Workshop Notes

Sixth International Workshop
"What can FCA do for Artificial Intelligence?"
FCA4AI 2018

International Joint Conference on Artificial Intelligence
European Conference on Artificial Intelligence
IJCAI/ECAI 2018

July 13 2018

Stockholm, Sweden

Editors
Sergei O. Kuznetsov (NRU HSE Moscow)
Amedeo Napoli (LORIA Nancy)
Sebastian Rudolph (TU Dresden)

http://fca4ai.hse.ru/2018/

Preface

The five preceding editions of the FCA4AI Workshop showed that many researchers working in Artificial Intelligence are deeply interested by a well-founded method for classification and mining such as Formal Concept Analysis (see `http://www.fca4ai.hse.ru/`). The first edition of FCA4AI was co-located with ECAI 2012 in Montpellier, the second one with IJCAI 2013 in Beijing, the third one with ECAI 2014 in Prague, the fourth on with IJCAI 2015 in Buenos Aires, and finally the fifth one with ECAI 2016 in The Hague. All the proceedings of the preceding editions are published as CEUR Proceedings (`http://ceur-ws.org/Vol-939/`, `http://ceur-ws.org/Vol-1058/`, `http://ceur-ws.org/Vol-1257/`, and `http://ceur-ws.org/Vol-1430/`, and `http://ceur-ws.org/Vol-1703/`).

This year, the workshop has again attracted many different researchers working on actual and important topics, e.g. theory, extensions of FCA (MDL), classification, mining of linked data, annotation, biclustering, recommendation and applications. This shows the diversity and the richness of the relations between FCA and AI.

Formal Concept Analysis (FCA) is a mathematically well-founded theory aimed at data analysis and classification. FCA allows one to build a concept lattice and a system of dependencies (implications) which can be used for many Artificial Intelligence needs, e.g. knowledge discovery, learning, knowledge representation, reasoning, ontology engineering, as well as information retrieval and text processing. As we can see, there are many "natural links" between FCA and Artificial Intelligence. Recent years have been witnessing increased scientific activity around FCA, in particular a strand of work emerged that is aimed at extending the possibilities of FCA w.r.t. knowledge processing, such as work on pattern structures and relational context analysis. These extensions are aimed at allowing FCA to deal with more complex than just binary data, both from the data analysis and knowledge discovery points of view and as well from the knowledge representation point of view, including, e.g., ontology engineering. All these investigations provide new possibilities for Artificial Intelligence activities in the framework of FCA. Accordingly, in this workshop, we are interested in main issues such as:

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

The workshop is dedicated to discuss such issues. This year, the papers submitted to the workshop were carefully peer-reviewed by three members of the program committee and 11 papers with the highest scores were selected. We thank all the PC members for their reviews and all the authors for their contributions.

The Workshop Chairs

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

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

Sebastian Rudolph
Technische Universität Dresden, Germany

Program Committee

Mehwish Alam (STLab, CNR Bologna, Italy)

Jaume Baixeries (UPC Barcelona, Catalunya)

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

Aleksey Buzmakov (National Research University HSE Perm, Russia)

Loïc Cerf (UFMG Belo-Horizonte, Brazil)

Victor Codocedo (UFTSM Santiago de Chile, Chile)

Mathieu D'Aquin (The National University of Ireland, Galway)

Florent Domenach (Akita International University, Japan)

Marianne Huchard (LIRMM/Université de Montpellier, France)

Dmitry I. Ignatov (National Research University HSE Moscow, Moscow, Russia)

Mehdi Kaytoue (INSA-LIRIS Lyon, France)

Jan Konecny (Palacky University, Olomouc, Czech Republic)

Florence Le Ber (ENGEES/Université de Strasbourg, France)

Nizar Messai (Université de Tours, France)

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

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

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

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

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

Henry Soldano (Université de Paris-Nord, France)

Laszlo Szathmary, University of Debrecen, Hungary

Diana Troancă (Babes-Bolyai University, Cluj-Napoca, Romania)

Renato Vimiero (UFPE Recife, Brazil)

Contents

Invited Talk
Inductive Reasoning with Conceptual Space Representations

Zied Bouraoui

Zied Bouraoui
CRIL CNRS & Université Artois
Lens, France
bouraoui@cril.univ-artois.fr

Abstract. Structured knowledge is playing an increasingly important role in areas such as natural language processing and information retrieval. Such applications differ from the settings that have traditionally been considered in the field of knowledge representation, in that they require knowledge bases with a wide coverage, even if that means accepting some inaccuracies. In this talk, I will present some methods for knowledge base completion. At the center of this work are conceptual spaces, which are geometric representations of knowledge that were proposed by Gärdenfors (2000) as an intermediate representation level between symbolic and connectionist representations. In conceptual spaces, objects from a domain of interest are represented as points in a metric space, and concepts are modeled as convex regions. I will first present how to learn conceptual space representations from data, and then introduce some inductive reasoning techniques that use conceptual spaces together with an efficient Bayesian inference machinery that allows us to find plausible missing facts and rules from a given knowledge base.

An Answer Set Programming environment for high-level specification and visualization of FCA

Lucas Bourneuf ✉

Université de Rennes 1
Campus de Beaulieu, 35042 Rennes cedex, France.
lucas.bourneuf@inria.fr

Abstract. This paper introduces Biseau, a programming environment dedicated to the exploration of relations through a graphical display. The use of Answer Set Programming enables the production of small code modules which are easy to maintain and debug since they are very close to the specifications. This paper shows how a mathematical framework such as Formal Concept Analysis can be efficiently described at the level of its properties, without needing a costly development process. We hope that it will help to quickly adapt a given code to the peculiarities of a data set, thereby speeding up the development of prototypes. Besides, it will also help the integration of the ideas of the FCA community in a readable and shareable format. From a practical point of view, Biseau provides an Answer Set Programming to (graphviz) dot compiler and uses the graphviz software to render in real-time the calculated graphs to user, for instance to produce concept lattices or aoc posets visualizations. Its relation with existing tools like LatViz and FCAbundles is also discussed.

1 Introduction

Large scale data production requires availability of high-level visualizations for their exploration. This is usually performed by building generic visualization models, that users may later use to explore their data. Thus, software environments oriented towards data mining use efficient implementations of data structures and their visualizations. For instance, in Formal Concept Analysis, LatViz is a lattice visualization software, allowing end-user to explore the lattice structure efficiently [1]. Lattice Miner builds and visualize Galois lattices and provides data mining tools to explore data [15]. FCA Tools Bundle consists in a web interface exposing multiple FCA-related tools for contexts and (ternary) concept lattices exploration [14]. In-Close algorithm reference implementation provides a concept trees visualization of contexts encoded in standard formats [3]. All these tools work with a formal model that provides an abstract view and a fixed search space on the data. Users cannot work on the model itself, they are expected to use the implemented methods, not to design new ones. In contrast, this paper introduces Biseau, a software focused on designing and exploring elements of the data structure, rather than the data itself. In this approach, data are only a support to the model validity, and the user's aim is the proper design of a general model. Biseau is a general purpose model builder that relies on graphs and logic languages.

Graphs are rendered in multiple ways, using field-specialized softwares like Cytoscape [21] in biology, graph-specific softwares (like LatViz for lattices), or more generalist like Tulip [4]. Another generic approach is dot, a graph description language specified by the graphviz software, which provides a gallery of visualization engines [6]. Dot is the internal graphical language used by Biseau (see Section 3).

Together with a graph data structure, Biseau offers a logical view of the associated exploration methods. A pure declarative language is used for this purpose, Answer Set Programming (ASP). It allows users to transcript the formal properties they are looking for in a straightforward way (see Section 2). ASP has already been applied to FCA to accomplish expressive query languages for formal contexts [12], later extended to n-adic FCA and improved with additional membership constraints, in order to handle large context exploration [20]. In our approach, ASP is also used for visualization.

Biseau is supplied with a graphical user interface and a command line interface to write an ASP encoding. Biseau uses this encoding as a script to generate the dot files and the resulting visualizations. The main interest of Biseau is therefore to build graph visualizations directly from formal relations. Biseau is not only dedicated to lattices, and their (efficient or scalable) exploration. It provides instead a general purpose programming environment that is able to visualize any ordered structure. Biseau is therefore suited for rapid design and easy testing of works or extensions in the framework of FCA. It is freely available under the GNU/GPL license[1].

The structure of this paper is as follows. Sections 2 and 3 quickly present the ASP and dot languages used by Biseau. Section 4 explains how Biseau takes advantage of these languages to allow the user to build models. Section 5 proposes as a case study the reconstruction of the Galois lattice. Section 6 shows how Biseau can easily handle some of the FCA extensions typically used in FCA applications as knowledge processing [19]. Finally the paper concludes by some insights about Biseau interest when used in FCA and in artificial intelligence.

2 Answer Set Programming

The following presentation of ASP is taken from [5]. For an in-depth dive into the language, the reader is redirected to [8].

ASP is a form of purely declarative programming oriented towards the resolution of combinatorial problems [17]. It has been successfully used for knowledge representation, problem solving, automated reasoning, and search and optimization. Unlike Prolog, ASP handles cross-references of rules, enabling the writing of code much closer to the specification. In the sequel, we rely on the input language of the ASP system Potassco (*Potsdam Answer Set Solving Collection* [8])

[1] `https://gitlab.inria.fr/lbourneu/biseau`

developed in Potsdam University. An ASP program consists of Prolog-like rules h :- $b_1, \ldots, b_m$, *not* $b_{m+1}, \ldots,$ *not* b_n, where each b_i and h are literals and *not* stands for *default negation*. Mainly, each literal is a predicate whose arguments can be constant atoms or variables over a finite domain. Constants start with a lowercase letter, variables start with an uppercase letter or an underscore (don't care variables). The rule states that the head h is proved to be true (h is in an answer set) if the body of the rule is satisfied, i.e. $b_1, \ldots, b_m$ are true and one can not prove that $b_{m+1}, \ldots, b_n$ are true. Note that the result is independent on the ordering of rules or of the ordering of literals in their body, as it is the case in Prolog. An ASP solver can compute one, several, or all the answer sets (stable models) that are solutions of the encoded problem. If the body is empty, h is a fact while an empty head specifies an integrity constraint. Together with model minimality, interpreting the program rules this way provides the stable model semantics (see [11] for details). In the head part, A *choice rule* of the form $\{p(X) : q(X)\}$ will generate $p(X)$ as the powerset of $q(X)$ for all values of X. In the body part, $\{p(\_)\}$ will count the number of atom p with one parameter, and $N = \{h\}$ evaluates N to the cardinal of the set of h. In the body part, $p(X) : q(X)$ holds if for all X, if $q(X)$ holds, then $p(X)$ holds. Finally, lines starting by `%` are comments. In practice, several syntactical extensions to the language that are not interesting for this paper are available. An example of ASP encoding is presented in Figure 1, using atoms to reproduce the context in Table 1 and a rule to build a bipartite graph linking objects and attributes.

```
% Facts.
age(john,7). age(eve,71). age(alice,15).
male(john). male(bob). female(alice).
mother(eve,bob).
% Rules.
rel(H,child):− age(H,A) ; A<12.
rel(H,adult):− age(H,A) ; A>=18.
rel(H,male):− male(H).
rel(H,female):− female(H).
rel(H,man) :− rel(H,male) ; rel(H,adult).
rel(H,boy) :− rel(H,male) ; rel(H,child).
rel(H,woman):− rel(H,female) ; rel(H,adult).
rel(H,girl) :− rel(H,female) ; rel(H,child).
rel(H,adult):− rel(H,male) ; not rel(H,boy).
rel(H,female):− mother(H,_).
% Build the visualization in Figures 2 and 3.
link(O,A):− rel(O,A).
```

Fig. 1: ASP program encoding the context in Table 1, in the form of `rel/2` relations between objects and attributes. The last line yield links/2 atoms that are compiled by Biseau as edges in the output dot file.

We used the Potassco system [9] that proposes an efficient implementation of ASP. ASP processing implies two steps, grounding and solving. The grounder generates a propositional program replacing variables by their possible values. The solver is in charge of producing the stable models (answer sets) of the propositional program. Of course, a dedicated algorithm for a specific problem will be generally more efficient than its equivalent compact ASP encoding. However, ASP systems are useful for the design of prototypes. It is an attractive alternative to standard imperative languages that enable fast developments.

	adult	child	female	male	boy	woman	man
alice			×				
bob	×			×			×
eve	×		×			×	
john		×		×	×		

Table 1: Formal context of human relations.

3 Graph Drawing With Dot

Dot is a graph description language, allowing one to generate a graph visualization from the definition of its content [6]. Dot enables the control of precise visual properties, such as node and edge labelling, position, shape, or color. For instance, the dot line `woman [color="blue"]` will color in blue the node labelled *woman*. The full language is defined by the graphviz graph visualization software, which provides multiple engines to interpret and compile dot encoded files to other formats, including images. Figure 2 shows an example of a working dot description, which given to a graphviz engine yields the visualization in Figure 3.

```
Digraph biseau_graph {
  node [penwidth="0.4" width="0.1"];
  edge [penwidth="0.4" arrowhead="none"];
  john->boy; john->male; john->child;
  eve->female; eve->woman; eve->adult;
  bob->man;  bob->adult; bob->male;
  alice->female;
}
```

Fig. 2: Dot encoding of the graph in Figure 3.

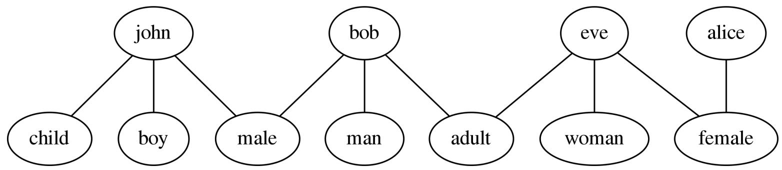


Fig. 3: Visualization of the relations described by context in Table 1.

4 From ASP to Dot With Biseau

Biseau allows the user to write some ASP encoding and retrieve in real-time the corresponding graph visualization. To achieve this, it implements an ASP to dot compiler and a Graphical User Interface that helps writing the ASP encoding and that performs automatically all necessary compilations.

As explained in Section 2, a given ASP encoding yields stable models consisting of true facts, which can be represented by atoms like `link(woman,human)`. For each stable models found from the ASP user encoding, Biseau will convert atoms into dot lines. For instance, the ASP atom `link(woman,human)` will translate to `woman -> human` in the dot output. This controlled vocabulary will be only partially explored in Section 5, but note that it maps the full dot language, including colors, shapes, and general graph options. A complete documentation is available online[2].

Because of the use of ASP to yield the dot description, the graph is therefore defined in intension: instead of describing manually all objects and properties, the user specify their definitions, and let the ASP solver infers all necessary relations. More generally, Biseau internal process can be seen as a compilation from ASP models to dot, then from dot to image (the last one being delegated to graphviz software, as seen in Section 3).

As a matter of example, the ASP encoding in Figure 1 will be compiled to the dot description in Figure 2, itself compiled to the image in Figure 3. If the ASP expression `color(A,blue):- rel(_,A).` was added to the ASP encoding in Figure 1, the final figure would show in blue all attributes nodes. The reader familiar with software engineering may recognize the use of ASP as a metamodel, and dot as the model.

Biseau can be extented with *scripts*, units of ASP (or Python) code to add to (or run on) the user encoding. They may expose some options to tune their behavior. Moreover, user can implement and add its own scripts to Biseau, allowing him (and others he shares with) to encapsulate ASP or Python programs that behave accordingly to their preferences. Biseau is shipped with scripts related to FCA, for data extraction from standard format like SLF or CXT, concept mining or lattice visualization (as shown in Section 5).

[2] `https://gitlab.inria.fr/lbourneu/biseau/blob/master/doc/user-doc.mkd`

5 Build and Visualize Galois Lattices With Biseau

This section shows how to build FCA basic mathematical relations in order to get a visualization of the Galois lattice in Biseau. The context in Table 1 will be used as case study, encoded in ASP using `rel/2` atoms as shown in the first five lines of Figure 1.

5.1 Mining the Formal Concepts

In a formal context defined by objects O, attributes A, and the binary relation $R \subseteq O \times A$, a formal concept is a pair (X, Y), such as:

$$X = \{y \in Y \mid (x, y) \in R \;\; \forall x \in X\} \quad (1)$$

$$Y = \{x \in X \mid (x, y) \in R \;\; \forall y \in Y\} \quad (2)$$

Where $X \subseteq O$ and $Y \subseteq A$. The search for formal concepts in ASP can be expressed like in the above definition:

```
ext(X):- rel(X,_) ; rel(X,Y): int(Y).
int(Y):- rel(_,Y) ; rel(X,Y): ext(X).
```

`rel(X,_)` fixes variable `X` as the first term of a relation, i.e. an object. Notation `rel(X,Y): int(Y)` ensure that there is a relation between `X` and all attributes of the intent. As a consequence, `ext(X)`, the extent, holds for all objects in relation with all attributes of the intent. The second rule is a symmetric definition for the concept's intent. For those familiar to Prolog, note that such a program would lead to an infinite loop. The treatment of loops is a nice feature of ASP that gives access to a fixed-point semantics. ASP search comes with the guarantee that all minimal fixed points will be enumerated. Therefore, each answer set is a different concept, or the supremum or infimum (where extent or intent are empty sets). To avoid the yield of supremum (infimum), one may include a constraint specifying that extent (intent) must include at least one element.

These models/concepts can be aggregated in order to produce an encoding containing `ext/2` (and `int/2`) atoms, where `ext(N,A)` (`int(N,A)`) gives an element of N-th concept's extent (intent). This numbering is arbitrary and serves no other purpose than identifying the different concepts.

5.2 Galois lattice

A Galois lattice is defined by the partial order on the concepts, i.e. a graph with concepts as nodes, and an edge between a concept and its successors in the ordering:

```
% Shortcut to infimum, supremum and concepts identifiers.
c(N):- ext(N,_).
c(N):- int(N,_).
% Ordering of two concepts: the first has all objects of the second.
```

```
5 contains(C1,C2):- c(C1) ; c(C2) ; C1!=C2 ; ext(C1,X): ext(C2,X).
6 % Concepts linked to another in the Galois Lattice.
7 link(C1,C3):- contains(C1,C3) ; not link(C1,C2): contains(C2,C3).
8 % Annotate nodes with extent and intent.
9 annot(upper,X,A):- ext(X,A).
10 annot(lower,X,B):- int(X,B).
```

These lines yield the visualization shown in Figure 4. Line 2 and 3 are here to enable the access to the infinum, supremum and concepts with one atom. Line 5 yields pairs of concepts that are included, based on their extent. Line 7 ensure that a link exists in the lattice between a concept $C1$ containing another concept $C3$ if there no link between $C1$ and a concept $C2$ smaller than $C3$. Finally, the `annot/3` atoms are a Biseau convention (just as `link/2` that define an edge in the dot output), allowing us to print the extent and intent of each concept, respectively above and below the node.

5.3 Reduced Labelling

The reduced labelling of a lattice is computed as the set of specific objects and attributes for each concept. This is easily defined as `specext/1` and `specint/1` atoms in ASP, using the following lines along the search for formal concepts in section 5.1:

```
% An outsider is any object or attribute linked to an attribute or object not in
    the concept.
outsider(X):- ext(X) ; rel(X,Z) ; not int(Z).
outsider(Y):- int(Y) ; rel(Z,Y) ; not ext(Z).
% The specific part of each concept contains no outsider.
specext(X):- ext(X) ; not outsider(X).
specint(Y):- int(Y) ; not outsider(Y).
```

With these lines and the collapsing into one model described in section 5.1, we obtain `specext/2` and `specint/2` atoms, describing the AOC poset elements, attached to each concept. We can then compute the reduced labelling of the lattice with the following lines, replacing the previously defined annot/3 definitions in section 5.2:

```
% Minimalist annotation of nodes with their extent/intent:
annot(upper,X,A):- specext(X,A).
annot(lower,X,B):- specint(X,B).
```

Using these definitions, Biseau produces the visualization shown in Figure 5.

6 Pulling Constraints On The Model

This section exposes the implementation in ASP and Biseau of some FCA variants and extensions often used in knowledge processing [19].

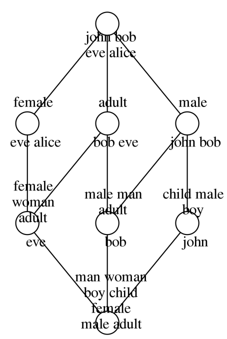


Fig. 4: Visualization of the Galois Lattice of context in Table 1 using Biseau, with extent and intent shown for each node/concept.

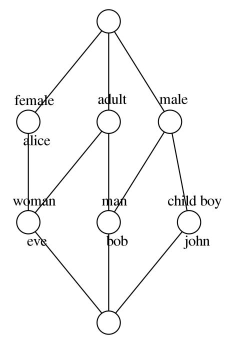


Fig. 5: Visualization of the Galois Lattice of context in Table 1 using Biseau, with reduced labelling.

6.1 Object and Property Oriented Concept Lattices

Following definitions from [23], it is also possible to encode the mining of object oriented concepts (X, Y) defined by $X = Y^\Diamond$ and $Y = X^\Box$, such as:

$$Y^\Diamond = \bigcup_{y \in Y} Ry \qquad\qquad X^\Box = \{y \in A | Ry \subseteq X\}$$

With $Ry = \{x \in O | (x, y) \in R\}$.

```
% Any object linked to an attribute in the intent is in the extent.
ext(X):- rel(X,Y) ; int(Y).
% Objects in the complementary set of the extent.
not_ext(Nx):- rel(Nx,_) ; not ext(Nx).
% The intent is made of attributes exclusively linked to objects of the extent.
int(Y):- rel(_,Y) ; not rel(Nx,Y): not_ext(Nx).
```

The code for property-oriented concepts is similar, and both replace the encoding in section 5.1.

6.2 Iceberg Lattices

The iceberg lattice, loosely defined as the Galois lattice stripped of all concepts with a too small support (i.e. number of objects in their extents) [22], can be built by discarding any model containing too few objects in his extent. For instance, the Figure 3 of [22], reproduced in this paper in Figure 6, shows the iceberg lattice of the running example MUSHROOMS database of `nbobj` objects with a minimal support of `minsupp`%. It can be reproduced by discarding models using a constraint:

```
% The number of ext/1 atoms must not fall behind the minimal.
:- {ext(_)} < nbobj*minsupp/100.
```

This constraint can be generated by Biseau knowing the number of objects and the minimal support.

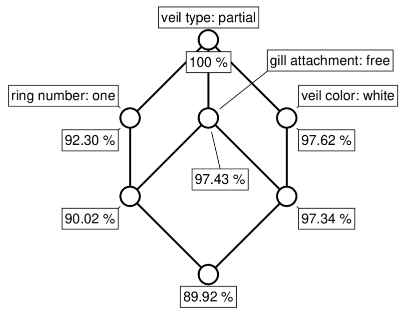


Fig. 6: Iceberg lattice with a minimal support of 85% of the MUSHROOMS database. Figure extracted from [22].

6.3 n-adic FCA

n-adic FCA [16] can be encoded the same way as regular FCA, by extending the number of parameters for `rel` atoms. For instance, in triadic FCA, conditions are given as the third argument of `rel/3` atoms, such as `rel(O,A,C)` is true when the relation between object O, attribute A and condition C holds. Triadic concepts can thus be generated using the following encoding:

```
ext(X):- rel(X,_,_) ; rel(X,A,C): int(A), cnd(C).
int(X):- rel(_,X,_) ; rel(O,X,C): ext(O), cnd(C).
cnd(X):- rel(_,_,X) ; rel(O,A,X): ext(O), int(A).
```

6.4 Pattern Structures

As introduced in [7], a pattern structure is a generalization of FCA applied on attributes structured in semi-lattices. Pattern concepts are pairs of objects and lattices, producing the expected pattern lattice. This technics have been applied to gene expression data [13]. Here, we reproduce the pattern lattice construction for an example of non-binary data from the same publication:

```
rel(1,1,5).   rel(1,2,7).   rel(1,3,6).   % 5 objects
rel(2,1,6).   rel(2,2,8).   rel(2,3,4).   % 3 situations
rel(3,1,4).   rel(3,2,8).   rel(3,3,5).   % one value from 4 to 9
rel(4,1,4).   rel(4,2,9).   rel(4,3,8).
rel(5,1,5).   rel(5,2,8).   rel(5,3,5).
```

Note that data are encoded in `rel/3` atoms over 5 objects, 3 conditions, and expression values in the interval [4; 9] associated with a given gene and condition, such as `rel(O,S,V)` holds when object O in situation S has an expression value of V. Similarly to section 5.1, we can enumerate the pattern concepts:

```
1  % Choose a subset of objects as the extent.
2  { ext(O): rel(O,_,_) }.
3  % The intervals of extent.
4  interval(C,Min,Max):- rel(_,C,_) ; Min=#min{V,O: rel(O,C,V), ext(O)} ;
5                        Max=#max{V,O: rel(O,C,V), ext(O)}.
6  % Object is valid on Condition.
7  valid_on(O,C):- rel(O,C,V) ; interval(C,Min,Max) ; Min<=V ; V<=Max.
8  % Object is valid for all Conditions.
9  valid(O):- rel(O,_,_) ; valid_on(O,C): rel(_,C,_).
10 % Avoid any model that do not include maximal number of objects.
11 :- not ext(O) ; valid(O).
```

The use of the meta-programming directives `#min` and `#max` allows us to retrieve the minimal and maximal value associated to the extent. Therefore, `interval(C,Min,Max)` stands for the minimal and maximal values on condition C, e.g. 5 and 6 for condition 1 when extent is $\{1, 2, 5\}$. Unlike the concept model seen in Section 5.1, this model relies on an explicit choice rule for the extent with subsequent constraints to ensure its maximality. Line 2 generates an answer set for each element of the power set of the object set. Following lines will discard answer sets that are not infinum, supremum or concept. Line 4 associate for each condition the minimal and maximal values over the extent. Line 7 selects an object and a condition such as they are associated to a value in the interval. Line 9 selects all objects that are valid for all conditions, and line 11 ensure that they belong to the extent.

The code in section 5.2 can be reused without modifications to produce and show the resulting pattern lattice.

7 Discussion & Conclusion

Using the ASP language in the Biseau environment, some well-known FCA structures (Galois, object-oriented, iceberg, integer pattern lattices) have been reconstructed. The main contribution of Biseau lies into the straightforward use of the structure specifications to produce a simple code and a proper visualization. To achieve that feat, Biseau is compiling a controlled subset of ASP atoms to dot lines, effectively building a dot formatted file that is compiled to an image by graphviz software. By letting the user manipulate the visualization with the full power of ASP, Biseau enables definition of graphs in intension. This gives an abstract access to dot expressions and lets the user focus on the fast prototyping of data exploration and the elaboration of mathematical properties. In other words, Biseau allows user to work on the model in which data are processed, instead of providing an implementation of a single model to be used on particular data, as usually performed in field-specialized softwares.

ASP limits lies into the absence of float numbers handling, and scaling problems inherited from the total grounding of data before solving. However, Potassco system users may benefit from several extensions of the language like linear programming [18] or propagators [10], allowing one to take advantage of other programming paradigms, or improving performances by an iterative replacement of bottlenecks by dedicated algorithms. For instance, the standard concept mining can be replaced by an implementation of the in-close algorithm [2].

Biseau current state is a very simple proof of concept, and therefore miss a lot of features typically found in Integrated Development Environments, that could help user to write, understand and debug produced ASP code.

Future work will focus on the Biseau generalization : other languages like GML allow to describe graphs, some are field-specific, and some enable outsourcing of the visualization to other (field-)specialized softwares. Future development of Biseau could provide support for ASP advanced features, and embedding of more scripts for FCA and its extensions.

References

1. M. Alam, T. N. N. Le, and A. Napoli. Steps towards interactive formal concept analysis with latviz. In *FCA4AI@ECAI*, 2016.
2. S. Andrews. In-close, a fast algorithm for computing formal concepts. 2009.
3. S. Andrews and L. Hirsch. A tool for creating and visualising formal concept trees. In *CEUR Workshop Proceedings*, volume 1637, pages 1–9. Tilburg University, 2016.
4. D. Auber, D. Archambault, R. Bourqui, M. Delest, J. Dubois, A. Lambert, P. Mary, M. Mathiaut, G. Mélançon, B. Pinaud, B. Renoust, and J. Vallet. TULIP 5. pages 1–28, Aug. 2017.
5. L. Bourneuf and J. Nicolas. *FCA in a Logical Programming Setting for Visualization-Oriented Graph Compression*, pages 89–105. Springer International Publishing, Cham, 2017.
6. E. R. Gansner and S. C. North. An open graph visualization system and its applications to software engineering. *Softw. Pract. Exper.*, 30(11):1203–1233, Sept. 2000.
7. B. Ganter and S. O. Kuznetsov. Pattern structures and their projections. In *International Conference on Conceptual Structures*, pages 129–142. Springer, 2001.
8. M. Gebser, R. Kaminski, B. Kaufmann, , M. Lindauer, M. Ostrowski, J. Romero, T. Schaub, and S. Thiele. *Potassco User Guide*, 2015.
9. M. Gebser, R. Kaminski, B. Kaufmann, M. Ostrowski, T. Schaub, and M. Schneider. Potassco: The Potsdam answer set solving collection. *AI Communications*, 24(2):107–124, 2011.
10. M. Gebser, R. Kaminski, B. Kaufmann, M. Ostrowski, T. Schaub, and P. Wanko. Theory Solving Made Easy with Clingo 5. In M. Carro, A. King, N. Saeedloei, and M. D. Vos, editors, *Technical Communications of the 32nd International Conference on Logic Programming (ICLP 2016)*, volume 52 of *OpenAccess Series in Informatics (OASIcs)*, pages 2:1–2:15, Dagstuhl, Germany, 2016. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik.
11. M. Gelfond and V. Lifschitz. Logic programs with classical negation. In *Proc. of the 7th International Conf. on Logic Programming (ICLP)*, pages 579–97, 1990.
12. P. Hitzler and M. Krötzsch. Querying formal contexts with answer set programs. In *Proc. of the 14th Int. Conf. on Conceptual Structures: Inspiration and Application*, ICCS'06, pages 260–273. Springer-Verlag, 2006.

13. M. Kaytoue, S. O. Kuznetsov, A. Napoli, and S. Duplessis. Mining gene expression data with pattern structures in formal concept analysis. *Information Sciences*, 181(10):1989 – 2001, 2011. Special Issue on Information Engineering Applications Based on Lattices.
14. L. L. Kis, C. Sacarea, and D. Troanca. Fca tools bundle-a tool that enables dyadic and triadic conceptual navigation. *Proc. of FCA4AI*, 2015.
15. B. Lahcen and L. Kwuida. Lattice miner: a tool for concept lattice construction and exploration. In *Suplementary Proceeding of International Conference on Formal concept analysis (ICFCA'10)*, 2010.
16. F. Lehmann and R. Wille. A triadic approach to formal concept analysis. In G. Ellis, R. Levinson, W. Rich, and J. F. Sowa, editors, *Conceptual structures: applications, implementation and theory*, number 954 in Lecture Notes in Artificial Intelligence, pages 32–43, Berlin–Heidelberg–New York, 1995. Springer–Verlag.
17. V. Lifschitz. What is answer set programming? In *Proc. of the 23rd National Conf. on Artificial Intelligence - Vol. 3*, AAAI'08, pages 1594–97. AAAI Press, 2008.
18. G. Liu, T. Janhunen, and I. Niemelä. Answer set programming via mixed integer programming. In *KR*, pages 32–42, 2012.
19. J. Poelmans, D. I. Ignatov, S. O. Kuznetsov, and G. Dedene. Formal concept analysis in knowledge processing: A survey on applications. *Expert Systems with Applications*, 40(16):6538 – 6560, 2013.
20. S. Rudolph, C. Săcărea, and D. Troancă. Membership constraints in formal concept analysis. In *Proc.of the 24th Int. Conf. on Artificial Intelligence*, IJCAI'15, pages 3186–3192. AAAI Press, 2015.
21. P. Shannon, A. Markiel, O. Ozier, N. S. Baliga, J. T. Wang, and D. e. a. Ramage. Cytoscape: A software environment for integrated models of biomolecular interaction networks. *Genome Research*, 13(11):2498–2504, 2003.
22. G. Stumme, R. Taouil, Y. Bastide, and L. Lakhal. Conceptual clustering with iceberg concept lattices. In *In: Proc. of GI-Fachgruppentreffen Maschinelles Lernen'01, Universität Dortmund*, 2001.
23. Y. Yao. Concept lattices in rough set theory. In *Fuzzy Information, 2004. Processing NAFIPS'04. IEEE Annual Meeting of the*, volume 2, pages 796–801. IEEE, 2004.

Three Approaches for Mining Definitions from Relational Data in the Web of Data

Justine Reynaud, Yannick Toussaint, and Amedeo Napoli

Université de Lorraine, CNRS, Inria, LORIA, F-54000 Nancy, France
firstname.lastname@loria.fr

Abstract. In this paper we study a classification process on relational data that can be applied to the web of data. We start with a set of objects and relations between objects, and extensional classes of objects. We then study how to provide a definition to classes, i.e. to build an intensional description of the class, w.r.t. the relations involving class objects. To this end, we propose three different approaches based on Formal Concept Analysis (FCA), redescription mining and Minimum Description Length (MDL). Relying on some experiments on RDF data from DBpedia, where objects correspond to resources, relations to predicates and classes to categories, we compare the capabilities and the complementarity of the three approaches. This research work is a contribution to understanding the connections existing between FCA and other data mining formalisms which are gaining importance in knowledge discovery, namely redescription mining and MDL.

Keywords: Relational data, Formal Concept Analysis, Redescription mining, DBpedia.

1 Introduction

In this work, we are interested in checking the completeness and the quality of RDF data in the linked open data (LOD), and the potential to discover definitions from these sets of linked data. Such definitions can be reused in the design of Knowledge Bases (KBs). This challenge is of main importance when we consider the masses of data which are currently published in LOD.

At an abstract level, we can view the current problem as follows. We have at hand a set of interconnected objects –objects connected by relations– just as an ABox in a description logics (DL) framework [2], and the objective is to classify the objects with respect to and in compliance with the connections they are involved in. Objects are classified in the same class, actually an extension, as soon as they share common elements. This sharing can be strict –elements are the same– or soft –elements are similar. Finally, we obtain a set of classes, possibly partially ordered, and their associated descriptions. These descriptions are important if not mandatory as they are a basis for building the definitions of classes. Definitions are considered as sets of necessary (NC) and sufficient conditions (SC) used for classifying new objects. If `x` is an instance of class `Red` then x has color red (NC), and conversely, if x has color red then x is an instance of class Red (SC).

Continuing the analogy with DLs, the idea in this paper is to build and apply induction rules having the form: $\texttt{r}(\texttt{x},\texttt{y})$ and $\texttt{y} : \texttt{C}$ then $\texttt{x} : \exists\texttt{r.C}$. This means that given a relation such as $\texttt{r}(\texttt{x},\texttt{y})$ between objects x and y, with y instance of class C, then we infer that x is an instance of a class say D whose description includes the expression $\exists\texttt{r.C}$, i.e. instances of D are related to instance(s) of C.

In this work, we aim to build definitions from RDF data. To this end, we use three approaches including Formal Concept Analysis (FCA), redescription mining and translation rule discovery. Then the main operations that we should perform are (i) the preparation of the data, (ii) the discovery of definitions, (iii) the evaluation of the quality of definitions. To compare the three algorithms, we run experiments on data extracted from DBpedia. This paper is in continuation of a line of research work on the discovery of definitions within RDF triples in the linked open data. The originality in this paper is to compare three approaches which are not based on the same principles but which can complement each other. Moreover, to the best of our knowledge, this is one of the first papers where such a study and comparison is drawn at a theoretical and practical level.

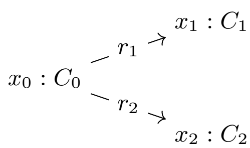
The paper is organized as follows. In the second section, we present the data on which we will be working and the basis of the classification process in the linked open data. The third section details the three classification approaches and their application. The following section is related to the experiments which have been carried out for evaluating the three approaches. Finally, a discussion, related and future work conclude the paper.

2 Data representation

In this section, we present basics of linked open data, and how we represent RDF triples as a formal context.

2.1 Linked Open Data

Linked open data (LOD) are relational data that can be seen as a set of interconnected *knowledge bases* (KB). A KB relies on two main components, a TBox which defines the *schema* of the KB and includes the concept definitions and the ABox which introduces individuals and the expressions in which individuals are involved. The basic units in a KB are RDF triples $\langle s, p, o\rangle$, which encode *subject–predicate–object* assertions. The elements of a triple can be a *resource* uniquely identified, a *literal* (values like strings, dates or integers) or a *blank node* (existential quantifier). For the sake of simplicity, in this paper we consider that $\langle s, p, o\rangle \in U \times U \times U$, where U is the set of all identified resources. Resources can refer to any object or abstraction and are identified by a URI (Uniform Resource Identifier). A URI is an address that is composed of two parts. The first part is the *namespace*, which indicates from which KB the resource comes from. The



(a) Relational data.

```
ex:x0 ex:r1 ex:x1 .
ex:x0 ex:r2 ex:x2 .
ex:x0 rdf:type ex:C0 .
ex:x1 rdf:type ex:C1 .
ex:x2 rdf:type ex:C2 .
```

(b) RDF data.

	M_C			M_D	
	C_0	C_1	C_2	$\exists r1{:}x_1$	$\exists r2{:}x_2$
x_0	×			×	×
x_1		×			
x_2			×		

(c) Formal context.

Fig. 1: Relational data, the associated set of RDF triples and the formal context built from RDF triples.

second part names the resource in this KB. The relation `rdf:type` is a specific relation of RDF which corresponds to the relation of instanciation.

LOD can be queried thanks to SPARQL queries. For example, the query `SELECT ?x WHERE {?x rdf:type ex:C0}` returns all the instances of C_0. Considering the example in Figure 1b, only $\mathtt{ex:x_0}$ is returned.

2.2 Formal Concept Analysis and RDF data

We rely on Formal Concept Analysis (FCA) from [6] in order to compare the approaches. Given G a set of objects, M a set of attributes and $I \subseteq G \times M$ a binary relation between G and M, (G, M, I) is a formal context. Derivation operators (denoted $.'$) for a set of entities $X \subseteq G$ and a set of attributes $Y \subseteq M$ are $X' = \{m \in M \mid \forall x \in X, xIm\}$ and $Y' = \{g \in G \mid \forall y \in Y, gIy\}$.

From RDF data describing a KB, we build a formal context where G is the set of subjects of the triples (i.e. $G = \{s \mid \langle s, p, o\rangle \in KB\}$) and M is the set of pairs (predicate, object) that appear in the RDF data (i.e. $M = \{(p, o) \mid \langle s, p, o\rangle \in KB\}$). The incidence relation is defined as $sI(p, o) \Leftrightarrow \langle s, p, o\rangle \in KB$.

The set of attributes is a partition of two sets: $M = M_C \cup M_D$ and $M_C \cap M_D = \emptyset$. The set M_C is the set of all attributes (p, o) such that $\mathtt{p} = \mathtt{rdf:type}$. Since all the resources in the range of `rdf:type` are classes, M_C corresponds to the set of all the classes we are trying to define. Hereafter, an attribute $(rdf : type, C)$ will simply be denoted C. The set M_D is the set of all attributes (p, o) such that $\mathtt{p} \neq \mathtt{rdf:type}$. Hereafter, an attribute $(p, o) \in M_D$ will be referred as a *description* and denoted $\exists p : o$ where o is an abbreviation of an abstract class containing only o. Considering the example Figure 1b, the associated context is presented Figure 1c.

Our goal is to build definitions of classes of the form $\mathtt{C} \equiv \mathtt{e_1} \sqcap \mathtt{e_2} \sqcap \ldots \sqcap \mathtt{e_n}$, where the $\mathtt{e_i}$ is an expression of the form $\exists\mathtt{r.x}$. To this end, we are searching for two sets of attributes $C \subseteq M_C$ and $D \subseteq M_D$ such that their derivations are the same ($C' = D'$). For example, in Figure 1, we have $\{C_0\}' = x_0$ and $\{\exists r_1{:}x_1, \exists r_2{:}x_2\}' = x_0$. Thus, the definition $C_0 \equiv \exists r_1{:}x_1 \sqcap \exists r_2{:}x_2$ can be constructed. Since data may be incomplete, it is possible that there is no equality between the derivations of a class and the derivation of its description, i.e. $\{\mathtt{C_i}\}' \neq \{\exists \mathtt{r} : \mathtt{x_j}\}'$. Therefore, we need to find some kind of approximation. This is allowed by the three algorithms presented in next section.

3 Rule mining algorithms

In this section, we briefly present the three approaches we are interested in, namely association rule mining, redescription mining and translation rule mining. Interested reader may refer to the original publications for further explanations.

3.1 Association rules

The goal of association rule mining [7] is to find dependencies between attributes. An association rule between two sets of attributes A and B, denoted $A \rightarrow B$ means that $A' \subseteq B'$. This rule has a *confidence* which can be considered as a conditional probability:

$$\text{conf}(A \rightarrow B) = \frac{|A' \cap B'|}{|A'|}$$

where $(.)'$ corresponds to the derivation operator. Confidence is used as a quality measure of the rule. An association rule is valid if its confidence is superior to a given threshold θ. When $conf(A \rightarrow B) = 1$, the rule is an implication, denoted by$A \Rightarrow B$. If $B \Rightarrow A$, then A and B form a definition, denoted by $A \equiv B$.

Since the confidence is not symmetric, $A \rightarrow B$ can be valid but $B \rightarrow A$ not valid. Potentially, an association rule $A \rightarrow B$ can be considered together with its reverse $B \rightarrow A$, and we can wonder how far they are from being implications. Accordingly, we introduce the notion of a quasi-definition which is to definition what association rule is to implication.

Definition 1 (Quasi-definition). *Given two sets of attributes A, B and a user-defined threshold θ, a quasi-definition $A \leftrightarrow B$ holds if $A \rightarrow B, B \rightarrow A$ and*

$$\min(\text{conf}(A \rightarrow B), \text{conf}(B \rightarrow A)) \geqslant \theta$$

The algorithm `Eclat` [11] is one of the existing algorithms for enumerating frequent itemsets. From frequent itemsets, association rules can be enumerated. Here, we use `Eclat` as implemented in the Coron system[1] for computing association rules. It exhaustively enumerates all the association rules that hold w.r.t. a given threshold. Here, we rely on `Eclat` to mine association rules.

Since we want to provide definitions of classes, we are interested in rules $X \rightarrow Y$ such that $X \subseteq M_C$ and $Y \subseteq M_D$ or, conversely, $X \subseteq M_D$ and $Y \subseteq M_C$. Given a rule R: $X \rightarrow Y$, the consequent can be decomposed into two rules R_C: $X \rightarrow Y_C$ and R_D: $X \rightarrow Y_D$ where $Y_C = Y \cap M_C$ and $Y_D = Y \cap M_D$ respectively. Since $Y_C \subseteq Y$, $Y' \subseteq Y'_C$, thus $|X' \cap Y'| \leqslant |X' \cap Y'_C|$, which means that if R holds, then R_C holds. Similarly, if R holds, then R_D holds.

We take advantage of this property to keep the quasi-definitions we are interested in. For example, $\exists r_1{:}x_1, C_0 \rightarrow \exists r_2{:}x_2$ is not kept because the antecedent include both categories and descriptions. On the other hand, $\exists r_1{:}x_1 \rightarrow$

[1] http://coron.loria.fr/

$\{\exists r_2{:}x_2, C_0\}$ can be decomposed into R_1: $\exists r_1{:}x_1 \rightarrow \exists r_2{:}x_2$ and R_2: $\exists r_1{:}x_1 \rightarrow C_0$. The rule R_2 is kept. If its converse is valid, we obtain the quasi-definition $C_0 \leftrightarrow \exists r_1{:}C_1$.

3.2 Redescriptions

Redescription mining [8] provides multiple characterizations of a given set of entities. Contrasting association rules, redescriptions rely on the separation of the set of attributes into *views*. The set of all views corresponds to a partition of the set of attributes. We work here with two views, corresponding to the two kinds of attributes we distinguished: M_C and M_D.

The similarity between the sets of attributes, coming from two different views, is measured thanks to the Jaccard coefficient:

$$\text{jacc}(A, B) = \frac{|A' \cap B'|}{|A' \cup B'|}$$

where $(.)'$ corresponds to the derivation operator. We say that the redescription holds if the Jaccard coefficient is above a given threshold. Contrary to confidence, the Jaccard coefficient is symmetric. A redescription with a Jaccard coefficient equal to 1 corresponds to a definition as introduced in the previous section. A redescription is necessarily a quasi-definition. Indeed,

$$\min(\text{conf}(A \rightarrow B), \text{conf}(B \rightarrow A)) \geqslant \text{jacc}(A, B).$$

Example 1. Given the context Figure 1, the two views are distinguished by the vertical line in gray. From this context, $\{C_0\} \leftrightarrow \{\exists r_1{:}x_1, \exists r_2{:}x_2\}$ is a redescription with a Jaccard coefficient of 1.

The algorithm `ReReMi` [5] is used in this work to mine redescriptions. It searches for a pair of attributes–one in each view–that may constitute a definition and tries to extend it by adding one attribute at each step. More than binary data, `ReReMi` also handles numerical and categorical data. It also enables to consider Boolean functions including conjunctions, disjunctions and negations over the attributes. Here, we only use a binary dataset and conjunctions of attributes in order to compare the results with the other algorithms.

3.3 Translation rules

The algorithm `Translator` [10] also relies on two views and searches for a set of associations between these two views, but the construction of the associations is based on a different approach, that is *minimum description length* (MDL).

