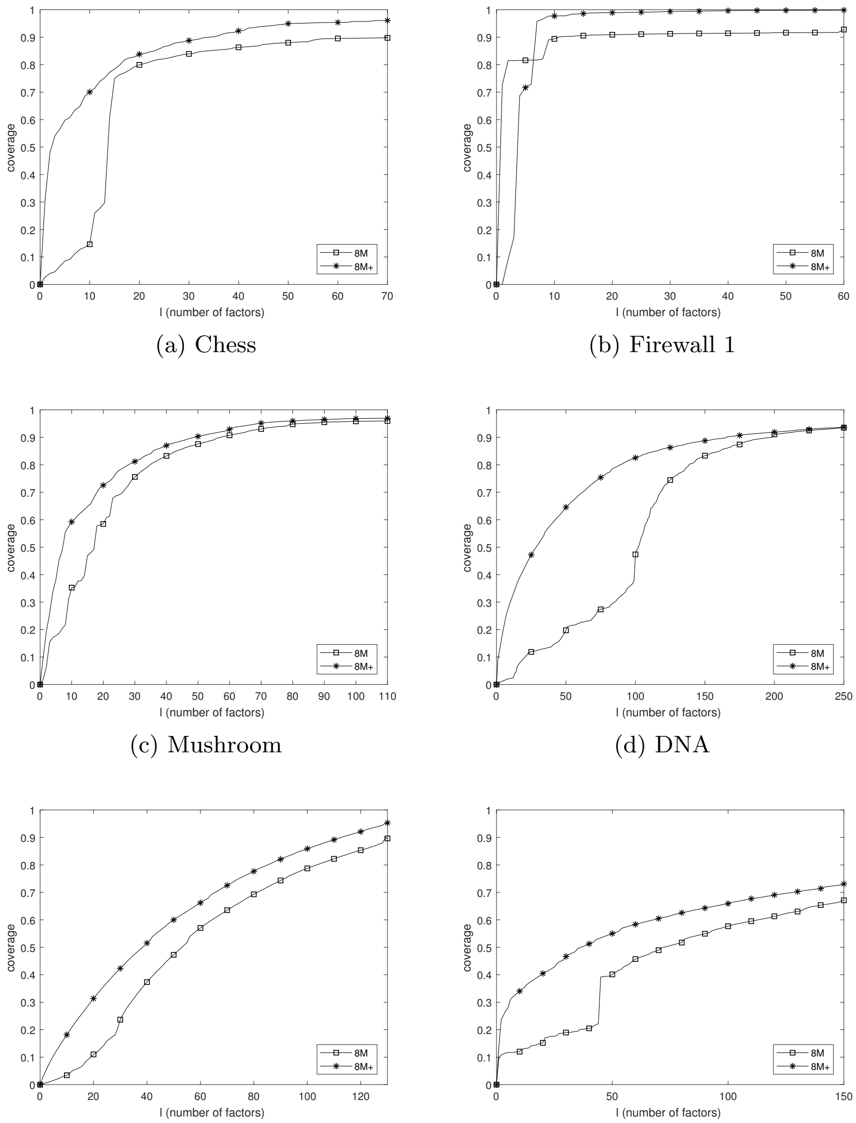
(c) Firewall 2

(d) Mushroom

(e) Set X1

(f) Set X2

Fig. 1: Coverage quality of the first l factors on real and synthetic data.



(a) Chess (b) Firewall 1

(c) Mushroom (d) DNA

(e) Paleo (f) Apj

Fig. 2: Coverage quality of the first l factors on real data: 8M vs. 8M+.

very interesting and find the idea of revisiting factors worth further exploration. Note that we did similar improvements with similar effects to Asso.

Table 1: Improvements to the GreConD algorithm

Dataset		orig.	0	0.01	0.02	0.03	0.04	0.05
Emea	k	42	34	29	26	25	24	23
	c	1.000	1.000	0.992	0.981	0.975	0.963	0.956
Chess	k	124	119	72	62	55	51	47
	c	1.000	1.000	0.991	0.981	0.970	0.962	0.952
Firewall 1	k	66	65	17	10	8	7	6
	c	1.000	1.000	0.990	0.981	0.972	0.964	0.953
Firewall 2	k	10	10	4	4	4	4	3
	c	1.000	1.000	0.998	0.998	0.998	0.998	0.958
Mushroom	k	120	113	81	73	69	65	61
	c	1.000	1.000	0.990	0.980	0.970	0.960	0.951

4 Conclusions

In addition to the fact that a description and detailed experimental evaluation of 8M were long due, we believe that the most interesting finding for future research is the property of 8M to revisit and possibly drop the previously computed factors. This idea is appealing particularly for algorithms performing general BMF, i.e. those committing overcovering error because, unlike the symmetric undercovering error, overcovering error can only increase if an algorithm does not revisit and modify the previously computed factors. Our straightforward implementation of this idea to GreConD and Asso yields an improvement which represents a promising sign of a usefulness of this idea, which hence needs to be further explored. Another topic worth further investigation is the regression procedure of 8M. While we described how it works, it is not yet properly understood why this procedure, which is analogous to statistical regression, actually works and delivers reasonable results.

Acknowledgment

R. Belohlavek acknolwedges support by the ECOP (Education for Competitiveness Operational Programme) project No. CZ.1.07/2.3.00/20.0059, which was co-financed by the European Social Fund and the state budget of the Czech Republic (the present research has been conducted in the follow-up period of this project), and by grant IGA 2018 of Palacký University Olomouc, No. IGA_PrF_2018_030.

References

1. Bache, K., Lichman, M.: UCI Machine Learning Repository [http://archive.ics.uci.edu/ml]. Irvine, CA: University of California, School of Information and Computer Science (2013)
2. Bartl, E., Belohlavek, R., Osicka, P., Řezanková, H.: Dimensionality Reduction in Boolean Data: Comparison of Four BMF Methods. CHDD 2012: 118–133. (2012)
3. Belohlavek, R., Vychodil, V.: Discovery of optimal factors in binary data via a novel method of matrix decomposition. J. Comput. Syst. Sci. **76**(1), 3–20 (2010)
4. Belohlavek R., Outrata J., Trnecka M.: Impact of Boolean factorization as preprocessing methods for classification of Boolean data. Ann. Math. Artif. Intell. **72**(1-2), 3-22 (2014)
5. Belohlavek, R., Trnecka, M.: From-below approximations in Boolean matrix factorization: Geometry and new algorithm. J. Comput. Syst. Sci. **81**(8), 1678–1697 (2015)
6. Belohlavek, R., Trnecka, M.: A new algorithm for Boolean matrix factorization which admits overcovering. Discrete Applied Mathematics (in press).
7. Dixon, W. J. (ed.): BMDP Statistical Software Manual. Berkeley, CA: University of California Press (1992)
8. Ene, A., Horne, W., Milosavljevic N., Rao, P., Schreiber, R., and Tarjan R. E.: Fast exact and heuristic methods for role minimization problems. In: SACMAT 2008, pp. 1–10. (2008)
9. Geerts, F., Goethals, B., Mielikäinen, T.: Tiling databases. In: Discovery Science 2004, pp. 278–289. (2004)
10. Lu, H., Vaidya, J., Atluri, V., Hong, Y.: Constraint-aware role mining via extended Boolean matrix decomposition. IEEE Trans. Dependable and Secure Comp. **9**(5), 655–669 (2012)
11. Lucchese, C., Orlando, S., Perego, R.: Mining top-K patterns from binary datasets in presence of noise. In: SIAM DM 2010, pp. 165–176. (2010)
12. Miettinen, P.: Sparse Boolean matrix factorizations. In: IEEE ICDM 2010, pp. 935–940. (2010)
13. Miettinen, P., Mielikäinen, T., Gionis, A., Das, G., Mannila, H.: The discrete basis problem. IEEE Trans. Knowledge and Data Eng. **20**(10), 1348–1362 (2008)
14. Miettinen, P., Vreeken J.: Model order selection for Boolean matrix factorization. In: ACM SIGKDD 2011, pp. 51–59. (2011)
15. Myllykangas, S. et al: 2006, DNA copy number amplification profiling of human neoplasms. Oncogene **25**(55), 7324–7332 (2006)
16. Outrata, J.: Boolean factor analysis for data preprocessing in machine learning. In: ICMLA 2010, pp. 899–902. (2010)
17. Stockmeyer, L., The set basis problem is NP-complete, Tech. Rep. RC5431, IBM, Yorktown Heights, NY, USA (1975)
18. Vaidya, J., Atluri, V., Guo, Q.: The role mining problem: finding a minimal descriptive set of roles. In: SACMAT 2007, pp. 175–184. (2016)
19. Xiang, Y., Jin, R., Fuhry, D., Dragan, F. F.: Summarizing transactional databases with overlapped hyperrectangles. Data Mining and Knowledge Discovery **23**, 215–251 (2011)

Can We Measure the Interpretability of Factors?

Martin Trnecka and Marketa Trneckova

Department of Computer Science
Palacky University Olomouc, Czech Republic
martin.trnecka@gmail.com, marketa.trneckova@gmail.com

Abstract. Decomposition of matrices over some finite scale received a considerable attention in data mining research. The methods that perform such decomposition can be viewed as an implementation of factor analysis. Surprisingly, the main motivation that is behind the factor analysis, the interpretation of the factors, is given only a very small amount of attention, or is completely neglected, in current research. In this paper, we are arguing that the interpretation of factors is an important part of matrix decomposition and we propose a novel measure, based on simple structure from factor analysis, enabling the intererability measurement. Furthermore, we present an experimental evaluation of selected decomposition algorithms via our metric.

1 Motivation

Decomposition of matrices over some finite scale—especially a case where the scale contains only two elements, namely zero and one, called the Boolean matrix decomposition—has become one of the standard methods in data mining with applications to many fields. In a broad sense, these methods may be considered as implementing the general idea of classical factor analysis introduced by psychologist Charles Spearman [16].

The motivation for the factor analysis comes from the psychology and the social sciences. The general aim is to simplify complex data. More precisely to describe original data via new more fundamental variables called factors.

Boolean matrix decomposition (BMF) and in general the decomposition of matrix over some finite scale, always reflects the ideas of factor analysis. This is not surprising, because BMF also comes from the psychology, where Boolean data often occur (see e.g. [7]). On the other hand some aspects of factor analysis are neglected in contemporary literature including matrix decomposition methods, namely the quality of factors. In factor analysis, the quality of factors, more precisely the interpretability of factors, is on the first place.

The current direction of research focuses on creating new algorithms and evaluating their quality in relation to the number of factors and the size of the input data covered. If an analysis of factor interpretability is contained, it is done by hand on a small number of datasets. The main reason is that the interpretation of factors is subjective and very tedious. In addition, contemporary literature lacks a uniform methodology or a metric to measure the interpretability of factors.

© Dmitry I. Ignatov, Lhouari Nourine (Eds.): CLA 2018, pp. 179–190,
ISBN 978–80–244–5328–6, Department of Computer Science, Palacký University
Olomouc, 2018.

The purpose of this paper is twofold. We argue that the interpretation of factors, which is often neglected, is important part of matrix decomposition method; and we propose a new measure, based on the simple structure from factor analysis, enabling an objective measurement of the interpretability of factors.

The main reason why we chose a simple structure as a criterion for the interpretability of factors is the historical interdependence of factor analysis and matrix decompositions. Simple structure is therefore the first choice, and for this reason this work should be seen as the first small step in the new field of research.

The rest of the paper is organized as follows. In the following Section 2 we provide a brief overview of current research involving matrix decomposition from the factor interpretation standpoint. Moreover, we describe in Section 2 basic approach to the factors interpretation in factor analysis. Then, in Section 3 basic notions, notation and formalization of the metric are presented. The metric is experimentally evaluated in Section 4. Section 5 draws a conclusion and future research direction.

2 Interpretation of Factors

2.1 Lost in the Flood of Algorithms

The current direction of matrix decomposition research focuses primarily on the production of new algorithms and improving existing ones. An overview of existing approaches and methods is beyond the scope of this paper (see e.g. [5,18] which provides comparison of the most commonly used algorithms). The factors interpretation has only a small amount of attention or it does not perform at all. Note that this is indeed a feature of contemporary research. Early works involving matrix decomposition usually contain a larger assessment—usually the analysis of the factors of one particular dataset—of factor interpretability.

One of the few exceptions is the work [4] that deals with the extensive detailed analysis of factors. However, this analysis is done manually. Works involving matrix decomposition do not contain any methodology or metrics to measure the interpretability of factors. This is very surprising, especially because these methods are inspired by classical factor analysis, where the interpretability of factors and its measurement is an elementary concept.

2.2 Good Factors Definition and Metric

In classical factor analysis, the question of whether the factor is good or bad is based on the law of parsimony, well known as Occam's razor, i.e. we should pick the simplest explanation of facts. A solution, which is selected via the parsimony law, is called *simple structure*.

In 1947 Thurstone proposed five simple criteria of simple structure in his work [17]. These can be seen as informal, vague and verbally described definition of good factors. Thurstone's criteria were as follows:

1. Each row of the rotated matrix should contain at least one zero.
2. In each factor the minimum number of zero loadings (see Section 3.2) should be the number of factors in the rotation.
3. For every pair of factors there should be variables with zero loadings on one and significant loadings on the other.
4. For every pair of factors a large portion of the loadings should be zero, at least in a matrix with a large number of factors.
5. For every pair of factors there should be only a few variables with significant loadings on both factors.

Two of these criteria, namely 3 and 5 are of overriding importance. Essentially, the criterion of simple structure is a factor matrix in which the factors each have a few high loadings.

Later in 1978, Cattell in [9], who continues in the Thurstone's work, argued that the simple structure factors are usually simple to interpret. There have been many attempts to formalize the simple structure (see e.g. [8]). The result of these attempts is an ad hoc formalization and a conclusion that there will never be a simple formula describing Thurstone's five criteria. Unfortunately, these approaches cannot be adopted in the case of decomposition of matrices over some finite scale, because they use a different calculus.

On the other hand, this kind of data can be handled using fuzzy logic. Moreover, in case of data over some finite scale, all Thurstone's criteria can be formalized via logical formulas. In the following section we will use the fuzzy logic to formalize Thurstone's criteria and create a metric allowing an objective analysis of factor interpretability.

3 Formalization of Metric

3.1 Basics from Fuzzy Logic

Fuzzy logic has been employed to handle the concept of partial truth, where the truth value may range between completely true and completely false. This approach has been proven to be useful in several areas and we refer to [1].

Let us consider a set L of truth values. We assume that this set is partially ordered (partial ordering is denoted by $\leq$), contains a least element 0 and a greatest element 1.

Let a and b are the truth degrees from L, then in L exists a truth value which is greater than both a and b. The least element that is greater or equal to both a and b is called *supremum* of a and b. Analogously, we can define *infimum* of a and b—the greatest element from L which is smaller or equal to both a and b. We define the *lower cone* of A by $\mathcal{L}(A) = \{a \in L | a \leq b \text{ for all } b \in A\}$ and the *upper cone* of A by $\mathcal{U}(A) = \{a \in L | b \leq a \text{ for all } b \in A\}$. If $\mathcal{L}(A)$ has a greatest element a, then a is called the *supremum* of A (denoted $\bigvee A$) and dually if $\mathcal{U}(A)$ has a least element a, then a is called the *infimum* of A (denoted $\bigwedge A$). In particular, we assume that the partial order $\leq$ makes L a complete lattice [12] (i.e., arbitrary infima $\bigwedge$ and suprema $\bigvee$ exist in L). This assumption is automatically satisfied if L is a finite chain (i.e. $a \leq b$ or $b \leq a$ for every $a, b \in L$), in which case $a \wedge b = \min(a, b)$ and $a \vee b = \max(a, b)$. We also need to define a logical conjunction operation (denoted by $\otimes$). We assume that $\otimes$ is commutative, associative, has 1 as its neutral element ($a \otimes 1 = a = 1 \otimes a$), and distributes over arbitrary suprema, i.e. $a \otimes (\bigvee_{j \in J} b_j) = \bigvee_{j \in J}(a \otimes b_j)$. This leads to if a and b are truth degrees of propositions p_1 and p_2, then $a \otimes b$ is the truth degree of proposition "p_1 and p_2".

Importantly, $\otimes$ induces another operation, $\rightarrow$, called the *residuum of* $\otimes$, which plays the role of the truth function of implication and is defined by

$$a \rightarrow b = \max\{c \in L \mid a \otimes c \leq b\}.$$

Residuum, which may be looked at as a kind of division, satisfies an important technical condition called adjointness:

$$a \otimes b \leq c \text{ iff } a \leq b \rightarrow c,$$

which is also utilized below. This leads to algebraic structures called *residuated lattices.*

3.2 Basic Notions of Matrix Decomposition

In general, matrix decomposition aims at whether data involving objects and their directly observable attributes may be explained by a smaller number of different, more fundamental attributes called factors. For example, whether performances of students (directly observable attributes) may be described by some treats of their intelligence (factors). Formally, the input data is represented by an $n \times m$ object–attribute matrix I and the "explanation" means a decomposition

$$I = A \circ B. \tag{1}$$

(exact or approximate) of I into a product $A \circ B$ of an $n \times k$ object–factor matrix A—called a score matrix in the factor analysis terminology—and a $k \times m$ factor–attribute matrix B—called a loading matrix in the factor analysis terminology. What kind of matrices (real, Boolean, or other) and what kind of product $\circ$ are involved determines the semantics of the factor model.

In this paper, we are mainly focused on the decomposition of matrices containing grades of certain scales L with the sup-$\otimes$ product. In particular, the matrix entry I_{ij} is a degree to which attribute j applies to object i. Similarly, A_{il} and B_{lj} are the degrees to which factor l applies to object i and the degree to which attribute j is a (one particular) manifestation of factor l. The case where the scale L contains only two degrees (0 and 1), is called the Boolean matrix decomposition.

Equation (1) has the following meaning. Object i has attribute j if and only if there exists factor l such that i has l (or, l applies to i) and j is one of the particular manifestations of l. The meaning can be described by the formula

$$(A \circ B)_{ij} = \bigvee_{l=1}^{k} A_{il} \otimes B_{lj},$$

Let us note, in the Boolean case ($L = \{0, 1\}$), the meaning of equation (1) may be described via formula

$$(A \circ B)_{ij} = \max_{l=1}^{k} \min(A_{il}, B_{lj}).$$

There exist two concrete variants of the decomposition problem. These two problems reflect two important views on matrix decomposition. The first one—the *discrete basis problem* (DBP) [14]—emphasizes the importance of the first k (presumably the most important) factors. The second one—the *approximate factorization problem* (AFP) [5]—emphasizes the need to account for (and thus to explain) a prescribed portion of data, which is specified by error ε.

The DBP and AFP problems are generally known in BMF, but both problems can be generalized to problems over some scale L. For this purpose we need to define closeness of matrices over L.

Let $s_L : L \times L \to [0, 1]$ be an appropriate function measuring closeness of degrees in L. For matrices $I, J \in L^{n\times m}$, put

$$s(I, J) = \frac{\sum_{i,j=1}^{n,m} s_L(I_{ij}, J_{ij})}{n \cdot m},$$

i.e. $s(I, J) \in [0, 1]$ is the normalized sum over all matrix entries of the closeness of the corresponding entries in I and J. In general, we require $s_L(a, b) = 1$ if and only if $a = b$, and $s_L(0, 1) = s_L(1, 0) = 0$, in which case $s(I, J) = 1$ if and only if $I = J$. We furthermore require that $a \leq b \leq c$ implies $s_L(a, c) \leq s_L(b, c)$. For the important case of L being a subchain of $[0, 1]$, s_L may be defined by

$$s_L(a, b) = a \leftrightarrow b,$$

where $a \leftrightarrow b = \min(a \to b, b \to a)$ is the so-called *biresiduum* (many-valued equivalence from a logical point of view) of a and b. Let us note that the closeness coincides with the notion coverage in several papers.

The generalization of the AFP and DBP to the general decomposition over scale L follows:

- *DBP(L)*: Given $I \in L^{n\times m}$ and a positive integer k, find $A \in L^{n\times k}$ and $B \in L^{k\times m}$ that maximize $s(I, A \circ B)$.
- *AFP(L)*: Given I and prescribed error $\varepsilon \in [0, 1]$, find $A \in L^{n\times k}$ and $B \in L^{k\times m}$ with k as small as possible such that $s(I, A \circ B) \geq \varepsilon$.

3.3 Interpretability Metric

We approach the formalization of Thurstone's five criteria according to the principles of mathematical fuzzy logic [1,12,13] as follows. We consider the factor model 1 and the Royce [15] definition of factor, namely "factor is a construct operationally defined by its factor loadings". In other words, factors are represented via attributes which are manifestation of them, i.e. factors are represented via rows of matrix B. This is very important aspect of our metric, because we can evaluate factors regardless of whether they do or or not contain noise—noise is a big issue in matrix decompositions, see e.g. [14].

The following formalization of the five criteria described in Section 2.2 utilizes operations over scale L. As far as the choice of the operations on L is concerned, we use the Łukasiewicz t-norm in the formalization, due to some of its intuitive properties. We describe each criterion via single logical formula with the truth degrees from L. The degree of fulfillment of each criterion is determined by the degree of fulfillment of a particular formula.

3.4 Formalization of Thurstone's Criteria

The first criterion. Each row contains at least one zero, i.e. for each factor there exist at least one attribute which is not particular manifestation of the factor. Formally, the first criterion can be described via formula $\forall i \exists j \neg B_{ij}$, i.e.

$$(\exists j \neg B_{1j}) \wedge (\exists j \neg B_{2j}) \wedge \cdots \wedge (\exists j \neg B_{kj}),$$

where $(\exists j \neg B_{ij}) = (\neg B_{i1}) \vee (\neg B_{i2}) \vee \cdots \vee (\neg B_{im})$.

The second criterion. In each factor, the minimum number of zero loadings should be the number of factors, i.e. in each factor, there is at least k attributes that are not manifestation of this factor. Formally,

$$\forall i \exists j_1 \exists j_2 \ldots \exists j_k (\neg B_{ij_1} \wedge \neg B_{ij_2} \wedge \ldots \neg B_{ij_k} \wedge j_1 \neq j_2 \neq \cdots \neq j_k).$$

The third criterion. For every pair of factors there should be variables with zero loadings on one and significant loadings on the other. Formally,

$$\forall i_1 \forall i_2 \exists j (B_{i_1 j} \wedge \neg B_{i_2 j}) \wedge (\neg B_{i_1 j} \wedge B_{i_2 j}).$$

The fourth criterion. For every pair of factors a large portion of loadings should be zero (at least in a matrix with large number of factors. We need to define what the "large portion" means, i.e. how many attributes do not manifest one or second (or both) factors. Let us denote "large portion" by lp and $B^{ij} = B_{i\_} \cup B_{j\_}$ ($B_{i\_}$ denotes row i of matrix B), than formally

$$\forall i_1 \forall i_2 \exists j_1 \exists j_2 \ldots \exists j_{lp} (\neg B^{i_1 i_2}_{j_1}) \wedge (\neg B^{i_1 i_2}_{j_2}) \wedge \cdots \wedge (\neg B^{i_1 i_2}_{j_{lp}}) \wedge j_1 \neq j_2 \neq \cdots \neq j_{lp}.$$

The fifth criterion. For every pair of factors there should be only a few attributes that manifest both factors. Similarly as in the previous case, we need to define a "few".

$$\forall i_1 \forall i_2 \exists j_1 \exists j_2 \ldots \exists j_{few} (B_{i_1 j_1} \wedge B_{i_2 j_1}) \wedge (B_{i_1 j_2} \wedge B_{i_2 j_2}) \wedge \ldots$$

$$\cdots \wedge (B_{i_1 j_{few}} \wedge B_{i_2 j_{few}}) \wedge j_1 \neq j_2 \neq \cdots \neq j_{few}.$$

The formalization via above presented logical formulas strictly says how a set of factors satisfies each criteria, but it does not take into account how many factors (pairs of factors) do not meet the criterion.

We can analogously define less strict measure which takes into account for how many factors (pairs of factors) each criterion holds, i.e instead of minimum value for each factor (pair of factors), we take mean of this values.

We denote the first variant of metric as the strict metric and the second variant as the partially strict metric in Section 4—which provides experimental evaluation of our metric.

4 Experimental Evaluation

The following section is devoted to the experimental evaluation of metrics described in Section 3. We compare three algorithms for the matrix decomposition problem, namely GreCond$_L$ [6], GreEss$_L$ [3] and Asso$_L$ [3]. The first two are based on formal concept analysis [11]. Let us note, these algorithms provide the decomposition of matrices over a finite scale L. All of them are inspired by the existing BMF algorithms.

4.1 Real-World Data

Since we are interested in the interpretability of factors, we perform experiments only on the real-world datasets—which, unlike synthetic data, are influenced by real factors. We used the following datasets.

`Dog breeds` dataset represents 151 dog breeds and their 13 attributes such as for example Playfulness, Protection ability, Affection or Ease of training. For detailed analysis see [3].

`Decathlon` extends the dataset from [6] to 28 athletes and their performance in 10 disciplines of decathlon. A detail analysis of this data can be found in [2].

`IPAQ` consists of international questionnaire data involving 4510 respondents answering 16 questions using a three-element scale regarding physical activity. The questions include those regarding their age, sex, body-mass-index (BMI), health, to what extent the person bicycles, walks, etc. For more detail see [3].

`Music` [3] consists of results of a study inquiring how people perceive speed of song depending on various song characteristics. The data consists of a 900 × 26 matrix over a six-element scale L, representing a questionnaire involving 30 participants who were presented 30 music samples.

`Rio` dataset [18] represents 87 × 31 matrix I obtained from `https://www.rio2016.com/en/medal-count` and consists of 87 countries that obtained any medal in one of 31 sport (such as Archery, Athletics, Badminton, Basketball, Boxing, ...) on Olympics games in Rio de Janeiro 2016. L contains four grades—1 means that country won at least one gold medal, $\frac{2}{3}$ at least silver medal, $\frac{1}{3}$ at least one bronze medal and 0 no medal in this sport. This dataset is very sparse in comparison with other presented datasets.

4.2 Assessment of the Interpretability Metric

Obtained results for each of five Thurstone's criteria and a total value of the simple structure are presented in Table 1 (strict measure) and Table 2 (partially strict measure). We provide the results for sets of factors with the values of closeness (column s) 0.75, 0.85, 0.9, 0.95 and 1 (which corresponds with the values of coverage 75%, 85%, 90%, 95% and 100% of the input data). Value NA means, that the particular algorithm can not obtain a prescribed coverage.

One may observe that the best results provides (in case of strict measure) GreEss$_L$ algorithm which outperforms GreConD$_L$ and Asso$_L$on `Breeds`, `Decathlon` and `IPAQ`. On the `Music` and `Rio` data, GreConD$_L$ produces slightly better results than GreEss$_L$. Asso$_L$ is not able to reach higher coverage and usually provides worse results, but on `Music` and `Rio` data it outperforms both GreConD$_L$ and GreEss$_L$.

We obtain similar results for partially strict measure. For this metric GreConD and GreEss produce higher values than in the case of strict metric. Additionally GreEss$_L$ outperforms GreConD$_L$ on `IPAQ` data and produces almost identical results on `Rio` data. Asso$_L$ produces very similar results as in the case of strict metric.

From Tables 1 and 2 it is obvious that the simple structure firstly fail on second criterion especially for high closeness (in both GreConD$_L$ and GreEss$_L$) since usually they need more factors than the number of attributes to achieve a prescribed coverage. Asso$_L$ is algorithm for solving DBP, usually it is not able to obtain full coverage of input data. On the other hand, first factors obtained by Asso$_L$ cover larger portion of data, so for example in `Rio` dataset we need only one factor to obtain coverage slightly higher than 90%. This is the reason why the total value of simple structure is equal to 1.

In [3] authors discuss problem of factor with values "around the middle". These factors are the reason why Asso$_L$ produces results which returns lower values on the criteria, that depend on the number of zeros, namely criterion 1, criterion 2 and criterion 5. In these criteria GreEss$_L$ returns better factorization than GreConD$_L$on almost all of the datasets. The reason is probably the logic behind the factor selection which particular algorithm utilizes. Unlike GreConD$_L$ algorithm GreEss$_L$ algorithm takes into account different role of entries, namely it utilizes the so-called essential entries [5].

Some observations that depend on data itself are for example, that `Decathlon` dataset does not contain any 0 as input value, so neither GreConD$_L$ nor GreEss$_L$fully satisfy criterion 1. `IPAQ` dataset has much more objects than attributes so we need more factors to obtain higher coverage. This is the reason why from closeness 0.9 it fails in criterion 2.

4.3 Application to Boolean Matrix Decompostion

BMF is probably the most popular class of matrix decomposition over finite scale—in this case the scale L contains only two elements, namely zero and one. Factors produced by BMF algorithms can be analyzed without any problems via our metric. We perform several experiments and we observe how good is the set of factors from the simple structure perspective.

There exist several algorithms for BMF based on different ideas. We used the following algorithms: GreConD, GreEss, Asso, Hyper, PaNDa, Tiling (for more details see e.g. [5]) and 8M [10]. Like in graded case, some of them are usually not able to achieve 100% coverage, namely Asso, PaNDa and 8M. We evaluate all of them on well known real data such as for example `Americas-small`, `DBLP`, `Emea`, `Chess` and `Mushroom`. All of them are well known and widely used. Description and characteristics of these datasets can be found e.g. in [5].

We present only basic observation. A broader analysis of results delivered by BMF algorithms is left to an extended version of this paper.

In the Boolean case, the third criterion is always true. It can be understood as: for every pair of factors, there should be an attribute which is manifestation of one of them and is not manifestation of the other one.

The best factors in terms of above defined measures are obtained by algorithm Hyper—for almost all datasets it returns value 1 (for closeness ≤ 0.95). The main reason is that Hyper usually selects factors including only one attribute. Such factors are really easy to interpret. On the other hand, this shows a drawback of the simple structure, because such factors are not useful.

GreEss, GreConD and Tiling algorithm returns comparable results (GreEss is slightly better than GreConD and both are slightly better than Tiling). All of them work in the similar way (they use formal concepts [11] as factors) and all of them do not meet the fifth criterion for high coverage.

8M, PaNDa and Asso return a set of factors which cover small amount data (in many cases they explain less that 80% of input data). Surprisingly, factors delivered via these algorithms produce the best results from the simple structure standpoint.

5 Conclusion and Future Research

We proposed a novel metric, based on a simple structure from factor analysis, for the measurement of the interpretability of factors delivered by matrix decomposition algorithms—more precisely algorithms that provide decomposition of matrices over some finite scale. Simple structure is defined via five criterion which we formalized via mathematics of Fuzzy logic. We proposed two variants of the metric, strict and partially strict and we experimentally evaluated the results produced by GreConD$_L$, GreEss$_L$ and Asso$_L$ algorithm. Additionally we provide a brief overview of experimental evaluation of selected BMF algorithms.

The observed results encourage us to the following future research directions. First, to explore different ways of the interpretability measuring. Second, to provide extensive evaluation of results produced by BMF algorithms.

Table 1: Strict metric

dataset	s	GreCond$_L$						Asso$_L$						GreEss$_L$					
		c1	c2	c3	c4	c5	total	c1	c2	c3	c4	c5	total	c1	c2	c3	c4	c5	total
Breeds	0.75	0.75	0.5	0.75	0.5	0.5	0.5	0.5	0.25	0.5	0.25	0.25	0.25	1	1	0.75	1	0.75	0.75
	0.85	0.5	0.5	0.75	0.5	0.5	0.5	0.5	0.25	0.5	0.25	0	0	1	0.75	0.75	0.75	0.5	0.5
	0.9	0.5	0.5	0.5	0.5	0.25	0.25	NA						1	0.75	0.75	0.75	0.5	0.5
	0.95	0.5	0	0.5	0.5	0.25	0	NA						1	0.25	0.75	0.75	0.5	0.25
	1	0.5	0	0.5	0.5	0.25	0	NA						1	0	0.75	0.75	0.5	0
Decathlon	0.75	0.75	0.5	0.75	0.5	0.5	0.5	0.5	0.5	1	1	1	0.5	0.75	0.5	0.5	0.5	0.5	0.5
	0.85	0.75	0.5	0.75	0.5	0.25	0.25	NA						0.75	0.5	0.5	0.5	0.5	0.5
	0.9	0.5	0.25	0.5	0.5	0.25	0.25	NA						0.75	0.25	0.5	0.5	0.5	0.25
	0.95	0.5	0.25	0.5	0.5	0.25	0.25	NA						0.75	0.25	0.5	0.5	0.5	0.25
	1	0.5	0	0.5	0.5	0.25	0	NA						0.75	0	0.5	0.5	0.25	0
IPAQ	0.75	1	1	1	1	1	1	1	0.5	0.5	0.5	0	0	1	1	1	1	1	1
	0.85	1	1	1	1	1	1	1	0	0.5	0.5	0	0	1	1	1	1	1	1
	0.9	1	0	1	1	1	0	NA						1	1	1	1	1	1
	0.95	1	0	1	1	1	0	NA						1	0	1	1	1	0
	1	1	0	0.5	1	1	0	NA						1	0	1	1	1	0
Music	0.75	0.5	0.33	1	0.33	0.17	0.17	0.5	0.33	0.67	0.33	0.33	0.33	0.33	0.33	0.83	0.33	0	0
	0.85	0.5	0.17	0.83	0.33	0.17	0.17	NA						0.33	0.17	0.83	0.33	0	0
	0.9	0.5	0.17	0.83	0.33	0.17	0.17	NA						0.33	0	0.83	0.33	0	0
	0.95	0.5	0	0.67	0.33	0.17	0	NA						0.33	0	0.83	0.33	0	0
	1	0.5	0	0.17	0.33	0.17	0	NA						0.33	0	0.17	0.33	0	0
Rio	0.75	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0.67	0.67
	0.85	1	0.67	1	1	0.67	0.67	1	1	1	1	1	1	1	1	1	1	0.67	0.67
	0.9	1	0.33	1	1	0.67	0.33	1	1	1	1	1	1	1	0.67	0.67	0.67	0.33	0.33
	0.95	1	0.33	1	0.67	0	0	NA						1	0.67	0.67	0.67	0.33	0.33
	1	1	0	0.33	0.67	0	0	NA						1	0	0.67	0.67	0.33	0

Table 2: Partially strict metric

dataset	s	GreCondD$_L$						Asso$_L$						GreEss$_L$					
		c1	c2	c3	c4	c5	total	c1	c2	c3	c4	c5	total	c1	c2	c3	c4	c5	total
Breeds	0.75	0.87	0.75	0.87	0.62	0.5	0.5	0.6	0.35	0.55	0.25	0.25	0.25	1	1	0.98	1	0.88	0.88
	0.85	0.83	0.75	0.91	0.61	0.51	0.51	0.6	0.25	0.52	0.25	0.24	0.24	1	0.93	0.94	0.99	0.84	0.84
	0.90	0.86	0.64	0.88	0.66	0.53	0.53	NA						1	0.81	0.94	0.95	0.83	0.81
	0.95	0.9	0.02	0.91	0.75	0.63	0.02	NA						1	0.6	0.95	0.96	0.84	0.6
	1	0.94	0	0.92	0.83	0.72	0	NA						1	0.02	0.96	0.96	0.85	0.02
Decathlon	0.75	0.75	0.5	0.75	0.5	0.5	0.5	0.5	0.5	1	1	1	0.5	0.75	0.67	0.67	0.5	0.5	0.5
	0.85	0.88	0.63	0.79	0.54	0.46	0.46	NA						0.8	0.55	0.75	0.55	0.5	0.5
	0.9	0.79	0.5	0.73	0.53	0.45	0.45	NA						0.83	0.54	0.78	0.58	0.5	0.5
	0.95	0.78	0.38	0.72	0.53	0.46	0.38	NA						0.84	0.5	0.78	0.6	0.51	0.5
	1	0.83	0	0.74	0.58	0.5	0	NA						0.82	0	0.73	0.58	0.49	0
IPAQ	0.75	1	1	1	1	1	1	1	0.5	0.67	0.5	0	0	1	1	1	1	1	1
	0.85	1	1	1	1	1	1	1	0.04	0.68	0.5	0.07	0.04	1	1	1	1	1	1
	0.9	1	0.86	1	1	1	0.86	NA						1	1	1	1	1	1
	0.95	1	0	1	1	1	0	NA						1	0.93	1	1	1	0.93
	1	1	0	1	1	1	0	NA						1	0	1	1	1	0
Music	0.75	0.92	0.89	1	0.79	0.62	0.62	0.58	0.5	0.67	0.33	0.33	0.33	0.89	0.86	0.99	0.73	0.57	0.57
	0.85	0.96	0.9	0.99	0.9	0.71	0.71	NA						0.95	0.9	1	0.88	0.8	0.8
	0.9	0.97	0.87	0.99	0.93	0.81	0.81	NA						0.96	0.85	1	0.92	0.89	0.85
	0.95	0.98	0.35	0.99	0.95	0.87	0.35	NA						0.97	0.67	1	0.94	0.92	0.67
	1	0.99	0	0.99	0.97	0.93	0	NA						0.98	0	0.99	0.96	0.94	0
Rio	0.75	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0.99	0.99
	0.85	1	0.98	1	1	0.99	0.98	1	1	1	1	1	1	1	1	1	1	0.99	0.99
	0.9	1	0.96	1	1	0.98	0.96	1	1	1	1	1	1	1	0.95	1	1	0.97	0.95
	0.95	1	0.93	1	1	0.95	0.93	NA						1	0.92	1	1	0.96	0.92
	1	1	0	0.99	0.99	0.9	0	NA						1	0	1	0.99	0.91	0

References

1. Belohlavek, R.: Fuzzy Relational Systems: Foundations and Principles. Kluwer Academic/Plenum Press New York (2002)
2. Belohlavek, R., Krmelova, M.: Factor analysis of sports data via decomposition of matrices with grades. In: Szathmary, L., Priss, U. (eds.) Proceedings of The Ninth International Conference on Concept Lattices and Their Applications. pp. 293–304 (2012)
3. Belohlavek, R., Krmelova, M.: Beyond boolean matrix decompositions: Toward factor analysis and dimensionality reduction of ordinal data. In: Xiong, H., Karypis, G., Thuraisingham, B.M., Cook, D.J., Wu, X. (eds.) 2013 IEEE 13th International Conference on Data Mining. pp. 961–966. IEEE Computer Society (2013)
4. Belohlavek, R., Krmelova, M.: Factor analysis of ordinal data via decomposition of matrices with grades. Ann. Math. Artif. Intell. **72**(1-2), 23–44 (2014)
5. Belohlavek, R., Trnecka, M.: From-below approximations in boolean matrix factorization: Geometry and new algorithm. J. Comput. Syst. Sci. **81**(8), 1678–1697 (2015)
6. Belohlavek, R., Vychodil, V.: Factor analysis of incidence data via novel decomposition of matrices. In: Ferré, S., Rudolph, S. (eds.) Proceedings of 7th International Conference, ICFCA 2009. Lecture Notes in Computer Science, vol. 5548, pp. 83–97. Springer (2009)
7. Boeck, P.D., Rosenberg, S.: Hierarchical classes: Model and data analysis. Psychometrika **53**, 361–381 (1988)
8. Carroll, J.B.: An analytical solution for approximating simple structure in factor analysis. Psychometrika **18**(1), 23–38 (1953)
9. Cattell, R.B.: The scientific use of factor analysis in behavioral and life sciences. Springer US (1978)
10. Dixon, W.: Bmdp statistical software manual to accompany the 7.0 software release, vols 1–3. (1992)
11. Ganter, B., Wille, R.: Formal concept analysis - mathematical foundations. Springer (1999)
12. Gottwald, S.: A Treatise on Many-Valued Logics, vol. 3. research studies press Baldock (2001)
13. Hajek, P.: Metamathematics of fuzzy logic, vol. 4. Springer Science & Business Media (1998)
14. Miettinen, P., Mielikäinen, T., Gionis, A., Das, G., Mannila, H.: The discrete basis problem. IEEE Trans. Knowl. Data Eng. **20**(10), 1348–1362 (2008)
15. Royce, J.R.: Factors as theoretical constructs. American Psychologist **18**(8), 522 (1963)
16. Spearman, C.: "General intelligence," objectively determined and measured. The American Journal of Psychology **15**(2), 201–292 (1904)
17. Thurstone, L.L.: Multiple factor analysis. University of Chicago Press: Chicago (1947)
18. Trneckova, M.: Formal concept analysis with ordinal attributes. Ph.D. thesis, Palacky University Olomouc, Czech Republic (2017)

Toward Factor Analysis of Educational Data

Eduard Bartl[1], Radim Belohlavek[1], and Alex Scharaschkin[2]

[1] Palacky University, Olomouc, Czech Republic,
eduard.bartl@upol.cz, radim.belohlavek@acm.org
[2] Oxford University, United Kingdom,
AScharaschkin@aqa.org.uk

Abstract. Factor analysis of Boolean and ordinal data became a significant research direction in data analysis. In this paper we present a case study involving a recently developed method of factor analysis of ordinal data which uses the apparatus of fuzzy logic and closure structures. In particular, the method uses formal concepts of the input data as factors and is utilized in our paper to analyze British educational data. The results of the analyses demonstrate that the method is capable of extracting natural and well-interpretable factors which provide insight into students' performances in tests. Our study represents an initial phase of a project of analyzing educational data by means of relational methods. Broader ramifications and further prospects regarding this project are also discussed.

1 Introduction and Paper Outline

Analysis of factors in various kinds of data represents an important topic in the domain of data analysis. The factors are thought of as hidden variables that are more fundamental than the directly observable variables using which the given data is described. Discovery of such factors enables one to better understand the data as well as to reduce its dimensionality. The best known factor-analytic methods are those designed for real-valued data and include the classical factor analysis, the singular value decomposition, principal component analysis, and non-negative matrix factorization; see e.g. [1,10,13,17,20]. As is well known, the application of such methods to Boolean and ordinal data is possible in principle but these classical methods suffer from poor interpretability when applied to such data. In the past years, a considerable effort has been devoted to the development of matrix methods for Boolean data; see e.g. [7,12,21,22,24] and the references therein. In our previous papers [4,5,6,8,9], we extended the factorization problem for Boolean data to ordinal data, which is of our main interest in this paper. Since in a development of new data analysis methods, explorations of real-case studies play a crucial role, our main aim in this paper is to add to our previous studies a further case study. The analyzed data comes from examination tests in the United Kingdom. Our efforts are part of a broader project whose goal is to explore methods of relational data analysis for analyzing educational data.

Our paper is organized as follows. In section 2, we present our method to the extent that both the principles as well as the user point of view are clarified.

© Dmitry I. Ignatov, Lhouari Nourine (Eds.): CLA 2018, pp. 191–206,
ISBN 978–80–244–5328–6, Department of Computer Science, Palacký University
Olomouc, 2018.

In section 3, which is the main section of this paper, we describe the data, our selected analyses of the data, and provide discussion of the results obtained. Section 4 concludes the paper by summarizing our results, putting our work in context, and describing our future goals.

