Proceedings of the International Workshop ”What can FCA do for Artificial Intelligence?” (FCA4AI 2014)

Sergei O. Kuznetsov, Amedeo Napoli, Sebastian Rudolph

Workshop Notes

International Workshop

“What can FCA do for Artificial Intelligence?”

FCA4AI

European Conference on Artificial Intelligence

ECAI 2014

August 19, 2014

Prague, Czech Republic

Editors

Sergei O. Kuznetsov (NRU HSE Moscow)

Amedeo Napoli (LORIA Nancy)

Sebastian Rudolph (TU Dresden)

http://fca4ai.hse.ru/2014/

Preface

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

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

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

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

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

The workshop is dedicated to discuss such issues. This year, the papers submitted to the workshop were carefully peer-reviewed by three members of the program committee and 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 Schools of Economics, Moscow, Russia

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

Sebastian Rudolph
Technische Universitaet Dresden, Germany

Program Committee

Mathieu D'Aquin (Open University, UK)

Jaume Baixeries, UPC Barcelona, Catalunya

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

Claudio Carpineto (Fondazione Ugo Bordoni, Roma, Italy)

Felix Distel (Technische Universitaet Dresden, Germany)

Florent Domenach (University of Nicosia, Cyprus)

Peter Eklund (University of Wollongong, Australia)

Cynthia-Vera Glodeanu (Technische Universitaet Dresden, Germany)

Marianne Huchard (LIRMM/Université de Montpellier, France)

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

Mehdi Kaytoue (INSA-LIRIS Lyon, France)

Florence Le Ber, Université de Strasbourg, France

Nizar Messai (Université de Tours, France)

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

Jan Outrata (Palacky University, Olomouc, Czech Republic)

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

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

Chedy Raïssi (Inria/LORIA Nancy, France)

Artem Revenko, Technische Universitaet Dresden, Germany

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

Baris Sertkaya (SAP Dresden, Germany)

Henry Soldano (Université de Paris-Nord, France)

Laszlo Szathmary, University of Debrecen, Hungary

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

Table of Contents

Contents

Abstraction, taxonomies, connectivity : from AI to FCA and back

Henry Soldano

Université Paris 13, Sorbonne Paris Cité, L.I.P.N UMR-CNRS 7030
F-93430, Villetaneuse, France

Abstract. We describe an experience of transfer and ideas exchange between AI and FCA. The original motivation was a data analysis problem in which there were objects, structured attributes together with a categorization of objects, leading to the idea that in some way the categorization should alter the selection of interesting patterns. On one hand, soon it appeared that to investigate the data it was interesting to use various degrees of coarseness not only on the pattern language but also on the extensions, i.e. the support, following the data mining terminology, of the patterns. On an other hand, closed patterns are known to summarize the whole set of frequent patterns, and FCA proposes to organize these closed patterns into a concept lattice, each node of which was a pair made of a closed pattern and its extension, but there were no known way to use categorization and relative coarseness in a flexible way. On the FCA technical side, this led us in particular to extend concept lattices to smaller conceptual structures, called abstract concept lattices, in which the extension of a term/motif/pattern in a set of objects is constrained by an external a priori view of the data together with a parameter controlling the degree of coarseness [1,2]. A closer view to the structure of the corresponding extensional space led us back to AI : we called such a structure an abstraction as it captured part of the notion of domain abstraction as it has been investigated in AI [3]. The most interesting transfer back to AI relied on the following observation : the set of abstract implications related to these abstract lattices had a particular meaning that was naturally expressed in modal logics. A direct consequence is that the notion of abstraction necessary to preserve the lattice structure of closed patterns, i.e. to preserve the concept lattice structure, defined a particular class of modal logics, we called modal logics of abstraction, whose properties led to a new kind of semantics [4]. In few words, in such a modal logics the modal connector, usually known as a "necessity" connector and represented as a square, could be translated as an "abstraction" operator, i.e. a sentence as "$\Box$ P" was understood as "Abstractly P". The corresponding semantics relied on a covering of the universe, and could not, except in particular cases, be translated as the standard "possible world" semantics of the most common modal logics.
More recently, new trends in AI and data mining orient research towards linked data. The same formal notion of abstraction can be defined on graphs, and this leads to a way to extract closed patterns from graphs whose vertices are objects described in a FCA framework, therefore allowing to investigate attributed graphs [5]. Finally, recent work on data mining discuss closure operators on partially ordered pattern languages weaker than lattices, as the set of connected subgraphs of some graph, which leads to extend formal concept analysis beyond the lattice

structure still preserving a large part of the nice formal structures and results of FCA[6].

References

1. Pernelle, N., Rousset, M.C., Soldano, H., Ventos, V.: Zoom: a nested Galois lattices-based system for conceptual clustering. J. of Experimental and Theoretical Artificial Intelligence **2/3**(14) (2002) 157–187
2. Ventos, V., Soldano, H.: Alpha Galois lattices: An overview. In Ganter, B., (Eds), R.G., eds.: International Conference on Formal Concept Analysis (ICFCA). Volume 3403 of Lecture Notes on Computer Science. Springer Verlag (2005) 298–313
3. Soldano, H., Ventos, V.: Abstract Concept Lattices. In Valtchev, P., Jäschke, R., eds.: International Conference on Formal Concept Analysis (ICFCA). Volume 6628 of LNAI., Springer, Heidelberg (2011) 235–250
4. Soldano, H.: A modal view on abstract learning and reasoning. In Genesereth, M., Revesz, P.Z., eds.: Ninth Symposium on Abstraction, Reformulation, and Approximation, SARA 2011, Cardona, Catalonia, SPAIN, AAAI Press (July 2011) 99–106
5. Soldano, H., Santini, G.: Graph abstraction for closed pattern mining in attributed network. In: European Conference in Artificial Intelligence (ECAI), IOS Press (2014) to appear
6. Soldano, H.: Closed patterns and abstraction beyond lattices. In Glodeanu, C., Kaytoue, M., Sacarea, C., eds.: International Conference on Formal Concept Analysis (ICFCA). Volume 8478 of LNAI., Springer, Heidelberg (2014) to appear

Using Formal Concept Analysis to Create Pathways through Museum Collections

Tim Wray, Peter Eklund**

Faculty of Engineering and Information Science
University of Wollongong
Northfields Avenue, NSW 2522, Australia

Abstract. This paper presents *A Place for Art* – an iPad app that allows users to explore an art collection via semantically linked pathways that are generated using Formal Concept Analysis. The app embraces the information seeking approach of exploration and is based on the idea that showing context and relationships among objects in a museum collection augments an interpretive experience. The fundamental interaction metaphor inherent in *A Place for Art* relies on Formal Concept Analysis so the interface has embedded within it the semantic clustering features of machine learning in artificial intelligence.

Keywords: Intelligent Interfaces, Formal Concept Analysis

This paper presents "A Place for Art", a working artefact developed by a team of developers of which the authors are members and that has been reported more extensively elsewhere [17] albeit not to a AI or FCA audience. The work can be framed as a contemporary extension of work using FCA for Information Retrieval[6, 3].