The associations consist in rules that enable building one context from the other, as shown in Figure 2. The set of rules has to be compact and representative. In one hand, it should cover most of the data. In the other hand, the rules have to be as small as possible in term of attributes. To check these two constraints, `Translator` relies on MDL. Given $K = (G, M, I)$ a context and $X \subseteq M$

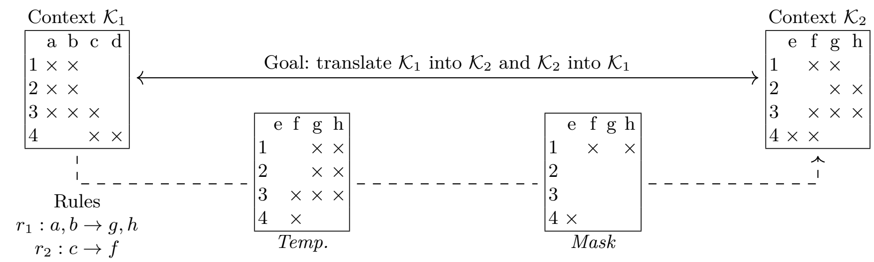


Fig. 2: `Translator` is searching for a set of rules that enables transforming $\mathcal{K}_2$ into $\mathcal{K}_1$ and $\mathcal{K}_1$ into $\mathcal{K}_2$. Here we represent only the construction of $\mathcal{K}_2$ from $\mathcal{K}_1$. For each object in $\mathcal{K}_1$ and for each rule, if the object has all the attributes of the condition, then all the attributes of the conclusion are added to $\mathcal{K}_2$.

a set of attributes, the *length* of X w.r.t. K corresponds to the minimum number of bits required in order to encode X. That is:

$$L(X) = -\sum_{x \in X} \log_2 P(x \mid K) \quad \text{where } P(x \mid K) = \frac{\mid x' \mid}{\mid G \mid}.$$

In [10], the authors compare the mining process to a translation task. A rule is considered as a *translation* from one context to an other. The underlying idea is that, with enough translation rules, one can build the first context from the other and *vice versa*. The general idea is depicted Figure 2. The errors introduced in the target context are fixed with a mask. Thus, the size of the mask corresponds to the number of errors added. The algorithm `Translator` compute rules step by step. At the beginning of the process, the mask corresponds to the target context. The algorithm searches for the rule which has the best trade-off between lowering density of the mask and not being too long, i.e. the rule which maximizes Δ:

$$\Delta(X \to Y) = \underbrace{L(Mask^-) - L(Mask^+)}_{\text{Information gain}} - \underbrace{L(X \cup Y)}_{\text{Rule length}}$$

where $Mask^+$ corresponds to the items added to the mask (errors introduced by the rule) and $Mask^-$ corresponds to items removed from the mask (errors fixed by the rule). Rules are added while $\Delta > 0$.

The mask is updated each time a rule is added. Since the information gain depends on the mask, the quality Δ of a rule depends on the rules that are already found. Thus, `Translator` is the only algorithm which takes into account rules already found to choose which rule is added.

4 Related work

In [1], authors rely on association rule mining to provide a navigation space over RDF resources. To this end, they search for implications and rank them w.r.t. the confidence of their converse. Our work is in the continuity of this one: our purpose is the same, but here we use two other approaches and compare them.

The AMIE algorithmn, extended to AMIE+ [4], is a reference for mining rules in KBs. Those rules have the form $B_1 \wedge B_2 \wedge \dots B_{n-1} \Rightarrow B_n$ where B_i is a relation between two objects $r(x_i, x_j)$. Authors add a constraint: all the variables have to appear twice in the rule, in different atoms. Our work is distinct from this one in two manners. We consider rules and their converse, and we do not focus on relations (i.e. predicates), but on the pair (predicate, object).

In a survey, Sertkaya [9] presents papers trying to bridge the gap between FCA and ontologies. In ontologies, the knowledge is constructed with top-down approaches (e.g experts who encode knowledge of a domain). At the contrary, in FCA, knowledge is discovered with a bottom-up approach, starting from facts and trying to generalize them. Thus, one way to take advantage of FCA is to allow bottom-up construction of ontologies and to complete existing ontologies. This is what is done in our approach: we start from RDF statements and try to find definitions of classes.

In [3], an extension to FCA for conceptual graphs, called G-FCA, is proposed. Compared to RDF graphs, conceptual graphs (CG) are oriented bipartite graphs. The two kinds of nodes are classes and relations. Contrasting RDF graphs which only consider binary relations, CGs handle n-ary relations. The approach enables to find *projected graph patterns*. A projected graph pattern is a pair containing a graph query and a set of candidate solutions. It is similar to a SPARQL query where the graph query is the intent and the candidate solutions are the extent. This work is complementary to our work in the sense that, instead of dealing with rule mining, it considers the full lattice.

5 Experiments

We run our experiments on *DBpedia* data, which is one of the most important knowledge bases of the linked open data, built from Wikipedia. We are interested in *categories* of *DBpedia*, that is, resources in the range of the relation `dct:subject`. Categories are a specific kind of classes. They are built from specific Wikipedia pages which lists other pages (for example the page Category:Smartphones[2]). The advantage of considering these categories instead of common classes is that there are much more categories than classes, and the only information about them provided in DBpedia is which resources belong to each category. Thus, finding why some resources are gathered together (for example, "because they all are smartphones") is an interesting challenge.

To this end, we extracted a subset of *DBpedia* thanks to a SPARQL query. The triples extracted are transformed in a context as presented in section 2.2. We run algorithms which are introduced above, then, we compare and evaluate the extracted quasi-definitions. Both data and results are available online[3].

[2] `https://en.wikipedia.org/wiki/Category:Smartphones`
[3] `https://gitlab.inria.fr/jreynaud/DefinitionMiningComparison`

Table 1: Statistics of the datasets extracted with the SPARQL query. `D` is one of the four domains, whereas the predicate `owl:objectProperty` ensures that $?o$ is a resource, and not a literal nor a blank node.

```
SELECT DISTINCT * WHERE {
 ?s ?p ?o .
 ?s dct:subject dbc:C .
 ?p a owl:ObjectProperty .
}
```

D	Triples	Objects	$\|M_C\|$	$\|M_D\|$
Turing_Award	2 642	65	503	857
Smartphones	8 418	598	359	1 730
Sports_cars	9 047	604	435	2 295
French_films	121 496	6 039	6 028	19 459

5.1 Methodology

We extracted four different subsets of triples, of different size and different domains, from *DBpedia*, with SPARQL queries. All the queries follow the same pattern. The datasets correspond to the categories `Smartphones`, `Sports_cars`, `Turing_Award_laureates` and `French_films`. Statistics of the datasets are provided Table 1.

For each dataset, the partition of the attributes is constructed as follows: M_C is the subset of attributes whose predicate is `dct:subject` whereas M_D is the set of attributes whose predicate differs from `dct:subject`. For `Eclat`, since both attributes of classes and descriptions are in the same context, the input data is one file which contains the context in a tabular format. For `ReReMi` and `Translator`, the input data are two tabular files.

5.2 Results

Each algorithm returns an ordered set of quasi-definitions. Each quasi-definition is manually evaluated by three phD students familiar with linked open data, playing the role of experts. Given a definition $C_0, \ldots, C_n \leftrightarrow D_0, \ldots, D_m$ from a dataset X, each evaluator answers the question " Taking X as a reference, is it true that *belonging to* C_0 *and* C_1 *. . . and* C_n and *having the properties* D_0 *and* D_1 *. . . and* D_m is equivalent ? " The final evaluation is the majority between the experts. Experts gave the same answer in 95.4% of the cases. If evaluated true, the quasi-definition is added to the set of definitions (see Fig. 3).

The comparison between the algorithms is based on definitions (i.e. quasi-definition evaluated as true by at least 2 experts) that have been extracted and categories that have been defined. Figure 4 shows two Venn diagrams for each dataset: one for the number of definitions extracted and one for the number of categories defined. In the dataset `Turing_Award_laureates`, for example, there are 22 definitions only extracted by `Eclat` and 8 definitions extracted by both `Eclat` and `Translator`. `Eclat` extracted 30 definitions in total. A category is considered as defined as soon as it appears in a definition. Therefore, in one definition, there can be one or more categories considered as defined.

	`Turing_Award_laureates`	
R	**Harvard_University_alumni** ↔ (almaMater Harvard_University)	R1
ET	**Harvard_University_alumni, Turing_Award_laureates** ↔ (a Agent), (a Person), (a Scientist), (almaMater Harvard_University)	R2
E	**Turing_Award_laureates** ↔ (a Agent), (a Person), (award Turing_Award)	R3
ET	**Turing_Award_laureates** ↔ (a Agent), (a Person), (a Scientist), (award Turing_Award)	R4
E	**Modern_cryptographers** ↔ (field Cryptography)	R5
	`Sports_cars`	
R	**McLaren_vehicles** ↔ (manufacturer McLaren_Automotive)	R6
R	**McLaren_vehicles** ↔ (assembly Surrey)	R7
ET	**McLaren_vehicles, Sports_cars** ↔ (a Automobile), (a MeanOfTransportation), (assembly Woking), (assembly Surrey), (assembly England), (bodyStyle Coupé), (manufacturer McLaren_Automotive)	R8
E	**McLaren_vehicles, Sports_cars** ↔ (a Automobile), (a MeanOfTransportation), (assembly England), (assembly Surrey), (bodyStyle Coupé)	R9
E	**McLaren_vehicles, Sports_cars** ↔ (a Automobile), (a MeanOfTransportation), (assembly Surrey), (bodyStyle Coupé)	R10
	`Smartphones`	
ET	**Firefox_OS_devices, Open-source_mobile_phones, Smartphones, Touchscreen_mobile_phones** ↔ (a Device), (operatingSystem Firefox_OS)	R11
R	**Nokia_mobile_phones** ↔ (manufacturer Nokia)	R12
ET	**Nokia_mobile_phones, Smartphones** ↔ (a Device), (manufacturer Nokia)	R13
R	**Samsung_Galaxy** ↔ (manufacturer Samsung_Electronics), (operatingSystem Android_(operating_system))	R14
ET	**Samsung_Galaxy, Samsung_mobile_phones, Smartphones** ↔ (a Device), (manufacturer Samsung_Electronics), (operatingSystem Android_(operating_system))	R15
	`French_films`	
R	**Pathé_films** ↔ (distributor Pathé)	R16
R	**Films_directed_by_Georges_Méliès** ↔ (director Georges_Méliès)	R17
ET	**Films_directed_by_Georges_Méliès, French_films, French_silent_short_films** ↔ (a Film), (a Wikidata:Q11424), (a Work), (director Georges_Méliès)	R18
ET	**Films_directed_by_Jean_Rollin, French_films** ↔ (a Film), (a Wikidata:Q11424), (a Work), (director Jean_Rollin)	R19
ET	**Film_scores_by_Gabriel_Yared, French_films** ↔ (a Film), (a Wikidata:Q11424), (a Work), (musicComposer Gabriel_Yared)	R20

Fig. 3: Definitions extracted by `Eclat`, `ReReMi` and `Translator` for each dataset. In order to be more readable, namespaces have been removed.

6 Discussion

Hereafter, we will denote $\mathcal{B}^X_{cand}$ the set of all the quasi-definitions extracted by the algorithm X and $\mathcal{B}^X_{def}$ the set of quasi-definitions from $\mathcal{B}^X_{cand}$ evaluated true by the experts, i.e. the set of definitions extracted by X. The set $\mathcal{B}_{cand}$ denotes the set of all the quasi-definitions definitions extracted, regardless the algorithm. Similarly, $\mathcal{B}_{def}$ denotes the set of all the definitions extracted.

6.1 Precision, recall and completeness

The precision of an algorithm X is $\frac{|\mathcal{B}^X_{def}|}{|\mathcal{B}^X_{cand}|}$. The precision of `ReReMi` has a high variability (from 33% to 75%) and is overall the weakest, especially for the dataset `French_films`. The precision of `Eclat` is stable (from 64% to 72%). `Translator` has the best precision which is always over 74%.

Table 2: Evaluation of the results. For each dataset, the number of quasi-definitions extracted ($|\mathcal{B}_{cand}|$) and evaluated true ($|\mathcal{B}_{def}|$) are reported, along with the average number of categories ($|\overline{C_i}|$) and descriptions ($|\overline{D_i}|$) per rule.

(a) `Turing_Award_laureates`

X	`Eclat`	`ReReMi`	`Translator`
$\|\mathcal{B}_{cand}\|$	47	12	11
$\|\mathcal{B}_{def}\|$	30	9	9
$\|\mathcal{B}^X_{def}\|/\|\mathcal{B}^X_{cand}\|$	.64	.75	.85
$\|\overline{C_i}\|$–$\|\overline{D_i}\|$	2–4	1–1	3–5

(b) `French_films`

X	`Eclat`	`ReReMi`	`Translator`
$\|\mathcal{B}_{cand}\|$	132	52	31
$\|\mathcal{B}_{def}\|$	95	30	23
$\|\mathcal{B}^X_{def}\|/\|\mathcal{B}^X_{cand}\|$	.72	.68	.74
$\|\overline{C_i}\|$–$\|\overline{D_i}\|$	2.8–4.5	1.3–1.4	2.6–4.1

(c) `Sports_cars`

X	`Eclat`	`ReReMi`	`Translator`
$\|\mathcal{B}_{cand}\|$	810	98	41
$\|\mathcal{B}_{def}\|$	521	57	31
$\|\mathcal{B}^X_{def}\|/\|\mathcal{B}^X_{cand}\|$	.64	.58	.76
$\|\overline{C_i}\|$–$\|\overline{D_i}\|$	4.3–7.8	1.6–1.8	3.1–3.1

(d) `Smartphones`

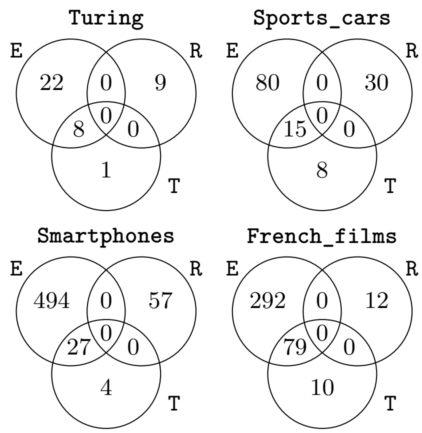
X	`Eclat`	`ReReMi`	`Translator`
$\|\mathcal{B}_{cand}\|$	546	36	93
$\|\mathcal{B}_{def}\|$	371	12	89
$\|\mathcal{B}^X_{def}\|/\|\mathcal{B}^X_{cand}\|$	.68	.33	.96
$\|\overline{C_i}\|$–$\|\overline{D_i}\|$	2.8–4.4	1.2–1.1	2.3–4.2

The recall could be defined as $\frac{|\mathcal{B}^X_{def}|}{|\mathcal{B}_{def}|}$. However, it cannot be used as a performance measure. Indeed, some of the definitions overlap (i.e. have attributes in common in both sides). This is the case for the rules R6 to R10 in Figure 3 : all the rules define the category `McLaren_vehicules`. Whereas `Translator` extracts only one rule (R8), `ReReMi` extracts 2 rules (R6 and R7) and `Eclat` extracts 9 rules (only 3 of them, R8 to R10, are reported here).

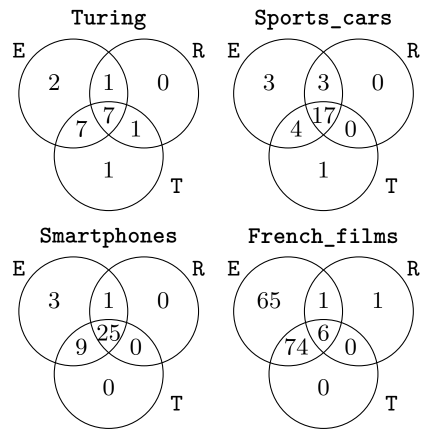
Given the valid quasi-definitions, the uncompleteness of the KB can be measured as the number of triples which can be inferred from the quasi-definitions and that are not already in the KB. For example, given the rule `Pathé_Films` $\leftrightarrow$ `(distributor Pathé)`, if a resource `r` belongs to `Pathé_Films` (i.e. ⟨`r, subject, Pathé_Films`⟩ $\in KB$), then the triple ⟨`r, distributor, Pathé`⟩ is expected to be in the KB. Conversely, if the triple ⟨`r, distributor, Pathé`⟩ belongs to the KB, then ⟨`r, subject, Pathé_Films`⟩ is expected to be in the KB. Figure 4c counts, for each dataset, the number of inferred triples that were not in the KB.

6.2 Shape and interpretation of the rules

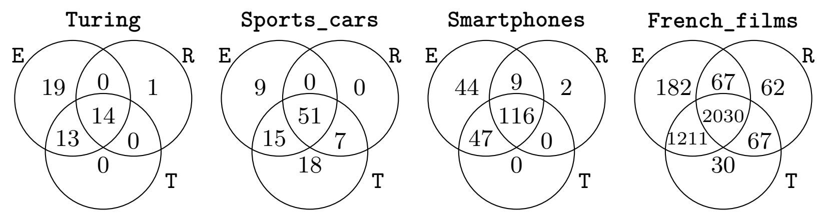
From Figure 4, 70% of the categories defined by `Eclat` or `Translator` are defined by both algorithms. However, `Translator` extracts much less rules than `Eclat` (until 16 times less for the dataset `Smartphones`). This is due to the extraction process of association rules: if the rule $A \rightarrow B$ has the same support as the rule $A \rightarrow \{B, C\}$, then only the rule $A \rightarrow \{B, C\}$ is kept. However, if the support of $A \rightarrow B$ is higher, both rules are kept. Consequently, `Eclat` mines rule which can differ from only one attribute (R9 and R10), contrary to `Translator` (only R8).



(a) Definitions extracted.



(b) Categories defined.



(c) Triples inferred.

Fig. 4: Definitions extracted, categories defined, and triples inferred by `Eclat`(`E`), `ReReMi`(`R`) and `Translator`(`T`) for each dataset.

None of the definitions mined by `ReReMi` are shared by `Eclat` or `Translator`. This is due to the heuristic used by `ReReMi`. If C is a category and D_1 and D_2 are two descriptions such that $C' = D_1' = D_2'$, then `ReReMi` generates the two definitions $C \leftrightarrow D_1$ and $C \leftrightarrow D_2$ rather than one definition $C \leftrightarrow \{D_1, D_2\}$ as `Eclat` does. This is the case for definitions R6 and R7 mined by `ReReMi`, and R8 mined by `Eclat`. If $C' = D_1'$ and $D_1' \subset D_2'$, `ReReMi` generates the definition $C \leftrightarrow D_1$ whereas `Eclat` generates $C \leftrightarrow \{D_1, D_2\}$, as shown with definitions R12 and R13 for example. Another consequence of the heuristic used by `ReReMi` is that it mines smaller definitions than definitions mined by `Eclat` and `Translator` wrt the number of attributes. On average, definitions mined by `ReReMi` have 1 or 2 attributes on each side whereas definitions mined by `Eclat` and `Translator` have 3 categories and 4 descriptions. These differences raise the question of the semantics of the conjunctions. Indeed, the semantics of the conjunctions in the definitions mined by `ReReMi` differs from the one in definition mined by `Eclat` and `Translator`. For example, in rule R15, the attribute `(a, Device)` can be removed without repercussion on the meaning. On the opposite side, in definition R14, no attribute can be removed without changing the meaning of the definition. That is, all the attributes are necessary. In our approach, it seems more interesting to consider only attributes that are necessary in the definition. Thus, R14 is better than R15 according to the ease of interpretation.

7 Conclusion

In this paper, we compared three algorithms to find definitions in the linked open data. Each algorithm has its specificities and we verified that these specificities are reflected in the results of our experiments. We showed that, despite their very different approaches, `Eclat` and `Translator` extract a lot of identical rules. At the opposite, `ReReMi`, in spite of a quality measure very similar to `Eclat`, extracts shorter rules. The advantage of each algorithm depends on the goal of the user. In our experiments, `Eclat` is the algorithm which defines the most of the categories, at the cost of a huge number of quasi-definitions extracted. `Translator` extracts significantly less quasi-definitions but defines less categories. `ReReMi`, despite a low number of categories defined, offers definitions easier to understand which do not include attributes that do not contribute to the definition.

Acknowledgements

This work has been conducted with the support of "Région Lorraine" and "Délégation Générale de l'Armement".

References

1. M. Alam, A. Buzmakov, V. Codocedo, and A. Napoli. Mining definitions from RDF annotations using formal concept analysis. In *IJCAI*, pages 823–829, 2015.
2. F. Baader, D. Calvanese, D. McGuinness, D. Nardi, and P. Patel-Schneider, editors. *The Description Logic Handbook*. Cambridge University Press, 2003.
3. S. Ferré and P. Cellier. Graph-FCA in Practice. In *Proceedings of 22nd ICCS*, pages 107–121, 2016.
4. L. A. Galárraga, C. Teflioudi, K. Hose, and F. M. Suchanek. Fast rule mining in ontological knowledge bases with AMIE+. *VLDB Journal*, 24(6):707–730, 2015.
5. E. Galbrun and P. Miettinen. From Black and White to Full Color: Extending Redescription Mining Outside the Boolean World. *Statistical Analysis and Data Mining*, 5(4):284–303, 2012.
6. B. Ganter and R. Wille. *Formal concept analysis - mathematical foundations*. Springer, 1999.
7. J. Han, J. Pei, and M. Kamber. *Data mining: concepts and techniques*. Elsevier, 2011.
8. N. Ramakrishnan, D. Kumar, B. Mishra, M. Potts, and R. F. Helm. Turning CARTwheels: an Alternating Algorithm for Mining Redescriptions. In *KDD'04*, pages 266–275, 2004.
9. B. Sertkaya. A survey on how description logic ontologies benefit from formal concept analysis. *CoRR*, abs/1107.2822, 2011.
10. M. van Leeuwen and E. Galbrun. Association Discovery in Two-View Data. *TKDE*, 27(12):3190–3202, Dec. 2015.
11. M. J. Zaki. Scalable algorithms for association mining. *TKDE*, 12(3):372–390, 2000.

Relational proportions between objects and attributes

Nelly Barbot[1], Laurent Miclet[1], and Henri Prade[2]

[1] Univ. Rennes, CNRS, IRISA, F-22305 Lannion, France,
nbarbot@irisa.fr, laurent.miclet@gmail.com,
[2] IRIT, CNRS & Univ. P. Sabatier, 31062 Toulouse Cedex 9, France,
prade@irit.fr

Abstract. Analogical proportions are statements of the form "A is to B as C is to D", where A, B, C, D are items of the same nature, or not. In this paper, we more particularly consider "relational proportions" of the form "object A has the same relationship with attribute a as object B with attribute b". We provide a formal definition for relational proportions, and investigate how they can be extracted from a formal context, in the setting of formal concept analysis.

Keywords: Analogy, analogical reasoning, analogical proportion, analogy in lattices, formal concept analysis.

1 Introduction

A statement such as "Carlsen is to chess as Mozart is to music" introduces Carlsen as a precocious virtuoso of chess, a quality that Mozart is well known to have concerning music. It relates two types of items, here people and activities. It is an example of what we call *relational proportions* which are statements of the form "object A has the same relationship with attribute a as object B with attribute b". This can be viewed as a special case of analogical proportions which are statements of the form "A is to B as C is to D". In the case where A, B, C, D are items which can be represented in terms of the same set of features, a formal definition has been proposed for analogical proportions in the setting of Boolean logic and then extended using multiple-valued logic for handling numerical features [2, 9], by stating that "A differs from B as C differs from D and B differs from A as D differs from C".

The nature of relational proportions suggests to handle them in the setting of formal concept analysis. This leads us to the question of defining analogical proportions between formal concepts. The paper first recalls the definition of analogical proportions in non distributive lattices, as already presented in [4]. Then it brings original material, firstly by studying the links between analogical proportions between formal concepts and analogical proportions between objects or attributes. It also shows how relational proportions can be obtained in a formal context from the identification of an analogical complex.

2 Analogical proportions: basics and formalization

Analogical proportions are usually characterized by three axioms. The first two axioms acknowledge the symmetrical role played by the pairs (x, y) and (z, t) in the proportion 'x is to y as z is to t', and enforce the idea that y and z can be interchanged if the proportion is valid, just as in the equality of two numerical ratios where means can be exchanged. This view dates back to Aristotle. A third (optional) axiom, called determinism, insists on the uniqueness of the solution $t = y$ for completing the analogical proportion in t: $(x : y :: x : t)$. These axioms are studied in [1].

Definition 1 (Analogical proportion). *An analogical proportion (AP) on a set X is a quaternary relation on X, i.e. a subset of X^4. An element of this subset, written $(x : y :: z : t)$, which reads 'x is to y as z is to t', must obey the following axioms:*

1. Reflexivity of 'as': $(x : y :: x : y)$
2. Symmetry of 'as': $(x : y :: z : t) \Leftrightarrow (z : t :: x : y)$
3. Exchange of means: $(x : y :: z : t) \Leftrightarrow (x : z :: y : t)$

Then, thanks to symmetry, it can be easily seen that $(x : y :: z : t) \Leftrightarrow (t : y :: z : x)$ should also hold (exchange of the extremes). According to the first two axioms, four other formulations are equivalent to the canonical form $(x : y :: z : t)$. Finally, the eight equivalent forms of an analogical proportion are: $(x : y :: z : t)$, $(z : t :: x : y)$, $(y : x :: t : z)$, $(t : z :: y : x)$, $(z : x :: t : y)$, $(t : y :: z : x)$, $(x : z :: y : t)$ and $(y : t :: x : z)$.

With respect to this axiomatic definition of AP, Stroppa and Yvon [3] have given another definition, based on the notion of factorization when the set of objects is a commutative semigroups. From these previous works, Miclet *et al.* [4] have derived the following definitions in the lattice framework.

Definition 2. *A 4-tuple (x, y, z, t) of a lattice $(L, \vee, \wedge, \leq)^4$ is a Factorial Analogical Proportion (FAP) $(x : y :: z : t)$ iff:*

$$\begin{array}{ll} x = (x \wedge y) \vee (x \wedge z) & x = (x \vee y) \wedge (x \vee z) \\ y = (x \wedge y) \vee (y \wedge t) & y = (x \vee y) \wedge (y \vee t) \\ z = (z \wedge t) \vee (x \wedge z) & z = (z \vee t) \wedge (x \vee z) \\ t = (z \wedge t) \vee (y \wedge t) & t = (z \vee t) \wedge (y \vee t) \end{array}$$

Definition 3. *A 4-tuple (x, y, z, t) of $(L, \vee, \wedge, \leq)^4$ is a Weak Analogical Proportion (WAP) when $x \wedge t = y \wedge z$ and $x \vee t = y \vee z$. It is denoted $x : y \, WAP \, z : t$.*

In the case of a distributive lattice (e.g. a Boolean lattice), this alternative definition is equivalent to the FAP. But, in general, a FAP is a WAP and the converse is false, which explains the use of adjective "weak" [4].

Example 1. Let us consider a finite set Σ and the associated Boolean lattice $(2^{\Sigma}, \cup, \cap, \leq)$. When saying of subsets x, y, z, t of Σ that "x is to y as z is to t", we express that x differs from y in the same way as z differs from t. For example, if $x = \{a, b, e\}$ and $y = \{b, c, e\}$, we see that to transform x into y, we have to remove a and add c. Now, if $z = \{a, d, e\}$, we can construct t with the same operations, to obtain $t = \{c, d, e\}$. In more formal terms, with this definition, the following properties are asked to x, y, z and t (with $x \setminus y = x \cap \neg y$): $x \setminus y = z \setminus t$ and $y \setminus x = t \setminus z$, which are equivalent to $x \cap t = y \cap z$ and $x \cup t = y \cup z$. These relations linking x, y, z, t are clearly symmetrical, and satisfy the exchange of the means. Hence it is a correct definition of the AP in the Boolean setting [2].

The next proposition gives a simple example of FAP in a lattice.

Proposition 1. *Let y and z be two elements of a lattice, the proportion $y : y \vee z :: y \wedge z : z$ is a FAP. We call it a Canonical Analogical Proportion (CAP).*

Proof. The first equality of Definition 2, namely $y = (y \wedge (y \vee z)) \vee (y \wedge (y \wedge z))$, is true since the right member is equal to $(y) \vee (y \wedge z) = y$. The verification of the three other equalities of Definition 2 is similar, using the absorption laws.

3 Analogical proportions in FCA

In order to derive more specifically the AP notion in a Formal Concept Analysis framework (FCA), we first recall some basic elements of FCA, before studying the relations between several kinds of AP and their characterization in FCA.

3.1 Formal concept analysis

FCA starts with a binary relation R defined between a set $\mathcal{O}$ of objects and a set $\mathcal{A}$ of attributes. The tuple $(\mathcal{O}, \mathcal{A}, R)$ is called a *formal context*. The notation $(o, a) \in R$ or oRa means that object o has attribute a. We denote $o^{\uparrow} = \{a \in \mathcal{A} | (o, a) \in R\}$ the attribute set of object o and $a^{\downarrow} = \{o \in \mathcal{O} | (o, a) \in R\}$ the object set having attribute a. Similarly, for any subset $\mathbf{o}$ of objects, $\mathbf{o}^{\uparrow}$ is defined as $\{a \in \mathcal{A} | a^{\downarrow} \supseteq \mathbf{o}\}$. Then a *formal concept* is defined as a pair $(\mathbf{o}, \mathbf{a})$, such that $\mathbf{a}^{\downarrow} = \mathbf{o}$ and $\mathbf{o}^{\uparrow} = \mathbf{a}$. One calls $\mathbf{o}$ the *extension* of the concept and $\mathbf{a}$ its *intension*.

The set of all formal concepts is equipped with a partial order (denoted $\leq$) defined as: $(\mathbf{o}_1, \mathbf{a}_1) \leq (\mathbf{o}_2, \mathbf{a}_2)$ iff $\mathbf{o}_1 \subseteq \mathbf{o}_2$ (or, equivalently, $\mathbf{a}_2 \subseteq \mathbf{a}_1$). Then it is structured as a lattice, called the *concept lattice* of R.

Example 2. The concept lattice of the following context R is shown in Figure 1.

	a_1	a_2	a_3	a_4	a_5	a_6	a_7	a_8	a_9
o_1		×	×	×			×	×	×
o_2	×		×	×	×		×		×
o_3	×	×		×		×		×	×
o_4	×	×	×		×	×			×
o_5	×	×	×	×	×	×	×	×	×

The following preliminaries are simple consequences of the definition of concept lattice and the Main Theorem of Formal Concepts [5, 6]. They allow for a quick demonstration of propositions in the next section.

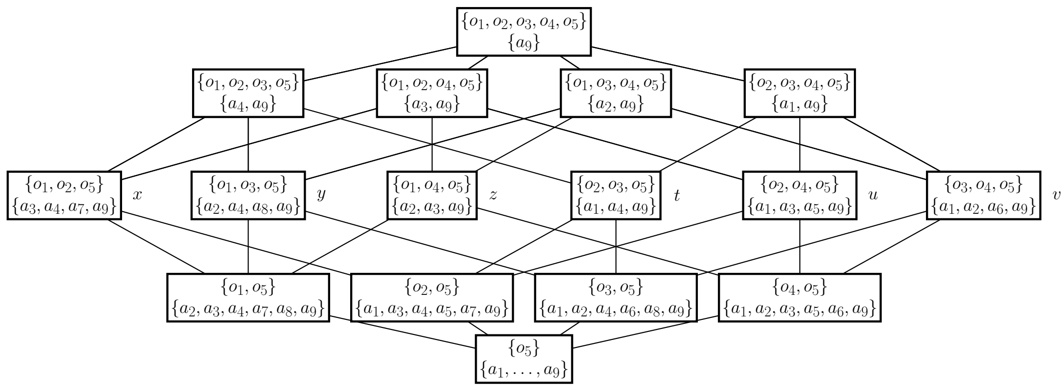


Fig. 1. The formal concept lattice of R (it is a Boolean lattice).

Preliminary 1 *Given two concepts* $x = (\mathbf{o}_x, \mathbf{a}_x)$ *and* $y = (\mathbf{o}_y, \mathbf{a}_y)$*, one has* $(\mathbf{o}_x \cup \mathbf{o}_y)^{\uparrow} = \mathbf{a}_x \cap \mathbf{a}_y$ *and* $(\mathbf{a}_x \cup \mathbf{a}_y)^{\downarrow} = \mathbf{o}_x \cap \mathbf{o}_y$*.*

Preliminary 2 *Given two concepts* $x = (\mathbf{o}_x, \mathbf{a}_x)$ *and* $y = (\mathbf{o}_y, \mathbf{a}_y)$*, one has* $\mathbf{o}_x \cup \mathbf{o}_y \subseteq \mathbf{o}_{x \vee y}$*,* $\mathbf{o}_x \cap \mathbf{o}_y = \mathbf{o}_{x \wedge y}$*,* $\mathbf{a}_x \cup \mathbf{a}_y \subseteq \mathbf{a}_{x \wedge y}$ *and* $\mathbf{a}_x \cap \mathbf{a}_y = \mathbf{a}_{x \vee y}$*.*

Preliminary 3 *Let* $\mathbf{o}$ *(resp.* $\mathbf{a}$*) be a subset of* $\mathcal{O}$ *(resp.* $\mathcal{A}$*), there exists at most one concept* x *such that* $\mathbf{o}_x = \mathbf{o}$ *(resp.* $\mathbf{a}_x = \mathbf{a}$*).*

3.2 Weak and strong analogical proportions in FCA

Since concepts are associated to a set of attributes and objects, the main objectives of this section are to relate the AP definitions with these sets and to study the links the AP on concept lattice and AP on object or attribute sets.

Proposition 2. *Let* x*,* y*,* z *and* t *be four concepts, one has:*
($x \vee t = y \vee z$ *iff* $\mathbf{a}_x \cap \mathbf{a}_t = \mathbf{a}_y \cap \mathbf{a}_z$*) and (*$x \wedge t = y \wedge z$ *iff* $\mathbf{o}_x \cap \mathbf{o}_t = \mathbf{o}_y \cap \mathbf{o}_z$*).*
As consequence, ($x : y\ WAP\ z : t$*) iff* $\mathbf{a}_x \cap \mathbf{a}_t = \mathbf{a}_y \cap \mathbf{a}_z$ *and* $\mathbf{o}_x \cap \mathbf{o}_t = \mathbf{o}_y \cap \mathbf{o}_z$*.*

Proof. From Preliminary 2, $x \vee t = y \vee z$ implies $\mathbf{a}_x \cap \mathbf{a}_t = \mathbf{a}_y \cap \mathbf{a}_z$ and conversely, $\mathbf{a}_x \cap \mathbf{a}_t = \mathbf{a}_y \cap \mathbf{a}_z$ implies $\mathbf{a}_{x \vee t} = \mathbf{a}_{y \vee z}$. Thus, $x \vee t = y \vee z$ using Preliminary 3. The proof of the second equivalence can be done in a similar manner.

Proposition 3. *Let* x*,* y*,* z *and* t *be four concepts, if (*$\mathbf{a}_x : \mathbf{a}_y :: \mathbf{a}_z : \mathbf{a}_t$ *or* $\mathbf{o}_x : \mathbf{o}_y :: \mathbf{o}_z : \mathbf{o}_t$*) then* $x : y\ WAP\ z : t$*.*

Proof. Let x, y, z and t be four concepts such that $\mathbf{a}_x : \mathbf{a}_y :: \mathbf{a}_z : \mathbf{a}_t$, or equivalently $\mathbf{a}_x \cap \mathbf{a}_t = \mathbf{a}_y \cap \mathbf{a}_z$ and $\mathbf{a}_x \cup \mathbf{a}_t = \mathbf{a}_y \cup \mathbf{a}_z$ (the APs between the subsets of attributes correspond to FAPs in the Boolean lattice of $(2^{\mathcal{A}}, \cup, \cap, \subseteq)$). Thanks to Proposition 2, $\mathbf{a}_x \cap \mathbf{a}_t = \mathbf{a}_y \cap \mathbf{a}_z$ is equivalent to $x \vee t = y \vee z$. Moreover, using Preliminary 1, we have $(\mathbf{a}_x \cup \mathbf{a}_t)^{\downarrow} = \mathbf{o}_x \cap \mathbf{o}_t$ and $(\mathbf{a}_y \cup \mathbf{a}_z)^{\downarrow} = \mathbf{o}_y \cap \mathbf{o}_z$. Thus, $\mathbf{a}_x \cup \mathbf{a}_t = \mathbf{a}_y \cup \mathbf{a}_z$ implies $\mathbf{o}_x \cap \mathbf{o}_t = \mathbf{o}_y \cap \mathbf{o}_z$. In the case where $\mathbf{o}_x : \mathbf{o}_y :: \mathbf{o}_z : \mathbf{o}_t$, the proof is similar since we also have $(\mathbf{o}_x \cup \mathbf{o}_t)^{\uparrow} = \mathbf{a}_x \cap \mathbf{a}_t$ and $(\mathbf{o}_y \cup \mathbf{o}_y)^{\uparrow} = \mathbf{a}_y \cap \mathbf{a}_z$.

Comments. The converse is false. Let us consider the following formal context

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

its concept lattice is displayed on Figure 2. Concepts $x = (\{o_1\}, \{a_3, a_4\})$, $y = (\{o_2\}, \{a_1, a_3\})$, $z = (\{o_3\}, \{a_2, a_4\})$ and $t = (\{o_4\}, \{a_1, a_2, a_5\})$ are in WAP, due to Proposition 2. However, the Boolean APs $\mathbf{a}_x : \mathbf{a}_y :: \mathbf{a}_z : \mathbf{a}_t$ and $\mathbf{o}_x : \mathbf{o}_y :: \mathbf{o}_z : \mathbf{o}_t$ are both false. The WAP between concepts is less restrictive than the AP between sets of attributes: in a WAP, objects are allowed to possess attributes which are not shared by any other object concerned in the WAP.

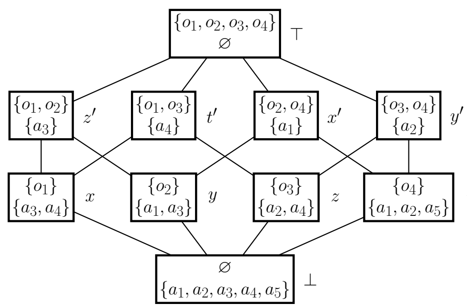


Fig. 2. In this lattice, x, y, z and t are in WAP but $\mathbf{a}_x : \mathbf{a}_y :: \mathbf{a}_z : \mathbf{a}_t$ and $\mathbf{o}_x : \mathbf{o}_y :: \mathbf{o}_z : \mathbf{o}_t$ are both false. Besides, x', y', z' and t' are in WAP and $\mathbf{o}_{x'} : \mathbf{o}_{y'} :: \mathbf{o}_{z'} : \mathbf{o}_{t'}$ is true, but $\mathbf{a}_{x'} : \mathbf{a}_{y'} :: \mathbf{a}_{z'} : \mathbf{a}_{t'}$ and the FAP $x' : y' :: z' : t'$ are both false.

We give now a proposition which leads us to a corollary in which is defined yet another analogical proportion between formal concepts, the strongest of all.

Proposition 4. *Let x, y, z and t be four concepts, if ($\mathbf{a}_x \cup \mathbf{a}_t = \mathbf{a}_y \cup \mathbf{a}_z$ and $\mathbf{o}_x \cup \mathbf{o}_t = \mathbf{o}_y \cup \mathbf{o}_z$) then the FAP $x : y :: z : t$ holds.*

Proof. $\mathbf{a}_x \cup \mathbf{a}_t = \mathbf{a}_y \cup \mathbf{a}_z$ implies that $\mathbf{a}_x = (\mathbf{a}_x \cap \mathbf{a}_y) \cup (\mathbf{a}_x \cap \mathbf{a}_z)$. It results that, using Preliminaries 1 and 2, $\mathbf{o}_x = (\mathbf{a}_x)^{\downarrow} = (\mathbf{a}_x \cap \mathbf{a}_y)^{\downarrow} \cap (\mathbf{a}_x \cap \mathbf{a}_z)^{\downarrow}$. Then,

$$\mathbf{o}_x = (\mathbf{a}_{x \vee y})^{\downarrow} \cap (\mathbf{a}_{x \vee z})^{\downarrow} = \mathbf{o}_{x \vee y} \cap \mathbf{o}_{x \vee z} = \mathbf{o}_{(x \vee y) \wedge (x \vee z)}$$

and Preliminary 3 permits to obtain $x = (x \vee y) \wedge (x \vee z)$.

In a same way, from $\mathbf{o}_x \cup \mathbf{o}_t = \mathbf{o}_y \cup \mathbf{o}_z$, we get that $\mathbf{o}_x = (\mathbf{o}_x \cap \mathbf{o}_y) \cup (\mathbf{o}_x \cap \mathbf{o}_z)$ and $\mathbf{a}_x = (\mathbf{o}_x)^{\uparrow} = (\mathbf{o}_x \cap \mathbf{o}_y)^{\uparrow} \cap (\mathbf{o}_x \cap \mathbf{o}_z)^{\uparrow}$. Then,

$$\mathbf{a}_x = (\mathbf{o}_{x \wedge y})^{\uparrow} \cap (\mathbf{o}_{x \wedge z})^{\uparrow} = \mathbf{a}_{x \wedge y} \cap \mathbf{a}_{x \wedge z} = \mathbf{a}_{(x \wedge y) \vee (x \wedge z)}.$$

Thus, $x = (x \wedge y) \vee (x \wedge z)$. All the equalities of Definition 2 are similarly checked.

Corollary 1. *Let x, y, z and t be four concepts, the following two conjunctions are equivalent:*

$$\mathbf{a}_x \cup \mathbf{a}_t = \mathbf{a}_y \cup \mathbf{a}_z \textit{ and } \mathbf{o}_x \cup \mathbf{o}_t = \mathbf{o}_y \cup \mathbf{o}_z$$
$$\mathbf{a}_x : \mathbf{a}_y :: \mathbf{a}_z : \mathbf{a}_t \textit{ and } \mathbf{o}_x : \mathbf{o}_y :: \mathbf{o}_z : \mathbf{o}_t$$

This characterizes a particular case of FAP between concepts that we call a Strong Analogical Proportion (SAP). It is denoted $x : y \ SAP \ z : t$. In other words, four concepts in analogical proportion on attributes and on objects are said to be in strong analogical proportion.

Proof. Let x, y, z and t be such that $\mathbf{a}_x \cup \mathbf{a}_t = \mathbf{a}_y \cup \mathbf{a}_z$ and $\mathbf{o}_x \cup \mathbf{o}_t = \mathbf{o}_y \cup \mathbf{o}_z$, Proposition 4 implies the FAP $x : y :: z : t$, and then $x : y \ WAP \ z : t$. Hence, using Proposition 2, we have $\mathbf{a}_x \cap \mathbf{a}_t = \mathbf{a}_y \cap \mathbf{a}_z$ and $\mathbf{o}_x \cap \mathbf{o}_t = \mathbf{o}_y \cap \mathbf{o}_z$. Consequently, $\mathbf{a}_x : \mathbf{a}_y :: \mathbf{a}_z : \mathbf{a}_t$ and $\mathbf{o}_x : \mathbf{o}_y :: \mathbf{o}_z : \mathbf{o}_t$. The converse is trivial.

Comments. From Corollary 1, $\mathbf{a}_x : \mathbf{a}_y :: \mathbf{a}_z : \mathbf{a}_t$ and $\mathbf{o}_x : \mathbf{o}_y :: \mathbf{o}_z : \mathbf{o}_t$ imply the FAP $x : y :: z : t$. However, the reciprocal is false. Let us consider the concept lattice displayed in Figure 2: we have the FAP $y : \top :: \bot : z$ (which is a CAP) but $\mathbf{o}_y \cup \mathbf{o}_z \neq \mathbf{o}_\top \cup \mathbf{o}_\bot$ and $\mathbf{a}_y \cup \mathbf{a}_z \neq \mathbf{a}_\top \cup \mathbf{a}_\bot$.

Example 3. In the Boolean lattice displayed in Figure 1, concepts x, y, u and v form a FAP but are not in SAP. Indeed, $\mathbf{a}_x : \mathbf{a}_y :: \mathbf{a}_u : \mathbf{a}_v$ does not hold. However, without changing the lattice, the formal context can be reduced to

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

and the reduced representation of x, y, u and v gives ($x : y \ SAP \ u : v$):

	a_1	a_2	a_3	a_4
$\mathbf{a}_x$			×	×
$\mathbf{a}_y$		×		×
$\mathbf{a}_u$	×		×	
$\mathbf{a}_v$	×	×		

	o_1	o_2	o_3	o_4
$\mathbf{o}_x$	×	×		
$\mathbf{o}_y$	×		×	
$\mathbf{o}_u$		×		×
$\mathbf{o}_v$			×	×

These observations stem from the fact that the FAP and WAP between concepts are directly related to the lattice whereas the Boolean AP between object or attribute sets directly depends on the formal context.

4 Formal concepts and relational proportion

4.1 From a RP to concepts in AP

In this section, we study if we can deduce from a relational proportion "A is the B of a", or "A is to a as B is to b", formal concepts in WAP and an analogical complex from this knowledge.

As an example, we have found in a web magazine[3] the following proportion "Massimiliano Alajmo is the Mozart of Italian cooking". The background knowledge allowing to understand this relational proportion is the following: music and Italian cooking are disciplines practiced by humans, such disciplines can be practiced with different levels of ability, Mozart is a musician and Mozart is a genius in music discipline. Since the quality "to be a genius" is not possessed by everybody, there must exist many "ordinary gifted" musicians. Then, the background knowledge can be expressed by the following formal context:

	a_1	a_2	a_3
o_1	×	×	
o_2	×		×

where o_1 stands for Mozart, o_2 for one of "ordinary gifted" musicians, a_1 is the attribute "practices music", a_2 "is a genius" and a_3 "has an ordinary ability".

Now, when the new data "Alajmo is the Mozart of Italian cooking" is introduced, the knowledge extends as follows: Alajmo practices Italian cooking, and he has something in common with Mozart that is not Italian cooking. The relational proportion is a reduced form of "Alajmo is to Italian cooking as Mozart is to music". Since Mozart has only the other attribute "Genius", Alajmo must have it. Moreover, since cooking is a discipline practiced by humans, there must exist some ordinary gifted Italian cook. At last, we must introduce the notion of non-genius in our universe. If we do not, we implicitly suppose that everybody is a genius for some activity. The knowledge is now as follows

	a_1	a_2	a_3	a_4
o_1	×	×		
o_2	×		×	
o_3		×		×
a_4			×	×

where o_3 stands for Alajmo, o_4 an ordinary gifted Italian cook and a_4 Italian cooking. This context is called the *analogical context*. Considering the associated concept lattice, the closest analogical proportion to "Alajmo is the Mozart of Italian cooking" is $(\{o_3\},\{a_2,a_4\}) : (\{o_4\},\{a_3,a_4\})\ WAP\ (\{o_1\},\{a_1,a_2\}) : (\{o_2\},\{a_1,a_3\})$ which translates into "Mozart is to some ordinary musician as Alajmo is to some ordinary cook".

More formally, from the relational proportion "o_1 is the o_2 of a", we can derive an analogical context as above. It is composed of objects o_1 and o_2, described by four attributes: a is possessed by o_1 and not by o_2, $\tilde{a}$ is possessed by o_2 and not by o_1, b is possessed both by o_1 and o_2 and $\tilde{b}$ is some attribute not possessed by o_1 nor o_2. Secondly we complete the context with two objects o_3 and o_4 that are the complements of o_2 and o_1 with respect to the four attributes. The resulted context is the analogical context where $a_1 = b$, $a_2 = a$, $a_3 = \tilde{a}$ and $a_4 = \tilde{b}$.

[3] http://www.slate.fr/story/43841/massimiliano-alajmo

4.2 Analogical complex

In the previous paragraph, it turns out that the analogical context is an interesting pattern, from which we can extract relational proportion. A more general definition of this pattern, named *analogical complex*, has been given in [7].

An analogical complex is a subcontext of a formal context described by:

		×	×
	×		×
×		×	
×	×		

associated with the binary matrix $AS = \begin{pmatrix} 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 \end{pmatrix}$. Matrix AS exhibits characteristic pattern of a Boolean analogical proportion [2] and is called an *analogical schema*. We write $AS(i, j)$ if its value at row i and column j is 1.