2 Our Method of Factor Analysis

2.1 The Basic Idea of Our Factor Model and Its Interpretation

The Factor Model We assume that the analyzed data is in the form of an $n \times m$ matrix I describing n objects (matrix rows) and m graded attributes (matrix columns). The matrix entries I_{ij} contain degrees (grades, levels) from a given scale L, such as $L = \{0, 1/2, 1\}$ or $L = \{0, 1/4, 1/2, 3/4, 1\}$. The entry I_{ij} represents the degree to which the object i has the attribute j. Thus, $I_{ij} = 0$ means that i does not have j at all, $I_{ij} = 1$ means that i has j to the full extent, and $I_{ij} = 3/4$ means that i has j to a large extent. For instance, the objects and attributes might be students and exam tests, respectively, and the entries I_{ij} might represent the extents to which student i succeeded in test j. The following is an example of a matrix I over the three-element scale:

$$\begin{pmatrix} 1/2 & 1 & 1 & 1/2 & 1/2 \\ 1/2 & 1 & 1 & 1/2 & 1/2 \\ 0 & 1/2 & 1/2 & 1 & 1 \\ 0 & 0 & 1 & 1/2 & 1 \end{pmatrix} \tag{1}$$

In our model, one looks for a decomposition (or, factorization) of the input $n \times m$ object-attribute matrix I into an (exact or approximate) product

$$I = A \circ B, \tag{2}$$

of an $n \times k$ object-factor matrix A and a $k \times m$ factor-attribute matrix B (the entries of both A and B are again degrees from the scale L).

The matrix product $\circ$ is defined by

$$(A \circ B)_{ij} = \bigvee_{l=1}^{k} A_{il} \otimes B_{lj}, \tag{3}$$

where $\otimes$ is an appropriate aggregation function generalizing the classical logical conjunction and $\bigvee$ is the supremum operation in the scale L ($\bigvee$ is max if L is a chain, i.e. linearly ordered); see below for details.

To understand the meaning of this factor model, consider first its particular case in which $L = \{0, 1\}$, i.e. the Boolean case. Then, (3) becomes

$$(A \circ B)_{ij} = \max_{l=1}^{k} \min(A_{il}, B_{lj}) \tag{4}$$

which is the well-known Boolean matrix product. Equivalently, (4) reads:

$$(A \circ B)_{ij} = 1 \text{ iff there exists } l \in \{1, \ldots, k\} \text{ such that } A_{il} = 1 \text{ and } B_{lj} = 1,$$

from which it is immediate that the factor model has the following meaning:

$$\begin{array}{l}\text{object } i \text{ has attribute } j \text{ if and only if}\\ \quad\text{there exists factor } l \text{ such that } i \text{ has } l \text{ (or, } l \text{ applies to } i\text{)}\\ \quad\text{and } j \text{ is one of the particular manifestations of } l,\end{array} \tag{5}$$

which may be regarded as a verbal description of the model given by (2). Such description is certainly appealing and well understandable.

With a general scale L, we approach the situation according to the principles of formal fuzzy logic [2,15,16] as follows. We consider the formulas $\varphi(i,l)$ saying "object i has factor l" and $\psi(l,j)$ saying "attribute j is a manifestation of factor l", and regard A_{il} as the truth degree $||\varphi(i,l)||$ of $\varphi(i,l)$, and B_{lj} as the truth degree $||\psi(l,j)||$ of $\psi(l,j)$, i.e.

$$||\varphi(i,l)|| = A_{il} \text{ and } ||\psi(l,j)|| = B_{lj}. \tag{6}$$

Now, according to fuzzy logic, the truth degree of the formula $\varphi(i,l)\&\psi(l,j)$ which says "object i has factor l and attribute j is a manifestation of factor l" is computed by

$$||\varphi(i,l)\&\psi(l,j)|| = ||\varphi(i,l)|| \otimes ||\psi(l,j)||,$$

where $\otimes : L \times L \to L$ is a truth function of many-valued conjunction & (several reasonable functions exist). Hence, the truth degree of $(\exists l)(\varphi(i,l)\&\psi(l,j))$ which says "there exists factor l such that object i has l and attribute j is a manifestation of l", i.e. the proposition involved in (5), is computed by

$$||(\exists l)(\varphi(i,l)\&\psi(l,j))|| = \textstyle\bigvee_{l=1}^{k} ||\varphi(i,l)|| \otimes ||\psi(l,j)||,$$

where $\bigvee$ denotes the supremum. Taking (6) into account, we see that a generalization of (4) to the case of multiple degrees in L is just given by the above formula (3). Therefore, even in presence of multiple degrees, the factor model (2) retains its simple meaning described by (5).

Scales of Degrees and Truth Functions $\otimes$ *and* $\to$ Technically, we assume that the grades are taken from a partially ordered bounded scale L of certain type. In particular, we assume that L conforms to the structure of a complete residuated lattice [14,25], used in fuzzy logic; see [15,16] for details. Grades of ordinal scales [19] are conveniently represented by numbers, such as the Likert scale $\{1,\dots,5\}$, which naturally appears in our experiments (see below). We assume that these numbers are normalized and taken from the unit interval $[0,1]$, i.e. they form the five-element scale $L = \{0, {}^1\!/_4, {}^1\!/_2, {}^3\!/_4, 1\}$ commonly used in fuzzy logic. In our analyses, we use the Łukasiewicz operations on this scale, i.e. we use

$$a \otimes b = \max(0, a+b-1) \text{ and } a \to b = \min(1, 1-a+b),$$

but many other examples are available; see e.g. [15].

2.2 Factors Utilized by Our Method

It follows from the above description that for any decomposition (2), the lth factor ($l \in \{1, \ldots, k\}$) is represented by two parts: the lth column A_l of A and the lth row $B_{l\_}$ of B. As shown in [4], optimal factors for a decomposition of I (see below) are provided by formal concepts associated to I. In detail, let $X = \{1, \ldots, n\}$ (rows/objects) and $Y = \{1, \ldots, m\}$ (columns/attributes). A formal concept of I is any pair $\langle C, D\rangle$ of L-sets (fuzzy sets, [14,26]) $C : \{1, \ldots, n\} \to L$ of objects and $D : \{1, \ldots, m\} \to L$ of attributes, see [3], that satisfies $C^{\uparrow} = D$ and $D^{\downarrow} = C$ where $^{\uparrow} : L^X \to L^Y$ and $^{\downarrow} : L^Y \to L^X$ are the concept-forming operators defined by

$$C^{\uparrow}(j) = \textstyle\bigwedge_{i\in X}(C(i) \to I_{ij}) \text{ and } D^{\downarrow}(i) = \textstyle\bigwedge_{j\in Y}(D(j) \to I_{ij}).$$

The set of all formal concepts of I is denoted by $\mathcal{B}(X, Y, I)$ or just $\mathcal{B}(I)$. $C(i) \in L$ and $D(j) \in L$ are interpreted as the degree to which factor l applies to object i and the degree to which attribute j is a manifestation of factor l. Using formal concepts as factors is optimal in the following sense [4]: Let for a set (we fix the numbering of its elements)

$$\mathcal{F} = \{\langle C_1, D_1\rangle, \ldots, \langle C_k, D_k\rangle\} \subseteq \mathcal{B}(X, Y, I)$$

of formal concepts denote by $A_{\mathcal{F}}$ and $B_{\mathcal{F}}$ the matrices defined by

$$(A_{\mathcal{F}})_{il} = (C_l)(i) \quad \text{and} \quad (B_{\mathcal{F}})_{lj} = (D_l)(j).$$

Then whenever $I = A \circ B$ for some $n \times k$ and $k \times m$ matrices A and B, there exists a set $\mathcal{F} \subseteq \mathcal{B}(X, Y, I)$ $|\mathcal{F}| \leq k$ such that $I = A_{\mathcal{F}} \circ B_{\mathcal{F}}$, i.e. optimal decompositions are attained by formal concepts as factors.

In our experiments, we use the basic greedy algorithm proposed in [8] for computing a set $\mathcal{F}$ of concepts for which $I = A_{\mathcal{F}} \circ B_{\mathcal{F}}$; see also [9] for computational complexity of the problem and the algorithm.

2.3 Explanation of Data by Factors

If a set $\mathcal{F} \subseteq \mathcal{B}(X, Y, I)$ of formal concepts of I satisfies $I = A_{\mathcal{F}} \circ B_{\mathcal{F}}$, we intuitively regard $\mathcal{F}$ as fully explaining the data represented by I and call $\mathcal{F}$ a set of *factor concepts*. In general, however, we are interested in small $\mathcal{F}$ for which I is close enough to the product $A_{\mathcal{F}} \circ B_{\mathcal{F}}$, i.e. to the data reconstructed from the factors in $\mathcal{F}$. To measure closeness of I and $A_{\mathcal{F}} \circ B_{\mathcal{F}}$, we use the function $s(I, A_{\mathcal{F}} \circ B_{\mathcal{F}})$ defined by

$$s(I, A_{\mathcal{F}} \circ B_{\mathcal{F}}) = \frac{\sum_{i,j=1}^{n,m}(I_{ij} \leftrightarrow (A_{\mathcal{F}} \circ B_{\mathcal{F}})_{ij})}{n \cdot m}, \tag{7}$$

where $\leftrightarrow$ is the biresiduum (i.e. many-valued logical equivalence). This function is proposed in the above-mentioned papers. In fact, it turned out during our experiments that we need a slight generalization of this function, which we describe below.

3 Educational Data and Its Factor Analysis

3.1 A Broader Context

Analyzing students' performance is a task which constantly occupies educators. On a small scale, teachers are naturally interested in performances of their individual students as well as of their classes to help their students improve, with respect to educational aims and objectives. On a large scale, understanding students' performance is of great interest at the national level: Education experts attempt to understand the effects of current curricula and approaches to education to possibly improve educational policies. Our project fits into this picture. We attempt to analyze students' performances as assessed by the tasks they attempts in examinations. Unlike the common approaches, which are mostly based on classical statistical methods, we propose to utilize the recently developed method of factor analysis of ordinal data described in the previous section. The limited extent of this paper prevents us from describing the results we obtained to any larger extent as well as from describing broader ramifications of the findings and comparison to analyses obtained by alternative methods. We therefore present a fraction of our results only, which nevertheless meets our primary purpose in this paper, namely to demonstrate that our method is capable of revealing natural and well-interpretable factors hidden in the outputs of educational assessments.

3.2 The Data

We analyze anonymized data coming from the official school-leaving (so-called A-level) examination tests that are used in the UK by universities to select students. In brief, our overall aim in this project is to see what factors may explain the students' performances. In addition, we are interested in the question of the so-called *construct validity* [23] of the examinations, namely the extent to which students' responses, assessed as being at a particular level, match the intentions of the assessment designers in terms of the qualitative performance standard intended to broadly characterise responses at that level. This is the kind of question that is difficult to study using traditional quantitative methods.

The data contains results of 2774 individual students' performances on a given examination in the subject "Government and Politics." The whole examination consists of four modules (i.e. four papers). Our data concerns the second module, which covers the current British governance.[3] Students choose two topics out of four possible. Our data contains the results for topic 2 (parliament) and topic 3 (the executive), which is the most popular combination. For each topic (2 and 3), the students answer three questions, i.e. six questions in total. The answer to each question, which is a piece of prose, is assessed by examiners with regard to three so-called assessment objectives, namely "knowledge and understanding," "analysis and evaluation," and "communication," with the exception

[3] The exam paper is available at `http://filestore.aqa.org.uk/sample-papers-and-mark-schemes/2016/june/AQA-GOVP2-QP-JUN16.PDF`.

of the first question in each topic for which only the first assessment objective is considered. As a result, 14 evaluations for each student examination (one for each topic, question, and permissible assessment objective) are obtained. A student obtains a mark in each of the 14 evaluations (the largest possible marks for the evaluations are mutually different in general). The sum of all marks gives the total mark of maximum value 80 assigned to this student, from which the grade for the student is obtained by a simple thresholding. The possible grades are A (the best grade, represented numerically by 5), B (represented by 4), C (3), D (2), E (1), and N (0). In addition, a simple scaling is defined for each of the 14 evaluations which assigns each possible mark for this evaluation a level on a five-element scale $0, \dots, 4$ with 4 indicating the best performance. This scaling brings the evaluation data on a common scale.[4] Two exceptions are attributes 5 and 8 which are mapped on a three-element scale (level 0 represented by 0, level 1-2 represented by 1.5, and level 3-4 represented by 3.5). We nevertheless embed this three-element scale to the five-element one by the assignment $0 \mapsto 0$, 1-2 $\mapsto 1$, and 3-4 $\mapsto 3$. Each student examination is thus described by 14 fuzzy (graded) attributes over a five-element scale $L = \{0, {}^1\!/\!_4, {}^1\!/\!_2, {}^3\!/\!_4, 1\}$ whose degrees represent the levels $0, \dots, 4$. A sample of the data sorted by the total marks is shown in Table 1: The first column represents the total marks, the second one represents the grades, and the remaining columns represent the 14 graded attributes, which are explained in Table 2.

3.3 Selected Analyses

The data described in the previous section may thus be represented by a 2774×14 matrix I with degrees in the scale $L = \{0, {}^1\!/\!_4, {}^1\!/\!_2, {}^3\!/\!_4, 1\}$, i.e. a matrix $I \in L^{2774 \times 14}$. A part of this matrix corresponding to the data from Table 1 is shown in Table 3.

We performed factor analyses of this data using the method described in section 2. In accordance to the intentions to understand the factors behind the various overall performance grades, we split the matrix I into 6 submatrices according to the grades. Thus, since there are 607 students who obtained grade A, we analyzed the corresponding 607×14 submatrix I_{A} of the whole matrix I and we performed this for the submatrices I_{G} for every grade G = A, …, E, N.

Since our algorithm computes the factors one by one, from the most significant ones in terms of data coverage to the least significant until an exact factorization of the input matrix I_{G} is obtained, we observed the coverage of the data I_{G} by the first factor, by the first two factors, by the first three factors, and in general by the first $l = 1, \dots, k$ factors where k is the total number of factors computed from the data. To measure coverage, which serves as an indicator of how well the data is explained by the factors, we first used the function (7) [5,6,8,9], which is a direct generalization of the coverage function from the Boolean case. We, however, observed a phenomenon not encountered in

[4] The marking scheme is described in http://filestore.aqa.org.uk/sample-papers-and-mark-schemes/2016/june/AQA-GOVP2-W-MS-JUN16.PDF.

Table 1: Sample of the examination data.

total	grade	k__cor__par	k__cor__exe	k__par__mac	e__par__mac	c__par__mac	k__cab__gov	e__cab__gov	c__cab__gov	k__par__mod	e__par__mod	c__par__mod	k__pow	e__pow	c__pow
78	5	4	4	4	4	3,5	4	4	3,5	4	4	4	4	4	4
77	5	4	4	4	4	3,5	3	2	3,5	4	4	4	4	4	4
76	5	3	3	4	4	3,5	4	4	3,5	4	4	4	4	4	4
75	5	2	4	4	4	3,5	3	3	3,5	4	4	4	4	4	4
75	5	3	3	4	3	3,5	4	4	3,5	4	4	4	4	4	4
74	5	4	3	4	3	3,5	4	4	3,5	4	4	4	4	4	4
73	5	4	3	4	3	3,5	3	3	3,5	4	4	4	4	4	4
73	5	3	4	4	3	3,5	4	3	3,5	4	4	4	4	4	4
73	5	3	3	4	4	3,5	4	3	3,5	4	4	4	4	3	4
73	5	4	3	4	4	3,5	4	4	3,5	3	4	4	4	4	4
73	5	4	4	4	3	3,5	4	3	1,5	4	4	4	4	4	4
73	5	3	4	4	3	3,5	3	4	3,5	4	4	4	4	4	4
73	5	4	4	4	4	3,5	3	3	3,5	4	4	4	4	4	4
73	5	3	3	4	4	3,5	3	4	3,5	4	4	4	4	4	4
73	5	3	2	3	3	3,5	4	4	3,5	4	3	4	4	4	4
72	5	4	4	4	4	3,5	4	4	3,5	4	4	4	3	3	3
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮

Table 2: Meaning of the 14 graded attributes.

attribute	label in table 3	description
1	k__cor__par	demonstrates knowledge of core parliamentary principles
2	k__cor__exe	demonstrates knowledge of core principles of the executive
3	k__par__mac	demonstrates knowledge of parliamentary machinery
4	e__par__mac	explains and analyses aspects of parliamentary machinery
5	c__par__mac	communicates effectively about aspects of parliamentary machinery
6	k__cab__gov	demonstrates knowledge of cabinet government
7	e__cab__gov	explains and analyses aspects of cabinet government
8	c__cab__gov	communicates effectively about aspects of cabinet government
9	k__par__mod	demonstrates knowledge of pros and cons of parliamentary models
10	e__par__mod	explains and analyses pros and cons of parliamentary models
11	c__par__mod	communicates effectively about pros and cons of parliamentary models
12	k__pow	demonstrates a knowledge of power structures in government
13	e__pow	explains and analyses power structures in government
14	c__pow	communicates effectively about power structures in government

Table 3: Matrix with grades, or a formal fuzzy context, representing the data from Table 1.

1	1	1	1	3/4	1	1	3/4	1	1	1	1	1	1
1	1	1	1	3/4	3/4	1/2	3/4	1	1	1	1	1	1
3/4	3/4	1	1	3/4	1	1	3/4	1	1	1	1	1	1
1/2	1	1	1	3/4	3/4	3/4	3/4	1	1	1	1	1	1
3/4	3/4	1	3/4	3/4	1	1	3/4	1	1	1	1	1	1
1	3/4	1	3/4	3/4	1	1	3/4	1	1	1	1	1	1
1	3/4	1	3/4	3/4	3/4	3/4	3/4	1	1	1	1	1	1
3/4	1	1	3/4	3/4	1	3/4	3/4	1	1	1	1	1	1
3/4	3/4	1	1	3/4	1	3/4	3/4	1	1	1	1	3/4	1
1	3/4	1	1	3/4	1	1	3/4	3/4	1	1	1	1	1
1	1	1	3/4	3/4	1	3/4	1/4	1	1	1	1	1	1
3/4	1	1	3/4	3/4	3/4	1	3/4	1	1	1	1	1	1
1	1	1	1	3/4	3/4	3/4	3/4	1	1	1	1	1	1
3/4	3/4	1	1	3/4	3/4	1	3/4	1	1	1	1	1	1
3/4	1/2	3/4	3/4	3/4	1	1	3/4	1	3/4	1	1	1	1
1	1	1	1	3/4	1	1	3/4	1	1	1	3/4	3/4	3/4
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮

the previous analyzes reported in the literature which is due to the considerable size of the data (a 2774 × 14 matrix over a 5-element scale is comparable to a $2774 \times (14 \cdot 5) = 2774 \times 70$ Boolean matrix [6]). Namely, the accumulation of the biresidua $I_{ij} \leftrightarrow (A_{\mathcal{F}} \circ B_{\mathcal{F}})_{ij}$ by the summation in (7) makes the algorithm select also flat factors. By "flat" we mean that the entries $C_l(i) \otimes D_l(j)$, by which the factor $\langle C_l, D_l \rangle$ contributes to the explanation of the input data, are close to $1/2$. In many such cases, we would naturally prefer factors that are less flat even though their coverage as measured by (7) is slightly smaller, because such factors are more discriminative and thus more informative. To solve this problem, we adjusted the function (7) as follows: The new function, $s_c(I, A_{\mathcal{F}} \circ B_{\mathcal{F}})$, is defined by

$$s_c(I, A_{\mathcal{F}} \circ B_{\mathcal{F}}) = \frac{\sum_{i,j=1}^{n,m} (c(I_{ij} \leftrightarrow (A_{\mathcal{F}} \circ B_{\mathcal{F}})_{ij}))}{n \cdot m}, \tag{8}$$

where $c : L \to L$ is an appropriate increasing function satisfying $c(0) = 0$ and $c(1) = 1$. In our analyses, we used $c(a) = a^{q\sqrt{mn}}$ and obtained satisfactory results for $q = 0.1$, which we report below. The effect of using $c(I_{ij} \leftrightarrow (A_{\mathcal{F}} \circ B_{\mathcal{F}})_{ij})$ is the following. The value $I_{ij} \leftrightarrow (A_{\mathcal{F}} \circ B_{\mathcal{F}})_{ij}$ measures closeness of the values at the $\langle i, j \rangle$ entry of the original matrix I and the matrix $A_{\mathcal{F}} \circ B_{\mathcal{F}}$ reconstructed from the computed set $\mathcal{F}$ of factors. Transforming this value by the monotone c emphasizes entries that are very close while inhibiting those that are not so close. The rate of inhibition is parameterized by the geometric mean $\sqrt{mn}$ of the number m of attributes and the number n of rows of the matrix, and a parameter q.

We now briefly describe the results for two grades, namely grade A and grade E. Grade A was attained by 607 students. Our algorithm obtained 36 factors from the 607×14 matrix I_{A}. The cumulative coverage of these factors is depicted in Fig. 1 and Table 4. The depicted coverage values corresponding to the sets $\mathcal{F}_l = \{F_1, \ldots, F_l\}$ consisting of the first l factors $F_i = \langle C_i, D_i \rangle$ computed by the algorithm are the values $s_c(I, A_{\mathcal{F}_l} \circ B_{\mathcal{F}_l})$ defined by (8). Thus, we can observe that the coverage by the first, the first two, and the first three factors is 0.417, 0.539, and 0.618, respectively. As one can see, a reasonable coverage (around 0.75 and more) is obtained by the first five factors already.

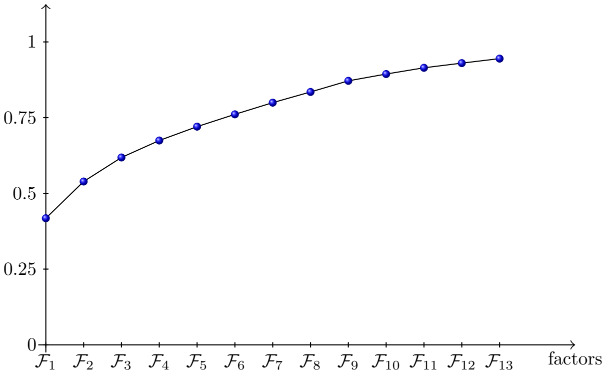


Fig. 1: Cumulative coverage by factors (grade A).

Table 4: Cumulative coverage by factors (grade A).

$\mathcal{F}_1$	$\mathcal{F}_2$	$\mathcal{F}_3$	$\mathcal{F}_4$	$\mathcal{F}_5$	$\mathcal{F}_6$	$\mathcal{F}_7$	$\mathcal{F}_8$	$\mathcal{F}_9$	$\mathcal{F}_{10}$	$\mathcal{F}_{11}$	$\mathcal{F}_{12}$	$\mathcal{F}_{13}$	...	$\mathcal{F}_{36}$
0.417	0.539	0.618	0.674	0.720	0.760	0.799	0.834	0.871	0.893	0.914	0.930	0.944	...	1

Let us now describe in detail the first three factors, i.e. the three most important ones according to the algorithm. The extent and the intent of the first factor, $F_1 = \langle C_1, D_1 \rangle$, is depicted in Fig. 2 and Fig. 3, respectively.

The intent, which conveys the meaning of each factor, is a fuzzy set assigning to every attribute y_i $(i = 1, \ldots, 14)$ a value in the scale $L = \{0, {}^1\!/_4, {}^1\!/_2, {}^3\!/_4, 1\}$. This value is interpreted as the degree to which the particular attribute y_i is a manifestation of the given factor. That is, the degree to which good performance on the attribute y_i accompanies the presence of the factor. Fig. 3 displays such a fuzzy set for the first factor obtained by the algorithm, i.e. the fuzzy set D_1.

The extent is a fuzzy set assigning to every student (with grade A) a degree in the scale L to which the student possesses the given factor. Fig. 2 presents such a fuzzy set, i.e. C_1, for the first factor obtained. Since there are 607 students with grade A, the graph of C_1 is somewhat condensed (there are 607 points on the horizontal axis, hence 607 vertical bars indicating the assigned grades).

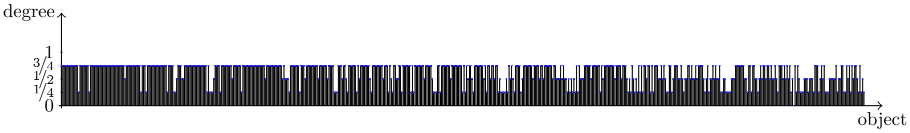


Fig. 2: Extent of F_1 (grade A).

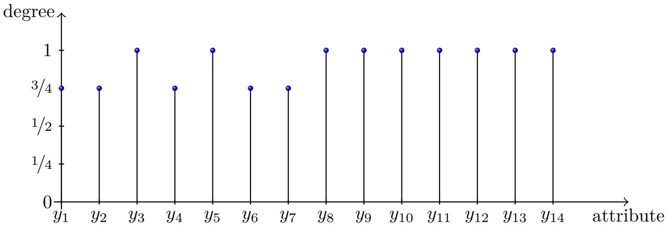


Fig. 3: Intent of F_1 (grade A).

In particular, one may observe from the intent D_1 and the description of the attributes $y_1, \ldots, y_{14}$ in Table 2 that the first factor may verbally be described as "excellent overall knowledge, and excellent analytical and communication skills," because the factor displays almost all attributes to the highest possible degree. From Fig. 2 one may see that this factor is possessed by most of the students who obtained grade A to the second-highest degree, $^3/_4$. Since the students are ordered on the horizontal axis by the their total marks, the graph also tells us that the factor appears in particular on the students with the highest total marks. Such a factor is a natural and expected one and from this viewpoint, our algorithm confirms the intuitive expectations.

The second factor, F_2, is depicted in Fig. 4 (extent) and Fig. 5 (intent). This factor may be interpreted as displaying very good overall knowledge with slightly limited communication skills and slightly limited knowledge of government power structures. Most of the students, particularly those with high total marks, possess this factor to a high degree, the best students even to the highest possible degree. Only one student does not possess this factor at all (i.e. the corresponding degree for this student in the extent is 0).

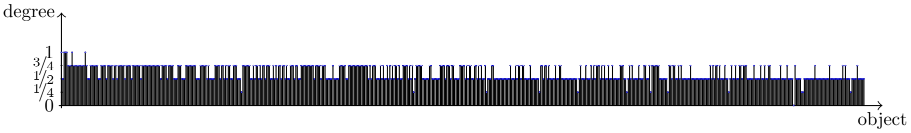


Fig. 4: Extent of F_2 (grade A).

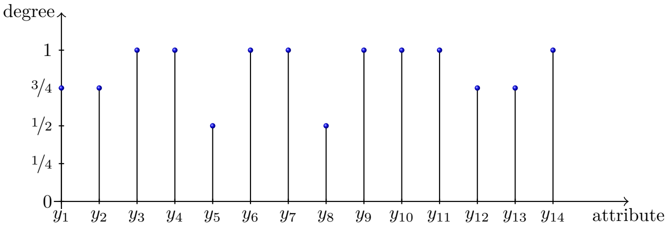


Fig. 5: Intent of F_2 (grade A).

The third factor, F_3, is depicted in Fig. 6 (extent) and Fig. 7 (intent). It may be interpreted as manifesting excellent knowledge in the first two questions (attributes y_1 and y_2), only a moderate knowledge of parliamentary machinery (y_3, y_4, and y_5), virtually no knowledge of cabinet government (y_6, y_7, and y_8), and reasonable knowledge of parliamentary models (y_9, y_{10}, and y_{11}) and government power structures (y_{12}, y_{13}, and y_{14}). This factor, which is possessed by many students to a high degree, is considerably discriminative and therefore interesting.

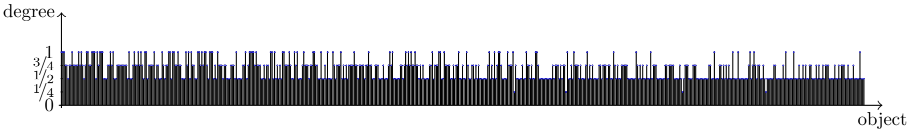


Fig. 6: Extent of F_3 (grade A).

Let us now turn to the analysis of performances of students who obtained grade E. Due to limited scope, our main purpose is to demonstrate that our method reveals different factors from the data for grade E compared to the data for grade A, which is in accordance with intuitive expectations. Grade E was attained by 322 students and our algorithm obtained 29 factors from the 322×14 matrix I_{E}. The cumulative coverage of these factors is depicted in Fig. 8 and Table 5. The depicted coverage values again correspond to the sets $\mathcal{F}_l = \{F_1, \ldots, F_l\}$ consisting of the first l factors $F_i = \langle C_i, D_i \rangle$ computed by the algorithm. As with grade A, we can observe that a reasonable coverage (around 0.75 and more) is obtained by the first five factors already.

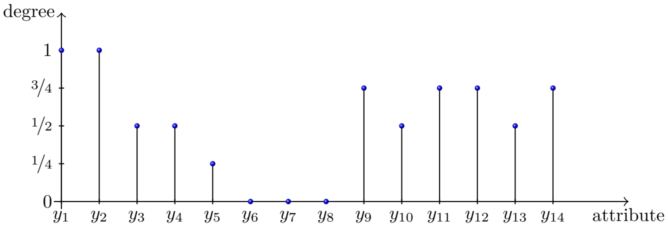


Fig. 7: Intent of F_3 (grade A).

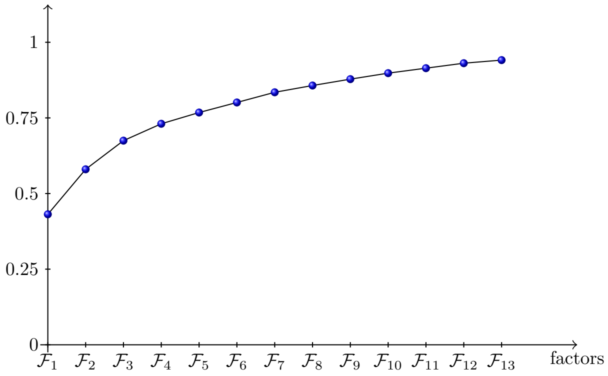


Fig. 8: Cumulative coverage by factors (grade E).

Table 5: Cumulative coverage by factors (grade E).

$\mathcal{F}_1$	$\mathcal{F}_2$	$\mathcal{F}_3$	$\mathcal{F}_4$	$\mathcal{F}_5$	$\mathcal{F}_6$	$\mathcal{F}_7$	$\mathcal{F}_8$	$\mathcal{F}_9$	$\mathcal{F}_{10}$	$\mathcal{F}_{11}$	$\mathcal{F}_{12}$	$\mathcal{F}_{13}$	…	$\mathcal{F}_{29}$
0.431	0.580	0.674	0.730	0.767	0.801	0.834	0.857	0.878	0.897	0.914	0.930	0.941	…	1

The most significant factor in the data for grade E is depicted in Fig. 9 (extent) and Fig. 10 (intent). We may observe that the factor is possessed by most students to the degree 3/4 and by a considerably high number of students even to the highest possible degree. This factor may be described as manifesting no or very limited knowledge in all questions except for questions regarding parliamentary models and governmental power structures, for which the students who possess this factor exhibit moderate performance with respect to all three assessment objectives.

The second most significant factor for grade E is depicted in Fig. 11 (extent) and Fig. 12 (intent). This factor is possessed by almost all students to degree 1/2 and by several of them even to degree 3/4. None of the students with grade E possesses this factor to the highest possible degree. The factor is manifested by a very limited knowledge of the first two questions (y_1 and y_2) and moderate knowledge of the remaining questions except for the question about governmental power structures where the manifested performance is severely limited.

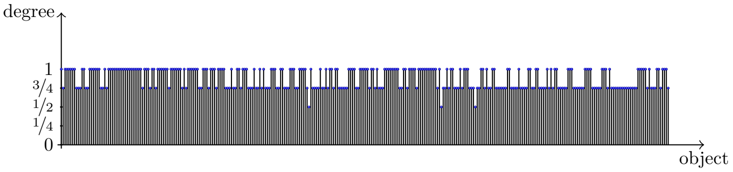


Fig. 9: Extent of F_1 (grade E).

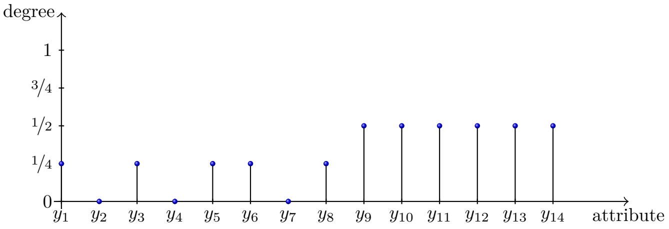


Fig. 10: Intent of F_1 (grade E).

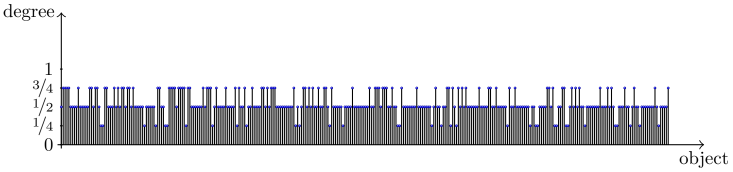


Fig. 11: Extent of F_2 (grade E).

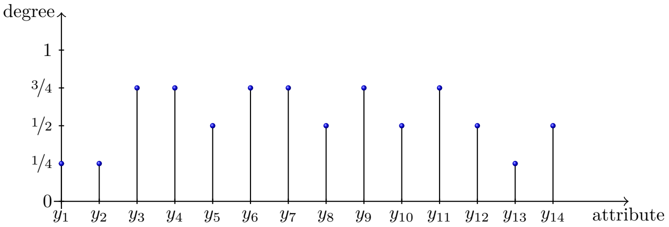


Fig. 12: Intent of F_2 (grade E).

The third most significant factor for grade E is depicted in Fig. 13 (extent) and Fig. 14 (intent). As is apparent from the intent of this factor, the factor is manifested by reasonably good knowledge in most of the questions. Nevertheless, the factor is possessed to very small degrees by the students with grade E and, therefore, is not as significant as the previous factors.

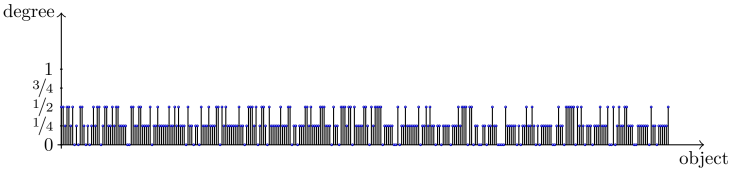


Fig. 13: Extent of F_3 (grade E).

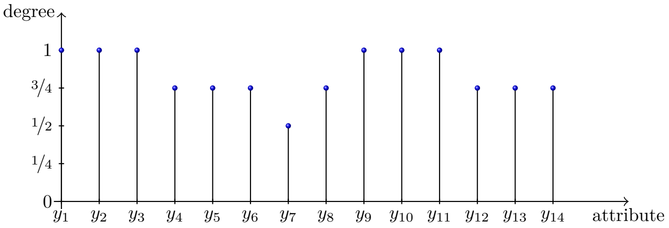


Fig. 14: Intent of F_3 (grade E).

4 Conclusions and Further Steps

The purpose of this paper is twofold. For one, we provide further analyses of real-world data using the recently developed method of factor analysis of ordinal data described in [4,5,6,8,9]. Secondly, we provide some first steps in our long-term project of utilizing relational methods of data analysis, in particular the methods related to formal concept analysis [11], in understanding students' performance data.

We demonstrated by our analyses that students' performance data, which consists of a collection of ordinal attributes, may naturally be subject to analysis by the methods we explore, in particular by the present method of factor analysis. We also demonstrated that the method yields naturally interpretable factors from data which are easy to understand, adding thus further evidence of a practical value of the method.

The limited scope of this paper does not allow us to go into the ramifications of our results obtained so far for educational policy makers. Formulating such ramifications is the ultimate goal for our research. Nevertheless, a proper methodology and experimental basis has first to be developed. Our present method and the reported experiments are to be considered as the first steps in this regard. The natural next steps seem to be the following. Firstly, we plan to further develop the present method of factor analysis. One direction is to adjust the method to be capable of extracting factors with a pattern preferred by the users of the method. An example is the flatness of the factors mentioned above. Another direction, which emerged during our experiments, is to allow a reasonable interaction with the user of the method. As the factors are generated one-by-one, we plan to provide the user with the option to accept or reject a candidate factor, hence the option to control the very process of factorization. In face of the extent of the data, we also plan to explore a possible statistical enhancement of our method. Secondly, we plan to compare the results of our factor analyses to the results obtained by alternative factor-analytic methods, as well as put our work in further works on analyzing educational data by relational methods, e.g. [18]. Thirdly, we plan to explore further methods related to formal concept analysis in analyzing students' performance data.

Acknowledgment

R. Belohlavek and E. Bartl acknolwedge support by the ECOP (Education for Competitiveness Operational Programme) project No. CZ.1.07/2.3.00/20.0059, which was co-financed by the European Social Fund and the state budget of the Czech Republic. The present research has been conducted in the follow-up period of this project.

References

1. Bartholomew, D.J., Knott M.: Latent Variable Models and Factor Analysis. 2nd Ed., Arnold, London (1999)
2. Belohlavek, R.: Fuzzy Relational Systems: Foundations and Principles. Kluwer, Academic/Plenum Publishers, New York (2002)
3. Belohlavek, R.: Concept lattices and order in fuzzy logic. Annals of Pure and Applied Logic **128** (1–3), 277–298 (2004)
4. Belohlavek, R.: Optimal decompositions of matrices with entries from residuated lattices. J. Logic and Computation **22** (6), 1405–1425 (2012)
5. Belohlavek, R., Krmelova, M.: Beyond Boolean matrix decompositions: Toward factor analysis and dimensionality reduction of ordinal data. In: 13th International Conference on Data Mining, pp. 961–966. IEEE (2013)
6. Belohlavek, R., Krmelova, M.: Factor analysis of ordinal data via decomposition of matrices with grades. Ann. Mathematics and Artificial Intelligence **72** (1), 23–44 (2014)
7. Belohlavek, R., Vychodil, V.: Discovery of optimal factors in binary data via a novel method of matrix decomposition. J. Computer and System Sciences **76** (1), 3–20 (2010)
8. Belohlavek, R., Vychodil, V.: Factor analysis of incidence data via novel decomposition of matrices. LNAI **5548**, 83—97 (2009)
9. Belohlavek, R., Vychodil, V.: Factorization of matrices with grades. Fuzzy Sets and Systems **292**, 85–97 (2016)
10. Eldén, L.: Matrix Methods in Data Mining and Pattern Recognition. SIAM, Philadelphia, PA (2007)
11. Ganter, B., Wille, R.: Formal Concept Analysis. Mathematical Foundations. Springer, Berlin (1999)
12. Geerts, F., Goethals, B., Mielikäinen, T.: Tiling databases. LNCS **3245**, 278–289 (2004)
13. Golub, G., Van Loan, C.: Matrix Computations. Johns Hopkins University Press (1996)
14. Goguen, J.A.: The logic of inexact concepts. Synthese **18**, 325–373 (1968–9)

15. Gottwald, S.: A Treatise on Many-Valued Logics. Research Studies Press, Baldock, Hertfordshire, England (2001)
16. Hájek, P.: Metamathematics of Fuzzy Logic. Kluwer, Dordrecht (1998)
17. Harman, H.H.: Modern Factor Analysis. 2nd Ed. The Univ. Chicago Press, Chicago (1970)
18. Ignatov, D.I., Mamedova, S., Romashkin, N., Shamshurin, I.: What Can Closed Sets of Students and Their Marks Say? EDM, 223–228 (2011)
19. Krantz, H.H., Luce, R.D., Suppes P., Tversky A.: Foundations of Measurement. Vol. I (Additive and Polynomial Representations), Vol. II (Geometric, Threshold, and Probabilistic Represenations), Vol. III (Represenations, Axiomatization, and Invariance). Dover Edition (2007)
20. Lee, D., Seung, H.: Learning the parts of objects by non-negative matrix factorization. Nature **401**, 788–791 (1999)
21. Miettinen, P., Mielikäinen, T., Gionis, A., Das, G., Mannila, H.: The discrete basis problem. IEEE Trans Knowledge and Data Eng **20** (10), 1348–1362 (2008)
22. Miettinen, P., Vreeken, J.: Model order selection for Boolean matrix factorization. Proc. ACM SIGKDD, 51–59 (2011)
23. Newton, P.E., Shaw, S.D.: Validity in Educational and Psychological Assessment. SAGE, London (2014)
24. Tatti, N., Mielikäinen, T., Gionis, A., Mannila, H.: What is the dimension of your binary data? In: The 2006 IEEE Conference on Data Mining, IEEE Computer Society, 603–612 (2006)
25. Ward, M., Dilworth, R.P.: Residuated lattices. Trans. Amer. Math. Soc. **45**, 335–354 (1939)
26. Zadeh, L.A.: Fuzzy sets. Inf. Control **8**, 338–353 (1965)

Interactive Query Refinement using Formal Concept Analysis

Tim Pattison

tim.pattison@dst.defence.gov.au
Defence Science & Technology Group
West Ave, Edinburgh
South Australia 5111

Abstract. Formal Concept Analysis (FCA) of a document corpus yields a concept lattice uniting the powersets of corpus terms and documents. Structural navigation of the directed graph connecting neighbours in this lattice affords interactive query refinement. The starting point for navigation is typically the closure of the conjunctive Boolean query with respect to the corpus. This paper describes the special treatment of "closure" terms – those in this closure but which are not specified by the user – and its implications for *implicit* structural navigation of the digraph through query editing. This approach, in which the user's choice of each additional query term is constrained to avoid the null result set, is contrasted with explicit structural navigation. If a term is added which does not co-occur with the terms already specified, the user must either remove the new term or choose which other term(s) to remove. A novel technique is used to present the user with options for those other terms.

1 Introduction

1.1 Formal Concept Analysis for Interactive Query Refinement

Formal Concept Analysis (FCA) has been used in corpus-based information retrieval for the construction and interactive refinement of conjunctive Boolean queries (see e.g. [15,14,16,6]). Here, a query is a set of terms, all of which must be present in each document retrieved in response to the query. The derivation operator, which maps the query onto the set of matching documents, induces a Galois connection between the set of potential queries – the powerset of terms appearing in the corpus – and the set of potential results – the powerset of documents in the corpus. This choice elegantly combines the query and result spaces into a unified structure – the concept lattice – which can be navigated either explicitly or implicitly by the user for interactive refinement of the query.

The set of documents returned by a query, and the set of query terms, augmented by any other terms – known as *closure* terms – which are common to all documents in the result set, constitute a *formal concept*. The association between terms present in the corpus and the documents in which they occur is known as a *formal context*, and is often specified in the form of a binary matrix. A simple example formal context is shown in Figure 1a.

© Dmitry I. Ignatov, Lhouari Nourine (Eds.): CLA 2018, pp. 207–218,
ISBN 978–80–244–5328–6, Department of Computer Science, Palacký University
Olomouc, 2018.

Given a formal context, FCA enumerates the set of formal concepts, partially orders them by document set subsumption, and computes the neighbour relation as the transitive reduction of this ordering relation. The result is a single-source, single sink, directed acyclic graph (DAG), whose vertices represent formal concepts and whose arcs connect neighbours.