A Place for Art showcases a collection of contemporary and Australian indigenous works from the University of Wollongong's Art Collection. It is a digitized companion piece to the print publication of the same name [9]. The app provides access to 77 works and accompanying short essay pieces that feature the history of the collection and significance to its local region. The key result is a semantic navigation concept driven interaction paradigm that has Formal Concept Analysis (FCA) at its heart. A demonstration of the design – an iPad app – is a very important companion to this written text and the reader is encouraged to download, install and run the app while reading the paper.

1 Pathways through an Art Collection using FCA

Interaction and navigation in A Place for Art relies on users creating and exploring their own path through the collection: an approach that is well supported by the literature on information seeking in museum collections. For example: Skov[13] found that online visitors demonstrated exploratory behaviors such as

** On leave in 2014 to the IT University of Copenhagen, Denmark serendipity and, when finding the unexpected, exhibited meaning making qualities, i.e. following paths and making implicit connections between objects. Further, Goodale et al. [7] conceptualize the pathway as a guiding metaphor to characterize the design of digital artifacts that support creative and divergent exploration of cultural data-sets. In their findings, the authors elaborate that pathways can be both used as a means to creatively explore a collection and also to structure the relationships among the objects.

Previous work has experimented with semantically linking museological content [12, 1], such as the semantically enriched search platform produced by Schreiber et al. [11]. In our work, we build a custom user interface that embraces the pathway metaphor by allowing the user to navigate clusters of related art content and 'branch off' at specific points of interest. To do this, we employ Formal Concept Analysis (FCA) to derive clusters of related artworks, and then exploit FCA's relational properties to generate semantically linked pathways. In this paper, we apply the terms *convergence* to describe the way FCA can be used to cluster related objects, and *divergence* to the way a visitor could potentially move through different sets of object clusters based on their shifting points of interest. We elaborate on this *convergence - divergence* approach to navigation, along with a brief overview of FCA in the following section.

1.1 Formal Concept Analysis, Convergence and Divergence

Formal Concept Analysis (FCA) was developed in the early 1980s as a mathematization of the human cognitive constructs of concepts and concept hierarchies [14]. Concepts are understood as basic units of thought shaped by observations of existing phenomena and formed in dynamic processes within social and cultural environments ([15], p. 2). According to its philosophical definition [15], a concept is composed of a set of objects as its extension, and all attributes, properties and meanings that apply to those objects as its intension. As an example, if one considers the idea of works that depict heavy industry and the Illawarra[1] (its intension), which we derived from the text analysis of the printed catalogue, there are 7 paintings (its extension) in the collection which have these attributes (see Table 1); a concept is therefore defined as the simultaneous perception of its intension and extension, i.e., the compositional qualities of those paintings (as attributes) and the actual paintings (as objects) defined via those attributes.

In FCA, concepts are mathematized as formal concepts defined as a pair (A, B) where A and B respectively are elements of the set of objects G (the formal concept's extension) and the set of attributes that describe those objects M (the formal concept's intension). Concepts are never perceived in isolation, but in the context of existing phenomena. By interpreting concepts in context, one can derive implications and perceive their relational and spatial properties to other concepts. For the purposes of interpreting works in A Place for Art, we consider groups of artworks that share equivalences as concepts that are a part of the 77 works that compose its collection, its context. In FCA, the context is mathematized as a formal context defined as a $\mathcal{K}(G, M, I)$ where G

[1] The Illawarra is the name of a region 80-160km south of Sydney, Australia.

Table 1. A sample of formal concepts from A Place for Art

Formal concept, expressed in natural language	No. of objects
paintings that depict the Illawarra	8
works that evoke identity issues and social critique	6
surreal works that depict animal imagery	6
vibrant and abstract paintings	11
intricate works that depict nature	6
vibrant works that evoke a sense of calm	6
works that depict heavy industry and the Illawarra	7

and M respectively describe its set of objects and attributes and I describes the associations between them. Formally, $I \subseteq G \times M$ is a binary relation where $(g, m) \in I$ is read object g has attribute m.

A valuable layer of meaning is added when concepts are perceived in context. One way of inferring meaning is by deriving attribute implications. These implications provide the assertion that within a given context, if all objects with X attributes also possess Y attributes, then X infers Y. Applying this form of inference gives the ability to gain insights into the implicit relationships and phenomena within the collection. In A Place for Art, it infers that, for example, all works that depict natural landscapes are painted with coarse brush strokes, or that all the depictions of heavy industry in the A Place for Art collection also all take place in the Illawarra. Using the latter assertion as an example, these attribute implications are formed by the way formal concepts are constructed: for a give formal concept (A, B), that has an attribute set $M =$ {'heavy industry'}, let G be composed of all objects that possess M, giving the result of: $G =$ {'Waiting, Port Kembla', 'Foundry Men', 'Steel Works BHP',...} Now let M be all attributes common to objects in G, giving the result of: $M =$ {'heavy industry', 'the Illawarra'} G and M are then combined as (A, B) to create a closed concept. The additional attributes that were derived from this operation give rise to their implication, in this case, 'heavy industry' $\rightarrow$ 'the Illawarra'. In A Place for Art, formal concepts are computed using the PCbO algorithm [8] but the choice of algorithm is not important.

Within its context of 77 artworks, there are a total of 330 formal concepts, 7 of these formal concepts are expressed in natural language are shown in Table 1. By deriving clusters from data and inferring association rules, formal concepts provide the mathematical realization of – what we term – *a convergence*: the way a group of otherwise disparate works of art are represented as a meaningful whole. A Place for Art also employs purpose built algorithms for describing concepts in natural language.

The examples shown here and in Table 1 are direct outputs generated from these algorithms. When a formal concept is expressed in natural language, the algorithm takes its intension and orders it based on their semantic qualifiers and parts of speech, such as whether they depict the work itself ('painting', 'screen-print', etc.), are adjectival ('surreal', 'vibrant', etc.) or are otherwise appended as clause fragments ('identity issues', 'a sense of calm', etc.). Using basic principles of grammar and sentence construction, these attributes are then conjoined to produce a statement. The algorithms also take into consideration whether the natural language statement should be expressed in a singular or plural form, given that, according to the principles of FCA, individual objects are also formal concepts. These natural language statements are used to convey the semantic meaning of the convergences as human-readable, narrative-like statements.

Meaning is further conveyed when concepts are observed in relation to one another. Concepts are inherently spatial and relational, as connections of concepts are networked to create a concept lattice [14] or are spatially conveyed via a measure of their concept similarity and distance. One common method of constructing a knowledge space in FCA is via the exploitation of the subconcept/superconcept relationship. Within a context, the complete set of formal concepts– ordered by this relationship – induces a concept hierarchy – an implicitly structured collation of human knowledge that can be represented visually as a concept lattice or line diagram.