Definition 4. *Given a formal context* $(\mathcal{O}, \mathcal{A}, R)$*, a set of objects* $\mathbf{o} \subseteq \mathcal{O}$*,* $\mathbf{o} = \mathbf{o}_1 \cup \mathbf{o}_2 \cup \mathbf{o}_3 \cup \mathbf{o}_4$*, a set of attributes* $\mathbf{a} \subseteq \mathcal{A}$*,* $\mathbf{a} = \mathbf{a}_1 \cup \mathbf{a}_2 \cup \mathbf{a}_3 \cup \mathbf{a}_4$*, and a binary relation* R*, the subcontext* $(\mathbf{o}, \mathbf{a})$ *forms an analogical complex* $(\mathbf{o}_{1,4}, \mathbf{a}_{1,4})$ *iff*

1. *the binary relation is compatible with the analogical schema AS:*
   $\forall i \in [1, 4],\ \forall o \in \mathbf{o}_i,\ \forall j \in [1, 4],\ \forall a \in \mathbf{a}_j,\ ((o, a) \in R) \Leftrightarrow AS(i, j).$
2. *The context is maximal with respect to the first property (*$\oplus$ *denotes the exclusive or and* $\setminus$ *the set-theoretic difference):*
   $\forall o \in \mathcal{O} \setminus \mathbf{o}, \forall i \in [1, 4],\ \exists j \in [1, 4], \exists a \in \mathbf{a}_j, ((o, a) \in R) \oplus AS(i, j).$
   $\forall a \in \mathcal{A} \setminus \mathbf{a}, \forall j \in [1, 4],\ \exists i \in [1, 4], \exists o \in \mathbf{o}_i, ((o, a) \in R) \oplus AS(i, j).$

An analogical complex is complete if none of sets $\mathbf{a}_1, \ldots, \mathbf{a}_4, \mathbf{o}_1, \ldots, \mathbf{o}_4$ *are empty.*

Comments.

1. In order to simplify the notations, the Cartesian products $\mathbf{o}_1 \times \ldots \times \mathbf{o}_4$ and $\mathbf{a}_1 \times \ldots \times \mathbf{a}_4$ are respectively denoted $\mathbf{o}_{1,4}$ and $\mathbf{a}_{1,4}$.
2. In [7], it has been shown that the set of the analogical complexes of any formal context is itself structured as a lattice.

Example 4. Let us consider a subcontext, called SmallZoo, extracted from the Zoo data base [8], it has been shown in [7] that 24 analogical complexes (18 complete ones) can be derived, like the following complete one:

SmallZoo	hair	feathers	eggs	milk	airborne	aquatic	predator	toothed
	a_0	a_1	a_2	a_3	a_4	a_5	a_6	a_7
o_0 aardvark	×			×			×	×
o_1 chicken		×	×		×			
o_2 crow		×	×		×		×	
o_3 dolphin				×		×	×	×
o_4 duck		×	×		×	×		
o_5 fruitbat	×			×	×			×
o_6 kiwi		×	×				×	
o_7 mink	×			×		×	×	×
o_8 penguin		×	×			×	×	
o_9 platypus	×		×	×		×	×	

		$\mathbf{a}_1$	$\mathbf{a}_2$			$\mathbf{a}_3$		$\mathbf{a}_4$
		a_5	a_0	a_3	a_7	a_1	a_2	a_4
$\mathbf{o}_1$	o_1					×	×	×
	o_2					×	×	×
$\mathbf{o}_2$	o_5		×	×	×			×
$\mathbf{o}_3$	o_8	×				×	×	
$\mathbf{o}_4$	o_7	×	×	×	×			

From the analogical complex structure, we derive a formal definition of a relational proportion.

Definition 5. *Let* $(\mathbf{o}_{1,4}, \mathbf{a}_{1,4})$ *be a complete analogical complex in a formal context, the following sets of objects and attributes are said to be in the* formal relational proportion $(\mathbf{o}_1$ *is to* $\mathbf{a}_3$ *as* $\mathbf{o}_2$ *is to* $\mathbf{a}_2)$*, and we write:* $(\mathbf{o}_1 \Updownarrow \mathbf{a}_3 \between \mathbf{o}_2 \Updownarrow \mathbf{a}_2)$.

Comments.

1. The reduced form of the relational proportion would be ($\mathbf{o}_1$ is the $\mathbf{o}_2$ of $\mathbf{a}_3$).
2. From the same complex, we can extract the 4 following formal relational proportions $\big(\mathbf{o}_1 \Updownarrow \mathbf{a}_4 \between \mathbf{o}_3 \Updownarrow \mathbf{a}_1\big)$, $\big(\mathbf{o}_2 \Updownarrow \mathbf{a}_4 \between \mathbf{o}_4 \Updownarrow \mathbf{a}_1\big)$ and $\big(\mathbf{o}_3 \Updownarrow \mathbf{a}_3 \between \mathbf{o}_4 \Updownarrow \mathbf{a}_2\big)$. Since the operator $\between$ is commutative, it gives a total of 8, but permuting the extreme and the means in a relational proportion may lead to awkward phrasings.

Example 5. Let us take the complex from SmallZoo described above. It implies all attributes but a_6 (predator) and objects o_1 and o_2 (chicken and crow), o_5 (fruitbat), o_8 (penguin) and o_7 (mink). From this context, the RP in reduced form "a fruitbat is the mink of airborne animals" can be derived for instance, meaning that fruitbat and mink have hair, are toothed and produce milk, but that the mink is aquatic at the contrary of the fruitbat. Of course, the interest of such phrases has to be taken in context: the SmallZoo data base is supposed to be the only knowledge.

4.3 WAP and analogical complex

In this section we explore the links between WAP between concepts and complete analogical complex, and then the formal relational proportion.

First, we are interested in defining a non degenerated WAP, called *complete*, forbidding inclusion between two of its concepts. It is a key notion for building WAPs between concepts with a sound cognitive interpretation.

Definition 6. *Let us consider* $(x : y\, WAP\, z : t)$*, this WAP is complete when*

1. *either* $(\mathbf{a}_x \cap \mathbf{a}_y) \setminus \mathbf{a}_\cap$, $(\mathbf{a}_x \cap \mathbf{a}_z) \setminus \mathbf{a}_\cap$, $(\mathbf{a}_y \cap \mathbf{a}_t) \setminus \mathbf{a}_\cap$ *and* $(\mathbf{a}_z \cap \mathbf{a}_t) \setminus \mathbf{a}_\cap$ *are nonempty (called complete WAP through attributes),*
2. *or* $(\mathbf{o}_x \cap \mathbf{o}_y) \setminus \mathbf{o}_\cap$, $(\mathbf{o}_x \cap \mathbf{o}_z) \setminus \mathbf{o}_\cap$, $(\mathbf{o}_y \cap \mathbf{o}_t) \setminus \mathbf{o}_\cap$ *and* $(\mathbf{o}_z \cap \mathbf{o}_t) \setminus \mathbf{o}_\cap$ *are nonempty (called complete WAP through objects).*

where $\mathbf{a}_\cap = \mathbf{a}_x \cap \mathbf{a}_y \cap \mathbf{a}_z \cap \mathbf{a}_t$ *and* $\mathbf{o}_\cap = \mathbf{o}_x \cap \mathbf{o}_y \cap \mathbf{o}_z \cap \mathbf{o}_t$.

Proposition 5. *1. A complete WAP is an* antichain *of concepts.*

2. *For a complete WAP through attributes,* $(x \vee y)$, $(x \vee z)$, $(y \vee t)$ *and* $(z \vee t)$ *are in antichain. Similarly, for a complete WAP through objects,* $(x \wedge y)$, $(x \wedge z)$, $(y \wedge t)$ *and* $(z \wedge t)$ *are in antichain.*
3. *A FAP in antichain forms a complete WAP through attributes and objects, and reciprocally.*

Proof. 1. Let us suppose that $(x : y\, WAP\, z : t)$ and $x \leq y$. From Preliminary 2, we get $\mathbf{a}_x \cap \mathbf{a}_y = \mathbf{a}_{x\vee y}$. Then $\mathbf{a}_x \cap \mathbf{a}_y = \mathbf{a}_y$ and using Proposition 2

$$\begin{aligned}\mathbf{a}_x \cap \mathbf{a}_z &= (\mathbf{a}_x \cap \mathbf{a}_y) \cap \mathbf{a}_z = \mathbf{a}_x \cap (\mathbf{a}_y \cap \mathbf{a}_z) \\ &= \mathbf{a}_x \cap (\mathbf{a}_x \cap \mathbf{a}_t) = \mathbf{a}_x \cap \mathbf{a}_t = \mathbf{a}_\cap \, .\end{aligned}$$

Thus, $\mathbf{a}_x \cap \mathbf{a}_z \setminus \mathbf{a}_\cap = \varnothing$ and $(x : y\, WAP\, z : t)$ is not a complete WAP.

2. From a complete WAP through attributes, $\mathbf{a}_{x\vee y} = \mathbf{a}_x \cap \mathbf{a}_y$ and three analog equalities hold. Due to this completeness, there is no inclusion between $\mathbf{a}_{x\vee y}$, $\mathbf{a}_{x\vee z}$, $\mathbf{a}_{z\vee t}$ and $\mathbf{a}_{y\vee t}$. The associated concepts are then in antichain.

3. Let us consider the FAP $x : y :: z : t$ where $\{x, y, z, t\}$ is an antichain. From Proposition 2, we have $x = (x \wedge y) \vee (x \wedge z)$ and $x = (x \vee y) \wedge (x \vee z)$ which are equivalent to $\mathbf{a}_x = \mathbf{a}_{x\wedge y} \cap \mathbf{a}_{x\wedge z}$ and $\mathbf{o}_x = \mathbf{o}_{x\vee y} \cap \mathbf{o}_{x\vee z}$ thanks to Preliminary 2. Similarly, $\mathbf{a}_y = \mathbf{a}_{x\wedge y} \cap \mathbf{a}_{y\wedge t}$ and $\mathbf{o}_x = \mathbf{o}_{x\vee y} \cap \mathbf{o}_{y\vee t}$ and

$$\begin{aligned}\mathbf{a}_\cap &= \mathbf{a}_x \cap \mathbf{a}_y \cap \mathbf{a}_z \cap \mathbf{a}_t \;=\; \mathbf{a}_x \cap \mathbf{a}_t \\ \mathbf{o}_\cap &= \mathbf{o}_x \cap \mathbf{o}_y \cap \mathbf{o}_z \cap \mathbf{o}_t \;=\; \mathbf{o}_x \cap \mathbf{o}_t\end{aligned}$$

due to Proposition 2 and the fact that a FAP is a WAP. Therefore, we have $\mathbf{a}_{x\wedge y} \setminus (\mathbf{a}_x \cap \mathbf{a}_t) \subseteq \mathbf{a}_x \cap \mathbf{a}_y \setminus \mathbf{a}_\cap$. Moreover, $\mathbf{a}_{x\wedge y} \setminus (\mathbf{a}_x \cap \mathbf{a}_t) = \mathbf{a}_{x\wedge y} \setminus \mathbf{a}_{x\vee t}$ is nonempty. Indeed, $\mathbf{a}_{x\wedge y} \setminus \mathbf{a}_{x\vee t} = \varnothing$ implies that $x \vee t \leq x \wedge y$ which is impossible since $\{x, y, z, t\}$ is an antichain. Similarly, we can prove that $\mathbf{o}_x \cap \mathbf{o}_y \setminus \mathbf{o}_\cap \neq \varnothing$.

Reciprocally, let us take a complete WAP through attributes and objects. From the previous properties, $\{x, y, z, t\}$ is an antichain, as well as $\{x \vee y, x \vee z, y \vee t, z \vee t\}$ and $\{x \wedge y, x \wedge z, y \wedge t, z \wedge t\}$. Therefore, these 12 concepts are distinct and it can be proved that they generate a Boolean sublattice. Because of the distributivity of this sublattice, the WAP $(x : y\, WAP\, z : t)$ is then a FAP.

In order to derive relational proportion from an analogical proportion between concepts, we consider a complete WAP through attributes (a similar reasoning can be done from a complete WAP through objects) and introduce a process to extract an analogical complex.

Due to the completeness, sets $\mathbf{a}_1 = (\mathbf{a}_z \cap \mathbf{a}_t) \setminus \mathbf{a}_\cap$, $\mathbf{a}_2 = (\mathbf{a}_y \cap \mathbf{a}_t) \setminus \mathbf{a}_\cap$, $\mathbf{a}_3 = (\mathbf{a}_x \cap \mathbf{a}_z) \setminus \mathbf{a}_\cap$ and $\mathbf{a}_4 = (\mathbf{a}_x \cap \mathbf{a}_y) \setminus \mathbf{a}_\cap$ are nonempty. We also define $\mathbf{o}_1 = \widetilde{\mathbf{o}_x}$ the set of objects proper to x (that appear in $\mathbf{o}_x$ but not in the objects of y, z and t) and similarly $\mathbf{o}_2 = \widetilde{\mathbf{o}_y}$, $\mathbf{o}_3 = \widetilde{\mathbf{o}_z}$ and $\mathbf{o}_4 = \widetilde{\mathbf{o}_t}$.

By construction, every object of $\mathbf{o}_1$ is in relation with every attribute of $\mathbf{a}_3 \cup \mathbf{a}_4$. It is also the case between $\mathbf{o}_2$ and $\mathbf{a}_2 \cup \mathbf{a}_4$, $\mathbf{o}_3$ and $\mathbf{a}_1 \cup \mathbf{a}_3$, $\mathbf{o}_4$ and $\mathbf{a}_1 \cup \mathbf{a}_2$. For all the other combinations, for instance $\mathbf{o}_1$ and $\mathbf{a}_1$, for any $o \in \mathbf{o}_1$, there exists $a \in \mathbf{a}_1$ such that o and a are not in relation. However, these properties do not guarantee that the subcontext $(\mathbf{o}_{1,4}, \mathbf{a}_{1,4})$ is an analogical schema, even if it is a closed schema. Indeed, it can exists an object $o \in \mathbf{o}_i$ in relation with an attribute $a \in \mathbf{a}_j$, where $(i, j) \in \{(1,1), (1,2), (2,1), (2,3), (3,2), (3,4), (4,3), (4,4)\}$. In such a case, either a or o is removed and this postprocessing permits to obtain an analogical schema. But this schema is not necessarily a complex, since the associated subcontext may be not maximal. Then a second postprocessing maximises the schema into complex, adding new attributes and/or objects chosen among those which do not appear in $\mathbf{a}_x \cup \ldots \cup \mathbf{a}_t$ nor $\mathbf{o}_x \cup \ldots \cup \mathbf{o}_t$. Finally, we check that the resulting analogical complexes are complete.

This method can lead to several complexes, according to the choices in both postprocessings. This set of complexes is a sub-lattice of the lattice of complexes.

Example 6. In SmallZoo, $x = (\{o_1, o_2, o_4\}, \{a_1, a_2, a_4\})$, $y = (\{o_5\}, \{a_0, a_3, a_4, a_7\})$, $z = (\{o_4, o_8\}, \{a_1, a_2, a_5\})$, $t = (\{o_7, o_9\}, \{a_0, a_3, a_5, a_6\})$ are concepts in complete WAP through attributes. At the beginning, $\mathbf{o}_1 = \{o_1, o_2\}$, $\mathbf{o}_2 = \{o_5\}$, $\mathbf{o}_3 = \{o_8\}$, $\mathbf{o}_4 = \{o_7, o_9\}$, $\mathbf{a}_1 = \{a_5\}$, $\mathbf{a}_2 = \{a_0, a_3\}$, $\mathbf{a}_3 = \{a_1, a_2\}$ and $\mathbf{a}_4 = \{a_4\}$ and, due to the relation between o_9 and a_2, the first postprocessing can remove (either o_9 or) a_2:

	a_5	a_0	a_3	a_1	a_2	a_4
o_1				×	×	×
o_2				×	×	×
o_5		×	×			×
o_8	×			×	×	
o_7	×	×	×			
o_9	×	×	×		×	

		$\mathbf{a}_1$	$\mathbf{a}_2$		$\mathbf{a}_3$	$\mathbf{a}_4$
		a_5	a_0	a_3	a_1	a_4
$\mathbf{o}_1$	o_1				×	×
	o_2				×	×
$\mathbf{o}_2$	o_5		×	×		×
$\mathbf{o}_3$	o_8	×			×	
$\mathbf{o}_4$	o_7	×	×	×		
	o_9	×	×	×		

After removing a_2, the right table is an analogical schema and we can check that it is maximal in SmallZoo. Note that if we had chosen to remove o_9, the postprocessings would have produced the analogical complex previously detailed in Example 4.

For example, from the complete analogical complex described above, we can derive the following relational proportion: "the chicken and the crow are to the feathers as the fruitbat is to the hair, the milk and the teeth". It makes sense when considering that all these animals share the attribute "airborne".

Likewise, another proportion from the same complex is "the fruitbat is to the airborne animals as the mink and the platypus are to the aquatic animals" (fruitbat, mink and platypus share the attributes hair and milk). The reduced form "the fruitbat is the mink of airborne animals" is the same as that of Example 5, Section 4.2, although the complexes involved are slightly different.

5 Conclusion

The paper has shown how relational proportions can be identified in a formal context. Relational proportions offer a basis for concise forms of explanations. Indeed, if B has some well-known features, the proportion "object A is to attribute a as object B is to attribute b" provides an argument for stating that "object A is the B of a", when A possesses these well-known features also, as in "Carlsen is the Mozart of chess". It is worth pointing out that two cognitive capabilities, namely conceptual categorization and analogical reasoning can be handled together in the setting of formal concept analysis. This introductory presentation has left aside the algorithmic side (based on the identification of formal complexes), which is discussed in the long version of the paper [10].

Our study of proportions between concepts explores a simple and fixed relation between concepts in a single lattice. It would be interesting to connect it with the general framework of Relational Concept Analysis (see e.g., [11]), and with a recent proposal based on antichains [12].

Following the pioneering work of Rumelhart and Abrahamson [13], a number of recent woks in computational linguistics (e.g., [14]) have been using a parallelogram-based modeling of analogical proportions in numerical settings, where words are represented by vectors of great dimension. Bridging this computational view of analogical proportions with the work presented here is certainly a challenging task for the future.

References

1. Lepage, Y.: De l'analogie rendant compte de la commutation en linguistique. Université Joseph Fourier, Grenoble (2001). www.slt.atr.jp/~lepage/pdf/dhdryl.pdf
2. Miclet, L., Prade, H.: Handling Analogical Proportions in Classical Logic and Fuzzy Logics Settings. In: 10th Eur. Conf. on Symbolic and Quantitative Approaches to Reasoning with Uncertainty, pp. 638–650, Verona (2009).
3. Stroppa, N., Yvon, F.: Du quatrième de proportion comme principe inductif : une proposition et son application à l'apprentissage de la morphologie. Traitement Automatique des Langues 47(2), 1–27 (2006).
4. Miclet, L., Barbot, N., Prade, H.: From Analogical Proportions in Lattices to Proportional Analogies in Formal Contexts. In: 21st of Eur. Conf. on Artificial Intelligence, Prague (2014).
5. Bělohlávek, R.: Introduction to formal concept analysis. Technical report, Palacký University, (2008).
6. Ganter, B., Wille, R.: Formal Concept Analysis: Mathematical Foundations. Springer-Verlag (1998).
7. Miclet, L., Nicolas, J.: From formal concepts to analogical complexes. In: 12th Int. Conf. on Concept Lattices and their App., pp. 159–170, Clermont-Ferrand (2015).
8. Lichman, M.: UCI Machine Learning Repository. University of California, Irvine, School of Information and Computer Sciences (2013). http://archive.ics.uci.edu/ml
9. Dubois, D., Prade, H., Richard, G.: Multiple-valued extensions of analogical proportions, Fuzzy Sets and Systems v.292, 193–202 (2016).
10. Barbot, N., Miclet, L., Prade, H.: Analogy between concepts. Submitted.
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, v. 67, 81-108 (2013).
12. Valverde-Albacete, F. J., Peláez-Moreno, C., Cabrera, I. P., Cordero, P., Ojeda-Aciego, M.: Formal independence analysis. In: 17th Int. Conf. Information Processing and Management of Uncertainty in Knowledge-Based Systems. Theory and Foundations, Jesús Medina, et al. (ed.), CCIS 853, Springer, 596–608, Cádiz (2018).
13. Rumelhart, D. E., Abrahamson, A. A.: A model for analogical reasoning. Cognitive Psychol. v.5, 1–28 (1973).
14. Mikolov, T., Sutskever, I., Chen, K., Corrado, G., Dean, J.: Distributed representations of words and phrases and their compositionality. Advances in Neural Information Processing Systems 26 (2013).

MDL for FCA: is there a place for background knowledge?

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. The Minimal Description Length (MDL) principle is a powerful and well founded approach, which has been successfully applied in a wide range of Data Mining tasks. In this paper we address the problem of pattern mining with MDL. We discuss how constraints – background knowledge on interestingness of patterns – can be embedded into MDL and argue the benefits of MDL over a simple selection of patterns based on measures.

1 Introduction

Formal Concept Analysis (FCA) is a formalism that can be applied to Knowledge Discovery and Data Mining. It is used commonly for solving a wide range of tasks: from pattern mining to design of ontologies.

Even controlled application of FCA in practice may result in exponentially large output, which entails additional steps aimed at reducing / selecting a small subset of concepts. The reduction of the number of formal concepts may be done during pre-/postprocessing stages.

In this paper we propose to combine two of the most common concept filtering approaches: the Minimal Description Length principle (MDL) [2, 6, 7, 12] and measure-based selection [8]. This combination tries to take the advantages of both methods and reduces the drawbacks of each one.

The idea of MDL is to select a subset of patterns that ensures the best compression of data. It has been embedded into FCA in a number of ways: for defining how many factors to use in Boolean matrix factorization (BMF) [3,10,11] or to get more diverse itemsets in frequent pattern mining (FIM) [1, 9, 13] or to select triclusters [14]. Being threshold-free, MDL provides a succinct non-redundant set of concepts. However, it has some shortcomings. Since the length minimisation is at least NP-complete, the implementation of MDL is based on heuristics. The selected itemsets cannot be interpreted easily by experts.

Unlike MDL, the selection of itemsets based of values of some measure is easy to interpret. A measure reflects the assumption on interestingness of patterns. Selecting the best itemsets w.r.t. the chosen measure one obtains patterns with the desired characteristics. However, this approach requires threshold and returns a lot of similar patterns.

In this paper we use the Krimp algorithm as an implementation of MDL principle to improve measure-based selection. Krimp is based on greedy covering of data by a set of patterns (subsets of attributes) called candidate set. The patterns in a candidate set are ordered w.r.t. the pattern length and its frequency. We propose to use different interestingness measures to order candidates. This modification allows for embedding background knowledge, i.e., our assumptions on interestingness. The aim of the ordering w.r.t. different measures is to improve measure-based pattern selection rather than to compress data the best. Using a preferable ordering one gets a good compression as well as only those patterns that satisfy defined constraints.

The rest of the paper has the following structure. In Section 2 we briefly recall the main notions of FCA. In Section 3 we describe the MDL principle and discuss how interestingness measures can be used within MDL. The benefits of our approach are discussed in Section 4, where we compare MDL-based with threshold-based measure selection. Section 5 gives the conclusion and discuss the direction of future work.

2 Formal Concept Analysis: 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, $M = \{m_1, m_2, \ldots, m_k\}$ 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 or itemset) $B \subseteq M$ 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 formal concept is said to cover set of objects A and set of attributes 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 can grow exponentially w.r.t. the size of a formal context, i.e., the number of objects in G and attributes in M. We say that a set of patterns $\mathcal{S}$ covers objects in G if $\bigcup_{B \in \mathcal{S}} B' = G$, where $B \subseteq M$. We are interested in a small set of patterns (intents) $\mathcal{S}$ that covers all objects and most of their attributes, i.e., $\left|\bigcup_{B \in \mathcal{S}} \{gIm \mid g \in B', m \in B\}\right| \approx |I|$.

Example. Let us consider a toy example. A formal context is given in Figure 1 (1). We consider 3 sets of itemsets (intents): $\mathcal{S}_2 = \{\{abc\}, \{bcde\}, \{de\}, \{cde\}, \{ac\}\}$, $\mathcal{S}_3 = \{\{bc\}, \{de\}, \{ac\}\}$ and $\mathcal{S}_4 = \{\{c\}, \{de\}\}$. The corresponding coverings of the context are given in Figure 1 (2-4). The intensity of colors is proportional to the number of times a particular "cross" is covered by intents. In our example "crosses" are covered by intents from 0 to 4 times. We count not only the number of intents, but also the number of covered "crosses", we call this value the rate of a cover relation, or $RCR = |\text{"crosses" that covered at least once}|/|I|$. It can be seen from the covering given in Figure 1 that $\mathcal{S}_3$ (Figure 1, (3)) provides the best covering w.r.t. the number of itemsets (intents) and the rate of covered elements in the object-attribute relation.

	a	b	c	d	e
1	x	x	x		
2		x	x	x	x
3				x	x
4	x		x	x	x
5	x		x		

(1) A formal context

	a	b	c	d	e
1	x	x	x		
2		x	x	x	x
3				x	x
4	x		x	x	x
5	x		x		

(2) Covering of objects with $\mathcal{S}_2 = \{\{abc\}, \{bcde\}, \{de\}, \{cde\}, \{ac\}\}$. RCR = 1.

	a	b	c	d	e
1	x	x	x		
2		x	x	x	x
3				x	x
4	x		x	x	x
5	x		x		

(3) Covering of objects with $\mathcal{S}_3 = \{\{bc\}, \{de\}, \{ac\}\}$. RCR = 1.

	a	b	c	d	e
1	x	x	x		
2		x	x	x	x
3				x	x
4	x		x	x	x
5	x		x		

(4) Covering of objects with $\mathcal{S}_4 = \{\{c\}, \{de\}\}$. RCR = 10/15.

Fig. 1. Formal context and its coverings.

The Minimal Description Length principle allows for covering with a substantial rate of "crosses" in I by a small number of patterns. In the next section we use MDL in a more general framework, i.e., in pattern mining. Intents of formal concepts, in turn, can be considered as patterns of a special kind.

3 Minimal Description Length: Basic Notions

MDL is aimed to find a subset of patterns that compresses data the best. In our study we use Krimp [13] as a practical implementation of this principle. In Section 3.1 we give a short description of it and in Section 3.2 we discuss how background knowledge can be embedded into MDL.

3.1 MDL in Practice: the Krimp Algorithm

The input of the algorithm is a dataset and a list of patterns (that are computed on the same dataset). The patterns are ordered w.r.t. their length and frequency. The result of Krimp is a two-column code table that consists of patterns and their encoding lengths (an examples of code tables are given in Figure 2). The objective of Krimp is minimization of the function

$$L(D, CT) = L(D \mid CT) + L(CT \mid D), \tag{1}$$

where $L(D \mid CT)$ is the length of the dataset $D = \{g' \mid g \in G\}$ encoded with the code table CT and $L(CT \mid D)$ is the length of the code table CT computed w.r.t. D. The objects are encoded by disjoint patterns in a greedy manner, i.e., starting from the top elements of CT. The length of pattern B is computed using an optimal prefix code given by Shannon entropy, i.e., the length $l(B) = -\log(u(B)/U)$ is inversely proportional to the usage $u(B) = |\{t \in D \mid B \in cover(t, CT)\}|$. The usage shows how many times B is used to cover objects in D, $U = \sum_{B \in CT} u(B)$ is the total usage of itemsets. We leave the details on itemset storage out of scope of this paper and take into account the compression related to a particular choice of itemsets, i.e., we use the simplified version of the lengths:

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

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

A code table is incrementally computed. At the beginning it contains only single-attribute patterns $\{\{m\} \mid m \in M\}$. A set of patterns – candidates in the code table – are ordered w.r.t. their length (intent cardinality) and frequency (extent cardinality). At each step the best candidate is added to the code table if its usage allows for smaller encoding length, otherwise it is removed from the code table and the candidate set.

Example. Let us consider how Krimp selects patterns using the running example (the context is given in Figure 1, (1)). Here we represent the context as a set of transactions, see Figure 2, (1). The main stages are given in Figure 2, (2-4). As candidates we use intents of formal concepts with the size of intent and extent exceeding 1. We sort them first by the size of intent and then by the size of extent (in descending order). Let us consider some steps of the algorithm.

Initial state (Figure 2, (2)): the code table consists of single-attribute patterns. Usage is equal to frequency. Sets of attributes in the dataset are covered by single-attribute patterns.

First step (Figure 2, (3)): An attempt to add the top pattern from the candidate set. Pattern ac is used to cover object g_1, g_4 and g_5 (Figure 2, (3)), the usage of single attributes a and c decreases by 3. The description length (see Formula 1) is computed for the updated code table and covering. Since the inclusion of ac into the code table provides smaller description length, ac is accepted for the code table.

Further, the top patterns one by one are used to minimize the description length.

Last step (Figure 2, (4)): The last candidate bc can cover only object g_2 (since subsets bc are partially covered by other members of the code table). It is not added since its inclusion in the code table does not provide better compression (i.e. smaller description length). The subset of MDL-optimal patterns is $\{\{ac\}, \{de\}\}$.

Objects	Desctiption
g_1	acb
g_2	bcde
g_3	de
g_4	acde
g_5	ac

(1) A formal context from Figure 1, (1) represented as a set of transactions

CT	
IS	U
a	3
b	2
c	4
d	3
e	3

CS
ac
de
bc

Covering
(a)(b)(c)
(b)(c)(d)(e)
(d)(e)
(a)(c)(d)(e)
(a)(c)

(2) Initial state.

CT	
IS	U
a	0
b	2
c	1
d, e	3
ac	3

CS
de
bc

Covering
(ac)(b)
(b)(c)(d)(e)
(d)(e)
(ac)(d)(e)
(ac)

(3) The state after adding *ac* to the code table.

CT	
IS	U
a,d,e	0
b	2
c	1
ac	3
de	3

CS

Covering
(ac)(b)
(b)(c)(de)
(de)
(ac)(de)
(ac)

(4) The last step. *bc* is not added to the code table.

Fig. 2. The main stages of the Krimp algorithm. "Covering" tables show the dataset with covering by itemsets from the corresponding code table above the covering, $(\cdot)$ depicts an itemset that covers some attributes of an object. CT is a two-column code table, where "IS" and"U" stand for itemsets and their usage in greedy covering, respectively. "CS" is a candidate set.

3.2 MDL in Practice: Compression under Constraints

The implementation of the MDL principle is based on heuristics and allows for the solution which is close to the optimal one. In practice, there exist several ways to select subsets of patterns that have almost the same size and ensure good compression. Thus, it becomes difficult to explain why a particular subset was chosen.

More than that, by selecting a subset of patterns one is interested in patterns that have particular properties, e.g., being stable w.r.t. noise, have high probability under certain condition, etc. Despite proper interpretability, the application of interestingness measures requires a threshold and results in a redundant set of patterns (quite similar patterns). As interestingness measures of concept (A, B) we took frequency $fr(B) = |A|$, i.e. the size of extent, length $len(B) = |B|$, i.e., the size of the intent, and lift $lift(B) = \prod_{b \in B} Pr(b)/Pr(B)$, where $Pr(\cdot) = |(\cdot)'|/|G|$.

In our study we combine the measure-based selection with Krimp to get a threshold-free approach that provides a small non-redundant subset of patterns having desired properties. The modified approach works as follows. First, all patterns are sorted w.r.t. chosen interestingness measures. Then the ordered set is considered as a candidate set. The greedy covering strategy (Krimp) is applied to select the most interesting and diverse patterns. The original workflow and the adapted version that is used in the paper are given in Figure 3.

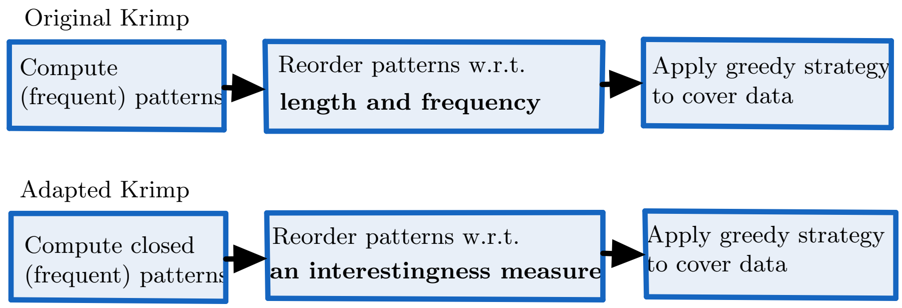


Fig. 3. The workflow for pattern mining by the original Krimp and its adapted version.

In the next section we show how the embedding of background knowledge (i.e. reordering of patterns w.r.t. interestingness measures) affects the results of pattern mining.

4 MDL in Closed Itemset Mining

In the worst case a concept lattice contains an exponential number of partially ordered intents (concepts), the application of MDL allows for the selection of a small subset of intents. Our experiments show that the application of the MDL principle allows for significant reduction in the number of patterns (up to 5% of the formal concepts, see Table 2). In the context of measure-based pattern mining, the application of MDL makes the measure-based selection threshold-free. More than that, a set of the MDL-optimal patterns has better characteristics than the top-n patterns. First of all, almost the same concepts (intents) are removed from the set of selected patterns. In our experiments we call this property "non-redundancy". For a set of patterns to be "non-redundant" means to have the following characteristics: differ from the most similar pattern in the set (i.e., distance to the 1st nearest neighbor), make shallow hierarchy by inclusion $B_1 \subset B_2 \subset \ldots \subset B_n$ (i.e., average length of the longest paths built from partially ordered itemsets) and do not have a lot of more general patterns $B_i \subset B$, $i \in [1, k]$ (the rate of patterns with children).

If we compare the sets of top-n and MDL-optimal patterns of the same size we will see, as a side effects of the "non-redundancy", that MDL-optimal patterns cover in total more data ("crosses" in a formal context) being diverse and interesting w.r.t. a given measure.

It is clear to see that MDL approach not only dispenses from predefined thresholds but also filter out similar interesting patterns and provides more comprehensive data description.

We examine the following orders of patterns: $area\_fr\_lift(B) = fr(B) \cdot lift(B)$, $area\_len\_fr(B) = len(B) \cdot fr(B)$, $area\_len\_lift(B) = len(B) \cdot lift(B)$ and sequential ordering by len and fr, len and $lift$, $lift$ and len (the patterns

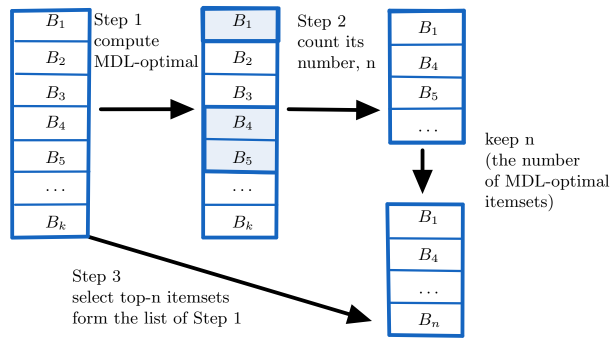


Fig. 4. The principle of computing MDL-optimal and top-n sets of patterns

are ordered by the chosen measure on Step 1 in Figure 4). An example of ordering for frequent closed itemsets (frequency is greater than 2) for the running example is given in Table 1.

Table 1. Values of interestingness measures and ordering of patterns for the running example from Figure 1, (1). An alternative ordering is given in $(\cdot)$, to select one ordering among the altervative ones additional rules are required to set.

Concepts (A, B)	fr $\|A\|$	len $\|B\|$	area (len × fr)	patterns ordered w.r.t. length and frequency **values of measures**	patterns ordered w.r.t. area_len_fr **values of measures**
({1245},{c})	4	1	4	{cde}; **3,2**	{de} ({ac}, {cde}); **6**
({234},{de})	3	2	6	{de} ({ac}); **2,3**	{ac} ({de}, {cde}); **6**
({145},{ac})	3	2	6	{ac} ({de}); **2,3**	{cde} ({de}, {ac}); **6**
({12},{bc})	2	2	4	{bc}; **2,2**	{c} ({bc}); **4**
({24},{cde})	2	3	6	{c}; **1,4**	{bc} ({c}); **4**

The discretized datasets from LUCS-KDD repository [4] were used in the study, the parameters of the datasets are given in Table 2. We split each dataset into 10 parts and in each of 10 experiments we use 9 of them as a training set and one part as a test set.

In this section we compare characteristics of MDL-optimal with top-n itemsets, patterns in both sets are ordered w.r.t. the same interestingness measure. The size of a set of top-n itemsets is equal to the size of a set of MDL-optimal patterns. The scheme of computing these sets is given in Figure 4. We compare the sets of patterns within the following properties: non-redundancy, data covering and representativeness.

Table 2. Characteristics of datasets

dataset	nmb. of obj.	nmb. of attr.	nmb. of concepts	Number of MDL-optimal					
				area fr_lift	area len_fr	area len_lift	len fr	len lift	lift len
breast	699	16	702	36.0	32.2	20.4	37.3	37.3	33.5
car	1 728	25	12 420	868.4	849.2	138.6	714.6	847.7	698.3
ecoli	336	29	690	58.8	55.9	16.4	64.9	65.6	55.9
iris	150	19	183	31.1	28.9	12.9	34.8	34.6	26.3
led7	3 200	24	3 808	108.0	118.3	64.2	108.7	108.7	130.3
pima	768	38	2 769	110.1	106.3	35.9	120.6	112.1	101.7

4.1 Non-redundancy

By redundant set of patterns we mean a set of patterns that contains a lot of similar itemsets. We measure redundancy by three parameters: average distance to the 1st nearest neighbor, average length of the longest paths built from partially ordered itemsets, and average number of itemsets that have at least one more general itemset (child).

Distance to the 1st nearest neighbor. To compute this parameter we represent patterns as binary vectors and take into account the smallest Euclidean distance between each pattern and the remaining patterns in the pattern set. The average value for a pattern set is taken as the average distance to the 1st nearest neighbor. A set containing a lot of similar patterns will have low average values, see Figure 5 (1) for an example.

As it can be seen from Figure 6 (1), the MDL principle provides much more distinctive itemsets. Top-n concepts have a lot of similar patterns, while MDL-optimal ones are pairwise distinctive (w.r.t. Euclidean distance).

Average length of the longest paths built from partially ordered itemsets. The patterns can be partially ordered by inclusion, i.e. $B_1 \subset B_2 \subset \ldots \subset B_n$, where B_n is the most specific patters and B_1 is the most general one. We call this ordered sequences paths. If $B_n \subseteq g'$ then $B_i \subseteq g'$ is guaranteed for all $i \in [1, n-1]$. Longer paths contain more patterns describing the same objects. Thus, a long path can be considered as an indicator of redundancy. In other words, these patterns characterize the same objects at different levels of abstraction and contain only a few new details w.r.t. the nearest neighbors in the path. Short paths correspond to "flat" structures with more varied patterns. An example of comparison of two tiny pattern sets is given in Figure 5, (2).

As we see in Figure 6 (b), for ordering w.r.t. *len* (see len_fr and len_lift) the MDL priciple does not provide any benefits, while its application to *area_len_sep* and $area\_sep\_lift$, $lift\_len\_fr$ allows for more flattened structures, even more flattened than with *len*. It means that pattern mining with *area_len_sep* and $area\_sep\_lift$, $lift\_len\_fr$ can be significantly improved by the application of MDL.

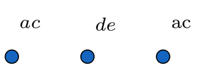
$\mathcal{S}_3$	binary representation (abcde)	nearest neighbor	Euclidean distance
bc	01100	ac	$\sqrt{2}$
de	00011	bc(ac)	2
ac	10100	bc	$\sqrt{2}$

The average distance is $(2 + 2\sqrt{2})/3$.

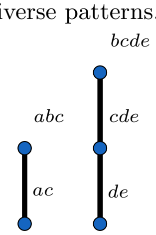
$\mathcal{S}_2$	binary representation (abcde)	nearest neighbor	Euclidean distance
bcde	01111	cde	$\sqrt{2}$
cde	00111	bcde (de)	$\sqrt{2}$
abc	11100	ac	$\sqrt{2}$
ac	10100	abc	$\sqrt{2}$
de	00011	cde	$\sqrt{2}$

The average distance is $\sqrt{2}$.

(1) Euclidean distances to the 1st nearest neighbors. The average distance for $\mathcal{S}_3$ is longer then for $\mathcal{S}_4$, thus $\mathcal{S}_3$ contains for diverse patterns.

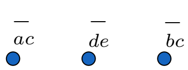


The average length of the longest paths is 1.

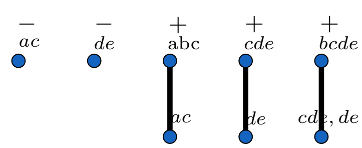


The average length of the longest paths is 2.5.

(2) The longest paths built on partially ordered patterns (by inclusion).



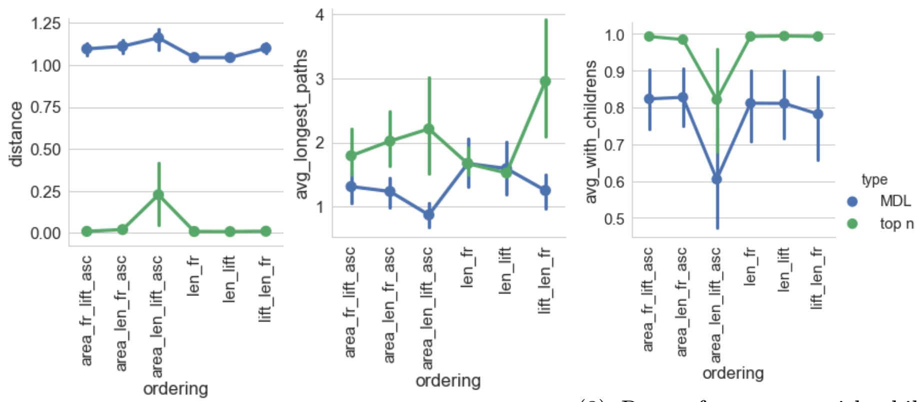
The rate of pattern with children is 0.



The rate of pattern with children is 3/5.

(3) The rate of patterns with children (i.e. more general / short patterns).

Fig. 5. Non-redundancy measures computed for patterns set given in Figure 1. The set $\mathcal{S}_3$ (column 1) is better than $\mathcal{S}_2$ (column 2) w.r.t. all the parameters: the average distance is higher, the average length of the longest paths and the rate of patterns with children are smaller.



(1) Distance to the 1NN

(2) The average path lengths

(3) Rate of patterns with children.

Fig. 6. Non-redundancy parameters: (1) the average distance to the 1st nearest neighbor for itemsets selected with MDL and top-n itemsets; (2) the average length of the longest paths computed on the chain of itemsets formed by inclusion of its attributes; (3) the average rate of itemsets with children. On the X-axis is different orderings of patterns, on the Y-axis is the values of the listed above non-redundancy parameters for MDL-optimal set (blue) and top-n (green) set of the same size.

Average number of itemsets with children (more general itemsets). This parameter characterizes the uniqueness of patterns in a set, absence of the second pattern $B_2 \subset B_1$ that characterizes the same subset as a more specific one. This parameter is related to the previous measure, but it indicates just an amount of itemsets having at least one more general itemset. An example of computing this parameter is given on Figure 5, (3).

The results of experiments (see Figure 6, (c)) show that the MDL principle selects more distinctive itemsets than top-n itemsets.

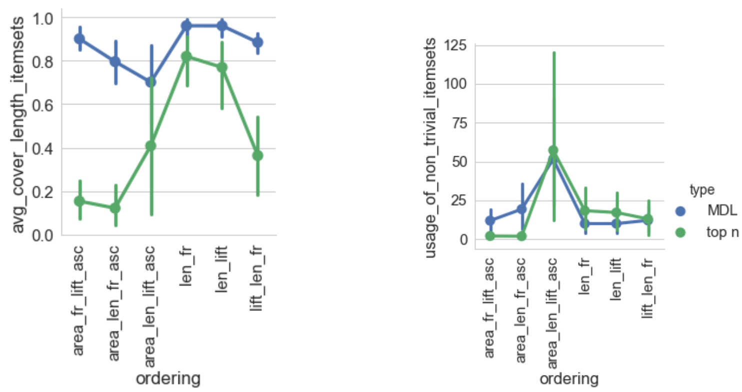
4.2 Data coverage

A subset of selected patterns can be considered as a concise representation of a dataset. Thus, it is important to know how much information is lost by compression. We measure this parameter by the rate of covered attributes. Values close to 1 correspond to the lossless compression.

The average covering rate is given in Figure 7 (1). With the same number of patterns MDL ensures better covering. For $area\_fr\_lift$, $area\_len\_fr$ and $area\_len\_lift$ MDL-optimal set covers much more data than top-n patterns.

4.3 Itemset typicality (representativeness)

In our experiments we also address typicality of patterns. In this study we measure it by the usage of patterns. To compute usage we consider the ordered



(1) Data coverage (2) Itemset typicality

Fig. 7. Pattern set parameters: (1) the average covering rate of itemsets (i.e. the rate of crosses covered by patterns); (2) the average itemset usage (reflects typicality/representativeness of patterns). On the X-axis is different orderings of patterns, on the Y-axis is the covering rate and the average itemset usage for MDL-optimal set (blue) and top-n (green) set of the same size.

patterns (in case of MDL, top patterns are those that have the shortest encoding length, for top-n they are top-patterns w.r.t. a chosen measure). The ordered patterns are used one by one to cover data. The attributes are covered only ones (disjoint covering by patterns). The number of times a patterns is used in the covering is its usage, thus the usage does not exceed the pattern frequency. For example, in Figure 2 (4), the frequency of bc is 2, but it can be used only one time to cover $(b)(c)(de)$, since in $(ac)(b)$ only b is left to cover.

It should be noted that it is not obvious which values are better. The usage serves to characterize a subset of patterns. The high values correspond to a subset of common patterns, while low values indicates that a subset contains less typical, but still interesting (w.r.t. interestingness measures) patterns.

Figure 7(2) shows the average usage for MDL-optimal and top-n patterns. The usage of MDL-optimal patterns is almost the same for different orders while the usage of top-n is dependent on ordering.

5 Conclusion

In the paper we propose a new approach to the measure-based pattern mining. It can be considered as an *"implementation of the MDL principle under constrains"* or *"embedding of background knowledge (on interestingness) into MDL"*. We took the Krimp algorithm as a basic implementation of MDL and studied a range of interestingness measures within it.

The proposed approach is a threshold-free method for the selection of a small set of patterns having desired properties. The chosen patterns are diverse and varied, they cover almost all attributes of objects.

The studied Krimp algorithm can be changed further to improve (closed) pattern mining as follows. The greedy strategy may be relaxed, i.e., overlapping patterns can be used to cover an object. Some additional mechanism may be proposed to deal with noisy data (missed values).