A lower [upper] neighbour of a concept represents a minimal conjunctive expansion [contraction] of the corresponding set of query terms with respect to the corpus, and the consequently contracted [expanded] result set[1]. These neighbours constitute the available options for incremental modification of the current query. The dominant paradigm for FCA-based query refinement involves explicit navigation via (undirected) edges in this graph – a process which we refer to as *structural navigation.* Navigation is typically constrained to neighbours of the query concept, although concepts beyond neighbours are sometimes also offered as candidates [12,5]. The graph can be presented to the user in its entirety [8], or, more usually, as a keyhole view of the neighbourhood of the current concept [2,4,3,7].

1.2 Query Editing with Ranked Refinement Options

Lindig [12] instead presented users with a list of terms in the current query, any one of which could be selected for removal, and a second list of terms which, when selected and thereby added to the query, would return a non-empty result set. The user is also shown the result set as a list of identifiers, and this list is updated with each term selection. This approach supports implicit navigation of the concept lattice to super- and sub-concepts – not necessarily neighbours – without explicitly exposing the user to the concept lattice or requiring them to comprehend it. To refine their query, the user is thereby focused on editing the query itself rather than navigating the concept lattice.

In contrast with structural navigation, the use of term lists as the primary interface for interactive query refinement naturally supports any scheme which ranks terms to assist the user's choice. Using Latent Semantic Analysis, the SORTEd prototype [13] ranks the lists of query and unused terms according to their "semantic" relevance to the result set of a conjunctive Boolean query. With the exception of term substitution, as described in Section 4, SORTEd implements the query editing techniques described in this paper.

1.3 Contribution

This paper describes a method by which the user explicitly edits a conjunctive Boolean query, subject to the constraint that the edited query must have a non-empty result set. To respect this constraint, the addition of a disjunctive term entails the removal of one or more terms previously entered by the user.

[1] Square brackets are used throughout this paper to indicate that a sentence is true both when read without the bracketed terms and when read with each bracketed term substituted for the term which precedes it.

Here we describe the term most recently entered by the user as *disjunctive* if its addition to the existing conjunctive Boolean query would return an empty result set and *conjunctive* otherwise. The user is alerted upon entry of a disjunctive term, and prompted to choose, from amongst automatically generated options, which term(s) should be removed. Removal of the disjunctive term is one of those options. An FCA-based technique is described for identifying those options which minimise the number of user-specified terms removed.

This approach to refining conjunctive Boolean queries is compared and contrasted with the explicit structural navigation of the concept lattice digraph. In particular, it demonstrates that: special treatment of closure terms is appropriate; not all neighbours of the query concept are reachable through the addition or removal of a single user-specified query term; and navigation is not constrained to neighbours.

1.4 Organisation

This paper is organised as follows. Section 2 commences with a brief introduction to the application of Formal Concept Analysis for interactive query refinement. It describes a query editing approach to interactive query refinement, comparing and contrasting it with structural navigation of the lattice digraph. Section 3 canvasses the computational implications of the query editing approach. A novel scheme is then described in Section 4 for computing term substitution options upon user entry of a disjunctive query term. Following a discussion in Section 5 of related work, the contributions of this paper are summarised in Section 6.

2 FCA for Interactive Query Refinement

2.1 Introduction

Formal Concept Analysis (FCA) derives multiple-inheritance class hierarchies from empirical data, and is commonly applied to information retrieval (see e.g. [6] for a recent survey). For classical information retrieval, each class or *concept* is a set of documents and a set of terms such that each document contains each of the terms [3]. A concept is maximal in the sense that there are no other documents containing all of the specified terms, and no other terms which are common to all of those documents. A concept is a *sub-concept* [*super-concept*] of another if it corresponds to a proper subset [superset] of its documents – or equivalently to a proper superset [subset] of its terms. The term "multiple-inheritance" refers to the fact that a concept can be a sub-concept of two or more others, between which a sub-concept relationship does not exist.

For a given document corpus, the corresponding set of concepts, partially ordered by the sub-concept relation between them, forms a *complete lattice*. This lattice can be represented as a directed graph – or *digraph* – whose vertices are the concepts and whose directed arcs are from concepts to their upper neighbours. A super-concept of a given concept is an *upper neighbour* if none of its sub-concepts are super-concepts of the given concept. Layered drawings of this digraph exist in which all arcs of the digraph are upwards, allowing arrows to be omitted from the arcs. The resultant graph drawing is known as a *Hasse diagram* (see e.g. [10]). Figure 1b shows the Hasse diagram corresponding to the formal context in Figure 1a.

At the top of the lattice digraph is the (sole) sink vertex, representing the *supremum* concept, which corresponds to the entire corpus and any terms common to all documents; at the bottom of the lattice digraph is the (sole) source vertex, representing the *infimum* concept, which corresponds to the set of all terms in the corpus and any documents which contain all of them. Each graph vertex is labelled with any term(s) for which the extent of the corresponding concept is exactly the set of documents which contain it, and with any document identifier(s) for which the intent is exactly the set of terms in that document. A concept is an attribute [object] concept for each attribute [object] with which the corresponding vertex is labelled.

2.2 Closure Terms

In corpus-based information retrieval applications of FCA, the user starts by specifying a set of terms which is to be used as a conjunctive query. The result set – the set of documents containing all of those terms – is easily determined, along with any additional terms which those documents have in common. To distinguish these additional terms from the query terms, we refer to them as *closure terms*. Closure terms, if any, are identified automatically and provide additional information about the result set – and indeed the corpus – which the user acquires without reading the constituent documents. Codocedo and Napoli [6] refer to the addition of closure terms as query "extension", and note the "exhaustive" exploitation of closure terms to provide feedback to the user on the corpus-specific context of their query.

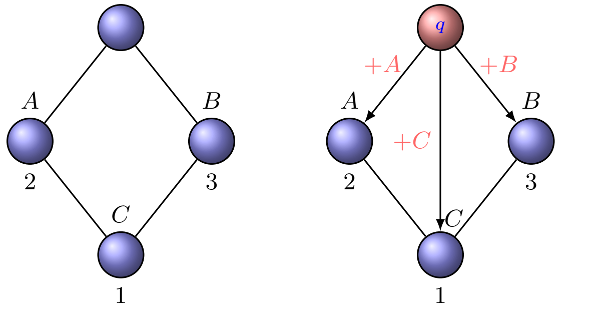
While closure terms are exposed to the user in SORTED, however, they are quarantined from user interaction during subsequent query editing, since the result set is unaffected by either: explicitly extending the user-specified query with closure terms; or removing closure terms from a conjunctive Boolean query which has been extended in this manner. The user must understand the definition of closure terms to properly interpret not only the information they impart about the corpus and the result set, but also their different treatment and behaviour in the user interface.

2.3 Query Refinement

Having specified a query which has a non-empty result set, the user can specialise or generalise the query as required. Query generalisation [specialisation] is typically implemented as the explicit selection of a super- [sub-] concept in the concept lattice, potentially via upward [downward] structural navigation. Some authors [3,14] refer to navigation to upper and lower neighbours as query "enlargement" and "refinement" respectively.

	A	B	C
1	✓	✓	✓
2	✓		
3		✓	

(a) Context



(b) Hasse diagram

(c) Specialisation options

Fig. 1: An example formal context, its corresponding Hasse diagram, and options for specialisation from query concept q.

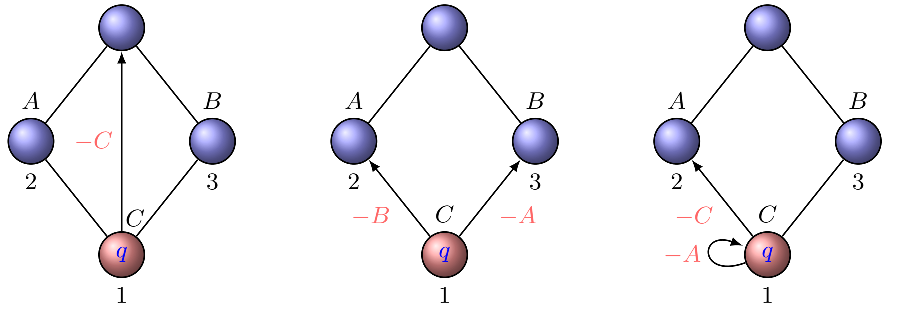
Alternatively, individual terms can be manually added to, or removed from, the query [12]. Query generalisation is achieved by removing one or more of the specified query terms. Query specialisation, on the other hand, involves adding conjunctive terms – those which occur in proper[2], non-empty subsets of the current result set of documents. In addition to requiring the user to select only from query terms in the case of generalisation, and only from conjunctive terms in the case of specialisation, the SORTeD user interface provides advanced knowledge of the effect each choice would have on the size of the result set. Constraining the choice of additional terms to conjunctive terms during query refinement also provides the user with explicit information about the terms and term combinations occurring in the corpus, which they would otherwise need to infer from failed queries.

2.4 Contrast with Structural Navigation

Section 2.3 described the interactive refinement of a conjunctive Boolean query through explicit editing of the query. Like explicit structural navigation of the concept lattice digraph, this approach to query refinement constrains the user's choice to the set of super- and sub- concepts of the current concept. Nevertheless, there are significant differences between the behaviour of these two approaches, and these are explained in this section.

The manual addition [removal] of a single term implicitly selects a sub- [super-] concept – but not necessarily a neighbour – of the current concept. The following example illustrates this point. Consider a corpus consisting of three documents $\{1, 2, 3\}$ containing combinations of three terms $\{A, B, C\}$. Its formal context is shown in Figure 1a and the corresponding Hasse diagram in Figure 1b. Starting at the supremum, as shown in Figure 1c, the user specifies

[2] this excludes both current query and closure terms, which occur in an improper subset.



(a) C

(b) AB

(c) AC

Fig. 2: Generalisation options for three query strings which select the infimum in Figure 1b. This query concept, q, is shown red. Directed edges from q, labelled with the attribute to be removed, enumerate the query generalisation options.

term A, B or C as their first search term. Term A selects the left lower neighbour of the supremum, since there is a document in the corpus containing only that term. Similarly, term B selects the right lower neighbour. However, if the user enters C as the first search term, the infimum is selected, since the only document in the corpus which contains term C also contains the (closure) terms A and B. Here the addition of a search term to the (empty) query implicitly navigates to a sub-concept of the current concept which is not a lower neighbour. As illustrated using the directed edge labelled "-C" in Figure 2a, its subsequent removal returns directly to the supremum, which is a super-concept but not upper neighbour of the infimum.

A given concept in the lattice can be reached using different sequences of query terms. For the example context of Figure 1, the infimum can be reached not only with the query C, but also with the queries AB, AC, BA and BC. Query sequences such as CA and ABC are not possible, since the last or last few terms are rendered closure terms – and therefore unavailable for subsequent user entry – by the entry of earlier terms in the sequence. In the case of CA, for example, the query C selects the infimum, at which point A becomes a closure term whose entry by the user would be nugatory.

Since only the query (vice closure) terms are available for removal, the query sequence used clearly affects the subsequent options for generalisation. Figure 2 illustrates the options for query generalisation for three of these queries. Figure 2b shows that for the query AB, either of the upper neighbours of the infimum is reachable by removing one of the search terms. In contrast, Figure 2c shows that the right-most concept cannot be reached by removing either term from the query AC. Whereas removing C from the search AC selects the left-most concept, removing A instead results in no change in the selected concept. In this case, A simply changes category from a user-specified query term to a closure term. Such nugatory interaction could be prevented by removing A from the set of terms which can be removed until such time as C has been removed.

A query editing approach to query refinement was described in Section 2.3. In this subsection its behaviour has been compared and contrasted with structural navigation of the concept lattice digraph. In particular, while all lower neighbours are reachable in the next move, the query used to reach the current concept may render some upper neighbours unreachable in the user's next move. Furthermore the concepts reachable in the next move may include super- [sub-] concepts which are not upper [lower] neighbours.

3 Computation for Query Editing

The query editing scheme described in Section 2.3 involves the addition or removal of a single user-specified query term. In order to inform the user of the anticipated effect on the size of the result set, and in the case of query specialisation, to constrain the choice of additional terms to those which produce a non-empty result set, it is necessary to calculate the set of concepts which can be reached in the user's next move.

A range of algorithms exist for the enumeration of concepts in a formal context (see e.g. [11,1]), and these can be readily modified to enumerate only those concepts which can be reached from the current concept by adding or removing a single query term. Reachable sub-concepts can be calculated by adding to the intent of the current concept each unused term in turn and calculating the result set. Here, an "unused" term is one which is not already in the intent of the current concept. Terms giving rise to an empty result set are candidates for term substitution, a technique for which is described in Section 4. Similarly, the reachable super-concepts can be calculated by removing each term in turn from the set of user-specified terms, calculating the result set, and optionally disallowing the removal of any terms for which the result set is not a proper superset of the extent of the current concept.

The on-demand computation scheme described in this section has the benefit that only those concepts required to inform the user's next move are computed at each step. The required set of concepts must be computed in interactive timescales, so that the user is not required to wait before contemplating their next move. Given that the computation is triggered by the user's move, however, and the user must assess the consequences of that move – viz. the intent and extent of the new query concept – before contemplating the next, this requirement is not onerous. The computational complexity of enumerating the reachable sub-concepts is $\mathcal{O}(t^2 d)$ Boolean AND operations, where t is the number of unique terms in the formal context and d is the number of documents. Whilst this has the potential to become prohibitive for large corpora, the computation is highly parallelisable.

The set of concepts reachable in the next move could be stored in case subsequent changes to the query return to the same concept. However, since the set of reachable super-concepts is dependent not only on the current concept, but also on the query used to reach it, more than one set of reachable super-concepts may need to be stored for a given concept.

4 Substituting Disjunctive Terms

Let the partially ordered set $\langle \mathfrak{B}, < \rangle$ be the concept lattice corresponding to a formal context constructed from the terms and documents of a nominated corpus. The elements of $\mathfrak{B}$ are the formal concepts and the relation $<$ between concepts corresponds to set inclusion between concept extents. $\langle \mathfrak{B}, < \rangle$ is a complete lattice, for which the greatest lower and least upper bounds, $\inf \mathfrak{S}$ and $\sup \mathfrak{S}$, respectively, on any $\mathfrak{S} \subseteq \mathfrak{B}$ exist and are unique [9].

Definition 1 *Let* $\mathfrak{X}(\mathfrak{S}) \triangleq \{x \in \mathfrak{B} | x \leq z, \forall z \in \mathfrak{S}\}$*. Then* $\inf \mathfrak{S} \in \mathfrak{X}(\mathfrak{S})$ *and* $\inf \mathfrak{S} \geq x, \forall x \in \mathfrak{X}(\mathfrak{S})$*.*

Definition 2 *Let* $\mathfrak{Y}(\mathfrak{S}) \triangleq \{y \in \mathfrak{B} | y \geq z, \forall z \in \mathfrak{S}\}$*. Then* $\sup \mathfrak{S} \in \mathfrak{Y}(\mathfrak{S})$ *and* $\sup \mathfrak{S} \leq y, \forall y \in \mathfrak{Y}(\mathfrak{S})$*.*

Let $q \in \mathfrak{B}$ denote the current query concept, $\alpha \in \mathfrak{B}$ an ancestor of $q < \alpha$, and $t \in \mathfrak{B}$ the attribute concept of a nominated disjunctive term. By definition, a disjunctive term is one which is not in the intent of q or any of its descendants, so that

$$q \not< t \tag{1a}$$

$$\inf \{q, t\} = \inf \mathfrak{B} \tag{1b}$$

Assume the existence of a procedure which identifies all distinct pairs (α', ω) such that

$$\alpha > q \tag{2a}$$

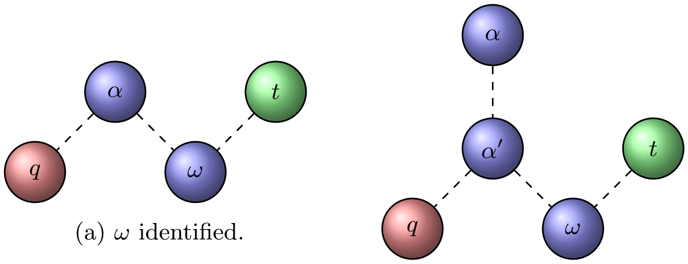
$$\omega \triangleq \inf\{\alpha, t\} \in \mathfrak{B} \tag{2b}$$

$$\alpha' \triangleq \sup\{\omega, q\} \in \mathfrak{B} \tag{2c}$$

Figure 3a shows the Hasse diagram for the poset $\langle \{q, \alpha, \omega, t\}, < \rangle$. The edges are shown dashed to indicate that they need not correspond to neighbour relations in the Hasse diagram for $\langle \mathfrak{B}, < \rangle$. From Equation 2b and Definition 1, $\omega \in \mathfrak{X}(\{\alpha, t\})$, and for any $x \in \mathfrak{X}(\{\alpha, t\})$, $x \leq \omega$. The intent of any $x < \omega$ is a superset of that of ω, and hence differs from those of α and t by more terms. In particular, Equation 2b ensures that the new query, ω, differs from $\alpha > q$ by the smallest number of (added) terms consistent with the constraint $\omega \in \mathfrak{X}(\{\alpha, t\})$. We note in passing that Equation 1a eliminates the possibility that $\omega = \alpha$.

Figure 3b shows the Hasse diagram for the poset $\langle \{q, \alpha, \alpha', \omega, t\}, < \rangle$, which differs from that in Figure 3a by the addition of α'. From Equation 2c and Definition 2, $\alpha' \in \mathfrak{Y}(\{q, \omega\})$ and for any $y \in \mathfrak{Y}(\{q, \omega\})$, $y \geq \alpha'$. From Equations 2a and 2c and Definition 2, $\alpha \in \mathfrak{Y}(\{q, \omega\})$ and hence

$$\alpha \geq \alpha' \tag{3}$$



(a) ω identified.

(b) α' identified.

Fig. 3: Hasse diagrams showing the ordering of concepts q, α, t, ω and α'.

Equation 2c ensures that α' differs from q by the smallest number of (removed) terms, and from ω by the smallest number of (added) terms, both consistent with the constraints $\alpha' \in \mathfrak{Y}(\{q, \omega\})$.

A procedure for enumerating all pairs satisfying Equation 2 is as follows. For each $\alpha > q$ encountered during an upwards, breadth-first traversal of $\langle \mathfrak{B}, < \rangle$ from q, find ω and then α' using Equations 2b and 2c. If $\alpha' \neq \alpha$, discard (α', ω). The upwards, breadth-first traversal ensures that the algorithm encounters α' as a candidate ancestor of q before – or strictly speaking not later than – $\alpha \geq \alpha'$. By an upwards, breadth-first traversal, we mean that all upper neighbours of a concept are visited before any more distant ancestors, *and* that a vertex is not visited until each of its lower neighbours has been. The latter requirement is necessitated by the possibility that the lattice digraph may contain parallel directed paths of unequal path length. If ω has been previously encountered, and hence also α' from Equation 2c, the pair (α', ω) has already been generated, and traversal can progress immediately to the next $\alpha > q$.

Proposition 1.

$$\omega = \inf \{\alpha', t\} \tag{4}$$

Proof. Define

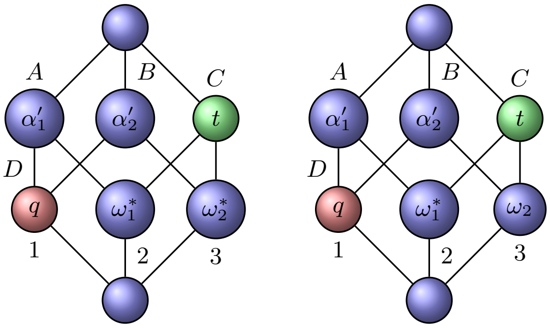
$$\omega' \triangleq \inf \{\alpha', t\} \tag{5}$$

Substituting Equation 3 into Definition 1 yields $\omega' \leq t$ and $\omega' \leq \alpha' \leq \alpha$, to which Definition 1 can be reapplied to give $\omega' \leq \inf \{\alpha, t\} = \omega$. But $\omega \leq \alpha'$ from Equation 2c and $\omega \leq t$ from Equation 2b, so $\omega \leq \inf \{\alpha', t\} = \omega'$. Hence $\omega' = \omega$.

Proposition 1 shows that evaluating the right-hand side of Equation 5 in the hope of finding a tighter lower bound ω' on α' and t is nugatory. Subject to the initial choice of α, the pair (α', ω) minimises the number of terms removed from and added to the current query q to include the disjunctive term for which t is the attribute concept. Having enumerated the possible choices for $\alpha > q$ to identify all unique pairs (α', ω), the problem reduces to choosing from amongst these the ω differing from q by the smallest number of terms. The cardinality of the set

	A	B	C	D
1	✓	✓		✓
2	✓		✓	
3		✓	✓	

(a) Context



(b) Query AB (c) Query AD

Fig. 4: Substitution of term C for queries AB and AD in example context.

difference between the intents of q and α' gives the number of terms removed from the existing query, while that of the set difference between the intents of ω and α' gives the number of terms added to it. The most straightforward approach is to minimise the sum of these two numbers. If preserving terms of the current query is a priority, then a weighted sum might be used in which the number of removed terms is weighted more highly than the number of added terms. The possible choices for the new query ω could be presented to the user ranked in order of increasing (weighted) term difference.

Applying this approach to the example context in Figure 4a with query AB yields the two jointly-optimal solutions ω^*_1 and ω^*_2 shown in Figure 4b, corresponding to ancestors α'_1 and α'_2, respectively, of the query concept q (shown red). Each involves removing one query term and adding the (formerly) disjunctive term C, for which the attribute concept t is shown green. The user might be asked to choose between them, possibly aided by additional information such as extent cardinality, or the semantic relevance of the additional terms.

In this section, we have so far ignored the distinction between user-specified and closure terms. This distinction should be taken into account to privelege options minimising the number of user-specified terms which must be removed to accommodate the new term. If, for example, the user-specified query against the formal context of Figure 4a were AD, then the removal of closure term B from the intent of the query concept should be preferred over that of query term A. Descendant ω^*_1 of α'_1 is therefore preferred over ω_2 of α'_2, as shown in Figure 4c, even though both involve the removal of two terms from the query concept.

Like conjunctive terms, disjunctive terms should be offered in a progressively-constrained list for addition to the query. Upon user selection of a disjunctive term, options would be presented showing which query (vice closure) terms must be removed to accommodate the new term. The terms to be removed might be shown struck out, and the terms to be added – other than the nominated disjunctive term – highlighted, but otherwise treated as closure terms.

5 Discussion and Related Work

If the user is not aided in their initial choice of query terms, it is not possible to guarantee that each term occurs in the corpus and that at least one document contains all terms. The query editing scheme described in this paper is therefore used *ab initio*, with the initial query being the set of terms associated with the supremum. In contrast, Carpineto and Romano [2] permitted unconstrained entry of an intial set of query terms, and described a composite method whereby the resultant query could be mapped onto the concept lattice. *Ab initio* computer-assisted query refinement requires the initial enumeration of all attribute concepts in the formal context and user interaction with the list of all terms in the corpus. Manual entry of terms, aided by term completion, partially alleviates the challenge posed by large corpora of finding terms in a long list.

6 Conclusion

This paper has described a method by which the user explicitly edits a conjunctive Boolean query, subject to the constraint that the edited query has a non-empty result set. Adoption of this query-editing approach, and in particular its use of lists to present options for term addition and removal, has allowed the SORTED prototype to offer the user ranked options for interactive query refinement. The addition of a disjunctive term entails the removal of one or more terms previously entered by the user. The user is alerted upon entry of a disjunctive term, and prompted to choose, from amongst automatically generated options, which term(s) should be removed. An FCA-based technique has been described for automatically identifying options minimising the number of user-specified terms removed.

This approach to refining conjunctive Boolean queries has been compared and contrasted with the explicit structural navigation of the concept lattice digraph. In particular: special treatment of closure terms is appropriate; not all neighbours of the query concept are reachable through the addition or removal of a single user-specified query term; and navigation is not constrained to neighbours.

References

1. Andrews, S.: A 'best-of-breed' approach for designing a fast algorithm for computing fixpoints of Galois connections. Information Sciences **295**, 633–649 (2015). https://doi.org/10.1016/j.ins.2014.10.011
2. Carpineto, C., Romano, G.: ULYSSES: a lattice-based multiple interaction strategy retrieval interface. In: Blumenthal, B., Gornostaev, J., Unger, C. (eds.) Human-Computer Interaction (EWHCI 1995). pp. 91–104. No. 1015 in LNCS, Springer, Berlin, Heidelberg (1995). https://doi.org/10.1007/3-540-60614-9_7
3. Carpineto, C., Romano, G.: Exploiting the potential of concept lattices for information retrieval with CREDO. Journal of Universal Computing **10**(8), 985–1013 (2004). https://doi.org/10.3217/jucs-010-08-0985

4. Carpineto, C., Romano, G.: Effective reformulation of Boolean queries with concept lattices. In: Andreasen, T., Christiansen, H., Larsen, H. (eds.) Flexible Query-Answering Systems (FQAS 1995). LNCS, vol. 1495, pp. 83–94. Springer, Berlin, Heidelberg (1998). https://doi.org/10.1007/BFb0055993
5. Codocedo, V., Lykourentzou, I., Napoli, A.: A semantic approach to concept lattice-based information retrieval. Annals of Mathematics and Artificial Intelligence **72**(1-2), 169–195 (2014). https://doi.org/10.1007/s10472-014-9403-0
6. Codocedo, V., Napoli, A.: Formal Concept Analysis and Information Retrieval – a survey. In: Baixeries, J., Sacarea, C., Ojeda-Aciego, M. (eds.) Formal Concept Analysis, Lecture Notes in Computer Science, vol. 9113, pp. 61–77. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-19545-2_4
7. Eklund, P., Wray, T., Goodall, P., Bunt, B., Lawson, A., Christidis, L., Daniels, V., Van Olffen, M.: Designing the digital ecosystem of the Virtual Museum of the Pacific. In: 3rd IEEE Intl. Conf. Digital Ecosystems and Technologies. pp. 377–383. IEEE (2009). https://doi.org/10.1109/DEST.2009.5276744
8. Eklund, P.W., Ducrou, J., Brawn, P.: Concept lattices for information visualization: Can novices read line diagrams? In: Eklund, P. (ed.) Proc. 2nd Intl. Conf. on Formal Concept Analysis (FCA'04). LNCS, vol. 2961, pp. 57–73. Springer, Berlin, Heidelberg (2004). https://doi.org/10.1007/978-3-540-24651-0_7
9. Ganter, B., Wille, R.: Formal Concept Analysis: Mathematical Foundations. Springer, Berlin, Heidelberg (1999). https://doi.org/10.1007/978-3-642-59830-2
10. Gross, J.L., Yellen, J., Zhang, P. (eds.): Handbook of Graph Theory. Chapman and Hall/CRC Press, New York, second edn. (2013)
11. Kuznetsov, S.O., Poelmans, J.: Knowledge representation and processing with Formal Concept Analysis. Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery **3**(3), 200–215 (2013). https://doi.org/10.1002/widm.1088
12. Lindig, C.: Concept-based component retrieval. In: Working notes of the IJCAI-95 workshop: Formal Approaches to the Reuse of Plans, Proofs, and Programs. pp. 21–25. Montreal, Canada (1995)
13. Pattison, T.: Interactive visualisation of formal concept lattices. In: Burton, J., Stapleton, G., Klein, K. (eds.) Joint Proc. 4th Intl. Workshop on Euler Diagrams (ED 2014) and the 1st Intl. Workshop on Graph Visualization in Practice (GViP 2014). CEUR Workshop Proceedings, vol. 1244, pp. 78–79 (Jul 28 – Aug 1 2014), `http://ceur-ws.org/Vol-1244/GViP-paper6.pdf`
14. Poelmans, J., Ignatov, D.I., Viaene, S., Dedene, G., Kuznetsov, S.O.: Text mining scientific papers: A survey on FCA-based information retrieval research. In: Perner, P. (ed.) Advances in Data Mining: Applications and Theoretical Aspects (ICDM 2012). pp. 273–287. Springer, Berlin, Heidelberg (2012). https://doi.org/10.1007/978-3-642-31488-9_22
15. Priss, U.: Formal Concept Analysis in Information Science. Annual Review of Information Science and Technology **40**, 521–543 (2006). https://doi.org/10.1002/aris.1440400120
16. Valverde-Albacete, F.J., Peláez-Moreno, C.: Systems vs. methods: an analysis of the affordances of Formal Concept Analysis for information retrieval. In: Carpineto, C., Kuznetsov, S.O., Napoli, A. (eds.) Proc. Workshop Formal Concept Analysis Meets Information Retrieval (FCAIR 2013). vol. 977 (2013), `http://ceur-ws.org/Vol-977/paper13.pdf`

Efficient Formal Concept Analysis through Recursive Context Partitioning

Tim Pattison, Aaron Ceglar, and Derek Weber

{tim.pattison,aaron.ceglar,derek.weber}@dst.defence.gov.au
Defence Science & Technology Group
West Ave, Edinburgh
South Australia 5111

Abstract. Formal Concept Analysis takes as input a bigraph known as a formal context. It produces a partially-ordered set of formal concepts which constitutes a complete lattice. This lattice can be represented as a directed acyclic graph, whose vertices are formal concepts and whose arcs connect neighbours in the ordering relation between them. This paper describes a divide-and-conquer technique for discovering and exploiting hierarchical structure in a formal context. Simultaneous hierarchical partitioning of both the context bigraph and the resultant lattice digraph is used to achieve efficient computation and, elsewhere, interactive visualisation of the concept lattice.

1 Introduction

Formal Concept Analysis (FCA) derives a multiple-inheritance class hierarchy from a formal context. A formal context consists of a set of objects, a set of attributes, and a binary relation between them. The classes derived by FCA are known as formal concepts. Each consists of a set of objects, called its extent, and a set of attributes, called its intent, such that each object in its extent, and no others, has all attributes in its intent. The set of formal concepts, when partially ordered by set inclusion, forms a complete lattice. This lattice can be efficiently represented as a directed acyclic graph (DAG), whose vertices are formal concepts, and whose adjacency relation is the transitive reduction of the ordering relation.

The number of formal concepts is bounded above by an exponential function of the number of objects and attributes in the context. Scaling FCA to the interactive analysis of large data sets poses two fundamental challenges: the time required to compute the concepts and construct the large lattice digraph; and the difficulty of meaningful and responsive user interaction with this digraph.

The times required to enumerate all formal concepts of a formal context, and to calculate the transitive reduction of the ordering relation between them, are both bounded above by a polynomial function of the number of formal concepts. One class of "divide and conquer" techniques tackles this inherent computational complexity by partitioning the context, performing FCA on each resultant sub-context, and combining the results. FCA is thereby mapped onto multiple independent processors, each performing FCA on a sub-context which is significantly

© Dmitry I. Ignatov, Lhouari Nourine (Eds.): CLA 2018, pp. 219–230,
ISBN 978–80–244–5328–6, Department of Computer Science, Palacký University
Olomouc, 2018.

smaller than the overall context. Several FCA algorithms use this approach for the enumeration of concepts [7,2] and construction of the lattice digraph [3,11].

This paper describes the Carve technique for recursively partitioning a formal context, which produces a corresponding hierarchical partition of the lattice digraph. Carve exploits structure which we have identified in an empirical author-publication context for co-authorship analysis, and which Bhatti et al. [4] found in software systems. We explain how recognising this structure leads to a novel divide-and-conquer algorithm for efficient FCA. The resultant hierarchical partitioning of the lattice digraph is exploited elsewhere for both layout of, and interaction with, the Hasse diagram [10].

Carve recursively partitions a formal context for analysis by *any* FCA algorithm which, or algorithms which collectively: enumerates the formal concepts; calculates the transitive reduction of the ordering relation; and returns the corresponding labelled DAG. It further assembles the resultant digraphs into that for the original context. Carve does not compete with existing FCA algorithms; it accelerates their application to formal contexts exhibiting the requisite structure.

This paper is organised as follows. Section 2 introduces relevant aspects of the theories of partial orders, FCA, graphs and graph drawing. Section 3 then establishes the theoretical foundations for the Carve algorithm, after which the algorithm is detailed in Section 4. A brief discussion of related work is presented in Section 5, followed by a summary in Section 6.

2 Preliminaries

This section introduces relevant aspects of the theories of partial orders and FCA, as well as graphs and graph drawing. It assumes relevant knowledge found in such textbooks as [5] and [8] respectively.

Definition 1 *A* formal context (G, M, I) *is a triple consisting of a set G of objects, a set M of attributes and a binary relation $I \subseteq G \times M$ such that an object $g \in G$ has attribute $m \in M$ iff $(g, m) \in I$.*

Definition 2 *The context bigraph for the formal context (G, M, I) is a bipartite graph having object vertex set G, attribute vertex set M, and edge set I.*

Definition 3 *A* connected component *of a context bigraph is a maximal subgraph in which a path exists between all pairs of vertices.*

The example formal context of Figure 1a is represented in Figure 1b as a context bigraph. Object vertices are drawn with grey fill and attribute vertices white. The context bigraph has three connected components.

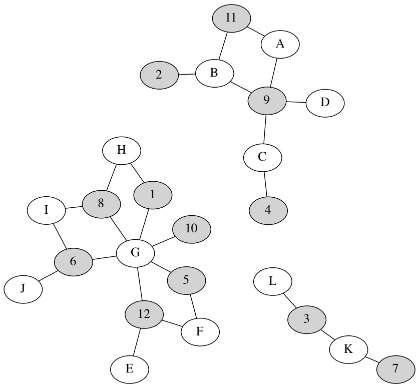
Definition 4 *The extent operator $^{\lhd} : \mathcal{P}(M) \to \mathcal{P}(G)$ and intent operator $^{\rhd} : \mathcal{P}(G) \to \mathcal{P}(M)$ return the maximal sets*

$$B^{\lhd} = \{g \in G \mid (g, m) \in I \;\; \forall\, m \in B\} \tag{1a}$$

$$A^{\rhd} = \{m \in M \mid (g, m) \in I \;\; \forall\, g \in A\} \tag{1b}$$

	A	B	C	D	E	F	G	H	I	J	K	L
1							✓	✓				
2		✓										
3											✓	✓
4			✓									
5						✓	✓					
6							✓		✓	✓		
7											✓	
8							✓	✓	✓			
9	✓	✓	✓	✓								
10							✓					
11	✓	✓										
12					✓	✓	✓					

(a) Formal context



(b) Context bigraph

Fig. 1: An example formal context and its corresponding bigraph.

of objects possessing all attributes in $B \subseteq M$ and attributes possessed by all objects in $A \subseteq G$.

Here, $\mathcal{P}(G)$ and $\mathcal{P}(M)$ denote the powersets – sets of all subsets – of G and M respectively. $m^{\triangleleft} \equiv \{m\}^{\triangleleft}$ is the set of objects which have attribute m and $g^{\triangleright} \equiv \{g\}^{\triangleright}$ is the set of attributes possessed by object g.

Proposition 1. *For $A \subseteq G$ and $B \subseteq M$*

$$B^{\triangleleft} = \{g \in G \mid g^{\triangleright} \supseteq B\} \tag{2a}$$

$$A^{\triangleright} = \{m \in M \mid m^{\triangleleft} \supseteq A\} \tag{2b}$$

Definition 5 *A* formal concept *of the context (G, M, I) is an ordered pair (A, B) with extent $\emptyset \subseteq A \subseteq G$ and intent $\emptyset \subseteq B \subseteq M$ satisfying*

$$A^{\triangleright} = B \tag{3a}$$

$$B^{\triangleleft} = A \tag{3b}$$

Definition 6 *A* biclique *(A, B) in the context bigraph (G, M, I) is a set $\emptyset \subset A \subseteq G$ of object vertices and a corresponding set $\emptyset \subset B \subseteq M$ of attribute vertices such that each vertex in A is adjacent to all vertices in B, and vice versa.*

Definition 7 *A biclique (A, B) is* maximal *if there is no biclique $(A', B') \neq (A, B)$ having $A' \supseteq A$ and $B' \supseteq B$.*

Proposition 2 (Gaume et el. [6]). *(A, B) is a formal concept in the formal context (G, M, I) having extent $A \neq \emptyset$ and intent $B \neq \emptyset$ iff (A, B) is a maximal biclique in the bigraph (G, M, I) having object vertex set A and attribute vertex set B.*

The elements of the set $\mathfrak{B}(G, M, I)$ of formal concepts are partially ordered by defining the relation $\leq$ between concepts $(A, B), (C, D) \in \mathfrak{B}(G, M, I)$ such that

$$A \subseteq C \iff (A, B) \leq (C, D) \iff B \supseteq D \tag{4}$$

Definition 8 *Let $\langle \mathfrak{P}; \leq \rangle$ be a finite partially-ordered set, and let $a, b \in \mathfrak{P}$ satisfy $a < b$ – i.e. $a \leq b$ and $a \neq b$. If $\nexists c \in \mathfrak{P}$ satisfying $a < c < b$, then we write $a \prec b$. Element a is then called a* lower neighbour *of b, and conversely b is an* upper neighbour *of a.*

Proposition 3. *FCA of the formal context (G, M, I) produces the lattice*

$$\langle \mathfrak{B}(G, M, I); \leq \rangle = \left\langle \mathfrak{C} \cup \bigvee_{c \in \mathfrak{C}} c \cup \bigwedge_{c \in \mathfrak{C}} c; \leq \right\rangle \tag{5}$$

where c ranges over the set $\mathfrak{C}$ of maximal bicliques in (G, M, I).

Thus the global supremum $\vee_{c \in \mathfrak{C}} c$ and infimum $\wedge_{c \in \mathfrak{C}} c$ of the set $\mathfrak{C}$ of maximal bicliques complete the set $\mathfrak{B} \supseteq \mathfrak{C}$ of formal concepts in cases where they have empty intent or extent, and are therefore not already included in $\mathfrak{C}$. We refer to the supremum and infimum collectively as the extrema.

The complete lattice $\langle \mathfrak{B}(G, M, I); \leq \rangle$ can be efficiently represented as a labeled digraph whose vertices are concepts and whose arcs connect neighbours. Each arc is directed from the lower to the upper neighbour, and concepts are comparable iff there exists a directed path between the corresponding vertices. For each concept (A, B), the corresponding vertex is labelled with attribute set $\mu \subseteq B$ and object set $\gamma \subseteq A$ defined as follows:

$$\mu := \{m \in M | m^{\triangleleft} = A\} \tag{6a}$$

$$\gamma := \{g \in G | g^{\triangleright} = B\} \tag{6b}$$

Definition 9 *The concept (A, B) is an* attribute concept *for each attribute $m \in \mu$ and an* object concept *for each object $g \in \gamma$.*

The resultant directed graph is a single-source, single-sink DAG. The source vertex `s`, corresponding to the infimum, has only outgoing arcs; the sink vertex `t`, corresponding to the supremum, has only incoming arcs. The extent [intent][1] of any concept is the set of all object [attribute] labels encountered on downward [upward] paths from that concept. Two lattices are order isomorphic iff their lattice digraphs, excluding labels, are isomorphic.

Proposition 4. *Let `t` and `s` be the sink and source vertices, respectively, of a lattice digraph. Denote by μ_t and γ_t the attribute and object label sets of `t`, and by μ_s and γ_s the attribute and object label sets of `s`. Then*

$$\begin{aligned} \mu_t &= G^{\triangleright} & \mu_s &= \{m \in M | m^{\triangleleft} = M^{\triangleleft}\} \\ \gamma_t &= \{g \in G | g^{\triangleright} = G^{\triangleright}\} & \gamma_s &= M^{\triangleleft} \end{aligned} \tag{7}$$

[1] Square brackets indicate that a sentence is true both when read without the bracketed terms and when read with each bracketed term substituted for the preceding term.

Proposition 4 indicates that t is labelled with attributes possessed by all objects and with objects possessing only these universal attributes. Similarly, s is labelled with objects possessing all attributes and with attributes possessed only by these universal objects.

A Hasse diagram [8] is a two-dimensional spatial embedding of the lattice digraph in which the vertical component of each arc is upwards, and edge directions are accordingly omitted. Attribute [object] labels are placed above [below] the labelled concept. Figure 2b shows the Hasse diagram for the context bigraph in Figure 2a. The labelling indicates, inter alia, that the supremum is an attribute concept for attribute G and an object concept for object 10. The vertex having attribute label set $\mu = \{\mathtt{F}\}$ and object label set $\gamma = \{\mathtt{5}\}$ in Figure 2b corresponds to the concept $(\{\mathtt{5}, \mathtt{12}\}, \{\mathtt{F}, \mathtt{G}\})$; it "inherits" attribute G from its upper neighbour and object 12 from its lower neighbour.

3 Foundations of Recursive Partitioning

3.1 Partitioning the Context Bigraph

Each biclique of the context bigraph must be entirely contained within a single connected component. The maximal bicliques of (G, M, I) can therefore be enumerated by dividing the bigraph into its connected components and enumerating the maximal bicliques of each component. Separate connected components, and hence also their bicliques, share neither object nor attribute vertices.

Proposition 5. *If* $\vee_{c \in \mathfrak{C}} c \in \mathfrak{C}$ *or* $\wedge_{c \in \mathfrak{C}} c \in \mathfrak{C}$ *then the bigraph* (G, M, I) *consists of a single connected component.*

3.2 Partitioning the Lattice Digraph

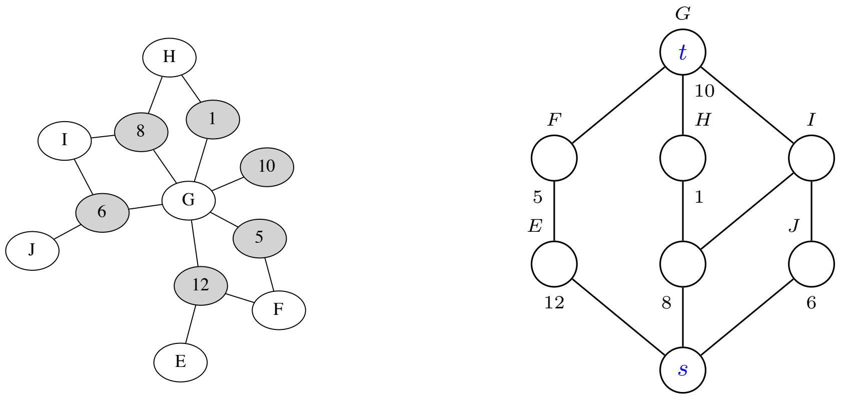
For any two concepts, the intent [extent] of their supremum [infimum] is the intersection of their intents [extents].

Proposition 6. *For any pair of maximal bicliques drawn from different connected components, their extrema are those of the set* $\mathfrak{C}$ *of maximal bicliques.*

Corollary 1 *No arcs exist between vertices of the lattice digraph corresponding to maximal bicliques from different connected components of* (G, M, I).

Corollary 1 indicates that partitioning the context bigraph into connected components partitions the vertices of the resultant lattice digraph, excepting the extrema, into disjoint sets between which there are no arcs. From the converse of Proposition 5, if (G, M, I) consists of more than one connected component, then the extrema are not maximal bicliques, and are not covered by Corollary 1.