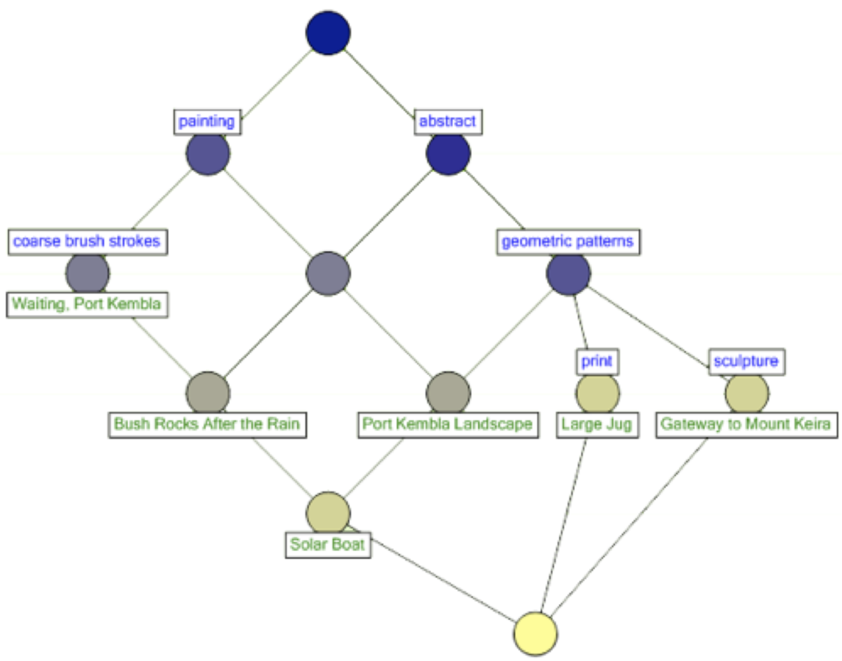


Fig. 1. A line diagram showing a small selection of artworks from A Place for Art.

Following the example in Fig. 1, the concept 'abstract paintings with geometric patterns and coarse brush strokes' – depicted by the artwork titled 'Solar Boat' as it appears bottom-right in the concept lattice is a subconcept of abstract paintings with geometric patterns – that also includes the artwork Port Kembla Landscape, appearing to its top-right – which in turn is also subconcept of both abstract paintings and abstract works with geometric patterns. In FCA, a formal concept (A, B) is a subconcept of (C, D) (expressed $(A, B) < (C, D)$) if $A \subseteq C$ and $B \supseteq D$. Likewise, a formal concept (A, B) is a superconcept of (C, D) (expressed as $(A, B) > (C, D)$) if $A \supseteq C$ and $B \subseteq D$.

Relations between concepts can be understood in terms of conceptual neighbors – concepts that are more general or more specific to one another within the concept hierarchy. A concept (A, B) is said to be the lower neighbor, of concept (C, D) if $(A, B) < (C, D)$ such that there is no concept (E, F) that gives rise to $(A, B) < (E, F) < (C, D)$. Likewise a concept (A, B) is said to be the upper neighbor, of concept (C, D) if $(A, B) > (C, D)$ such that there is no concept (E, F) that gives rise to $(A, B) > (E, F) > (C, D)$. Following the running example, the concepts 'abstract paintings with geometric patterns and coarse brush strokes', 'abstract paintings and abstract works with geometric patterns' are all conceptual neighbors of 'abstract paintings with geometric patterns'.

Concepts can also be related in terms of similarity: i.e., certain concepts can be considered conceptually similar based on sharing some common objects and attributes, with the mathematics of such described in Formica [5]. Furthermore, concept similarity provides a fast approximation for identifying a concept s neighbors. In Table 2, for example, for the formal concept 'abstract paintings with geometric patterns', its immediate lower neighbor 'vibrant and abstract paintings with geometric patterns' is identified as its most similar concept. It is also partially similar to the more distant 'energetic and vibrant paintings', more so than the notionally relevant paintings that depict the Illawarra and the almost irrelevant works that depict animal imagery.

These concept similarity metrics are also valid for comparing object-to-concept as well as concept-to-concept relationships, since, according to the mathematics of FCA, formal concepts can represent individual objects called object concepts. For instance, consider the print 'Illawarra Flame Tree and Bowerbird' (Fig. 2) which is, according to the natural language description of its object concept, "an intricate and vibrant print that depicts animal imagery, the Illawarra and nature and has red and blue tones" Using the concept similarity metrics described above, we can observe the multiplicity of contexts that this object can be interpreted in, and determine which concepts are 'most' similar to the artwork (shown below in Table 3).

Table 2. Concepts similar to abstract paintings with geometric patterns

Formal concept, expressed in natural language	**Similarity Score**
vibrant and abstract paintings with geometric patterns	0.80
energetic and abstract paintings with geometric patterns	0.73
abstract paintings	0.60
energetic and vibrant paintings	0.32
paintings that depict the Illawarra	0.16
works that depict animal imagery	0.03

Examining the objects in these contexts and ordering them by relevance gives the ability to induce new objects and draw equivalencies between then.

This provides the basis of how *divergences* work in A Place for Art. Divergent exploration is based on the idea that every turning point within the collection should infer new objects and enlighten new connections. Hence, divergences have two design criteria: a) they should infer new objects based on similarity of an object of interest and b) the resulting pathways should always infer new objects that have not yet been presented previously by prior convergences. This approach avoids repetition and circularity navigating the information space, where the sum of divergences affords a gradual unveiling of the collection by highlighting new works of interest.

Table 3. Concepts similar to “Illawarra Flame Tree and Bowerbird”.

Formal concept, expressed in natural language	Similarity	#objects
intricate prints that depict animal imagery and nature	0.50	2
works that depict animal imagery and have red tones	0.38	2
intricate works that depict nature	0.21	3
works that have blue tones	0.13	8
works that depict animal imagery	0.11	11
vibrant works	0.08	23

Fig. 2. “Illawarra Flame Tree and Bowerbird”.

When considering what objects to show in a divergence, all the other objects represented by the ‘pivot point’ are determined, along with the total set of objects in prior convergences. Based on the object depicted in its pivot point, an object concept is constructed, in which a set of formal concepts containing that object are retrieved. Using concept similarity metrics, the formal concept that is selected is the one that has the highest similarity score containing objects not part of a prior set of convergences. These new objects are then reconstructed as a formal concept, so that any additional attributes are implied from this reduced set, which is then presented as an adjoining pathway from the pivot point.

2 Refections on Divergence/Convergence

The convergent-divergent interaction paradigm is also supported by the theory and philosophy of Formal Concept Analysis in two ways. Formal concepts – just like the human concepts they are modeled on "express subjectivity and emotions" [15]. In the context of museum collections, this provides the ability to model human meaning and thought in the form of sentiments and conjectures; within the A Place for Art, the idea that certain works have 'warm tones' or 'evoke a sense of calm'. It also offers a way of creating inferences and structures from these conjectures without their explicit encoding in other formal knowledge representation schemas.

The second implication concerns the relational qualities of the conceptual structure, a quality best observed from the concept lattice (Fig 1). Wille [14] introduces the notion of conceptual landscapes as a metaphor to describe the inherently spatial properties of human knowledge, drawing parallels to Murray's [10] landscape paradigm. Whereas Murray's perspective refers to spatial qualities of navigating information spaces, Wille's conceptual landscapes describe the way humans produce, communicate and consume knowledge. Yet, like Murray, Wille alludes to the metaphorical adoption of landscape motifs that dictate the way humans interact with information spaces, and shares the view that computers are a medium, rather than a container for the storage and display of data. He argues "The idea of a landscape is becoming increasingly influential in the field of knowledge representation and processing. Especially, the frequently used term of 'navigation' suggests this idea is becoming a leading metaphor. That view is also supported by the development of computers as a medium. This development shows that it is time for explicating the pragmatic landscape paradigm for knowledge processing" (Wille[16]).