References

1. Aggarwal, C.C., Han, J.: Frequent pattern mining. Springer (2014)
2. Barron, A., Rissanen, J., Yu, B.: The minimum description length principle in coding and modeling. IEEE Transactions on Information Theory **44**(6), 2743–2760 (1998)
3. Belohlavek, R., Trnecka, M.: From-below approximations in boolean matrix factorization: Geometry and new algorithm. Journal of Computer and System Sciences **81**(8), 1678–1697 (2015)
4. Coenen, F.: The lucs-kdd discretised/normalised arm and carm data library. Department of Computer Science, The University of Liverpool, UK (2003), http://www.csc.liv.ac.uk/ frans/KDD/Software/LUCS_KDD_DN
5. Ganter, B., Wille, R.: Formal concept analysis: Logical foundations. Springer Verlag Berlin, RFA (1999)
6. Grünwald, P.D.: The minimum description length principle. MIT press (2007)
7. Hansen, M.H., Yu, B.: Model selection and the principle of minimum description length. Journal of the American Statistical Association **96**(454), 746–774 (2001)
8. Kuznetsov, S.O., Tatiana, M.: On interestingness measures of formal concepts. Information Sciences **442-443**, 202 – 219 (2018)
9. Li, J., Li, H., Wong, L., Pei, J., Dong, G.: Minimum description length principle: Generators are preferable to closed patterns. In: AAAI. pp. 409–414 (2006)
10. Miettinen, P., Vreeken, J.: Model order selection for boolean matrix factorization. In: Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining. pp. 51–59. ACM (2011)
11. Miettinen, P., Vreeken, J.: Mdl4bmf: Minimum description length for boolean matrix factorization. ACM Transactions on Knowledge Discovery from Data (TKDD) **8**(4), 18 (2014)
12. Siebes, A.: Mdl in pattern mining a brief introduction to krimp. In: Glodeanu, C.V., Kaytoue, M., Sacarea, C. (eds.) Formal Concept Analysis. pp. 37–43. Springer International Publishing, Cham (2014)
13. Vreeken, J., Van Leeuwen, M., Siebes, A.: Krimp: mining itemsets that compress. Data Mining and Knowledge Discovery **23**(1), 169–214 (2011)
14. Yurov, M., Ignatov, D.I.: Turning krimp into a triclustering technique on sets of attribute-condition pairs that compress. In: Polkowski, L., Yao, Y., Artiemjew, P., Ciucci, D., Liu, D., Ślezak, D., Zielosko, B. (eds.) Rough Sets. pp. 558–569. Springer International Publishing, Cham (2017)

Rectangle and Square Coverings of Tolerance Spaces and their Direct Product

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. This article is a sequel to the paper "Blocks of the Direct Product of Tolerance Relations" [7]. The square cover number of the direct product of tolerance spaces and the rectangle cover number of the direct product of formal contexts is treated. Furthermore, we compare rectangle and square covers of tolerance spaces.

Keywords: tolerance relation, formal concept analysis, direct product, factor analysis, rectangle cover, square cover.

1 Introduction

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*. An introduction to tolerance spaces together with applications can be found in [10] and [11].

For a tolerance τ on V, a non-empty subset $S \subseteq V$ induces a *square* in τ if $S \times S \subseteq \tau$. If S is maximal with respect to set inclusion, then $S \times S$ defines a *maximal square*.

The set of all maximal squares of $\mathbb{T}$ is denoted by $\mathrm{Sq}(\mathbb{T})$ and 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*, $\mathrm{sc}(\mathbb{T})$, of a tolerance space $\mathbb{T}$, as the minimal number of maximal squares necessary to cover τ.

$$\mathrm{sc}(\mathbb{T}) := \min\{k \mid \exists\, \mathcal{S} \subseteq \mathrm{Sq}(\mathbb{T}),\ \tau = \bigcup \mathcal{S},\ |\mathcal{S}| = k\}. \tag{1}$$

In [7] the direct product (defined in Section 2) of tolerance spaces was treated by means of formal concept analysis, which lead to the conjecture:

Conjecture 1. *Let $\mathbb{T}_1$ and $\mathbb{T}_2$ be tolerance spaces. For their direct product $\mathbb{T}_1 \check{\times} \mathbb{T}_2$ it holds that* $\mathrm{sc}(\mathbb{T}_1 \check{\times} \mathbb{T}_2) = \mathrm{sc}(\mathbb{T}_1) + \mathrm{sc}(\mathbb{T}_2)$.

When we analysed Conjecture 1, it turned out that it is not valid in general. Still, we will provide a sufficient condition for this conjecture to hold (Section 5). The meta framework for this will be formal concept analysis, introduced in

Section 2, together with some tools from graph theory (Section 4). Additionally, we will treat the rectangle cover number of the direct product of formal contexts in Section 3 and Section 6 provides example classes of tolerance spaces for which the square cover number and rectangle cover number are equal. Lastly, Section 7 analyses a construction principle for tolerance spaces, which is based on formal contexts.

2 Formal Concept Analysis

In this section, we will provide the definitions and facts from formal concept analysis (see [5]) that will be used in the sequel.

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

$$A^I := \{m \in M \mid \forall a \in A : (a, m) \in I\} = \bigcap_{a \in A} \{a\}^I,$$

$$B_I := \{g \in G \mid \forall b \in B : (g, b) \in I\} = \bigcap_{b \in B} \{b\}_I.$$

If $A^I = B$ and $B_I = A$, the pair (A, B) is called a *formal concept* (or in short *concept*) with *extent* A and *intent* B. 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 $(A_1, B_1) \leq (A_2, B_2) :\Leftrightarrow A_1 \subseteq A_2$. The *complementary context* is defined as $\mathbb{K}^c = (G, M, I^c) := (G, M, (G \times M) \setminus I)$ and the *dual context* as $\mathbb{K}^d := (M, G, I^{-1})$, with the *inverse relation* $I^{-1} := \{(m, g) \in M \times G \mid (g, m) \in I\}$.

Let $\dot\cup$ denote the disjoint union of sets. We define four binary operations on contexts $\mathbb{K}_1 = (G_1, M_1, I_1)$ and $\mathbb{K}_2 = (G_2, M_2, I_2)$.

The *direct product* $\mathbb{K}_1 \check{\times} \mathbb{K}_2 := (G_1 \times G_2, M_1 \times M_2, I_1 \check{\times} I_2)$ with

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

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

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

the *direct sum* $\mathbb{K}_1 \oplus \mathbb{K}_2 := (G_1 \dot\cup G_2, M_1 \dot\cup M_2, I_1 \dot\cup I_2 \dot\cup G_1 \times M_2 \dot\cup G_2 \times M_1)$,

and the *disjoint union* $\mathbb{K}_1 \dot\cup \mathbb{K}_2 := (G_1 \dot\cup G_2, M_1 \dot\cup M_2, I_1 \dot\cup I_2)$.

The two products fulfill De Morgan laws

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

the relation $I_1 \check{\times} I_2$ can be expressed as

$$I_1 \check{\times} I_2 = (G_1 \times M_1) \hat{\times} I_2 \cup I_1 \hat{\times} (G_2 \times M_2), \tag{3}$$

and we will denote the incidence relation of the direct sum by $I_1 \oplus I_2$.

A context $\mathbb{K}$ is *crossed*, if the *adjacency matrix* $\mathcal{A}_I$ of its incidence I has at least one full row and one full column. If $\mathcal{A}_I$ has at least one empty row and one empty column, we say that $\mathbb{K}$ is *co-crossed*. In case of two crossed contexts, we can express the concept lattice of the cardinal product as the direct product (in terms of Universal Algebra) of each factors concept lattice (see [3]).

$$\underline{\mathfrak{B}}(\mathbb{K}_1 \hat{\times} \mathbb{K}_2) \cong \underline{\mathfrak{B}}(\mathbb{K}_1) \times \underline{\mathfrak{B}}(\mathbb{K}_2). \tag{4}$$

The concept lattice of the direct sum is isomorphic to the direct product of each components concept lattice too[1].

$$\underline{\mathfrak{B}}(\mathbb{K}_1 \oplus \mathbb{K}_2) \cong \underline{\mathfrak{B}}(\mathbb{K}_1) \times \underline{\mathfrak{B}}(\mathbb{K}_2). \tag{5}$$

Let $\mathbb{P} = (P, \leq_{\mathbb{P}}, 0_{\mathbb{P}}, 1_{\mathbb{P}})$ and $\mathbb{L} = (L, \leq_{\mathbb{L}}, 0_{\mathbb{L}}, 1_{\mathbb{L}})$ be bounded posets such that $P \cap L = \emptyset$. The poset $\mathbb{S} = (S, \leq, 0, 1)$, with $P^* := P \backslash \{0_{\mathbb{P}}, 1_{\mathbb{P}}\}$, $L^* := L \backslash \{0_{\mathbb{L}}, 1_{\mathbb{L}}\}$, $S^* := P^* \cup L^*$, $S := S^* \cup \{0, 1\}$ and $\leq := \leq_{\mathbb{P}} \cup \leq_{\mathbb{L}} \cup \{0\} \times S \cup S \times \{1\}$, is called the *horizontal sum* of $(\mathbb{P}, \mathbb{L})$ and is denoted by $\mathbb{P} \hat{+} \mathbb{L} := \mathbb{S}$.

For the disjoint union of two contexts, the resultant concept lattice is the horizontal sum of each components concept lattice.

$$\underline{\mathfrak{B}}(\mathbb{K}_1 \dot{\cup} \mathbb{K}_2) \cong \underline{\mathfrak{B}}(\mathbb{K}_1) \hat{+} \underline{\mathfrak{B}}(\mathbb{K}_2). \tag{6}$$

Next, since a concept (A, B) with non-empty sets A and B induces a *maximal rectangle* $A \times B$ in I, we define the *rectangle cover number* (see also [12]), $\mathrm{rc}(\mathbb{K})$, of a context $\mathbb{K}$ as

$$\mathrm{rc}(\mathbb{K}) := \min\{k \mid \exists\, \mathcal{F} \subseteq \mathfrak{B}(\mathbb{K}),\ I = \bigcup_{(A,B) \in \mathcal{F}} A \times B,\ |\mathcal{F}| = k\}. \tag{7}$$

The *Boolean rank*, $\mathrm{r_B}(C)$, 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 (see [1]). In [1] it is implicitly shown that:

$$\mathrm{rc}(\mathbb{K}) = \mathrm{r_B}(\mathcal{A}_I). \tag{8}$$

[1] The condition to be crossed is not necessary for Identity 5.

Lastly, we recall some aspects of dimension theory. For a concept lattice $\underline{\mathfrak{B}}(\mathbb{K})$, its 2-*dimension*, $\dim_2(\underline{\mathfrak{B}}(\mathbb{K}))$, is the smallest number of chains of cardinality 2 in whose direct product it can be order-embedded. Since the n-fold direct product of chains of cardinality 2 is isomorphic to the powerset lattice of the n-element set $\underline{n}$, there exists $\varphi : \underline{\mathfrak{B}}(\mathbb{K}) \to \mathfrak{P}(\underline{n})$ with $(A, B) \leq (C, D) \Longleftrightarrow \varphi(A, B) \leq \varphi(C, D)$.

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 $(h, m) \in F$. This is equivalent to $\underline{\mathfrak{B}}(G, M, F)$ being a chain. The *length* l of F is defined as $l(F) = |\mathfrak{B}(G, M, F)| - 1$. For a context $\mathbb{K}$ its *Ferrers 2-dimension*, $\mathrm{fdim}_2(\mathbb{K})$, is the smallest number of Ferrers relations F_t, $t \in T$ with $l(F_t) < 2$, so that $I = \bigcap_{t \in T} F_t$.

The above defined dimensions are equal and are related to the rectangle cover number via the complementary context, that is:

$$\mathrm{rc}(\mathbb{K}) = \mathrm{fdim}_2(\mathbb{K}^c) = \dim_2(\underline{\mathfrak{B}}(\mathbb{K}^c)). \tag{9}$$

3 The Rectangle Cover Number of the Direct Product of Formal Contexts

In this section, we will treat the rectangle cover number of the direct product of two contexts $\mathbb{K}_1$ and $\mathbb{K}_2$. From Identity 3, it follows that $\mathrm{rc}(\mathbb{K}_1 \check{\times} \mathbb{K}_2) \leq \mathrm{rc}(\mathbb{K}_1) + \mathrm{rc}(\mathbb{K}_2)$. We will provide a sufficient condition for equality. Therefore, we will need a proposition about the Ferrers 2-dimension of the direct sum of two contexts. This proposition and its use in Theorem 1 is inspired by [14].

Proposition 1. *For the direct sum of two contexts* $\mathbb{K}_1 = (G_1, M_1, I_1)$ *and* $\mathbb{K}_2 = (G_2, M_2, I_2)$, *it holds that* $\mathrm{fdim}_2(\mathbb{K}_1 \oplus \mathbb{K}_2) = \mathrm{fdim}_2(\mathbb{K}_1) + \mathrm{fdim}_2(\mathbb{K}_2)$.

Proof. The claim follows from Identity 9 with interchanged roles of $\mathbb{K}$ and $\mathbb{K}^c$, and the structure of the relation $(I_1 \oplus I_2)^c$ depicted in Figure 1.

Fig. 1. The relation $I_1 \oplus I_2$ and $(I_1 \oplus I_2)^c$ of the direct sum of $\mathbb{K}_1$ and $\mathbb{K}_2$.

$I_1 \oplus I_2$	M_1	M_2
G_1	I_1	$G_1 \times M_2$
G_2	$G_2 \times M_1$	I_2

$(I_1 \oplus I_2)^c$	M_1	M_2
G_1	I_1^c	$\emptyset$
G_2	$\emptyset$	I_2^c

Theorem 1. *Let* $\mathbb{K}_1$ *and* $\mathbb{K}_2$ *be co-crossed contexts. For the rectangle cover number of their direct product it holds that:*

$$\mathrm{rc}(\mathbb{K}_1 \check{\times} \mathbb{K}_2) = \mathrm{rc}(\mathbb{K}_1) + \mathrm{rc}(\mathbb{K}_2).$$

Proof. First, we notice that $\mathbb{K}_1^c$ and $\mathbb{K}_2^c$ are crossed contexts. From Proposition 1, and Identity 2, 4, 5 and 9, we conclude that

$$\begin{aligned}
\mathrm{rc}(\mathbb{K}_1 \check{\times} \mathbb{K}_2) &= \mathrm{fdim}_2((\mathbb{K}_1 \check{\times} \mathbb{K}_2)^c) \\
&= \mathrm{fdim}_2(\mathbb{K}_1^c \hat{\times} \mathbb{K}_2^c) \\
&= \dim_2(\underline{\mathfrak{B}}(\mathbb{K}_1^c \hat{\times} \mathbb{K}_2^c)) \\
&= \dim_2(\underline{\mathfrak{B}}(\mathbb{K}_1^c) \times \underline{\mathfrak{B}}(\mathbb{K}_2^c)) \\
&= \dim_2(\underline{\mathfrak{B}}(\mathbb{K}_1^c \oplus \mathbb{K}_2^c)) \\
&= \mathrm{fdim}_2(\mathbb{K}_1^c \oplus \mathbb{K}_2^c) \\
&= \mathrm{fdim}_2(\mathbb{K}_1^c) + \mathrm{fdim}_2(\mathbb{K}_2^c) \\
&= \mathrm{rc}(\mathbb{K}_1) + \mathrm{rc}(\mathbb{K}_2).
\end{aligned}$$

Remark 1. In case that the hypothesis for both factors to be co-crossed does not hold, the simplest example to consider would be $\mathbb{I} := (\{g\}, \{m\}, \{g\} \times \{m\})$. It is crossed and for any non-empty context $\mathbb{K}$, it holds that $\mathrm{rc}(\mathbb{K} \check{\times} \mathbb{I}) = 1 < \mathrm{rc}(\mathbb{K}) + 1 = \mathrm{rc}(\mathbb{K}) + \mathrm{rc}(\mathbb{I})$.

Without providing a formal definition, we restate Theorem 1 for Boolean matrices. Since in this case the term direct product would not be appropriate, we will use the established notion *Cartesian sum* from graph theory (see [9]). Identity 8 implies:

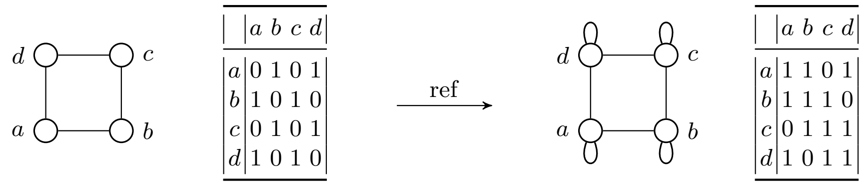
Corollary 1. *Let A_1 and A_2 be Boolean matrices with at least one empty row and one empty column. For the Boolean rank of their Cartesian sum it holds that* $\mathrm{r_B}(A_1 \check{\times} A_2) = \mathrm{r_B}(A_1) + \mathrm{r_B}(A_2)$.

4 Edge Clique Covers of Simple Graphs

In order to analyse the square cover number of tolerance spaces, we will use some results from graph theory which will be introduced in this section. A *graph* is considered as a relational structure $\mathbb{G} = (V, E)$ with *vertex set* V and an irreflexive, symmetric binary relation $E \subseteq V \times V$. Let E^{ref} denote the *reflexive closure* of E. The reflexive closure of $\mathbb{G}$ is defined as $\mathbb{G}^{\mathrm{ref}} := (V, E^{\mathrm{ref}})$. It follows that the graph $\mathbb{G}^{\mathrm{ref}}$ defines a tolerance space. On the contrary, let $\mathbb{T}$ be a tolerance space, then $\mathbb{G}_{\mathbb{T}}$ denotes the underlying graph.

As usual, K_n denotes the *complete graph* with n vertices and $K_{m,n}$ the *complete bipartite graph* with disjoint vertex sets A and B, such that $|A| = m$ and $|B| = n$. An *n-clique* (or just *clique*) of $\mathbb{G}$ is a complete subgraph $K_n \leq \mathbb{G}$. Every clique of a graph $\mathbb{G}$ induces a clique in the reflexive closure $\mathbb{G}^{\mathrm{ref}}$ and every isolated vertex of $\mathbb{G}$ induces a 1-clique in $\mathbb{G}^{\mathrm{ref}}$. The difference between cliques and reflexive cliques is that the latter one can be identified with a formal concept and especially with a maximal square in E^{ref} in the sense of tolerance relations. Figure 2 provides an example.

Fig. 2. The reflexive closure of a 4 cycle is depicted.



	a	b	c	d
a	0	1	0	1
b	1	0	1	0
c	0	1	0	1
d	1	0	1	0

	a	b	c	d
a	1	1	0	1
b	1	1	1	0
c	0	1	1	1
d	1	0	1	1

The *edge clique cover number* of a graph $\mathbb{G}$, $\theta_e(\mathbb{G})$, is the smallest number of cliques such that their edges cover the edges of $\mathbb{G}$. For a graph with n vertices, we have that $\theta_e(\mathbb{G}) \leq \lfloor n^2/4 \rfloor$, in which equality holds for the graph $K_{\lfloor n/2 \rfloor, \lceil n/2 \rceil}$ (see [13]).

$$\theta_e(K_{\lfloor n/2 \rfloor, \lceil n/2 \rceil}) = \lfloor n^2/4 \rfloor. \tag{10}$$

The following proposition relates θ_e to the square covering number.

Proposition 2. *Let* $\mathbb{G}$ *be a graph and* $\mathbb{T}$ *be a tolerance space. It holds that* $\theta_e(\mathbb{G}) = \mathrm{sc}(\mathbb{G}^{\mathrm{ref}})$ *and* $\mathrm{sc}(\mathbb{T}) = \theta_e(\mathbb{G}_{\mathbb{T}})$*.*

Another graph parameter related to cliques is the *vertex clique cover number*, $\theta_v(\mathbb{G})$, that is the smallest number of cliques, such that their vertices cover all vertices of $\mathbb{G}$.

Lastly, we describe the concept lattice of a graph $\mathbb{G}$ and the relationship between graph homomorphisms (edge preserving maps) and certain maps between concept lattices of graphs. In [15], concept lattices of graphs are studied under the name *neighborhood ortholattice*. It is shown that $\underline{\mathfrak{B}}(\mathbb{G})$ is a *complete ortholattice*, that is a complete bounded lattice with an involutory antiautomorphism c, such that $x \leq c(x)$ implies $x = 0$. We define an abstract *orthogonality relation* through $x \perp y :\Longleftrightarrow x \leq c(y)$.

An *orthomap* between complete ortholattices preserves order and orthogonality, and maps only the bottom element of the domain lattice to the bottom element of the codomain lattice. The next theorem relates graph homomorphisms to orthomaps.

Theorem 2 ([15]). *A graph homomorphism from* $\mathbb{G}_1$ *to* $\mathbb{G}_2$ *exists if and only if there exists an orthomap from* $\underline{\mathfrak{B}}(\mathbb{G}_1)$ *to* $\underline{\mathfrak{B}}(\mathbb{G}_2)$*.*

Furthermore, it is shown in [15] that the concept lattice of K_n is isomorphic to the powerset lattice of an n-element set: $\underline{\mathfrak{B}}(K_n) \cong \mathfrak{P}(\underline{n})$.

5 The Square Cover Number of the Direct Product of Tolerance Spaces

This section treats Conjecture 1, *i.e.*, for tolerance spaces $\mathbb{T}_1$ and $\mathbb{T}_2$:

$$\mathrm{sc}(\mathbb{T}_1 \check{\times} \mathbb{T}_2) = \mathrm{sc}(\mathbb{T}_1) + \mathrm{sc}(\mathbb{T}_2).$$

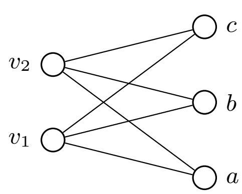
First, similar to Remark 1, we see that Conjecture 1 is false for an arbitrary tolerance space $\mathbb{T}_1$, and $\mathbb{T}_2 = (\{v\}, \{v\}, \{v\} \times \{v\})$. Second, due to Identity 3, it always holds that $\mathrm{sc}(\mathbb{T}_1 \check{\times} \mathbb{T}_2) \leq \mathrm{sc}(\mathbb{T}_1) + \mathrm{sc}(\mathbb{T}_2)$. The question for which tolerance spaces equality holds remains. An analogue to Theorem 1 can not exist, since tolerance spaces, due to their reflexivity, can not be co-crossed.

In order to be able to make use of the the rectangle cover number, we will relate the square cover number, $\mathrm{sc}(\mathbb{T})$, to the rectangle cover number, $\mathrm{rc}(\mathbb{T})$. Since every square is also a rectangle, it holds that for any tolerance space, the rectangle cover number is less or equal to the square cover number.

$$\mathrm{rc}(\mathbb{T}) \leq \mathrm{sc}(\mathbb{T}). \tag{11}$$

But, the reverse inequality is wrong in general. To see this, we notice that $\mathrm{rc}(\mathbb{T}) = \mathrm{r_B}(\mathcal{A}_\tau) \leq |V|$ (see [1]). Consequently, a tolerance space with square cover number larger than $|V|$ would provide a counter example. From Identity 10 and Proposition 2, we conclude that $6 = \mathrm{sc}(K_{2,3}^{\mathrm{ref}}) > \mathrm{rc}(K_{2,3}^{\mathrm{ref}}) = 5$ (see Fig. 3).

Fig. 3. The graph $K_{2,3}$ and the adjacency matrix of $K_{2,3}^{\mathrm{ref}}$.



	v_1	v_2	a	b	c
v_1	1	0	1	1	1
v_2	0	1	1	1	1
a	1	1	1	0	0
b	1	1	0	1	0
c	1	1	0	0	1

This motivates the following definition.

Definition 1. *We will say that a tolerance space* $\mathbb{T}$ *has the* balanced covering property *(in short* BCP*) if* $\mathrm{sc}(\mathbb{T}) = \mathrm{rc}(\mathbb{T})$*.*

This definition leads immediately to:

Theorem 3. *Let* $\mathbb{T}_1$ *and* $\mathbb{T}_2$ *be tolerance spaces with the BCP, such that* $\mathrm{rc}(\mathbb{T}_1 \check{\times} \mathbb{T}_2) = \mathrm{rc}(\mathbb{T}_1) + \mathrm{rc}(\mathbb{T}_2)$*. It follows that* $\mathrm{sc}(\mathbb{T}_1 \check{\times} \mathbb{T}_2) = \mathrm{sc}(\mathbb{T}_1) + \mathrm{sc}(\mathbb{T}_2)$*.*

Proof. It always holds that $\mathrm{sc}(\mathbb{T}_1 \check{\times} \mathbb{T}_2) \leq \mathrm{sc}(\mathbb{T}_1) + \mathrm{sc}(\mathbb{T}_2)$. From the BCP of $\mathbb{T}_1$ and $\mathbb{T}_2$, and Inequality 11, we conclude the reverse direction

$$\mathrm{sc}(\mathbb{T}_1) + \mathrm{sc}(\mathbb{T}_2) = \mathrm{rc}(\mathbb{T}_1) + \mathrm{rc}(\mathbb{T}_2) = \mathrm{rc}(\mathbb{T}_1 \check{\times} \mathbb{T}_2) \leq \mathrm{sc}(\mathbb{T}_1 \check{\times} \mathbb{T}_2).$$

6 Tolerance Spaces with the balanced covering property

In this section, we will provide examples of tolerance spaces which have the BCP.

Example 1. *The following is inspired by [8]. A covering $\mathcal{H} \subseteq \mathfrak{P}(V)$ of V is* irredundant *if $\mathcal{H}\backslash\{X\}$ is not a covering of V for any $X \in \mathcal{H}$. An irredundant covering induces the tolerance $\tau_{\mathcal{H}} := \bigcup\{X \times X \mid X \in \mathcal{H}\}$ with underlying tolerance space $\mathbb{T}_{\mathcal{H}} := (V, \tau_{\mathcal{H}})$. It follows that* $\mathrm{sc}(\mathbb{T}_{\mathcal{H}}) \leq |\mathcal{H}|$*. Since, $\mathcal{H}$ is an irredundant covering, for every $X \in \mathcal{H}$ there exists $v \in X$, such that $X \times X$ is the only maximal square which is covering (v, v). Hence, the squares $X \times X$ with $X \in \mathcal{H}$ are* mandatory *(see [1] for* mandatory factors *in the sense of factor analysis) for every covering of $\tau_{\mathcal{H}}$, which implies* $\mathrm{sc}(\mathbb{T}_{\mathcal{H}}) = \mathrm{rc}(\mathbb{T}_{\mathcal{H}}) = |\mathcal{H}|$.

Furthermore, note that tolerances induced by irredundant coverings can be considered as the reflexive closure of graphs $\mathbb{G}$ with $\theta_e(\mathbb{G}) = \theta_v(\mathbb{G})$ (see [2] Theorem 1) and that equivalence relations are a special case of such tolerances.

Next, we will describe the structure of the graphs $\mathbb{G} = (V, E)$ whose underlying relation E is the complement of a tolerance induced by an irredundant covering.

Theorem 4. *Let $\mathbb{T} = (V, \tau)$ be a tolerance space and $\mathbb{G} = (V, E)$ the graph defined through $\mathbb{G} := \mathbb{T}^c$. If the tolerance τ is induced by an irredundant covering $\mathcal{H} \subseteq \mathfrak{P}(V)$ with $|\mathcal{H}| = n$, then $\mathbb{G}$ is a connected graph with K_n as a retract*[2].

Proof. In [8] it is shown that $\underline{\mathfrak{B}}(\mathbb{G})$ is an atomistic boolean lattice if τ is induced by an irredundant covering. Since we only consider finite tolerance spaces, this means that $\underline{\mathfrak{B}}(\mathbb{G})$ is isomorphic to a powerset lattice. We denote the isomorphism by Φ and show that it is an orthomap.

Since it is an isomorphism it preserves order and only the bottom element of the domain lattice is mapped to the bottom element of the codomain lattice. Consequently, just the preservation of orthogonality is left to show:

$$x \perp y \Rightarrow x \leq c(y) \Rightarrow \Phi(x) \leq \Phi(c(y)) = c(\Phi(y))^3 \Rightarrow \Phi(x) \perp \Phi(y).$$

The same holds for the inverse Φ^{-1} so that we have orthomaps $\underline{\mathfrak{B}}(\mathbb{G}) \to \mathfrak{P}(\underline{n})$ and $\mathfrak{P}(\underline{n}) \to \underline{\mathfrak{B}}(\mathbb{G})$. Theorem 2 implies the existence of two graph homomorphisms $\varphi_1 : \mathbb{G} \to K_n$ and $\varphi_2 : K_n \to \mathbb{G}$. Since φ_2 must be an embedding, it can be defined such that $\varphi_1 \circ \varphi_2 = \mathrm{id}_{\mathbb{G}}$ holds.

Lastly, we notice that $\mathbb{G}$ must be connected. Otherwise $\underline{\mathfrak{B}}(\mathbb{G})$ would be equal to the horizontal sum of the connected components of $\mathbb{G}$ (see Identity 6). But $\underline{\mathfrak{B}}(\mathbb{G}) \cong \mathfrak{P}(\underline{n})$ implies that for $n \geq 3$ and $n = 1$, the concept lattice $\underline{\mathfrak{B}}(\mathbb{G})$ can not

[2] A graph G is a *retract* of H if there exist graph homomorphisms $\varphi : G \to H$ and $\psi : H \to G$ such that the composite $\psi \circ \varphi$ is the identity on G.

[3] The last equality is a consequence of the fact that the isomorphic image of an orthocomplemented lattice is again an orthocomplemented lattice. Just define $c(\Phi(x)) := \Phi(c(x))$. Since the powerset lattice has a unique orthocomplementation, this is the only possible choise for c.

be horizontally decomposed. For $n = 2$, the graph $\mathbb{G}$ must have two connected components such that their concept lattice is a chain. This is a contradiction, since the underlying relation of a graph can not be a Ferrers relation.

Example 2. *In this example we generalize the construction of* $(K_{\lfloor n/2\rfloor,\lceil n/2\rceil})^{\mathrm{ref}}$ *(see Section 4 and Figure 3). For this purpose let* $\mathbb{K}$ *be a context. We consider* $\mathbb{T} := (\mathbb{K}\dot\cup\mathbb{K}^d)^{\mathrm{ref}}$*, the reflexive closure of the union of* $\mathbb{K}$ *and* $\mathbb{K}^d$*. This construction yields the reflexive closure of a bipartite graph with disjoint vertex sets* G *and* M*, such that we draw a line from* $g \in G$ *to* $m \in M$ *whenever* gIm *holds. It follows that every element of* I *induces a maximal clique in this bipartite graph and hence a maximal square in* $(I \dot\cup I^{-1})^{\mathrm{ref}}$ *(Fig. 4). In [6] the concepts of* $\underline{\mathfrak{B}}(\mathbb{T})$ *are*

Fig. 4. The reflexive closure of $I \dot\cup I^{-1}$, where E_X denotes the identity on X .

$(I \dot\cup I^{-1})^{\mathrm{ref}}$:	G	M
G	E_G	I
M	I^{-1}	E_M

characterized. Let $\{a\}, A \subseteq G$ *and* $\{b\}, B \subseteq M$*. The following types of concepts can occur. First, concepts which represent a row or column in* $(I \dot\cup I^{-1})^{\mathrm{ref}}$,

$$(\{a\}, \{a\} \cup A^I),\ (\{b\}, B_I \cup \{b\}),\ (\{a\} \cup B, \{a\}),\ (A \cup \{b\}, \{b\}),$$

and second concepts from $\underline{\mathfrak{B}}(\mathbb{K})$*, that is* (A, A^I), (B, B_I)*, as well as the above mentioned squares* $(\{a\} \cup \{b\}, \{a\} \cup \{b\})$*. It follows that for* $|G| + |M| < |I|$*, we have that* $\mathrm{rc}(\mathbb{T}) = |G| + |M| < \mathrm{sc}(\mathbb{T}) = |I|$*. In the next step we remove elements from* I *until* $|G| + |M| = |I|$*, which gives us* $\mathrm{rc}(\mathbb{T}) = \mathrm{sc}(\mathbb{T}) \leq |G| + |M|$*. If* $|G| + |M| > |I|$*, it still holds that* $\mathrm{rc}(\mathbb{T}) = \mathrm{sc}(\mathbb{T})$.

Finally, we notice that a graph $\mathbb{G}$ *with* $\mathbb{G}^c = (\mathbb{K} \dot\cup \mathbb{K}^d)^{\mathrm{ref}}$ *consists of complete graphs* $K_{|G|}$ *and* $K_{|M|}$*, such that their vertices are symmetrically connected through the context* $\mathbb{K}^c$*.*

Example 3. *A further example is the* symmetrization $\mathbb{K}^s$ *of a context* $\mathbb{K}$ *(see [7]). It is defined as* $\mathbb{K}^s := \mathbb{K} \oplus \mathbb{K}^d = (G \dot\cup M, G \dot\cup M, I \cup I^{-1} \cup G \times G \cup M \times M)$ *(Fig. 5). Every concept* (A, B) *of* $\mathbb{K}$ *induces a maximal square* $(A \cup B) \times (A \cup B)$*.*

More generally, every concept of $\mathbb{K}^s$ *has the form* $(A \cup D, B \cup C)$*, in which* (A, B) *and* (C, D) *are concepts of* $\mathbb{K}$*. Hence, a minimal rectangle cover of* $\mathbb{K}$ *induces a set of maximal squares which cover* I *and* I^{-1}*, but* $G \times G$ *and* $M \times M$ *may not be covered. It follows that* $\mathrm{rc}(\mathbb{K}) \leq \mathrm{rc}(\mathbb{K}^s) \leq \mathrm{rc}(\mathbb{K}) + 2$ *and that* $\mathrm{rc}(\mathbb{K}) \leq \mathrm{sc}(\mathbb{K}^s) \leq \mathrm{rc}(\mathbb{K}) + 2$*. If* $\mathrm{sc}(\mathbb{K}^s) = \mathrm{rc}(\mathbb{K})$*, the BCP* $\mathrm{sc}(\mathbb{K}^s) = \mathrm{rc}(\mathbb{K}^s)$ *follows from Inequality 11.*

Fig. 5. The symmetrization of $\mathbb{K}$.

$\mathbb{K}^s$	G	M
G	$G \times G$	I
M	I^{-1}	$M \times M$

A graph $\mathbb{G}$ with $\mathbb{G}^c = \mathbb{K} \oplus \mathbb{K}^d$ consists of two empty graphs on G and M, such that their vertices are symmetrically connected via $\mathbb{K}^c$

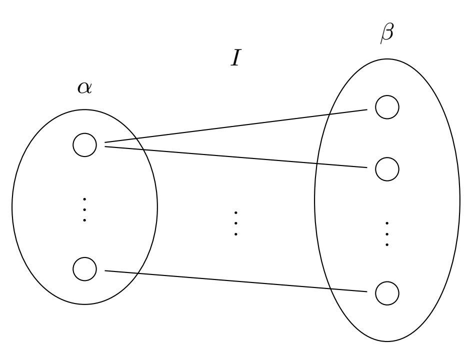
7 Construction of Tolerance Spaces

In this section, we will analyse a construction principle for tolerance spaces which is based on formal contexts. Example 2 and 3 suggest that we consider a triple $(\mathbb{A}, \mathbb{K}, \mathbb{B})$ with tolerance spaces $\mathbb{A} = (G, G, \alpha)$, $\mathbb{B} = (M, M, \beta)$ and a context $\mathbb{K} = (G, M, I)$. That triple defines the tolerance space $\mathbb{T} := (G \cup M, I \cup I^{-1} \cup \alpha \cup \beta)$ (Fig. 6).

Fig. 6. The triple $(\mathbb{A}, \mathbb{K}, \mathbb{B})$ defines the tolerance space $\mathbb{T}$.

τ :	G	M
G	α	I
M	I^{-1}	β

Fig. 7. The bipartite structure of the triple $(\mathbb{A}, \mathbb{K}, \mathbb{B})$.



The interaction of α, I and β determines the structure of the tolerance space. This fact can be interpreted as a bipartite graph defined through I, such that α and β are tolerance relations on the disjoint vertex sets (Fig. 7). In the context of clique partitions of graphs, this was already observed in [4].

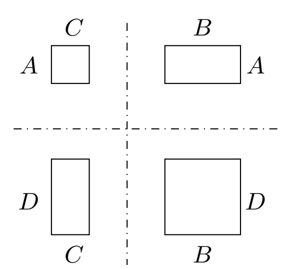
If I is the empty relation, then $\mathrm{rc}(\mathbb{T}) = \mathrm{rc}(\mathbb{A})+\mathrm{rc}(\mathbb{B})$ and $\mathrm{sc}(\mathbb{T}) = \mathrm{sc}(\mathbb{A})+\mathrm{sc}(\mathbb{B})$. For $||G| - |M|| \leq 1$ and $I = G \times M$, as well as α, β equal to the identity relation, the square cover number $\mathrm{sc}(\mathbb{T})$ is maximal. In this case, an increase of the elements of α and β can only reduce the square cover number. Generally, if I has many edges, more edges in α and β are necessary for a small square cover number, because one has to connect the edges of I into just a few maximal squares.

For arbitrary I, α and β, we state the following theorem.

Theorem 5. *Let $(\mathbb{A}, \mathbb{K}, \mathbb{B})$ be defined as above. For $A, C \subseteq G$ and $B, D \subseteq M$, let $A \times B$ and $C \times D$ be subsets of I. If $A \times C \subseteq \alpha$ and $D \times B \subseteq \beta$, then $(A \cup D) \times (B \cup C)$ is a rectangle of the underlying tolerance space. This rectangle is maximal if A, B, C and D are maximal with respect to the above stated inclusions. For $A = C$ and $B = D$ a (maximal) square is induced.*

Proof. *If $C \times D \subseteq I$, then $D \times C \subseteq I^{-1}$. The rest of the proof is graphical (Fig. 8).*

Fig. 8. The structure of maximal rectangles.



Note that in order to induce a maximal rectangle, neither $A \times B$ and $C \times D$ have to be maximal in I, nor $A \times C$ and $D \times B$ have to be maximal in α and β.

Corollary 2. *Let $\mathbb{T} = (\mathbb{A}, \mathbb{K}, \mathbb{B})$ be defined as above. A concept $(A, B) \in \mathfrak{B}(\mathbb{K})$ induces a maximal rectangle in $\mathbb{T}$ if there exists no rectangle $C \times D \subseteq I$ such that $A \times C \subseteq \alpha$ and $D \times B \subseteq \beta$. Furthermore, $(A, C) \in \mathfrak{B}(\mathbb{A})$ and $(D, B) \in \mathfrak{B}(\mathbb{B})$ induce a maximal rectangle in $\mathbb{T}$ if there exist no rectangles $A \times B \subseteq I$ and $C \times D \subseteq I$.*

8 Conclusion

This paper analysed rectangle covers of the direct product of formal contexts. If the contexts are co-crossed, then the rectangle cover number of the direct product is equal the sum of each factors' rectangle cover number.

In the next step, we treated the square cover number of the direct product of tolerance spaces. If each factor has the balanced covering property (BCP), which means that its square cover number is equal to its rectangle cover number, then additivity of the rectangle cover number with respect to the direct product transfers to the square cover number.

Lastly, we provided a variety of examples for tolerance spaces which have the BCP, analysed the corresponding graphs and introduced a construction principle for tolerance spaces based on formal contexts.

Acknowledgments. Finally, we want to express our thanks to the anonymous referees for their suggestions to improve our paper.

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., Ravindra, G.: Line-clique cover number of a graph. Indian Nat. Sci. Acad. Proc. **41**, 289–293 (1975)
3. Deiters, K., Erné, M.: Sums, products and negations of contexts and complete lattices. Algebra Universalis **60**, 469–496 (2009)
4. Erdös, P., Faudree, R., Ordman, E.T.: Clique partitions and clique coverings. Discrete Mathematics **72**, 93–101 (1988)
5. Ganter, B., Wille, R.: Formal Concept Analysis: Mathematical Foundations. Springer-Verlag New York, Inc. (1997)
6. Garanina, N.O., Grebeneva, J.V., Shilov, N.V.: Towards description logic on concept lattices. Proc. of the Tenth Int. Conf. CLA pp. 287–292 (2013)
7. Jäkel, C., Schmidt, S.: Blocks of the direct product of tolerance relations. Proceedings of the ISMIS 2017 pp. 587–596 (2017)
8. Järvinen, J., Radeleczki, S.: Rough sets determined by tolerances. International Journal of Approximate Reasoning **55**, 1419–1438 (2014)
9. Liu, D.D.F., Zhu, X.: Rough sets determined by tolerances. Discrete Mathematics **308**, 5928–5936 (2008)
10. Peters, J.F., Wasilewski, P.: Tolerance spaces: Origins, theoretical aspects and applications. Information Sciences **195**, 211–225 (2012)
11. Pogonowski, J.: Introduction to Graph Theory. University Press, Institute of Linguistics, Adam Mickiewicz University (1981)
12. Pourmoradnasseri, M., Theis, D.O.: The rectangle covering number of random boolean matrices. The Electronic J. of Combinatorics **24** (2017)
13. Pullman, N.J.: Clique coverings of graphs - a survey. Casse L.R.A. Combinatorial Mathematics X, Lecture Notes in Mathematics pp. 72–85 (1983)
14. Reuter, K.: On the dimension of the cartesian product of relations and orders. Order **6**, 277–293 (1989)
15. Walker, J.W.: From graphs to ortholattices and equivariant maps. Journal of Combinatorial Theory **B 35**, 171–192 (1983)

The theory and practice of coupling formal concept analysis to relational databases

Jens Kötters and Peter W. Eklund†

†School of Information Technology
Deakin University, Geelong, Australia

Abstract. In Formal Concept Analysis, a many-valued context is a collection of objects described by attributes that take on more than binary values, such as age (as integers or ranges of integer values) or color (a list or even a hierarchy of color combinations). Conceptual scaling is the process by which such a many-valued context is transformed into a formal context, by associating a concept lattice with the many-valued context. A many-valued context can be compared to a single table in a relational database populated with multiple rows and non-binary values. A generalization of conceptual scaling as a relational database as a whole should take into account the relations between objects, as expressed by means of foreign keys. Previous approaches to scaling a relational database (e.g. relational scaling) take such relations into account, but either do not maintain a separation between objects and values, which is characteristic for the unary case, or result in unary contexts only. In the approach presented in this paper, the use of n-ary scales is suggested, whereby a relational database is transformed into a family of n-ary contexts (a so called power context family). This paper describes the fundamentals of a Web application that allows connection to a relational database, its scaling interactively into a power context family, and navigation within that context family.

Keywords: Formal concept analysis, relational databases, conjunctive queries, data navigation, power context families, conceptual scaling.

1 Background and Motivation

In a previous paper [12], a variant of Formal Concept Analysis was introduced that uses conjunctive queries as concept intents, and resulting tables as concept extents. The resulting complete lattice of these conjunctive-query/table pairs[1] is a mathematical model of the information space over a relational database (accessible through conjunctive querying). Conjunctive queries correspond to a subset of logical formulas over a relational signature Σ (i.e. Σ is the query vocabulary), and thus have interpretations in a relational structure, which represents the database.

[1] The actual concept lattices have been defined as certain sub-lattices $\mathfrak{C}[\{x_1, \ldots, x_n\}]$.

When representing a database by a relational structure, we have to decide whether the carrier set should consist of *table entries* or *table rows*. The importance of this decision is that it determines what the query variables represent; queries will then either be formulated in the *domain calculus* or in the *tuple calculus*(cf. [1, p.74]). The domain calculus is a natural choice if all database tables represent relations (not just technically, but also conceptually) between objects. We might then employ a simple one-to-one correspondence between databases and relational structures: the n-column tables are precisely the n-ary relations in the relational structure, and the query vocabulary Σ is the set of table names, which are used as n-ary predicates.

In practice, databases are centered around object tables in the ORM-style, and the above modeling option does not reflect how users conceptualize database content. Because objects are represented by table rows, these should constitute the carrier set, so that the relational structure provides interpretations for the tuple calculus (which is also used by SQL [1, p.74]). But, unlike in the hypothetical case above, there is no immediate suitable choice of relations. While the signature Σ should express conditions in a WHERE-clause, meaningless comparisons (e.g. $t.age = t.shoe\_size$) should be eliminated in the formation of concepts.

In this paper, we propose a method to build meaningful query vocabulary around a relational database with reasonable effort. To this end, we utilize an alternative formalization [13] of the conjunctive-query lattice model [12] consistent with Wille's concept graphs [18], work which relates FCA to Sowa's Conceptual Graphs [17]. In particular, the relational database is represented by a power context family [18], i.e. by a sequence of formal contexts.

Power context families and relational structures correspond to each other in an almost trivial way: the columns of the n-th context correspond to n-ary relations, i.e. its attributes are the n-ary symbols of Σ. But the change of formalism encourages to think about databases in terms of conceptual scaling [5, Section 1.3]. Conceptual scaling is a method of deriving a formal context from a *many-valued context*, which can be seen as a database table with value columns only (no foreign keys). Huchard et. al. [10] have presented a way to obtain a context family from a database (a set of many-valued contexts, together with binary inter-object relations), but only unary contexts (where attributes represent object properties) are obtained. The idea of scaling a database into a power context family has been formulated under the title of *relational scaling* [14, 8], but the domain calculus has been used, which may have been the reason for a conflation of objects and values, in so doing destroying the analogy to conceptual scaling. Our scaling approach adheres to the analogy by maintaining a clear separation between objects and values, which in turn leads to generic and reusable scales so that (after the initial creation of scales) scaling a database can be done on a point-and-click basis.

Section 2 describes the graph representation of conjunctive queries that is used throughout this paper. The SQL translation of graphs is detailed in Sect. 3. The scaling approach is described in Section 4. The scales are not only used for the creation of the power context family; they define facets, which control the user options in a navigation application. Navigation in the power context family is discussed in Section 5. In this context, we revisit previous work on concept graphs and propose a new definition.

2 Conjunctive Queries

Conjunctive queries are a natural subset of database queries with nice theoretical properties [6]. Different representations for conjunctive queries are in popular use, including tableaux, formulas and Datalog rules [1]. In this paper, we represent conjunctive queries with *windowed intension graphs* [13] (similar to conceptual graphs [17]). An example windowed intension graph is shown in Fig. 2. This represents a query for 20th-century-born British authors who published in the 21st century. The rectangles are called *object nodes* and the rounded rectangles *relation nodes.* All object nodes of a query take on the role of variables. Colored nodes represent the subject(s) of the query; they are called *subject nodes.* Only object nodes can be subject nodes. The *window* represents the choice of subject nodes; so it can be thought of as a window into the data. Every object node carries a *sort* label, and a subject node carries in addition a *marker*, so that the combined label is of the form *sort/marker*. Every subject node is associated with a column in the result table, and the marker specifies the name of that column. The available sorts are precisely the table names in the database.