Figure 3a uses rounded boxes as containers to illustrate this partitioning of the concepts for the example context in Figure 1. The concepts derived from the largest connected component of the context bigraph – with the exception of the



(a) Bigraph

(b) Hasse diagram

Fig. 2: Bigraph and Hasse diagram for largest connected component of Figure 1b.

infimum which does not correspond to a biclique − are in the pink container. These concepts and the ordering relation between them are the same as those in Figure 2b, which resulted from FCA of the largest component in isolation. As expected, there are no edges (arcs) between concepts in different containers.

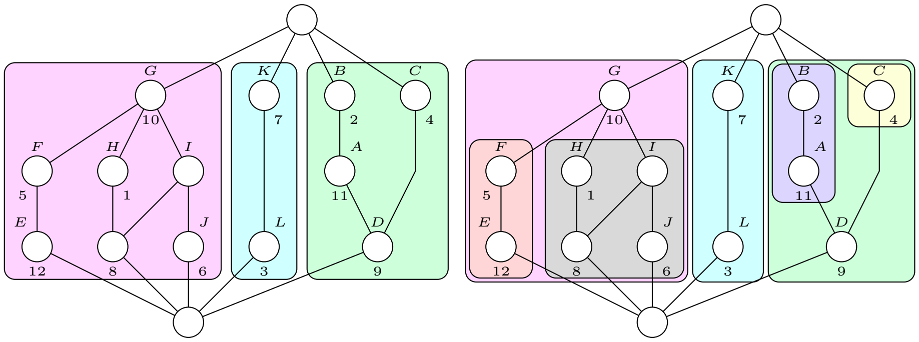
A context bigraph exhibiting two or more connected components can be partitioned into those components, those components analysed separately to identify their maximal bicliques and compute the transitive reduction of their partial ordering, and the sink [source] vertices of each resultant sub-graph of the lattice digraph connected to the global sink [source] to form the lattice digraph. FCA of the context (G, M, I) is thereby reduced to independent FCA for each of the sub-contexts $(G_i, M_i, I \cap (G_i \times M_i))$. Some care is required (see Section 3.5) to ensure the proper handling of the extrema of each sub-context.

3.3 Reducing the Context

Proposition 7. *Let (G', M', I') be formed from (G, M, I) by removing object set γ_s, attribute set μ_t, and all edges incident on either. Then $\langle \mathfrak{B}(G', M', I'); \leq \rangle$ is order isomorphic to $\langle \mathfrak{B}(G, M, I); \leq \rangle$.*

Proposition 8. *Let (G', M', I') be formed from (G, M, I) by removing object set γ_t, attribute set μ_s, and all edges incident on either. Then $\langle \mathfrak{B}(G', M', I'); \leq \rangle$ is order isomorphic to $\langle \mathfrak{B}(G, M, I); \leq \rangle$ iff the supremum and infimum of $\mathfrak{B}(G, M, I)$ have more than one lower and upper neighbour respectively.*

Propositions 7 and 8 involve the application of row and column reduction [5] to the objects and attributes for which the extrema are object and attribute concepts. They allow us to safely remove from the context all fully-connected attributes and objects, and in many cases any objects and attributes which are isolated after this removal.



(a) Initial partitioning of maximal bicliques

(b) Recursive partitioning

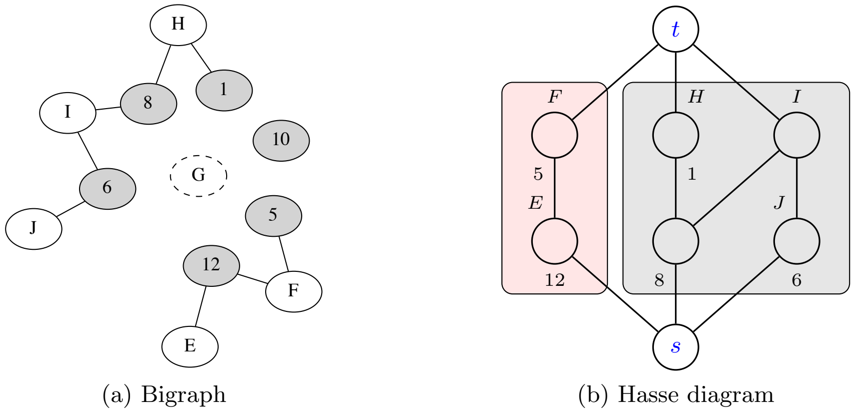
Fig. 3: Hasse diagram for the context in Figure 1 showing partitioning of its maximal bicliques. Figure 3b adds an inclusion tree layout (nested containers) to illustrate the recursive execution of `Carve()`.

Figure 4a shows the connected components of the bigraph in Figure 2a after removal of the universal attribute `G` and its incident edges. Object `10` is now isolated, and cannot therefore participate in a biclique. Figure 4b shows the Hasse diagram for this sub-context after removal of both `G` and `10`. As indicated by the shaded containers, the maximal bicliques are partitioned into two sets, corresponding to the two remaining connected components of the bigraph, between which there are no connections. Comparison of Figures 4b and 2b confirms that the lattices before and after extremum reduction are order isomorphic. Since the supremum has more than one lower neighbour, removal of object `10` satisfies the condition of Proposition 8 for lattice isomorphism.

Let (G', M', I') denote the result of applying Propositions 7 and 8 to (G, M, I). The extrema of this *extremum-reduced* context are no longer object or attribute concepts, and accordingly have no labels. Whereas the concept lattices for (G, M, I) and (G', M', I') are order isomorphic, the corresponding lattice digraphs will therefore differ in the labelling of the supremum and infimum. Restoring these labels, which are now stored in $\gamma_s, \mu_s, \gamma_t$ and μ_t, converts the lattice digraph for (G', M', I') into that for (G, M, I). For example, restoring the label sets $\gamma_t = \{\texttt{10}\}$ and $\mu_t = \{\texttt{G}\}$ to the supremum converts the Hasse diagram in Figure 4b to that in Figure 2b.

3.4 Recursive Partitioning

We have seen that the removal of any universal attributes and objects, as per Proposition 7, can disconnect the context bigraph, and thereby pave the way for partitioning of (G, M, I). If (G, M, I) is instead a sub-context resulting from a previous partition, removal of these objects and attributes may facilitate partitioning of that sub-context, and hence recursive partitioning of the global context.



(a) Bigraph (b) Hasse diagram

Fig. 4: Removal of attribute G from the bigraph in Figure 2a reveals new connected components and a corresponding partition of its maximal bicliques.

The execution of a procedure which removes from the context bigraph the objects and attributes identified in Propositions 7 and 8, along with their adjacent edges, identifies the connected components in the remaining context bi-graph, and calls itself to process each of the identified components, is described by its recursion tree. Figure 5 shows this tree superimposed on the context bigraph from Figure 1 using an inclusion tree layout. Each rounded rectangular container corresponds to a vertex in the recursion tree. A container includes another iff the former corresponds to an ancestor of the latter in the recursion tree.

This tree also constitutes a partition tree for the context bigraph, in which the non-leaf nodes are labelled with the objects $\gamma_t \cup \gamma_s$ and attributes $\mu_t \cup \mu_s$ associated with the extrema of the corresponding sub-context. In Figure 5, these objects and attributes appear only in non-leaf containers.

The leaf nodes of this tree correspond to sub-contexts whose lattice digraphs are either trivial – consisting of one or two vertices – or otherwise not amenable to further partitioning. Each non-trivial leaf-node sub-context must be processed by a suitable FCA algorithm which returns the corresponding lattice digraph labelled as per (6). The resultant single-source, single-sink digraphs form the

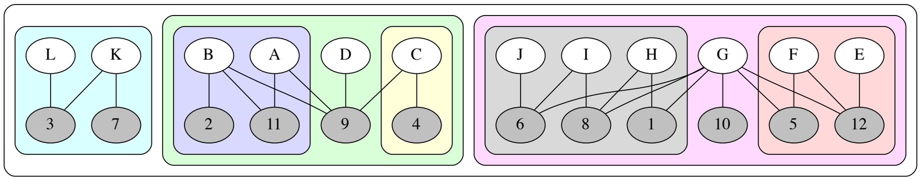


Fig. 5: Context bigraph from Figure 1 with recursion tree superimposed using inclusion tree layout.

building blocks which must then be appropriately assembled and interconnected to produce the lattice digraph for the overall context. Importantly, this assembly can be performed progressively in a single pass back up the recursion tree.

3.5 Assembling the Lattice Digraph

Assembling the lattice digraph for a given sub-context involves either connecting or merging the source (infimum) and sink (supremum) vertices $\mathtt{s}', \mathtt{t}'$ of the digraphs for each of its immediate sub-contexts with their counterparts $\mathtt{s}, \mathtt{t}$ for the current sub-context. In this section we examine the circumstances under which merging and connection, respectively, are appropriate.

Proposition 9. *Let (G', M', I') be an extremum-reduced context consisting of two or more connected components. Let (A, B) be the supremum [infimum] of one of these connected components, (G_i, M_i, I_i), and let t' [s'] be the corresponding vertex of the lattice digraph for that sub-context. Then $(A, B) \in \mathfrak{B}(G', M', I')$ iff $\mu_{t'} \neq \emptyset$ [$\gamma_{s'} \neq \emptyset$].*

If the supremum [infimum] of (G_i, M_i, I_i) has an attribute [object] label, then an arc is added to connect it to its counterpart in the parent context (G', M', I'). This ensures that each maximal biclique in (G_i, M_i, I_i) has a directed path to the lattice supremum, and from the infimum, of the parent context. If the supremum [infimum] of (G_i, M_i, I_i) does not have an attribute [object] label, it should instead be merged with its counterpart in the parent context. The merge operation replaces two digraph vertices with a single vertex having the unions of their upper neighbours, lower neighbours, attribute labels and object labels.

4 Carve

Algorithm 1 lists the Carve algorithm for simultaneous recursive partitioning of a formal context and its corresponding lattice digraph. `Carve()` takes as input a context bigraph (G, M, I) and returns, by reference to its source and sink vertices, the corresponding lattice digraph, labelled as per (6). The Boolean parameter ϕ represents the level of recursion, and should be set to zero for the initial call. `Carve()` invokes the following functions:

`FindComponents`(G', M', I'): Takes as input a context bigraph (G', M', I') and returns an unordered set of its connected component bigraphs.

`FCA`(G', M', I'): Takes as input a formal context (G', M', I') which cannot be further decomposed by `Carve()` and returns the resultant lattice digraph. This function can be implemented using *any* FCA algorithm (see e.g. [9,1]) which enumerates the concepts, calculates the transitive reduction of the ordering relation between them, and calculates the labelling defined in (6).

`Connect`(a, b): Creates an arc from vertex a to vertex b of the lattice digraph.

`Merge`(a, b): Applied to lattice digraph vertices a and b, it modifies vertex b to have label sets and adjacent arcs, both incoming and outgoing, which are the unions of those of a and b. Thus vertex a is merged with vertex b.

Algorithm 1 Build lattice digraph through recursive partitioning of context.

```
Require: $I \subseteq G \times M$
Ensure: $(s, t)$ are (source,sink) vertices of labelled lattice digraph for $(G, M, I)$
Function $(s, t) = $ Carve$(G, M, I, \phi)$
$\mu_t \leftarrow G^{\rhd}$
if $\mu_t = M$ then
    $\gamma_t \leftarrow G$
    return $(t, t)$
end if
if $\mu_t \neq \emptyset$ or $\phi = 0$ then
    $\gamma_t \leftarrow \{g \in G | g^{\rhd} = \mu_t\}$
else
    $\gamma_t \leftarrow \emptyset$
end if
$\gamma_s \leftarrow M^{\lhd}$
if $\gamma_s \neq \emptyset$ or $\phi = 0$ then
    $\mu_s \leftarrow \{m \in M | m^{\lhd} = \gamma_s\}$
else
    $\mu_s \leftarrow \emptyset$
end if
$M' \leftarrow M \setminus (\mu_t \cup \mu_s)$
$G' \leftarrow G \setminus (\gamma_t \cup \gamma_s)$
if $M' = \emptyset$ or $G' = \emptyset$ then
    Connect$(s, t)$
    return $(s, t)$
end if
$I' \leftarrow I \cap (G' \times M')$
if $\mu_t \neq \emptyset$ or $\gamma_s \neq \emptyset$ or $\phi = 0$ then
    $Q \leftarrow$ FindComponents$(G', M', I')$
else
    $Q \leftarrow \{(G, M, I)\}$
end if
if $|Q| = 1$ and $\gamma_t = \emptyset$ and $\mu_s = \emptyset$ then
    $(s', t') \leftarrow$ FCA$(G', M', I')$
    Merge$(t', t)$
    Merge$(s', s)$
else
    for all $(G_i, M_i, I_i) \in Q$ do
        $(s', t') \leftarrow$ Carve$(G_i, M_i, I_i, 1)$
        if $\mu_{t'} \neq \emptyset$ then
            Connect$(t', t)$
        else
            Merge$(t', t)$
        end if
        if $\gamma_{s'} \neq \emptyset$ then
            Connect$(s, s')$
        else
            Merge$(s', s)$
        end if
    end for
end if
return $(s, t)$
```

Lines 2 to 24 of `Carve()` create and label the source and sink vertices of the sub-lattice digraph, return the lattice digraph in cases where it is trivial, and if not, convert the context (G, M, I) to its extremum-reduced form (G', M', I'). Lines 25 to 29 partition (G', M', I') into its connected component bigraphs. Lines 30 to 48 calculate the sub-lattice digraph for each of these components and connect or merge it into the source and sink vertices of the lattice digraph for (G, M, I). From the perspective of `FCA()`, `Carve()` is a pre-processor which reduces the context to an extremum-reduced, connected bigraph, and a post-processor which connects the resultant sub-lattice digraph into that for the original context.

Figure 3b uses an inclusion tree layout to illustrates the recursive execution of `Carve()` when applied to the context of Figure 1. The containers are filled with the same colours as their counterparts in Figure 5, in which the same recursion tree is overlaid on the context bigraph. The single vertex in the yellow container is returned by line 5 of Algorithm 1. Those in the light grey container are generated by a call to `FCA()` at line 31, and those pairs in the remaining three leaf-node containers are returned by line 22.

5 Related Work

Berry et al. [3] proposed a divide-and-conquer approach to FCA based on the identification of a vertex separator known as a clique minimal separator. A clique minimal separator is an attribute or an object, or an attribute-object pair, whose removal disconnects the context bigraph. The vertices of a bigraph having one or more such separators are "partitioned" by notionally removing each separator in turn, and replicating it across each of the adjacent connected components of the resultant bigraph. Berry et al. [3] nominated an algorithm which efficiently identifies these separators and described a method for combining the concept lattices produced by applying FCA to each resultant sub-context.

Decomposition using clique minimal separators produces a set of atomic bigraphs which by definition contain no clique minimal separators. FCA of each atomic bigraph produces corresponding lattice digraphs from which the original lattice digraph is reconstructed. In contrast with Carve, the resultant decomposition of the lattice digraph is not hierarchical, and reconstruction is complicated, inter alia, by the fact that edges can be required between constituent digraphs.

Valtchev et al. [11] described a procedure for recursive binary partitioning of an arbitrary formal context and the assembly of the overall lattice digraph from the digraphs arising from the sub-contexts. Only the attribute [object] set is partitioned, so that each sub-context contains all objects [attributes]. Whilst postulating that an "optimal" partition of the context would be one which minimised the size of the component digraphs, Valtchev et al. [11] left open the question of how to choose such a partition. In contrast, Carve discovers and exploits the structure of amenable contexts to recursively partition both the object and attribute sets, and significantly simplifies digraph assembly for this special case.

6 Summary

This paper has established the theoretical foundations of, and then detailed, the Carve algorithm, a divide-and-conquer technique for discovering and exploiting hierarchical structure in a formal context. Hierarchical partitioning of both the formal context and the resultant concept lattice have been used to achieve efficient computation of the lattice digraph. The discovered structure can be exploited for improved layout of, and interaction with, the Hasse diagram [10].

References

1. Andrews, S.: A 'best-of-breed' approach for designing a fast algorithm for computing fixpoints of Galois connections. Information Sciences **295**, 633–649 (2015). https://doi.org/10.1016/j.ins.2014.10.011
2. Baklouti, F., Levy, G.: Parallel algorithms for general Galois lattices building. In: Proc. Workshop on Distributed Data and Structures (WDAS 2003). Proceedings in Informatics, Carleton Scientific (2003)
3. Berry, A., Pogorelcnik, R., Sigayret, A.: Vertical decomposition of a lattice using clique separators. In: Napoli, A., Vychodil, V. (eds.) Concept Lattices and their Applications (CLA 2011). CEUR Workshop Proceedings, vol. 959, pp. 15–29 (2011), http://ceur-ws.org/Vol-959/
4. Bhatti, M.U., Anquetil, N., Huchard, M., Ducasse, S.: A catalog of patterns for concept lattice interpretation in software reengineering. In: Zhang, D., Reformat, M., Gokhale, S., Maldonado, J.C. (eds.) Proc. 24th Intl. Conf. Software Engineering & Knowledge Engineering (SEKE'2012). pp. 118–124. Knowledge Systems Institute Graduate School (2012)
5. Ganter, B., Wille, R.: Formal Concept Analysis: Mathematical Foundations. Springer, Berlin, Heidelberg (1999). https://doi.org/10.1007/978-3-642-59830-2
6. Gaume, B., Navarro, E., Prade, H.: A parallel between extended Formal Concept Analysis and bipartite graphs analysis. In: Hüllermeier, E., Kruse, R., Hoffmann, F. (eds.) Computational Intelligence for Knowledge-Based System Design. LNCS, vol. 6178, pp. 270–280. Springer, Berlin (2010). https://doi.org/10.1007/978-3-642-14049-5_28
7. Gély, A.: A generic algorithm for generating closed sets of a binary relation. In: Ganter, B., Godin, R. (eds.) Formal Concept Analysis. LNCS, vol. 3403, pp. 223–234. Springer, Berlin (2005). https://doi.org/10.1007/978-3-540-32262-7_15
8. Gross, J.L., Yellen, J., Zhang, P. (eds.): Handbook of Graph Theory. Chapman & Hall/CRC Press, New York, second edn. (2013)
9. Kuznetsov, S.O., Poelmans, J.: Knowledge representation and processing with Formal Concept Analysis. Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery **3**(3), 200–215 (2013). https://doi.org/10.1002/widm.1088
10. Pattison, T., Weber, D., Ceglar, A.: Enhancing layout and interaction in Formal Concept Analysis. In: Proc. 2014 IEEE Pacific Visualization Symposium. pp. 248–252. IEEE (2014). https://doi.org/10.1109/PacificVis.2014.21
11. Valtchev, P., Missaoui, R., Lebrun, P.: A partition-based approach towards constructing Galois (concept) lattices. Discrete Mathematics **256**(3), 801–829 (2002). https://doi.org/10.1016/S0012-365X(02)00349-7

Sequential Pattern Mining using FCA and Pattern Structures for Analyzing Visitor Trajectories in a Museum

Nyoman Juniarta, Miguel Couceiro, Amedeo Napoli, and Chedy Raïssi

Université de Lorraine, CNRS, Inria, LORIA, F-54000 Nancy, France
{nyoman.juniarta, miguel.couceiro, amedeo.napoli, chedy.raissi}@loria.fr

Abstract. This paper presents our work on mining visitor trajectories in Hecht Museum (Haifa, Israel), within the framework of CrossCult European Project about cultural heritage. We present a theoretical and practical research work about the characterization of visitor trajectories and the mining of these trajectories as sequences. The mining process is based on two approaches in the framework of FCA, namely the mining of subsequences without any constraint and the mining of frequent contiguous subsequences. Both approaches are based on pattern structures. In parallel, a similarity measure allows us to build a hierarchical classification which is used for interpretation and characterization of the trajectories w.r.t. four well-known visiting styles.

Keywords: FCA, pattern structures, sequence clustering, sequential pattern mining

1 Introduction

This paper is related to the CrossCult European Project about cultural heritage (http://www.crosscult.eu/). The general idea of CrossCult is to support the emergence of a European cultural heritage by allowing visitors in different locations (e.g. museum, city, archaeological site) to consider their visit at a European level by using adapted computer-based devices.

In this project, we are mainly interested in the analysis of visitor trajectories and recommendation. The trajectory of a visitor in a specific location is considered as a multi-dimensional sequence depending on a number of variables, such as space (e.g. paths, rooms, environment), time (e.g. hour, day, season, news), history and geography (e.g. town, region, country. . .). Moreover, additional domain knowledge and general knowledge bases such as DBpedia, Freebase or YAGO can be reused to draw inferences and improve the quality of both analysis and recommendation.

Here, we have two main objectives, (i) the mining of visitor trajectories based on sequence mining, and (ii) the characterization of a trajectory in terms of the subsequences which are mined. We assume that the subsequences are related to the visiting styles, the visit content, and the environment. Thus subsequences can

© Dmitry I. Ignatov, Lhouari Nourine (Eds.): CLA 2018, pp. 231–242,
ISBN 978–80–244–5328–6, Department of Computer Science, Palacký University
Olomouc, 2018.

be used for analyzing the trajectory of a visitor and for making recommendations all along the visit. Moreover, the occurrences of some subsequences at a given moment within a trajectory can witness a change of behavior –which triggers in turn a change in the recommendations.

In the present paper, we discuss theoretical and practical work about the definition of visitor trajectories and the mining of these trajectories as sequences. The mining process is based on two approaches about sequence mining in Formal Concept Analysis (FCA [1]): `MRGS` for "Mining Rare General Subsequences" [2] and `MFCS` for "Mining Frequent Contiguous Subsequences" [3]. The first approach mines rare subsequences in a general way, i.e. gaps may appear in the subsequences, while the second approach searches for frequent subsequences without any gap (a kind of substrings). If the original paper about `MRGS` [2] was interested in rare subsequences, this is no more the case here and we work on frequent subsequences as well. We also reuse the similarity measure sim_{ACS} developed for analyzing the trajectories of patients between hospitals [4,5]. This similarity measure allows us to build a hierarchical classification that will play a role of "reference classification". For analyzing and interpreting the trajectories of visitors, it is interesting to compare the outputs of `MRGS` and `MFCS` algorithms w.r.t. the clustering produced by sim_{ACS}. Moreover, these outputs and the clustering as well are analyzed thanks to four theoretical visiting styles, namely "ant", "butterfly", "fish" and "grasshopper" [6].

Several challenges are faced in this research work in the FCA framework: the mining of complex sequential data and dynamics in adapting two algorithms based on pattern structures, the analysis of the trajectories based on jumping emerging patterns and clustering. Here, data are not necessarily big but are rather complex and multidimensional and this should be taken into account.

The paper is organized as follows. Section 2 recalls the basic definitions about sequence mining that are useful for understanding the present work. Then, Sect. 3 presents the characteristics of the dataset that was used as a basis for the current work. In Sect. 5 and Sect. 6, we present first the application of clustering on data enabling to build classes of visitors, and then the application of two algorithms for mining interesting subsequences. Finally, in Sect. 7, an interpretation of the results and a discussion on the characterization of the visitor trajectories conclude the paper.

2 Sequence Mining

2.1 Basic Definitions

Pattern mining is the task of finding repeated occurrences in a dataset. For example, in a dataset about customer transactions, an objective can be to find a set of items that are frequently ordered in a single transaction. Another complex objective is to detect a set of items that are likely ordered within certain transactions. These specific tasks in pattern mining are related to sequential pattern mining. We recall below the most important definitions that we need in the following.

Definition 1. A sequence is an ordered list $\langle s_1 s_2 \dots s_m \rangle$, where s_i is an itemset $\{i_1, \dots, i_n\}$. Here, m is the *size* of a sequence. The *length* of a sequence is the total number of items, or $\sum |s_i|$.

For example, $\langle \{a, b\}\{a, c, d\} \rangle$ is a sequence with size 2, since it contains two itemsets. Its length is 5.

Definition 2. A sequence $s = \langle s_1 s_2 \dots s_m \rangle$ is a subsequence of $s' = \langle s'_1 s'_2 \dots s'_n \rangle$, denoted by $s \preceq s'$, if there exist indices $1 \leq i_1 < i_2 < \dots < i_m \leq n$ such that $s_j \subseteq s'_{i_j}$ for all $j = 1 \dots m$ and $m \leq n$.

Therefore, the sequence $\langle \{a\}\{d\} \rangle$ is a subsequence of $\langle \{a, b\}\{a, c, d\} \rangle$, while sequence $\langle \{c\}\{d\} \rangle$ is not.

One way of evaluating the quality of a subsequence is to compute the support of the subsequence. Given a user-defined threshold, the subsequence can be "frequent", i.e. the support is above the threshold, or "rare", i.e. the support is below the threshold.

Definition 3. Let $\mathcal{S}$ be a sequential database. The *support* of a sequence s in $\mathcal{S}$ is: $support(s, \mathcal{S}) = |\{s_i \in \mathcal{S}; s \preceq s_i\}|$

There exist algorithms which can retrieve all frequent sequences [7,8]. A long sequence can have a combinatorial number of subsequences. Thus, if a long sequence is frequent, these algorithms return all of its subsequences. This leads to the retrieval of many uninteresting patterns. This issue has been studied in [9,10,11] by introducing the concept of "closed sequence". They narrow the output by disregarding sequences which have another supersequence with the same support (hence not closed).

Beside mining frequent sequences, another complex task is clustering. To achieve such a task, a distance or a similarity measure between two sequences has to be defined. The similarity measure sim_{ACS} was proposed in [5], which counts the number of all common subsequences (ACS). This measure is formulated as:

$$sim_{ACS}(S_i, S_j) = \frac{\phi_C(S_i, S_j)}{\max\{\phi_D(S_i), \phi_D(S_j)\}} \tag{1}$$

where $\phi_C(S_i, S_j)$ is the number of all common distinct subsequences between S_i and S_j, while $\phi_D(S_i)$ is the number of all distinct subsequences of S_i.

In this paper, we reuse sim_{ACS} with a restriction. Actually, we consider sequences whose itemsets include only one item. For simplicity, we omit the curly brackets to describe an itemset. Therefore we will write $\langle \{a\}\{d\}\{e\} \rangle$ as $\langle a, d, e \rangle$.

2.2 Sequence Mining in FCA

In this section we briefly present the two algorithms that are adapted for mining the trajectories of visitors in a museum, namely `MFCS` [3] and `MRGS` [2]. The names of the algorithms are not used as such in the papers but here we use them by commodity. Both algorithms are original and very efficient, and among the few algorithms performing sequence mining in the framework of FCA.

MFCS was originally introduced for mining trajectories of patients in hospitals. The algorithm is based on pattern structures and projections, and stability as well. One important characteristic of MFCS is that it mines contiguous subsequences, or stated differently, subsequences without any gap between items. This is due to the fact that physicians are mainly interested in consecutive events when analyzing healthcare trajectories. In addition, but this is not needed in our framework, MFCS is able to take into account a partial ordering – given by domain knowledge for example – defined on the items composing the sequences.

MRGS is also a sequence miner based on pattern structures but with a different purpose. The objective of MRGS is to mine rare rather than frequent subsequences, and in particular long subsequences with special characteristics. The algorithm is based on a specific pattern structure of subsequences, where the similarity operation is based on the discovery of common close subsequences (SCCS operation illustrated in a next section). The SCCS operation is based on a directed graph of alignments (DAG of alignments) which guide the mining of common subsequences. The algorithm shows very good performances and is most probably one of the few algorithms whose objective is the mining of rare subsequences. In our framework, we adapted MRGS and the support threshold for comparison purposes with frequent subsequences. However, we will use in our context MRGS as a standard sequence miner and we will be interested in frequent subsequences.

3 The Dataset of Museum Visitors

3.1 The Museum

In the framework of the CrossCult project, we are working on a specific dataset about the trajectories of 254 visitors in Hecht Museum in Haifa, Israel [12]. In the raw dataset, a visitor trajectory contains a list of visited items, where each visit is composed of three elements namely "start time", "end time", and "item name". An example is presented in Table 1.

Table 1: An example of one visitor trajectory

Start	Finish	Item
12:55:39	12:58:05	Crafts and Arts
12:58:06	12:58:22	Religion and Cult
12:58:22	12:58:27	Building Methods and Facilities
12:58:29	13:05:09	Wooden Tools

A visitor can have visits with various time lengths. In order to obtain more meaningful results and to reduce the complexity, we only consider visits lasting at least 90 seconds, but this is a parameter that can be relaxed or more constrained. Thirty-eight trajectories have no visit more than this threshold, so they are ignored, leaving us with 216 trajectories. Moreover, we model each trajectory as a sequence of visited items. Therefore, for trajectory in Table 1, the corresponding sequence is ⟨`Crafts and Arts`, `Wooden Tools`⟩. This preprocessing results in sequences of various size. Forty-five sequences have only one itemset, while three sequences have more than 15 itemsets.

Table 2: Grouping of museum items

Category	Items and their ID
1	`Entrance Reuben Hecht (101), Symbols Jewish Menorah (102), Persian Cult (103), Jerusalem Photo (104)`
...	...
8	`Upper Floor Entrance (801), Coins (802), Seven Species (803), Oil Lamps (804), Weights (805), Temple Mount (806), Jerusalem (807), Greece Egypt (808), Cyprus (809), Gems (810)`

We group the museum items according to their location, so that we obtain 8 categories of items. Some of them are listed in Table 2. We convert the raw dataset into sequences of items, where each item is represented by its ID. We define the IDs such that we can infer the category of an item by its first digit. Therefore, we obtain a dataset of 216 sequences of visitor trajectories – named V_1–V_{216} – where each sequence is composed by a list of IDs, as illustrated in Table 3.

Table 3: Examples of visitor trajectories

Visitor	Trajectory
V_1	⟨`101, 101, 401, 704`⟩
V_2	⟨`102, 402, 808, 206, 808`⟩
V_3	⟨`302, 102, 201, 302, 705, 402, 802`⟩
V_4	⟨`104, 704, 602, 302, 402, 103`⟩

3.2 The Four Visiting Styles

In a seminal work about the typing of visitor styles in a museum [6], four main behaviors have been detected and described, leading to different recommendations all along a visit [13,14]. These four styles are summarized below:

- The *ant* denotes a visitor who will surely see all the works following their location order in the museum. Then the recommendation can be the following

item, but depending also on some environmental factors such as the crowd in the museum, the accessibility of the item and the fatigue of the visitor.
- The *grasshopper* denotes a visitor who will see only certain artworks, jumping from one to the other. Then, to encourage such a person to visit more items, the recommendation can be to visit items having a content similar to items already visited.
- The *butterfly* denotes a visitor wanting to discover some and not all artworks, without having any exact preferences. Then, the recommendation is open and can be based on surprise (items which are very different one from the other).
- The *fish* denotes a visitor who does not feel that much interested in the artworks and stays most of the time in the center of the rooms without any precise objective. Then the recommendation can be to visit the most famous items in the museum which are the closer to the current visitor location, for encouraging the visitor to continue the visit and gain more interest.

Indeed, a visitor can change his/her style during a visit and other elements may be of importance, e.g. crowd or fatigue of the visitor.

4 The Workflow for Analyzing the Trajectories

In the following, one objective is to map specific subsequences included in the visitor trajectories to each visiting style for characterizing more precisely the style and then making smarter recommendations. To identify the behavior of each visitor, we propose the following workflow:

1. Cluster the visitor trajectories and attach a label for each visitor (Sect. 5).
2. Create two concept lattices using `MFCS` and `MRGS` over the whole dataset (Sect. 6.1).
3. From the two lattices, find jumping emerging patterns (JEPs) for each label (Sect. 7.2).
4. Based on their JEPs, these labels are then mapped into four visiting styles that has been explained in Sect. 3.2.

5 The Clustering of Trajectories

In this first experiment, we reuse the sim_{ACS} similarity measure for clustering the visitor trajectories. The idea is to check whether it is possible to distinguish the four visiting styles introduced in Section 3.2. We applied hierarchical clustering[1] based on sim_{ACS} to build a distance matrix between individuals. From the resulting dendrogram, we retained 5 clusters denoted by “A”, “B”, “C”, “D”, and “E”. Four of them are expected to match the four visiting patterns, namely *ant*, *butterfly*, *fish*, and *grasshopper*. The last cluster will gather all non-classified

[1] We used the *hclust* method from the R software [15].

trajectories. These five clusters have various sizes. Cluster “A”, “B”, “C”, “D”, and “E” have 11, 11, 59, 102, and 33 visitors respectively.

Actually, it is not easy to directly match the five clusters to corresponding visiting styles. For doing so, we will analyze the subsequences that can be attached to each cluster of trajectories. The benefit of the clustering is actually to provide a label among “A”, “B”, “C”, “D”, and “E” to the visitors. Thanks to these labels, we can perform a search for the so-called “jumping emerging patterns” and attach a characterization to the clusters based on the mined subsequences.

6 The Mining of Trajectories Considered as Sequences

6.1 Mining Subsequences with MFCS and MRGS

Below, we explain the application of the MFCS and MRGS algorithms to the museum dataset and the building of an associated concept lattice. Moreover, as will be discussed in the next section, the jumping sequential patterns which are mined will help us to characterize the visitor trajectories.

In MFCS and MRGS, pattern structures are used for mining sequences. The similarity operator ($\sqcap$) between any two sets of sequences is defined as the set of closed common subsequences (SCCS) in the two input sequences. Then, given two sequences, say $S_1 = \langle \texttt{401,502,503} \rangle$ and $S_2 = \langle \texttt{401,503,502} \rangle$, the similarity between these descriptions is:

$$\begin{aligned}\delta(S_1) \sqcap \delta(S_2) &= \{\langle \texttt{401,502,503} \rangle\} \sqcap \{\langle \texttt{401,503,502} \rangle\} \\ &= \{\langle \texttt{401,502} \rangle, \langle \texttt{401,503} \rangle\}\end{aligned}$$

In the dataset, the items are grouped into categories and the SCCS calculation is performed, checking whether two items belong to the same category. Using the MFCS algorithm [3] it becomes:

$$\begin{aligned}\delta(S_1) \sqcap \delta(S_2) &= \{\langle \texttt{401,502,503} \rangle\} \sqcap \{\langle \texttt{401,503,502} \rangle\} \\ &= \{\langle \texttt{502} \rangle, \langle \texttt{503} \rangle, \langle \texttt{401,5,5} \rangle\}\end{aligned}$$

It should be noticed that MFCS mines contiguous subsequences, i.e. in Definition 2, $i_k = i_{k-1} + 1$ for all $k \in \{2, 3, \ldots, m\}$.

In parallel, the default similarity operator of MRGS can be modified to accommodate our needs, such that non-contiguous common subsequences can be mined:

$$\begin{aligned}\delta(S_1) \sqcap \delta(S_2) &= \{\langle \texttt{401,502,503} \rangle\} \sqcap \{\langle \texttt{401,503,502} \rangle\} \\ &= \{\langle \texttt{401,502} \rangle, \langle \texttt{401,503} \rangle, \langle \texttt{401,5,5} \rangle\}\end{aligned}$$

Then, based either on MFCS or MRGS, a concept has an extent including a set of trajectories and an intent including a set of common subsequences. Again, it should be noticed that, based on whether a subsequence is contiguous or not, the resulting concept lattices are different.

For example, the concepts corresponding to Table 3 are shown in Table 4. Notice that both algorithms obtain a concept whose extent is V_2, V_3, V_4, albeit with different intent. Based on `MRGS`, the common subsequence of V_2, V_3, V_4 is $\langle 1, 402\rangle$, while according to `MFCS`, their common subsequences are $\langle 1\rangle$ and $\langle 402\rangle$. This is because items 1 and 402 are not contiguous in V_3 and V_4.

Table 4: The concepts that are computed by of `MFCS` and `MRGS` from four visitors in Table 3

Extent	Intent (`MFCS`)	Intent (`MRGS`)
V_1	⟨101,101,401,704⟩	
V_2	⟨102,402,808,206,808⟩	
V_3	⟨302,102,201,302,705,402,802⟩	
V_4	⟨104,704,602,302,402,103⟩	
$V_{1,2}$	⟨1,4⟩	*not present*
$V_{1,4}$	⟨1⟩, ⟨4⟩, ⟨704⟩	⟨1,1⟩, ⟨1,4⟩, ⟨1,704⟩
$V_{2,3}$	⟨2⟩, ⟨102⟩, ⟨402,8⟩	⟨102,402,8⟩, ⟨102,2,8⟩
$V_{3,4}$	⟨1⟩, ⟨302⟩, ⟨402⟩, ⟨7⟩	⟨1,302,402⟩, ⟨302,1⟩, ⟨1,7,402⟩
$V_{1,3,4}$	⟨1⟩, ⟨4⟩, ⟨7⟩	⟨1,4⟩, ⟨1,7⟩
$V_{2,3,4}$	⟨1⟩, ⟨402⟩	⟨1,402⟩
$V_{1,2,3,4}$	⟨1⟩, ⟨4⟩	⟨1,4⟩

6.2 Jumping Emerging Patterns

FCA is a non supervised classification process that can be turned into a supervised process thanks to the adding of a target attribute in the context, generally corresponding to a target class. Then the idea is to search for the so-called "Jumping Emerging Patterns" (JEPs) [16]. We have already applied this approach in [17] for analyzing and characterizing clusters of biological inhibitors. Here we adapt the same idea for characterizing this time the clusters of visitors discovered with the similarity measure sim_{ACS}.

More precisely, five clusters are discovered by classifying visitor trajectories with sim_{ACS}. These same trajectories are then considered as sequences composed of subsequences. Then a set of characteristic subsequences is extracted and these subsequences are used as "attributes" in a formal context where objects are visitor trajectories. The resulting formal context is completed with an extra attribute corresponding to the "cluster information", i.e. the cluster in which the trajectory is classified according to sim_{ACS}. A concept lattice can then be built from this completed context.

More interestingly, the cluster information is used for characterizing the concepts whose extents include trajectories of a single cluster. The intents – made of subsequences – of these particular concepts are JEPs, and as such they can be used to characterize the corresponding clusters. For example, if the extent of the concept $(\{V_{103}, V_{165}, V_{188}\}, \{\langle 4\rangle, \langle \mathtt{1}\rangle, \langle \mathtt{306}\rangle, \langle \mathtt{701,707}\rangle\})$ includes visitors from cluster B only, then its intent is JEPs for that cluster.

7 Discussion

7.1 About Interesting Subsequences

The first part of Table 5 shows some interesting contiguous subsequences from 4677 concepts discovered by MFCS. Thirty-three persons are visiting three items contiguously in category 1 of items located near the entrance. This is interesting to be noticed, as visitors are likely to spend more time in rooms located near the entrance, because they are arriving, they have high interest, and they are not tired. Then items of importance could be placed near the entrance for getting sufficient interest from visitors.

Thirteen people visit an item in category 7 – this category corresponds to items in the room of "Ancient Ship" which is one of the most famous items in this museum – right after an item in category 1. This is a characteristic of *grasshopper*, because 1 and 7 are separated by many other categories. These visitors have a specific interest for the "Ancient Ship" in the museum, since they skip all the items located between entrance and "Ancient Ship" (both categories can be considered as "far" from each others).

From 8019 concepts obtained by MRGS, some subsequences are presented in the second part of Table 5. The subsequence $\langle \mathtt{1,1}\rangle$ has a support of 69 with MFCS, and it has quite a similar support (66) with MRGS. Then we can draw the same conclusion, meaning that when a person visits two items in category 1, it is likely in continuation (to be compared with the preceding subsequence $\langle \mathtt{1,1,1}\rangle$).

Now, more interestingly, there are 38 persons visiting an item in category 3 after category 1, while much less persons (9) are doing the opposite. A similar conclusion can also be drawn with pairs $\langle \mathtt{4,7}\rangle$ (31) and $\langle \mathtt{7,4}\rangle$ (11). Based on such an observation, we can infer that visiting a museum is an "oriented activity"and that some directions are more preferred than others or "naturally followed", just as it is the case for the ordering of the rooms existing in the museum. By contrast, only a few visitors are quitting the "natural flow" and go "backward". Among these visitors, we can probably find visitors searching for more precision about preceding visited items.

7.2 Cluster Characterization

Now we are interested in characterizing the five clusters that were introduced in the previous section. For doing so, JEPs are searched in the two concept lattices obtained with MFCS and MRGS algorithms. Some of these concepts are listed in Table 6 and Table 7.

First, from both MFCS and MRGS, we cannot find any satisfying concept for JEP of cluster "E". This is because among all the concepts whose extent is

Table 5: Some interesting subsequences mined by MFCS (left) and MRGS (right)

Subsequence	Support
$\langle\texttt{1,1,1}\rangle$	33
$\langle\texttt{1,7}\rangle$	13
$\langle\texttt{1,1}\rangle$	66

Subsequence	Support
$\langle\texttt{1,3}\rangle$	38
$\langle\texttt{3,1}\rangle$	9
$\langle\texttt{4,7}\rangle$	31
$\langle\texttt{7,4}\rangle$	11
$\langle\texttt{1,1}\rangle$	69

exclusively from cluster "E", none of them has more than one visitor. If we consider the dataset, among 33 members of cluster "E", 32 of them visit less than 2 items during their whole visit. We can assume that they are visitors that are not really interested in visiting the museum. Therefore, we can quote safely label this cluster as *fish*.

Cluster "D" is more easily distinguishable. Based on subsequences of concept FD2–FD4, many visitors in this class skip some items. Also, in concept RD1 and RD2, some of them visit other items after item 701. This is not a natural direction, because items in category 7 are located farther from the entrance than items in category 4 or 5. We can interpret the visitors of this cluster as *grasshopper*, since they "jump" from one item to another.

Clusters "A", "B", and "C" are relatively similar to each other. The visitors associated to these clusters follow an *ant* behavior: a natural flow (based on RA1–RC1) and no "jump" (based on FA1–FC2). However, in FC3, three visitors visit 101, then 102, then back again to 101, indicating rather a *butterfly* behavior.