From this proposition, Wille defines the practice and discipline of Conceptual Knowledge Processing [4, 14] as a set of techniques that make use of a variety of conceptual structures to augment human activities in knowledge representation, processing and communication. Within this framework, Wille defines the act of identification – the positioning and contextualization of objects, concepts or data elements in relation to other objects, concepts or data elements and exploration – understood as the act of seeking without a goal, or where the item in question is vague or not well known.

3 Conclusion

The mathematization of the convergent-divergent paradigm of navigation and its implementation in A Place for Art highlight the spatial properties of its interaction and of its underlying conceptual structures. These structures recognize the inherently polyvalent nature of knowledge and their interpretation that orbits and interpretation of museum objects [2], and, through the convergent-divergent interaction paradigm, A Place for Art intends to afford creative exploration of these structures in a non-didactic way. Evaluation of the interface among user communities reinforces the success of the approach and these results are described in the presentation at FCA4AI but are to be fully published elsewhere.

References

1. L. Aroyo, N. Stash, Y. Wang, P. Gorgels, and L. Rutledge. CHIP Demonstrator: Semantics-Driven Recommendations and Museum Tour Generation. In K. Aberer, N. Choi, K-S.and Noy, D. Allemang, K. Lee, L. Nixon, J. Golbeck, P. Mika, D. Maynard, R. Mizoguchi, G. Schreiber, and P. Cudré-Mauroux, editors, *The Semantic Web*, volume 4825 of *LNCS*, pages 879–886. Springer Berlin / Heidelberg, 2007.
2. S. Cairns. Mutualizing museum knowledge: Folksonomies and the changing shape of expertise. curator. *The Museum Journal*, 56, 2013.
3. C. Carpineto and G. Romano. *Concept Data Processing: Theory and Practice*. Wiley, 2004.
4. P. Eklund and R. Wille. Semantology as basis for conceptual knowledge processing. In *Formal Concept Analysis*, pages 18–38. Springer Berlin Heidelber, 2007.
5. A. Formica. Concept similarity in formal concept analysis: An information content approach. knowledge-based systems. *Knowledge-Based Systems*, 21(1):80–87, 2008.
6. R. Godin, R. Missaoui, and A. April. Experimental comparison of navigation in a galois lattice with conventional information retrieval methods. *International journal of Man-Machine Studies*, 38(5):747–767, May 1993.
7. P. Goodale, P. Clough, N. Ford, M. Hall, M. Stevenson, S. Fernando, N. Aletras, K. Fernie, O. Archer, and A. Polo. User-Centred Design to Support Exploration and Path Creation in Cultural Heritage Collections, 2012.
8. P. Krajca, J. Outrata, and V. Vychodil. Parallel recursive algorithm for formal concept analysis. *6th International Conference on Concept Lattices and their Applications*, pages 71–80, 2008.
9. A. Lawson and C. Judd. *A Place for Art*. University of Wollongong Press, 2012.
10. J. Murray. *Inventing the Medium*. MIT Press, Cambridge, MA, 1012.
11. G. Schreiber, A. Amin, L. Aroyo, M. van Assem, V. de Boer, L. Hardman, M. Hildebrand, B. Omelayenko, J. van Osenbruggen, A. Tordai, J. Wielemaker, and B. Wielinga. Semantic annotation and search of cultural-heritage collections: The MultimediaN E-Culture demonstrator. *Web Semantics: Science, Services and Agents on the World Wide Web*, 6(4):243–249, 2008.
12. A Sheth, S. Staab, M. Dean, M. Paolucci, D. Maynard, T. Finin, K. Thirunarayan, J. Wielemaker, M. Hildebrand, J. van Ossenbruggen, and G. Schreiber. *Thesaurus-Based Search in Large Heterogeneous Collections*, volume 5318, pages 695–708. Springer Berlin / Heidelberg, 2010.
13. M. Skov. *The Reinvented Museum : Exploring Information Seeking Behaviour in a Digital Museum Context*. PhD thesis, The University of Copenhagen, Royal School of Library and Information Science, 2009.
14. R. Wille. Conceptual landscapes of knowledge: A pragmatic paradigm of knowledge processing. *Classification in the Information Age*, pages 344–356, 1999.
15. R. Wille. Formal concept analysis as mathematical theory of concepts and concept hierarchies. In R. Wille B. Ganter, G. Stumme, editor, *Formal Concept Analysis*, volume 3626, pages 47–70. Springer Verlag, 2005.
16. R. Wille. Restructuring lattice theory: An approach based on hierarchies of concepts. In S. Ferre and S. Rudolph, editors, *Formal Concept Analysis*, volume 5548, pages 314–339. Springer Verlag, 2009.
17. T. Wray, P. Eklund, and K. Kautz. Pathways through information landscapes: Alternative design criteria for digital art collections. In *The 34th International Conference on Information Systems (ICIS)*. AIS Press, 2013.

Can FCA-based Recommender System Suggest a Proper Classifier?

Yury Kashnitsky and Dmitry I. Ignatov

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

Abstract. The paper briefly introduces multiple classifier systems and describes a new algorithm, which improves classification accuracy by means of recommendation of a proper algorithm to an object classification. This recommendation is done assuming that a classifier is likely to predict the label of the object correctly if it has correctly classified its neighbors. The process of assigning a classifier to each object is based on Formal Concept Analysis. We explain the idea of the algorithm with a toy example and describe our first experiments with real-world datasets.

1 Introduction

The topic of Multiple Classifier Systems (MCSs) is well studied in machine learning community [1]. Such algorithms appear with different names – mixture of experts, committee machines, classifier ensembles, classifier fusion and others.

The underlying idea of all these systems is to train several (base) classifiers on a training set and to combine their predictions in order to classify objects from a test set [1]. This idea probably dates back to as early as the 18^{th} century. The Condorcet's jury theorem, that was formulated in 1785 in [2], claims that if a population makes a group decision and each voter most likely votes correctly, then adding more voters increases the probability that the majority decision is correct. The probability that the majority votes correctly tends to 1 as the number of voters increases. Similarly, if we have multiple weak classifiers (meaning that classifier's error on its training data is less than 50% but greater than 0%), we can combine their predictions and boost the classification accuracy as compared to those of each single base classifier.

Among the most popular MCSs are bagging [3], boosting [7], random forests [9], and stacked generalization (or stacking) [10].

In this paper, we present one more algorithm of such type – Recommender-based Multiple Classifier System (RMCS). Here the underlying proposition is that a classifier is likely to predict the label of the object from a test set correctly if it has correctly classified its neighbors from a training set.

The paper is organized as follows. In chapter 2, we discuss bagging, boosting and stacking. In Section 3, we introduce basic definitions of Formal Concept

Analysis (FCA). Section 4 provides an example of execution of the proposed RMCS algorithm on a toy synthetic dataset. Then, Section 5 describes the RMCS algorithm itself. Further, the results of the experiments with real data are presented. Section 7 concludes the paper.