Each relation node is connected to object nodes by $n \geq 1$ outgoing arrows, labeled from 1 to n. A node with n outgoing arrows is said to have *arity* n. Two or more arrows may point to the same object node. A relation node carries one or more labels of the form *facet:attribute*. An *attribute* is a name for an n-ary relation, and a *facet* acts as a namespace for attributes. The label *facet:attribute* states that *attribute* applies to the objects at the end of the arrow tips (the i-th arrow points to the i-th argument). If a node label comprises two or more attributes, they must belong to the same facet.

A *relation sort* is an n-tuple $(s_1, \ldots, s_n)$ of table names. A facet only provides attributes of a single relation sort $(s_1, \ldots, s_n)$, which means that its attributes may only occur on n-ary relation nodes whose i-th arrows point to objects of sort s_i $(i = 1, \ldots, n)$. The nationality and DOB facets of Fig. 2 provide unary relations on Authors, pubdate provides unary relations on Books and wrote binary relations from Authors to Books.

Formally, an intension graph (cf. Fig. 1) is a 4-tuple (V, E, ν, κ) where V is the set of object nodes, E is the set of relation nodes, $\nu(e) := (v_1, \ldots, v_n)$ is the n-tuple of object nodes connected to $e \in E$, and $\kappa(u)$ is the label on the node $u \in V \cup E$. In the original definition of intension graphs [13], ν was required to be injective, so it was omitted. This is convenient for theory, but multiple edges make sense when working with facets.

A windowed intension graph is formalized as a pair $(\lambda, \mathcal{G})$, where $\mathcal{G}$ is an intension graph and $\lambda : X \nrightarrow V_{\mathcal{G}}$ is a partial map from a set X of *markers* to the object nodes of $\mathcal{G}$.

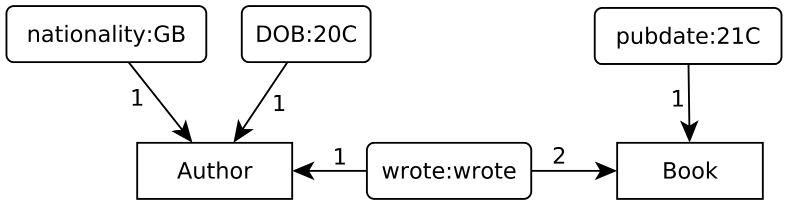


Fig. 1. Intension Graph of the statement "20th-century-born British authors who published in the 21st century"

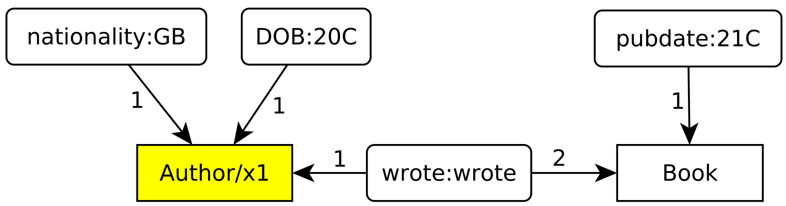


Fig. 2. Windowed Intension Graph of the statement "20th-century-born British authors who published in the 21st century"

Book

title	author	publication_date
Alice in Wonderland	1	1865-11-26
To the Lighthouse	2	1927-05-05
The Hitchhiker's Guide to the Galaxy	3	1979-10-12
Trigger Warning	4	2015-02-03
Harry Potter and the Deathly Hallows	5	2007-07-21
The Casual Vacancy	5	2012-09-27
The Shining	6	1977-01-28
Doctor Sleep	6	2013-09-24
The Da Vinci Code	7	2003-03-18
Inferno	7	2013-03-14

Author

id	first_name	last_name	nationality	date_of_birth
1	Lewis	Carroll	British	1832-01-27
2	Virginia	Woolf	British	1882-01-25
3	Douglas	Adams	British	1952-03-11
4	Neil	Gaiman	British	1960-11-10
5	J. K.	Rowling	British	1965-07-31
6	Stephen	King	American	1947-09-21
7	Dan	Brown	American	1964-06-22

Result Table

x1
Neil Gaiman
J. K. Rowling

Fig. 3. Database for the running example, consisting of tables `Book` and `Author`, and result table for the query in Fig. 2.

3 SQL Translation

A facet c provides an *SQL-interpretation* of its attributes by means of a function Φ_c, which maps an attribute a to a WHERE-condition $\Phi_c(a)$ in the syntax of the target database. If the attribute a has sort $(s_1, \ldots, s_n)$, then $\Phi_c(a)$ contains the placeholders $t_1, \ldots, t_n$, which have to be replaced by the respective arguments in every concrete case.

The expressions below interpret the attributes of the graph in Fig. 2 in terms of the schema of the database in Fig. 3.

$$\Phi_{\mathsf{wrote}}(\mathsf{wrote}) \equiv t_1.\mathsf{id} = t_2.\mathsf{author} \tag{1}$$

$$\Phi_{\mathsf{nationality}}(\mathsf{GB}) \equiv t_1.\mathsf{nationality}\text{="GB"} \tag{2}$$

$$\Phi_{\mathsf{DOB}}(\text{20C}) \equiv t_1.\mathsf{date\_of\_birth}\ \text{BETWEEN "1999-01-01" AND "1999-12-31"} \tag{3}$$

$$\Phi_{\mathsf{pubdate}}(\text{21C}) \equiv t_1.\mathsf{publication\_date}\ \text{BETWEEN "2000-01-01" AND "2099-12-31"} \tag{4}$$

To obtain a result table as in Fig. 3, we have to specify in addition how the objects of each sort (i.e. the rows of each table) are printed. This is achieved by fixing an *output expression* Ω_s for each sort s. For the sorts of the database in Fig. 3 we specify

$$\Omega_{\mathsf{Author}} \equiv CONCAT(t1.first\_name, \text{" "}, t1.last\_name) \quad , \tag{5}$$

$$\Omega_{\mathsf{Book}} \equiv t1.title \quad . \tag{6}$$

The SQL translation of a windowed intension graph is thus given by a statement of the following form:

$$\begin{aligned} \text{SELECT DISTINCT } & \Omega_{\text{sort}(u_1)}(u_1) \text{ AS } x_1 \text{ ,...,} \\ & \Omega_{\text{sort}(u_m)}(u_m) \text{ AS } x_m \\ \text{FROM } & \text{sort}(v_1) \text{ AS } v_1 \text{ ,...,} \\ & \text{sort}(v_n) \text{ AS } v_n \\ \text{WHERE } & \Phi_{c_1}(a_1)(v_{11}, ..., v_{1n_1}) \text{ AND ...} \\ \text{AND } & \Phi_{c_k}(a_k)(v_{k1}, ..., v_{kn_k}) \end{aligned} \tag{7}$$

In eqn. (7), the object nodes are represented by variables $v_1, \ldots, v_n$, and the FROM-clause can be seen as a variable declaration, which declares v_i to be of sort sort(v_i). In SQL terminology, v_i is called a *table alias*. The WHERE-clause contains for each attribute $c_j : a_j$ a WHERE-condition $\Phi_{c_j}(a_j)(v_{j1}, \ldots, v_{jn_j})$, where $\Phi_{c_j}(a_j)(v_{j1}, \ldots, v_{jn_j})$ denotes substitution of t_i by the applicable table alias v_{ji}. The FROM-WHERE-part realizes the query's underlying intension graph (cf. Fig. 1). The SELECT-clause defines an output column for each marker x_i on a subject node u_i. In SQL terminology, x_i is called a *column alias*.

So far, result tables display objects, but no values are shown beyond those that occur in the output expression (cf. Fig. 3). In a navigation application user interface (cf. Sect. 5), we envision relation nodes as controls to show in addition (or hide) the column values associated with a facet, by modifying the SELECT-clause in eqn. (7). These additional columns are not part of the concept extent, but they are of course informative. In the next section, we make precise how facets are associated with column values.

4 Database Scaling

A *syntactic interpretation* defines the symbols of a given signature by expressions over another signature (cf. [4]). The attributes provided by a facet c can be understood as symbols of a relational signature. The SQL-interpretation Φ_c is in this sense a syntactic interpretation: namely it interprets the attributes of c in a given database schema S (which is not exactly a signature, but the database-theoretic analogue).

Each facet c defines a context $\mathbb{K}_c$. This is how the extension of the attribute a of $\mathbb{K}_c$ is defined:

$$\begin{aligned} \text{SELECT } & \Omega_{s_1}(t_1),\ \ldots,\ \Omega_{s_n}(t_n) \\ \text{FROM } & s_1 \text{ AS } t_1,\ \ldots,\ s_n \text{ AS } t_n \\ \text{WHERE } & \Phi_c(a)(t_1,\ldots,t_n) \end{aligned} \tag{8}$$

Using eqns. (5) and (3) in eqn. (8), the 20C column in Fig. 4 is obtained. In this manner, the contexts for the DOB and pubdate facets (Figs. 4 and 5) can be derived from the database. The SQL definitions of the attributes *19C*, *20C* and

DOB	19C	20C	21C
Lewis Carroll	×		
Virginia Woolf	×		
Douglas Adams		×	
Neil Gaiman		×	
J. K. Rowling		×	
Stephen King		×	
Dan Brown		×	

Fig. 4. Context for the DOB facet

pubdate	19C	20C	21C
Alice in Wonderland	×		
To the Lighthouse		×	
Hitchhiker's Guide		×	
Harry Potter 7			×
The Casual Vacancy			×
Trigger Warning			×
The Shining		×	
Doctor Sleep			×
The Da Vinci Code			×
Inferno			×

Fig. 5. Context for the pubdate facet

21, which occur in both contexts, are not defined in the facets. Instead, every facet imports its attributes from exactly one underlying scale. A scale in FCA is a formal context which describes *values*; examples are ordinal scales (Fig. 7) and nominal scales (Fig. 8) [5]. The scale that underlies the DOB and pubdate facets is the Centuries scale in Fig. 11. The scales that we use to scale databases (i.e. generate context families from databases) should describe values that can occur in a database column; for the Centuries scale, these are ISO 8601 dates. Of course, it is generally not efficient or even possible to represent scales in the computer as cross-tables; we would expect the Centuries scale to describe all possible dates that can occur in a column, and not just the 17 dates of Fig. 11. A scale for a database must be able to produce an SQL definition for an attribute. The SQL definition for the 20C attribute is

$$z_1 \text{ BETWEEN "1999-01-01" AND "1999-12-31"} \quad , \tag{9}$$

where $z_1, z_2, z_3, \ldots$ are variables reserved for values. A facet *binds* a scale to one or more columns (each variable z_i is bound to a column). The Centuries scale is a unary scale, so its attributes are described by a single variable z_1. The DOB facet binds z_1 to the Author.date_of_birth column (which yields Φ_{DOB}, cf. (3)), whereas pubdate binds z_1 to Book.publication_date. [2]. Scales encode the actual logic, whereas a facet merely translates a relation between values into a relation between objects, by means of a syntactic substitution that is specified by the binding. The scale interface and facet class are shown in Fig. 12.

Examples for binary scales are equality scales (Fig. 9), which have been used in a prototype to generate binary single-column contexts for foreign keys, or distance scales (Fig. 10), which can be used to measure spatial distance between objects, or time spans between events. A comparison to the classic unary scales (Figs. 7 and 8) shows that these binary scales are in the same spirit.

A syntactic interpretation provides, in addition to symbol definitions, a formula that defines the carrier of the derived structure (cf. [4]). In our scaling approach, this is the object set of the derived context. We call this formula a *domain expression* and denote it by $\Phi_c(*)$, where $*$ is a special symbol. The domain expression for the contexts in Figs. 4 and 5 is a sort restriction. Ideally, the domain expression would also be a WHERE-condition, but SQL requires special treatment in this case. However, a WHERE-condition is allowed in addition to the sort restriction. The domain expression for the wrote facet is "$t1$.id=$t2$.author", where $t1$ is an Author and $t2$ is a Book. On top of this, the *wrote* facet uses a distance scale, bound to Author.date_of_birth and Book.publication_date, to measure at what age an author wrote a particular book. A facet supports renaming of attributes to allow for more expressive attribute names than the generic names provided by the scales. The context derived from the wrote facet is the bottom context in Fig. 6.

The contexts that are derived from the facets can be assembled into a power context family (Fig. 6). Working with power context families is more convenient for mathematical investigations, whereas working with the contexts derived from the facets (as in Figs. 4 and 5) is more convenient for practical work. As with the scales, it is not necessary that the power context family, or individual facets, are explicitly constructed. The power context family has been realized in a prototype as a virtual layer around the database, although the computation of refinement options (cf. Section 5) required an additional query, and SQL does not adequately support all types of scales (such as taxonomies), which may require post-processing of result tables.

[2] Direct specification of $\Phi_c(a)$ in the facet (thus by-passing scales) is not supported.

0	sort: Author	sort: Book
Lewis Carroll	×	
Virginia Woolf	×	
Douglas Adams	×	
Neil Gaiman	×	
J. K. Rowling	×	
Stephen King	×	
Dan Brown	×	
Alice in Wonderland		×
To the Lighthouse		×
Hitchhiker's Guide		×
Harry Potter 7		×
The Casual Vacancy		×
Trigger Warning		×
The Shining		×
Doctor Sleep		×
The Da Vinci Code		×
Inferno		×

1	nationality: GB	nationality: USA	DOB: 19C	DOB: 20C	DOB: 21C	pubdate: 19C	pubdate: 20C	pubdate: 21C
Lewis Carroll	×		×					
Virginia Woolf	×		×					
Douglas Adams	×			×				
Neil Gaiman	×			×				
J. K. Rowling	×			×				
Stephen King		×		×				
Dan Brown		×		×				
Alice in Wonderland						×		
To the Lighthouse							×	
Hitchhiker's Guide							×	
Harry Potter 7								×
The Casual Vacancy								×
Trigger Warning								×
The Shining							×	
Doctor Sleep								×
The Da Vinci Code								×
Inferno								×

2	wrote: wrote	wrote: age≤30	wrote: age≤40	wrote: age≤50
(Lewis Carroll, Alice in Wonderland)	×		×	×
(Virginia Woolf, To the Lighthouse)	×			×
(Douglas Adams, Hitchhiker's Guide)	×	×	×	×
(Neil Gaiman, Trigger Warning)	×			
(J. K. Rowling, Harry Potter 7)	×			×
(J. K. Rowling, The Casual Vacancy)	×			×
(Stephen King, The Shining)	×	×	×	×
(Stephen King, Doctor Sleep)	×			
(Dan Brown, The Da Vinci Code)	×		×	×
(Dan Brown, Inferno)	×			×

Fig. 6. Power Context Family

Ordinal	≤1	≤2	≤3	≤4	≤5
1	×	×	×	×	×
2		×	×	×	×
3			×	×	×
4				×	×
5					×

Fig. 7. Ordinal Scale

Nominal	=1	=2	=3	=4	=5
1	×				
2		×			
3			×		
4				×	
5					×

Fig. 8. Nominal Scale

Equality	=
(1,1)	×
(1,2)	
(1,3)	
(2,1)	
(2,2)	×
(2,3)	
(3,1)	
(3,2)	
(3,3)	×

Fig. 9. Equality Scale

Distance	=0	≤1	≤2
(1,1)	×	×	×
(1,2)		×	×
(1,3)			×
(2,1)		×	×
(2,2)	×	×	×
(2,3)		×	×
(3,1)			×
(3,2)		×	×
(3,3)	×	×	×

Fig. 10. Distance Scale

Centuries	19C	20C	21C
1832-01-27	×		
1865-11-26	×		
1882-01-25	×		
1927-05-05		×	
1947-09-21		×	
1952-03-11		×	
1960-11-10		×	
1964-06-22		×	
1965-07-31		×	
1977-01-28		×	
1979-10-12		×	
2003-03-18			×
2007-07-21			×
2012-09-27			×
2013-03-14			×
2013-09-24			×
2015-02-03			×

Fig. 11. Centuries Scale

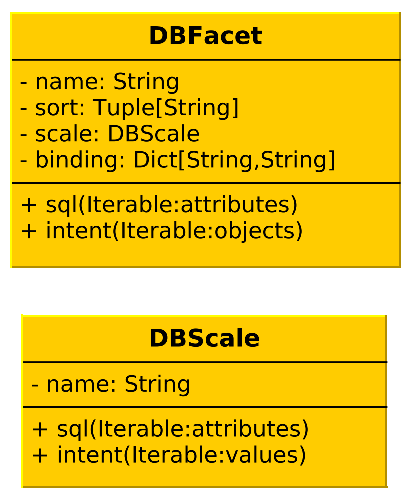


Fig. 12. The main API functions of the DBFacet class and DBScale interface. The internal representation of scales is up to the implementation.

5 Navigation using Projectional Concept Graphs

The ideas of the previous sections can be turned to account in a navigation application. The viability of its core features has already been explored in a prototype; a full version will be presented in an upcoming paper.

The application provides for two roles, user and admin. In the admin role, one can connect to an existing database, view its schema and a list of available scales, bind scales to database columns (thus scaling the database) and store the binding, together with the database connection info, in a file (i.e. the database can be read-only). A binding, together with the database that it references, constitutes a virtual power context family.

In the user role, one can choose from a list of available power context families to navigate in. Note that the user does not need to know that the power context family originates from a relational database, and indeed, there could be different back ends for different sources of data, such as RDF or object-oriented databases (although we have only worked this out for relational databases with SQL access). Different user interfaces are possible, but it is instructive to assume that a conjunctive query looks to the user like the graph in Fig. 2. As mentioned in Sect. 2, it is formalized by a windowed intension graph $(\lambda, \mathcal{G})$.

A *solution* of an intension graph $\mathcal{G}$ in a power context family $\vec{\mathbb{K}}$ is formalized by a map $\varphi : \mathcal{G} \to \vec{\mathbb{K}}$ from object nodes to objects (of the context $\mathbb{K}_0$). The set of all solutions is denoted by $\mathcal{S}(\mathcal{G}, \vec{\mathbb{K}})$. For a windowed intension graph $(\lambda, \mathcal{G})$, the rows in the result table are the maps $\lambda \circ \varphi$ with $\varphi \in \mathcal{S}(\mathcal{G}, \vec{\mathbb{K}})$. In the following, we introduce projectional concept graphs as a basic structure for navigation.

Definition 1 (Projectional Concept Graph). *A* projectional concept graph *is a 5-tuple* $(V, E, \nu, \kappa, \mathrm{ext}_{\vec{\mathbb{K}}})$ *comprised of an intension graph* $\mathcal{G} := (V, E, \nu, \kappa)$ *and its* extension map

$$\mathrm{ext}_{\vec{\mathbb{K}}}(v) := \{\varphi(v) \mid \varphi \in \mathcal{S}(\mathcal{G}, \vec{\mathbb{K}})\} \tag{10}$$

for a given power context family $\vec{\mathbb{K}}$ *with* $\mathcal{S}(\mathcal{G}, \vec{\mathbb{K}}) \neq \emptyset$ *(i.e.* $\mathrm{ext}_{\vec{\mathbb{K}}}(v) \neq \emptyset$ *for all* $v \in V$*). We call* $\mathrm{ext}_{\vec{\mathbb{K}}}(v)$ *the* node extent *of* v.

The node extent of the `Author` node in Fig. 1 is the extent of the windowed intension graph in Fig. 2. It is thus an extent in the lattice $\underline{\mathcal{B}}_1(\vec{\mathbb{K}})$ of unary concepts over the power context family $\vec{\mathbb{K}}$ (cf. [13]). Therefore, projectional concept graphs should indeed be considered concept graphs.

Considering node extents rather than whole result tables spares the user going through large result tables; the navigation approach allows however to place windows of arbitrary size on the graph, if the specific combinations of objects in the solution are of interest.

Refinement options are given by a triple $(E^+, \kappa^+, \theta^+)$. For each $v \in V$, $E^+(v)$ is a set of facets for which a new relation node can be connected to v, extending the graph structure. For each $u \in V \cup E$, $\kappa^+(u)$ is a set of scale concepts which can replace $\kappa(u)$. And θ^+ is a set of pairs of object nodes in the graph which can be merged. All refinement options lead to a refined projectional concept graph that has at least one solution in $\vec{\mathbb{K}}$.

6 Related Work

Concept graphs and power context families were defined by Wille [18]. Relational context families, used in Relational Concept Analysis (RCA) developed by Huchard et. al, [9] are similar to power context families but define different contexts for objects of different sorts. The contexts derived from facets in this paper represent facets of such sort contexts. Faceted navigation on the basis of FCA was suggested by Priss [15] and later developed by Eklund and Ducrou [3]. In RCA, conceptual scaling is generalized to relations but produces unary contexts only. The idea of scaling databases into power context families was formulated by Prediger and Wille [14] and expanded on by Hereth [8]. The scales presented there correspond to facets in our work; a central idea to conceptual scaling, the translation of properties of values into properties of objects, is not reflected in this work, but is addressed by the scales in our work.

From the beginning, conceptual graphs have been considered as a database interface [16]. Their translation into logical formulas, stated by Sowa [17], seems to imply that an interpretation as conjunctive queries is intended for conceptual graphs with variables. An SQL translation of Wille's concept Graphs, which treats concept graphs as conjunctive queries, is described by Groh and Eklund [7]. Interpretations were provided by a power context family, but it was encoded in a database (which imposes a particular format), not derived from the database, so scaling (as we define it here) was not involved. Both object and relation nodes were considered variables, whereas in the present work, only object nodes are considered variables.

An intension graph can be thought of as Wille's abstract concept graph (see [18, p. 300]). A concept graph (in the standard definition) defines in addition a realization ϱ which, like the extension map in Def. 1, assigns to each object node a nonempty set of objects. In a concept graph, the elements of the sets $\varrho(v)$ can be freely combined to obtain solutions, whereas for projectional concept graphs, each element of $\varrho(v)$ is part of *some* solution.

The navigation approach of Sect. 5 has been described in [11]. FCA-based navigation in relational data is also the subject of [2].

7 Conclusion

This paper describes the fundamental theory of a Web application that allows connection to a relational database, its scaling interactively into a power context family, and navigation within that context family. The conceptual scaling approach used is based on a syntactic interpretation of attributes, which results in an FCA-based method to build suitable query vocabulary around a relational database. Generic and reusable scales allow easy database scaling on a point-and-click basis. Scales constitute facets in a faceted-navigation approach based on projectional concept graphs, which are a new class of concept graphs.

References

1. Abiteboul, S., Hull, R., Vianu, V.: Foundations of Databases. Addison-Wesley (1995)
2. Azmeh, Z., Huchard, M., Napoli, A., Hacene, M.R., Valtchev, P.: Querying relational concept lattices. In: Proc. of the 8th Intl. Conf. on Concept Lattices and their Applications (CLA'11). pp. 377–392 (2011)
3. Ducrou, J., Eklund, P.: Faceted document navigation using conceptual structures. In: Hitzler, P., Schärfe, H. (eds.) Conceptual Structures in Practice, pp. 245–271. Chapman & Hall/CRC (2009)
4. Ebbinghaus, H.D., Flum, J., Thomas, W.: Mathematical Logic. Springer, New York, second edn. (1994)
5. Ganter, B., Wille, R.: Formal concept analysis: mathematical foundations. Springer, Berlin (1999)
6. Gottlob, G., Leone, N., Scarcello, F.: On tractable queries and constraints. In: Bench-Capon, T.J., Soda, G., Tjoa, A.M. (eds.) Database and Expert Systems Applications. pp. 1–15. Springer Berlin Heidelberg, Berlin, Heidelberg (1999)
7. Groh, B., Eklund, P.W.: Algorithms for creating relational power context families from conceptual graphs. In: Proceedings of ICCS 1999. pp. 389–400 (1999), `https://doi.org/10.1007/3-540-48659-3_24`
8. Hereth, J.: Relational scaling and databases. In: Priss, U., Corbett, D., Angelova, G. (eds.) Proceedings of ICCS 2002. LNAI, vol. 2393, pp. 62–76. Springer, Heidelberg (2002)
9. Huchard, M., Hacene, M.R., Roume, C., Valtchev, P.: Relational concept discovery in structured datasets. Annals of Mathematics and Artificial Intelligence 49(1-4), 39–76 (2007)
10. Huchard, M., Roume, C., Valtchev, P.: When concepts point at other concepts: the case of UML diagram reconstruction. In: Proceedings of FCAKDD 2002. pp. 32–43 (2002)
11. Kötters, J.: Object configuration browsing in relational databases. In: Valtchev, P., Jäschke, R. (eds.) Proceedings of ICFCA 2011. LNCS, vol. 6628, pp. 151–166. Springer (2011)
12. Kötters, J.: Concept lattices of a relational structure. In: Pfeiffer, H.D., Ignatov, D.I., Poelmans, J., Gadiraju, N. (eds.) Proceedings of ICCS 2013. LNCS, vol. 7735, pp. 301–310. Springer (2013)
13. Kötters, J.: Intension graphs as patterns over power context families. In: Huchard, M., Kuznetsov, S. (eds.) Proceedings of CLA 2016. CEUR Workshop Proceedings, vol. 1624. CEUR-WS.org (2016)
14. Prediger, S., Wille, R.: The lattice of concept graphs of a relationally scaled context. In: William M. Tepfenhart, W.R.C. (ed.) Proceedings of ICCS 1999. LNAI, vol. 1640, pp. 401–414. Springer (1999)
15. Priss, U.: Lattice-based information retrieval. Knowledge Organization 27(3), 132–142 (2000)
16. Sowa, J.F.: Conceptual graphs for a data base interface. IBM Journal of Research and Development 20(4), 336–357 (1976)
17. Sowa, J.F.: Conceptual Structures: Information Processing in Mind and Machine. Addison-Wesley, Reading, MA (1984)
18. Wille, R.: Conceptual graphs and formal concept analysis. In: Lukose, D., Delugach, H.S., Keeler, M., Searle, L., Sowa, J.F. (eds.) Proceedings of ICCS 1997, 5th International Conference on Conceptual Structures. LNCS, vol. 1257, pp. 290–303. Springer, Heidelberg (1997)

Generalized metrics with applications to ratings and formal concept analysis

Tobias Gäbel-Hökenschnieder[1], Thorsten Pfeiffer[2], and Stefan E. Schmidt[1]

[1] Institut für Algebra, Technische Universität Dresden,
[2] Bad Kreuznach, Germany
tgh@mail.de, thorsten.b.pfeiffer@gmail.com, midt1@msn.com

Abstract. We introduce an order theoretic approach to generalized metrics that covers various concepts of distance. In particular, we point out the role of supermodular mappings on lattices, which we then apply in diverse settings such as comparison of ratings and formal concept lattices.

Keywords: Generalized metric, supermodular mappings, comparison of ratings, semimodular lattices, formal concept analysis

1 Introduction

Generalized metrics recently have become of increased interest for modeling *directed distances* with values in qualitative measurement spaces including ordered monoids and lattices. In [8] generalized metrics are proposed which turned out to be relevant for formal concept analysis and closure operators (see [4], [9], [10]).

In this paper, we will apply generalized metrics in order to compare *ratings*, that is comparing the rating methodologies of different rating agencies with different result scales. We analyze suitable result scales for the rating process and show that ratings are not limited to chain lattices but can as well use certain *semimodular lattices* as target. The paper also considers applications to formal concept analysis covering the extensional as well as the intensional point of view.

For our approach *supermodularity* plays an important role, which goes beyond ideas of measurement associated with Dempster-Shafer-Theory (see [10]).

2 A prior result on generalized metrics

In this section we recall a theorem on generalized metrics (compare [8]). We start with

Definition 1 ([8]) *$\mathcal{M} = (M, *, \varepsilon, \leq)$ is an* **ordered monoid** *if $\mathbb{M} := (M, *, \varepsilon)$ is a monoid and $(M, \leq)$ is a poset such that $a \leq b$ implies $c * a \leq c * b$ and $a * c \leq b * c$, for all $a, b, c \in M$.*

The class of ordered monoids is quite large. Examples are:

- $(\mathbb{R}, +, 0, \leq)$ and $(\mathbb{R}_+, *, 1, \leq)$ under the natural ordering of the real numbers;
- for any set E, $(\mathcal{P}(E), \cup, \subseteq, \emptyset)$ and $(\mathcal{P}(E), \cap, \subseteq, E)$;
- a meet-semilattice $(L, \wedge, 1_L, \leq_L)$ bounded from above by 1_L and a join-semilattice $(L, \vee, 0_L, \leq_L)$ bounded from below by 0_L.

In order to distinguish the respective order relations, in the following we will use the symbol "$\leq_{\mathbb{P}}$" for the order relations of a given poset $\mathbb{P}$ and "$\leq$" of a given ordered monoid $\mathcal{M}$, respectively:

Definition 2 ([8]) *Let $\mathbb{P} = (P, \leq_{\mathbb{P}})$ be a poset and $\mathcal{M} = (M, *, \varepsilon, \leq)$ be an ordered monoid. A mapping*

$$\varDelta\colon \leq_{\mathbb{P}} \longrightarrow M$$

is called **functorial w. r. t.** $(\mathbb{P}, \mathcal{M})$*, if*

- *for all $p \in P$: $\quad \varDelta(p,p) = \varepsilon$,*
- *for all $p, t, q \in P$ with $p \leq_{\mathbb{P}} t \leq_{\mathbb{P}} q$: $\quad \varDelta(p,t) * \varDelta(t,q) = \varDelta(p,q)$.*

Furthermore, $\varDelta$ is called **weakly positive***, if $\varepsilon \leq \varDelta(p,q)$ for all $(p,q) \in \ \leq_{\mathbb{P}}$.*

In case $\mathbb{P} = (P, \leq_{\mathbb{P}})$ is a lattice, $\varDelta$ is called **supermodular** *w. r. t. $(\mathbb{P}, \mathcal{M})$ (resp.* **modular***), if $\varDelta(p \wedge q, q) \leq \varDelta(p, p \vee q)$ (resp. equality) holds for all $p, q \in P$.*

So far, functorial mappings are only defined on the order relation $\leq_{\mathbb{P}} \subseteq P \times P$. In order to extend functorial mappings from the ordering $\leq_{\mathbb{P}}$ to its superset $P \times P$, we need

Definition 3 ([8]) *Let P be a set, and $\mathcal{M} = (M, *, \varepsilon, \leq)$ be an ordered monoid. A function $d\colon P \times P \longrightarrow M$ is called* **generalized quasi-metric (GQM) w. r. t.** $(P, \mathcal{M})$*, if*

(A0) *for all $(p,q) \in P \times P$: $\quad \varepsilon \leq d(p,q)$*

(A1) *for all $p \in P$: $\quad d(p,p) = \varepsilon$*

(A2) *for all $p, t, q \in P$: $\quad d(p,q) \leq d(p,t) * d(t,q)$*

If in addition, (A3) holds, d is a **generalized metric (GM) w. r. t.** $(P, \mathcal{M})$*:*

(A3) *for all $(p,q) \in P \times P$: $\quad d(p,q) = \varepsilon = d(q,p) \quad \Longrightarrow \quad p = q$*

It is not quite obvious if functorial maps can be extended from $\leq_{\mathbb{P}}$ to the superset $P \times P$, which gives rise to the following

Question: For a given $\Delta\colon \leq_{\mathbb{P}} \longrightarrow M$*, does there exist a generalized quasi-metric* $d\colon P \times P \longrightarrow M$ *w. r. t.* $(P, \mathcal{M})$ *which extends* Δ *such that* $d|_{\leq_{\mathbb{P}}} = \Delta$*?*

We find a positive answer and sufficient conditions in the following

Theorem 1 ([8]) *Let* $\mathbb{P} = (P, \leq_{\mathbb{P}})$ *be a lattice and let* $\mathcal{M} = (M, *, \varepsilon, \leq)$ *be an ordered monoid. If a map* $\Delta\colon \leq_{\mathbb{P}} \longrightarrow M$ *is weakly positive, supermodular and functorial w. r. t.* $(\mathbb{P}, \mathcal{M})$*, then*

$$d\colon P \times P \longrightarrow M, (p, q) \mapsto \Delta(p \wedge q, q)$$

is a GQM w. r. t. $(P, \mathcal{M})$*.*

3 Application to ratings

In this section we formalize the *rating process* and show how to compare ratings from different sources.

Let O be a finite set of *objects* to be rated, prominent examples are financial entities which issue debt. There are different (credit) rating agencies applying different ratings, where a (credit) *rating* is a mapping $A : O \to C(n){:=}\{0, \ldots, n\}$. "0" represents the lowest (credit) quality, "n" the highest, and $C(n)$ is called *rating scale*. It is clear that $C(n)$ is a complete lattice, naturally and totally ordered by "$\leq$", and n is called *length* of the *chain* $C(n)$. Our goal is to compare the results of two different rating agencies. The two agencies rate the same objects but they apply different rating methodologies, which leads to the

Question: Given two ratings A and B from different sources, which one is more progressive?

Progressive in this context means systematically giving a better rating to the same set of objects. Such "optimism" might lead to an underestimation of the underlying risks compared to the less progressive view since the more progressive view tends to ask for a lower risk premium.

Input: O a set, a finite chain $S := C(n) = \{0, \ldots, n\}$, two ratings $A, B : O \to S$

Definition 4 (Rating *B* is progressive given rating *A*)

$$D^{+}(A, B) := \sum_{o \in O\colon A(o) \leq B(o)} \mathit{rank}\ B(o) - \mathit{rank}\ A(o)$$

where the natural rank function in a chain *is given by rank* $:= s,\ s \in S$

We immediately notice:

- $D^+(A,B)$ is well defined and finite if O is finite: since there are only finitely many objects to be rated, we do not need to worry about non finite or even non countable sets.
- $D^+(A,B) \geq 0$
- $D^+(A,B) = 0$ iff $\forall o \in O : B(o) \leq A(o)$

A little less obvious is the following property: $D^+(A,B)$ is "triangular", i.e. $\forall E\colon O \to S$: $D^+(A,B) \leq D^+(A,E) + D^+(E,B)$. To see this we apply Theorem 1 as follows:

The set $\mathcal{O}$ of all ratings $O : A \to S$ is endowed with a natural order: $A \leq_{\mathbb{O}} B$ if $A(o) \leq B(o)$ for all $o \in O$. We write $\mathbb{O} = (\mathcal{O}, \leq_{\mathbb{O}})$. $\mathbb{O}$ is even a lattice where $(A \vee B)(o) = \max(A(o), B(o))$ and $(A \wedge B)(o) = \min(A(o), B(o))$.

For $A \leq_{\mathbb{O}} E$ define $\Delta^+\colon \leq_{\mathbb{O}} \to \mathbb{N} \cup \{0\}$ via $\Delta^+(A,E) := \sum_{o \in O} rank\ E(o) - rank\ A(o)$. Δ^+ is functorial, since $\Delta^+(A,E) = \Delta^+(A,B) + \Delta^+(B,E)$ for the totally ordered triple $A \leq_{\mathbb{O}} B \leq_{\mathbb{O}} E$. Since $\min(a,b) + \max(a,b) = a + b$ for all real numbers a, b, Δ^+ is even a modular map.

Applying Theorem 1, thus D^+, which is the the extension of Δ^+, is triangular.

Usually $D^+(A,B) \neq D^+(B,A)$, i.e. D^+ is not symmetric. If $D^+(A,B) > D^+(B,A)$ then A is more *conservative* than B, and B is more progressive than A. In order to measure a symmetric distance between ratings, we proceed as follows:

Input: O a finite set, a finite chain $C(n)$, two ratings $A, B : O \to C(n)$

Definition 5 (Distance between ratings A and B)

$$D(A,B) := D^+(A,B) + D^+(B,A)$$

Being the L^1-distance of the rankings, D is symmetric: $D(A,B) = D(B,A)$, and $D(A,B) = 0 = D(B,A)$ if and only if $A = B$.

We will use D to derive a brute-force algorithm to solve the following issue: Rating scales do not need to be identical since different raters might use different rating scales:

Question: how can we compare ratings in case the rating scales are of different length?

The algorithm we propose will use embeddings (order preserving injections) of one chain into the other and minimize the distance D over all possible embeddings. For example, there are 3 possibilities of embedding $C(1)$ into $C(2)$, and 6 possibilities of embedding $C(1)$ into $C(3)$.

Input: O a finite set, ratings $A : O \to C(k)$, $B : O \to C(n)$, $k, n \in \mathbb{N}$ with $k \leq n$

Algorithm 1 (Scaling with minimal distance) –

– *Run through all embeddings* $E_i : C(k) \to C(n)$

– *Calculate* $E_i \circ A$ *and* $D(B,\ E_i \circ A)$ *for each embedding* E_i

– *Pick (one of) the* E_i *with minimal distance* $D(B,\ E_i \circ A)$

Comments: This algorithm is based on the implicit assumption, that both rating agencies are subject matter experts and "know what they are doing", which is reflected in building the minimum of the distances over all possible embeddings. No (subjective) expert opinion or management discretion is needed to decide before hand on the best possible embedding: instead, the algorithm increases objectivity in the sense that the best embedding is chosen purely based on the input data.

4 Generalized targets for ratings

So far, we only have used finite chains - i.e. totally ordered sets - for the rating process. In this section we will generalize the target sets of the rating process, another application of Theorem 1 will help us to answer the following

Questions: Are we limited to totally ordered sets? What about more general lattices as target of ratings? Which lattices will work?

The idea is to use Theorem 1 to "extend" the distances defined above, which essentially compares positions in a finite chain. To this end, we need

Definition 6 (Jordan-Dedekind chain condition) *A poset* P *is said to satisfy the* Jordan-Dedekind chain condition *if any two maximal chains between the same elements of* P *have the same finite length, where a chain* $C \subseteq P$ *is called* maximal *if, for any chain* $D \subseteq P$*,* $C \subseteq D$ *implies* $C = D$*.*

If $p, q \in P$ with $p \leq q$, then p, q are contained in at least one chain in P. In order to measure a distance Δ between p and q using the natural rank function as introduced in Definition 4, we can take any *maximal chain* between p and q, and the Jordan-Dedekind chain condition makes sure that this procedure is independent of choice of the maximal chain, and thus the following is well defined: $\Delta(p, q) := \text{length}(C) = \text{rank}(q)$ (in C) for any maximal chain C with $p, q \in C$.

The lattice depicted in Figure 1 violates the Jordan-Dedekind chain condition and serves as counter example: the chain on the left side yields $\Delta(x \wedge y, x \vee y) = 2$, the chain on the right would yield 3 as distance Δ between $x \wedge y$ and $x \vee y$.

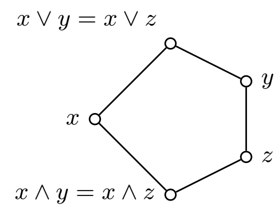


Fig. 1: A lattice violating the Jordan-Dedekind chain condition

Slightly more general formulated, in a poset P with Jordan-Dedekind chain condition which has a smallest element 0_P we can define rank(q) in the same way for every element $q \in P$ as length(C) for any maximal chain containing 0_P and q. This rank function $\Delta\colon P \to \mathbb{N} \cup \{0\}$ is weakly positive and functorial. If Δ happens to be also supermodular, then applying Theorem 1 all together we get

Corollary 1 *Let P be a lattice with Jordan-Dedekind chain condition and supermodular rank function Δ. Then*

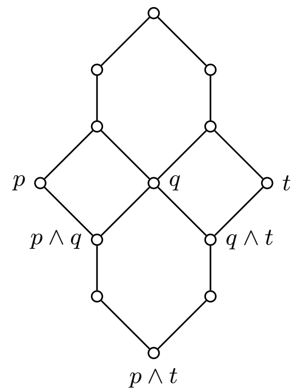
$$\boxed{d\colon P \times P \longrightarrow M, \quad (p,q) \mapsto \Delta(p \wedge q, q)}$$

is a GQM w. r. t. $(P, \mathbb{N} \cup \{0\})$. Furthermore, given two ratings $A, B : O \to P$,

$$D^{+}(A,B) := \sum_{o \in O\colon A(o) \leq B(o)} rank\ B(o) - rank\ A(o)$$

is a also a GQM w. r. t. $(O, \mathbb{N} \cup \{0\})$.

Remark: The Jordan-Dedekind chain condition per se is not enough, as we can deduct from the lattice depicted in Figure 2.



$$d(p,t) = rank(p) - rank(p \wedge t) = 3 - 0 = 3$$
$$d(p,q) = rank(p) - rank(p \wedge q) = 3 - 2 = 1$$
$$d(q,t) = rank(q) - rank(q \wedge t) = 3 - 2 = 1$$

Hence, $d(p,t) = 3 > 2 = d(p,q) + d(q,t)$, and d is not a triangular metric.

Fig. 2: A complete lattice satisfying the Jordan-Dedekind chain condition but bearing a non triangular metric based on the rank function

So with the help of Corollary 1 we can give a positive answer: not only simple chains are suitable targets for the rating process, much more, there is the huge class of lattices which allow for a (finite) Jordan-Dedekind chain condition together with a supermodular rank function as rating targets. In particular, modular lattices of finite length will work very well, where a lattice is called *modular* if it does not contain a sublattice of the form in Figure 1.

But we are not limited to modular lattices. In Figure 3 there is an example of a *lower semimodular* lattice which is not modular.

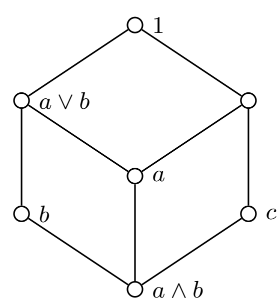


Fig. 3: A non modular lattice satisfying the Jordan-Dedekind chain condition

A lattice L is called *lower semimodular* if $\forall a, b \in L\colon\ b \lessdot a \vee b \Rightarrow a \wedge b \lessdot a$, where we write $a \lessdot b$ if $a < b$ and $a < x \leq b$ implies $x = b$. Every modular is lower semimodular, but the converse is obviously not true. In particular, the rank function of the lattice depicted in Figure 3 is only supermodular but not modular since $rank(b)+rank(c) = 2 < 3 = rank(1) = rank(b\vee c)+rank(b\wedge c)$.

This behavior of the rank function is somewhat typical, as we can see by the following characterization of lower semimodular lattices:

Theorem 2 *Let L be a lattice bounded from below such that any chain between any two elements of L is finite. L is lower semimodular if and only if L possesses a rank function r such that $\forall x, y \in L$:*

$$rank(x) + rank(y) \leq rank(x \vee y) + rank(x \wedge y).$$

L is modular if and only if $\forall x, y \in L$:

$$rank(x) + rank(y) = rank(x \vee y) + rank(x \wedge y).$$

Proof: this is the dual version of Theorem 2.27 from [1].

So lower semimodular lattices, bounded from below such that any chain between any two elements is finite, are exactly the appropriate class of lattices for our purposes.

Furthermore, we can generalize the scaling Algorithm 1 to this class of lattices using *rank preserving* mappings, where a mapping φ between two lattices L and L', which both possess a well-defined rank function, is called *rank preserving* if $rank(u) \leq rank(v)$ implies $rank(\varphi(u)) \leq rank(\varphi(v))$ for all $u, v \in L$.

Input: O a finite set, ratings $A : O \to L$, $B : O \to L'$ for lower semimodular lattices L, L', where the finite number of elements of L' is denoted by n, and k denotes the number of elements of L such that $k \leq n$.

Algorithm 2 (Extended scaling with minimal distance) –

- *Run through all rank preserving injections $E_i : L \to L'$*
- *Calculate $E_i \circ A$ and $D(B,\ E_i \circ A)$ for each embedding E_i*
- *Pick (one of) the E_i with minimal distance $D(B,\ E_i \circ A)$*

Actually, this algorithm is the same as Algorithm 1, but applied to rank preserving injections instead of order preserving embeddings.

An example with only two rank preserving injections is depicted in Figure 4. Should we pick $\varphi(1) = a$ and $\varphi(2) = b$ or should we opt for the other possibility $\varphi(1) = b$ and $\varphi(2) = a$? Based on data, we would pick the possibility with the minimum distance.

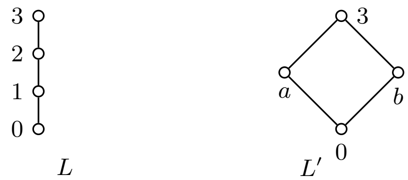


Fig. 4: Lattices L, L' which allow only for two rank preserving injections $\varphi\colon L \to L'$

5 Application to concept lattices

In order to keep the paper self-contained, we give a very short summary of formal concept lattices:

A *formal context* is a triple $\mathbb{K} = (G, M, I)$, where G is a set of objects, M is a set of attributes, and $I \subseteq G \times M$ is a binary incidence relation that expresses which objects have which attributes. For subsets $X \subseteq G$ of objects and subsets $Y \subseteq M$ of attributes, one defines the following mappings between the power sets of G and M:

- $G \supseteq X \mapsto X^{\triangleright} = \{m \in M\colon (x, m) \in I \text{ for every } x \in X\}$, and dually
- $M \supseteq Y \mapsto Y^{\triangleleft} = \{g \in G\colon (g, y) \in I \text{ for every } y \in Y\}$.

Clearly, $X_1 \subseteq X_2$ implies $X_1^{\triangleright} \supseteq X_2^{\triangleright}$ and $Y_1 \supseteq Y_2$ implies $Y_1^{\triangleleft} \subseteq Y_2^{\triangleleft}$. By a *formal concept* of the context $\mathbb{K}$ is understood a pair (X, Y) with $X \subseteq G$, $Y \subseteq M$ such that $X^{\triangleright} = Y$ and $Y^{\triangleleft} = X$. The set X is called the *extent* of the concept, and the set Y is referred to as *intent* of the concept. (X_1, Y_1) is called a *subconcept* of (X_2, Y_2) if $X_1 \subseteq X_2$, and we write $(X_1, Y_1) \preceq (X_2, Y_2)$. The class $\mathfrak{B}\mathbb{K}$ of all formal concepts of a given context $\mathbb{K}$ turns out to be ordered by $\preceq$, and even to be a complete lattice (cfr. Theorem 3 in chapter 1 of [9]), where supremum resp. infimum of two formal concepts are defined by

- $(X_1, Y_1) \vee (X_2, Y_2) = ((X_1 \cup X_2)^{\triangleright\triangleleft}, Y_1 \cap Y_2)$, resp.
- $(X_1, Y_1) \wedge (X_2, Y_2) = (X_1 \cap X_2, (Y_1 \cup Y_2)^{\triangleleft\triangleright})$

The lattice $\mathfrak{B}\mathbb{K}$ is called *concept lattice* of the context $\mathbb{K} = (G, M, I)$.

One consequence is that the mapping $G \supseteq X \mapsto X^{\triangleright\triangleleft} \subseteq G$ is a *closure mapping*, and therefore $\#(X) \leq \#(X^{\triangleright\triangleleft})$, where $\#(A)$ is the *count measure* of a set A, i.e. counting the number of elements of A. After these preparations we can derive

Proposition 1 *For $\alpha := (A_1, A_2)$, $\beta := (B_1, B_2) \in \mathfrak{B}\mathbb{K}$ with $\alpha \leq \beta$, the map*

$$\Delta\colon \leq_{\mathfrak{B}\mathbb{K}} \longrightarrow \mathbb{N} \cup \{0\}, \quad (\alpha, \beta) \mapsto \Delta(\alpha, \beta) := \#(B_1 - A_1). \tag{1}$$

is functorial, weakly positive and supermodular.