Table 6: Interesting concepts discovered by the MFCS algorithm

Concept ID	Extent	Intent	Support	Cluster
FA1	$\{V_{70}, V_{107}, V_{121}, V_{133}, V_{201}, V_{202}\}$	$\{\langle\texttt{1,1,402}\rangle, \langle\texttt{103}\rangle, \langle\texttt{2}\rangle\}$	6	A
FA2	$\{V_{70}, V_{93}, V_{107}, V_{121}\}$	$\{\langle\texttt{402}\rangle, \langle\texttt{103,104}\rangle\}$	4	A
FB1	$\{V_{103}, V_{165}, V_{188}\}$	$\{\langle\texttt{4}\rangle, \langle\texttt{1}\rangle, \langle\texttt{306}\rangle, \langle\texttt{701,707}\rangle\}$	3	B
FC1	$\{V_{4}, V_{8}, V_{28}, V_{32}, V_{84}, V_{152}\}$	$\{\langle\texttt{102}\rangle, \langle\texttt{101,1,101}\rangle\}$	6	C
FC2	$\{V_{53}, V_{152}, V_{169}, V_{189}, V_{190}, V_{203}\}$	$\{\langle\texttt{7}\rangle, \langle\texttt{102,4}\rangle\}$	6	C
FC3	$\{V_{4}, V_{8}, V_{32}\}$	$\{\langle\texttt{101,102,101}\rangle\}$	3	C
FD1	$\{V_{54}, V_{105}, V_{139}, V_{168}\}$	$\{\langle\texttt{202,4}\rangle\}$	4	D
FD2	$\{V_{139}, V_{168}\}$	$\{\langle\texttt{202,405,701}\rangle\}$	2	D
FD3	$\{V_{46}, V_{47}\}$	$\{\langle\texttt{101,602}\rangle\}$	2	D
FD4	$\{V_{89}, V_{163}\}$	$\{\langle\texttt{602,203}\rangle\}$	2	D

7.3 Conclusion

In this article, we have presented our experiments in mining visitor trajectories that are modeled as sequences of items. We incorporated a classification of

Table 7: Interesting concepts discovered by the MRGS algorithm

Concept ID	Extent	Intent	Support	Cluster
RA1	$\{V_{70}, V_{107}, V_{121}, V_{133}, V_{201}, V_{202}\}$	$\{\langle\texttt{1,1,402,2}\rangle, \langle\texttt{1,1,4}\rangle, \langle\texttt{103,402,2}\rangle, \langle\texttt{103,4}\rangle\}$	6	A
RB1	$\{V_{142}, V_{183}, V_{192}\}$	$\{\langle\texttt{102,1,1,1,1}\rangle, \langle\texttt{102,103,1,1}\rangle, \langle\texttt{1,1,1,1,1}\rangle, \langle\texttt{1,103,1,1}\rangle\}$	3	B
RC1	$\{V_4, V_8, V_{28}, V_{84}, V_{152}\}$	$\{\langle\texttt{1,1,1,101}\rangle, \langle\texttt{1,101,1,101}\rangle, \langle\texttt{1,1,1,1}\rangle, \langle\texttt{1,101,1,1}\rangle, \langle\texttt{101,1,1,1}\rangle, \langle\texttt{101,101,1,1}\rangle, \langle\texttt{101,101,101}\rangle, \langle\texttt{102,101}\rangle, \langle\texttt{102,1}\rangle\}$	5	C
RD1	$\{V_{71}, V_{79}\}$	$\{\langle\texttt{701,504}\rangle\}$	2	D
RD2	$\{V_{97}, V_{98}\}$	$\{\langle\texttt{701,406}\rangle\}$	2	D

museum items and built a concept lattice using pattern structures. We applied two sequence miners based on FCA to the visitor trajectories, namely MFCS and MRGS, to discover interesting contiguous and general subsequences.

Our result highlight some interesting patterns that may define visitor behaviors. This can help museum researchers to analyze and evaluate the placement of items and the visiting styles. Moreover, we have also studied the possibility of clustering the visitors based on a concept lattice. These clusters can be analyzed to build a recommendation system for future visitors, but we did not yet study this aspect until now.

In this paper, we only included in the sequences partial information about the museum. More interesting results can be obtained if other elements are taken into account, such as more general knowledge about history and geography, and duration and time of the visit... Furthermore, the selection of interesting concepts can be also guided by computing the stability of the concepts [18]. Finally, from a more dynamic point of view, ongoing information such as comments and state of the visitor during the visit could be also considered for analysis and in-line recommendation.

References

1. Ganter, B., Wille, R.: Formal Concept Analysis: Mathematical Foundations, Berlin: Springer Science & Business Media (1999)
2. Codocedo, V., Bosc, G., Kaytoue, M., Boulicaut, J.F., Napoli, A.: A proposition for sequence mining using pattern structures. Proc. Int. Conf. Formal Concept Analysis, 106–121 (2017)
3. Buzmakov, A., Egho, E., Jay, N., Kuznetsov, S.O., Napoli, A., Raïssi, C.: On mining complex sequential data by means of FCA and pattern structures. Int. J. General Syst. 45 (2), 135–159 (2016)
4. Egho, E., Jay, N., Raïssi, C., Ienco, D., Poncelet, P., Teisseire, M., Napoli, A.: A contribution to the discovery of multidimensional patterns in healthcare trajectories. J. Intell. Inform. Syst. 42 (2), 283–305 (2014)

5. Egho, E., Raïssi, C., Calders, T., Jay, N., Napoli, A.: On measuring similarity for sequences of itemsets. Data Min. Knowl. Discov. 29 (3), 732–764 (2015)
6. Véron, E., Levasseur, M.: Ethnographie de l'exposition. Paris: Bibliothèque Publique d'Information (1983)
7. Han, J., Pei, J., Mortazavi-Asl, B., Pinto, H., Chen, Q., Dayal, U., Hsu, M.: PrefixSpan: Mining sequential patterns efficiently by prefix-projected pattern growth. Proc. 17th Int. Conf. Data Eng., 215–224 (2001)
8. Ayres, J., Flannick, J., Gehrke, J., Yiu, T.: Sequential pattern mining using a bitmap representation. Proc. 8th ACM SIGKDD Int. Conf. Knowl. Discov. Data Min., 429–435 (2002)
9. Yan, X., Han, J., Afshar, R.: CloSpan: Mining: Closed sequential patterns in large datasets. Proc. Int. Conf. Data Min., 166–177 (2003)
10. Wang, J., Han, J.: BIDE: Efficient mining of frequent closed sequences. Proc. 20th Int. Conf. Data Eng., 79–90 (2004)
11. Gomariz, A., Campos, M., Marin, R., Goethals, B.: ClaSP: an efficient algorithm for mining frequent closed sequences. Pacific-Asia Conf. Knowl. Discov. Data Min., 50–61 (2013)
12. Lanir, J., Kuflik, T., Dim, E., Wecker, A.J., Stock, O.: The influence of a location-aware mobile guide on museum visitors' behavior. Interact. Comput. 25 (6), 443–460 (2013)
13. Zancanaro, M., Kuflik, T., Boger, Z., Goren-Bar, D., Goldwasser, D.: Analyzing museum visitors' behavior patterns. Int. Conf. User Modeling, 238–246 (2007)
14. Kuflik, T., Boger, Z., Zancanaro, M.: Analysis and prediction of museum visitors' behavioral pattern types. Ubiquitous Display Environments. Berlin: Springer, 161–176 (2012)
15. R Core Team: R: A Language and Environment for Statistical Computing. R Foundation for Statistical Computing, Vienna, Austria. (2014)
16. Dong, G., Li, J.: Efficient mining of emerging patterns: Discovering trends and differences. Proc. 5th ACM SIGKDD Int. Conf. Knowl. Discov. Data Min., 43–52 (1999)
17. Asses, Y., Buzmakov, A., Bourquard, T., Kuznetsov, S.O., Napoli, A.: A Hybrid Classification Approach based on FCA and Emerging Patterns - An application for the classification of biological inhibitors. Proc. CLA Volume 972 CEUR Workshop, 211–222 (2012)
18. Kuznetsov, S.O., Ignatov, D.I.: Concept stability for constructing taxonomies of web-site users. arXiv preprint arXiv:0905.1424 (2009)

Pattern Setups and Their Completions

Aimene Belfodil[1,4], Sergei O. Kuznetsov[2,3], and Mehdi Kaytoue[1,5]

[1] Univ Lyon, INSA Lyon, CNRS, LIRIS UMR 5205, F-69621, LYON, France
[2] National Research University Higher School of Economics, Moscow, Russia
[3] St. Petersburg Department of Steklov Mathematical Institute of Russian Academy of Sciences, Russia
[4] Mobile Devices Ingénierie, 100 Avenue Stalingrad, 94800, Villejuif, France
[5] Infologic, 99 avenue de Lyon, 26500 Bourg-Lès-Valence, France
firstname.surname@insa-lyon.fr, skuznetsov@hse.ru

Abstract. Pattern mining consists in discovering interesting patterns in data. For that, algorithms rely on smart techniques for enumerating the pattern search space and, generally, focus on compressed collections of patterns (e.g. closed patterns), to avoid redundancy. Formal Concept Analysis offers a generic framework, called *pattern structures*, to formalize many types of patterns, such as closed itemsets, intervals, graph and sequence sets. Additionally, it provides generic algorithms to enumerate *all* closed patterns and *only* them. The only condition is that the pattern space is a meet-semilattice, which, unfortunately does not always hold (e.g., for sequential and graph patterns). In this paper, we discuss *pattern setups*, a tool that models pattern search spaces relying only on posets. Subsequently, we revisit some techniques transforming pattern setups to a pattern structure using set of patterns, namely *completions*, and we state *a necessary and sufficient condition* for a pattern setup completion using *antichains* to be a *pattern structure*.

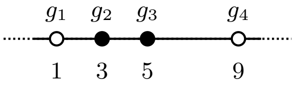
1 Introduction

Pattern mining is the task of finding interesting patterns in a predefined search space. A generic tool for defining such a pattern search space is pattern structures [10,14] based on *Formal Concept Analysis (FCA)* [9]. Indeed, itemsets, interval [12], convex [3], partition [2] pattern spaces among others can be modeled within the pattern structure framework. However, since pattern structures rely on meet-semilattices, some pattern spaces that are only posets cannot be "directly" defined using such a tool.

Fig. 1 depicts a dataset of 4 objects described by attribute `"value"` and labeled positive or negative. Consider the task of finding "good" rules $d \to +$ in this dataset with d a description given by attribute `value`. Rather than considering the usual meet-semilattice of intervals [12]; descriptions d are restrained to intervals of the form $(v]$ and $[v)$ or singleton $\{v\} \subseteq \mathbb{R}$ (see Fig. 1 - right). Descriptions form a poset $(\mathcal{D}, \supseteq)$ but not a meet-semilattice. For instance, the set $\{\{3\},\{5\}\}$ has not a meet, since lower bounds of $\{\{3\},\{5\}\}$ has two maximal elements w.r.t. $\supseteq$ (*i.e.* $[3)$ and $(5]$). Hence, the triple $(\mathcal{G}, (\mathcal{D}, \supseteq), \delta)$ with $\delta : g \mapsto \{\texttt{value}(g)\}$ does

© Dmitry I. Ignatov, Lhouari Nourine (Eds.): CLA 2018, pp. 243–253,
ISBN 978–80–244–5328–6, Department of Computer Science, Palacký University
Olomouc, 2018.

`object`	`value`	`label`
g_1	1	$-$
g_2	3	$+$
g_3	5	$+$
g_4	9	$-$



`Description language:`

$\texttt{value} = v \mid v \in \mathbb{R}$

$\texttt{value} \leq v$

$\texttt{value} \geq v$

Fig. 1. Dataset (left), its representation in $\mathbb{R}$ with black dots representing positive objects (center) and the description language (right).

not form a pattern structure [10] since $\{\delta(g_2), \delta(g_3)\} = \{\{3\}, \{5\}\}$ has not a meet. It does form actually a **pattern setup** [16] which is based on a poset.

Description spaces like the one in Fig. 1 are numerous. For instance, sequential patterns [1] ordered by *"is subsequence of"* do not form a meet-semilattice [19] [7]. Sequential meet-semilattice in FCA [6] [7] refers usually to **set of sequences** rather than to the poset of **sequences**. Same holds for the graph meet-semilattice from [13]. In general, the base pattern setup is transformed to a pattern structure using sets of descriptions thus providing a richer space (language). Such transformations are naturally called *completions*. For example, in Fig. 1, restriction $\{\texttt{value} \geq 3, \texttt{value} \leq 5\}$ is equivalent to $3 \leq \texttt{value} \leq 5$ and does not belong to the base description language.

Understanding properties of pattern setups independently from their completions is fundamental for answering many practical questions. For instance, consider the question *"What are the best descriptions covering all positive instances?"*. If **better** stands for **more relevant than** as in relevance theory [11], the answer will be the two best incomparable rules $\texttt{value} \geq 3 \rightarrow +$ and $\texttt{value} \leq 5 \rightarrow +$ rather than only one in the completion $3 \leq \texttt{value} \leq 5 \rightarrow +$.

In this paper, after recalling basic facts on pattern structures in Section 2, we discuss the properties of a pattern setup in Section 3. Subsequently, Section 4 revisits pattern setup completions and states a necessary and sufficient condition on a pattern setup to augment it to a pattern structure using antichains of patterns [13,6,7]. We coin pattern setups verifying such a condition *pattern hyperstructures*. The basic order-theoretic concepts we use in this paper are well-presented in [18] and [9]. Table 1 resumes the used notations.

Table 1. Notations

$\wp(E)$	Powerset of E	$\{S \mid S \subseteq E\}$
$f[S]$	Image of S by f	$\{f(s) \mid s \in S\} \subseteq F$ with $S \subseteq E$ and $f : E \rightarrow F$
$(\mathcal{D}, \sqsubseteq)$	Poset	partially ordered set. Below $S \subseteq \mathcal{D}$ is a subset
$\downarrow S$	Down closure	$\{d \in \mathcal{D} \mid (\exists s \in S)\, d \sqsubseteq s\}$. For $s \in S : \downarrow \{s\} \triangleq \downarrow s$
$\uparrow S$	Up closure	$\{d \in \mathcal{D} \mid (\exists s \in S)\, s \sqsubseteq d\}$. For $s \in S : \uparrow \{s\} \triangleq \uparrow s$
S^ℓ	Lower bounds	$\{d \in \mathcal{D} \mid (\forall s \in S)\, d \sqsubseteq s\}$. For $s \in S : \{s\}^\ell \triangleq s^\ell$
S^u	Upper bounds	$\{d \in \mathcal{D} \mid (\forall s \in S)\, s \sqsubseteq d\}$. For $s \in S : \{s\}^u \triangleq s^u$
$min(S)$	Minimal elements	$\{s \in S \mid d \in S, d \sqsubseteq s \Rightarrow d = s\}$
$max(S)$	Maximal elements	$\{s \in S \mid d \in S, s \sqsubseteq d \Rightarrow d = s\}$
$\bigsqcap S$	Meet	greatest element of S^ℓ if exists ($d \in S^\ell \Leftrightarrow d \sqsubseteq \bigsqcap S$)
$\bigsqcup S$	Join	smallest element of S^u if exists ($d \in S^u \Leftrightarrow \bigsqcup S \sqsubseteq d$)

2 Basic Definitions

In general, pattern search spaces are formalized by partially ordered sets. In this paper, we call **description space (language)** or **pattern space (language)** any *poset* $\underline{\mathcal{D}} := (\mathcal{D}, \sqsubseteq)$. Elements of $\mathcal{D}$ are called *descriptions* or *patterns.* For any $c, d \in \mathcal{D}$, $c \sqsubseteq d$ is read "*c subsumes d*" or "*c* is *less restrictive than d*".

Pattern structures is an extension of the basic FCA model [10]. Objects in a pattern structure have descriptions in a meet-semilattice. **Pattern setups** [16] generalize pattern structures by demanding only a partial order on descriptions.

Definition 1. *A* ***pattern setup*** *is a triple* $\mathbb{P} = (\mathcal{G}, \underline{\mathcal{D}}, \delta)$ *where* $\mathcal{G}$ *is a* set *of objects,* $\underline{\mathcal{D}}$ *is a description space and* $\delta : \mathcal{G} \to \mathcal{D}$ *is a mapping that takes each object* $g \in \mathcal{G}$ *to a description* $\delta(g) \in \mathcal{D}$. *We say that an object* $g \in \mathcal{G}$ ***realizes*** *a description* $d \in \mathcal{D}$ *or d* ***covers*** *g if* $d \sqsubseteq \delta(g)$. *A pattern setup* $\mathbb{P}$ *is said to be a* ***pattern structure*** *if every subset of* $\delta[\mathcal{G}] = \{\delta(g) \mid g \in \mathcal{G}\}$ *has a* ***meet*** *in* $\mathcal{D}$.

Note that a necessary and sufficient condition on $\underline{\mathcal{D}}$ to have a pattern structure $(\mathcal{G}, \underline{\mathcal{D}}, \sqsubseteq)$ on any finite set of objects $\mathcal{G}$ and any mapping $\delta : \mathcal{G} \to \mathcal{D}$ is that $\underline{\mathcal{D}}$ is a **meet-semilattice** with a **top element** ($\mathcal{D}$ has its greatest element).

In **pattern structures**, two derivation operators map posets $(\wp(\mathcal{G}), \subseteq)$ and $(\mathcal{D}, \sqsubseteq)$ to each other. They are usually both denoted by $(\cdot)^{\square}$. For more clarity we use *ext* and *int* notation.

Definition 2. *The* ***extent operator*** *of a pattern setup, denoted by ext, takes each description* $d \in \mathcal{D}$ *to the set of objects in* $\mathcal{G}$ ***realizing*** *it. In the case the pattern setup is a* pattern structure*, the* ***intent operator****, denoted by int, takes a subset of objects* $A \subseteq \mathcal{G}$ *to the largest common description in* $\mathcal{D}$ ***covering*** *them, with* $\bigsqcap$ *representing the meet in* $\underline{\mathcal{D}}$*, we have:*

$$ext : \mathcal{D} \to \wp(\mathcal{G}), d \mapsto \{g \in \mathcal{G} \mid d \sqsubseteq \delta(g)\} \qquad int : \wp(\mathcal{G}) \to \mathcal{D}, A \mapsto \bigsqcap \delta[A]$$

The size $|ext(d)|$ *is called the* ***support*** *of d and is denoted by* $support(d)$.

In a *pattern structure* $\mathbb{P}$, the pair of operators (ext, int) forms a **Galois connection** between posets $(\wp(\mathcal{G}), \subseteq)$ and $(\mathcal{D}, \sqsubseteq)$. Thus, $ext \circ int$ and $int \circ ext$ form **closure operators** on the two posets and $(ext[\mathcal{D}], \subseteq)$ is a $\bigcap$**-structure** (*i.e.* a complete lattice). Moreover, another *complete lattice* isomorphic to poset $(ext[\mathcal{D}], \subseteq)$ is used. It is called **pattern concept lattice** and is denoted by $\underline{\mathfrak{B}}(\mathbb{P}) = (\mathfrak{B}(\mathbb{P}), \leq)$. Elements of $\underline{\mathfrak{B}}(\mathbb{P})$ are called **pattern concepts** and are of the form (A, d_A) with $A = ext(d_A)$ and $d_A = int(A)$. Pattern concepts are ordered as follows: $(A, d_A) \leq (B, d_B) \Leftrightarrow A \subseteq B \Leftrightarrow d_B \sqsubseteq d_A$.

3 Pattern Setups

Whereas basic FCA considers only binary data (descriptions are sets), pattern structures can handle more complex languages. Nevertheless, it fails at considering some types of patterns, which do not make a semi-lattice. Before diving into details, consider the following example.

$\mathcal{G}$	$\delta(\cdot)$
g_1	$c \rightarrow a \rightarrow b$
g_2	$c \rightarrow b \rightarrow b \rightarrow a$
g_3	a
g_4	$b \rightarrow b \rightarrow c$

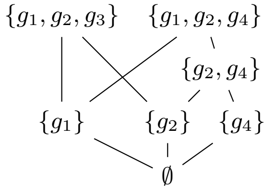


Fig. 2. The table (left) represents the pattern setup considered in example 1. The diagram (right) represents the poset $(ext[\mathcal{D}], \subseteq)$ (not making a closure system).

Example 1. Let be the dataset with the set of objects $\mathcal{G} = \{g_1, g_2, g_3, g_4\}$ in Fig. 2 (left). The description space $(\mathcal{D}, \sqsubseteq)$ contains all **non empty sequences** that can be built using items in $\{a, b, c\}$. It is ordered by the relationship **"is substring of"** (i.e. is subsequence of -without gaps-) denoted by $\sqsubseteq$. Such an order does not form a meet-semilattice. Indeed, consider sequences $\delta(g_1) = c \rightarrow a \rightarrow b$ and $\delta(g_2) = c \rightarrow b \rightarrow b \rightarrow a$; clearly, their common lower bounds $\{\delta(g_1), \delta(g_2)\}^\ell = \{a, b, c\}$ form an antichain and hence $\{\delta(g_1), \delta(g_2)\}$ has no meet. It follows that $(\mathcal{G}, \underline{\mathcal{D}}, \delta)$ is a *pattern setup* but not a *pattern structure*.

Important Remark. *Unless otherwise mentioned, in this paper $\mathbb{P} = (\mathcal{G}, \underline{\mathcal{D}}, \delta)$ denotes a* ***pattern setup*** *where $\mathcal{G}$ is a* ***non-empty finite set****. Hence, theorems and propositions in this paper are guaranteed to be valid only if $\mathcal{G}$ is* ***finite****.*

It is clear that the *extent operator* (see Definition 2) can be used in the general case of pattern setups since it requires only the order. However, the *intent* of a set may not exist. We define here the **cover operator**.

Definition 3. *We call* ***cover operator****, denoted by cov, the operator that takes each subset $A \subseteq \mathcal{G}$ to the set of common descriptions in $\mathcal{D}$* ***covering*** *them:*

$$cov : \wp(\mathcal{G}) \rightarrow \wp(\mathcal{D}), A \mapsto \delta[A]^\ell = \{d \in \mathcal{D} \mid (\forall g \in A)\ d \sqsubseteq \delta(g)\}$$

Note that ext and cov are **order reversing mappings** in a pattern setup: $(\forall A, B \subseteq \mathcal{G}) A \subseteq B \Rightarrow cov(B) \subseteq cov(A)$ and $(\forall c, d \in \mathcal{D}) c \sqsubseteq d \Rightarrow ext(d) \subseteq ext(c)$.

Some sets of objects $A \subseteq \mathcal{G}$ can have no common descriptions covering them. In other words, $cov(A) = \emptyset$. Definition 4 develops a categorization of sets $A \subseteq \mathcal{G}$.

Definition 4. *Let A be a subset of $\mathcal{G}$. A is said to be an* ***extent*** *iff $\exists d \in \mathcal{D} \mid A = ext(d)$. A is said to be* ***coverable*** *iff $\exists d \in \mathcal{D} : A \subseteq ext(d)$, otherwise it is said to be* ***non coverable****. The set of all possible* ***extents*** *is denoted by $\mathbb{P}_{ext}$ and given by $\mathbb{P}_{ext} = ext[\mathcal{D}]$. The set of* ***coverable sets*** *is given by $\downarrow \mathbb{P}_{ext}$.*

Example 2. Consider Fig. 2. Set $A = \{g_2, g_4\}$ has cover $cov(A) = \{b, b \rightarrow b, c\}$. Moreover, A is an **extent** since $ext(b \rightarrow b) = A$. Set $B = \{g_3\}$ is **not an extent** since the only covering description is "a" for which $ext(a) = \{g_1, g_2, g_3\}$ (B is **coverable**). Set $C = \{g_3, g_4\}$ is **non coverable**, since g_3 and g_4 do not have common symbols.

Remark 1. As in **pattern structures** [10], **pattern implication** and **object implication** can be defined thanks to extent and cover operators, respectively. For $c, d \in \mathcal{D}$, the **pattern implication** $c \rightarrow d$ holds if $ext(c) \subseteq ext(d)$. Conversely, for $A, B \subseteq \mathcal{G}$, the **object implication** $A \rightarrow B$ holds if $cov(A) \subseteq cov(B)$.

3.1 On Maximal Descriptions in a Cover

In pattern structures, **intent** operator associates the largest common description to a set of objects. However, **cover** operator takes a set of objects to the set of all descriptions covering them. Such a set can be huge and needs to be "compressed". Since we have no guarantees of the existence of the **largest** common description, a reasonable suggestion would be to consider the **maximal** ones.

Definition 5. *The set of* ***maximal descriptions*** *covering a subset $A \subseteq \mathcal{G}$, denoted by $cov^{\mathcal{I}}(A)$, is given by:*

$$cov^{\mathcal{I}}(A) = max(cov(A)) = max(\delta[A]^{\ell})$$

Remark 2. Maximal descriptions are sometimes called **support-closed descriptions** [5]. A description d is said to be *support-closed* if $\forall c \in \mathcal{D}$ such that for $d \sqsubseteq c$ and $c \neq d$ we have $support(c) < support(d)$ $(i.e.\ ext(c) \subsetneq ext(d))$.

3.2 On Upper Approximations of a Set

In pattern structures a description $d \in \mathcal{D}$ **covering** the set A will **certainly cover** $ext(int(A))$. This observation is important when it comes to search for positive or negative hypotheses in a labeled dataset [13] or in other words classification association rules [15]. Indeed, if $\mathcal{G}^+$ represents the whole set of positive objects in the dataset $\mathcal{G}$ and if we want exactly one rule which covers all positive instances, the best one (in terms of relevance [11]) will be the rule $int(\mathcal{G}^+) \rightarrow +$ with confidence $|\mathcal{G}^+|/|ext(int(\mathcal{G}^+))|$. For pattern structures $ext(int(A))$ is the closure of A, in Rough Set Theory [17], $ext(int(A))$ can be seen as the **upper approximation** of A in $\mathcal{G}$. To have a similar notion for pattern setups, we define what we call **upper-approximation extents**.

Definition 6. *The* ***set of upper-approximation extents*** *of a subset $A \subseteq \mathcal{G}$, denoted by $\overline{A}$, is given by: $\overline{A} = min(\{E \in ext[\mathcal{D}] \mid A \subseteq E\}) = min(\uparrow A \cap ext[\mathcal{D}])$.*

Example 3. Extent $A = \{g_2, g_4\}$ in Fig. 2 has a unique upper-approximation (i.e. $\overline{A} = \{\{g_2, g_4\}\}$). Coverable subset $B = \{g_1, g_2\}$ has multiple upper approximations $\overline{B} = \{\{g_1, g_2, g_3\}, \{g_1, g_2, g_4\}\}$. Non coverable subset $C = \{g_3, g_4\}$ has no upper approximations (i.e. $\overline{C} = \emptyset$). Proposition 1 formalizes these observations.

Proposition 1. *For any $A \subseteq \mathcal{G}$, we have:*

$$A \in \mathbb{P}_{ext} \Leftrightarrow \overline{A} = \{A\}\ \textbf{(1)}\ A \in \downarrow \mathbb{P}_{ext} \Leftrightarrow \overline{A} \neq \emptyset\ \textbf{(2)}\ \overline{A} = min(ext[cov(A)])\ \textbf{(3)}$$

Proof. We demonstrate each property independently:

1. $A \in ext[\mathcal{D}] \Leftrightarrow \overline{A} = \{A\}$: For $(\Leftarrow)$, $\overline{A} \subseteq ext[\mathcal{D}]$, thus $A \in ext[\mathcal{D}]$. For $(\Rightarrow)$, we have $A \in \uparrow A \cap ext[\mathcal{D}]$ thus $\overline{A} = min(\uparrow A \cap ext[\mathcal{D}]) = \{A\}$.
2. $A \in \downarrow ext[\mathcal{D}] \Leftrightarrow \overline{A} \neq \emptyset$: For $(\Rightarrow)$, we have $A \in \downarrow ext[\mathcal{D}]$, that is $\exists B \in ext[\mathcal{D}]$ s.t. $A \subseteq B$ $(i.e\ B \in \uparrow A)$. Thus $\uparrow A \cap ext[\mathcal{D}]$ is not empty and so does $\overline{A}$ (since $\uparrow A \cap ext[\mathcal{D}]$ is finite). For $(\Leftarrow)$, we have $\overline{A} \neq \emptyset$, that is $\uparrow A \cap ext[\mathcal{D}] \neq \emptyset$ thus $\exists B \in ext[\mathcal{D}]$ such that $A \subseteq B$ thus $A \in \downarrow ext[\mathcal{D}] = \downarrow \mathbb{P}_{ext}$.

3. We show $ext[cov(A)] = \uparrow A \cap ext[\mathcal{D}]$. Let $B \subseteq \mathcal{G}$. We have $B \in ext[cov(A)] \Leftrightarrow \exists d \in cov(A)$ s.t. $B = ext(d) \Leftrightarrow \exists d \in \mathcal{D}\ \forall g \in A : d \sqsubseteq \delta(g) \Leftrightarrow \exists d \in \mathcal{D}$ $A \subseteq ext(d) = B \Leftrightarrow B \in ext[\mathcal{D}] \cap \uparrow A$. This concludes the proof. □

For a set $A \subseteq \mathcal{G}$ s.t. $\overline{A} \neq \emptyset$, any description covering A **contains at least** elements of $\bigcap \overline{A}$, the intersection of upper approximations of A.

Proposition 2. *$I : \wp(\mathcal{G}) \to \wp(\mathcal{G}), A \mapsto \bigcap \overline{A} = \bigcap \{E \in ext[\mathcal{D}] \mid A \subseteq E\}$ is a **closure operator** on $(\wp(\mathcal{G}), \subseteq)$. Moreover, we have $\overline{I(A)} = \overline{A}$ for any $A \subseteq \mathcal{G}$.*

Proof. In fact, this proposition is a direct application of the following Lemma.

Lemma 1. *Let $(P, \leq)$ be a complete lattice with $\bigwedge$ its meet and let $E \subseteq P$ be a subset. The mapping $\phi_E : P \to P, p \mapsto \phi_E(p) = \bigwedge \{e \in E \mid p \leq e\}$ is a **closure operator** on $(P, \leq)$ and $\phi_E[P]$ is a **meet-structure** in poset $(P, \leq)$.*

We prove Lemma 1. **(1)** ϕ_E **is extensive**. Trivially $p \in \{e \in E \mid p \leq e\}^{\ell}$ for $p \in P$. Since the meet is the greatest element, we conclude: $p \leq \phi_E(p)$. **(2)** ϕ_E **is monotone**. Let $p_1, p_2 \in P$ s.t. $p_1 \leq p_2$, we have $\{e \in E \mid p_2 \leq e\} \subseteq \{e \in E \mid p_1 \leq e\}$ thus $\{e \in E \mid p_1 \leq e\}^{\ell} \subseteq \{e \in E \mid p_2 \leq e\}^{\ell}$. We conclude that $\phi_E(p_1) \leq \phi_E(p_2)$. **(3)** ϕ_E **is idempotent**. Let us show $\{e \in E \mid \phi_E(p) \leq e\} = \{e \in E \mid p \leq e\}$. Inclusion $\subseteq$ is verified since $p \leq \phi_E(p)$. Inclusion $\supseteq$ comes from the definition since $\phi_E(p)$ is a lower bound of $\{e \in E \mid p \leq e\}$, then for any $e \in E$ such that $p \leq e$ we have $\phi_E(p) \leq e$. The idempotence is straightforward.

Proposition 2 is a corollary of Lemma 1 since $(\wp(\mathcal{G}), \subseteq)$ is a complete lattice in which the meet is the set intersection. Indeed, $I \triangleq \phi_{ext[\mathcal{D}]}$ is a closure operator on $(\wp(\mathcal{G}), \subseteq)$. $\overline{I(A)} = \overline{A}$ comes directly from the previous idempotence proof. □

Proposition 3. *For any $g \in \mathcal{G}$ we have $\overline{\{g\}} = \{ext(\delta(g))\}$. That is, any singleton set $\{g\} \subseteq \mathcal{G}$ is coverable and has a **unique upper-approximation**.*

Proof. Let $g \in \mathcal{G}$, we have $\delta(g) \in cov(\{g\})$, thus $ext(\delta(g)) \in ext[cov(\{g\})]$. Let us show that $ext(\delta(g))$ is a lower bound of $ext[cov(\{g\})]$. We have by definition: $cov(\{g\}) = \{d \in \mathcal{D} \mid d \sqsubseteq \delta(g)\}$. Thus, $\forall d \in cov(\{g\}) : d \sqsubseteq \delta(g)$. Since ext is an order reversing operator, we obtain: $\forall A \in ext[cov(\{g\})] : ext(\delta(g)) \subseteq A$. Thus $ext(\delta(g))$ is the smallest element of $ext[cov(\{g\})]$. That is $\overline{\{g\}} = \{ext(\delta(g))\}$. □

3.3 Linking Upper Approximations and Maximal Descriptions

Now that we have both upper approximations and maximal covering descriptions, a judicious question would be:

Question 1. *What is the relationship between $cov^{\mathcal{I}}(A)$ and $\overline{A}$ for $A \subseteq \mathcal{G}$?*

Before diving into more details, consider the example below.

Example 4. Consider Fig. 2. **Extent** $A = \{g_2, g_4\}$ has $cov(A) = \{b, b \to b, c\}$. Hence, $cov^{\mathcal{I}}(A) = \{b \to b, c\}$. Therefore, $ext[cov^{\mathcal{I}}(A)] = \{\{g_2, g_4\}, \{g_1, g_2, g_4\}\}$. Besides, $\overline{A} = \{\{g_2, g_4\}\}$. Thus, counter-intuitively, $\overline{A}$ is not equal to $ext[cov^{\mathcal{I}}(A)]$.

Furthermore, when $\underline{\mathcal{D}}$ is an **infinite poset**, the set $cov^{\mathcal{I}}(A)$ might not "hold" all the information contained in $cov(A)$. That is to say: "knowing only maximal covering descriptions does not allow us to deduce the set of covering ones". Speaking formally, $cov^{\mathcal{I}}(A)$ does not necessarily verify $cov(A) = \downarrow cov^{\mathcal{I}}(A)$.

4 Pattern Hyper-Structures and Completions

There is a standard construction how an ordered set of descriptions can be turned in a semilattice of descriptions. For example, see [13], where a semilattice on sets of graphs with labeled vertices and edges is constructed from the order given by subgraph isomorphism relation. Let $\underline{\mathcal{D}} = (\mathcal{D}, \sqsubseteq)$ be a poset and let $\mathcal{A}(\mathcal{D})$ be the set of its antichains. It is possible to define a *partial order* by letting $S_1, S_2 \in \mathcal{A}(\mathcal{D})$ [4]: $S_1 \leq S_2$ iff $\forall s_1 \in S_1 \, \exists s_2 \in S_2 : s_1 \sqsubseteq s_2$ (i.e. $S_1 \subseteq \downarrow S_2$). Please note that this relation does not define an order on $\wp(\mathcal{D})$. It does define only a **pre-order** since *anti-symmetry* does not hold (see [8]).

In fact, when $\mathcal{D}$ is **finite**, $(\mathcal{A}(\mathcal{D}), \leq)$ is a **distributive lattice** where the meet and the join are given by $S_1 \wedge S_2 = max(\downarrow S_1 \cap \downarrow S_2)$ and $S_1 \vee S_2 = max(S_1 \cup S_2)$, respectively. Thus, one can build a pattern structure with $(\mathcal{A}(\mathcal{D}), \leq)$ which embeds the pattern setup $\mathbb{P} = (\mathcal{G}, \underline{\mathcal{D}}, \delta)$ for any finite set of objects $\mathcal{G}$. We call such a **pattern structure** the **antichain completion** of $\mathbb{P}$.

Definition 7. *Let $\mathbb{P} = (\mathcal{G}, \underline{\mathcal{D}}, \delta)$ be a **pattern setup**, the **antichain completion** of $\mathbb{P}$ is the **pattern setup** denoted by $\mathbb{P}^{\triangledown}$ and given by:*

$$\mathbb{P}^{\triangledown} = (\mathcal{G}, (\mathcal{A}(\mathcal{D}), \leq), \sigma : g \mapsto \{\delta(g)\})$$

Consider the following question for the more general case where $\underline{\mathcal{D}}$ is **infinite**:

Question 2. *What is a necessary and sufficient condition on $\mathbb{P}$ that makes $\mathbb{P}^{\triangledown}$ a **pattern structure**?*

We have seen that the finiteness of $\underline{\mathcal{D}}$ is a **sufficient** condition (i.e. $(\mathcal{A}(\mathcal{D}), \leq)$ is a distributive lattice), but not a necessary one. Definition 1 requires for $\mathbb{P}^{\triangledown}$ that for any $A \subseteq \mathcal{G}$ we have $\sigma[A] = \{\{\delta(g)\} \mid g \in A\}$ has a meet in $(\mathcal{A}(\mathcal{D}), \leq)$. This condition is verified **iff** pattern setup $\mathbb{P}$ satisfies the following condition:

$$\forall A \subseteq \mathcal{G} : cov(A) = \delta[A]^{\ell} \;\; = \;\; \downarrow max(\delta[A]^{\ell}) = \downarrow cov^{\mathcal{I}}(A) \qquad (1)$$

and then $cov^{\mathcal{I}}(A)$ is the meet of $\sigma[A]$ in $(\mathcal{A}(\mathcal{D}), \leq)$ (see Theorem 2).

Definition 8. *A pattern setup $(\mathcal{G}, \underline{\mathcal{D}}, \delta)$ is said to be a **pattern hyper-structure** if condition (1) holds.*

Note that the term **hyper** in **pattern hyper-structure** comes from the notion of hyper-lattices briefly introduced in [19]. Note also that graphs ordered by subgraph isomorphism relation [13] induces a pattern hyper-structure but not a pattern structure. Same remark holds for sequential patterns [6,7] under the assumption of the existence of the largest sequence **1** subsumed by all sequences.

Remark 3. When $\underline{\mathcal{D}}$ is **infinite**, poset $(\mathcal{A}(\mathcal{D}), \leq)$ remains to be a join-semilattice but not necessarily has *finite meets* (as it is when $\underline{\mathcal{D}}$ is **finite**). It was shown in [4] that a **necessary and sufficient condition** on $(\mathcal{A}(\mathcal{D}), \leq)$ to have *finite meets* is as follows:

$$\forall S_1, S_2 \in \mathcal{A}(\mathcal{D})\ \exists S \in \mathcal{A}(\mathcal{D}) : \downarrow S_1 \cap \downarrow S_2 = \downarrow S \tag{2}$$

In this case, S represents the meet (infimum) of $\{S_1, S_2\}$ in $\mathcal{A}(\mathcal{D})$; moreover, $(\mathcal{A}(\mathcal{D}), \leq)$ becomes a distributive lattice. Requiring $(\mathcal{A}(\mathcal{D}), \leq)$ to have a meet is a **sufficient condition**, but still **not necessary** for the *antichain completion* to be a pattern structure. In fact, condition (2) implies condition (1) (case $\mathcal{G}$ finite). Moreover, an example from [4] shows that condition (2) does not always hold even if $(\mathcal{D}, \sqsubseteq)$ is a meet-semilattice. Example 5 shows that condition (1) does not always hold.

Example 5. Consider the pattern setup $(\mathcal{G}, (\mathcal{D}, \supseteq), \delta)$ with $\mathcal{G} = \{g_1, g_2\}$, $\mathcal{D} = \{[a,b] \subseteq \mathbb{R} \mid a, b \in \mathbb{R}\} \backslash [1,3]$, $\delta(g_1) = \{1\}$ and $\delta(g_2) = \{3\}$. The considered pattern setup does not verify condition (1) since $\delta[\mathcal{G}]^{\ell}$ is clearly not empty while $max(\delta[\mathcal{G}]^{\ell}]) = \emptyset$. $(\mathcal{G}, (\mathcal{D}, \supseteq), \delta)$ is thus not a pattern hyper-structure.

Theorem 1. *For any* ***pattern hyper-structure*** $(\mathcal{G}, \underline{\mathcal{D}}, \delta)$*, we have:*

$$\forall A \subseteq \mathcal{G} : \overline{A} = min\left(ext\left[cov^{\mathcal{I}}(A)\right]\right)$$

Proof. Let $A \in \wp(\mathcal{G})$. If $cov(A) = \emptyset$, then the property trivially holds since $cov^{\mathcal{I}}(A) = cov(A) = \emptyset$. Let $A \in \downarrow ext[\mathcal{D}]$ (i.e. $cov(A) \neq \emptyset$). We need to show the following equivalent proposition $min(ext[cov(A)]) = min(ext[max(cov(A))])$.

We start by showing $min(ext[cov(A)]) \subseteq min(ext[max(cov(A))])$. Let $B \in min(ext[cov(A)])$. Since $B \in ext[cov(A)]$, $\exists d \in cov(A)$ s.t. $B = ext(d)$. Since $cov(A) = \downarrow max(cov(A))$ ($(\mathcal{G}, \underline{\mathcal{D}}, \delta)$ is a pattern hyper-structure), there exists $c \in max(cov(A))$ s.t. $d \sqsubseteq c$. Since extent is an order revering mapping, $C = ext(c) \subseteq ext(d) = B$. Supposing that $C \in ext[max(cov(A))]$ s.t. $C \subsetneq B$ contradicts the fact that $B \in min(ext[cov(A)])$ since $C \subsetneq B$ and $C \in ext[cov(A)]$. It follows that $C = B \in ext[max(cov(A))]$. Again, supposing that $B \notin min(ext[max(cov(A))])$ leads to a contradiction ($\exists D \in ext[max(cov(A)] \subseteq ext[cov(A)]$ s.t. $D \subsetneq B$ while $B \in min(ext[cov(A)])$). We conclude that $B \in min(ext[max(cov(A))])$.

It remains to show that $min(ext[cov(A)]) \supseteq min(ext[max(cov(A))])$. Suppose the converse: $\exists E \in min(ext[max(cov(A))])$ such that $E \notin min(ext[cov(A)])$, we have $E \in ext[max(cov(A))] \subseteq ext[cov(A)]$. Since $E \notin min(ext[cov(A)])$ and $ext[cov(A)]$ is finite, we obtain that $\exists F \in min(ext[cov(A)])$ such that $F \subsetneq E$. The first inclusion implies $F \in min(ext[max(cov(A))])$. This is a contradiction, since at the same time $F \subsetneq E$ and $E \in min(ext[max(cov(A))])$.

We conclude that $\overline{A} = min(ext[cov(A)]) = min(ext[cov^{\mathcal{I}}(A)])$. □

Theorem 1 answers question 1. It says that in a **pattern hyper-structure** rather than considering all covering descriptions to compute $\overline{A}$, it is sufficient to consider only maximal covering ones. Theorem 2 answers question 2:

Theorem 2. *Let $\mathbb{P} = (\mathcal{G}, \underline{\mathcal{D}}, \delta)$ be a pattern setup, the* ***antichain completion*** *of $\mathbb{P}$ is a* ***pattern structure*** iff *$\mathbb{P}$ is a* ***pattern hyper-structure****. Operators $ext^{\triangledown}$ and $int^{\triangledown}$ denote extent and intent of $\mathbb{P}^{\triangledown}$ and are given by*

$$(\forall S \in \mathcal{A}(\mathcal{D}))\; ext^{\triangledown}(S) = \bigcap ext[S] \qquad (\forall A \subseteq \mathcal{G})\; int^{\triangledown}(A) = cov^{\mathcal{I}}(A)$$

Moreover, we have $\mathbb{P}^{\triangledown}_{ext} = \{\bigcap S \mid S \subseteq \mathbb{P}_{ext}\}$. Note that $\mathcal{G} \in \mathbb{P}^{\triangledown}_{ext}$.