2 Multiple Classifier Systems

In this chapter, we consider several well-known multiple classier systems.

2.1 Bagging

The *bootstrap sampling* technique has been used in statistics for many years. *Bootstrap aggregating*, or *bagging*, is one of the applications of bootstrap sampling in machine learning. As sufficiently large data sets are often expensive or impossible to obtain, with bootstrap sampling, multiple random samples are created from the source data by sampling with replacement. Samples may overlap or contain duplicate items, yet the combined results are usually more accurate than a single sampling of the entire source data achieves.

In machine learning the bootstrap samples are often used to train classifiers. Each of these classifiers can classify new instances making a prediction; then predictions are combined to obtain a final classification.

The aggregation step of bagging is only helpful if the classifiers are different. This only happens if small changes in the training data can result in large changes in the resulting classifier – that is, if the learning method is unstable [3].

2.2 Boosting

The idea of *boosting* is to iteratively train classifiers with a weak learner (the one with error better than 50% but worse than 0%) [4]. After each classifier is trained, its accuracy is measured, and misclassified instances are emphasized. Then the algorithm trains a new classifier on the modified dataset. At classification time, the boosting classifier combines the results from the individual classifiers it trained.

Boosting was originally proposed by Schapire and Freund [5,6]. In their *Adaptive Boosting*, or *AdaBoost*, algorithm, each of the training instances starts with a weight that tells the base classifier its relative importance [7]. At the initial step the weights of n instances are evenly distributed as $\frac{1}{n}$ The individual classifier training algorithm should take into account these weights, resulting in different classifiers after each round of reweighting and reclassification. Each classifier also receives a weight based on its accuracy; its output at classification time is multiplied by this weight.

Freund and Schapire proved that, if the base classifier used by AdaBoost has an error rate of just slightly less than 50%, the training error of the meta-classifier will approach zero exponentially fast [7]. For a two-class problem the base classifier only needs to be slightly better than chance to achieve this error rate. For problems with more than two classes less than 50% error is harder to achieve. Boosting appears to be vulnerable to overfitting. However, in tests it rarely overfits excessively [8].

2.3 Stacked generalization

In *stacked generalization*, or *stacking*, each individual classifier is called a *level-0 model*. Each may vote, or may have its output sent to a *level-1 model* – another classifier that tries to learn which level-0 models are most reliable. Level-1 models are usually more accurate than simple voting, provided they are given the class probability distributions from the level-0 models and not just the single predicted class [10].

3 Introduction to Formal Concept Analysis

3.1 Main definitions

A *formal context* in FCA is a triple $K = (G, M, I)$, where G is a set of objects, M is a set of attributes, and the binary relation $I \subseteq G \times M$ shows which object possesses which attribute. gIm denotes that object g has attribute m. For subsets of objects and attributes $A \subseteq G$ and $B \subseteq M$ *Galois operators* are defined as follows:

$$A' = \{m \in M \mid gIm \ \forall g \in A\},$$
$$B' = \{g \in G \mid gIm \ \forall m \in B\}.$$

A pair (A, B) such that $A \subseteq G, B \subseteq M, A' = B$ and $B' = A$, is called a *formal concept* of a context K. The sets A and B are closed and called the *extent* and the *intent* of a formal concept (A, B) respectively. For the set of objects A the set of their common attributes A' describes the similarity of objects of the set A and the closed set A'' is a cluster of similar objects (with the set of common attributes A') [11].

The number of formal concepts of a context $K = (G, M, I)$ can be quite large ($2^{min\{|G|,|M|\}}$ in the worst case), and the problem of computing this number is #P-complete [12]. There exist some ways to reduce the number of formal concepts, for instance, choosing concepts by stability, index or extent size [13].

For a context (G, M, I), a concept $X = (A, B)$ is *less general than or equal to* a concept $Y = (C, D)$ (or $X \leq Y$) if $A \subseteq C$ or, equivalently, $D \subseteq B$. For two concepts X and Y such that $X \leq Y$ and there is no concept Z with $Z \neq X, Z \neq Y, X \leq Z \leq Y$, the concept X is called a *lower neighbor* of Y, and Y is called an *upper neighbor* of X. This relationship is denoted by $X \prec Y$. Formal concepts, ordered by this relationship, form a *complete concept lattice* which might be represented by a *Hasse diagram* [14]. Several algorithms for building formal concepts (including *Close by One*) and constructing concept lattices are studied also in [14].

One can address to [11] and [15] to find some examples of formal contexts, concepts and lattices with their applications. Chapter 4 also shows the usage of FCA apparatus in a concrete task.

However, in some applications there is no need to find all formal concepts of a formal context or to build the whole concept lattice. Concept lattices, restricted to include only concepts with frequent intents, are called *iceberg lattices*. They were shown to serve as a condensed representation of association rules and frequent itemsets in data mining [15].

Here we modified the *Close by One* algorithm slightly in order to obtain only the upper-most concept of a formal context and its lower neighbors. The description of the algorithm and details of its modification is beyond the scope of this paper.

4 A toy example

Let us demonstrate the way RMCS works with a toy synthetic dataset shown in Table 1. We consider a binary classification problem with 8 objects comprising a training set and 2 objects in a test set. Each object has 4 binary attributes and a target attribute (class). Suppose we train 4 classifiers on this data and try to predict labels for objects 9 and 10.

Using FCA terms, we denote by $G = \{1, 2, 3, 4, 5, 6, 7, 8, 9, 10\}$ — the whole set of objects, $G_{test} = \{9, 10\}$ — the test set, $G_{train} = G \backslash G_{test}$ — the training set, $M = \{m_1, m_2, m_3, m_4\}$ — the attribute set, $C = \{cl_1, cl_2, cl_3, cl_4\}$ — the set of classifiers.

Table 1. A sample data set of 10 objects with 4 attributes and 1 binary target class

G/M	m_1	m_2	m_3	m_4	Label
1	×	×		×	1
2	×			×	1
3		×	×		0
4	×		×	×	1
5	×	×	×		1
6		×	×	×	0
7	×	×	×		1
8			×	×	0
9	×	×	×	×	?
10		×		×	?

Table 2. A classification context

G/C	cl_1	cl_2	cl_3	cl_4
1	×		×	×
2		×	×	
3	×			×
4		×	×	
5	×	×		
6	×	×		×
7		×		×
8		×	×	×

Here we run leave-one-out cross-validation on this training set for 4 classifiers. Further, we fill in Table 2, where a cross for object i and classifier cl_j means that cl_j correctly classifies object i in the process of cross-validation. To clarify, a cross for object 3 and classifier cl_4 means that after being trained on the whole training set but object 3 (i.e. on objects $\{1, 2, 4, 5, 6, 7, 8\}$), classifier cl_4 correctly predicted the label of object 3.

Let us consider Table 2 as a formal context with objects G and attributes C (so now classifiers play the role of attributes). We refer to it as classification context. The concept lattice for this context is presented in Fig. 1.