Proof. Let $\gamma \coloneqq (C_1, C_2)$ such that $\alpha \leq \beta \leq \gamma$.

- Firstly, we can calculate

$$\begin{aligned}\Delta(\alpha,\gamma) &= \#C_1 - \#A_1 \\ &= (\#C_1 - \#B_1) + (\#B_1 - \#A_1) \\ &= \Delta(\beta,\gamma) + \Delta(\alpha,\beta).\end{aligned}$$

  Secondly, we see that

$$\Delta(\alpha,\alpha) = \#(A_1 - A_1) = \#\emptyset = 0.$$

  Hence, Δ is functorial.

- Δ is weakly positive, since $0 \leq \Delta(\alpha,\beta)$ holds for all $\alpha, \beta \in \mathfrak{B}\mathbb{K}$.
- To show that Δ is supermodular, we start to calculate $\Delta(\alpha \wedge \beta, \beta)$ and $\Delta(\alpha, \alpha \vee \beta)$ separately:

$$\begin{aligned}\Delta(\alpha \wedge \beta, \beta) &= \Delta\big((A_1 \cap B_1, (A_2 \cup B_2)^{\lhd\rhd}), (B_1, B_2)\big) \\ &= \#\big(B_1 - (A_1 \cap B_1)\big) \\ &= \#B_1 - \#(A_1 \cap B_1).\end{aligned}$$

$$\begin{aligned}\Delta(\alpha, \alpha \vee \beta) &= \Delta\big((A_1, A_2), (A_1 \cup B_1)^{\rhd\lhd}, A_2 \cap B_2)\big) \\ &= \#\big((A_1 \cup B_1)^{\rhd\lhd} - A_1)\big) \\ &= \#\big(A_1 \cup B_1)^{\rhd\lhd}\big) - \#A_1.\end{aligned}$$

  Consequently, since $X \mapsto X^{\rhd\lhd}$ is a closure mapping:

$$\begin{aligned}\Delta(\alpha \wedge \beta, \beta) &= \#B_1 - \#(A_1 \cap B_1) \\ &= \#(A_1 \cup B_1) - \#A_1 \\ &\leq \#\big(A_1 \cup B_1)^{\rhd\lhd}\big) - \#A_1 \\ &= \Delta(\alpha, \alpha \vee \beta).\end{aligned}$$

  Hence, Δ is supermodular. ◇

All together, we now can introduce a *generalized metric* for concept lattices as follows:

Theorem 3 *Let Δ be as in (1) and $\alpha \coloneqq (A_1, A_2)$, $\beta \coloneqq (B_1, B_2) \in \mathfrak{B}\mathbb{K}$.*

Then the map

$$d\colon \mathfrak{B}\mathbb{K} \times \mathfrak{B}\mathbb{K} \longrightarrow \mathbb{N} \cup \{0\}, \quad (\alpha, \beta) \mapsto d(\alpha, \beta) \coloneqq \Delta(\alpha \wedge \beta, \beta)$$

is a GM.

Proof. This is a consequence of Proposition 1 together with Theorem 1. ◇

In our considerations we have focused on the extent. Likewise, there is also an intensional point of view for generalized metrics. In general, there are always two types of generalized metrics:

① $d_{ext}(\alpha, \beta) \coloneqq \#(B_1 - A_1)$

② $d_{int}(\alpha, \beta) \coloneqq \#(B_2 - A_2)$

6 Conclusions

- In order to compare ratings, we propose a sound directed metric in order to measure how **progressive** or **conservative** ratings are.
- **Scaling:** For chains S, S' of different size we propose an algorithmic solution.
- **Posets as target:** As target other then simply chains there is the huge class of lattices which allow for a (finite) Jordan-Dedekind chain condition together with a supermodular rank function. In particular, lower semimodular lattices of finite length will work very well. Also, our scaling algorithm based on minimal distances **extends to this class of lattices**.
- **Formal concept analysis:** Our concept of generalized metrics carries over to concept lattices, where we can cover the extensional as well as the intensional point of view.

Bibliography

[1] Aigner, M.: *Combinatorial Theory.* Springer, pp. 1-483, 1979

[2] Birkhoff, G.: *Lattice Theory.* American Mathematical Society Colloquium Publications, Vol. 25, Providence, 1967

[3] Blyth, T.S.; Janowitz, M.F.: *Residuation Theory.* Pergamon Press, pp. 1-382, 1972

[4] Carpineto, C.; Romano, G.: *Concept Data Analysis: Theory and Applications.* John Wiley & Sons, 2004

[5] Davey, B. A.; Priestly, H. A.: *Introduction to Lattices and Order* Cambridge University Press, 1990

[6] Deza, M.; Deza, E.: *Encyclopedia of Distances.* Springer, Berlin–Heidelberg, 2009

[7] Deza, M., Grishukhin, V. P.; Deza, E.: *Cones of weighted quasimetrics, weighted quasihypermetrics and of oriented cuts.* Abstracts of the International Conference "Mathematics of Distances and Applications", July 02-05, 2012, Varna, Bulgaria. ITHEA, Sofia, Bulgaria, 2012

[8] Gäbel-Hökenschnieder, T.; Pfeiffer, T.; Al Salamat, M.; Schmidt, S. E.: *Generalized metrics and their relevance for FCA and closure operators.* CLA 2016. Proceedings of the Thirteenth International Conference on Concept Lattices and Their Applications, pp. 175-188. National Research University Higher School of Economics, Russia, Moscow, Russia, 2016

[9] Ganter, B.; Wille, R.: *Formal Concept Analysis: Mathematical Foundations.* Springer-Verlag, Berlin Heidelberg, 1999

[10] Kwuida, L.; Schmidt, S. E.: *Valuations and closure operators on finite lattices.* Discrete Applied Mathematics. Vol 159, Issue 10., 2011

[11] Lengnink, K.: *Formalisierungen von Ähnlichkeit aus Sicht der Formalen Begriffsanalyse.* Shaker Verlag, Aachen, 1996

Binary Lattices

Célia Châtel, François Brucker, Pascal Préa

Aix Marseille Univ, Université de Toulon, CNRS, LIS (UMR 7020), Marseille, France
École Centrale Marseille
{celia.chatel, francois.brucker, pascal.prea}@lis-lab.fr

Abstract. A concept lattice is said to be binary if every formal concept covers at most two other concepts and is covered by at most two. These particular lattices can be seen as a generalization of decision trees (which rely on binary yes/no decisions). A non-binary lattice is binarizable if and only if it can be embedded into a binary lattice. We show in this paper that crown-free lattices are exactly binarizable ones. We also provide an algorithm which binarizes any crown-free concept lattice by adding and modifying a minimum number of concepts.

Keywords: crown-free lattice, formal concept analysis, decision systems

1 Introduction

Binary decision systems (choosing one among two possibilities) are usually more interpretable and clearer than more complex systems (choosing one among k) and for many data structures, the binary case is the standard case. For instance, in machine learning, decision trees [3] can be defined with any number of children per nodes, but one generally uses binary decision trees.

Moreover, the use of decision trees for prediction can be seen as recursively asking whether a particular individual has or not a chosen attribute hence nodes can be seen as sets of individuals sharing some attribute(s). As formal concepts are elements of concept lattices which also represent objects with common attributes, concept lattices and decision trees are strongly linked : concept lattices can be seen as a collection of overlapping decision trees [2].

We study binary lattices associated with formal contexts. By binary, we mean a concept lattice such that each formal concept covers at most two other concepts and is covered by at most two concepts. We will focus on binarizable lattices, i.e. lattices which can be embedded into a binary lattice. This is similar to the transformation of a non-binary node in a decision tree into an equivalent sequence of binary nodes. In a concept lattice, this amounts to adding some new formal concepts and modifying some existing concepts by adding some objects and/or attributes.

We show in this paper that binarizable lattices are exactly crown-free lattices. Crown-free lattices are an interesting case of lattices as they only have a polynomial number of elements, admit strong properties and a convenient graphical representation [4], [5]. These lattices are equivalent to totally balanced hypergraphs which can be seen as a generalization of trees (they are hypergraphs with no special cycle) and can be characterized by a sequence of trees [7].

This paper is organized as follows: the next section contains basic results and definitions linked to crown-free lattices and formal concept analysis. Section 3 presents our algorithm of binarization of a crown-free set system used to prove the equivalence between crow-free and binarisable lattices. Section 4 gives an illustrative example of binarization applied to formal concept analysis. Finally, Section 5 concludes and gives some topics of future research.

2 Preliminaries

In this paper, all the sets, posets and lattices are finite.

A *poset* (partially ordered set) is a pair $(A, \leq)$ such that A is a nonempty set and $\leq$ a reflexive, antisymetric, transitive binary relation on A. $(A, \geq)$ is called the *dual* of $(A, \leq)$.

Definition 1. *A poset $(A, \leq)$ can be* embedded *into a poset $(B, \leq)$ if there exists $f : A \to B$ such that for all $A_1, A_2 \in A$, $A_1 \leq A_2$ if and only if $f(A_1) \leq f(A_2)$*

In a poset $(A, \leq)$, we note $\prec$ the *covering relation* : $\forall U, V \in A$, $U \prec V$ (V *covers* U or U *is covered* by V) if and only if $U < V$ and $\nexists X \in A$, $U < X < V$. One can then represent a poset by its Hasse diagram. On such a diagram, each element of A is represented by a node and $U, V \in A$ are linked by a segment going upward if and only if $U \prec V$.

A poset $(L, \leq)$ is a lattice if $\inf\{U, V\}$ (the largest element that is smaller than or equal to U and V, written $U \wedge V$ and also called the *meet* of U and V) and $\sup\{U, V\}$ (the smallest element that is larger than or equal to U and V, written $U \vee V$ and also called the *join* of U and V) exist for all $U, V \in L$.

In formal context analysis, a *formal context* is a triplet $K = (G, M, I)$ with G a set of objects, M a set of attributes and $I \subseteq G \times M$ a binary relation.

Definition 2. *A* formal concept *associated with a formal context $K = (G, M, I)$, is a pair (A, B) with:*

- $A \subseteq G$, $B \subseteq M$
- $\{y \in M | \forall x \in A,\ xIy\} = B$
- $\{x \in G | \forall y \in B,\ xIy\} = A$

A is called an extent *and B is called an* intent*.*

The *concept lattice* $\mathcal{L}_K$ associated with a formal context K is the lattice of all concepts of the formal context with $(A_1, B_1) \leq (A_2, B_2)$ if and only if $A_1 \subseteq A_2$ and $B_2 \subseteq B_1$. An example of a concept lattice is represented in Figure 1 with blue semicircles representing attributes and black semicircles representing objects and Table 1 gives the concepts of this formal context. We will call *extent*

$(\emptyset, \{c_1, c_2, c_3, c_4, c_5, c_6, c_7, c_8\})$
$(\{\boldsymbol{r_2}\}, \{c_6, c_7, c_8\})$
$(\{\boldsymbol{r_1}\}, \{c_3, c_4, \boldsymbol{c_5}\})$
$(\{\boldsymbol{r_3}\}, \{c_2, c_3, c_4, c_6, c_8\})$
$(\{\boldsymbol{r_5}\}, \{\boldsymbol{c_1}, c_2, c_3\})$
$(\{r_2, \boldsymbol{r_4}\}, \{\boldsymbol{c_7}, c_8\})$
$(\{r_2, r_3\}, \{\boldsymbol{c_6}, c_8\})$
$(\{r_1, r_3\}, \{c_3, \boldsymbol{c_4}\})$
$(\{r_3, r_5\}, \{\boldsymbol{c_2}, c_3\})$
$(\{r_2, r_3, r_4\}, \{\boldsymbol{c_8}\})$
$(\{r_1, r_3, r_5\}, \{\boldsymbol{c_3}\})$
$(\{r_1, r_2, r_3, r_4, r_5\}, \emptyset)$

Table 1: Example of formal concepts (associated with Table 2)

	c_1	c_2	c_3	c_4	c_5	c_6	c_7	c_8
r_1			×	×	×			
r_2						×	×	×
r_3		×	×	×		×		×
r_4							×	×
r_5	×	×	×					

Table 2: Example of a cross table associated with Table 1

lattice of a concept lattice the lattice $(A, \subseteq)$ with A the set of extents of the formal concepts and *intent lattice* the lattice $(B, \subseteq)$ with B the set of intents. These two lattices are the dual of each other and the concept lattice can be seen as merging them hence working on the intent lattice or on the extent lattice is equivalent to working on the concept lattice.

The intent lattice and the extent lattice of a concept lattice are lattices whose elements are subsets of the same set. They are *set systems*.

Definition 3. *S is a* set system *on a set V if :*

- $S \subseteq 2^V$,
- *S is closed under intersection (i.e. $A \in S, B \in S \implies A \cap B \in S$),*
- *S has a minimum and a maximum element.*

Note that all the definitions given in this paper for lattices (embdedded, binary, crown-free, ...) can be extended to set systems as for a given set system S, $(S, \subseteq)$ is a lattice.

Formal contexts can also be represented as matrices, rows being the objects and columns the attributes they can have. Table 2 gives the matrix associated with the concept lattice of Figure 1 (with 1 replaced by $\times$ symbols and 0 replaced by a blank for readability purposes).

Moreover, every finite lattice $\mathcal{L} = (L, \leq)$ can be associated with a formal context $K_{\mathcal{L}}$ ([1], [6]) such that its concept lattice $\mathcal{L}_{K_{\mathcal{L}}}$ is equivalent to the initial lattice.

We will focus on a particular type of lattices : *crown-free* lattices (i.e. lattices with no crown).

Definition 4. *A* crown *is a poset $(X_1, X'_1, \ldots, X_n, X'_n)$ such that for all $i \geq 2$, $X_i < X'_{i-1}$, $X_i < X'_i$ and $X_1 < X'_n$, $X_1 < X'_1$ and there is no other comparability relation between these elements.*

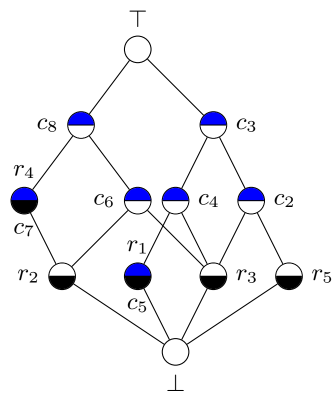
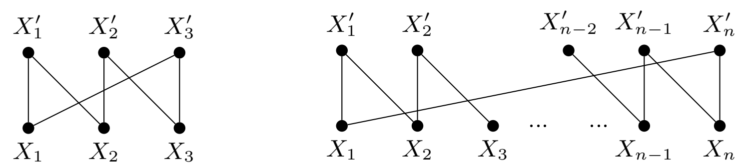


Fig. 1: Concept lattice associated with the formal context of Table 2

The Hasse diagrams of a 3-crown and of an n-crown are given in Figure 2. The concept lattice represented in Figure 1 does not contain any crown. One can remark that if $(L, \leq)$ is a 3-crown free lattice, for all $A, B, C \in L$, $A \wedge B \wedge C \in \{A \wedge B, A \wedge C, B \wedge C\}$.



(a) 3-crown (b) n-crown

Fig. 2: Hasse diagram of a crown

3 Binary lattices and crown-free lattices

In binary decision trees, each internal node has exactly two children. More generally, binary trees are trees such that each node has at most two children. Moreover the structure of tree implies that every node has only one parent. The use of a decision tree is based on the simple idea to ask successive yes/no questions about an individual in order to classify it according to the known individuals. We extend the definition of these binary structures to lattices, taking into account that elements of a lattice can be covered by more than one element. In these structures, yes/no questions about a single attribute are replaced by a question about two different attributes. An object can then have only one of the two attributes or both of them.

Definition 5. *Let $\mathcal{L} = (L, \leq)$ be a finite lattice. $\mathcal{L}$ is said to be* binary *if*

$$\forall v \in L, v \neq \bot \implies \begin{cases} |\{u \in L \mid u \prec v\}| \leq 2 \\ |\{w \in L \mid v \prec w\}| \leq 2 \end{cases}$$

If only the first condition is respected, $\mathcal{L}$ is said to be lower-binary*.*

We will characterize binary lattices and those which can be embedded into a binary lattice.

Definition 6. *Let $\mathcal{L} = (L, \leq)$ be a finite lattice. $\mathcal{L}$ is* binarizable *if and only if there exists a lattice $\mathcal{L}' = (L', \leq)$ such that $\mathcal{L}$ can be embedded in $\mathcal{L}'$ and $\mathcal{L}'$ is binary.*

We will first prove that binarizable lattices are crown-free (Proposition 8). The intuition of this result can be summarized as follows : considering an element Y covering more than three elements, in an attempt to binarize the lattice, Y would cover two elements Y' and Y'' which must be the joins of the incomparable elements they cover. We show that in a crown, $Y' = Y$ or $Y'' = Y$ which is not possible. The intuition is easy to understand on a 3-crown. For example in Figure 2, it is impossible to create the union of X_1' and X_2' without X_3'.

Property 7. Let $\mathcal{L} = (L, \leq)$ be a finite lattice and $X_1, X_1', \dots, X_n, X_n' \in L$. If $(X_1, X_1', \dots, X_n, X_n')$ is a crown of $\mathcal{L}$ then $(X_1, X_1 \vee X_2, \dots, X_n, X_n \vee X_1)$ is a crown of $\mathcal{L}$.

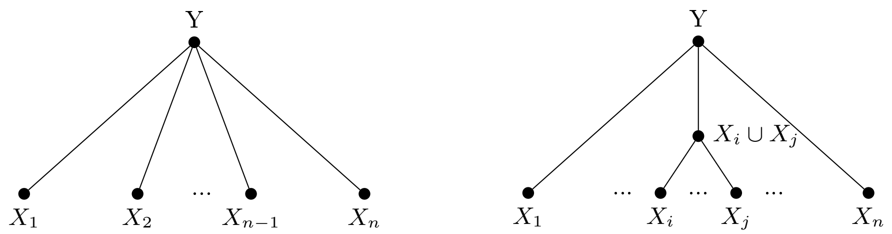
Proof. We show that $X_i \vee X_{i+1} \parallel X_j$ for all $j \neq i, i+1 \mod n$. Indeed, $X_i \parallel X_j$ and $X_{i+1} \parallel X_j$ by definition of a crown hence $X_i \vee X_{i+1} \not\leq X_j$. Moreover $X_j \parallel X_i'$ by definition of a crown. Yet for all $i \leq n$, $X_i \leq X_i'$ and $X_{i+1} \leq X_i'$ hence $X_i \vee X_{i+1} \leq X_i'$. Hence $X_j \not\leq X_i \vee X_{i+1}$. Hence $X_j \parallel X_i \vee X_{i+1}$. □

We will therefore only consider cycles of the form $(X_1, X_1 \vee X_2, \dots, X_n, X_n \vee X_1)$ in the proofs.

Proposition 8. *Let $\mathcal{L} = (L, \leq)$ be a finite lattice. If $\mathcal{L}$ is binarizable then $\mathcal{L}$ is crown-free.*

Proof. We first show that any binary lattice is crown-free by induction on the size of the crown. Suppose $(L, \leq)$ is a binary lattice containing a 3-crown. Let $(X_1, X_1 \vee X_2, X_2, X_2 \vee X_3, X_3, X_3 \vee X_1)$ a 3-crown and $Y = \sup(X_1, X_2, X_3) = \sup(X_1 \vee X_2, X_2 \vee X_3, X_3 \vee X_1)$. $(L, \leq)$ is binary so Y covers at most two elements Y' and Y'' and $\{X_i \leq Y' \cup Y''\} = \{X_1, X_2, X_3\}$. Y is the supremum of $X_1 \vee X_2, X_2 \vee X_3, X_3 \vee X_1$ and $Y' \prec Y, Y'' \prec Y$ hence, by the pigeonhole property, we can suppose without loss of generality that $X_2 \vee X_3 \leq Y'$ and $X_3 \vee X_1 \leq Y'$. Hence $X_1 < Y'$ and $X_2 < Y'$ so $X_1 \vee X_2 \leq Y'$ hence $Y' \geq Y$ which is a contradiction.

Suppose that any $(L, \leq)$ containing a crown of size inferior or equal to $n-1$ (with $n > 3$) is not binary. Let $(L, \leq)$ a binary lattice containing an n-crown. Let $(X_1, X_1 \vee X_2, \dots X_n, X_n \vee X_1)$ a crown and $Y = \sup(X_1, \dots X_n)$. $(L, \leq)$ is



(a) Initial configuration (b) Example of first step

Fig. 3: Binarization

binary hence Y covers at most two elements Y' and Y''. $Y = \sup(X_1, \ldots, X_n)$ hence $1 \leq |\{X_i \leq Y' | i \leq n\}| < n$ and $1 \leq |\{X_i \leq Y'' \mid i \leq n\}| < n$. Y is binary hence $\{X_i \leq Y' \mid i \leq n\} \cup \{X_i \leq Y'' \mid i \leq n\} = \{X_i \mid 1 \leq i \leq n\}$ so we can suppose without loss of generality that $|\{X_i \leq Y' \mid i \leq n\}| \geq 2$. Suppose moreover that $X_1 \not\leq Y'$. Let $j = \min\{i \leq n \mid X_i \leq Y'\}$ and $j' = \max\{i \leq n \mid X_i \leq Y'\}$. $(X_{j'}, X_{j'} \vee X_{j'+1 \mod n}, X_{j'+1 \mod n} \ldots, X_n, X_n \vee X_1, X_1, X_1 \vee X_2, X_2 \ldots, X_j, Y')$ is a crown of size inferior or equal to $n-1$ and superior to 3. Indeed for all $i < j$ and for all $i > j'$, $X_i \not\leq Y'$. Hence the lattice has a crown of size inferior or equal to $n-1$ hence by induction hypothesis the lattice is not binary.

By definition of embedding and of a crown, for all lattice $(L, \leq)$ containing a crown, if $(L, \leq)$ can be embedded in $(L', \leq)$ by a function f, then $(X_1, X_1', \ldots X_n, X_n')$ is a crown in $(L, \leq)$ if and only if $(f(X_1), f(X_1'), \ldots, f(X_n), f(X_n'))$ is a crown in $(L', \leq)$. By the previous result, if $(L', \leq)$ has a crown then $(L', \leq)$ is not binary hence if $(L, \leq)$ has a crown, $(L, \leq)$ is not binarizable.

In this proof we only used the fact that for all Y in a binary lattice, Y covers at most two elements. If Y is covered by more than 3 elements, the same proof can be applied to the dual of the lattice in which Y covers more than 3 elements. □

We will now show that any crown-free lattice can be embedded in a binary lattice (Proposition 14). We will work on set systems associated with lattices and use Algorithm 1 in order to transform any crown-free set system into a lower-binary set system. Each non-binary element is transformed into a binary one by creating unions of some of the elements it covers. In order to keep the closure under intersection of the set system, the elements used to created the new elements have to be chosen wisely.

Definition 9. *Let $\{X_1, \ldots, X_n\}$ be a set of incomparable subsets of the same set. X_i and X_j are said to be of* maximal intersection *among $\{X_1, X_2, \ldots X_n\}$ if and only if there does not exist $k \neq i, j$ such that $X_i \cap X_j \subsetneq X_i \cap X_k$ or $X_i \cap X_j \subsetneq X_j \cap X_k$.*

One step of the process is illustrated in Figure 3.

Algorithm 1: Lower-binarization of a crown-free set system

Data: S a set system

Result: $\mathcal{B}(S)$ a set system such that every element covers at most two other elements and S can be embedded into $\mathcal{B}(S)$

$\mathcal{B}(S) = S$
for $Y \in S$ **do**
  $C = \{X \in \mathcal{B}(S) \mid X \prec Y\}$
  while $|C| > 2$ **do**
    Find X_i, X_j of maximal intersection among C
    $\mathcal{B}(S) = \mathcal{B}(S) \cup \{X_i \cup X_j\}$
    $C = C \cup \{X_i \cup X_j\} \backslash \{X_i, X_j\}$
  return $\mathcal{B}(S)$

The process adds as few elements as possible to the set system to make it binary. Indeed, if an element Y covers k elements $X_1, \ldots, X_k$, our construction adds exactly $k-2$ elements.

The following technical lemmas will be used to prove Proposition 13. Note that all these lemmas only require the set system to have no 3-crowns and not to be crown-free.

Lemma 10. *Let L be a set system with no 3-crown and $\{X_1, \ldots, X_n\} \subset L$ a set of incomparable elements of L.*

Let X_i and X_j of maximal intersection among $\{X_1, \ldots X_n\}$.

Then,

$$\forall l \in \{1, \ldots, n\}, X_l \cap (X_i \cup X_j) = \begin{cases} X_i \cap X_l \\ or \\ X_j \cap X_l \end{cases}$$

Proof. Let X_i, X_j of maximal intersection among $\{X_1, \ldots, X_n\}$.

L has no 3-crowns, so $X_i \cap X_j \cap X_l \in \{X_i \cap X_j, X_i \cap X_l, X_j \cap X_l\}$.

X_i and X_j are of maximal intersection so $X_i \cap X_j \not\subseteq X_l \cap X_i$ and $X_i \cap X_j \not\subseteq X_l \cap X_j$. Hence $X_i \cap X_j \cap X_l \in \{X_i \cap X_l, X_j \cap X_l\}$ i.e. $X_j \cap X_l \subseteq X_i \cap X_l$ or $X_i \cap X_l \subseteq X_j \cap X_l$. Yet

$$\forall l, X_l \cap (X_i \cup X_j) = (X_i \cap X_l) \cup (X_j \cap X_l)$$

Hence

$$\forall l, X_l \cap (X_i \cup X_j) = \begin{cases} X_i \cap X_l \\ or \\ X_j \cap X_l \end{cases} \iff \begin{cases} X_j \cap X_l \subseteq X_i \cap X_l \\ or \\ X_i \cap X_l \subseteq X_j \cap X_l \end{cases}$$

□

Lemma 11. *Let L be a set system with no 3-crown and $\{X_1, \ldots, X_n\} \subset L$ the set of elements covered by $Y \in L$. Taking X_i, X_j, two elements of maximal intersection among $\{X_1, \ldots X_n\}$:*

$$\forall Z \in L \backslash \{X_1, \ldots, X_n, Y\}, Z \cap (X_i \cup X_j) \in L \cup \{X \cup Y\}.$$

Proof. If $Z \cap X_i \subseteq X_j$ then $Z \cap (X_i \cup X_j) = Z \cap X_j$. As L is closed under intersection, $Z \cap X_j \in L$. The same goes for X_j.
Suppose now $Z \cap X_i \backslash \{X_j\} \neq \emptyset$ and $Z \cap X_j \backslash \{X_i\} \neq \emptyset$. Hence $X_i \cap X_j \subseteq Z$, as L has no 3-crowns. So $X_i \cap X_j \subseteq Z \cap Y$.
If $Y \subseteq Z$, the result is obvious. If $Z \parallel Y$ or $Z \subsetneq Y$ then $Z \cap Y \subsetneq Y$ so there exists k such that $Z \cap Y \subseteq X_k$ as Y exactly covers the elements $(X_1, \ldots, X_n)$. $X_i \cap X_j \subseteq Z \cap Y \subseteq X_k$ and X_i, X_j are of maximal intersection so $k = i$ or $k = j$. Hence $Z \cap Y \subseteq X_i$ (or symmetrically $Z \cap Y \subseteq X_j$) which leads to $Z \cap Y \subseteq Z \cap X_i$ so $Z \cap Y \subseteq Z \cap (X_i \cup X_j)$.
Moreover as $X_i \cup X_j \subseteq Y$, $Z \cap (X_i \cup X_j) \subseteq Z \cap Y$, by double inclusion $Z \cap (X_i \cup X_j) = Z \cap Y$. Yet $Z \cap Y \in L$ as L is closed under intersection, which completes the proof. □

Lemma 12. *Let L be a set system with no 3-crown and $\{X_1, \ldots, X_n\} \in L$ the set of elements covered by $Y \in L$. For all X_i, X_j of maximal intersection among $\{X_1, \ldots, X_n\}$, all elements of $\{X_1, \ldots X_n\} \cup \{X_i \cup X_j\} \setminus \{X_i, X_j\}$ are incomparable.*

Proof. We prove that for all $k \neq i, j$, $X_i \cup X_j \parallel X_k$.
As, for all $k \neq i, j$, X_i, X_j and X_k are incomparable, $X_i \nsubseteq X_k$ and $X_j \nsubseteq X_k$ hence $X_i \cup X_j \nsubseteq X_k$.
As L has no 3-crown, $X_i \cap X_j \cap X_k \in \{X_i \cap X_j, X_i \cap X_k, X_j \cap X_k\}$ so as X_i and X_j are of maximal intersection, $X_i \cap X_j \not\subset X_i \cap X_k$ and $X_i \cap X_j \not\subset X_j \cap X_k$. So $X_i \cap X_j \cap X_k \in \{X_i \cap X_k, X_j \cap X_k\}$. Suppose $X_i \cap X_j \cap X_k = X_i \cap X_k$. We then have $X_i \cap X_k \subseteq X_j$. Moreover, as $(X_1, \ldots X_n)$ are incomparable, there exists $x \in X_k$ such that $x \notin X_j$. $X_i \cap X_k \subseteq X_j$ hence $x \notin X_i$. So there exists $x \in X_k, x \notin X_i \cup X_j$ so $X_k \nsubseteq X_i \cup X_j$. □

Proposition 13. *Algorithm 1 applied on a set system S returns a set system $\mathcal{B}(S)$ such that:*

- *for all $X \in \mathcal{B}(S)$, X covers at most two elements,*
- *$S \subseteq \mathcal{B}(S)$,*
- *for all $X \in S$, X is covered by the same number of elements in S and in $\mathcal{B}(S)$.*

Proof. By Lemma 10 and 11, creating new elements as described in the construction preserves the closure under intersection of the system. Lemma 10 shows that the intersection of any element of the considered part of the system and the new element is already in the system and Lemma 11 shows the same for other elements. Moreover, the minimum element and the maximum element are unchanged as the algorithm only adds unions of two elements hence the system is a set system. Lemma 12 shows that if X_i and X_j are chosen from the set of incomparable elements $\{X_1, \ldots, X_n\}$ covered by an element Y to create a new element, the only changes in the covering relation of the system are $X_i \cup X_j \prec Y$, $X_i \prec X_i \cup X_j$ and $X_j \prec X_i \cup X_j$. Hence, no new crown is added and the set system is still crown-free. Moreover, for $k = i, j$, X_k is not anymore covered by

Y but by $X_i \cup X_j$ which keeps unchanged the number of covering elements of X_k. The other elements are unchanged. Lemma 12 also proves that the process can be iterated if the set system still has no 3-crowns.

The algorithm ends when all elements of the initial system cover at most two elements. Moreover, the elements added by the construction cover exactly two elements by construction. □

Proposition 14. *Let $\mathcal{L} = (L, \leq)$ be a finite lattice. If $\mathcal{L}$ is crown-free then $\mathcal{L}$ is binarizable.*

Proof. Let $(A, \subseteq)$ be the extent lattice of $K_{\mathcal{L}}$ (the formal context associated with $\mathcal{L}$). Algorithm 1 can be applied on the set system A. By Proposition 13, the obtained set system $\mathcal{B}(A)$ is such that $A \subseteq \mathcal{B}(A)$ and $\mathcal{B}(A)$ is lower-binary. Moreover $\mathcal{L}$ can trivially be embedded in the lattice $\mathcal{L}' = (\mathcal{B}(A), \subseteq)$.

Let $(B, \subseteq)$ be the intent lattice of $\mathcal{L}'$. By Proposition 13, $\mathcal{B}(B)$, the result of Algorithm 1 applied on B, is binary. Finally, $\mathcal{L}'$ can be embedded in the binary lattice $\mathcal{L}'' = (\mathcal{B}(B), \subseteq)$ so $\mathcal{L}$ can also be embedded into $\mathcal{L}''$ by transitivity of the embedding. □

Propositions 8 and 14 prove the main result of this paper.

Theorem 15. *Let $(L, \leq)$ be a finite lattice. $(L, \leq)$ is binarizable if and only if $(L, \leq)$ is crown-free.*

4 Example

We will now apply our binarization algorithm on a small concept lattice. Table 3a gives a small formal context describing some animals to apply our algorithm. The formal concepts associated with this context are given in Table 3b. The representation of the Hasse diagram of the concept lattice associated with this formal context (Figure 4) gives an easy way to see non-binary formal concepts.

Here, the concept representing the *duck* is covered by three concepts and the concept representing the attribute *swim* covers three concepts. This representation also shows that the lattice is crown-free so it it binarizable and our algorithm can be applied.

A binarization of the set system associated with the extent lattice of this concept lattice gives the Hasse diagram given in Figure 5a with the new latent node (a new attribute) represented as a rectangle. We deliberately ignore the fact that the element $\perp$ is not binary as binarizing it would not increase interpretability or help using the model in machine learning. The Hasse diagram associated with a binarization of the extents and of the intents is represented in Figure 5b and is associated with the formal context of Table 4. The process creates two new formal concepts : one associated with a new object *obj* (which could be interpreted as the existence of a bird eating seeds, for example a canary) and one associated with a new attribute *att* (which suggests the existence of an attribute allowing to distinguish salmon, shark, barracuda and crocodile from frog and duck) and modifies

	scale	teeth	swim	fly	seed	feather	air
salmon	×		×				
shark		×	×				
barracuda	×	×	×				
frog			×				×
crocodile		×	×				×
eagle				×		×	×
ostrich					×	×	×
duck			×	×	×	×	×

(a) Cross-table of animals data

({∅}, {scale, teeth, swim, fly, seed, feather, air})
({*crocodile*}, {teeth, swim, air})
({*duck*}, {swim, fly, seed, feather, air})
({*barracuda*}, {scale, teeth, swim})
({*ostrich*, duck}, {*seed*, feather, air})
({*salmon*, barracuda}, {*scale*, swim})
({*eagle*, duck}, {*fly*, feather, air})
({*shark*, barracuda, crocodile}, {*teeth*, swim})
({*frog*, crocodile, duck}, {air, swim})
({eagle, ostrich, duck}, {*feather*, air})
({frog, crocodile, eagle, ostrich, duck}, {*air*})
({salmon, shark, barracuda, frog, crocodile}, {*swim*})

(b) Formal concepts of animals formal context

Table 3: Animal example formal context

	scale	teeth	**att**	swim	fly	seed	feather	air
salmon	×		×	×				
shark		×	×	×				
barracuda	×	×	×	×				
frog				×				×
crocodile		×	×	×				×
eagle					×		×	×
ostrich						×	×	×
obj					×	×	×	×
duck				×	×	×	×	×

Table 4: Binarized formal context

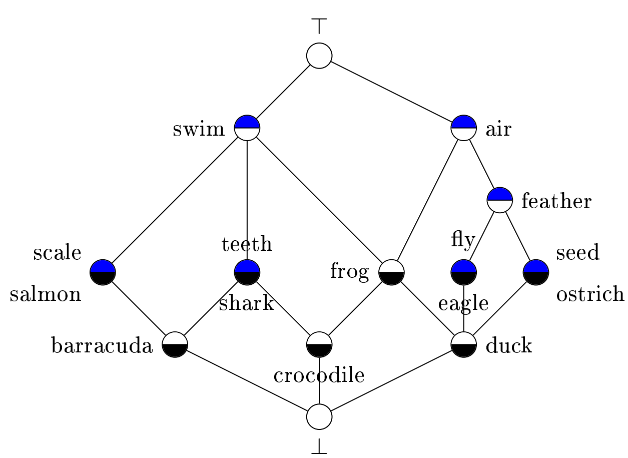
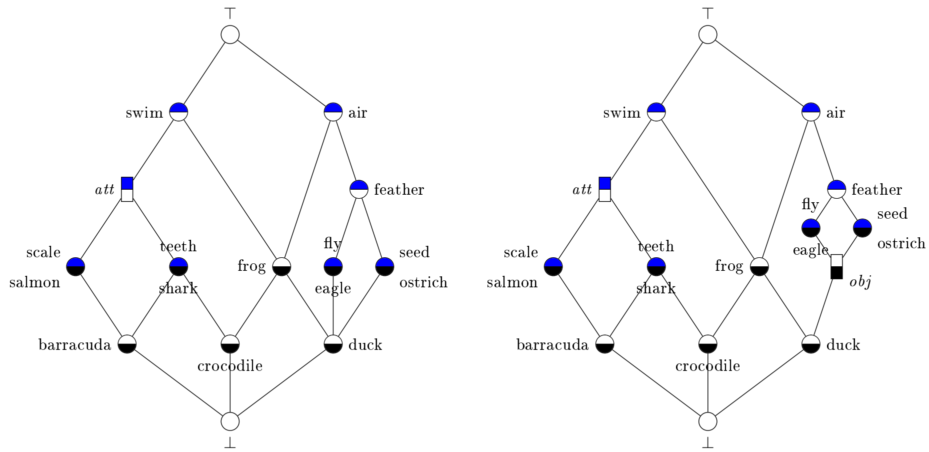


Fig. 4: Hasse diagram of the lattice associated with Table 3b



(a) Binarization of the extents

(b) Binarization of the extents and the intents

Fig. 5: Hasse diagram of binarization of the lattice associated with Table 3b

other formal concepts using this new attribute and object. The binarization process does not give a unique solution. Indeed, *att* could have been associated with a new concept $\{(shark, frog, barracuda, crocodile, duck), (teeth, att, swim)\}$ instead of $\{salmon, shark, barracuda, crocodile), (scale, att, swim)\}$ as the two intersections concerned are incomparable. This allows to make interactive systems giving the user different choices to binarize each concept. The model can then be used in the same way a decision tree is used : beginning from the top of the lattice, a new element is propagated in the nodes asking at each node whether it has the attribute of the right child or of the left child of the current node beforce classifying it as close to some known concept.

5 Conclusion

We presented a simple and efficient algorithm to transform a crown-free set system into a binary one. This construction allows us to prove the equivalence between binary lattices and crown-free ones. Our algorithm can be easily used in formal context analysis in order to modify a concept lattice to obtain a binary one, adding some objects and attributes. Moreover, our algorithm can be independently applied on the intents only or on the extents only or on both. The system being binary, it is easy to interpret and understand. The equivalence between binary lattices and crown-free ones makes of crown-free lattices a perfect candidate to extend machine learning ideas developped in decision trees to more complex systems. Indeed, it can then be used in machine learning to predict the class of a given object by propagating it in the concept lattice recursively asking whether the object has or not the attribute represented by the predecessor of the concept in the lattice. Moreover, the binarization we proposed is not unique but always adds the same number of elements to the lattice, which allows the development of interactive systems to build the model. Topics for future work include :

- a top-down process to build binary lattices inspired from decision trees,
- machine learning applications of lattice structures,
- study of the interest of intersecting classes in the machine learning perspective and the classification one.

References

1. Marc Barbut and Bernard Monjardet. *Ordre et classification: algèbre et combinatoire*. Hachette, 1970. Google-Books-ID: n3BpSgAACAAJ.
2. Radim Belohlavek, Bernard De Baets, Jan Outrata, and Vilem Vychodil. Inducing decision trees via concept lattices. *International Journal of General Systems*, 38(4):455–467, May 2009.
3. Leo Breiman, Jerome H. Friedman, Richard A. Olshen, and Charles J. Stone. *Classification and regression trees*. 1984.
4. François Brucker and Alain Gély. Parsimonious cluster systems. *Advances in Data Analysis and Classification*, 3(3):189–204, November 2009.
5. François Brucker and Pascal Préa. Totally Balanced Formal Context Representation. In Jaume Baixeries, Christian Sacarea, and Manuel Ojeda-Aciego, editors, *Formal Concept Analysis*, number 9113 in Lecture Notes in Computer Science, pages 169–182. Springer International Publishing, June 2015.
6. Bernhard Ganter and Rudolf Ville. *Formal Concept Analysis - Mathematical Foundations*. Springer edition, 1997.
7. Jenö Lehel. A characterization of totally balanced hypergraphs. *Discrete Mathematics*, 57(1–2):59–65, November 1985.

Biclustering Based on FCA and Partition Pattern Structures for Recommendation Systems

Nyoman Juniarta[1], Victor Codocedo[2], Miguel Couceiro[1], and Amedeo Napoli[1]

[1] Université de Lorraine, CNRS, Inria, LORIA, F-54000 Nancy, France
{nyoman.juniarta, miguel.couceiro, amedeo.napoli}@loria.fr
[2] Universidad Técnica Federico Santa María, Santiago, Chile
victor.codocedo@inria.cl

Abstract. This paper focuses on item recommendation for visitors in a museum within the framework of European Project CrossCult about cultural heritage. We present a theoretical research work about recommendation using biclustering. Our approach is based on biclustering using FCA and partition pattern structures. First, we recall a previous method of recommendation based on constant-column biclusters. Then, we propose an alternative approach that incorporates an order information and that uses coherent-evolution-on-columns biclusters. This alternative approach shares some common features with sequential pattern mining. Finally, given a dataset of visitor trajectories, we indicate how these approaches can be used to build a collaborative recommendation strategy.

Keywords: biclustering, FCA, pattern structures, recommendation

1 Introduction

CrossCult (http://www.crosscult.eu) is a European project whose idea is to support the emergence of a European cultural heritage by allowing visitors in different cultural sites (e.g. museum, historic city, archaeological site) to improve the quality of their visit by using adapted computer-based devices and to consider the visit at a European level. Such improvement can be accomplished by studying, among others, the possibility to build a dynamic recommendation system. This system should be able to produce a relevant suggestion on which part of a cultural site may be interesting for a specific visitor.

Here, our objective is to study a dynamic recommendation system for visitors in a museum. Given a new visitor V_n, the task is to suggest a museum item that may be interesting for him/her. Based on how a suggestion is made to a new visitor V_n, a recommendation system can be classified into one of the three following categories [1]:

- *Content-based recommendations*: The system makes a suggestion based only on the previous visited items of V_n. For example, if V_n visited mostly the items from prehistoric era, then the system recommends another item from that era.

– *Collaborative recommendations*: The system looks for previous users who have similar interest to V_n, and makes a suggestion based on their visited items. For example, if many of V_n's similar users have visited item I, then the system recommends this item.
– *Hybrid approaches*: The combination of content-based and collaborative approaches.

Our method belongs to the second category (collaborative recommendation). First we group all previous users based on their visit trajectories using biclustering. When V_n arrives, we try to find a G_s, i.e. a group of visitors who shares a similar interest to V_n. Then, based on the behavior of the visitors in G_s, we can suggest one item that may be interesting for V_n.

In this paper we will recall an approach in [6] that uses partition pattern structures to obtain biclusters with constant (or similar) values on the columns. Then we will propose an alternative approach that relies on this approach to mine another type of biclusters: those with coherent evolution on the columns (`CEC` biclusters). This bicluster type is useful when we are dealing with a dataset of trajectories where each trajectory corresponds to an ordered list of items. Furthermore, the mining of `CEC` biclusters can be related to sequential pattern mining, which we will explore in this paper.

This paper is organized as follows. First, we mention some related works about recommendation in Section 2. Then the basic background on biclustering is given in Section 3. Section 4 explains how to perform biclustering using partition pattern structures. The application of biclustering to recommendation systems will be presented in Section 5. Finally, we conclude our paper and outline some future works in Section 6.

2 Related work

In this section, we will describe related work about recommendation systems, biclustering, and Formal Concept Analysis (FCA).

FCA has been studied in collaborative movie recommendations for a user by looking at the ratings given by other users. In [5], FCA is used to generate a lattice from a binary matrix (with users as rows and movies as columns) as the formal context. This matrix is derived from a rating dataset which is binarized, such that the matrix contains only the information whether a user has rated a movie. The lattice is then drawn to select some neighbors – i.e. users who have rated the same movies as the new user – regardless of the rating values. In this way, the exhaustive search of neighbors can be avoided. The neighbors' ratings can be then studied to recommend movies rated by the neighbors but not yet rated by the new user.

Pattern structures [8,13] are a generalization of FCA, where the objects have more complex descriptions (e.g. sequence, graph, etc.). FCA was also extended into Triadic Concept Analysis, and it was shown in [13] that triadic concepts are in 1-1-correspondence with maximal biclusters of similar values.

Partition pattern structures are an instance of the pattern structure framework. They were used in a collaborative movie recommendation [7] by identifying similar-column biclusters within the rating matrix. Such a bicluster corresponds to a set of users with similar rating behavior (hence similar interest) across a set of movies. Therefore, to recommend a movie to a new user, there is a search for biclusters whose users have similar interest to him/her. Then, a recommendation is given by looking at the movies rated by users in the biclusters but not yet rated by the new user. Using the real MovieLens data, a study was also conducted based on Boolean matrix factorization [2].

Moreover, recommendation systems based on FCA and/or biclustering have been applied to other real world problems such as detection of future advertising terms for a company [10], educational orientation of Russian school graduates [11], and idea recommendation at a crowdsourcing project of Witology company [9]. Furthermore, a unified taxonomy of biclustering methods was proposed in [12].

3 Biclustering

In this section, we will recall the basic background and discuss illustrative examples of the different types of biclusters as described in [14]. We consider a dataset composed of a set of objects, each of which has values over a set of attributes. This dataset can be represented as a numerical matrix, where each cell ij indicates the value of object i w.r.t. attribute j.

One may be interested in finding which subset of objects possesses the same values w.r.t. a subset of attributes. Regarding the matrix representation, this is equivalent to the problem of finding a submatrix that has a constant value over all of its elements (example in Table 1). This task is called biclustering with constant values, which is a simultaneous clustering of the rows and columns of a matrix.

Table 1. A bicluster with constant value (shaded)

1	1	4	3	5
1	1	2	5	1
3	3	4	2	1

Other than constant values, the bicluster approach also focused on finding other types of submatrices, as shown in Table 2. A bicluster with constant columns (rows) is a submatrix where each column (row) has the same value, as illustrated in Table 2a (Table 2b, resp.).

In a bicluster with additive coherent values, the value of each cell ij follows the equation $\gamma + \alpha_i + \beta_j$, where γ is a constant, α_i is a constant value for row i, and β_j is a constant value for column j. For example, if $\gamma = 1$, $(\alpha_1, \alpha_2, \alpha_3, \alpha_4) = (3, 2, 4, 6)$, and $(\beta_1, \beta_2, \beta_3, \beta_4) = (0, -2, 1, -1)$, then we can obtain the bicluster

Table 2. Examples of some types of biclusters. (a) Constant columns, (b) constant rows, (c) additive coherent values, (d) multiplicative coherent values, (e) coherent evolution on the columns, and (f) coherent evolution on the rows.