Proof. Let us show that if $\mathbb{P}$ is a pattern hyper-structure then $\mathbb{P}^{\triangledown}$ is a pattern structure. $\mathbb{P}^{\triangledown}$ is a pattern structure **iff** every subset of $\sigma[\mathcal{G}]$ has a meet in $(\mathcal{A}(\mathcal{D}), \leq)$. For $A \subseteq \mathcal{G}$ we have $\sigma[A]^{\ell} = \{S \in \mathcal{A}(\mathcal{D}) \mid (\forall g \in A)\, S \subseteq\downarrow \delta(g)\} = \{S \in \mathcal{A}(\mathcal{D}) \mid S \subseteq \delta[A]^{\ell}\}$ where $\delta[A]^{\ell}$ and $\sigma[A]^{\ell}$ denote respectively the lower bounds of $\delta[A]$ w.r.t. $\sqsubseteq$ and the lower bounds of of $\sigma[A]$ w.r.t. $\leq$ (recall that $\delta[A]^{\ell} = \bigcap_{g \in A} \downarrow \delta(g)$). In this proof $\downarrow$ refers to the down-closure related to $\sqsubseteq$.

- ($\Rightarrow$) Let $A \subseteq \mathcal{G} : \delta[A]^{\ell} =\downarrow max(\delta[A]^{\ell})$. Thus $\sigma[A]^{\ell} = \{S \in \mathcal{A}(\mathcal{D}) \mid S \subseteq\downarrow max(\delta[A]^{\ell})\} = \{S \in \mathcal{A}(\mathcal{D}) \mid S \leq max(\delta[A]^{\ell})\}$. Since $max(\delta[A]^{\ell}) \in \mathcal{A}(\mathcal{D})$, so $max(\delta[A]^{\ell})$ is the meet of $\sigma[A]$ in $\mathcal{A}(\mathcal{D})$.
- ($\Leftarrow$) $\mathbb{P}^{\triangledown}$ is pattern structure is **equivalent** to say: $\forall A \subseteq \mathcal{G}$, $\sigma[A]$ has a meet $M \in \mathcal{A}(\mathcal{D})$. That is, for $A \subseteq \mathcal{G}$: $\forall S \in \mathcal{A}(\mathcal{D}) : S \subseteq \delta[A]^{\ell} \Leftrightarrow S \subseteq\downarrow M$. Particularly, for $S = \{d\}$ with $d \in \mathcal{D}$, we deduce that: $\forall d \in \delta[A]^{\ell} : d \in\downarrow M$. Thus, $\delta[A]^{\ell} \subseteq\downarrow M$. Moreover, since $M \subseteq \delta[A]^{\ell}$ ($M \in \sigma[A]^{\ell}$) and $\downarrow$ is a closure operator on $(\wp(\mathcal{D}), \subseteq)$ we have by monotony $\downarrow M \subseteq \delta[A]^{\ell} \subseteq\downarrow M$. We conclude that $\delta[A]^{\ell} =\downarrow M$ (note that $\downarrow \delta[A]^{\ell} = \delta[A]^{\ell}$). It follows that $M = max(\delta[A]^{\ell})$ that is $\delta[A]^{\ell} =\downarrow max(\delta[A]^{\ell})$. This concludes the equivalence.

Let us now define $int^{\triangledown}$ and $ext^{\triangledown}$. The previous proof shown that for $A \subseteq \mathcal{G}$ we have $int^{\triangledown}(A) = max(\delta[A]^{\ell}) = cov^{\mathcal{I}}(A)$ (The meet of $\sigma[A]$ is $max(\delta[A]^{\ell})$). For $ext^{\triangledown}$ operator, let $S \in \mathcal{A}(\mathcal{D})$. We have: $ext^{\triangledown}(S) = \{g \in \mathcal{G} \mid S \leq \sigma(g)\} = \{g \in \mathcal{G} \mid S \subseteq\downarrow \delta(g)\} = \{g \in \mathcal{G} \mid (\forall d \in S)\, d \sqsubseteq \delta(g)\} = \bigcap_{d \in S} ext(d) = \bigcap ext[S]$.

Let us show that $\mathbb{P}^{\triangledown}_{ext} = \{\bigcap S \mid S \subseteq \mathbb{P}_{ext}\}$. By definition of $ext^{\triangledown}$, the property $\mathbb{P}^{\triangledown}_{ext} \subseteq \{\bigcap S \mid S \subseteq \mathbb{P}_{ext}\}$ holds. For the inverse inclusion, it is sufficient to show that $\mathbb{P}_{ext} \subseteq \mathbb{P}^{\triangledown}_{ext}$ (since $(\mathbb{P}^{\triangledown}_{ext}, \subseteq)$ is closed under intersection). Let $A \in \mathbb{P}_{ext}$. $\exists d \in \mathcal{D}$ s.t. $A = ext(d)$. Since $\{d\} \in \mathcal{A}(\mathcal{D})$, and $ext^{\triangledown}(\{d\}) = ext(d) = A$. We conclude that $A \in \mathbb{P}^{\triangledown}_{ext}$ and $\mathbb{P}^{\triangledown}_{ext} = \{\bigcap S \mid S \subseteq \mathbb{P}_{ext}\}$. □

There is a completion that transforms any pattern setup to a pattern structure. This completion relies on the **Dedekind-MacNeille completion**.

Definition 9. *The family of subsets of $\mathcal{D}$ given by $DM(\mathcal{D}) = \{A^{\ell} \mid A \subseteq \mathcal{D}\}$ is a $\bigcap$-structure and is called the* ***Dedekind-MacNeille Completion*** *of $\underline{\mathcal{D}}$.*

Theorem 3. *Let $\mathbb{P} = (\mathcal{G}, \underline{\mathcal{D}}, \delta)$ be a* ***pattern setup****. The* ***Direct Completion*** *of $\mathbb{P}$ is the* ***pattern structure*** *denoted by $\mathbb{P}^{\blacktriangledown}$ and given by:*

$$\mathbb{P}^{\blacktriangledown} = (\mathcal{G}, (DM(\mathcal{D}), \subseteq), \gamma : g \mapsto \downarrow \delta(g))$$

Operators $ext^{\blacktriangledown}$ and $int^{\blacktriangledown}$ denote extent and intent of $\mathbb{P}^{\blacktriangledown}$ and are given by

$$(\forall S \in DM(\mathcal{D}))\; ext^{\blacktriangledown}(S) = \bigcap ext[S] \qquad (\forall A \subseteq \mathcal{G})\; int^{\blacktriangledown}(A) = cov(A) = \delta[A]^{\ell}$$

Moreover, we have $\mathbb{P}^{\blacktriangledown}_{ext} = \{\bigcap S \mid S \subseteq \mathbb{P}_{ext}\}$.

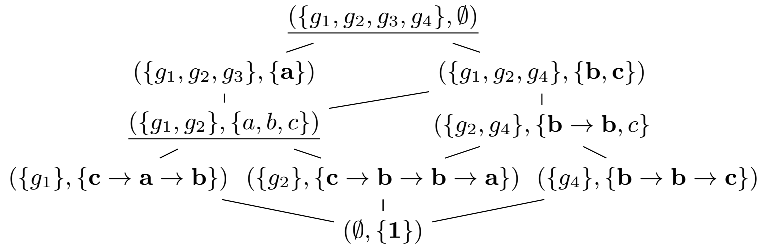

$\underline{(\{g_1, g_2, g_3, g_4\}, \emptyset)}$
$(\{g_1, g_2, g_3\}, \{\mathbf{a}\})$
$(\{g_1, g_2, g_4\}, \{\mathbf{b}, \mathbf{c}\})$
$\underline{(\{g_1, g_2\}, \{a, b, c\})}$
$(\{g_2, g_4\}, \{\mathbf{b} \to \mathbf{b}, c\}$
$(\{g_1\}, \{\mathbf{c} \to \mathbf{a} \to \mathbf{b}\})$
$(\{g_2\}, \{\mathbf{c} \to \mathbf{b} \to \mathbf{b} \to \mathbf{a}\})$
$(\{g_4\}, \{\mathbf{b} \to \mathbf{b} \to \mathbf{c}\})$
$(\emptyset, \{\mathbf{1}\})$

Fig. 3. Concept lattice $\underline{\mathfrak{B}}(\mathbb{P}^{\triangledown})$.

Proof. According to definition 9, $(DM(\mathcal{D}), \subseteq)$ is a complete lattice closed under intersection. Thus, the pattern setup $\mathbb{P}^{\blacktriangledown}$ is a pattern structure.

By definition, we have $int^{\blacktriangledown}(A) = \bigcap_{g \in A} \downarrow \delta(g) = \delta[A]^{\ell} = cov(A)$ since the meet in $\mathbb{P}^{\blacktriangledown}$ is $\bigcap$. For the extent operator $ext^{\blacktriangledown}$. Let $S \in DM(\mathcal{D})$, we have $ext^{\blacktriangledown}(S) = \{g \in \mathcal{G} \mid S \subseteq\downarrow \delta(g)\} = \{g \in \mathcal{G} \mid (\forall d \in S)\, d \sqsubseteq \delta(g)\} = \bigcap ext[S]$.

Let us show that $\mathbb{P}^{\blacktriangledown}_{ext} = \{\bigcap S \mid S \subseteq \mathbb{P}_{ext}\}$. Thanks to $ext^{\blacktriangledown}$ definition, property $\mathbb{P}^{\triangledown}_{ext} \subseteq \{\bigcap S \mid S \subseteq \mathbb{P}_{ext}\}$ holds. For the inverse inclusion, it is sufficient to show that $\mathbb{P}_{ext} \subseteq \mathbb{P}^{\blacktriangledown}_{ext}$ (since $(\mathbb{P}^{\blacktriangledown}_{ext}, \subseteq)$ is closed under intersection). Let $A \in \mathbb{P}_{ext}$, $\exists d \in \mathcal{D}$ such that $A = ext(d)$. We have $ext^{\blacktriangledown}(\{d\}^{\ell}) = ext^{\blacktriangledown}(\downarrow d) = \{g \in \mathcal{G} \mid\downarrow d \subseteq\downarrow \delta(g)\} = \{g \in \mathcal{G} \mid d \sqsubseteq \delta(g)\} = ext(d) = A$. We conclude that $A \in \mathbb{P}^{\blacktriangledown}_{ext}$ and $\mathbb{P}^{\blacktriangledown}_{ext} = \{\bigcap S \mid S \subseteq \mathbb{P}_{ext}\}$, which completes the proof. □

Example 6. Fig. 3 depicts the concept lattice $\underline{\mathfrak{B}}(\mathbb{P}^{\triangledown})$ associated to the *antichain completion* of the pattern hyper-structure $\mathbb{P}$ considered in Fig. 2 (*i.e.,* the description space is augmented with the top element $\mathbf{1}$). One can see that there are two new (underlined) extents $\{g_1, g_2\}$ and $\{g_1, g_2, g_3, g_4\}$ in $\mathbb{P}^{\triangledown}_{ext} \backslash \mathbb{P}_{ext}$. For instance, consider the intent of $\{g_1, g_2\}$ in the completion, each pattern d has extent $ext(d) \supsetneq \{g_1, g_2\}$. Extent $\{g_1, g_2, g_3, g_4\}$ is **non coverable** in $\mathbb{P}$ and thus $int^{\triangledown}(\{g_1, g_2, g_3, g_4\}) = max(cov(\{g_1, g_2, g_3, g_4\})) = max(\emptyset) = \emptyset$.

5 Conclusion

In this paper, we have developed a better understanding of pattern setups, a framework that models pattern spaces relying only on a poset. Next, we studied the usual transformation of pattern setups to pattern structures using antichains. We have shown that such a completion does not always produce a pattern structure unless the pattern setup is a pattern hyper-structure. Finally, we have shown that a natural completion of a pattern setup to a pattern structure exists thanks to the Dedekind-MacNeille completion. This work paves the way to answer an important question: *How to enumerate extents of a pattern setup without "visiting" the whole set of extents of its associated completion to a pattern structure?*

Acknowledgments

This work has been partially supported the Association Nationale Recherche Technologie (ANRT) French program. Sections 1, 2 and 3.1 were written by

Sergei O. Kuznetsov supported by the Russian Science Foundation under grant 17-11-01276 and performed at St. Petersburg Department of Steklov Mathematical Institute of Russian Academy of Sciences, Russia.

References

1. Agrawal, R., Srikant, R.: Mining sequential patterns. In: Data Engineering. pp. 3–14. IEEE (1995)
2. Baixeries, J., Kaytoue, M., Napoli, A.: Characterizing functional dependencies in formal concept analysis with pattern structures. Ann. Math. Artif. Intell. **72**(1-2), 129–149 (2014)
3. Belfodil, A., Kuznetsov, S.O., Robardet, C., Kaytoue, M.: Mining convex polygon patterns with formal concept analysis. In: IJCAI. pp. 1425–1432 (2017)
4. Boldi, P., Vigna, S.: On the lattice of antichains of finite intervals. CoRR **abs/1510.03675** (2016), `https://arxiv.org/abs/1510.03675`
5. Boley, M., Horváth, T., Poigné, A., Wrobel, S.: Listing closed sets of strongly accessible set systems with applications to data mining. Theor. Comput. Sci. **411**(3), 691–700 (2010)
6. Buzmakov, A., Egho, E., Jay, N., Kuznetsov, S.O., Napoli, A., Raïssi, C.: On mining complex sequential data by means of FCA and pattern structures. Int. J. General Systems **45**(2), 135–159 (2016)
7. Codocedo, V., Bosc, G., Kaytoue, M., Boulicaut, J., Napoli, A.: A proposition for sequence mining using pattern structures. In: ICFCA. LNCS (10308). 106–121. (2017)
8. Crampton, J., Loizou, G.: The completion of a poset in a lattice of antichains. International Mathematical Journal **1**(3), 223–238 (2001)
9. Ganter, B., Wille, R.: Formal Concept Analysis. Springer (1999)
10. Ganter, B., Kuznetsov, S.O.: Pattern structures and their projections. In: ICCS. LNCS (2120). 129–142. (2001)
11. Garriga, G.C., Kralj, P., Lavrac, N.: Closed sets for labeled data. Journal of Machine Learning Research **9**, 559–580 (2008)
12. Kaytoue, M., Kuznetsov, S.O., Napoli, A.: Revisiting Numerical Pattern Mining with Formal Concept Analysis. In: IJCAI. pp. 1342–1347 (2011)
13. Kuznetsov, S.O.: Learning of simple conceptual graphs from positive and negative examples. In: PKDD. LNCS (1704). 384–391., vol. 1704, pp. 384–391 (1999)
14. Kuznetsov, S.O.: Pattern structures for analyzing complex data. In: RSFDGrC. LNCS (5908). 33–44. (2009)
15. Liu, B., Hsu, W., Ma, Y.: Integrating classification and association rule mining. In: KDD. pp. 80–86 (1998)
16. Lumpe, L., Schmidt, S.E.: Pattern structures and their morphisms. In: CLA. CEUR Workshop Proceedings, vol. 1466, pp. 171–179 (2015)
17. Pawlak, Z.: Rough sets. International Journal of Parallel Programming **11**(5), 341–356 (1982)
18. Roman, S.: Lattices and Ordered Sets. Springer New York (2008)
19. Zaki, M.J.: SPADE: an efficient algorithm for mining frequent sequences. Machine Learning **42**(1/2), 31–60 (2001)

A New Method for Inheriting Canonicity Test Failures in Close-by-One Type Algorithms

Simon Andrews[0000−0003−2094−7456]

Conceptual Structures Research Group
Communication and Computing Research Centre
Department of Computing
Faculty of Arts, Computing, Engineering and Sciences
Sheffield Hallam University, Sheffield, UK
s.andrews@shu.ac.uk

Abstract. Close-by-One type algorithms are efficient algorithms for computing formal concepts. They use a mathematical canonicity test to avoid the repeated computation of the same concept, which is far more efficient than methods based on searching. Nevertheless, the canonicity test is still the most labour intensive part of Close-by-One algorithms and various means of avoiding the test have been devised, including the ability to inherit test failures at the next level of recursion. This paper presents a new method for inheriting canonicity test failures in Close-by-One type algorithms. The new method is simpler than the existing method and can be amalgamated with other algorithm features to further improve efficiency. The paper recaps an existing algorithm that does not feature test failure inheritance and an algorithm that features the existing method. The paper then presents the new method and a new algorithm that incorporates it. The three algorithms are implemented on a 'level playing field' with the same level of optimisation. Experiments are carried out on the implemented algorithms, using a representative range of data sets, to compare the number of inherited canonicity test failures and the computation times. It is shown that the new algorithm, incorporating the new method of inheriting canonicity test failures, gives the best performance.

Keywords: Formal Concept Analysis · FCA · FCA algorithms · Computing formal concepts · Canonicity test · Inheriting canonicity test failures· Close-by-One · FCbO · In-Close

1 Introduction

In the development o fast algorithms to compute formal concepts, the discovery of the so-called 'canonicity test', whereby the attributes in a concept could be examined to determine its newness in the computation, gave rise to the original Close-by-One (CbO) algorithm [8]. The canonicity test has proved to be fundamental in the efficient computation of formal concepts, being far more efficient than previous methods of determining the newness of a concept based on

© Dmitry I. Ignatov, Lhouari Nourine (Eds.): CLA 2018, pp. 255–266,
ISBN 978–80–244–5328–6, Department of Computer Science, Palacký University Olomouc, 2018.

searching, and was integral to the subsequent CbO algorithm presented in [6]. Nevertheless, the canonicity test is still the most labour intensive part of CbO-type algorithms and various means of avoiding or improving the test have been devised, giving rise to a number of advances in CbO-type algorithms including FCbO [7, 9], In-Close2 [3] and In-Close4 [4]. FCbO introduced a combined 'breadth and depth' approach to computation that allowed child concepts to fully inherit their parent's attributes. In-Close2 then added a modified, 'partial-closure', canonicity test to reduce the computation required in the test. FCbO also introduced a technique whereby failed canonicity tests could be inherited, thereby avoiding many canonicity tests. In-Close4 made use of empty intersections between the current concept extent and attribute-extents in the computation to also avoid canonicity tests.

This paper describes a new method of inheriting failed canonicity tests that is simpler than the method used by FCbO. Furthermore, the method can be amalgamated with existing efficiency features to further improve performance.

The rest of this paper is structured as follows: The paper will use the algorithm In-Close4 [4] as the framework in which to incorporate the new inheritance method, so Section 2 is a recap of that algorithm. Section 3 is a recap of the FCbO algorithm, describing its method of inheriting failed canonicity tests. Section 4 describes the new method of inheriting failed canonicity tests and incorporates it into In-Close4, creating a new algorithm, In-Close5. It should be noted that In-Close1, In-Close2 and In-Close3 are previous versions of In-Close, as presented in [2]. Section 5 describes the implementation of In-Close4, FCbO and In-Close5 on a 'level playing field' using the same programming optimisations. Section 5 also shows how the new method of inheriting failed canonicity tests can be amalgamated with existing efficiency features to further improve performance. Section 6 presents a series of experiments and results, comparing the performance of In-Close4, FCbO and In-Close5. Finally, Section 7 provides some concluding remarks and ideas for further work.

2 Recap of the In-Close4 Algorithm

Below is a recap of the In-Close4 algorithm, as presented in [4]. In-Close4 combines the efficiency of a partial-closure canonicity test [2] with full inheritance of the parent intent. The full inheritance is achieved by adapting and incorporating the combined breadth-first and depth-first approach of FCbO [7, 9]. The main cycle is completed before passing to the next level, so that all the attributes of a parent intent can be passed down to the next level. Child intents only have to be finished off by adding attributes that are not in the parent intent. During the main cycle, whilst the current intent is being closed, new extents that pass the canonicity test are stored in a queue, similar to the queue in FCbO, to be processed after the main cycle has completed. In-Close4 also makes use of empty intersections when the current extent is intersected with the next attribute-extent (next column) in the formal context: empty intersections are inherited so that they can be skipped at subsequent levels in the computation and, whenever an empty intersection occurs, the algorithm forgoes the canonicity test.

The In-Close4 algorithm is invoked with an initial pair $(A, B) = (X, \emptyset)$, where A is a set of objects (extent) and B is a set of attributes (intent) and X is the set of all objects in the formal context, and initial attribute $y = 0$. Y is the set of all attributes in the formal context and Y_j is the set of all attributes up to (but not including) j. The algorithm is also invoked with an empty set of attributes, $P = \emptyset$, in which to store subsequent empty intersections.

Note that forgoing the canonicity test after an empty intersection means that the algorithm is incomplete, in that it will not compute the concept $(Y, \emptyset)$. However, it is a simple task to add it afterwards, if it exists: If $Y^{\downarrow} = \emptyset$ then add $(\emptyset, Y)$ to the set of computed concepts.

In-Close4

ComputeConceptsFrom($(A, B), y, P$)

```
1  for $j \leftarrow y$ upto $n - 1$ do
2      if $j \notin B$ and $j \notin P$ then
3          $C \leftarrow A \cap \{j\}^{\downarrow}$
4          if $C \neq \emptyset$ then
5              if $C = A$ then
6                  $B \leftarrow B \cup \{j\}$
7              else
8                  if $B \cap Y_j = C^{\uparrow j}$ then
9                      PutInQueue($C, j$)
10         else
11             $P \leftarrow P \cup \{j\}$
12 ProcessConcept($(A, B)$)
13 $Q \leftarrow P$
14 while GetFromQueue($C, j$) do
15     $D \leftarrow B \cup \{j\}$
16     ComputeConceptsFrom($(C, D), j + 1, Q$)
```

A line by line explanation of In-Close4 is as follows:

Line 1 - Iterate across the formal context, from a starting attribute y up to attribute $n - 1$, where n is the number of attributes in the context.

Line 2 - Skip attributes already in B. Because intents inherit all of their parent's attributes, these can be skipped. Also skip any attributes in P as these are inherited empty intersections - if the parent extent resulted in an empty intersection, so will its children since they are all subsets of the parent.

Line 3 - Form an extent, C, by intersecting the current extent, A, with the next attribute-extent (column of objects) in the context.

Line 4 - If the extent, C, is not empty...

Line 5 - If the extent, C, equals the extent of the concept whose intent is currently being closed, A, then...

Line 6 - ...add the current attribute, j, to the intent being closed, B.

Line 7 - Otherwise, test the canonicity using the partial-closure canonicity test [1]: $^{\uparrow}$ is the standard closure operator in FCA and $^{\uparrow_j}$ is a modification meaning "close up to, but not including, attribute j".

Line 8 - If the canonicity test is passed...

Line 9 - ...place the new extent, C, and the location where it was found, j, in a queue for later processing.

Line 10 - If the extent, C, is empty...

Line 11 - ... add the current attribute to P so that the empty intersection can be inherited.

Line 12 - Pass concept (A, B) to the notional procedure `ProcessConcept` to process it in some way (for example, storing it in a data base of concepts).

Line 13 - Store P in Q ready to pass the attributes resulting in empty intersections to the next level.

Line 14 - The queue is processed by obtaining from the queue each new extent and the location it was found.

Line 15 - Each new partial intent, D, inherits all the attributes from its completed parent intent, B, along with the attribute, j, where its extent was found.

Line 16 - Recursively call `ComputeConceptsFrom` to compute child concepts from $j + 1$ and to complete the intent D.

3 Recap of the FCbO Algorithm

Below is a recap of the FCbO algorithm [7, 9] as presented in [2]. FCbO introduced the feature of inherited canonicity test failures to improve the performance of CbO-type algorithms, along with the combined breadth/depth first approach to enable full inheritance of parent intents. The inheritance of test failures is achieved by recording intents that are not canonical as N^js, where j is the current attribute, thus enabling subsequent levels to compare these failed intents against the current one and thus avoid the computation of a repeated concept without the need for the original canonicity test. FCbO is invoked with the initial concept $(A, B) = (X, X^{\uparrow})$, initial attribute $y = 0$ and a set of empty N^ys, $\{N^y = \emptyset \mid y \in Y\}$.

Line 1 - Pass concept (A, B) to the notional procedure `ProcessConcept` to process it in some way (for example, storing it in a set of concepts).

Line 2 - Iterate across the context, from starting attribute y up to attribute $n - 1$.

Line 3 - M^j is set to the latest intent that failed the canonicity test at attribute j, N^j.

Line 4 - Skip attributes in B and those that have an inherited record of failure.

FCbO

```
ComputeConceptsFrom($(A, B), y, \{N^y \mid y \in Y\}$)
1  ProcessConcept($(A, B)$)
2  for $j \leftarrow y$ upto $n - 1$ do
3      $M^j \leftarrow N^j$
4      if $j \notin B$ and $N^j \cap Y_j \subseteq B \cap Y_j$ then
5          $C \leftarrow A \cap \{j\}^{\downarrow}$
6          $D \leftarrow C^{\uparrow}$
7          if $B \cap Y_j = D \cap Y_j$ then
8              PutInQueue ($(C, D), j$)
9          else
10             $M^j \leftarrow D$
11 while GetFromQueue($(C, D), j$) do
12     ComputeConceptsFrom($(C, D), j + 1, \{M^y \mid y \in Y\}$)
```

Line 5 - Otherwise, form an extent, C, by intersecting the current extent, A, with the next column of objects in the context.

Line 6 - Close the extent to form an intent, D.

Line 7 - Perform the canonicity test.

Line 8 - If the concept is a new one, store it in a queue along with the attribute it was computed at.

Line 10 - Otherwise set the record of failure for attribute j, M^j, to the intent that failed the canonicity test.

Line 11 - Get each stored concept from the queue...

Line 12 - ...and pass it to the next level, along with the stored starting attribute for the next level and the failed intents from this level.

4 A New Method of Inheriting Failed Canonicity Tests

The method of inheriting failed canonicity tests employed by FCbO requires the manipulation and storage of a two-dimensional array to represent intents that fail the canonicity test. A total of n intents are required, and, although the use of pointers in a optimised implementation avoids the need for copying intents, they still need to be computed and stored. This results in computational overheads so that, even though a significant number of canonicity test are avoided [9], algorithms such as In-Close4 are still able to outperform FCbO [2,4].

However, it is possible to obtain the inheritance of failed canonicity tests with a simpler method. Firstly consider the criteria for failure in In-Close4: the test will fail if there exists an attribute in $C^{\uparrow_j}$ that is not in $B \cap Y_j$. In other words, when there is an attribute before j (but not in the current intent, B) who's attribute-extent contains the extent, C - in which case the extent, C, will already have been computed. Now consider the starting attribute, y, for the current cycle (Line 1 of In-Close4). Let us say, in a failed canonicity test, that the smallest attribute in $C^{\uparrow j}$ that is not in $B \cap Y_j$ is i. If $i \geq y$ then an extent, H, where $C \subseteq H$, will have been discovered in the current cycle at i (and be waiting in the current queue). And there may be other extents, discovered after i but before j that are also supersets of C and also in the queue. Thus, if $i \geq y$, the current attribute, j, will be required at the next level to be examined by the children in the queue: C may be canonical with respect to one of the children or j may be an attribute in the intent of a child and thus required to be added. However, if $i < y$, the concept with extent C and its children will have already been computed and processed. Thus no children in the current queue, or subsequent children, need examine j. In other words, if $i < y$ then j can be inherited as a canonicity test failure - all subsequent children can skip j in the cycle. All that is required is to maintain a set of such attributes that can be passed down to the next level in the algorithm.

The new algorithm, In-Close5, below, is In-Close4 with the new method of inheriting failed canonicity tests added. It is invoked in the same way as In-Close4 but with the addition of an initially empty set of attributes, $N = \emptyset$, in which to store canonicity test failures.

In-Close5

ComputeConceptsFrom($(A, B), y, P, N$)

```
for $j \leftarrow y$ upto $n - 1$ do
    if $j \notin B$ and $j \notin P$ and $j \notin N$ then
        $C \leftarrow A \cap \{j\}^{\downarrow}$
        if $C \neq \emptyset$ then
            if $C = A$ then
                $B \leftarrow B \cup \{j\}$
            else
                if $B \cap Y_j = C^{\uparrow j}$ then
                    PutInQueue($C, j$)
                else
                    if $min(C^{\uparrow j}) < y$ then
                        $N \leftarrow N \cup \{j\}$
        else
            $P \leftarrow P \cup \{j\}$
ProcessConcept($(A, B)$)
$Q \leftarrow P$
$M \leftarrow N$
while GetFromQueue($C, j$) do
    $D \leftarrow B \cup \{j\}$
    ComputeConceptsFrom($(C, D), j + 1, Q, M$)
```

The new lines in In-Close5 are as follows:

Line 2 - As well as skipping inherited attributes in the intent, $j \notin B$, and inherited empty intersections, $j \notin P$, the algorithm now also skips inherited canonicity test failures, $j \notin N$.

Line 11 - If the canonicity test (Line 8) is failed, a test is carried out comparing the smallest attribute in $C^{\uparrow_j}$ with y. If the attribute is smaller than y then...

Line 12 - ...j is added to the set of canonicity test failures, N.

Line 17 - Store N in M ready to pass the canonicity test failures to the next level.

5 Implementation

The three algorithms, In-Close4, FCbO and In-Close5, were implemented in ANCII C using the same data structures, data pre-processing and level of optimisation to create a 'level playing field' for comparing their performance. The key optimisations are described below.

The use of Bit-Arrays Implementations of CbO-type algorithms, such as In-Close and FCbO, typically use a bit-array to represent the formal context. This allows operations on the formal context, such as closure operations, to be implemented using bit-wise operators in the manner of fine-grained parallel processing. In a typical 64-bit architecture, this means that 64 cells of the formal context can be operated on simultaneously. Using bits to represent cells of the formal context also allows more of the context to be retained in cache memory.

Using a Local Boolean Copy of the Current Intent Typical implementations of CbO-type algorithms maintain a global data structure to store integer representations of concept intents (integers mapping to formal attributes) but, at the same time, also use a Boolean (bit-array) representation of the current intent to facilitate an efficient implementation of the test for inherited attributes, $j \notin B$.

Efficient Implementation of the Partial-Closure Canonicity Test in In-Close Algorithms In practice, it is not necessary to always close the new extent up to the current attribute. It is only necessary to find the *first instance* where $B \cap Y_j$ and $C^{\uparrow_j}$ do not agree. Thus failure is typically detected before j is reached, thus saving additional time. In FCbO, however, a full-closure, $C^{\uparrow}$ is *always* required because, if the test is passed, it provides the closure of the concept intent, or, if the test is failed, it provides the failed intent to be stored in M^j. In In-Close, new concept intents are closed at the next level, during the main cycle, whenever $C = A$ by $B \leftarrow B \cup \{j\}$ (Lines 5 and 6 of In-Close4, for example). Furthermore, given that the test $C = A$ is provided at no computational cost, as a by-product of the intersection in $C \leftarrow A \cap \{j\}^{\downarrow}$, the overheads of the closure are close to zero. This also means that savings are made by In-Close algorithms when canonicity tests succeed. Here, the partial closure, $C^{\uparrow_j}$ is carried out up to j, compared to the full closure, $C^{\uparrow}$, in FCbO.

Amalgamation of Efficiency Features in In-Close5 In implementation, the set of inherited empty intersections, the set of inherited canonicity test failures and the local, Boolean, copy of the current intent can be amalgamated into a single bit-array, in effect reducing the test in Line 2 of In-Close5, $j \notin B$ And $j \notin P$ And $j \notin N$ to a single test, $j \notin Z$, where $Z = B \cup P \cup N$. Lines 6, 12 and 14 will all become $Z \leftarrow Z \cup \{j\}$, thus updating the same bit-array in the implementation (of course the update of the global set of intents in the implementation, required by Line 6, remains unchanged). Amalgamating the three sets of attributes also means there are overhead savings made from reduced parameter passing.

6 Evaluation of Performance

In this section, In-Close4, FCbO and In-Close5 are evaluated by comparing their performance over a varied range of data sets. The experiments are divided into three groups: 1) real data sets, 2) artificial data sets, and 3) randomised data sets. In each case, the time taken to compute all formal concepts is measured along with the number of canonicity tests carried out.

The experiments were conducted on a standard 64-bit Intel architecture, using a PC with an Intel Core i7-2600 3.40GHz CPU and 8GB of RAM. To cater for any inconsistency of system performance, due to background system processes, for example, each experiment was conducted multiple times and the average time taken for each.

Real Data Set Experiments. Four real data sets were used in the experiments: *Mushroom*, *Adult* and *Internet Ads*, taken from the UCI Machine Learning Repository [5] and *Student*, an anonymised data set from an internal student experience survey carried out at Sheffield Hallam University, UK. The data sets were selected to represent a broad range of features, in terms of size and density, and the UCI ones, in particular, are well known and used in FCA work.

The results of the experiments are given in Table 1 (timings) and Table 2 (canonicity tests).

In-Close5 was fastest for the *Mushroom*, *Adult* and *Student* data sets, and equal fastest, with In-Close4, for the *Internet Ads* data set. In-Close5 used the fewest canonicity tests for the *Adult* and *Internet Ads* data sets and was not far behind FCbO for the *Mushroom* and *Student* data sets.

Artificial Data Set Experiments. Artificial data sets were used that, although randomised, the randomisation was constrained by properties of real data sets, such as many-valued attributes having a pre-defined number of unique

Table 1. Real data set results (timings in seconds).

Data set	Mushroom	Adult	Internet Ads	Student
$\vert G\vert \times \vert M\vert$	$8,124 \times 126$	$32,561 \times 124$	$3,279 \times 1,565$	587×145
Density	17.36%	11.29%	0.97%	24.50%
#Concepts	233,116	1,388,469	16,570	$22,760,243$
FCbO	0.23	1.46	0.21	8.80
In-Close4	0.19	0.88	**0.07**	4.65
In-Close5	**0.18**	**0.85**	**0.07**	**4.31**

Table 2. Real data set results (canonicity tests).

Data set	Mushroom	Adult	Internet Ads	Student
FCbO	**331,106**	2,029,933	363,568	**40,630,663**
In-Close4	429,974	1,707,707	91,029	53,162,649
In-Close5	332,449	**1,667,052**	**67,715**	41,048,752

values. Three data sets, *M7X10G120K*, *M10X30G120K* and *T10I4D100K*, were used to provide a range of features in terms of size and density.

The timing results of the artificial data set experiments are given in Table 3 and the comparison of the number of canonicity tests carried out is given in Table 4. For all three data sets, In-Close5 was quickest and performed the fewest canonicity tests.

Table 3. Artificial data set results (timings in seconds).

Data set	M7X10G120K	M10X30G120K	T10I4D100K
$\vert G\vert \times \vert M\vert$	$120,000 \times 70$	$120,000 \times 300$	$100,000 \times 1,000$
Density	10.00%	3.33%	1.01%
#Concepts	1,166,326	4,570,493	2,347,376
FCbO	1.35	15.45	23.83
In-Close4	0.77	5.60	6.56
In-Close5	**0.69**	**5.35**	**5.81**

Random Data Set Experiments. Three series of random data experiments were carried out, testing the effect of variation of the number of attributes, context density, and number of objects, respectively:

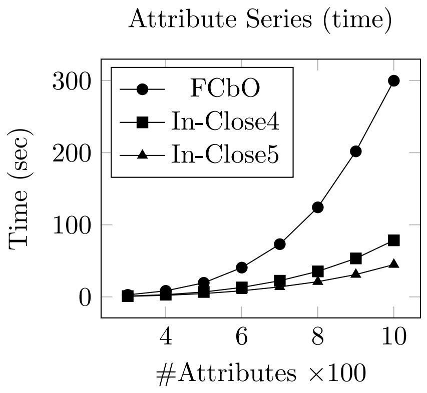
- *Attributes series* - with 5% density and 5,000 objects, the number of attributes was varied between 300 and 1,000. The number of concepts varied from approximately 1,000,000 to 22,000,000.

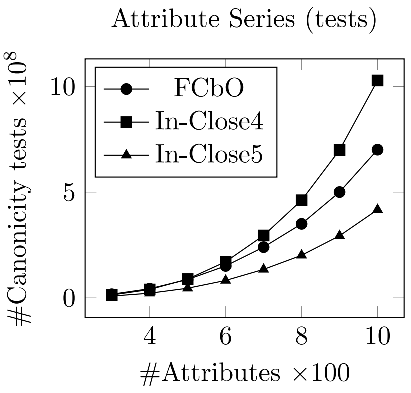
Table 4. Artificial data set results (canonicity tests).

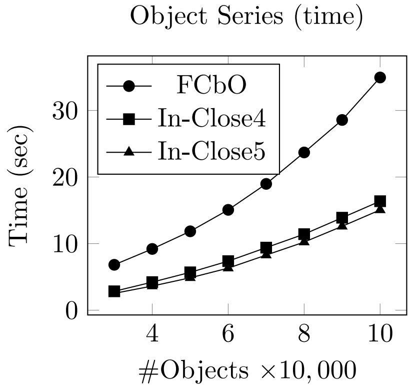
Data set	M7X10G120K	M10X30G120K	T10I4D100K
FCbO	4,640,906	167,814,522	75,281,105
In-Close4	2,360,015	29,686,007	21,262,544
In-Close5	**2,339,951**	**26,593,944**	**14,907,484**

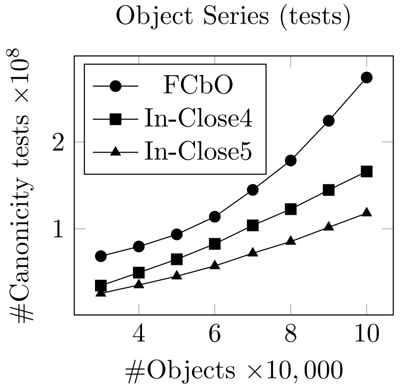
- *Objects series* - with 5% density and 200 attributes, the number of objects was varied between 30,000 and 100,000. The number of concepts varied from approximately 4,000,000 to 22,000,000.
- *Density series* - with 200 attributes and 10,000 objects, the density of 1s in the context was varied between 3 and 10%. The number of concepts varied from approximately 200,000 to 19,000,000.

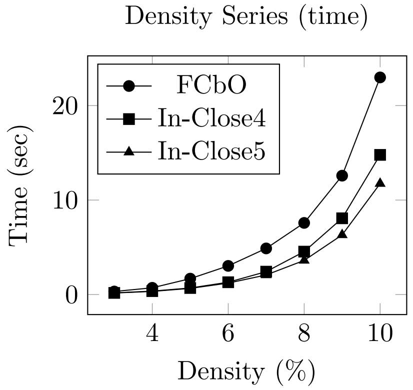
The results of the random data set timings are shown in the plots below. In all three series, In-Close5 performed the fewest canonicity tests and was fastest. It is interesting to note that In-Close4 often performed fewer canonicity tests than FCbO (particularly apparent in the *Object* series). One might therefore deduce that the *Object* series data sets gave rise to large numbers of empty intersections - perhaps not surprising as the number of objects is increased at a relatively low density in a randomised formal context.

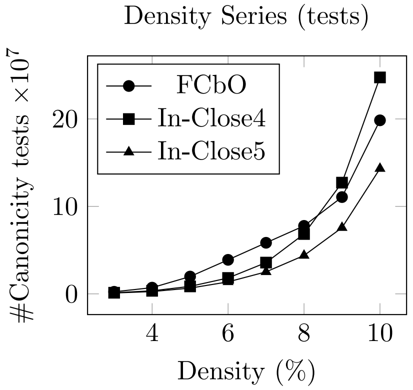

Attribute Series (time)


Attribute Series (tests)









7 Conclusions

In conclusion, the performance of In-Close5 clearly demonstrates the efficiency savings provided by the new method of inheriting canonicity test failures when its results are compared to those of In-Close4 (the same algorithm but without canonicity test failure inheritance). In-Close5 clearly outperforms FCbO, the algorithm that features the existing method of inheriting canonicity test failures. Although FCbO's method inherits more test failures than the new method, the simplicity of the new method warrants its attention as a useful contribution to the area. It was shown in In-Close3 [2] that incorporating FCbO's method gave little improvement of performance, due to the computational overheads of implementing it, whereas it is show here that the incorporation of the new method does improve performance significantly.

An implementation of In-Close5 is available, free and open source, at `https://sourceforge.net/projects/inclose/`.

References

1. Andrews, S.: A partial-closure canonicity test to increase the efficiency of cbo-type algorithms. In: Proceedings of the 21st International Conference on Conceptual Structures. pp. 37–50. Springer (2014)
2. Andrews, S.: A best-of-breed approach for designing a fast algorithm for computing fixpoints of galois connections. Information Sciences **295**, 633–649 (2015)
3. 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 - Proceedings of the 19th International Conference on Conceptual Structures (ICCS). pp. 50–62. Springer (2011)
4. Andrews, S.: Making use of empty intersections to improve the performance of cbo-type algorithms. In: Proceedings of the 14th International Conference on Formal Concept Analysis. pp. 56–71. Springer (2017)
5. Frank, A., Asuncion, A.: UCI machine learning repository: http://archive.ics.uci.edu/ml (2010)
6. Krajca, P., Outrata, J., Vychodil, V.: Parallel recursive algorithm for FCA. In: Belohavlek, R., Kuznetsov, S. (eds.) Proceedings of Concept Lattices and their Applications, 2008. pp. 71–82 (2008)
7. Krajca, P., Vychodil, V., Outrata, J.: Advances in algorithms based on CbO. In: Kryszkiewicz, M., Obiedkov, S. (eds.) CLA 2010. pp. 325–337. University of Sevilla (2010)
8. Kuznetsov, S.O.: Mathematical aspects of concept analysis. Mathematical Science **80**(2), 1654–1698 (1996)
9. Outrata, J., Vychodil, V.: Fast algorithm for computing fixpoints of Galois connections induced by object-attribute relational data. Information Sciences **185**(1), 114–127 (Feb 2012). https://doi.org/10.1016/j.ins.2011.09.023

The Distributive, Graded Lattice of $\mathcal{EL}$ Concept Descriptions and its Neighborhood Relation

Francesco Kriegel[0000−0003−0219−0330]

Institute of Theoretical Computer Science,
Technische Universität Dresden, Dresden, Germany
francesco.kriegel@tu-dresden.de

Abstract. For the description logic $\mathcal{EL}$, we consider the neighborhood relation which is induced by the subsumption order, and we show that the corresponding lattice of $\mathcal{EL}$ concept descriptions is distributive, modular, graded, and metric. In particular, this implies the existence of a rank function as well as the existence of a distance function.

Keywords: Description logic · Distributive lattice · Modular lattice · Graded lattice · Metric lattice · Rank function · Distance function · Neighborhood relation · Upper neighbor · Lower neighbor

1 Introduction

Description Logics [3] are a family of well-founded languages for knowledge representation with a strong logical foundation as well as a widely explored hierarchy of decidability and complexity of common reasoning problems. The several reasoning tasks allow for an automatic deduction of implicit knowledge from given explicitly represented facts and axioms, and many reasoning algorithms have been developed. Description Logics are utilized in many different application domains, and in particular provide the logical underpinning of *Web Ontology Language (OWL)* [7] and its profiles.

$\mathcal{EL}$ is an example of a description logic with tractable reasoning problems, i.e., the usual inference problems can be decided in polynomial time, cf. Baader, Brandt, and Lutz in [2]. From a perspective of *lattice theory*, $\mathcal{EL}$ has not been deeply explored yet. Of course, it is apparent that the subsumption $\sqsubseteq$ with respect to some TBox $\mathcal{T}$ constitutes a quasi-order. Furthermore, in description logics supremums in the corresponding ordered set are usually called *least common subsumers*, and these exist in all cases if either no TBox is present, or if greatest fixed-point semantics are applied. Apart from that not much is known about the lattice of $\mathcal{EL}$ concept descriptions. In this document, we shall consider the neighborhood relation which is induced by the subsumption order, and we shall show that the lattice of $\mathcal{EL}$ concept descriptions is distributive, modular, graded, and metric. In particular, this implies the existence of a rank function as well as the existence of a distance function.