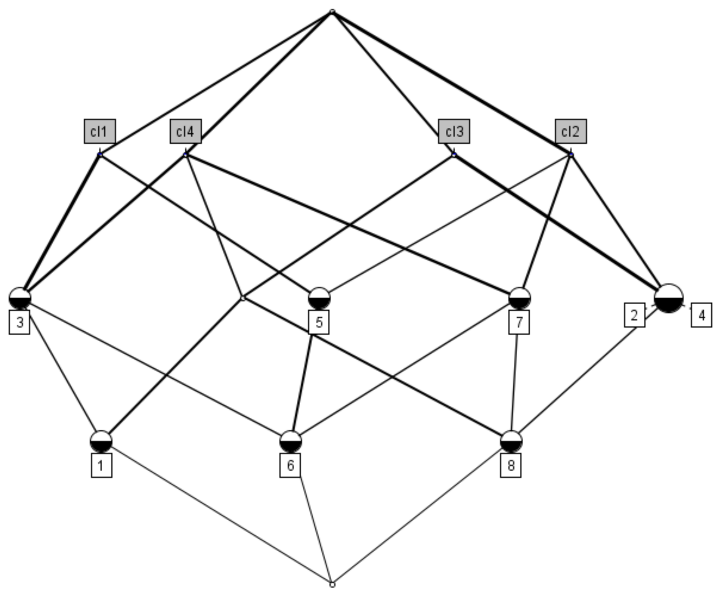


Fig. 1. The concept lattice of the classification context

As it was mentioned, the number of formal concepts of a context $K = (G, M, I)$ can be exponential in the worst case. But for the toy example it is possible to draw the whole lattice diagram. Thankfully, we do not need to build the whole lattice in RMCS algorithm — we only keep track of its top concepts.

Here are these top concepts: $(G, \emptyset)$, $(\{1, 3, 5, 6\}, \{cl_1\})$, $(\{2, 4, 5, 6, 7, 8\}, \{cl_2\})$, $(\{1, 2, 4, 8\}, \{cl_3\})$, $(\{1, 3, 6, 7, 8\}, \{cl_4\})$.

To classify objects from G_{test}, we first find their k nearest neighbors from G_{train} according to some distance metric. In this case, we use $k = 3$ and Hamming distance. In these conditions, we find that three nearest neighbors of object 9 are 4, 5 and 7, while those of object 10 are 1, 6 and 8.

Then, we take these sets of nearest neighbors $Neighb_9 = \{4, 5, 7\}$ and $Nieghb_{10} = \{1, 6, 8\}$, and find maximal intersections of these sets with the extents of formal concepts presented above (ignoring the concept $(G, \emptyset)$). The intents (i.e. classifiers) of the corresponding concepts are given as recommendations for the objects from G_{test}. The procedure is summarized in Table 3.

Table 3. Recommending classifiers for objects from G_{test}

G_{test}	1^{st} nearest neighbor	2^{nd} nearest neighbor	3^{rd} nearest neighbor	Neighbors	Classification concept which extent gives the maximal intersection with the Neighbors set	Recommended classifier
9	4	5	7	$\{4, 5, 7\}$	$(\{2, 4, 5, 6, 7, 8\}, \{cl_2\})$	cl_2
10	1	6	8	$\{1, 6, 8\}$	$(\{1, 3, 6, 7, 8\}, \{cl_4\})$	cl_4

Finally, the RMCS algorithm predicts the same labels for objects 9 and 10 as classifiers cl_2 and cl_4 do correspondingly.

Lastly, let us make the following remarks:

1. We would not have ignored the upper-most concept with extent G if it did not have an empty intent. That is, if we had the top concept of the classification context in a form $(G, \{cl_j\})$ it would mean that cl_j correctly classified all objects from the training set and we would therefore recommend it to the objects from the test set.
2. One more situation might occur that two or more classifiers turn out to be equally good at classifying objects from G_{train}. That would mean that the corresponding columns in classification table are identical and, therefore, the intent of some classification concept is comprised of more than one classifier. In such case, we do not have any argument for preferring one classifier to another and, hence, the final label would be defined as a result of voting procedure among the predicted labels of these classifiers.
3. Here we considered an input dataset with binary attributes and a binary target class. However, the idea of the RMCS algorithm is still applicable for datasets with numeric attributes and multi-class classification problems.

5 Recommender-based Multiple Classifier System

In this section, we discuss the Recommender-based Multiple Classifier System (RMCS). The pseudocode of the RMCS algorithm is presented in the listing Algorithm 1.

The inputs for the algorithm are the following:

1. $\{X_{train}, y_{train}\}$ — is a training set, X_{test} — is a test set;

2. $C = \{cl_1, cl_2, ..., cl_K\}$ — is a set of K base classifiers. The algorithm is intended to perform a classification accuracy exceeding those of base classifiers;
3. $dist(x_1, x_2)$ — is a distance function for objects which is defined in the attribute space. This might be the Minkowski (including Hamming and Euclidean) distance, the distance weighted by attribute importance and others.
4. k, n_fold — are parameters. Their meaning is explained below;
5. $topCbO(context)$ — is a function for building the upper-most concept of a formal context and its lower neighbors. Actually, it is not an input for the algorithm but RMCS uses it.

The algorithm includes the following steps:

1. Cross-validation on the training set. All K classifiers are trained on $n\_folds-1$ folds of X_{train}. Then a classification table (or context) is formed where a cross is put for object i and classifier cl_j if cl_j correctly classifies object i after training on $n\_folds-1$ folds (where object i belongs to the rest fold);
2. Running base classifiers. All K classifiers are trained on the whole X_{train}. Then, a table of predictions is formed where (i, j) position keeps the predicted label for object i from X_{test} by classifier cl_j;
3. Building top formal concepts of the classification context. The *topCbO* algorithm is run in order to build upper formal concepts of a classification context. These concepts have the largest possible number of objects in extents and minimal possible number of classifiers in their intents (not counting the upper-most concept);
4. Finding neighbors of the objects from X_{test}. The objects from the test set are processed one by one. For every object from X_{test} we find its k nearest neighbors from X_{train} according to the selected metric $sim(x_1, x_2)$. Let us say these k objects form a set $Neighbors$. Then, we search for a concept of a classification context which extent yields maximal intersection with the set $Neighbors$. If the intent of the upper-most concept is an empty set (i.e., no classifier correctly predicted the labels of all objects from X_{train}, which is mostly the case), then the upper-most concept $(G, \emptyset)$ is ignored. Thus, we select a classification concept, and its intent is a set of classifiers C_{sel};
5. Classification. If C_{sel} consists of just one classifier, we predict the same label for the current object from X_{test} as this classifier does. If there are several selected classifiers, then the predicted label is defined by majority rule.

6 Experiments

The algorithm, described above, was implemented in Python 2.7.3 and tested on a 2-processor machine (Core i3-370M, 2.4 HGz) with 3.87 GB RAM.

We used four UCI datasets in these experiments - `mushrooms`, `ionosphere`, `digits`, and `nursery`.[1] Each of the datasets was divided into training and test sets in proportion 70:30.