(a)

4	2	5	3
4	2	5	3
4	2	5	3
4	2	5	3

(b)

1	1	1	1
2	2	2	2
4	4	4	4
3	3	3	3

(c)

4	2	5	3
3	1	4	2
5	3	6	4
7	5	8	6

(d)

4	2	5	3
2	1	2.5	1.5
8	4	10	6
6	3	7.5	4.5

(e)

1	2	4	3
3	5	7	6
2	3	8	4
4	5	9	8

(f)

1	2	4	3
0	1	1	2
5	4	6	4
6	5	7	5

in Table 2c. Similarly, we can obtain a bicluster with multiplicative coherent values as shown in Table 2d using a constant for each row and each column. The main difference is that, instead of adding, we multiply them.

Another interesting type is the `CEC` bicluster, also known as order-preserving submatrix [4]. In this type of bicluster, each row induces the same linear order across all columns. For example, in the bicluster in Table 2e, each row follows *column1* $\leq$ *column2* $\leq$ *column4* $\leq$ *column3*. Moreover, a bicluster with coherent evolution on the rows can be defined similarly, as shown in Table 2f.

Those different types of biclusters are useful when we are interested in identifying a group of people who behave similarly according to a set of attributes. This group identification is necessary in the task of collaborative recommendation, because in the process of making a suggestion to a person, we first identify the people who are similar to him/her. As suggested in [3] the bicluster-based recommendation may give a better performance than state-of-the-art recommendation algorithms.

4 Biclustering Using Partition Pattern Structures

Biclustering has many common elements with FCA. In FCA, from a binary matrix we try to find a maximal submatrix whose elements are 1. In other words, the objective is to identify maximal constant-value biclusters (but only for biclusters whose values are 1). Hence, a formal concept can be considered as a bicluster of objects and attributes. Furthermore, formal concepts are arranged in a concept lattice, that can describe the hierarchical relation among all biclusters.

Consider the matrix given by Table 3, where we are interested in finding constant-column biclusters. We recognize that the values of $\mathtt{m}_1$ "break" the objects into two sets: $\{\mathtt{g}_1, \mathtt{g}_2\}$ and $\{\mathtt{g}_3, \mathtt{g}_4\}$. The same "break" is also obtained from the values of $\mathtt{m}_4$. In particular, we can see that the pair $(\{\mathtt{g}_1, \mathtt{g}_2\}, \{\mathtt{m}_1, \mathtt{m}_4\})$ corresponds to a constant-column bicluster. Therefore, it is possible to mine this type of bicluster using this "breaking" – or "partitioning" – technique. Moreover, this technique can be performed using partition pattern structures – an extension of FCA.

In this section, first we will recall the constant-column biclustering approach using partition pattern structures [6]. After that, we will explain the possibility to perform `CEC` biclustering based on this approach.

4.1 Biclustering with Constant Columns

A partition $\mathtt{d} = \{p_i\}$ of a set $\mathtt{G}$ is a collection of $p_i \subseteq \mathtt{G}$ such that:

$$\bigcup_{p_i \in \mathtt{d}} p_i = \mathtt{G} \quad \text{and} \quad p_i \cap p_j = \emptyset \quad \text{whenever} \quad i \neq j. \tag{1}$$

Notice that when calculating the initial partitions, missing values can produce overlapping partitions (i.e. $p_i \cap p_j \neq \emptyset$). Consider the dataset given by Table 3 that has $\mathtt{G} = \{\mathtt{g}_1, \mathtt{g}_2, \mathtt{g}_3, \mathtt{g}_4\}$ as the set of objects and $\mathtt{M} = \{\mathtt{m}_1, \mathtt{m}_2, \mathtt{m}_3, \mathtt{m}_4, \mathtt{m}_5\}$ as the set of attributes. Here we can define a partition mapping $\delta : \mathtt{M} \rightarrow \mathtt{D}$. The partition is based on the fact that the values of the attribute are equal for all objects in a subset. For example, $\delta(\mathtt{m}_1) = \{\{\mathtt{g}_1, \mathtt{g}_2\}, \{\mathtt{g}_3, \mathtt{g}_4\}\}$ because $\mathtt{G}$ is partitioned as such regarding the value of $\mathtt{m}_1$, whereas $\delta(\mathtt{m}_4) = \{\{\mathtt{g}_1, \mathtt{g}_2\}, \{\mathtt{g}_1, \mathtt{g}_3, \mathtt{g}_4\}\}$. This partition overlaps on $\mathtt{g}_1$ since this object has a missing value on $\mathtt{m}_4$. Intuitively, $\mathtt{g}_1$ can be grouped with either $\{\mathtt{g}_2\}$ or $\{\mathtt{g}_3, \mathtt{g}_4\}$ w.r.t. $\mathtt{m}_4$.

Table 3. A dataset with 4 objects and 5 attributes

	$\mathtt{m}_1$	$\mathtt{m}_2$	$\mathtt{m}_3$	$\mathtt{m}_4$	$\mathtt{m}_5$
$\mathtt{g}_1$	1	5	3	?	7
$\mathtt{g}_2$	1	1	4	2	7
$\mathtt{g}_3$	2	5	4	5	3
$\mathtt{g}_4$	2	5	4	5	7

The space $\mathtt{D}$ of all partitions over $\mathtt{G}$ is a complete lattice, where the meet and join of two partitions $\mathtt{d}_1 = \{p_i\}$ and $\mathtt{d}_2 = \{p_j\}$ are defined as:

$$\mathtt{d}_1 \sqcap \mathtt{d}_2 = \left(\bigcup_{i,j} p_i \cap p_j \right)^+ \tag{2}$$

$$\mathtt{d}_1 \sqcup \mathtt{d}_2 = \left(\bigcup_{p_i \cap p_j \neq \emptyset} p_i \cup p_j \right)^+ \tag{3}$$

where $(.)^+$ is a closure that preserves only the maximal components in $\mathtt{d}$. For example, $\delta(\mathtt{m}_1) \sqcap \delta(\mathtt{m}_4) = \{\{\mathtt{g}_1, \mathtt{g}_2\}, \{\mathtt{g}_1\}, \{\mathtt{g}_3, \mathtt{g}_4\}\}^+ = \{\{\mathtt{g}_1, \mathtt{g}_2\}, \{\mathtt{g}_3, \mathtt{g}_4\}\}$, and $\delta(\mathtt{m}_1) \sqcup \delta(\mathtt{m}_4) = \{\{\mathtt{g}_1, \mathtt{g}_2\}, \{\mathtt{g}_1, \mathtt{g}_2, \mathtt{g}_3, \mathtt{g}_4\}, \{\mathtt{g}_1, \mathtt{g}_3, \mathtt{g}_4\}\}^+ = \{\{\mathtt{g}_1, \mathtt{g}_2, \mathtt{g}_3, \mathtt{g}_4\}\}$.

The order between any two partitions is given by the subsumption relation:

$$\mathtt{d}_1 \sqsubseteq \mathtt{d}_2 \Longleftrightarrow \mathtt{d}_1 \sqcap \mathtt{d}_2 = \mathtt{d}_1 \tag{4}$$

Given a set of attributes $\mathtt{M}$, a partition space $\mathtt{D}$, and a mapping δ, a partition pattern structures for constant-column biclustering is determined by the triple

$(\mathtt{M}, (\mathtt{D}, \sqcap), \delta)$. A pair $(\mathtt{A}, \mathtt{d})$ is then called a partition pattern concept (pp-concept) iff $\mathtt{A}^\square = \mathtt{d}$ and $\mathtt{d}^\square = \mathtt{A}$, where:

$$\mathtt{A}^\square = \prod_{\mathtt{m} \in \mathtt{A}} \delta(\mathtt{m}) \qquad \mathtt{A} \subseteq \mathtt{M} \tag{5}$$

$$\mathtt{d}^\square = \{\mathtt{m} \in \mathtt{M} | \mathtt{d} \sqsubseteq \delta(\mathtt{m})\} \qquad \mathtt{d} \in \mathtt{D} \tag{6}$$

For any partition component $p \in \mathtt{d}$, each pair $(p, \mathtt{A})$ corresponds to a constant-column bicluster. For example, from the concept $(\{\mathtt{m}_1, \mathtt{m}_4\}, \{\{\mathtt{g}_1, \mathtt{g}_2\}, \{\mathtt{g}_3, \mathtt{g}_4\}\})$, two biclusters can be obtained: $(\{\mathtt{g}_1, \mathtt{g}_2\}, \{\mathtt{m}_1, \mathtt{m}_4\})$ and $(\{\mathtt{g}_3, \mathtt{g}_4\}, \{\mathtt{m}_1, \mathtt{m}_4\})$.

4.2 Biclustering with Coherent Evolution on the Columns

In a dataset of movie ratings, bicluster with constant columns is useful to identify a set of users with the same taste regarding a set of movies. Another interesting problem arises, e.g. when the dataset contains watching order. In that case, we may be interested in finding a set of users who watch a set of movies in the same order. This problem corresponds to CEC biclustering, where the objective is to find a set of rows which has coherent evolution over a set of columns, as previously described in Section 3. In the current section, we will explain the possible application of partition pattern structures to discover CEC biclusters.

Table 4. A dataset with 5 objects and 5 attributes

	$\mathtt{m}_1$	$\mathtt{m}_2$	$\mathtt{m}_3$	$\mathtt{m}_4$	$\mathtt{m}_5$
$\mathtt{g}_1$	1	2	3	4	5
$\mathtt{g}_2$	4	2	1	?	3
$\mathtt{g}_3$	2	3	4	1	1
$\mathtt{g}_4$	5	4	2	3	1
$\mathtt{g}_5$	2	1	5	4	3

Consider the dataset given by Table 4, with the set of attributes $\mathtt{G} = \{\mathtt{g}_1, \mathtt{g}_2, \mathtt{g}_3, \mathtt{g}_4, \mathtt{g}_5\}$. First, we have to list each pair of attributes and the partition according to the pair's evolution. For the pair $\mathtt{p}_{1,2} = (\mathtt{m}_1, \mathtt{m}_2)$, the partition is $\{\{\mathtt{g}_1, \mathtt{g}_3\}, \{\mathtt{g}_2, \mathtt{g}_4, \mathtt{g}_5\}\}$ because in $\mathtt{g}_1$ and $\mathtt{g}_3$, $\mathtt{m}_1$ is less than $\mathtt{m}_2$, whereas in $\mathtt{g}_2$, $\mathtt{g}_4$, and $\mathtt{g}_5$, $\mathtt{m}_1$ is greater. As in Subsection 4.1, missing values generate an overlapping partition. For instance, the pair $\mathtt{p}_{1,4}$ gives rise to the partition is $\{\{\mathtt{g}_1, \mathtt{g}_2, \mathtt{g}_5\}, \{\mathtt{g}_2, \mathtt{g}_3, \mathtt{g}_4\}\}$. Furthermore, two columns with the same value (e.g. $\mathtt{g}_3$ in $\mathtt{m}_4$ and $\mathtt{m}_5$) can also produce an overlapping partition because, by our definition of CEC bicluster in Section 3, they satisfy $\mathtt{m}_4 \leq \mathtt{m}_5$ and $\mathtt{m}_5 \leq \mathtt{m}_4$. Therefore, the partition for $\mathtt{p}_{4,5}$ is $\{\{\mathtt{g}_1, \mathtt{g}_2, \mathtt{g}_3\}, \{\mathtt{g}_2, \mathtt{g}_3, \mathtt{g}_4, \mathtt{g}_5\}\}$. Some pairs and their partitions are listed in Table 5.

Since a CEC partition is defined by at least two attributes, the partition mapping becomes $\gamma : \mathtt{P} \rightarrow \mathtt{D}$. For instance, $\gamma(\mathtt{p}_{1,2}) = \{\{\mathtt{g}_1, \mathtt{g}_3\}, \{\mathtt{g}_2, \mathtt{g}_4, \mathtt{g}_5\}\}$.

Table 5. Some examples of partitions over Table 4

Pair	Partition
$\mathtt{p}_{1,2}$	$\{\{\mathtt{g}_1, \mathtt{g}_3\}, \{\mathtt{g}_2, \mathtt{g}_4, \mathtt{g}_5\}\}$
$\mathtt{p}_{1,3}$	$\{\{\mathtt{g}_1, \mathtt{g}_3, \mathtt{g}_5\}, \{\mathtt{g}_2, \mathtt{g}_4\}\}$
$\mathtt{p}_{1,4}$	$\{\{\mathtt{g}_1, \mathtt{g}_2, \mathtt{g}_5\}, \{\mathtt{g}_2, \mathtt{g}_3, \mathtt{g}_4\}\}$
$\mathtt{p}_{2,3}$	$\{\{\mathtt{g}_1, \mathtt{g}_3, \mathtt{g}_5\}, \{\mathtt{g}_2, \mathtt{g}_4\}\}$
$\mathtt{p}_{2,5}$	$\{\{\mathtt{g}_1, \mathtt{g}_2, \mathtt{g}_5\}, \{\mathtt{g}_3, \mathtt{g}_4\}\}$

As in Subsection 4.1, given a set of attribute pairs $\mathtt{P}$, a partition space $\mathtt{D}$, and the mapping function γ, a partition pattern structures for coherent-evolution biclustering is determined by the triple $(\mathtt{P}, (\mathtt{D}, \sqcap), \gamma)$. A pp-concept is a pair $(\mathtt{B}, \mathtt{d})$ such that $\mathtt{B}^{\square} = \mathtt{d}$ and $\mathtt{d}^{\square} = \mathtt{B}$, where:

$$\mathtt{B}^{\square} = \prod_{\mathtt{p} \in \mathtt{B}} \gamma(\mathtt{p}) \qquad \mathtt{B} \subseteq \mathtt{P} \tag{7}$$

$$\mathtt{d}^{\square} = \{\mathtt{p} \in \mathtt{P} | \mathtt{d} \sqsubseteq \gamma(\mathtt{p})\} \qquad \mathtt{d} \in \mathtt{D} \tag{8}$$

Here, the extent of a pp-concept is a set of attribute pairs. We can obtain a CEC bicluster in a pp-concept if there is a clique among the attributes in the pairs. For example, consider the pp-concept $\mathtt{ppc}_1$ with extent $\{\mathtt{p}_{1,2}, \mathtt{p}_{1,3}, \mathtt{p}_{2,3}\}$ and intent $\{\{\mathtt{g}_1, \mathtt{g}_3\}, \{\mathtt{g}_5\}, \{\mathtt{g}_2, \mathtt{g}_4\}\}$. Its extent forms a clique among $\mathtt{m}_1$, $\mathtt{m}_2$, and $\mathtt{m}_3$, since all pairings of any two of those attributes are included.

If a pp-concept $(\mathtt{B}, \mathtt{d})$ contains a set of attributes $\mathtt{A}$ that forms a clique, then each pair $(p, \mathtt{A})$, for any partition component $p \in \mathtt{d}$, corresponds to a CEC bicluster. For example, from $\mathtt{ppc}_1$, we can obtain bicluster $(\{\mathtt{g}_1, \mathtt{g}_3\}, \{\mathtt{m}_1, \mathtt{m}_2, \mathtt{m}_3\})$.

4.3 Comparison with Sequential Pattern Mining

A sequence is an ordered list $\langle s_1 s_2 \dots s_m \rangle$, where s_i is an itemset $\{i_1, \dots, i_n\}$. 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 $m \leq n$. For example, 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.

Notice that the problem of retrieving CEC biclusters can be thought of as a particular type of sequential pattern mining where each itemset is composed by only one item, i.e. the sequences are an ordered list of items. Mining sequential patterns means retrieving frequent subsequences (i.e., subsequences that are present in more than n sequences) and for which there exist many efficient algorithms [16,15].

The CEC biclustering differs from sequential pattern mining when we allow overlaps in the partitions. Consider Table 4 as a sequential dataset. Each number in row x column y corresponds to the itemset when item y appears in the sequence x. For example, the sequence of object $\mathtt{g}_3$ is $\langle \{\mathtt{m}_4, \mathtt{m}_5\}\{\mathtt{m}_1\}\{\mathtt{m}_2\}\{\mathtt{m}_3\} \rangle$.

Let us consider items $\mathtt{m}_1$, $\mathtt{m}_4$, and $\mathtt{m}_5$. According to sequential pattern mining, $\mathtt{g}_3$ is different from $\mathtt{g}_4$, because $\mathtt{m}_4$ and $\mathtt{m}_5$ appear in the same itemset in $\mathtt{g}_3$. On the other hand, according to CEC biclustering with overlaps, $\mathtt{g}_3$ is similar to $\mathtt{g}_4$, because in both objects $\mathtt{m}_5 \leq \mathtt{m}_4 \leq \mathtt{m}_1$.

5 Recommendation

In the context of CrossCult, we are working on a visitor dataset that comprises several trajectories in a museum. Within this project, our main objective is to build a dynamic recommendation system for new visitors. This system should be able to suggest a museum item to visitors based on their trajectories and by looking at the trajectories of previous visitors. Also, it should be able to update the suggestion as they move inside the museum.

5.1 Matrix as order of interest

For each item in the museum, we can measure (e.g. by rating, duration of visit, etc.) their level of interestingness from a set of visitors. An example is shown in Table 6, where the number in cell xy is the ranking of item y according to visitor x. Here we have 3 visitors ($\mathtt{v}_1$, $\mathtt{v}_2$, and $\mathtt{v}_3$) in the database and 1 target visitor ($\mathtt{v}_a$). Among the existing visitors, only $\mathtt{v}_1$ has complete values over all five items. According to this visitor $\mathtt{i}_1$ is the best, followed by $\mathtt{i}_2$, $\mathtt{i}_3$, $\mathtt{i}_4$, and the worst $\mathtt{i}_5$. For $\mathtt{v}_2$ ($\mathtt{v}_3$), the order of interest of $\mathtt{i}_2$ ($\mathtt{i}_4$ resp.) is not known.

The target visitor ($\mathtt{v}_a$) has visited only three items, with the same order of preference in $\mathtt{i}_1$ and $\mathtt{i}_2$. This visitor will be included in the bicluster ($\{\mathtt{v}_1, \mathtt{v}_2, \mathtt{v}_a\}$, $\{\mathtt{i}_1, \mathtt{i}_2, \mathtt{i}_3\}$). The other two members of this bicluster do not agree on the order of interest of $\mathtt{i}_4$ and $\mathtt{i}_5$. Hence, we should suggest $\mathtt{i}_5$ to him/her, since one visitor ($\mathtt{v}_2$) similar to him/her ranked it first.

Table 6. Order of interest of 5 items, observed from certain visitors

	$\mathtt{i}_1$	$\mathtt{i}_2$	$\mathtt{i}_3$	$\mathtt{i}_4$	$\mathtt{i}_5$
$\mathtt{v}_1$	1	2	3	4	5
$\mathtt{v}_2$	3	?	4	2	1
$\mathtt{v}_3$	2	4	3	?	1
$\mathtt{v}_a$	1	1	2	?	?

5.2 Matrix as order of visit

Consider the dataset given by Table 7 about 4 visitors in a museum with 7 items. As explained in Subsection 4.3, this table can be regarded as a sequential dataset. The numbers in row x indicate the path of visitor $\mathtt{v}_x$. For example, the path of $\mathtt{v}_2$ is $\mathtt{i}_6 \rightarrow \mathtt{i}_1 \rightarrow \mathtt{i}_4 \rightarrow \mathtt{i}_2 \rightarrow \mathtt{i}_3 \rightarrow \mathtt{i}_7 \rightarrow \mathtt{i}_5$.

Now we have a new visitor $\mathtt{v}_a$ who recently arrives to the museum. He/She visits $\mathtt{i}_3$, followed by $\mathtt{i}_4$. Our task is to recommend a new item to her, by studying

Table 7. Order of visit of 7 items, observed from certain visitors

	i_1	i_2	i_3	i_4	i_5	i_6	i_7
v_1	1	2	3	4	5	6	7
v_2	2	4	5	3	7	1	6
v_3	4	2	1	5	6	3	7
v_4	7	3	1	4	2	6	5
v_a	?	?	1	2	?	?	?

the CEC biclusters over the first four visitors. Some of those biclusters are listed in Table 8. From B7, we can see that all four visitors visit i_7 after i_2. In B3, v_1 and v_3 follow the same order w.r.t. $\{i_3, i_4, i_5, i_7\}$: $i_3 \to i_4 \to i_5 \to i_7$. These two visitors also agree in the order of items $\{i_1, i_4, i_5, i_7\}$, as seen in B4.

Table 8. Some CEC biclusters in Table 7

#	Visitors	Items (in order)
B1	v_1	$i_1, i_2, i_3, i_4, i_5, i_6, i_7$
B2	v_2	$i_6, i_1, i_4, i_2, i_3, i_7, i_5$
B3	v_1, v_3	i_3, i_4, i_5, i_7
B4	v_1, v_3	i_1, i_4, i_5, i_7
B5	v_1, v_4	i_3, i_5, i_6
B6	v_2, v_3	i_6, i_1, i_4, i_5
B7	v_1, v_2, v_3, v_4	i_2, i_7

Those CEC biclusters can be studied to give a recommendation to v_a by focusing on those visitors that are similar to him/her. Thus, we can propose a recommendation strategy that follows sequential patterns in the dataset. The idea behind is the following: if many visitors have the path $i_a \to i_b \to i_c$, then we should recommend item i_c to a visitor who has done $i_a \to i_b$.

Since v_a has path $i_3 \to i_4$, we focus on the CEC biclusters that have those two items, i.e. B1, B2, and B3. One of those biclusters (B2) has a different ordering (i_3 after i_4), and thus we filter it out. Then, in B3 for example, the path is $i_3 \to i_4 \to i_5 \to i_7$. Therefore, we can recommend i_5 to v_a.

6 Conclusion

In this work, we have explored an approach to build collaborative recommendation strategy for visitors in a museum. This strategy takes into account the order of interest or the order of visit for each visitor, and we showed how to use CEC biclustering to obtain a set of similar visitors. We also presented a technique for mining CEC biclusters based on FCA using partition pattern structures. This recommendation strategy can be applied to any dataset where the order of items is relevant.

As future work, we intend to explore recommendations based on the order of visit. The problem to be solved is how to model visitors who visit a single item multiple times (for example, $\mathtt{i}_1$, $\mathtt{i}_2$, and back to $\mathtt{i}_1$).

Another noteworthy question is how to measure the "score" of each bicluster in order to rank recommendations for a new visitor. Ranking candidate items was studied in [3] for constant-value biclusters, and it is possible to extend this work to `CEC` biclusters. Moreover, further comparisons of `CEC` biclustering and sequential pattern mining should be investigated, in particular, regarding their complexities and their results. Finally, an implementation of the `CEC` biclustering using partition pattern structures and an empirical study on real-world data should be performed to measure its complexity and efficiency.

References

1. Adomavicius, G., Tuzhilin, A.: Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. IEEE transactions on knowledge and data engineering **17**(6), 734–749 (2005)
2. Akhmatnurov, M., Ignatov, D.I.: Context-aware recommender system based on boolean matrix factorisation. In: CLA. pp. 99–110 (2015)
3. Alqadah, F., Reddy, C.K., Hu, J., Alqadah, H.F.: Biclustering neighborhood-based collaborative filtering method for top-n recommender systems. Knowledge and Information Systems **44**(2), 475–491 (2015)
4. Ben-Dor, A., Chor, B., Karp, R., Yakhini, Z.: Discovering local structure in gene expression data: the order-preserving submatrix problem. Journal of computational biology **10**(3-4), 373–384 (2003)
5. du Boucher-Ryan, P., Bridge, D.: Collaborative recommending using formal concept analysis. Knowledge-Based Systems **19**(5), 309–315 (2006)
6. Codocedo, V., Napoli, A.: Lattice-based biclustering using partition pattern structures. In: Proceedings of the Twenty-first European Conference on Artificial Intelligence. pp. 213–218. IOS Press (2014)
7. Codocedo-Henríquez, V.: Contributions à l'indexation et à la récupération d'information utilisant l'analyse formelle de concepts. Ph.D. thesis, Université de Lorraine (2015), http://www.theses.fr/2015LORR0143
8. Ganter, B., Kuznetsov, S.O.: Pattern structures and their projections. In: International Conference on Conceptual Structures. pp. 129–142. Springer (2001)
9. Ignatov, D.I., Kaminskaya, A.Y., Konstantinova, N., Malyukov, A., Poelmans, J.: FCA-based recommender models and data analysis for crowdsourcing platform Witology. In: International Conference on Conceptual Structures. pp. 287–292. Springer (2014)
10. Ignatov, D.I., Kuznetsov, S.O., Poelmans, J.: Concept-Based Biclustering for Internet Advertisement. In: 2012 IEEE 12th International Conference on Data Mining Workshops. pp. 123–130. Brussels, Belgium (Jun 2009), http://arxiv.org/abs/0906.4982, arXiv: 0906.4982
11. Ignatov, D.I., Poelmans, J., Zaharchuk, V.: Recommender system based on algorithm of bicluster analysis RecBi. arXiv preprint arXiv:1202.2892 (2012)
12. Ignatov, D.I., Watson, B.W.: Towards a unified taxonomy of biclustering methods. arXiv preprint arXiv:1702.05376 (2017)

13. Kaytoue, M., Kuznetsov, S.O., Macko, J., Napoli, A.: Biclustering meets triadic concept analysis. Annals of Mathematics and Artificial Intelligence **70**(1-2), 55–79 (2014)
14. Madeira, S.C., Oliveira, A.L.: Biclustering algorithms for biological data analysis: a survey. IEEE/ACM Transactions on Computational Biology and Bioinformatics (TCBB) **1**(1), 24–45 (2004)
15. Wang, J., Han, J.: BIDE: Efficient mining of frequent closed sequences. In: Data Engineering, 2004. Proceedings. 20th International Conference on. pp. 79–90. IEEE (2004)
16. Yan, X., Han, J., Afshar, R.: CloSpan: Mining: Closed sequential patterns in large datasets. In: Proceedings of the 2003 SIAM international conference on data mining. pp. 166–177. SIAM (2003)

Combining Concept Annotation and Pattern Structures for Guiding Ontology Mapping

Pierre Monnin[1], Amedeo Napoli[1], and Adrien Coulet[1,2]

[1] LORIA (CNRS, Inria Nancy-Grand Est, Université de Lorraine), Vandœuvre-Lès-Nancy, France
pierre.monnin@loria.fr
[2] BMIR, Stanford University, USA

Abstract. Formal Concept Analysis (FCA) is a mathematical framework classifying in formal concepts a set of objects w.r.t. their common attributes. To this aim, FCA relies on a binary incidence relationship indicating whether an object has an attribute. On one hand, in order to consider more complex descriptions for objects (*e.g.*, intervals, graphs), FCA has been extended with Pattern Structures. On the other hand, in a previous work, we introduced the notion of Concept Annotation, adding a third dimension to formal concepts, computed over the extent, without modifying the original classification. In this paper, we combine Concept Annotation with the formalism of Pattern Structures and we consider multiple annotation possibilities, *i.e.*, multiple annotations for one concept and computing the annotation over the intent. We illustrate our approach and its interest with two use cases: *(i)* suggesting mappings between ontology classes and *(ii)* finding specific classes frequently associated as domain and range of a predicate.

Keywords: Concept Annotation · Pattern Structures · Formal Concept Analysis

1 Introduction

An ontology is a formal representation of a particular domain [8], consisting of two parts: an *assertion component* (*ABox*) and a *terminological component* (*TBox*). In the *TBox*, classes and predicates between classes are defined while in the *ABox*, individuals instantiate classes and predicates. For example, the drug *codein* can be considered as an individual instantiating the class *Analgesics*.

Nevertheless, there is a need for structure between ontologies. Indeed, individuals of an *ABox* may instantiate classes from several ontologies. For example, in the medical domain, individuals representing diseases may instantiate classes of several ontologies of diseases, such as MeSH (*Medical Subject Headings*), ICD-9-CM and ICD-10-CM (*International Classification of Diseases version 9 and 10, Clinical Modification*). Corresponding classes from distinct ontologies may be mapped thanks to equivalence relationships resulting from an alignment process [5]. This alignment may be either a manual review by an expert or a semi-automatic process.

On the other hand, within an ontology, additional knowledge and structuring may be discovered. Thus, the *ABox* of an ontology may also contains predicate assertions, *i.e.*, relations between individuals. For example, in DrugBank, a database of drug information, a drug can be associated with a gene, indicating an interaction between them. Thereby, the gene *VKORC1* is indicated to be an inhibitor of the drug *warfarin*. Genes and drugs instantiate classes of ontologies, such as ATC (*Anatomical Therapeutic Chemical Classification System*) for drugs and GO (*Gene Ontology*) for genes. Therefore, from the predicate assertions, we could discover classes of drugs and genes frequently associated as domain and range of a predicate. Such domain and range associations could be interesting as they may indicate common properties of the gene class or the drug class.

In this article, we aim at addressing both of the aforementioned use cases, *(i)* suggesting equivalence relationships between ontology classes and *(ii)* discovering classes frequently associated in domain and range of a predicate, with Formal Concept Analysis (FCA) [7]. FCA is a well-founded mathematical framework adapted to knowledge engineering purposes [1, 3, 4] that groups objects w.r.t. their common attributes in formal concepts. The association of an attribute to an object is expressed thanks to a binary relationship. Formal concepts are organized by a partial order in a hierarchical structure called a lattice. In a previous work [9], we introduced an extension of FCA, called *Concept Annotation*, allowing to compare the hierarchical structure of the lattice with the class hierarchy of an ontology for possible refactoring. The main interest of Concept Annotation resides in adding a third dimension to a previously generated lattice without changing its structure. The main contribution of the present article is to extend the initial definition of Concept Annotation by *(i)* associating multiple Concept Annotations with formal concepts, and *(ii)* using a formalism similar to Pattern Structures [6] to express the annotations. Pattern Structures are another extension of FCA dealing with objects having complex (non binary) descriptions (*e.g.*, graphs, numerical values, classes of an ontology). In the following, we assume that the reader is familiar with the basics of FCA and Pattern Structures.

This article is organized as follows. In Section 2, necessary basics about ontologies are recalled. In Section 3, we present the initial definition of Concept Annotation as well as our proposed extension. Then, in Sections 4 and 5, we illustrate the combination of Concept Annotation and Pattern Structures by, respectively, suggesting equivalence relationships between classes and discovering classes frequently associated as domain and range of a predicate. Finally, in Sections 6 and 7, we discuss the results we obtain on the two considered use cases as well as the next challenges to tackle.

2 Basics about Ontologies

An ontology is a formal representation of a particular domain [8]. It is composed of two parts, the *TBox* and the *ABox*. The *TBox* defines classes and relationships between them. We denote $\mathcal{C}(\mathcal{O})$ the set of classes of the ontology $\mathcal{O}$. Classes of an ontology are instantiated by individuals of the *ABox*. These individuals can also be involved in instantiation of relationships (such as the drug-gene relationship in DrugBank previously explained).

Classes of an ontology are organized in a partial order by a subsumption relation denoted by $\sqsubseteq$. Considering two classes C and D, $C \sqsubseteq D$ states that every instance of C is also an instance of D. The *least common subsumer* (sometimes named the *lowest common ancestor*) of two classes C_1 and C_2 of an ontology is the most specific class subsuming both C_1 and C_2. It is denoted by $\text{lcs}(C_1, C_2)$. In $\mathcal{EL}$ ontologies where no cycle appears, the lcs of two classes always exists [2]. Considering a set of classes $\mathcal{C}_n = \{C_1, C_2, \ldots, C_n\}$, we define the set of the most specific classes of $\mathcal{C}_n$ as $\min \mathcal{C}_n = \{C \in \mathcal{C}_n \mid \nexists\, D \in \mathcal{C}_n,\ D \sqsubseteq C\}$.

3 Concept Annotation

Concept Annotation is an extension of FCA that we introduced in a previous work [9]. In this section, we first recall basics about our initial work and then present the proposed extension.

3.1 Basics about Concept Annotation

In Concept Annotation, standard FCA is applied on a formal context to build a concept lattice. Then, for each formal concept, an annotation is computed, adding a third dimension to each concept.

For example, in [9], a lattice is built from a formal context (G, M, I) where G is a set of individuals from the *ABox* of an ontology $\mathcal{O}$ and M is a set of predicates of this ontology. $(g, m) \in I$ indicates that the individual g is involved in a relationship whose predicate is m. Then, classes of the *TBox* instantiated by the individuals in G are added as annotation. The resulting hierarchy formed by the lattice and the annotations is compared to the class hierarchy of the ontology. To compute the annotation, we defined a derivation operator $(\cdot)^\diamond : 2^G \to 2^{\mathcal{C}(\mathcal{O})}$, where $2^{\mathcal{C}(\mathcal{O})}$ corresponds to the powerset of the set of classes of the ontology. Considering a formal concept (A, B) where A is a set of individuals, the annotation is defined thanks to the derivation operator as follows: $A^\diamond = \bigcap_{g \in A} \{C \mid \mathcal{O} \models C(g)\}$[3]. Intuitively, it corresponds to the set of all classes of $\mathcal{O}$ instantiated by all individuals in A.

3.2 Combining Concept Annotation and Pattern Structures

In this paper, we propose to combine Concept Annotation with Pattern Structures. Indeed, such combination will allow us to only keep as annotation the most specific classes instantiated by all individuals.

To do so, we define a function $\delta : 2^G \to 2^{\mathcal{C}(\mathcal{O})}$ that associates an individual from G with the set of the most specific classes of the ontology $\mathcal{O}$ that this individual instantiates. Formally, given $g \in G$, $\delta(g) = \min \{C \in \mathcal{C}(\mathcal{O}) \mid \mathcal{O} \models C(g)\}$.

[3] $C(g)$ indicates that g is an instance of C in the Description Logics formalism

$\delta(g)$ is considered as the description of the individual g. Given two individuals, $g_1, g_2 \in G$, we define a similarity operator $\sqcap$ to compare their two descriptions as follows:

$$\delta(g_1) \sqcap \delta(g_2) = \min \{\text{lcs}(C_1, C_2) \mid \forall\ (C_1, C_2) \in \delta(g_1) \times \delta(g_2)\}$$

The use of the *least common subsumer* allows to compute the most specific classes of the ontology that both g_1 and g_2 instantiates. Finally, given a formal concept (A, B), we annotate it thanks to the new derivation operator $(\cdot)^\circ : 2^G \to 2^{\mathcal{C}(\mathcal{O})}$ defined as $A^\circ = \prod_{g \in A} \delta(g)$.

In the following sections, the defined functions and operators will be restricted to specific ontologies. Thus, δ_i, $\sqcap_i$ and $(\cdot)^{\circ_i}$ will only be applicable to classes of an ontology $\mathcal{O}_i$.

4 Suggesting Mappings between Classes of Ontologies

In this section, we illustrate the combination of Concept Annotation and Pattern Structures with the use case of suggesting equivalence relationships between classes of two ontologies, denoted by $\mathcal{O}_1$ and $\mathcal{O}_2$. To generate these mappings, we consider individuals that instantiate classes of both $\mathcal{O}_1$ and $\mathcal{O}_2$. They also instantiate classes of another ontology, denoted by $\mathcal{O}_{ref}$, that is considered in this setting as the reference ontology, *i.e.* the feature we consider to build the original classification of the set of instances. In order to avoid complexity problems, we keep $\mathcal{O}_{ref}$ of a small size.

4.1 Classifying Individuals w.r.t. $\mathcal{O}_{ref}$ in a Concept Lattice

The first step is to build a concept lattice classifying the individuals w.r.t. the classes of $\mathcal{O}_{ref}$ they instantiate. To this aim, we use the pattern structure $(G, (2^{\mathcal{C}(\mathcal{O}_{ref})}, \sqcap_{ref}), \delta_{ref})$. G is the set of individuals and $2^{\mathcal{C}(\mathcal{O}_{ref})}$ is the powerset of the set of classes of $\mathcal{O}_{ref}$. δ_{ref} and $\sqcap_{ref}$ are defined as in Subsection 3.2 w.r.t. the classes of $\mathcal{O}_{ref}$. From this pattern structure definition, we obtain pattern concepts (A, D) organized in a concept lattice, where $A \subseteq G$ is a set of individuals and $D \in 2^{\mathcal{C}(\mathcal{O}_{ref})}$ is a set of the most specific classes of $\mathcal{O}_{ref}$ that all individuals in A instantiate.

4.2 Annotating the Concept Lattice with $\mathcal{O}_1$ and $\mathcal{O}_2$

In this next step, classes from $\mathcal{O}_1$ and $\mathcal{O}_2$ are considered for annotating the concept lattice. To this aim, we define *two* annotations (one per ontology) using the formalism defined in Subsection 3.2. Thereby, we use two functions δ_1 and δ_2 to associate an individual g with the set of the most specific classes from $\mathcal{O}_1$ and $\mathcal{O}_2$ that this individual instantiates. $\sqcap_1$ and $\sqcap_2$ are used to compute the similarity between descriptions of two individuals w.r.t. the two considered ontologies $\mathcal{O}_1$ and $\mathcal{O}_2$. Finally, to compute the two annotations for each pattern concept (A, D), two derivation operators $(\cdot)^{\circ_1} : 2^G \rightarrow 2^{\mathcal{C}(\mathcal{O}_1)}$ and $(\cdot)^{\circ_2} : 2^G \rightarrow 2^{\mathcal{C}(\mathcal{O}_2)}$ are defined as follows:

$$A^{\circ_1} = \sqcap_{1\, g \in A} \delta_1(g) \quad A^{\circ_2} = \sqcap_{2\, g \in A} \delta_2(g)$$

As a result, for each pattern concept (A, D), an annotated pattern concept $(A, D, A^{\circ_1}, A^{\circ_2})$ is obtained where A°_1} (resp. A°_2}) is the set of the most specific classes of $\mathcal{O}_1$ (resp. $\mathcal{O}_2$) that all individuals in A instantiate.

4.3 Reading Mappings from the Lattice

Let us consider an annotated pattern concept $(A, D, A^{\circ_1}, A^{\circ_2})$. From the previous definitions, we know that $A^{\circ_1} \subseteq C(\mathcal{O}_1)$ contains the set of the most specific classes of $\mathcal{O}_1$ that all individuals in A instantiate. Similarly, $A^{\circ_2} \subseteq C(\mathcal{O}_2)$ contains the set of the most specific classes of $\mathcal{O}_2$ that all individuals in A instantiate. Therefore, considering each pair of classes $(C_1, C_2) \in A^{\circ_1} \times A^{\circ_2}$, we know that they are instantiated by the same set of individuals, *i.e.*, individuals in A. Therefore, an equivalence relationship is suggested between C_1 and C_2, based on the instances of the two classes.

5 Discovering Associated Classes as Domain and Range of a Predicate

In this section, we illustrate how Concept Annotation and Pattern Structures could be used to discover the most specific classes frequently associated as domain and range of a predicate from its instantiations in an *ABox*. Such domain and range characterization could indeed indicate common behavior at the class level. For example, considering DrugBank, a gene can be involved with a drug in a relationship whose action is specified (*e.g.*, inhibitor, antagonist). Such relationships correspond to assertions of a predicate. As drugs instantiate classes of ATC and genes instantiate classes of GO, we could discover that instances of a GO class, considered as a family of genes, are frequently indicated as inhibitors of instances of an ATC class, considered as a family of drugs.

In the following paragraphs, we consider individuals, instantiating classes of an ontology $\mathcal{O}_1$, that are involved in relationships with other individuals, instantiating classes of an ontology $\mathcal{O}_2$[4].

5.1 Classifying Relationships in a Concept Lattice

First, a classification of relationships is established. To this aim, we consider the formal context (G, M, I) where G is the set of individuals instantiating classes from $\mathcal{O}_1$ and M is the set of individuals instantiating classes from $\mathcal{O}_2$. Given $g \in G$, and $m \in M$, $(g, m) \in I$ if and only if a relationship between g and

[4] It is not necessary for $\mathcal{O}_1$ and $\mathcal{O}_2$ to be distinct.

m exists. Standard binary FCA is applied on this formal context to generate the associated formal concepts organized in a concept lattice. In the DrugBank example, G is the set of genes and M is the set of drugs. Considering $g \in G$ and $m \in M$, we have $(g, m) \in I$ if and only if a relationship between the gene g and the drug m exists in DrugBank.

5.2 Annotating the Concept Lattice with $\mathcal{O}_1$ and $\mathcal{O}_2$

Consider a formal concept (A, B) from the concept lattice. $A \subseteq G$ is a set of individuals instantiating classes of $\mathcal{O}_1$ and $B \subseteq M$ is a set of individuals instantiating classes of $\mathcal{O}_2$. From the definition of the derivation operators in FCA, we know that every individual in A is in relationship with every individual in B. To find the most specific ontology classes involved as domain and range of the predicate, *two* annotations for formal concepts are defined (one per ontology) using the formalism defined in Subsection 3.2. We use two functions δ_1 and δ_2 to associate an individual with the set of the most specific classes from $\mathcal{O}_1$ and $\mathcal{O}_2$ that this individual instantiates. δ_1 is applied on individuals $g \in A$ and δ_2 is applied on individuals $m \in B$. $\sqcap_1$ and $\sqcap_2$ are used to compute the similarity between descriptions of two individuals w.r.t. the two considered ontologies $\mathcal{O}_1$ and $\mathcal{O}_2$. Finally, to compute the two annotations for each formal concept (A, B), two derivation operators $(\cdot)^{\circ_1} : 2^G \to 2^{\mathcal{C}(\mathcal{O}_1)}$ and $(\cdot)^{\circ_2} : 2^M \to 2^{\mathcal{C}(\mathcal{O}_2)}$ are defined as follows:

$$A^{\circ_1} = \bigsqcap_{g \in A}{}_1\, \delta_1(g) \quad B^{\circ_2} = \bigsqcap_{m \in B}{}_2\, \delta_2(m)$$

It is noteworthy that $(\cdot)^{\circ_2}$ is applied on the intent of the formal concept. As a result, each formal concept (A, B) is replaced by the annotated concept $(A, B, A^{\circ_1}, B^{\circ_2})$ where A°_1} (resp. B°_2}) is the set of the most specific classes of $\mathcal{O}_1$ (resp. $\mathcal{O}_2$) that all individuals in A (resp. in B) instantiate.

5.3 Reading the Domain and Range of a Relation from the Annotated Lattice

Let us consider an annotated concept $(A, B, A^{\circ_1}, B^{\circ_2})$. Every individual in A is in a relationship with every individual in B. Furthermore, A°_1} is the set of the most specific classes of $\mathcal{O}_1$ that all individuals in A instantiate. Similarly, B°_2} is the set of the most specific classes of $\mathcal{O}_2$ that all individuals in B instantiate. Therefore, from this annotated concept, for the considered predicate, classes from A°_1} as domain are associated with classes from B°_2} as range.

Along the hierarchy of the lattice, the two annotations A°_1} and B°_2} behave in the opposite way. Indeed, A°_1}, computed on the extent, contains more specific classes when the number of individuals in the extent decreases, *i.e.*, when browsing the lattice top to bottom. On the contrary, B°_2}, computed on the intent, contains more specific classes when the number of individuals in the intent decreases, *i.e.*, when browsing the lattice bottom to top. Consequently, in annotated concepts at the top of the lattice, general classes of $\mathcal{O}_1$ will be involved in domains and specific classes of $\mathcal{O}_2$ will be involved in ranges. In annotated concepts at the bottom of the lattice, specific classes of $\mathcal{O}_1$ will act as domain and general classes of $\mathcal{O}_2$ will act as ranges. Therefore, this structure could be of interest in an interactive setting with an expert. Indeed by browsing the lattice depending on her specific constraints on the domains or ranges to discover, the expert could leverage on the lattice order to obtain more general or specific classes involved.

6 Discussion

Regarding the suggestion of equivalence relationships between classes of two ontologies, the approach needs to be validated on a real dataset where mappings already exist. One main identified drawback is that the current annotation process works under the *Closed World Assumption*. Thus, mappings are suggested considering that all instantiations are correct and none is missing. As many datasets are under the *Open World Assumption*, the suggested mappings may only be applicable on the considered set of individuals but may not be applied to other sets of individuals. Therefore, there is a need to validate the suggested mappings with an expert. One next challenge would be to define a new derivation operator to compute an annotation working under the *Open World Assumption*.

The original lattice is built from the set of individuals and the classes of a reference ontology $\mathcal{O}_{ref}$. However, other features could be considered for this original classification, such as the relationships involving the individuals, similarly to our previous work [9]. The choice of the features to consider are of importance as they impact the original lattice, which is the the "pivot" structure from which equivalence relationships are suggested. One user can then choose the specific features to consider to guide the generated mappings. In the proposed approach, equivalence relationships are suggested by considering annotated concepts separately. Nevertheless, the subsumption relations between concepts could be considered to suggest mappings of the form of subsumption relations between classes of the two considered ontologies. Such setting could therefore be used to align and structure folksonomies of various users in a social network.

Finally, the suggestion of mappings may not be the only use case of interest for the proposed setting. Indeed, by annotating the original lattice with classes of different versions of the same ontology, concept drift could be highlighted. For example, a semantic change in the classes between two versions of an ontology would be indicated by discovering in the annotations that the same set of individuals does not instantiate the same classes between the two versions.

Regarding the use case of discovering classes frequently associated as domain and range of a predicate, we could also benefit from an experiment on a real dataset. In this setting, as previously mentioned, there is an issue in selecting the interesting domains and ranges. Additionally to the interactive discovery process previously mentioned, various metrics could be considered to highlight interesting annotated concepts. For example, notions of *support* or *confidence* based on the cardinal number of the extent and / or intent could be of interest here. Similarly to the suggestion of equivalence relationships, this work does not benefit from the subsumption relations between formal concepts. Such relations could be used to build a hierarchical organization of associations of domains and ranges. However, as the two annotations behave in opposite ways, this hierarchy should not be read as an order.

7 Conclusion

In this paper, we combine Concept Annotation with the formalism of Pattern Structures. To illustrate our approach and its interest, we consider two use cases: suggesting equivalence relationships between classes of two ontologies and discovering classes frequently associated as domain and range of a predicate. The formalism of Pattern Structures is an advantage compared to our original work on Concept Annotation as it enables more complex annotations, using descriptions of objects and similarity operations. Moreover, we illustrate some other annotation possibilities: multiple annotations for one concept and annotations computed on the intent instead of the extent. As a result, starting from a concept lattice considered as a "pivot" structure, it is possible to obtain a complex structure representing or highlighting several relations between its components. The next challenges lies in experimenting our approach on real datasets and formalizing the properties of the annotation.

Acknowledgments

This work is supported by the *PractiKPharma* project, founded by the French National Research Agency (ANR) under Grant No.: ANR-15-CE23-0028, and by the *Snowball* Inria Associate Team.