2 The Description Logic $\mathcal{EL}$

In this section we shall introduce the syntax and semantics of the light-weight description logic $\mathcal{EL}$ [3,2]. Throughout the whole document assume that Σ is a signature, i.e.,

© Dmitry I. Ignatov, Lhouari Nourine (Eds.): CLA 2018, pp. 267–278,
ISBN 978–80–244–5328–6, Department of Computer Science, Palacký University
Olomouc, 2018.

$\Sigma = \Sigma_\mathsf{C} \uplus \Sigma_\mathsf{R}$ is a disjoint union of a set Σ_C of *concept names* and a set Σ_R of *role names*. An *$\mathcal{EL}$ concept description*over Σ is a term that is constructed by means of the following inductive rule where $A \in \Sigma_\mathsf{C}$ and $r \in \Sigma_\mathsf{R}$.

$$C ::= \top \mid A \mid C \sqcap C \mid \exists\, r.\, C$$

The set of all $\mathcal{EL}$ concept descriptions over Σ is denoted by $\mathcal{EL}(\Sigma)$. The *size* $||C||$ of an $\mathcal{EL}$ concept description C is the number of nodes in its syntax tree, and we can recursively define it as follows: $||\top|| \coloneqq 1$, $||A|| \coloneqq 1$, $||C \sqcap D|| \coloneqq ||C|| + 1 + ||D||$, and $||\exists\, r.\, C|| \coloneqq 1 + ||C||$. A *concept inclusion* is an expression $C \sqsubseteq D$ where both the *premise* C as well as the *conclusion* D are concept descriptions. A *terminological box* (abbrv. *TBox*) is a finite set of concept inclusions.

An *interpretation* $\mathcal{I} \coloneqq (\Delta^\mathcal{I}, \cdot^\mathcal{I})$ over Σ consists of a non-empty set $\Delta^\mathcal{I}$, called the *domain*, and an *extension function* $\cdot^\mathcal{I}$ that maps concept names $A \in \Sigma_\mathsf{C}$ to subsets $A^\mathcal{I} \subseteq \Delta^\mathcal{I}$ and maps role names $r \in \Sigma_\mathsf{R}$ to binary relations $r^\mathcal{I} \subseteq \Delta^\mathcal{I} \times \Delta^\mathcal{I}$. Then, the extension function is canonically extended to all $\mathcal{EL}$ concept descriptions by the following definitions.

$$\bot^\mathcal{I} \coloneqq \emptyset \qquad \top^\mathcal{I} \coloneqq \Delta^\mathcal{I} \qquad (C \sqcap D)^\mathcal{I} \coloneqq C^\mathcal{I} \cap D^\mathcal{I}$$
$$(\exists\, r.\, C)^\mathcal{I} \coloneqq \{\, d \in \Delta^\mathcal{I} \mid \exists\, e \in \Delta^\mathcal{I} \colon (d,e) \in r^\mathcal{I} \text{ and } e \in C^\mathcal{I} \,\}$$

A concept inclusion $C \sqsubseteq D$ is *valid* in $\mathcal{I}$ if $C^\mathcal{I} \subseteq D^\mathcal{I}$. We then also refer to $\mathcal{I}$ as a *model* of $C \sqsubseteq D$, and denote this by $\mathcal{I} \models C \sqsubseteq D$. Furthermore, $\mathcal{I}$ is a *model* of a TBox $\mathcal{T}$, symbolized as $\mathcal{I} \models \mathcal{T}$, if each concept inclusion in $\mathcal{T}$ is valid in $\mathcal{I}$. The relation $\models$ is lifted to TBoxes as follows. A concept inclusion $C \sqsubseteq D$ is *entailed* by a TBox $\mathcal{T}$, denoted as $\mathcal{T} \models C \sqsubseteq D$, if each model of $\mathcal{T}$ is a model of $C \sqsubseteq D$ too. We then also say that C is *subsumed* by D with respect to $\mathcal{T}$. A TBox $\mathcal{T}$ *entails* a TBox $\mathcal{U}$, symbolized as $\mathcal{T} \models \mathcal{U}$, if $\mathcal{T}$ entails each concept inclusion in $\mathcal{U}$, or equivalently if each model of $\mathcal{T}$ is also a model of $\mathcal{U}$. In case $\mathcal{T} = \emptyset$ we may ommit the prefix "$\emptyset \models$". However, then we have to carefully interpret an expression $C \sqsubseteq D$—it either just denotes a concept inclusion, i.e., an axiom, without stating where it is valid; or it expresses that C is subsumed by D (w.r.t. $\emptyset$), i.e., $C^\mathcal{I} \subseteq D^\mathcal{I}$ is satisfied in all interpretations $\mathcal{I}$.

Two $\mathcal{EL}$ concept descriptions C and D are *equivalent* with respect to $\mathcal{T}$, and we shall write $\mathcal{T} \models C \equiv D$, if $\mathcal{T} \models \{C \sqsubseteq D,\ D \sqsubseteq C\}$. As a further abbreviation, let $\mathcal{T} \models C \sqsubsetneq D$ if both $\mathcal{T} \models C \sqsubseteq D$ and $\mathcal{T} \not\models C \sqsupseteq D$, and we then say that C is *strictly subsumed* by D with respect to $\mathcal{T}$. In the sequel of this document we may also write $C \leq_\mathcal{T} D$ instead of $\mathcal{T} \models C \leq D$ where $\leq$ is some suitable relation symbol, e.g., $\sqsubseteq$.

It is not hard to find $\mathcal{EL}$ concept descriptions that are equivalent, i.e., have the same extension in *all* interpretations, but are not equal. It is therefore helpful for technical details to have a unique *normal form* for $\mathcal{EL}$ concept descriptions. According to [4,9] an $\mathcal{EL}$ concept description C can be transformed into a *reduced form* that is equivalent to C by exhaustive application of the *reduction rule* $D \sqcap E \mapsto D$ whenever $\emptyset \models D \sqsubseteq E$ to the subconcepts of C (modulo commutativity and associativity of $\sqcap$). It is immediately clear that each $\mathcal{EL}$ concept description C is essentially a conjunction of other $\mathcal{EL}$ concept descriptions that are no conjunctions. In particular, if we define $\mathsf{Conj}(C)$ as the set of all top-level conjuncts in C, then C has the form $\bigsqcap \mathsf{Conj}(C)$ (modulo commutativity and associativity of $\sqcap$).

It is readily verified that the *subsumption* $\sqsubseteq_\emptyset$ constitutes a quasi-order on $\mathcal{EL}(\Sigma)$. Hence, the quotient of $\mathcal{EL}(\Sigma)$ with respect to the induced *equivalence* $\equiv_\emptyset$ is an ordered set. In what follows we will not distinguish between the equivalence classes and their representatives. Furthermore, $\top$ is the greatest element, and the quotient set $\mathcal{EL}(\Sigma)/{\equiv_\emptyset}$ is a lattice that we shall symbolize by $\boldsymbol{\mathcal{EL}}(\Sigma)$. It is easy to verify that the conjunction $\sqcap$ corresponds to the finitary *infimum* operation. In a description logic allowing for disjunctions $\sqcup$, it dually holds true that the disjunction $\sqcup$ corresponds to the finitary *supremum* operation. Unfortunately, this does not apply to our considered description logic $\mathcal{EL}$. As an obvious solution, we can simply define the notion of a *supremum* specifically tailored to the case of $\mathcal{EL}$ concept descriptions as follows. The *supremum* or *least common subsumer* (abbrv. *LCS*) of two $\mathcal{EL}$ concept descriptions C and D is an $\mathcal{EL}$ concept description E with the following properties.

1. $\emptyset \models \{C \sqsubseteq E,\ D \sqsubseteq E\}$
2. For each $\mathcal{EL}$ concept description F, if $\emptyset \models \{C \sqsubseteq F,\ D \sqsubseteq F\}$, then $\emptyset \models E \sqsubseteq F$.

Since all least common subsumers of C and D are unique up to equivalence, we may denote a representative of the corresponding equivalence class by $C \vee D$. It is well known that LCS-s always exist in $\mathcal{EL}$; in particular, the least common subsumer $C \vee D$ can be computed, modulo equivalence, by means of the following recursive formula.

$$\begin{aligned} C \vee D = {} & \bigsqcap(\Sigma_{\mathsf{C}} \cap \mathsf{Conj}(C) \cap \mathsf{Conj}(D)) \\ & \sqcap \bigsqcap\{\, \exists\, r.\, (E \vee F) \mid r \in \Sigma_{\mathsf{R}},\ \exists\, r.\, E \in \mathsf{Conj}(C), \text{ and } \exists\, r.\, F \in \mathsf{Conj}(D) \,\} \end{aligned}$$

It is easy to see that the equivalence $\equiv_\emptyset$ is compatible with both $\sqcap$ and $\vee$. Of course, the definition of a LCS can be extended to an arbitrary number of arguments in the obvious way, and we shall then denote the LCS of the concept descriptions C_t, $t \in T$, by $\bigvee\{\, C_t \mid t \in T\,\}$.

3 The Neighborhood Problem

In this section we consider the *neighborhood problem* for $\mathcal{EL}$. We have already seen that the set of $\mathcal{EL}$ concept descriptions constitutes a lattice. It is only natural to consider the question whether there exists a neighborhood relation which corresponds to the subsumption order. Remark that for an order relation $\leq$ on some set P its *neighborhood relation* or *transitive reduction* is defined as

$$\prec \mathrel{:=} \lneq \setminus (\lneq \circ \lneq) = \{\, (p, q) \mid p \lneq q \text{ and there exists no } x \text{ such that } p \lneq x \lneq q \,\}.$$

Clearly, if P is finite, then the transitive closure $\prec^+$ equals the irreflexive part $\lneq$. However, there are infinite ordered sets where this does not hold true; even worse, there are cases where $\prec^+$ is empty. Consider, for instance, the set $\mathbb{R}$ of real numbers with their usual ordering $\leq$. It is well-known that $\mathbb{R}$ is dense in itself, that is, for each pair $x \lneq y$, there is another real number z such that $x \lneq z \lneq y$—thus, there are no neighboring real numbers. In general, we say that $\leq$ is *neighborhood generated* if $\prec^+ = \lneq$ is satisfied. Clearly, $\leq$ is a neighborhood generated order relation if, and only if, there is a finite path $p = x_0 \prec x_1 \prec \ldots \prec x_n = q$ for each pair $p \leq q$. An alternative formulation is the following. $\leq$ is not neighborhood generated if, and only if, there exists some pair $p \lneq q$ such that every finite path $p = x_0 \lneq x_1 \lneq \ldots \lneq x_n = q$ can be refined, that is, there is some index i and an element y such that $x_i \lneq y \lneq x_{i+1}$. Of course, if the order relation $\leq$ is *bounded*, i.e., for each element $p \in P$, there exists a finite upper bound on the lengths of $\lneq$-paths issuing from p, then $\leq$ is neighborhood generated.

In the sequel of this section, we shall address the neighborhood problem from different perspectives. We first consider the general problem of existence of neighbors, and then provide means for the computation of all upper neighbors and of all lower neighbors, respectively, in the cases where these exist. As it will turn out, neighbors only exist for all concept descriptions in the description logic $\mathcal{EL}$ without any TBox. The presence of either a TBox or of the bottom concept description $\bot$ prevents the existence of neighbors for some concept descriptions. Furthermore, the extensions of $\mathcal{EL}$ with greatest fixed-point semantics also allow for the construction of concept descriptions that do not possess neighbors. Eventually, a complexity analysis shows that deciding neighborhood in $\mathcal{EL}$ is in P, and that all upper neighbors of an $\mathcal{EL}$ concept description can be computed in deterministic polynomial time.

Definition 1. *Consider a signature Σ, let $\mathcal{T}$ be a TBox over Σ, and further assume that C and D are concept descriptions over Σ. Then, C is a* lower neighbor *or a* most general strict subsumee *of D with respect to $\mathcal{T}$, denoted as $\mathcal{T} \models C \prec D$, if the following statements hold true.*

1. $\mathcal{T} \models C \sqsubsetneq D$
2. *For each concept description E over Σ, it holds true that $\mathcal{T} \models C \sqsubseteq E \sqsubseteq D$ implies $\mathcal{T} \models E \equiv C$ or $\mathcal{T} \models E \equiv D$.*

Additionally, we then also say that D is an upper neighbor *or a* most specific strict subsumer *of C with respect to $\mathcal{T}$, and we may also write $\mathcal{T} \models D \succ C$.*

We first observe that neighborhood of concept descriptions is not preserved by the concept constructors. It is easy to see that $\emptyset \models A \sqcap B \prec A$. However, it holds true that $\emptyset \models \exists\, r. (A \sqcap B) \sqsubsetneq \exists\, r. A \sqcap \exists\, r. B \sqsubsetneq \exists\, r. A$, which shows $\emptyset \not\models \exists\, r. (A \sqcap B) \prec \exists\, r. A$. Furthermore, we have that $\emptyset \models A \sqcap B \sqcap (A \sqcap B) \equiv A \sqcap (A \sqcap B)$, and consequently $\emptyset \not\models A \sqcap B \sqcap (A \sqcap B) \prec A \sqcap (A \sqcap B)$. There are according counterexamples when neighborhood with respect to a non-empty TBox is considered.

It is easily verified that neighborhood with respect to the empty TBox $\emptyset$ does not coincide with neighborhood w.r.t. a non-empty TBox $\mathcal{T}$. For instance, $\emptyset \models A \prec \top$ holds true, but $\{\top \sqsubseteq A\} \models A \equiv \top$. For the converse direction, consider the counterexample where $\{A \sqsubseteq B,\ B \sqsubseteq A\} \models A \sqcap B \prec \top$ and $\emptyset \models A \sqcap B \sqsubsetneq A \sqsubsetneq \top$.

3.1 The Empty TBox

Since Baader and Morawska showed in [4, Proof of Proposition 3.5] that $\sqsubseteq_\emptyset$ is bounded, we can immediately draw the following conclusion.

Proposition 2. *The subsumption relation $\sqsubseteq_\emptyset$ is neighborhood generated.*

After this first promising result, we continue with describing the neighborhood relation $\prec_\emptyset$. For this purpose, we define $\mathsf{Upper}(C) \coloneqq \{\, D \mid C \prec_\emptyset D \,\}$ as the set of all upper neighbors of a concept description C, and accordingly $\mathsf{Lower}(C)$ contains exactly all lower neighbors of C.

There is a well-known recursive characterization of $\sqsubseteq_\emptyset$ as follows: $C \sqsubseteq_\emptyset D$ if, and only if, $A \in \mathsf{Conj}(D)$ implies $A \in \mathsf{Conj}(C)$ for each concept name A, and for each $\exists\, r. F \in \mathsf{Conj}(D)$, there is some $\exists\, r. E \in \mathsf{Conj}(C)$ such that $E \sqsubseteq_\emptyset F$. With the help of that we can prove that there is the following necessary condition for neighboring concept descriptions.

Lemma 3. *Let C and D be some reduced $\mathcal{EL}$ concept descriptions over a signature Σ. If $\emptyset \models C \prec D$, then exactly one of the following statements holds true.*

1. *There is a concept name $A \in \Sigma_{\mathsf{C}}$ such that $\emptyset \models C \equiv D \sqcap A$.*
2. *It holds true that $\mathsf{Conj}(C) \cap \Sigma_{\mathsf{C}} = \mathsf{Conj}(D) \cap \Sigma_{\mathsf{C}}$, and there is exactly one existential restriction $\exists\, r.\, E \in \mathsf{Conj}(C)$ such that for all existential restrictions $\exists\, s.\, F \in \mathsf{Conj}(D)$, it holds true that $r = s$ and $\emptyset \models E \sqsubseteq F$ imply $\emptyset \not\models E \equiv F$.*

By means of the previous lemma we can deduce the following two propositions that describe the sets $\mathsf{Upper}(C)$ and $\mathsf{Lower}(C)$.

Proposition 4. *For each reduced $\mathcal{EL}$ concept description C over some signature Σ, the following recursive equation is satisfied (modulo equivalence).*

$$\begin{aligned}\mathsf{Upper}(C) = \{ \textstyle\bigsqcap \mathsf{Conj}(C) \setminus \{A\} \mid A \in \mathsf{Conj}(C) \,\} \\ \cup \{ \textstyle\bigsqcap \mathsf{Conj}(C) \setminus \{\exists\, r.\, E\} \cup \{\, \exists\, r.\, F \mid F \in \mathsf{Upper}(E) \,\} \mid \exists\, r.\, E \in \mathsf{Conj}(C) \,\}\end{aligned}$$

For instance, consider the concept description $A \sqcap \exists\, r.\, B \sqcap \exists\, s.\, (A \sqcap B)$. It is in reduced form and has three upper neighbors, namely $\exists\, r.\, B \sqcap \exists\, s.\, (A \sqcap B)$, $A \sqcap \exists\, r.\, \top \sqcap \exists\, s.\, (A \sqcap B)$, and $A \sqcap \exists\, r.\, B \sqcap \exists\, s.\, A \sqcap \exists\, s.\, B$.

Proposition 5. *For every $\mathcal{EL}$ concept description C over some signature Σ, the following equation is satisfied (modulo equivalence).*

$$\begin{aligned}\mathsf{Lower}(C) = \{\, C \sqcap A \mid A \in \Sigma_{\mathsf{C}} \text{ and } \emptyset \not\models C \sqsubseteq A \,\} \\ \cup \left\{ C \sqcap \exists\, r.\, D \;\middle|\; \begin{array}{l} r \in \Sigma_{\mathsf{R}},\ D \text{ is most general such that } \emptyset \not\models C \sqsubseteq \exists\, r.\, D, \\ \text{and } \emptyset \models C \sqsubseteq \exists\, r.\, E \text{ for all } E \text{ with } \emptyset \models D \prec E \end{array} \right\}\end{aligned}$$

While the recursive characterization of Upper in Proposition 4 immediately yields a procedure for enumerating all upper neighbors of a given concept description, the situation is not that apparent for lower neighbors. We can, however, constitute a procedure for computing lower neighbors by means of Proposition 5. Let C be an $\mathcal{EL}$ concept description over some signature Σ. Proceed as follows.

1. For each concept name $A \in \Sigma_{\mathsf{C}}$ with $\emptyset \not\models C \sqsubseteq A$, output $C \sqcap A$ as a lower neighbor of C.
2. For each role name $r \in \Sigma_{\mathsf{R}}$, recursively proceed as follows.
   (a) Let $D \coloneqq \top$.
   (b) While $\emptyset \models C \sqsubseteq \exists\, r.\, D$, replace D with a lower neighbor of D.
   (c) If $\emptyset \models C \sqsubseteq \exists\, r.\, E$ for all E with $\emptyset \models D \prec E$, then output $C \sqcap \exists\, r.\, D$ as a lower neighbor of C.

Eventually, we finish our investigations of $\prec_{\emptyset}$ with a complexity analysis. Using induction on the role depth of C, we can prove that, for each reduced $\mathcal{EL}$ concept description C, the set $\mathsf{Upper}(C)$ can be computed in deterministic time $\mathcal{O}(||C||^2)$. It is then apparent that the following proposition holds true.

Proposition 6. *The neighborhood relation $\prec_{\emptyset}$ can be decided in polynomial time, that is, $\prec_{\emptyset} \in \mathbf{P}$, and the mapping Upper is computable in deterministic polynomial time.*

3.2 The Bottom Concept Description

Now consider the extension of $\mathcal{EL}$ with the *bottom concept description* $\bot$ the semantics of which is defined as $\bot^{\mathcal{I}} := \emptyset$ for any interpretation $\mathcal{I}$. Then $\sqsubseteq_\emptyset$ is not bounded, since the following infinite chain exists.

$$\emptyset \models \bot \sqsubsetneq \ldots \sqsubsetneq (\exists\, r.)^{n+1}\, \top \sqsubsetneq (\exists\, r.)^{n}\, \top \sqsubsetneq \ldots \sqsubsetneq \exists\, r.\, \exists\, r.\, \top \sqsubsetneq \exists\, r.\, \top \sqsubsetneq \top$$

However, $\sqsupseteq_\emptyset$ is still well-founded, since whenever a chain starts with $\bot$, then the second element must be a satisfiable concept description, that is, some C with $C \not\equiv_\emptyset \bot$, after which the chain can only have a bounded number of elements. Furthermore, $\sqsubseteq_\emptyset$ is not neighborhood generated, as $\bot$ does not have any upper neighbors. To see this, consider a concept description C such that $\bot \sqsubsetneq_\emptyset C$; it then follows that $\bot \sqsubsetneq_\emptyset C \sqcap \exists\, r.\, C \sqsubsetneq_\emptyset C$.

3.3 A Non-Empty TBox

A similar situation arises when considering subsumption with respect to a non-empty TBox $\mathcal{T}$. Firstly, consider the simple signature Σ where $\Sigma_{\mathsf{C}} := \{A\}$ and $\Sigma_{\mathsf{R}} := \{r\}$ and define the TBox $\mathcal{T} := \{\top \sqsubseteq \exists\, r.\, \top,\ A \sqsubseteq \exists\, r.\, A\}$. We obtain that the quotient $\mathcal{EL}(\Sigma)/{\equiv_{\mathcal{T}}}$ consists of the classes $[(\exists\, r.)^n\, A]$ for $n \in \mathbb{N}$, and of the class $[\top]$. Furthermore, these concept descriptions are linearily ordered as follows.

$$\mathcal{T} \models A \sqsubsetneq \exists\, r.\, A \sqsubsetneq \exists\, r.\, \exists\, r.\, A \sqsubsetneq \exists\, r.\, \exists\, r.\, \exists\, r.\, A \sqsubsetneq \ldots \sqsubsetneq \top$$

Consequently, $\top$ does not have lower neighbors, and we also conclude that $\sqsubseteq_{\mathcal{T}}$ is not bounded and $\sqsupseteq_{\mathcal{T}}$ is not well-founded.

Secondly, we show that there is a TBox $\mathcal{T}$ and a concept description without any upper neighbors with respect to $\sqsubseteq_{\mathcal{T}}$. We try to keep things simple, and consider a rather small signature, namely Σ defined by $\Sigma_{\mathsf{C}} := \{A, B\}$ and $\Sigma_{\mathsf{R}} := \{r\}$, and we define a TBox by $\mathcal{T} := \{\exists\, r.\, A \sqsubseteq A,\ B \sqsubseteq A,\ B \equiv \exists\, r.\, B\}$. It can be shown that, for each $\mathcal{EL}(\Sigma)$ concept description C, either C is equivalent to B w.r.t. $\mathcal{T}$ or there exists an $n \in \mathbb{N}$ such that $\mathcal{T} \models B \sqsubsetneq (\exists\, r.)^n\, A \sqsubsetneq C$, i.e., B does not have upper neighbors with respect to $\mathcal{T}$.

Proposition 7. *There is some TBox $\mathcal{T}$ for which the subsumption relation $\sqsubseteq_{\mathcal{T}}$ is not neighborhood generated.*

3.4 Greatest Fixed-Point Semantics

Unfortunately, the situation is also not rosy for extensions of $\mathcal{EL}$ with *greatest fixed-point semantics* [1,11]. It then also holds true that $\sqsubseteq_\emptyset$ is neither bounded nor neighborhood generated, and $\sqsupseteq_\emptyset$ is not well-founded. One culprit is a concept description which represents a cycle, for instance $\nu\, X.\, \exists\, r.\, X$, the extension of which is maximal w.r.t. the property of containing elements that have some other element in that extension as an r-successor.

4 The Distributive, Graded Lattice of $\mathcal{EL}$ Concept Descriptions

The goal of this section is to explore the properties of the lattice of $\mathcal{EL}$ concept descriptions ordered by subsumption with respect to the empty TBox. In particular, Blyth [5, Chapters 4 and 5] shows that it suffices to investigate whether this lattice is distributive and of locally finite length, such that as an immediate corollary we then obtain that also the Jordan-Dedekind chain condition is satisfied, which states that for each pair $C \sqsubseteq_\emptyset D$, all maximal chains in the intervall $[C, D]$ have the same length. Furthermore, this length can then be utilized to define a distance between C and D, and in particular to measure a distance from each concept description C to the top concept description $\top$, which we call the rank of C.

Lemma 8. *For each signature Σ, the lattice $\boldsymbol{\mathcal{EL}}(\Sigma)$ is distributive, i.e., for all concept descriptions $C, D, E \in \mathcal{EL}(\Sigma)$, it holds true that*

$$\begin{aligned} &\emptyset \models C \sqcap (D \vee E) \equiv (C \sqcap D) \vee (C \sqcap E), \\ \textit{and} \quad &\emptyset \models C \vee (D \sqcap E) \equiv (C \vee D) \sqcap (C \vee E). \end{aligned}$$

Lemma 9. *For each signature Σ, the lattice $\boldsymbol{\mathcal{EL}}(\Sigma)$ is of locally finite length, that is, for all concept descriptions $C, D \in \mathcal{EL}(\Sigma)$ with $\emptyset \models C \sqsubseteq D$, every chain in the interval $[C, D]$ has a finite length.*

According to Blyth [5, Chapters 4 and 5], the following statements are obtained as immediate consequences of Lemmas 8 and 9.

Corollary 10. *1. For each signature Σ, the lattice $\boldsymbol{\mathcal{EL}}(\Sigma)$ is modular, i.e., for all concept descriptions $C, D, E \in \mathcal{EL}(\Sigma)$, it holds true that*

$$\begin{aligned} &\emptyset \models (C \sqcap D) \vee (C \sqcap E) \equiv C \sqcap (D \vee (C \sqcap E)), \\ &\emptyset \models (C \vee D) \sqcap (C \vee E) \equiv C \vee (D \sqcap (C \vee E)), \\ &\emptyset \models C \sqsubseteq D \quad \textit{implies} \quad \emptyset \models C \vee (E \sqcap D) \equiv (C \vee E) \sqcap D, \\ \textit{and} \quad &\emptyset \models C \sqsupseteq D \quad \textit{implies} \quad \emptyset \models C \sqcap (E \vee D) \equiv (C \sqcap E) \vee D. \end{aligned}$$

2. For each signature Σ, the lattice $\boldsymbol{\mathcal{EL}}(\Sigma)$ is both upper and lower semi-modular, i.e., for all concept descriptions $C, D \in \mathcal{EL}(\Sigma)$, it holds true that

$$\emptyset \models C \sqcap D \prec C \quad \textit{if, and only if,} \quad \emptyset \models D \prec C \vee D.$$

3. For each signature Σ, the lattice $\boldsymbol{\mathcal{EL}}(\Sigma)$ satisfies the Jordan-Dedekind chain condition, *i.e., for all concept descriptions $C, D \in \boldsymbol{\mathcal{EL}}(\Sigma)$ with $\emptyset \models C \sqsubsetneq D$, it holds true that all maximal chains in the interval $[C, D]$ have the same length.*

The notion of a rank function can be defined for ordered sets. The following definition specifically tailors this notion for the lattice $\boldsymbol{\mathcal{EL}}(\Sigma)$.

Definition 11. *An $\mathcal{EL}$* rank function *is a mapping $|\cdot| \colon \mathcal{EL}(\Sigma) \to \mathbb{N}$ with the following properties.*

1. $|\top| = 0$
2. $\emptyset \models C \equiv D$ *implies* $|C| = |D|$ *(equivalence closed)*
3. $\emptyset \models C \sqsubsetneq D$ *implies* $|C| \gneq |D|$ *(strictly order preserving)*
4. $\emptyset \models C \prec D$ *implies* $|C| + 1 = |D|$ *(neighborhood preserving)*

For an $\mathcal{EL}$ concept description C, we say that $|C|$ is the rank *of C.*

Proposition 12. *For each $C \in \mathcal{EL}(\Sigma)$, let $|C| \coloneqq 0$ if $\emptyset \models C \equiv \top$, and otherwise set*

$$|C| \coloneqq \mathsf{max}\{\, n+1 \mid \exists\, D_1,\ldots,D_n \in \mathcal{EL}(\Sigma)\colon \emptyset \models C \prec D_1 \prec \ldots \prec D_n \prec \top \,\}.$$

Then, $|\cdot|$ is an $\mathcal{EL}$ rank function.

Since $\boldsymbol{\mathcal{EL}}(\Sigma)$ satisfies the Jordan-Dedekind chain condition, we infer that in order to compute the rank $|C|$ of an $\mathcal{EL}$ concept description C over Σ with $\emptyset \not\models C \equiv \top$, we simply need to find *one* chain $\emptyset \models C \prec D_1 \prec D_2 \prec \ldots \prec D_n \prec \top$, and then it follows that $|C| = n+1$. Furthermore, $|C| = 0$ if $\emptyset \models C \equiv \top$.

Corollary 13. *For each signature Σ, the lattice $\boldsymbol{\mathcal{EL}}(\Sigma)$ is graded.*

The next lemma provides an equation for the rank of a conjunction of n concept descriptions. By induction over n, it follows from Lemma 9, Corollary 10, and [5, Theorem 4.6].

Lemma 14. *Let $\mathcal{C}$ be a set of n $\mathcal{EL}$ concept descriptions over Σ. Then, the following equation holds true.*

$$|\bigsqcap \mathcal{C}| = \sum_{i=1}^{n} (-1)^{i+1} \cdot \sum_{\mathcal{D} \in \binom{\mathcal{C}}{i}} |\bigvee \mathcal{D}|$$

Let $C = A_1 \sqcap \ldots \sqcap A_m \sqcap \exists\, r_1.\, C_1 \sqcap \ldots \sqcap \exists\, r_n.\, C_n$ be a reduced $\mathcal{EL}$ concept description. Then its rank can be computed as follows, cf. Lemma 14.

$$\begin{aligned} |C| &= |A_1 \sqcap \ldots \sqcap A_m \sqcap \exists\, r_1.\, C_1 \sqcap \ldots \sqcap \exists\, r_n.\, C_n| \\ &= |A_1 \sqcap \ldots \sqcap A_m| + |\exists\, r_1.\, C_1 \sqcap \ldots \sqcap \exists\, r_n.\, C_n| - |\top| \\ &= m + |\exists\, r_1.\, C_1 \sqcap \ldots \sqcap \exists\, r_n.\, C_n| \end{aligned}$$

Furthermore, it holds true that $\emptyset \models \exists\, r.\, C \vee \exists\, s.\, D \equiv \top$ if $r \neq s$. It follows that we can further simplify the rank computation as follows.

$$\begin{aligned} |\exists\, r_1.\, C_1 \sqcap \ldots \sqcap \exists\, r_n.\, C_n| &= |\bigsqcap\{\, \bigsqcap\{\, \exists\, r_i.\, C_i \mid i \in \{1,\ldots,n\} \text{ and } r_i = r \,\} \mid r \in \Sigma_{\mathsf{R}} \,\}| \\ &= \sum_{r \in \Sigma_{\mathsf{R}}} |\bigsqcap\{\, \exists\, r_i.\, C_i \mid i \in \{1,\ldots,n\} \text{ and } r_i = r \,\}| \end{aligned}$$

The rank of the conjunction of existential restrictions can be computed by means of Lemma 14, and finally it is readily verified that the rank of one existential restriction $\exists\, r.\, C$ satisfies the following equation.

$$|\exists\, r.\, C| = 1 + |\bigsqcap\{\, \exists\, r.\, D \mid \emptyset \models C \prec D \,\}|$$

Definition 15. *An $\mathcal{EL}$* metric *or $\mathcal{EL}$* distance function *is a mapping $\mathsf{d}\colon \mathcal{EL}(\Sigma) \times \mathcal{EL}(\Sigma) \to \mathbb{N}$ with the following properties.*

1. $\mathsf{d}(C, D) \geq 0$ *(non-negative)*
2. $\mathsf{d}(C, D) = 0$ *if, and only if,* $\emptyset \models C \equiv D$ *(equivalence closed)*

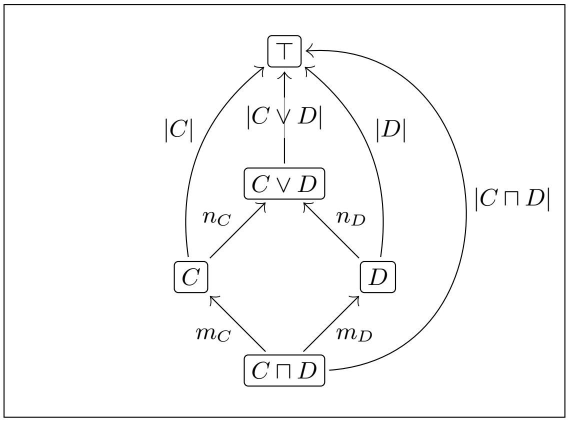


Fig. 1. Obtaining a distance function from the rank function

3. $\mathsf{d}(C, D) = \mathsf{d}(D, C)$ *(symmetric)*
4. $\mathsf{d}(C, E) \leq \mathsf{d}(C, D) + \mathsf{d}(D, E)$ *(triangle inequality)*

We then also say that $\mathsf{d}(C, D)$ *is the* distance *between* C *and* D.

Lemma 14 for the case $n = 2$ yields that in the rectangle shown in Figure 1 opposite edges have the same length, where length means length of a maximal chain between the endpoints. It is easy to see that $|C \sqcap D| = |C| + m_C = |D| + m_D$ and $|C \vee D| = |C| - n_C = |D| - n_D$. Thus, we infer that $m_C = |C \sqcap D| - |C| = |D| - |C \vee D| = n_D$, and similarily that $m_D = n_C$. Consequently, we can define an $\mathcal{EL}$ distance function in the following way.

Proposition 16. *For all* $C, D \in \mathcal{EL}(\Sigma)$*, define*

$$\mathsf{d}(C, D) \coloneqq |C \sqcap D| - |C \vee D|.$$

Then, d *is an* $\mathcal{EL}$ *metric.*

We can justify the name of a distance function as follows. If we consider the graph of $\mathcal{EL}$ concept descriptions such that edges exist exactly between neighboring concept descriptions, that is, if we consider the graph $(\mathcal{EL}(\Sigma), \prec_\emptyset \cup \succ_\emptyset)$, then the distance $\mathsf{d}(C, D)$ is the length of a shortest path between C and D.

Corollary 17. $\boldsymbol{\mathcal{EL}}(\Sigma)$ *is a metric lattice, i.e., a lattice which is also a metric space.*

It is easy to verify that $\boldsymbol{\mathcal{EL}}(\Sigma)$ is complete, not bounded, not precompact, not compact, locally compact, proper if Σ is finite, neither connected nor path connected, and separable. The induced topology τ_d is perfectly normal Hausdorff or T_6. Furthermore, all subsets are both open and closed, and all mappings $f \colon \boldsymbol{\mathcal{EL}}(\Sigma) \to (X, \mathsf{d}')$ are continuous.

Lemma 18. *Let* $C \in \mathcal{EL}(\Sigma)$*, then* $\mathsf{d}(C, \bigvee \mathsf{Upper}(C)) = |\mathsf{Upper}(C)|$.

According the the previous lemma, we can compute ranks as follows.

1. Let $D \coloneqq C$ and $r \coloneqq 0$.

2. While $\emptyset \not\models D \equiv \top$, compute the set $\mathsf{Upper}(D)$ of upper neighbors of D, set $r \coloneqq r + |\mathsf{Upper}(D)|$ and $D \coloneqq \bigvee \mathsf{Upper}(D)$.
3. Return r.

In [6] Ecke, Peñaloza, and Turhan defined the notion of a concept similarity measure as a function of type $\mathcal{EL}(\Sigma) \times \mathcal{EL}(\Sigma) \to [0,1]$, and then considered so-called *relaxed instances* of concept descriptions with respect to ontologies. Simply speaking, a is a relaxed instance of C if there is a concept that is similar enough to C and has a as an instance. It is straight-forward to consider these relaxed instances also with respect to the distance function we have just introduced. More formally, we define them as follows.

Definition 19. *Consider an interpretation $\mathcal{I}$ over some signature Σ and a concept description $C \in \mathcal{EL}(\Sigma)$, and let $n \in \mathbb{N}$. Then, the expression $\mathsf{Q} \leq n.\, C$ is called a* relaxed concept description*, and its extension is defined by*

$$(\mathsf{Q} \leq n.\, C)^{\mathcal{I}} \coloneqq \bigcup \{\, D^{\mathcal{I}} \mid D \in \mathcal{EL}(\Sigma) \text{ and } \mathsf{d}(C, D) \leq n \,\}.$$

Suppose that $\mathcal{O}$ is an ontology over some signature Σ, and further let $a \in \Sigma_{\mathsf{I}}$ be an individual name, $C \in \mathcal{EL}(\Sigma)$ a concept description, and $n \in \mathbb{N}$. We then say that a is a relaxed instance *of C with respect to $\mathcal{O}$ and distance threshold n, denoted as $\mathcal{O} \models a \mathrel{\varepsilon} \mathsf{Q} \leq n.\, C$, if it holds true that $a^{\mathcal{I}} \in (\mathsf{Q} \leq n.\, C)^{\mathcal{I}}$ for each model $\mathcal{I}$ of $\mathcal{O}$.*

For transforming our distance function d into a similarity function $\mathsf{s}\colon \mathcal{EL}(\Sigma) \times \mathcal{EL}(\Sigma) \to [0,1]$ we can proceed as follows. We begin with transforming d into a metric with range $[0,1)$. For that purpose, we choose an order-preserving, sub-additive function $f\colon [0,\infty) \to [0,1)$ with $\ker(f) = \{0\}$. Note that a function $f\colon [0,\infty) \to \mathbb{R}$ is sub-additive if $f'' < 0$ and $f(0) = 0$. Then $f \circ \mathsf{d}$ is such a metric with range $[0,1)$. Suitable functions are the following.

- $f\colon x \mapsto \frac{x}{1+x}$ or more generally $f\colon x \mapsto (\frac{x}{1+x})^y$ for $y > 0$
- $f\colon x \mapsto 1 - \frac{1}{2^x}$ or more generally $f\colon x \mapsto 1 - y^x$ for $y \in (0,1)$

Then, $\mathsf{s} \coloneqq 1 - f \circ \mathsf{d}$ is a similarity function on $\mathcal{EL}(\Sigma)$. It is easy to verify that then s satifies the following properties which have been defined by Lehmann and Turhan in [10], for all $\mathcal{EL}$ concept descriptions C, D, E over Σ.

1. $\mathsf{s}(C,D) = \mathsf{s}(D,C)$ (symmetric)
2. $1 + \mathsf{s}(C,D) \geq \mathsf{s}(C,E) + \mathsf{s}(E,D)$ (triangle inequality)
3. $\emptyset \models C \equiv D$ implies $\mathsf{s}(C,E) = \mathsf{s}(D,E)$ (equivalence invariant)
4. $\emptyset \models C \equiv D$ if, and only if, $\mathsf{s}(C,D) = 1$ (equivalence closed)
5. $\emptyset \models C \sqsubseteq D \sqsubseteq E$ implies $\mathsf{s}(C,D) \geq \mathsf{s}(C,E)$ (subsumption preserving)
6. $\emptyset \models C \sqsubseteq D \sqsubseteq E$ implies $\mathsf{s}(C,E) \leq \mathsf{s}(D,E)$ (reverse subsumption preserving)

However, as it turns out such a similarity measure $1 - f \circ \mathsf{d}$ does not satisfy the property of *structural dependance*. For instance, consider a signature Σ without role names and such that $\Sigma_C \coloneqq \{A\} \cup \{\, B_n \mid n \in \mathbb{N} \,\}$. It is now readily verified that

$$(1 - f \circ \mathsf{d})(A \sqcap \bigsqcap \{\, B_\ell \mid \ell \leq n \,\}, \bigsqcap \{\, B_\ell \mid \ell \leq n \,\}) = 1 - f(1)$$

for all $n \in \mathbb{N}$, and since $f(1) > 0$ we conclude that the sequence does not converge to 1 for $n \to \infty$.

For extending our rank function $|\cdot|$ and our distance function d to $\mathcal{EL}^{\bot}$, we can simply define $|\bot| \coloneqq \infty$, $\mathsf{d}(\bot, \bot) \coloneqq 0$, and $\mathsf{d}(\bot, C) \coloneqq \mathsf{d}(C, \bot) \coloneqq \infty$ for $\emptyset \models C \not\equiv \bot$. When transforming the extended metric into a similarity measure then two concept descriptions have a similarity of 0 if, and only if, exactly one of them is unsatisfiable. In $\mathcal{EL}$ without the bottom concept description $\bot$, a similarity of 0 can never occur when utilizing the above construction.

We close this section with some first investigations on the complexities of decision problems and computation problems related to the introduced rank function. So far, it is unknown whether the rank function can be tractably computed, i.e., in deterministic polynomial time. However, if $|C|$ is computed in the naïve way by constructing an arbitrary chain of neighbors from C to $\top$ and then determinining its length, at least deterministic exponential time with respect to the size of C is necessary. To see this, consider the concept description $C_n \coloneqq \exists\, r. \bigsqcap\{A_1, \ldots, A_n\}$ for each $n \in \mathbb{N}$. It is well-known that there are exponentially many subsets of $\{A_1, \ldots, A_n\}$ with $\lfloor \frac{n}{2} \rfloor$ elements; let $X_1, \ldots, X_\ell$ be an enumeration of these, and define $D_m \coloneqq \bigsqcap\{\, \exists\, r. \bigsqcap X_i \mid i \in \{m, \ldots, \ell\}\,\}$. Clearly, then $C_n \sqsubsetneq_\emptyset D_1 \sqsubsetneq_\emptyset D_2 \sqsubsetneq_\emptyset \cdots \sqsubsetneq_\emptyset D_\ell \sqsubsetneq_\emptyset \top$ is an exponentially long chain of strict subsumptions. We conclude that $|C_n|$ is at least exponential in n.

Given a concept description C and a natural number n (in binary encoding), then we can decide in triple exponential time whether the rank of C is equal to n, at most n, or at least n. A procedure can construct a chain of n neighbors and then check whether $\top$ is reached. If n is fixed, then this requires only deterministic polynomial time.

5 Conclusion

We have investigated the *neighborhood problem* for the description logic $\mathcal{EL}$ and some of its variants. We found that existence of neighbors can in general only be guaranteed for the case of $\mathcal{EL}$ without a TBox, without the bottom concept description, and without greatest fixed-point semantics. The presence of a TBox, the bottom concept description, or greatest fixpoint semantics allow for the construction of concept descriptions that do not have neighbors in certain directions. For the case of $\mathcal{EL}$ we proposed sound and complete procedures for deciding neighborhood as well as for computing all upper neighbors and all lower neighbors, respectively. Furthermore, we have shown that deciding neighborhood and computing all upper neighbors requires only deterministic polynomial time. The complexity of computing all lower neighbors is currently an open question; possibly there exists a less expensive procedure than the one presented here.