[1] `http://archive.ics.uci.edu/ml/datasets`

Algorithm 1 Recommender-based Multiple Classifier System

Input: $\{X_{train}, y_{train}\}, X_{test}$ — are training and test sets, $C = \{cl_1, cl_2, ..., cl_K\}$ — is a set of base classifiers, $topCbO(context, n)$ — is a function for building the uppermost concept of a formal context and its lower neighbors, $dist(x_1, x_2)$ — is a distance function defined in the attribute space, k — is a parameter (the number of neighbors), n_fold — is the number of folds for cross-validation on a training set
Output: y_{test} — are predicted labels for objects from X_{test}

$train\_class\_context = [\ ][\ ]$ — is a 2-D array
$test\_class\_context = [\ ][\ ]$ — is a 2-D array
for $i \in 0 \dots len(X_{train}) - 1$ **do**
  for $cl \in 0 \dots len(C) - 1$ **do**
    train classifier cl on $(n\_fold - 1)$ folds not including object $X_{train}[i]$
    $pred$ = predicted label for $X_{train}[i]$ by classifier cl
    $train\_class\_context[i][cl] = (pred == y_{train}[i])$
  end for
end for
for $cl \in 0 \dots len(C) - 1$ **do**
  train classifier cl on the whole X_{train}
  $pred$ = predicted labels for X_{test} by classifier cl
  $test\_class\_context[:][cl] = pred$
end for
$top\_concepts = topCbO(class\_context)$
for $i \in 0 \dots len(X_{test}) - 1$ **do**
  $Neighbors = k$ nearest neighbors of $X_{test}[i]$ from X_{train} according to $sim(x_1, x_2)$
  $concept = argmax(c.extent \cap Neighbors), c \in top\_concepts$
  $C_{sel} = concept.intent$
  $labels$ = predictions for $X_{test}[i]$ made by classifiers from C_{sel}
  $y_{test}[i] = argmax(count\_freq(labels))$
end for

Table 4. Classification accuracy of 6 algorithms on 4 UCI datasets: mushrooms (1), ionosphere (2), digits (3), and nursery (4)

Data	SVM, RBF kernel (C=1, γ=0.02)	Logit (C=10)	kNN (euclidean, k=3)	RMCS (k=3, n_folds=4)	Bagging SVM (C=1, γ=0.02) 50 estimators	AdaBoost on decision stumps, 50 iterations
1	0.998 t=0.24 sec.	0.996 t=0.17 sec.	0.989 t=1.2*10^{-2} sec.	0.997 t=29.45 sec.	0.998 t=3.35 sec.	0.998 t=44.86 sec.
2	0.906 t=5.7*10^{-3} sec.	0.868 t=10^{-2} sec.	0.858 t=8*10^{-4} sec.	0.933 t=3.63 sec.	0.896 t=0.24 sec.	0.934 t=22.78 sec.
3	0.917 t=0.25 sec.	0.87 t=0.6 sec.	0.857 t=1.1*10^{-2} sec.	0.947 t=34.7 sec.	0.92 t=4.12 sec.	0.889 t=120.34 sec.
4	0.914 t=3.23 sec.	0.766 t=0.3 sec.	0.893 t=3.1*10^{-2} sec.	0.927 t=220.6 sec.	0.913 t=38.52 sec.	0.903 t=1140 sec.

Data	SVM, RBF kernel ($C=10^3$, $\gamma=0.02$)	Logit ($C=10^3$)	kNN (minkowski, p=1, k=5)	RMCS (k=5, n_folds=10)	Bagging SVM ($C=10^3$, $\gamma=0.02$) 50 estimators	AdaBoost on decision stumps, 100 iterations
1	0.998 t=0.16 sec.	0.999 t=0.17 sec.	0.999 t=$1.2*10^{-2}$sec.	0.999 t=29.45 sec.	0.999 t=3.54 sec.	0.998 t=49.56 sec.
2	0.906 t=$4.3*10^{-3}$ sec.	0.868 t=10^{-2} sec.	0.887 t=$8*10^{-4}$ sec.	0.9 t=3.63 sec.	0.925 t=0.23 sec.	0.934 t=31.97 sec.
3	0.937 t=0.22 sec.	0.87 t=0.6 sec.	0.847 t=$1.1*10^{-2}$ sec.	0.951 t=34.7 sec.	0.927 t=4.67 sec.	0.921 t=131.6 sec.
4	0.969 t=2.4 sec.	0.794 t=0.3 sec.	0.945 t=$3*10^{-2}$ sec.	0.973 t=580.2 sec.	0.92 t=85.17 sec.	0.912 t=2484 sec.

We ran 3 classifiers implemented in `SCIKIT-LEARN` library [2](written in Python) which served as base classifiers for the RMCS algorithm as well. These were a Support Vector Machine with Gaussian kernel (`svm.SVC()` in `Scikit`), logistic regression (`sklearn.linear_model.LogisticRegression()`) and k Nearest Neighbors classifier (`sklearn.neighbors.classification.KNeighborsClassifier()`).

The classification accuracy of each classifier on each dataset is presented in Table 4 along with special settings of parameters. Moreover, for comparison, the results for `Scikit`'s implementation of bagging with SVM as a base classifier and AdaBoost on decision stumps [3] are presented.

As we can see, RMCS outperformed its base classifiers in all cases, while it turned out to be better than bagging only in case of multi-class classification problems (datasets `digits` and `nursery`).

7 Conclusion

In this paper, we described the underlying idea of multiple classifier systems, discussed bagging, boosting and stacking. Then, we proposed a multiple classifier system which turned out to outperform its base classifiers and two particular implementations of bagging and AdaBoost in two multi-class classification problems.

Our further work on the algorithm will continue in the following directions: exploring the impact of different distance metrics (such as the one based on attribute importance or information gain) on the algorithm's performance, experimenting with various types of base classifiers, investigating the conditions preferable for RMCS (in particular, when it outperforms bagging and boosting), improving execution time of the algorithm and analyzing RMCS's overfitting.

[2] http://scikit-learn.org
[3] https://github.com/pbharrin/machinelearninginaction/tree/master/Ch07

Acknowledgements. The authors would like to thank their colleague from Higher School of Economics, Sergei Kuznetsov, Jaume Baixeries and Konstantin Vorontsov for their inspirational discussions which directly or implicitly influenced this study.

References

1. Izenman, A. J.: Committee Machines. Modern Multivariate Statistical Techniques. pp. 505—550. Springer New York (2008).
2. Condorcet, M.-J.-A.-N.: Essay on the Application of Analysis to the Probability of Majority Decisions. (1785)
3. Breiman, L.: Bagging predictors. Machine Learning. 24(2), 123–140. (1996)
4. Schapire, R. E.: The Strength of Weak Learnability. Machine Learning. 5, 197—227 (1990)
5. Freund, Y: Boosting a Weak Learning Algorithm by Majority. Information and Computation. 121(2), 256—285 (1995)
6. Freund, Y., Schapire, R. E.: A Decision-Theoretic Generalization of On-Line Learning and an Application to Boosting. Journal of Computer and System Sciences. 55, 119—139 (1997)
7. Freund, Y., Schapire, R. E.: A Short Introduction to Boosting. (1999).
8. Dietterich, T. G.: Ensemble Methods in Machine Learning. Multiple Classifier Systems, LBCS-1857. pp. 1—15. Springer (2000).
9. Breiman, L.: Random Forests. Machine Learning. 45(1), 5—32 (2001)
10. Wolpert, D. H.: Stacked Generalization. Neural Networks, 5, 241—259 (1992)
11. Ganter, B., Wille, R.: Formal Concept Analysis: Mathematical Foundations. Springer-Verlag New York, Inc., Secaucus, NJ, USA (1997).
12. Kuznetsov, S. O.: On Computing the Size of a Lattice and Related Decision Problems. Order. 18(4), 313—321 (2001).
13. Kuznetsov, S. O.: On stability of a formal concept. Annals of Mathematics and Artificial Intelligence. 49(1-4), 101—115 (2007)
14. Kuznetsov, S. O., Obiedkov, S.: Comparing Performance of Algorithms for Generating Concept Lattices. Journal of Experimental and Theoretical Artificial Intelligence. 14, 189—216 (2002).
15. Stumme, G., Taouil, R., Bastide, Y., Pasquier, N., Lakhal, L.: Intelligent Structuring and Reducing of Association Rules with Formal Concept Analysis. In: Baader, F., Brewka, G., and Eiter, T. (eds.) KI 2001: Advances in Artificial Intelligence. pp. 335—350. Springer Berlin Heidelberg (2001)