References

1. Baader, F., Ganter, B., Sertkaya, B., Sattler, U.: Completing Description Logic Knowledge Bases Using Formal Concept Analysis. In: IJCAI 2007, Proceedings of the 20th International Joint Conference on Artificial Intelligence, Hyderabad, India, January 6-12, 2007. pp. 230–235 (2007), http://ijcai.org/Proceedings/07/Papers/035.pdf
2. Baader, F., Küsters, R., Molitor, R.: Computing least common subsumers in description logics with existential restrictions. In: Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence, IJCAI 99, Stockholm, Sweden, July 31 - August 6, 1999. 2 Volumes, 1450 pages. pp. 96–103 (1999), http://ijcai.org/Proceedings/99-1/Papers/015.pdf
3. Bendaoud, R., Napoli, A., Toussaint, Y.: Formal Concept Analysis: A Unified Framework for Building and Refining Ontologies. In: Knowledge Engineering: Practice and Patterns, 16th International Conference, EKAW 2008, Acitrezza, Italy, September 29 - October 2, 2008. Proceedings. pp. 156–171 (2008). https://doi.org/10.1007/978-3-540-87696-0_16

4. Coulet, A., Domenach, F., Kaytoue, M., Napoli, A.: Using Pattern Structures for Analyzing Ontology-Based Annotations of Biomedical Data. In: Formal Concept Analysis, 11th International Conference, ICFCA 2013, Dresden, Germany, May 21-24, 2013. Proceedings. pp. 76–91 (2013). https://doi.org/10.1007/978-3-642-38317-5_5
5. Euzenat, J., Shvaiko, P.: Ontology Matching, Second Edition. Springer (2013)
6. Ganter, B., Kuznetsov, S.O.: Pattern Structures and Their Projections. In: Conceptual Structures: Broadening the Base, 9th International Conference on Conceptual Structures, ICCS 2001, Stanford, CA, USA, July 30-August 3, 2001, Proceedings. pp. 129–142 (2001). https://doi.org/10.1007/3-540-44583-8_10
7. Ganter, B., Wille, R.: Formal concept analysis - mathematical foundations. Springer (1999)
8. Gruber, T.R.: A translation approach to portable ontology specifications. Knowledge acquisition **5**(2), 199–220 (1993)
9. Monnin, P., Lezoche, M., Napoli, A., Coulet, A.: Using Formal Concept Analysis for Checking the Structure of an Ontology in LOD: The Example of DBpedia. In: Foundations of Intelligent Systems - 23rd International Symposium, ISMIS 2017, Warsaw, Poland, June 26-29, 2017, Proceedings. pp. 674–683 (2017). https://doi.org/10.1007/978-3-319-60438-1_66

Understanding Collaborative Filtering with Galois Connections

Dmitry I. Ignatov[0000−0002−6584−8534] and Vladimir Kursitys

National Research University Higher School of Economics, Moscow, Russia
dignatov@hse.ru, kursitys-vladimir@rambler.ru

Abstract. In this paper, we explain how Galois connection and related operators between sets of users and items naturally arise in user-item data for forming neighbourhoods of a target user or item for Collaborative Filtering. We compare the properties of these operators and their applicability in simple collaborative user-to-user and item-to-item setting. Moreover, we propose a new neighbourhood-forming operator based on pair-wise similarity ranking of users, which takes intermediate place between the studied closure operators and its relaxations in terms of neighbourhood size and demonstrates comparatively good Precision-Recall trade-off. In addition, we compare the studied neighbourhood-forming operators in the collaborative filtering setting against simple but strong benchmark, the SlopeOne algorithm, over bimodal cross-validation on MovieLens dataset.

Keywords: Collaborative Filtering · Galois Connection · Recommender Systems · Neighbourhood-forming operators

1 Introduction

Galois connections of different types as well as closure and kernel operators play important role not only in mathematics [14] but also in analysis of relational data, for example, object-attribute tables also known as transactional databases [13,40,39], information systems [16,38], formal contexts [18,33], user-item rating matrices [9,28], etc.

Thus, it has been shown that Boolean matrix factorisation performed by means of Galois operators on user-item binary matrix (obtained from user-item rating matrix under proper scaling) is not worse than ordinary SVD to capture similarity between users and items in terms of MAE, Precision and Recall [28]. The so called concept lattice, an ordered hierarchy of maximal submatrices (users, items) generated by Galois operators, has been proposed as a global search space for nearest neighbours of users and items [9]; however, such a lattice might be huge even for sparse input rating matrices and its generating is costly in terms of time and storage memory. An interesting attempt to scale this approach and form only necessary relevant neighbourhoods of users and items via Galois operators has been done in [5].

However, a systematic study of those useful connections and operators as well as their variants suitable for Collaborative filtering has not been performed yet. In this study, we introduce and discuss Galois operators for collaborative filtering setting to form neighbourhoods of users and items (as well as their sets in group recommendation scenario) and sets of prospective relevant items to rank by means of ordinary user-based (or item-based) approaches.

The remainder of the paper consists of five sections. In Section 2, we recall several definitions of Galois or derivation operators from Formal Concept Analysis and the associated closure operators. In Section 3, we explain how the existing operators can be used to form neighbourhoods of users (items) for a target user (item) as well as sets of prospective relevant items to recommend. Section 6 presents several simple experiments with user-based and item-based approaches where the formed neighbourhoods and sets of items used as parameters. Section 7 discusses related work and Section 8 concludes the paper.

2 Galois connections and related operators

First, we give the definition of Galois connection.

Let $\varphi : P \to Q$ and $\psi : Q \to P$

be maps between two ordered sets $(P, \leq)$ and $(Q, \leq)$. The pair of such maps is called a Galois connection between the ordered sets if:

a. $p_1 \leq p_2 \Rightarrow \varphi p_1 \geq \varphi p_2$;
b. $q_1 \leq q_2 \Rightarrow \varphi q_1 \geq \varphi q_2$;
c. p $\leq \psi\varphi p$ and $q \leq \varphi\psi q$.

The operators φ and ψ are called Galois operators.

Let us define concrete version of Galois operators as it is done in Formal Concept Analysis (FCA) [18] over relational object-attribute tables but in collaborative filtering setting. Here, the role of objects is played by users and the role of attributes by items.

Let us consider a triple (U, I, R) called formal context in FCA, where U is a set of users, I is a set of items, and $R \subseteq U \times I$. A pair $(u, i) \in R$ iff user $u \in U$ rated (liked or browsed) item $i \in I$.

In this case, for a subset of users $X \subseteq U$ and a subset of items $Y \subseteq I$ Galois operators (prime or derivation operators), $(\cdot)' : 2^U \to 2^I$ and $(\cdot)' : 2^I \to 2^U$, are defined as follows:

$X' = \{i \mid uRi \text{ for all } u \in X\}$,

$Y' = \{u \mid uRi \text{ for all } i \in Y\}$.

In fact, X' is the set of items that every user from X rated and Y' are those users, who rated every item from Y.

One may check that two operators $(\cdot)'$ form a Galois connection between $(2^U, \subseteq)$ and $(2^I, \subseteq)$.

Moreover, one may prove that $(\cdot)''$ is a closure operator, i.e. for $X, Z \subseteq U$ (or for $X, Z \subseteq I$).

1. $X \subseteq Z \Rightarrow X'' \subseteq Z''$ (monotony);
2. $X \subseteq X''$ (extensity);
3. $X' = X'''$ (idempotency).

A monotone and idempotent operator $op(\cdot)$ on 2^U is called a kernel operator iff for $X \subseteq U$: $op(X) \subseteq X$ (intensity). Operators with intensity property play important role in Social Choice Theory since they help to select relevant alternatives from their input set [4]. We provide an example of kernel operator in section 3.

Let us discuss the meaning of several important properties of the introduced Galois operators in terms of Collaborative Filtering domain.

For $X, X_1, X_2 \subseteq U$ (similarly for $Y, Y_1, Y_2 \subseteq I$):

4. $X_1 \subseteq X_2 \Rightarrow X_2' \subseteq X_1'$ (antitony);
5. $X' = X'''$.

The fourth property means that the more users we add to the initial set X_1, the less is the number of their co-rated items (this property have been exploited in classic itemset mining algorithm, Apriori, in [2]). To understand the meaning of the remaining properties we need to discuss the interpretation of the result of $(\cdot)''$ to $X \subseteq U$. The first prime returns the set X' of all co-rated items for users from X, the second prime returns the set X'' of all users who rated all items from X'. In fact, this set X'' may become larger than X or remain the same (Property 2). If we have a group of users Z and its subgroup X, then after looking at the items that X and Z rated, i.e. X' and Z', we obtain by Property 4 that the set of items X' is larger or equal to Z'. By applying Property 4 one more time, we obtain that X'' is a subgroup of Z'' or equal to it. Property 2 says that by passing through items X' co-rated by X, we may obtain some more users who rated all items X' as well, i.e. our overlooked neighbours at the beginning. The third property says that it is not necessary to look at the co-rated items of the group X'' since everyone who rated all items from X' is in X''. That is X'' is a fixed point of operator $(\cdot)''$. These fixed points correspond to the called formal concepts in FCA, i.e. pairs (X'', X') for $X \subseteq U$ (for itemsets they are defined similarly).

In collaborative filtering setting, for a particular target user u from U we are mainly interested in $\{u\}'$, the items rated by u, and $\{u\}''$, all users from U, who rated all items $\{u\}'$. However, if we would require to show new items that those users also rated, applying the prime operator one more time, we would obtain $\{u\}''' = \{u\}'$, i.e. nothing to potentially recommend. One of the remedies would be to delete u from $\{u\}''$ and obtain $(\{u\}'' \setminus \{u\})' \setminus \{u\}'$, however we prefer to examine a richer set of possible alternatives and study their properties.

3 Connections for Collaborative Filtering

Let (U, I, R) be a formal context, then for a subset of users $X \subseteq U$ and a subset of items $Y \subseteq I$ then neighbourhood-forming operators, $(\cdot)^\diamond : 2^U \to 2^I$ and $(\cdot)^\diamond : 2^I \to 2^U$, are defined as follows:

$X^\diamond = \{i \mid uRi \text{ for some } u \in X\}$,
$Y^\diamond = \{u \mid uRi \text{ for some } i \in I\}$.

In fact, $X^\diamond$ can be considered as a query "show me all that have been bought by at least some user from X for $X \subseteq U$; $Y^\diamond$ is interpreted as all users that bought at least one item from Y for $Y \subseteq I$.

Property 1. $X^\diamond = \bigcup\limits_{x \in X} x'$ and $Y^\diamond = \bigcup\limits_{y \in Y} y'$.

Property 2. $X, Z \subseteq U \Rightarrow X^\diamond \subseteq Z^\diamond$ (monotony of $(\cdot)^\diamond$) (similarly for $X, Z \subseteq I$).

Now we have 2^2 combinations of operators $(\cdot)'$ and $(\cdot)^\diamond$ to form neighbours, and 2^3 operator combinations to list potentially relevant items. Let us figure it out theoretically which of the proposed combinations are relevant for collaborative filtering.

Theorem 1. *For $X, Z \subseteq U$ (similarly for $X, Z \subseteq I$) the following properties fulfil:*

- *If $X \subseteq Z$ then*
  1. *$X'^\diamond \supseteq Z'^\diamond$ (antitony);*
  2. *$X^{\diamond\prime} \supseteq Z^{\diamond\prime}$ (antitony);*
  3. *$X^{\diamond\diamond} \subseteq Z^{\diamond\diamond}$ (monotony);*
- 4. *$X \subseteq X'^\diamond$ (extensity);*
  5. *$X \supseteq X^{\diamond\prime}$ (intensity);*
  6. *$X \subseteq X^{\diamond\diamond}$ (extensity);*
- 7. *$X'^\diamond = X'^{\diamond\prime\diamond}$ (idempotency);*
  8. *$X^{\diamond\prime} = X^{\diamond\prime\diamond\prime}$ (idempotency);*
  9. *$(\cdot)^{\diamond\diamond}$ is not idempotent.*

Corollary 1. *Operator $(\cdot)^{\diamond\prime}$ is a kernel operator (antitone, extensive, and idempotent).*

In what follows, we mainly concentrate on one target user u, its neighbours founded by double combination of the derivation and neighbourhood forming operators, and items potentially relevant for that user obtained by triple combinations of those operators.

Lemma 1. *For $u \in U$: $u' = u^\diamond$ (similarly for $i \in I$).*

Theorem 2. *For $u \in U$ the following inclusions hold:*
$u'^{\diamond\prime} = u^{\diamond\diamond\prime} \subseteq u''' = u^{\diamond\prime\prime} = u' = u^{\diamond\prime\diamond} = u''^\diamond \subseteq u'^{\diamond\diamond} = u^{\diamond\diamond\diamond}$.

Thus, every triple operator on the left hand side from u' does not bring new items in comparison to those that u is rated. So, potentially we are interested in those from the right hand side, namely, $u''^\diamond$ and $u'^{\diamond\diamond}$.

Since we should eliminate the target user u from the set of his neighbours and his rated items from the set of potentially relevant items. Let us introduce two final neighbourhood forming operators:

$\mathcal{N}_{ij}(u) = u^{ij} \setminus u$ and $\mathcal{N}_{ijk}(u) = u^{ijk} \setminus u^i$, where $i, j, k \in \{\diamond, \prime\}$.

There is also an option to subtract u from its neighbourhood as soon as possible in the chain of operators applied to u. So, there is one more variant for forming potentially relevant items:

$\mathcal{N}_{ijk}^{-u}(u) = (u^{ij} \setminus u)^k \setminus u^i$, where $i, j, k \in \{\diamond, \prime\}$.

4 Similarity measure inspired by Galois operators

Given two users $u, \tilde{u} \in U$, define for them a measure of similarity as $\rho : U^2 \to \mathbb{N}$, $\rho(u, \tilde{u}) = |\{u, \tilde{u}\}'|$. For each user, we will reorder all users in ascending order: $\rho(u, u_1) \leq \rho(u, u_2) \leq \cdots \leq \rho(u, u_n)$. So, each user u generates its renumbering of the set U. Let us define the neighbourhood-forming operator $(.)^{\triangle m} : U \to 2^U$ as follows: $\{u\}^{\triangle m} = \bigcup_{i=1}^{m} \{u_i\}$.

Property 3. $\forall u \in U \; |\{u\}''| \leq m \leq |\{u\}^{\diamond\diamond}| \Leftrightarrow \{u\}'' \subseteq \{u\}^{\triangle m} \subseteq \{u\}^{\diamond\diamond}$

The neighbours of the target user will be considered as $\{u\}^{\triangle m} \setminus \{u\}$ for a given operator. This operator is useful because we exactly specify the number of neighbours, thereby solving the problem of the lack of neighbours or presence of too many of them. Then, we take the k nearest of these neighbours by the other measure (e.g. *cossim*). Since we will look for new items for the prediction in the set $(\{u\}^{\triangle m} \setminus \{u\})' \setminus \{u\}'$, we may need to solve the optimisation problem in this case for m and k choice w.r.t. MAE or Precision and Recall.

5 Algorithm

1. Find neighbours to the target user u using one of the following methods:
   - $\{u\}'' \setminus \{u\}$
   - $\{u\}'^{\diamond} \setminus \{u\}$
   - $\{u\}^{\triangle m} \setminus \{u\}$.
2. Find the set of top k nearest neighbours to the target user u among the neighbours found at the previous step using a measure of similarity *sim*. Denote this set by $\mathcal{N}_k$
3. Find new items for the user u using the method:

$$(\mathcal{N}_k \setminus \{u\})' \setminus \{u\}'$$

4. Make a prediction of the rating for each items found in the previous step. Choose top n of them, if necessary.

Since steps 1 and 3 have been considered, let us discuss steps 2 and 4.

5.1 Similarity measure

Note that this step is not necessary, but useful. The number of neighbours found at the first step can be large, which leads to fewer items for the recommendation. For our experiments, we will use the cosine measure of similarities. This measure is recognised as one of the best estimators of users' similarity [19]. Let u, $\tilde{u} \in U$, $r_{u,i}$ and $r_{\tilde{u},i}$ be the ratings of item $i \in I$ by users u and $\tilde{u}$, respectively, and the vector $\boldsymbol{r}_u = (r_{u,1}, r_{u,2}, \ldots, r_{u,n})$ be the vector of user ratings u. Then we define the cosine measure of users' similarity $cossim : U^2 \to [0, 1]$ as follows[1]:

$$cossim(u, \tilde{u}) = \frac{\boldsymbol{r}_u \cdot \boldsymbol{r}_{\tilde{u}}}{\|\boldsymbol{r}_u\| \, \|\boldsymbol{r}_{\tilde{u}}\|} = \frac{\sum_{i \in I} r_{u,i} r_{\tilde{u},i}}{\sqrt{\left(\sum_{i \in I} (r_{u,i})^2\right) \left(\sum_{i \in I} (r_{\tilde{u},i})^2\right)}}.$$

5.2 Rating prediction

The predicted rating $\hat{r}_{u,i}$ for an item $i \in I$ by a user $u \in U$ is a weighted combination of selected neighbours ratings, which is calculated as a weighted deviation from the average ratings of the neighbours. The general prediction formula is below:

$$\hat{r}_{u,i} = \bar{r}_u + \frac{\sum_{\tilde{u} \in U} (r_{\tilde{u},i} - \bar{r}_{\tilde{u}}) \, sim(u, \tilde{u})}{\sum_{\tilde{u} \in U} | \, sim(u, \tilde{u}) \, |}.$$

6 Experiments

6.1 Data

For test the model, we used data from the GroupLens[2] web site [20]. The data was collected through the MovieLens[3] recommender service during the seven-month period from September 19th, 1997 through April 22nd, 1998. This data has been cleaned up – users who had less than 20 ratings were removed from this data set.

This data set consists of:

- 100 000 ratings (1-5) from 943 users on 1682 movies.
- Each user has rated at least 20 movies.

The data represents 100 000 lines of the form:

$$| \; user\ id \; | \; item\ id \; | \; rating \; | \; timestep \; |.$$

[1] the formula should be adjusted by considering only commonly rated items in the numerator in case of missing ratings by u or $\tilde{u}$

[2] https://grouplens.org/datasets/movielens/

[3] https://movielens.umn.edu

6.2 Training/test set split procedure

We will partially imitate online testing when only a part of information on ratings for test users is known, as our operators are more focused on building recommendations than on forecasting ratings. We will follow the bimodal cross-validation procedure from [29]. To do this, we first find the sets U_{hidden} and I_{hidden}, where:

- U_{hidden} is a randomly selected 20% of all users U,
- I_{hidden} is a randomly selected 20% of all items I.

Then we hide all the information about the ratings at the intersection (U_{hidden}, I_{hidden}) as shown below and call this matrix *trainset*

$$
\begin{array}{c|cccc|ccc}
 & & & & & & I_{hidden} & \\
\hline
 & r_{1,1} & r_{1,2} & \cdots & r_{1,n} & r_{1,n+1} & \cdots & r_{1,l} \\
 & r_{2,1} & r_{2,2} & \cdots & r_{2,n} & r_{1,n+1} & \cdots & r_{2,l} \\
 & \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\
 & r_{m,1} & r_{m,2} & \cdots & r_{m,n} & r_{1,n+1} & \cdots & r_{m,l} \\
\hline
 & r_{m+1,1} & r_{m+1,2} & \cdots & r_{m+1,n} & * & \cdots & * \\
U_{hidden} & \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\
 & r_{k,1} & r_{k,2} & \cdots & r_{k,n} & * & \cdots & *
\end{array}
$$

where $r_{u,i}$ is the rating item i by users u or $*$ if this user did not rate this item yet.
Similarly, *testset* is a matrix containing all the hidden information.
Each experiment will be carried out 100 times to eliminate the dependence on random partitioning.

6.3 Adjusted Precision and Recall

We used standard measures to compare studied models: *Precision* and *Recall*. They can be defined as follows:

$$Precision = \frac{|\{relevant\} \cap \{retrieved\} \cap I_{hidden}|}{|\{retrieved\} \cap I_{hidden}|},$$

$$Recall = \frac{|\{relevant\} \cap \{retrieved\} \cap I_{hidden}|}{|\{relevant\} \cap I_{hidden}|},$$

where for user $u \in U_{hidden}$:

- $\{relevant\}$ is the set of all items that the user u rated,
- $\{retrieved\}$ is the set of all items that we recommended to the user u.

Note special cases:

- $Precision = 1$, if $\{retrieved\} \cap I_{hidden} = \varnothing$,
- $Recall = 1$, if $\{relevant\} \cap I_{hidden} = \varnothing$.

6.4 Testing models

We will test the models based on the algorithm described in Section 5. Since these algorithms different only in the initial obtaining of the neighbours of the target user, we denote them as $/\!/$, $/\!\diamond$, and $\triangle m$.

Model $/\!/$ For this experiment, after applying all the operators, we took $top_k = 5$ of the nearest neighbours by the cosine similarity measure. The result can be observed in table 1:

Table 1. Precision and Recall for the model $/\!/$ over 100K MovieLens dataset

top n recommendation	Precision	Recall	time, sec
1	1.8%	0.01%	6.28
2	0.8%	0.06%	6.28
3	0.7%	0.08%	6.28
⋮	⋮	⋮	⋮
all	7.2%	99%	6.28

Further studies of this model were not carried out since unacceptable trade-off between such low values of Precision and Recall. The problem with this model is that we have neighbours only for $< 3\%$ users, and according to our Galois operator we have $\{\varnothing\}' = I$. Thus, we get the same predicted rating for all available movies, equals the average rating of the target user. Therefore we obtain low values of Precision and Recall.

Model $/\!\diamond$ For this experiment, after applying all the operators, we took $top_k = 5$ of the nearest neighbours by the cosine measure. The result can be observed in table 2:

Table 2. Precision and Recall for the model $/\!\diamond$ over 100K MovieLens dataset

top n recommendation	Precision	Recall	time, sec
1	97.6%	0.5%	1.51
2	97.4%	0.6%	1.51
3	97.6%	0.7%	1.51
⋮	⋮	⋮	⋮
all	97.1%	0.9%	1.51

Further studies of this model were not carried out. In this model, we have a very good Precision, but its Recall does not suit us. This is due to the fact that in this case we have too many neighbours. The average number of neighbours is more than half of the total number of users U. Therefore, the cosine measure does not correctly give us top_k nearest neighbours, e.g., 100% of similarity for the only one commonly rated item. After applying our Galois operator, we get that for $\approx 93\%$ users we have nothing to recommend.

Model $\triangle m$ For this experiment, $top_m = 50$ of the nearest neighbours by measure based on the Galois connection was taken and then $top_k = 5$ of the nearest neighbours by the cosine measure were selected.The result can be observed in table 3:

Table 3. Precision and Recall for the model $\triangle m$ over 100K MovieLens dataset

top n recommendation	Precision	Recall	time, sec
1	72.8%	4.9%	1.44
3	69.5%	10.4%	1.44
5	67.5%	13.3%	1.44
10	67.0%	16.3%	1.44
⋮	⋮	⋮	⋮
all	65.8%	17.7%	1.44

This model produced acceptable results, and thus we can work with it. We have at least one recommendation for $\approx 87.3\%$ users. It can be considered normal for our model. The maximum possible number of recommendations for the user on average is 3-4 movies. So we do not have high recall for all of recommendations. Possibly, this problem should be solved in the transition to larger data (like 10M ratings) or its portion with moderately large profiles of users. Also note that Precision is not greatly reduced by recommending a large number of movies. This is unusual for most recommendation systems. So we can immediately recommend to the user u all the movies that fall into the set $(\mathcal{N}_k \setminus \{u\})' \setminus \{u\}'$ and do not predict the ratings for these movies.

Now we need to understand whether it is necessary to solve the optimisation problem. To do this, we first look at how the values of Precision, Recall, and F_1 score change, fixing k, where F_1 score is considered according as follows: $F_1 = 2\,\frac{Precision \cdot Recall}{Precision + Recall}$.

Then fixing m and see how the Precision and Recall change in depending on k.

We can see from figures 1, 2, 3, and 4 that the Precision is directly proportional to m and k, and the Recall is inversely proportional to m and k. Therefore, it is not necessary to solve an optimisation problem for any of the parameters.

Next, we experimented with the MovieLens dataset composed by 1M ratings and look at the results. For this experiment, $top_m = 100$ of the nearest neighbours by measure based on the Galois operators and then $top_k = 10$ of the nearest neighbours by the cosine measure were taken. We have found out

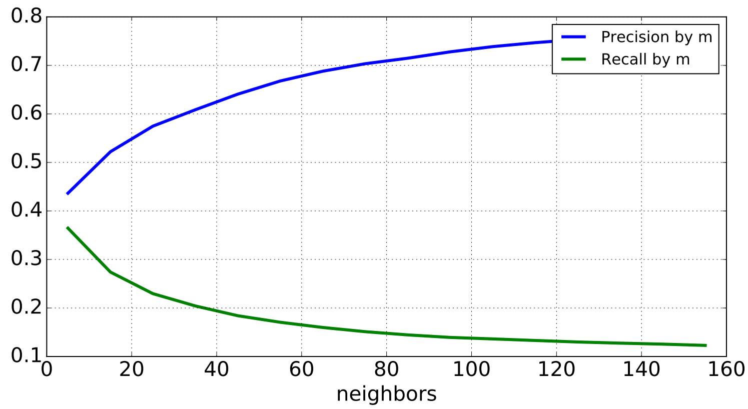


Fig. 1. Precision and Recall by m for the model $\triangle m$

that the Precision is not greatly reduced by recommending a larger number of movies. Therefore, we made an estimate of Prediction and Recall only for all possible recommendations. The result can be observed in table 4:

Table 4. Precision and Recall for the model $\triangle m$ over 1M MovieLens dataset

top n recommendation	Precision	Recall	time, sec
all	61.4%	13.1%	25.7

From this experiment we can conclude that Precision varies slightly and Recall has a small increase in the larger sample. However, we have at least one recommendation for $\approx$ 97.3% users. That can be considered as an acceptable result since in most cases even a few recommendations is enough.

6.5 Slope One

For comparison, we took a well-known model Slope One [34]. In a reputed survey [10] it was shown that this model is one of the best for offline predicting the rating of items, while in our test we see how this model works for making recommendations.The result can be observed in table 5.

This model works worse than ours. This is due to the fact that we do not limit the set of items for prediction.

We can see in figure 5, that sorting by the top n predicted ratings does not give strong effect on the precision of the recommendation by SlopeOne.

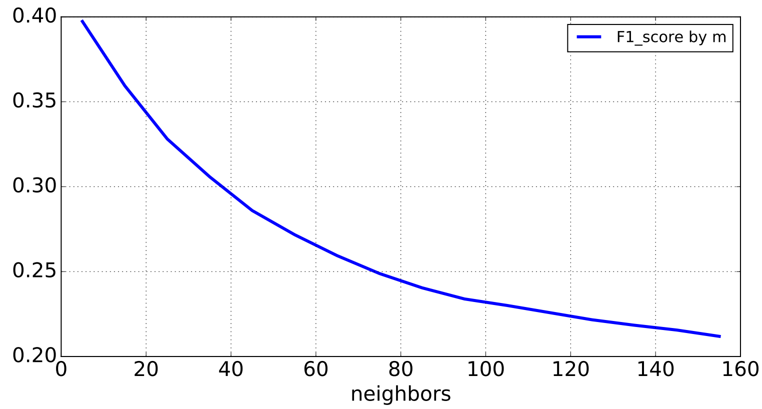


Fig. 2. F_1 score by m for the model $\triangle m$

Table 5. Precision and Recall for SlopeOne over 100K MovieLens dataset

top n recommendation	Precision	Recall	time, sec
1	3.5%	0.2%	7.66
3	8.1%	1.2%	7.66
5	10.2%	2.5%	7.66
10	13.1%	6.8%	7.66
⋮	⋮	⋮	⋮
all	7.7%	100%	7.66

7 Related Work

To the best of our knowledge, the first paper that uses Formal Concept Analysis (FCA) and Galois Connections for Collaborative Filtering was [9]. Later on, a paper on concept-based biclustering for making recommendations over firms-terms contextual advertising problem appeared [25] based on a prior study on the same datset from Yahoo! (former Overture) with spectral clustering techniques [41]; its latest version with revisited experiments and study of biclustering properties is presented in [26]. In parallel, maximal-inclusion biclusters (in fact, formal concepts) were used in similar collaborative filtering scenario [37] based on BiMax algorithm from [36]. A reincarnated study in explicit FCA-terms was done in [6] with large real commercial datasets like PayPal. FCA-based biclusters were also used [21] for recommender system to facilitate educational orientation of Russian school graduates. In [27,23], the authors used concept-based biclustering for making recommendations for crowdsourcing platform Witology to find similar user's ideas and the so-called users-antagonists for stronger team building.

As for interval-like ratings ranges, recommendations based on pattern structures [17] (an extension of FCA-approach for complex data) were firstly intro-

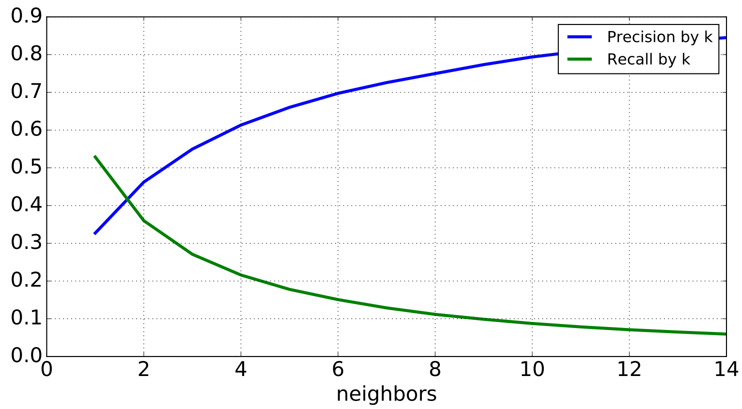


Fig. 3. Precision and Recall by k for the model $\triangle m$

duced and compared with SlopeOne approach in [24]. Another attempt to build interval-based biclusters on MovieLens data was done in [11].

Since after its success in the NetflixPrize competition, a widely accepted method for Recommender Systems is matrix factorization [32], the question on whether Boolean Matrix Factorisation (BMF) provides a competitive approach here emerged. The first answer was received in two works [35,28], where BMF-based solution was compared with Singular Value Decomposition for Collaborative Filtering in terms of MAE and demonstrated equal quality. In the subsequent paper [3], BMF was studied against SVD (compared in terms of MAE, Precision and Recall) over matrices extended by user's and item's features representing the so-called context-aware approach. The main advantages of BMF lie in its high interpretability and promising efficiency of bit-wise Boolean operations whereas its main drawback resides in higher complexity due to combinatorial nature of the optimal number of factors determination (the cover or dimension problem) [8,7].

A separate venue is recommendation for Folksonomies based on higher order extenstions of FCA; let us cite only one recent paper with a detailed introduction of the problem [30].

As an unexpected example, FCA-based collaborative filtering can be also used as an ensemble technique to suggest a proper classifier within classification framework [31].

An interested reader may also refer to a tutorial on FCA for Information Retrieval (IR) [22] and related fields with examples of FCA-based recommender systems as well as a survey on FCA for IR [12].

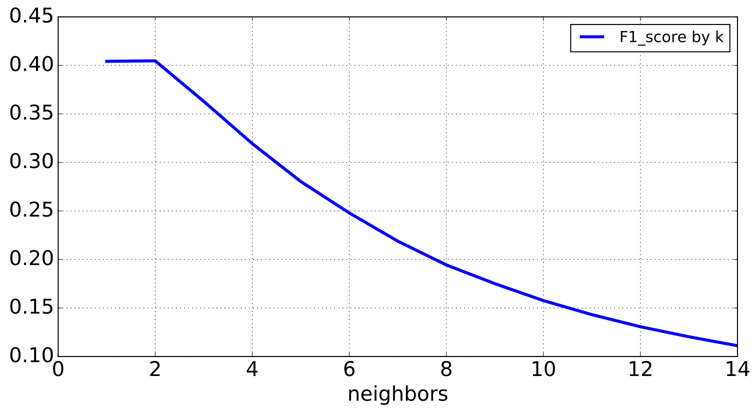


Fig. 4. F_1 score by k for the model $\triangle m$

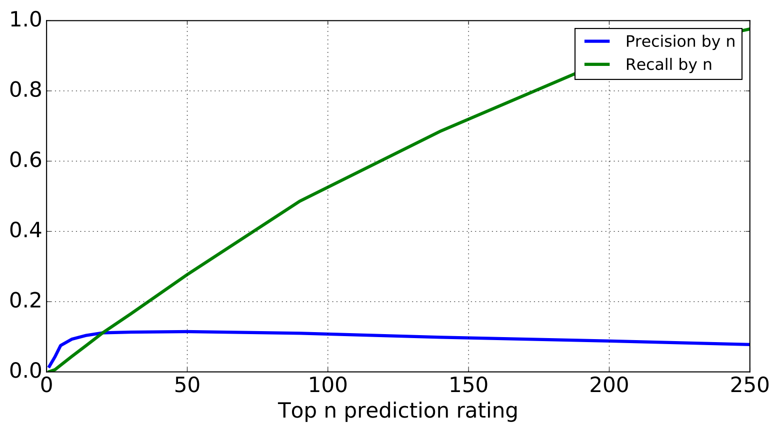


Fig. 5. Precision and Recall for SlopeOne over 100K MovieLens dataset

8 Conclusion

The obtained results seems to be a promising attempt to rethink neighbourhood-based methods in terms of Galois connections and see their theoretical comprehensiveness and limits.

The problem of finding items that the user has not yet looked at, but should see in the near future is relevant for many models. We have managed to treat this problem with the help of Galois operators. Thus this paper being not only an interesting theoretical exercise, again indirectly confirmed the hypothesis: users who rate the same items tend to rate other items similarly.

As for possible venues of the forthcoming work one may take: 1) group recommendations by means of Galois operators; 2) explicit decomposition of $\triangle m$ operator into a combination of two operators from the set of users to the set of items and vice versa; 3) extensive set of experiments with other large real datasets and more recent nearest-neighbours based techniques [1]; 4) scalability issues. A richer set of possible neighbourhood forming operators can be potentially found in [15].

Acknowledgements The first author would like to thank our colleagues either for their piece of advice or earlier contribution, Rakesh Agrawal, Marat Akhamtnurov, Shlomo Berkovskiy, Vladimir Bobrikov, Ivo Düntsch, Chandan Reddy, Mehdy Kaytoue, Sergei Kuznetsov, Denis Kornilov, Maria Mikhailova, Amedeo Napoli, Elena Nenova, Alexander Tuzhilin, and Konstantin Vorontsov.

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

References

1. Adamopoulos, P., Tuzhilin, A.: On over-specialization and concentration bias of recommendations: Probabilistic neighborhood selection in collaborative filtering systems. In: Proceedings of the 8th ACM Conference on Recommender Systems. pp. 153–160. RecSys '14, ACM, New York, NY, USA (2014). https://doi.org/10.1145/2645710.2645752, http://doi.acm.org/10.1145/2645710.2645752
2. Agrawal, R., Srikant, R.: Fast algorithms for mining association rules in large databases. In: VLDB'94, Proceedings of 20th International Conference on Very Large Data Bases, September 12-15, 1994, Santiago de Chile, Chile. pp. 487–499 (1994), http://www.vldb.org/conf/1994/P487.PDF
3. Akhmatnurov, M., Ignatov, D.I.: Context-aware recommender system based on boolean matrix factorisation. In: Yahia, S.B., Konecny, J. (eds.) Proceedings of the Twelfth International Conference on Concept Lattices and Their Applications, Clermont-Ferrand, France, October 13-16, 2015. CEUR Workshop Proceedings, vol. 1466, pp. 99–110. CEUR-WS.org (2015), http://ceur-ws.org/Vol-1466/paper08.pdf
4. Aleskerov, F., Bouyssou, D., Monjardet, B.: Utility Maximization, Choice and Preference. Springer Berlin Heidelberg, Berlin, Heidelberg (2007)
5. Alqadah, F., Reddy, C.K., Hu, J., Alqadah, H.F.: Biclustering neighborhood-based collaborative filtering method for top-n recommender systems. Knowl. Inf. Syst. **44**(2), 475–491 (2015). https://doi.org/10.1007/s10115-014-0771-x, http://dx.doi.org/10.1007/s10115-014-0771-x
6. Alqadah, F., Reddy, C.K., Hu, J., Alqadah, H.F.: Biclustering neighborhood-based collaborative filtering method for top-n recommender systems. Knowl. Inf. Syst. **44**(2), 475–491 (2015). https://doi.org/10.1007/s10115-014-0771-x, https://doi.org/10.1007/s10115-014-0771-x
7. Belohlávek, R., Trnecka, M.: From-below approximations in boolean matrix factorization: Geometry and new algorithm. J. Comput. Syst. Sci. **81**(8), 1678–1697

(2015). https://doi.org/10.1016/j.jcss.2015.06.002, https://doi.org/10.1016/j.jcss.2015.06.002
8. Belohlávek, 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). https://doi.org/10.1016/j.jcss.2009.05.002, https://doi.org/10.1016/j.jcss.2009.05.002
9. du Boucher-Ryan, P., Bridge, D.G.: Collaborative recommending using formal concept analysis. Knowl.-Based Syst. **19**(5), 309–315 (2006). https://doi.org/10.1016/j.knosys.2005.11.017, http://dx.doi.org/10.1016/j.knosys.2005.11.017
10. Cacheda, F., Carneiro, V., Fernández, D., Formoso, V.: Comparison of collaborative filtering algorithms: Limitations of current techniques and proposals for scalable, high-performance recommender systems. ACM Trans. Web **5**(1), 2:1–2:33 (Feb 2011). https://doi.org/10.1145/1921591.1921593, http://doi.acm.org/10.1145/1921591.1921593
11. Codocedo, V., Napoli, A.: Lattice-based biclustering using partition pattern structures. In: Schaub, T., Friedrich, G., O'Sullivan, B. (eds.) ECAI 2014 - 21st European Conference on Artificial Intelligence, 18-22 August 2014, Prague, Czech Republic - Including Prestigious Applications of Intelligent Systems (PAIS 2014). Frontiers in Artificial Intelligence and Applications, vol. 263, pp. 213–218. IOS Press (2014). https://doi.org/10.3233/978-1-61499-419-0-213, https://doi.org/10.3233/978-1-61499-419-0-213
12. Codocedo, V., Napoli, A.: Formal concept analysis and information retrieval - A survey. In: Baixeries, J., Sacarea, C., Ojeda-Aciego, M. (eds.) Formal Concept Analysis - 13th International Conference, ICFCA 2015, Nerja, Spain, June 23-26, 2015, Proceedings. Lecture Notes in Computer Science, vol. 9113, pp. 61–77. Springer (2015). https://doi.org/10.1007/978-3-319-19545-2_4, https://doi.org/10.1007/978-3-319-19545-2_4
13. Cristofor, D., Cristofor, L., Simovici, D.A.: Galois connections and data mining. J. UCS **6**(1), 60–73 (2000). https://doi.org/10.3217/jucs-006-01-0060, http://dx.doi.org/10.3217/jucs-006-01-0060
14. Denecke, K., Erné, M., Wismath, S.L.: Galois Connections and Applications. Springer Netherlands (2004)
15. Dubois, D., Prade, H., Rico, A.: The cube of opposition: A structure underlying many knowledge representation formalisms. In: Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence, IJCAI 2015, Buenos Aires, Argentina, July 25-31, 2015. pp. 2933–2939 (2015), http://ijcai.org/Abstract/15/415
16. Düntsch, I., Gediga, G.: Approximation operators in qualitative data analysis. In: Theory and Applications of Relational Structures as Knowledge Instruments, COST Action 274, TARSKI, Revised Papers. pp. 214–230 (2003). https://doi.org/10.1007/978-3-540-24615-2_10, http://dx.doi.org/10.1007/978-3-540-24615-2_10
17. Ganter, B., Kuznetsov, S.: Pattern structures and their projections. In: Delugach, H., Stumme, G. (eds.) Conceptual Structures: Broadening the Base, Lecture Notes in Computer Science, vol. 2120, pp. 129–142. Springer Berlin Heidelberg (2001). https://doi.org/10.1007/3-540-44583-8_10, http://dx.doi.org/10.1007/3-540-44583-8_10
18. Ganter, B., Wille, R.: Formal Concept Analysis: Mathematical Foundations. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 1st edn. (1999)

19. Gunawardana, A., Shani, G.: A survey of accuracy evaluation metrics of recommendation tasks. J. Mach. Learn. Res. **10**, 2935–2962 (Dec 2009), http://dl.acm.org/citation.cfm?id=1577069.1755883
20. Harper, F.M., Konstan, J.A.: The movielens datasets: History and context. TiiS **5**(4), 19:1–19:19 (2016). https://doi.org/10.1145/2827872, http://doi.acm.org/10.1145/2827872
21. Ignatov, D., Poelmans, J., Zaharchuk, V.: Recommender System Based on Algorithm of Bicluster Analysis RecBi. In: Ignatov, D., Poelmans, J., Kuznetsov, S. (eds.) CEUR Workshop proceedings Vol-757, CDUD'11 - Concept Discovery in Unstructured Data. pp. pp. 122–126 (2011)
22. Ignatov, D.I.: Introduction to formal concept analysis and its applications in information retrieval and related fields. In: Braslavski, P., Karpov, N., Worring, M., Volkovich, Y., Ignatov, D.I. (eds.) Information Retrieval - 8th Russian Summer School, RuSSIR 2014, Nizhniy, Novgorod, Russia, August 18-22, 2014, Revised Selected Papers. Communications in Computer and Information Science, vol. 505, pp. 42–141. Springer (2014). https://doi.org/10.1007/978-3-319-25485-2_3, https://doi.org/10.1007/978-3-319-25485-2_3
23. Ignatov, D.I., Kaminskaya, A.Y., Konstantinova, N., Malioukov, A., Poelmans, J.: Fca-based recommender models and data analysis for crowdsourcing platform witology. In: Graph-Based Representation and Reasoning - 21st International Conference on Conceptual Structures, ICCS 2014, Iaşi, Romania, July 27-30, 2014, Proceedings. pp. 287–292 (2014). https://doi.org/10.1007/978-3-319-08389-6_24, http://dx.doi.org/10.1007/978-3-319-08389-6_24
24. Ignatov, D.I., Kornilov, D.: RAPS: A recommender algorithm based on pattern structures. In: Proceedings of the 4th International Workshop "What can FCA do for Artificial Intelligence?", FCA4AI 2015, co-located with the International Joint Conference on Artificial Intelligence (IJCAI 2015), Buenos Aires, Argentina, July 25, 2015. pp. 87–98 (2015), http://ceur-ws.org/Vol-1430/paper9.pdf
25. Ignatov, D.I., Kuznetsov, S.O.: Concept-based Recommendations for Internet Advertisement. In: Belohlavek, R., Kuznetsov, S.O. (eds.) Proc. CLA 2008. CEUR WS, vol. Vol. 433, pp. 157–166. Palack University, Olomouc, 2008 (2008)
26. Ignatov, D.I., Kuznetsov, S.O., Poelmans, J.: Concept-based biclustering for internet advertisement. In: ICDM Workshops. pp. 123–130 (2012)
27. Ignatov, D.I., Mikhailova, M., Zakirova, A.Y., Malioukov, A.: Recommendation of ideas and antagonists for crowdsourcing platform witology. In: Information Retrieval - 8th Russian Summer School, RuSSIR 2014, Nizhniy, Novgorod, Russia, August 18-22, 2014, Revised Selected Papers. pp. 276–296 (2014). https://doi.org/10.1007/978-3-319-25485-2_9, http://dx.doi.org/10.1007/978-3-319-25485-2_9
28. Ignatov, D.I., Nenova, E., Konstantinova, N., Konstantinov, A.V.: Boolean matrix factorisation for collaborative filtering: An fca-based approach. In: Artificial Intelligence: Methodology, Systems, and Applications - 16th International Conference, AIMSA 2014, Varna, Bulgaria, September 11-13, 2014. Proceedings. pp. 47–58 (2014). https://doi.org/10.1007/978-3-319-10554-3_5, http://dx.doi.org/10.1007/978-3-319-10554-3_5
29. Ignatov, D.I., Poelmans, J., Dedene, G., Viaene, S.: A new cross-validation technique to evaluate quality of recommender systems. In: Kundu, M., Mitra, S., Mazumdar, D., Pal, S. (eds.) Perception and Machine Intelligence, Lecture Notes in Computer Science, vol. 7143, pp. 195–202. Springer Berlin Heidel-

berg (2012). https://doi.org/10.1007/978-3-642-27387-2_25, http://dx.doi.org/10.1007/978-3-642-27387-2_25
30. Jelassi, M.N., Yahia, S.B., Nguifo, E.M.: Towards more targeted recommendations in folksonomies. Social Netw. Analys. Mining **5**(1), 68:1–68:18 (2015). https://doi.org/10.1007/s13278-015-0307-8, https://doi.org/10.1007/s13278-015-0307-8
31. Kashnitsky, Y., Ignatov, D.I.: Can fca-based recommender system suggest a proper classifier? In: Proceedings of the 4th International Workshop "What can FCA do for Artificial Intelligence"? co-located with the European Conference on Artificial Intelligence, FCA4AI@ECAI 2014. pp. pp. 17–26 (2014), http://arxiv.org/abs/1504.05473
32. Koren, Y., Bell, R., Volinsky, C.: Matrix factorization techniques for recommender systems. Computer **42**(8), 30–37 (2009). https://doi.org/http://doi.ieeecomputersociety.org/10.1109/MC.2009.263
33. 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 (2005). https://doi.org/10.1007/11528784_11, http://dx.doi.org/10.1007/11528784_11
34. Lemire, D., Maclachlan, A.: Slope one predictors for online rating-based collaborative filtering. In: Proceedings of the 2005 SIAM International Conference on Data Mining. pp. 471–475 (2005). https://doi.org/10.1137/1.9781611972757.43, http://epubs.siam.org/doi/abs/10.1137/1.9781611972757.43
35. Nenova, E., Ignatov, D.I., Konstantinov, A.V.: An fca-based boolean matrix factorisation for collaborative filtering. In: International Workshop FCA meets IR at ECIR 2013. vol. Vol. 977, CEUR Workshop Proceeding, pp. 57–73 (2013)
36. Prelic, A., Bleuler, S., Zimmermann, P., Wille, A., Bühlmann, P., Gruissem, W., Hennig, L., Thiele, L., Zitzler, E.: A systematic comparison and evaluation of biclustering methods for gene expression data. Bioinformatics **22**(9), 1122–1129 (2006). https://doi.org/10.1093/bioinformatics/btl060, https://doi.org/10.1093/bioinformatics/btl060
37. Symeonidis, P., Nanopoulos, A., Papadopoulos, A.N., Manolopoulos, Y.: Nearest-biclusters collaborative filtering based on constant and coherent values. Inf. Retr. **11**(1), 51–75 (2008)
38. Wolski, M.: Galois connections and data analysis. Fundam. Inform. **60**(1-4), 401–415 (2004), http://content.iospress.com/articles/fundamenta-informaticae/fi60-1-4-26
39. Zaki, M.J., Meira Jr., W.: Data Mining and Analysis: Fundamental Concepts and Algorithms. Cambridge University Press (2014), http://www.dataminingbook.info/
40. Zaki, M.J., Hsiao, C.: Efficient algorithms for mining closed itemsets and their lattice structure. IEEE Trans. Knowl. Data Eng. **17**(4), 462–478 (2005). https://doi.org/10.1109/TKDE.2005.60
41. Zhukov, L.E.: Spectral clustering of large advertiser datasets. Technical report, Overture R&D (April 2004)