As further results, we have proven that the lattice of $\mathcal{EL}$ concept descriptions is distributive, modular, graded, and metric. In particular, this means that there exists a rank function as well as a distance function on this lattice. Some first complexity results on problems related to these rank and distance functions were found. However, the exact complexities are currently unknown; we do not know whether the presented upper bounds are sharp, and lower bounds are also not available. This implies that there could possibly exist faster procedures for computing ranks and distances than those introduced in this document. In particular, better formulas for computing or approximating ranks of $\mathcal{EL}$ concept description should be sought. Some initial experiments could lead to the claim that the rank of an $\mathcal{EL}$ concept description with role depth n, that is, for which the nesting depth of existential quantifiers is n, could be n-exponential in the size of C.

As an important consequence we infer that the algorithm *NextClosures* [8] can be utilized for enumerating canonical bases of closure operators in $\mathcal{EL}$.

Other possible future research could consider extensions to more expressive description logics. Of course, these logics should be considered without TBoxes for deciding existence of neighbors in general. Eventually, a further direction for future research is a more fine-grained characterization of existence of neighbors. That is, given a description logic where neighbors need not exist in general, how can we decide whether a concept description has neighbors and how can we enumerate these?

Acknowledgements The author gratefully thanks both Franz Baader and Bernhard Ganter for fruitful discussions on neighborhood generated orders, and furthermore thanks the anonymous reviewers for their constructive hints and helpful remarks.

References

1. Baader, F.: Terminological Cycles in a Description Logic with Existential Restrictions. In: Gottlob, G., Walsh, T. (eds.) IJCAI-03, Proceedings of the Eighteenth International Joint Conference on Artificial Intelligence, Acapulco, Mexico, August 9-15, 2003. pp. 325–330. Morgan Kaufmann (2003)
2. Baader, F., Brandt, S., Lutz, C.: Pushing the $\mathcal{EL}$ Envelope. In: Kaelbling, L.P., Saffiotti, A. (eds.) IJCAI-05, Proceedings of the Nineteenth International Joint Conference on Artificial Intelligence, Edinburgh, Scotland, UK, July 30 - August 5, 2005. pp. 364–369. Professional Book Center (2005)
3. Baader, F., Horrocks, I., Lutz, C., Sattler, U.: An Introduction to Description Logic. Cambridge University Press (2017)
4. Baader, F., Morawska, B.: Unification in the Description Logic $\mathcal{EL}$. Logical Methods in Computer Science **6**(3) (2010)
5. Blyth, T.S.: Lattices and Ordered Algebraic Structures. Universitext, Springer London, 1st edn. (2005)
6. Ecke, A., Peñaloza, R., Turhan, A.: Similarity-based Relaxed Instance Queries. Journal of Applied Logic **13**(4), 480–508 (2015)
7. Hitzler, P., Krötzsch, M., Rudolph, S.: Foundations of Semantic Web Technologies. Chapman and Hall/CRC Press (2010)
8. Kriegel, F.: NextClosures with Constraints. In: Huchard, M., Kuznetsov, S. (eds.) Proceedings of the Thirteenth International Conference on Concept Lattices and Their Applications, Moscow, Russia, July 18-22, 2016. CEUR Workshop Proceedings, vol. 1624, pp. 231–243. CEUR-WS.org (2016)
9. Küsters, R.: Non-Standard Inferences in Description Logics, Lecture Notes in Computer Science, vol. 2100. Springer (2001)
10. Lehmann, K., Turhan, A.: A Framework for Semantic-Based Similarity Measures for $\mathcal{ELH}$-Concepts. In: del Cerro, L.F., Herzig, A., Mengin, J. (eds.) Logics in Artificial Intelligence - 13th European Conference, JELIA 2012, Toulouse, France, September 26-28, 2012. Proceedings. Lecture Notes in Computer Science, vol. 7519, pp. 307–319. Springer (2012)
11. Lutz, C., Piro, R., Wolter, F.: Enriching $\mathcal{EL}$-Concepts with Greatest Fixpoints. In: Coelho, H., Studer, R., Wooldridge, M. (eds.) ECAI 2010 - 19th European Conference on Artificial Intelligence, Lisbon, Portugal, August 16-20, 2010, Proceedings. Frontiers in Artificial Intelligence and Applications, vol. 215, pp. 41–46. IOS Press (2010)

Characterizing Covers of Functional Dependencies using FCA

Victor Codocedo[1], Jaume Baixeries[2], Mehdi Kaytoue[3], and Amedeo Napoli[4]

[1] Universidad Técnica Federico Santa María. Campus San Joaquín, Santiago de Chile.
[2] Universitat Politècnica de Catalunya. 08032, Barcelona. Catalonia.
[3] Infologic, 99 Avenue de Lyon, F-26500, Bourg-lès-Valence, France
[4] LORIA (CNRS - Inria Nancy Grand Est - Université de Lorraine), B.P. 239, F-54506, Vandœuvre-lès-Nancy.

Corresponding author: `victor.codocedo@inf.utfsm.cl`

Abstract. Functional dependencies (FDs) can be used for various important operations on data, for instance, checking the consistency and the quality of a database (including databases that contain complex data). Consequently, a generic framework that allows mining a sound, complete, non-redundant and yet compact set of FDs is an important tool for many different applications. There are different definitions of such sets of FDs (usually called *covers*).

In this paper, we present the characterization of two different kinds of covers for FDs in terms of pattern structures. The convenience of such a characterization is that it allows for an easy implementation of efficient mining algorithms which can later be easily adapted to other kinds of similar dependencies. Finally, we present empirical evidence that the proposed approach can perform better than a state-of-the-art FD miner in large databases.

1 Introduction

Functional Dependencies (FDs) are a keystone of the relational database model, since they allow checking the consistency, maintaining the quality of a database [8, 10, 9], and guiding its design [20]. In addition, they have been used to study information integration in the Web of data with varying degrees of quality [24, 25], or to check the data completeness in DBpedia [1]. Therefore, the computation of a succinct representation of a set of FDs (usually refered to as a **cover**), is of interest to various fields of knowledge discovery and representation, specially, if this computation can easily be extended to other kinds of dependencies (e.g. relaxed versions of FDs [5]).

The computation of FD *covers* is a popular topic in the database literature. As a reference, in [21], seven algorithms to mine FDs and compute their covers are reviewed and grouped into three families: *lattice transversal algorithms*, *difference/agree sets algorithms*, and *dependency induction* algorithms. The characterization of FDs with FCA and pattern structures is also presented in [2] and

© Dmitry I. Ignatov, Lhouari Nourine (Eds.): CLA 2018, pp. 279–290,
ISBN 978–80–244–5328–6, Department of Computer Science, Palacký University
Olomouc, 2018.

a generalization to other types of FDs is given in [6]. For a detailed review on the characterization of FDs and FCA, see [3].

In this paper, we characterize the representations of FD covers using pattern structures, an extension of FCA dealing with complex object representations [18].

On the one hand, we adapt the definition of a canonical direct basis of implications with proper premises [4, 22] to the formalism of pattern structures, in order to prove that this basis is equivalent to a reduced non-redundant set of FDs, better known as the canonical cover (Section 3.1). We show that the canonical cover can be characterized using the arrow relation ($\swarrow$) between an attribute and a pattern defined over a partition pattern structures (PPS). On the other hand, we discuss on the relation between the Stem Base of implications [12] with the Minimal Cover of dependencies, a sound, complete, non-redundant set of FDs that has minimum cardinality w.r.t. any other equivalent cover (Section 3.2). We show that the latter can be characterized by pseudo-extents of a PPS.

Finally, in Section 4 we empirically compare these two ways of computing FD covers with the algorithm TANE [16]. This algorithm is one of the most efficient FD mining algorithms and according to [21], it is the base for the family of "lattice transversal algorithms" serving as the baseline to validate our approach with a state-of-the-art FD miner.

2 Theoretical Background

Let $\mathcal{U}$ be an ordered set of attributes, and let 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 $t : \mathcal{U} \rightarrow Dom$, and a table T is a set of tuples. Usually a table is represented as a matrix, as in Table 1, where the set of tuples (or objects) is $T = \{ t_1, t_2, \ldots, t_7 \}$ with attributes $\mathcal{U} = \{ a, b, c, d, e \}$. We use *table, dataset, set of tuples* as equivalent terms. We overload the functional notation of a tuple in such a way that, given a tuple $t \in T$, we say that $t(X \subseteq \mathcal{U})$ is a tuple with the values of t in the ordered attributes $x_i \in X$ defined as $t(X) = \langle t(x_1), t(x_2), \ldots, t(x_n) \rangle$. For example, we have that $t_2(\{ a, c \}) = \langle t_2(a), t_2(c) \rangle = \langle 2, 1 \rangle$. In this article, the set notation is dropped: instead of $\{ a, b \}$ we use ab.

2.1 Functional Dependencies and their Covers

Definition 1 ([23]). *Let T be a set of tuples, and $X, Y \subseteq \mathcal{U}$. A* **functional dependency (FD)** *$X \rightarrow Y$ holds in T if:*

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

For instance, the functional dependency $d \rightarrow e$ holds in T (Table 1), whereas the functional dependency $e \rightarrow d$ does not hold since $t_4(e) = t_5(e)$ but $t_4(d) \neq t_5(d)$.

For a given set of functional dependencies F, we can use the three Armstrong's axioms (reflexivity, augmentation and transitivity) to derive a larger set of FDs [20]. We will call F^* the set of FDs derived from F by reflexivity and augmentation, and F^+ the set of FDs derived by reflexivity, augmentation and transitivity. Two sets of FDs F and H are said to be equivalent $F \equiv H \iff F^+ = H^+$.

Let F be a set of FDs from a database T, F is said to be *sound* if all FDs in F hold in T. In addition, F is said to be *complete* if all FDs that hold in T can be derived from F. Let $X \to Y$ be any FD in F, then F is said to be *non-redundant* if $F \backslash \{X \to Y\} \not\equiv F$, and *non-redundant w.r.t. augmentation* iff $(F \backslash \{X \to Y\})^* \neq F^*$

A set F is said to be *left-reduced* if for all $X \to Y \in F$ and $Z \subsetneq X$ we have that $(F \backslash \{X \to Y\}) \cup \{Z \to Y\} \not\equiv F$. Dually, it is said to be *right-reduced* if for all $X \to Y \in F$ and $Z \subsetneq Y$ we have that $(F \backslash \{X \to Y\}) \cup \{X \to Z\} \not\equiv F$. F is said to be *reduced* if it is simultaneously *left* and *right-reduced.*

Let F be a reduced set of FDs, then $G = \{X \to y \mid X \to Y \in F, y \in Y\}$ is the *splitting* of F and $G \equiv F$. Let F be a *reduced* set, its splitting is called a *canonical cover.* A *canonical cover* is a *left-reduced*, *non-redundant w.r.t. augmentation* set of FDs with a single element in their right hand side (a split set) [19]. A different definition presents the canonical cover in a *saturated* version requiring uniqueness in their left hand side [17] losing the single element condition in the right hand side. For the sake of compatibility with FCA implication covers we will favor the first definition. Notice that *canonical covers* can be redundant w.r.t. transitivity. For example the *canonical cover* of Table 1 contains $\{c \to b, c \to e, d \to e, bd \to c, be \to c\}$ where $bd \to c$ would be redundant w.r.t. transitivity as $bd \to bde \to c$.

Finally, a set F is said to be a *minimum cover* if it has as few FDs as any other equivalent set of FDs. For example, the *minimum cover* of Table 1 contains FDs $\{c \to be, d \to e, be \to c\}$. Notice that the minimum cover is not restricted to be reduced, so it is not presented with split sets. Secondly, the minimum cover contains exactly one fewer FD than the canonical cover, namely $bd \to c$.

2.2 Formal Concept Analysis, Implication Systems and FDs

For the sake of brevity we do not provide a description of the Formal Concept Analysis (FCA) framework. The notation used in this article follows [13] where $\mathcal{K} = (G, M, I)$ is a formal context of objects G, attributes M and incidence relation I, with formal concepts (A, B) where $A' = B$ and $B' = A$.

Implications are relations established between attribute sets from a formal context $\mathcal{K}$. Implications are analogous to FDs and they can be used to characterize them [2]. Because of this, we will notate an implication similarly to an FD. Implication systems (sets of implications) can also characterize FD covers [4].

An implication $X \to Y$ holds in $\mathcal{K}$ for $X, Y \subseteq M$ if $Y \subseteq X''$. Let T be a set of tuples and $\mathcal{U}$, a set of attributes in a table (such as the one in Table 1). We define the set $Pair(T) = \binom{T}{2}$ set of all subsets T with exactly two elements, and the incidence set I such that $((t_i, t_j), x) \in I \iff t_i[x] = t_j[x]$, $\forall x \in \mathcal{U}$, $\forall t_i, t_j \in T$. It can be shown that an FD $X \to Y$ holds in the database if and only if $X \to Y$ is an implication of the formal context $\mathcal{K} = (Pair(T), \mathcal{U}, I)$ [13].

$\mathcal{K}$ is called the *binary codification* of table T. For example, Table 3 contains the binary codification of Table 1. In Table 3 we can observe the implication $d \rightarrow e$ which can be verified as an FD in Table 1.

The previous statement entails that FDs and implications in $\mathcal{K}$ are in 1-1 correspondence. Moreover, the corresponding definition of a canonical cover of FDs is equivalent to that of a *canonical-direct unitary basis* of implications as shown in [4] where the equivalent *left-minimal basis* is described as containing *minimal functional dependencies*, i.e. those in a canonical cover.

Table 1: Example of a table T

	a	b	c	d	e
t_1	1	1	1	1	1
t_2	2	1	1	1	1
t_3	3	1	2	2	2
t_4	3	2	3	2	2
t_5	3	1	2	3	2
t_6	1	1	2	2	2
t_7	1	1	2	4	2

Table 2: Representation context

	a	b	c	d	e
$\delta(a)$	×				
$\delta(b)$		×			
$\delta(e)$					×
$\delta(a) \sqcap \delta(b)$	×	×	↙		↙
$\delta(a) \sqcap \delta(e)$	×				×
$\delta(d) \sqcap \delta(e)$				×	×
$\delta(a) \sqcap \delta(d) \sqcap \delta(e)$	×	↙	↙	×	×
$\delta(b) \sqcap \delta(c) \sqcap \delta(e)$		×	×		×
$\delta(a) \sqcap \delta(b) \sqcap \delta(c) \sqcap \delta(e)$	×	×	×	↙	×
$\delta(b) \sqcap \delta(c) \sqcap \delta(d) \sqcap \delta(e)$	↙	×	×	×	×
$\delta(a) \sqcap \delta(b) \sqcap \delta(c) \sqcap \delta(d) \sqcap \delta(e)$	×	×	×	×	×

Table 3: Binary codification of Table 1

	a	b	c	d	e
(t_1, t_2)		×	×	×	×
(t_1, t_3)		×			
(t_1, t_4)					
(t_1, t_5)		×			
(t_1, t_6)	×	×			
(t_1, t_7)	×	×			
(t_2, t_3)		×			
(t_2, t_4)					
(t_2, t_5)		×			
(t_2, t_6)		×			
(t_2, t_7)		×			
(t_3, t_4)	×			×	×
(t_3, t_5)	×	×	×		×
(t_3, t_6)		×	×	×	×
(t_3, t_7)		×	×		×
(t_4, t_5)	×				×
(t_4, t_6)				×	×
(t_4, t_7)					×
(t_5, t_6)		×	×		×
(t_5, t_7)		×	×		×
(t_6, t_7)	×	×	×		×

2.3 Partition Pattern Structures

A Partition Pattern Structure (PPS) is a type of pattern structure [14] that deals with, as the name suggests, object representations in the form of set partitions. PPS have shown to be useful for mining biclusters [7] and, more importantly, relations between partition pattern concepts have been used to characterize FDs of different kinds [3].

The formalization of a database T with attributes $\mathcal{U}$ as a PPS is as follows. A partition d of T is a set $\mathrm{d} \subseteq \wp(T)$ (powerset of T) of disjoint non-empty subsets of T such that for any two different elements $K_i, K_j \in \mathrm{d}$ we have that $K_i \cap K_j = \varnothing$ and $\bigcup_{K \in \mathrm{d}} K = T$. Let D be the set of all possible partitions of T, then any two partitions $\mathrm{d}_1, \mathrm{d}_2 \in \mathrm{D}$ can be ordered by a *coarser/finer* relation denoted $\mathrm{d}_1 \sqsubseteq \mathrm{d}_2$ (d_1 is finer than d_2) iff $(\forall\ K_i \ \in\ \mathrm{d}_1)(\exists\ K_j \in\ \mathrm{d}_2)$ such that $K_i \subseteq K_j$. The similarity operator is defined as $\mathrm{d}_1 \sqcap \mathrm{d}_2 = \{K_i \cap K_j \mid K_i \in \mathrm{d}_1, K_j \in \mathrm{d}_2\}$. From this, it follows that $(\mathrm{D}, \sqsubseteq)$ is a complete lattice with supremum and infimum defined respectively as $\top = \{\{T\}\}$ and $\bot = \{\{t\} \mid t \in T\}$.

The set of attributes $\mathcal{U}$ is mapped onto D through a function δ which for a given attribute $x \in \mathcal{U}$ yields a partition $\delta(x) \in \mathrm{D}$ representing the equivalence relations over T for values of attribute x. With this, we can configure the PPS $(\mathcal{U}, (\mathrm{D}, \sqsubseteq), \delta)$ with derivation operators $X^{\square} = \bigsqcap_{x \in X} \delta(x)$ denoting the equivalence relations implied by attributes in X, and $\mathrm{d}^{\square} = \{x \in \mathcal{U} \mid \delta(x) \sqsubseteq \mathrm{d}\}$ denoting all attributes with associated equivalence relations finer than d. (X, d) is a partition pattern concept when $X^{\square} = \mathrm{d}$ and $\mathrm{d}^{\square} = X$.

Theorem 1 (Proposition 2 in [3]). *Let $(Pair(T), \mathcal{U}, I)$ be the binary codification of a table within a database, and $(\mathcal{U}, (\mathrm{D}, \sqsubseteq), \delta)$ its PPS representation:*

$$(W, X) \in (Pair(T), \mathcal{U}, I) \iff (X, \mathrm{d}) \in (\mathcal{U}, (\mathrm{D}, \sqsubseteq), \delta)$$

The proof of Theorem 1 can be found in [3]. Theorem 1 presents an important property of the PPS that states that $X \subseteq \mathcal{U}$ is an extent in $(\mathcal{U}, (\mathrm{D}, \sqsubseteq), \delta)$ if and only if it is also an intent in $(Pair(T), \mathcal{U}, I)$ (the relation between the set of tuple pairs $W \subseteq Pair(T)$ and the partition $\mathrm{d} \in \mathrm{D}$ can be formalized as well but it is of no interest to our development). Theorem 1 is very important since it entails that the lattices derived from $(Pair(T), \mathcal{U}, I)$ and $(\mathcal{U}, (\mathrm{D}, \sqsubseteq), \delta)$ are isomorphic. Consequently, implication $X \to Y$ in $(Pair(T), \mathcal{U}, I)$ can be found as a relation between extents in $(\mathcal{U}, (\mathrm{D}, \sqsubseteq), \delta)$ (extent implication) such that $Y \subseteq X^{\square\square}$.

3 Covers and Pattern Structures

In this section we present two different kinds of covers for FDs: canonical covers (Section 3.1) and minimal covers (Section 3.2), as well as their characterization in terms of Pattern Structures. This section uses existing well-known results in FCA, which are reviewed here for the sake of readability.

Section 3.1 presents how a canonical cover for FDs can be computed using PPS, according to the results in [4, 22]. Section 3.2 introduces a novel characterization of the minimum cover of FDs by means of pseudo-extents of PPS [13]. The interest of these results is not limited to computing FD covers, but also for generalizations of FDs [3, 6].

3.1 Characterizing a Canonical Cover of FDs

The characterization of a canonical cover of FDs using FCA is straightforward. In a nutshell, a canonical cover of FDs is analogous to a *canonical direct unitary basis* of implications [4] as presented in Section 2.2. In this section we recall some of these ideas and show how they can be simply adapted to the framework of PPS.

Firstly, let $(\mathcal{U}, (\mathrm{D}, \sqsubseteq), \delta)$ be a PPS, we define $\underline{\mathrm{D}} = \{\mathrm{d} \in \mathrm{D} \mid \mathrm{d}^{\square\square} = \mathrm{d}\}$ as the set of all closed partition patterns in D. The formal context $(\mathcal{U}, \underline{\mathrm{D}}, J)$ with $(\mathrm{d}, x) \in J \iff \mathrm{d} \sqsubseteq \delta(x)$ is called a representation context of $(\mathcal{U}, (\mathrm{D}, \sqsubseteq), \delta)$ and their corresponding concept lattices are isomorphic [18]. For the sake of readability of the following definitions, we define the representation context as $(\underline{\mathrm{D}}, \mathcal{U}, J)$ instead of $(\mathcal{U}, \underline{\mathrm{D}}, J)$ (Table 2). By transitivity of equivalence, it is clear that $(\underline{\mathrm{D}}, \mathcal{U}, J)$ is isomorphic to $(Pair(T), \mathcal{U}, I)$ as defined in Section 2.3 and as such, implications in $(\underline{\mathrm{D}}, \mathcal{U}, J)$ correspond to FDs. For example, Table 2 (not considering elements $\swarrow$) contains the representation context $(\underline{\mathrm{D}}, \mathcal{U}, J)$ of the PPS derived from Table 1. Notice that objects are closed intersections of object representations, e.g. $\delta(d)$ does not appear since $\delta(d)^{\square\square} = \delta(d) \sqcap \delta(e)$. With this, the canonical direct basis of implications in $(\underline{\mathrm{D}}, \mathcal{U}, J)$ (and thus, canonical cover of FDs) is determined by the set of proper premises of elements in $\mathcal{U}$.

Theorem 2 (Corollary 1 in [22]). *$X \subseteq \mathcal{U} \backslash \{y\}$ is a premise of $y \in \mathcal{U}$ iff X is a hypergraph transversal of $\mathcal{H}_y^{\swarrow}$ defined as :*

$$\mathcal{H}_y^{\swarrow} = \{((\mathcal{U} \backslash \{y\})) \backslash \mathrm{d}' \mid \mathrm{d} \in \underline{\mathrm{D}}, \mathrm{d} \swarrow y\}$$

The set of all proper premises of y is the minimum hypergraph transversal $Tr(\mathcal{H}_y^{\swarrow})$.

A detailed description on the development of Theorem 2 can be found in [22]. Providing a formal definition of hypergraph transversals is out of the scope of this article, however we briefly mention that this formalization can also be made considering set collections (instead of hypergraphs) and minimum hitting sets (instead of minimum hypergraph transversals) [11]. This problem is also analogous to the vertex cover problem [12].

Theorem 2 provides a formal description for the proper premises of a given attribute $y \in \mathcal{U}$ that in turn yields the canonical cover of functional dependencies. However this approach requires the creation of the representation context which is a middle step in the overall calculation process. Actually, by analyzing the arrow relation between d and y we can observe that the representation context is not necessary. Consider that in $(\underline{\mathrm{D}}, \mathcal{U}, J)$, $\mathrm{d}' = \{x \in \mathcal{U} \mid (\mathrm{d}, x) \in J (\iff \mathrm{d} \sqsubseteq \delta(x))\}$ and thus d' is equivalent to $\mathrm{d}^{\square}$ for any $\mathrm{d} \in \underline{\mathrm{D}}$. Moreover, in $(\mathcal{U}, (\mathrm{D}, \sqsubseteq), \delta)$, $\mathrm{d}^{\square} \subsetneq \mathrm{h}^{\square} \iff \mathrm{h} \sqsubsetneq \mathrm{d}$ since $\mathrm{h} = \mathrm{h}^{\square\square}$ and $\mathrm{d} = \mathrm{d}^{\square\square}$. With this, we can rewrite the arrow definition as follows.

$$\begin{aligned} \mathrm{d} \swarrow y &\iff (\mathrm{d}, y) \notin J \text{ and if } \mathrm{d}' \subsetneq \mathrm{h}' \text{ then } (\mathrm{h}, y) \in J \\ &\iff \mathrm{d} \not\sqsubseteq \delta(y) \text{ and if } \mathrm{d}^{\square} \subsetneq \mathrm{h}^{\square} \text{ then } \mathrm{h} \sqsubseteq \delta(y) \\ &\iff \mathrm{d} \not\sqsubseteq \delta(y) \text{ and if } \mathrm{h} \sqsubsetneq \mathrm{d} \text{ then } \mathrm{h} \sqsubseteq \delta(y) \end{aligned}$$

The last result shows that $\mathrm{d} \swarrow y$ in $(\underline{\mathrm{D}}, \mathcal{U}, J)$ can be defined directly over the PPS. Intuitively, this definition corresponds to $y \nearrow \mathrm{d}$ in $(\mathcal{U}, \underline{\mathrm{D}}, J)$ and thus, in $(\mathcal{U}, (\mathrm{D}, \sqsubseteq), \delta)$. With these elements we can finally propose a characterization for the canonical cover of functional dependencies in $(\mathcal{U}, (\mathrm{D}, \sqsubseteq), \delta)$ as follows.

Corollary 1. *Let $(\mathcal{U}, (\mathrm{D}, \sqsubseteq), \delta)$ be a partition pattern structure and $Tr(\mathcal{H})$ denote the hypergraph transversal of $\mathcal{H}$, then with*

$$\begin{aligned} \mathcal{L}_{cc} &= \{X \to y \mid y \in \mathcal{U}, X \in Tr(\mathcal{H}_y^{\nearrow})\} \\ \mathcal{H}_y^{\nearrow} &= \{((\mathcal{U} \backslash \{y\})) \backslash \mathrm{d}' \mid \mathrm{d} \in \underline{\mathrm{D}}, y \nearrow \mathrm{d}\} \\ y \nearrow \mathrm{d} &\iff \mathrm{d} \not\sqsubseteq \delta(y) \textit{ and if } \mathrm{h} \sqsubsetneq \mathrm{d} \textit{ then } \mathrm{h} \sqsubseteq \delta(y) \end{aligned}$$

$\mathcal{L}_{cc}$ is a canonical cover of functional dependencies.

For the running example, let us calculate the proper premises of attribute c using the arrow relations in Table 3. We have $c \nearrow (\delta(a) \sqcap \delta(b))$ and $c \nearrow (\delta(a) \sqcap \delta(d) \sqcap \delta(e))$. With this, we have the hypergraph $\mathcal{H}_c^{\nearrow} = \{\{d, e\}, \{b\}\}$ for which the minimum transversal hypergraph is $Tr(\mathcal{H}_c^{\nearrow}) = \{\{b, d\}, \{b, e\}\}$. Correspondingly, we have the FDs $bd \rightarrow c$ and $be \rightarrow c$ which are included in the canonical cover.

3.2 Characterizing a Minimal Cover of FDs

We introduce a novel characterization of a minimal cover of FDs by means of pseudo-intents, and its generalization using pseudo-extents of a PPS.

The stem base of implications, or Duquenne-Guigues basis [15], is a *sound*, *complete* and *non-redundant* basis which also has minimum cardinality among the sets of implications for a given formal context. We show how this can be used to characterize a minimal cover of FDs in a rather simple manner. Prior to introducing the stem base, let us define pseudo-closed sets [13].

Definition 2. *(Pseudo-closed sets) Let $P \mapsto P''$ be a closure operator over a set M, then P is a pseudo-closed set if and only if:*

$$P \neq P'' \tag{1}$$

$$Q \subsetneq P \text{ and } Q \text{ is a pseudo-closed set} \implies Q'' \subseteq P \tag{2}$$

Given a formal context (G, M, I), pseudo-closed sets $A \subseteq G$ are called pseudo-extents, while pseudo-closed sets $B \subseteq M$ are called pseudo-intents. A stem base of implications, or Duquenne-Guigues basis, can be defined as follows:

Theorem 3 ([13]). *(Duquenne-Guigues Basis) The set of implications:*

$$\mathcal{L} = \{P \rightarrow P'' \mid P \text{ is a pseudo-intent}\} \tag{3}$$

is sound, complete and non-redundant.

Theorem 4 ([13]). *Every complete set of implications contains an implication $X \rightarrow Y$ with $X'' = P''$ for every pseudo-intent P of (G, M, I)*

Theorem 4 entails that the stem base of implications has minimum cardinality with respect to any equivalent set of implications of (G, M, I). With this and the previous observation that FDs are in 1-1 correspondence with implications in $(Pair(T), \mathcal{U}, I)$, we can derive the following corollary.

Corollary 2. *Let $(Pair(T), \mathcal{U}, I)$ be the binary codification of a table with tuples T and attributes $\mathcal{U}$, the set of FDs $\mathcal{L} = \{P \rightarrow P'' \mid P$ is a pseudo-intent$\}$ is a minimal cover.*

Corollary 2 provides a novel characterization of the minimal cover of FDs through pseudo-intents of a formal context. Given the existing relation between $(Pair(T), \mathcal{U}, I)$ and $(\mathcal{U}, (\mathrm{D}, \sqsubseteq), \delta)$, we can generalize the characterization of the minimal cover over the latter. Observe that in the PPS $(\mathcal{U}, (\mathrm{D}, \sqsubseteq), \delta)$, we maintain the notion of pseudo-extents for a pseudo-closed set $X \subseteq \mathcal{U}$ with $X \mapsto X^{\square\square}$.

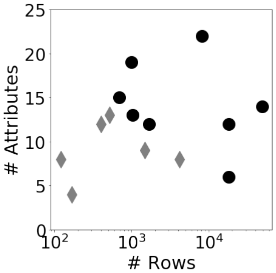


Fig. 1: Datasets and the number of rows and columns they contain.

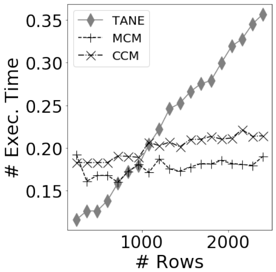


Fig. 2: Performance Comparison when increasing tuples: FCA vs TANE

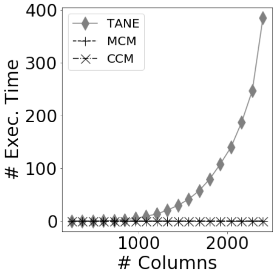


Fig. 3: Performance Comparison when increasing attributes: FCA vs TANE

Proposition 1. *Let $(\mathcal{U}, (\mathrm{D}, \sqsubseteq), \delta)$ be the PPS representation of a database, then the set of functional dependencies $\mathcal{L}_{mc}$ defined below is a minimal cover.*

$$\mathcal{L}_{mc} = \{X \to X^{\square\square} \mid X \textit{ is a pseudo-extent}\} \tag{4}$$

The proof of Proposition 1 follows from Theorem 1 and the fact that for a set $X \in \mathcal{U}$ the closure operator $X \mapsto X''$ is exactly equivalent to $X \mapsto X^{\square\square}$ and consequently, the set of pseudo-intents in $(Pair(T), \mathcal{U}, I)$ is the same as the set of pseudo-extents in $(\mathcal{U}, (\mathrm{D}, \sqsubseteq), \delta)$. Thus, because of Corollary 2, Proposition 1 holds.

Table 4: Dataset details, Execution Times in Seconds, and Number of Mined Rules for CCM (Canonical Cover Miner), MCM (Minimal Cover Miner) and TANE. CC: Canonical Cover, MC: Minimal Cover. Datasets in boldface represent those in which FCA performed better than TANE.

			CCM		MCM		TANE	
Dataset	# Tuples	# Attributes	# CC Deps	Time [S]	# MC Deps	Time [S]	# CC Deps	Time [S]
Mushroom	8124	22	3605	23887	1509	12684	-	-
Adult	48842	14	78	90.41	42	71.55	78	123.13
Credit	690	15	1099	3.07	253	1.82	1099	2.54
PGLW	17995	6	5	0.67	2	0.35	5	0.48
PGLW (2xA)	17995	12	38	1.81	15	1.18	38	7.45
Forest Fires	516	13	442	0.46	138	0.31	442	0.49
Forest Fires (2xT)	1032	13	442	1.27	138	0.78	442	2.34
ncvoter	1000	19	775	2.47	193	1.63	775	2.07
Diagnostics	120	8	37	0.08	17	0.06	37	0.06
Abalone	4177	8	137	0.41	40	0.29	137	0.32
CMC	1473	9	1	0.56	1	0.49	1	0.52
Hughes	401	12	3	0.12	3	0.17	3	0.06
Servo	167	4	1	0.05	1	0.03	1	0.02
Caulkins	1685	12	227	0.66	67	0.53	227	0.95

4 Experimental Evaluation

In this section we present a brief experimental comparison of both introduced approaches versus TANE [16], a state-of-the-art FD miner. TANE is a highly optimized *apriori*-based algorithm that generates a canonical cover of FDs. The goal of this evaluation is to study the comparative benefits of using FCA versus a traditional approach such as TANE. TANE was re-implemented for our experiments[5]. For the sake of repeatibility, the code used to run the experiments was made available at GitHub. Both the minimal cover miner (MCM) [6], and the canonical cover miner (CCM)[7] were implemented using Python's pipy FCA library `fca`[8].

Experiments were performed over 12 datasets extracted from the UCI Machine Learning repository[9], the JASA Data Archive[10] and Metanome's repeatability Web page[11]. Details on the number of rows and columns for each dataset are provided in the first two columns of Table 4. In addition to these datasets, we created synthetic versions by multiplying the rows or the columns of a given dataset. Experiments were performed on a virtual machine with 4 cores running at 2.7 Ghz and equipped with 32 GB of RAM memory.

4.1 Results & Discussion

Table 4 presents the main results of applying our approach and TANE on each dataset to mine the Minimal Cover and the Canonical Cover, respectively. The table contains the execution times for each approach and the number of dependencies mined. Datasets in boldface represent those for which CCM or MCM performed substantially better than TANE. For the Mushroom dataset, TANE was not able to obtain results before running out of memory, thereby no information is provided in the table. Table 4 also reports in two synthetic datasets, namely PGLW (2xA) which contains two horizontal copies of the PGLW dataset resulting in twice as many attributes. Forest Fires (2xT) contains two vertical copies of Forest Fires resulting in twice as many tuples. All Canonical Covers mined by TANE have been reduced to a Minimal Cover to verify the consistency of our approach.

Out of the 12 datasets, CCM and MCM performed better in the **largest** (both in rows and columns). This is better illustrated in Figure 1 where datasets are represented as points in a *number of rows-number of columns* space. Circles represent datasets for which CCM or MCM performed better while diamonds, where TANE did. Notice that the X axis is provided in logarithmic scale. The figure shows that most of the datasets where TANE performs better are in

[5] `https://github.com/cododedo/tane/tree/cla18`

[6] `https://github.com/cododedo/fd_miner/tree/cla18`

[7] `https://github.com/cododedo/uis_miner/tree/cla18`

[8] `https://pypi.org/project/fca/`

[9] `https://archive.ics.uci.edu/ml/index.php`

[10] `http://lib.stat.cmu.edu/jasadata/`

[11] `https://hpi.de/naumann/projects/repeatability/data-profiling/fds.html`

the lower-left region of the figure, representing small datasets. FCA-based approaches perform better in datasets in all other regions, including the upper-right which contains datasets with many rows and columns.

Synthetic datasets in Table 4 show evidence that FCA scales better when duplicating the dataset. When duplicating attributes the difference is particularly dramatic since TANE is over 13 times slower while our approach, only 3. To study this further, we created two sets of synthetic datasets. The first set (vertical set) was created by incrementally multiplying vertically the *Diagnostics* datasets (with 8 attributes and 120 tuples) generating 19 versions of 240, 360, 480 tuples, up to a dataset containing 2400 tuples. The second set (horizontal set) was created using the same idea but in a horizontal manner generating 19 versions of 16, 24, 32, up to 160 attributes. Since most of the datasets of the second set were too big for TANE, they were trunked to 40 tuples.

Figure 2 depicts the increasing time for CCM, MCM and TANE on the vertical set, i.e. when increasing the number of tuples. We can observe that all three approaches scale linearly w.r.t. the number of tuples, even when CCM and MCM seem to have a more stable behavior. Vertical multiplication of datasets yield the same number of FDs than the original set, since the relation between attributes remains unchanged. Thus, we can assume that other algorithms based on TANE could achieve a similar performance to CCM or MCM provided some optimizations.

On the other hand, this do not seem to be the case for the horizontal set. Figure 3 shows that CCM and MCM remain very stable when varying the number of attributes, while TANE's execution time grows exponentially. Indeed, this great difference in performance is due to the way in which we use FCA to find FDs which differs from TANE's strategy. Using FCA we calculate closures which quickly group attribute copies avoiding unnecessary intersections. Instead, TANE computes each attribute combination rendering the exponential growth in the computation time. We stress that this is not simply an extreme case from which our approach takes advantage, but actually a very good illustration of the benefits of using a closure operator to navigate the space of FDs. Closures enable CCM and MCM to avoid unnecessary computations not only when we have redundant attributes, but also whenever possible in the lattice of the powerset of attributes. Finally, we do not discuss on the differences between CCM and MCM strategies as these are detailed in [22].

5 Conclusions

We have presented a new characterization of a minimal cover of functional dependencies (FDs) by means of the stem base (or Duquenne-Guigues basis) of a partition pattern structure. We have presented the mechanisms through which this characterization can be exploited to efficiently mine the minimal cover. Furthermore, we have described the algorithms that implement these mechanisms and show empirical evidence that our characterization performs better than a state-of-the-art FD miner, namely TANE, in larger databases containing many rows and columns.

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).

References

1. Alam, M., Buzmakov, A., Codocedo, V., Napoli, A.: Mining definitions from RDF annotations using formal concept analysis. In: Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence, IJCAI 2015, Buenos Aires, Argentina, July 25-31, 2015. pp. 823–829. AAAI Press (2015)
2. Baixeries, J., Kaytoue, M., Napoli, A.: Computing Functional Dependencies with Pattern Structures. In: Concept Lattices and their Applications. pp. 175–186 (2012)
3. Baixeries, J., Kaytoue, M., Napoli, A.: Characterizing Functional Dependencies in Formal Concept Analysis with Pattern Structures. Annals of Mathematics and Artificial Intelligence 72(1–2), 129–149 (Oct 2014)
4. Bertet, K., Monjardet, B.: The multiple facets of the canonical direct unit implicational basis. Theoretical Computer Science 411(22-24), 2155–2166 (2010)
5. Caruccio, L., Deufemia, V., Polese, G.: Relaxed Functional Dependencies – A Survey of Approaches. IEEE Transactions on Knowledge and Data Engineering 28(1), 147–165 (2016)
6. Codocedo, V., Baixeries, J., Kaytoue, M., Napoli, A.: Characterization of Order-like Dependencies with Formal Concept Analysis. In: Concept Lattices and Their Applications. vol. 1624, pp. 123–134 (2016)
7. Codocedo, V., Napoli, A.: Lattice-based biclustering using Partition Pattern Structures. In: Proceedings of ECAI 2014. Frontiers in Artificial Intelligence and Applications, vol. 263, pp. 213–218. IOS Press (2014)
8. Fan, W.: Dependencies revisited for improving data quality. In: Proceedings of the Twenty-Seventh ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems, PODS 2008, June 9-11, 2008, Vancouver, BC, Canada. pp. 159–170 (2008)
9. Fan, W.: Data quality: From theory to practice. SIGMOD Record 44(3), 7–18 (2015)
10. Fan, W., Geerts, F.: Foundations of Data Quality Management. Synthesis Lectures on Data Management, Morgan & Claypool Publishers (2012)
11. Gainer-Dewar, A., Vera-Licona, P.: The minimal hitting set generation problem: Algorithms and computation. SIAM Journal on Discrete Mathematics 31(1), 63–100 (2017)
12. Ganter, B., Obiedkov, S.: Conceptual Exploration. Springer, Berlin (2016)
13. Ganter, B., Wille, R.: Formal Concept Analysis. Springer, Berlin (1999)
14. Ganter, B., Kuznetsov, S.O.: Pattern Structures and their projections. In: Delugach, H.S., Stumme, G. (eds.) Conceptual Structures: Broadening the Base. LNCS, vol. 2120, pp. 129–142. Springer (2001)
15. Guigues, J.L., Duquenne, V.: Familles minimales d'implications informatives résultant d'un tableau de données binaires. Mathématiques et Sciences Humaines 95, 5–18 (1986)

16. Huhtala, Y., Kärkkäinen, J., Porkka, P., Toivonen, H.: TANE: An Efficient Algorithm for Discovering Functional and Approximate Dependencies. Computer Journal 42(2), 100–111 (1999)
17. Korth, H.F., Silberschatz, A.: Database System Concepts. McGraw-Hill, Inc., New York, NY, USA (1986)
18. Kuznetsov, S.O.: Galois Connections in Data Analysis: Contributions from the Soviet Era and Modern Russian Research. In: Formal Concept Analysis, Foundations and Applications. pp. 196–225. Lecture Notes in Computer Science 3626, Springer (2005)
19. Maier, D.: Theory of Relational Databases. Computer Science Press (1983)
20. Mannila, H., Räihä, K.J.: The Design of Relational Databases. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA (1992)
21. Papenbrock, T., Ehrlich, J., Marten, J., Neubert, T., Rudolph, J.P., Schönberg, M., Zwiener, J., Naumann, F.: Functional Dependency Discovery: An Experimental Evaluation of Seven Algorithms. Proceedings of VLDB Endowment 8(10), 1082–1093 (2015)
22. Ryssel, U., Distel, F., Borchmann, D.: Fast algorithms for implication bases and attribute exploration using proper premises. Annals of Mathematics and Artificial Intelligence 70(1), 25–53 (2014)
23. Ullman, J.: Principles of Database Systems and Knowledge-Based Systems, volumes 1–2. Computer Science Press, Rockville (MD), USA (1989)
24. Yu, Y., Heflin, J.: Extending functional dependency to detect abnormal data in RDF graphs. In: The Semantic Web - ISWC 2011 - 10th International Semantic Web Conference, Bonn, Germany, October 23-27, 2011, Proceedings, Part I. pp. 794–809 (2011)
25. Yu, Y., Li, Y., Heflin, J.: Detecting abnormal semantic web data using semantic dependency. In: Proceedings of the 5th IEEE International Conference on Semantic Computing (ICSC 2011), Palo Alto, CA, USA, September 18-21, 2011. pp. 154–157. IEEE Computer Society (2011)

Author Index

Editors:	Dmitry I. Ignatov, Lhouari Nourine
Title:	CLA 2018, Proceedings of the Fourteenth International Conference on Concept Lattices and Their Applications
Technical Editor:	Jan Outrata, `jan.outrata@upol.cz`
Cover Design:	Martin Trnečka, `martin.trnecka@gmail.com`
Executive Editor:	Karel Lemr
In-house Editor:	Otakar Loutocký

Publisher & Print:	Palacký University Olomouc
	Křížkovského 8, 771 47 Olomouc, Czech Republic
	`www.vydavatelstvi.upol.cz`
	`www.e-shop.upol.cz`
	`vup@upol.cz`

Place, year, edition:	Olomouc, Czech Republic, 2018, 1st
Page count:	xii+291
Impression:	70
Archived at:	`cla.inf.upol.cz`

The publication has not been subjected to any language editing review by the publishing house.

Not for sale

VUP 2018/0132

ISBN 978–80–244–5328–6