Bicluster enumeration using Formal Concept Analysis

Víctor Codocedo and Amedeo Napoli

LORIA - CNRS - INRIA - Université de Lorraine, BP 239, 54506
Vandœuvre-les-Nancy.
victor.codocedo@loria.fr, amedeo.napoli@loria.fr,

Abstract. In this work we introduce a novel technique to enumerate constant row/column value biclusters using formal concept analysis. To achieve this, a numerical data-table (standard input for biclustering algorithms) is modelled as a many-valued context where rows represent objects and columns represent attributes. Using equivalence relations defined for each single column, we are able to translate the bicluster mining problem in terms of the partition pattern structure framework. We show how biclustering can benefit from the FCA framework through its robust theoretical description and efficient algorithms. Finally, we show how this technique is able to find high quality biclusters (in terms of the mean squared error) more efficiently than a state-of-the-art bicluster algorithm.

1 Introduction

Biclustering has become a fundamental tool for bioinformatics and gene expression analysis [4]. Different from standard clustering where objects are compared and grouped together based on their full descriptions, biclustering generates groups of objects based on a subset of their attributes, values or conditions. Thus biclusters are able to represent object relations in a local scale instead of the global representation given by an object cluster [12]. In this sense, biclustering has many elements in common with Formal Concept Analysis (FCA) [6]. In FCA objects are grouped together by the attributes they share in what is called a formal concept. Furthermore, formal concepts are arranged in a hierarchical and overlapping structure denominated a concept lattice. Hence a formal concept can be considered as a bicluster of objects and attributes representing relations in a local scale, while the lattice structure gives a description in the global scale. FCA is not only analogous to biclustering, but has much to offer in terms of mining techniques and algorithms [10]. The concept lattice can also provide biclusters with an overlapping hierarchy which has been reported as an important feature for bicluster analysis [15]. Recently, some approaches considering the use of FCA algorithms to mine biclusters from a numerical data-table have been introduced showing good potential [8, 7]. In this work, we present a novel technique for lattice-based biclustering using the pattern structure framework [5], an extension of FCA to deal with complex data. More specifically, we propose a technique for mining biclusters with similar row/column values, a specialization of biclustering focused on mining attributes with coherent variations, i.e. the difference between two attributes is the same for a group of objects [12]. We show that, by the use of partition pattern structures [1], we can find high quality maximal biclusters (w.r.t. the mean squared error). Finally, we compare our approach with a standard constant row value algorithm [3], showing the capabilities and limitations of our approach.

The remainder of this paper is organized as follows. The basics of biclustering are introduced in Section 2. Section 3 presents our approach and Section 4 presents the experiments and initial findings of our biclustering technique. Finally, Section 5 concludes our article and presents some new perspectives of research.

2 Biclustering definitions

A numerical data-table is a matrix $\mathcal{M}$ where $\mathcal{M}_{ij}$ indicates the value of an object $\mathtt{g}_i \in \mathtt{G}$ w.r.t. the attribute $\mathtt{m}_j \in \mathtt{M}$ with $i \in [1..|\mathtt{G}|]$ and $j \in [1..|\mathtt{M}|]$ ($|\cdot|$ represents set cardinality). A bicluster of $\mathcal{M}$ is a submatrix $\mathcal{B}$ where each value $\mathcal{B}_{ij}$ satisfies a given restriction. According to [4, 12], there are five different restrictions which we summarize in Table 1.

Constant values	$\mathcal{B}_{ij} = c$	Within the submatrix, all values are equal to a constant $c \in \mathbb{R}$ ($\mathbb{R}$ indicates real values).
Constant row values	$\mathcal{B}_{ij}=c + \alpha_i$	Within the submatrix, all the values in a given row i are equal to a constant c and a row adjustment $\alpha_i \in \mathbb{R}$.
Constant column values	$\mathcal{B}_{ij}=c + \alpha_j$	Within the submatrix, all the values in a given column j are equal to a constant c and a column adjustment $\alpha_j \in \mathbb{R}$.
Coherent values	$\mathcal{B}_{ij}= c + \alpha_i + \beta_j$	Within the submatrix, all the values in a given column j are equal to a constant c, a row adjustment α_i and a column adjustment β_j. Instead of addition, the model can also consider multiplicative factors.
Coherent evolution		Values in the submatrix induce a linear order.

Table 1: Types of biclusters.

Similar values instead of constant values When noise is present in a data-table, it is difficult to search for constant values. Several approaches have tackled this issue in different ways, e.g. by the use of evaluation functions [14], equivalence relations [2, 13] and tolerance relations [7]. The most common way is establishing a threshold $\theta \in \mathbb{R}$ to enable the similarity comparison of two different values $w_1, w_2 \in \mathbb{R}$. We say that $w_1 \simeq_\theta w_2$ (values are similar) iff $|w_1 - w_2| \leq \theta$. Thus, constant values are a special case of similar values when $\theta = 0$. Using this, we can redefine the first three types of biclusters as follows:

1. Similar values: $\mathcal{B}_{ij} \simeq_\theta \mathcal{B}_{kl}$.

2. Similar row/column values:
   (a) Similar row values: $\mathcal{B}_{ij} \simeq_\theta \mathcal{B}_{il}$.
   (b) Similar column values: $\mathcal{B}_{ij} \simeq_\theta \mathcal{B}_{kj}$.

Example 1. *With $\theta = 1$, Table 2 shows in its upper left corner a bicluster with similar values (dark grey). The upper right corner represents a similar column bicluster (light grey). Lower left corner considering $\{\mathtt{g}_3, \mathtt{g}_4\}$ and $\{\mathtt{m}_1, \mathtt{m}_2\}$ (not marked in the table) represents a similar row bicluster.*

	$\mathtt{m}_1$	$\mathtt{m}_2$	$\mathtt{m}_3$	$\mathtt{m}_4$	$\mathtt{m}_5$
$\mathtt{g}_1$	1	2	2	1	6
$\mathtt{g}_2$	2	1	1	0	6
$\mathtt{g}_3$	2	2	1	7	6
$\mathtt{g}_4$	8	9	2	6	7

Table 2: Bicluster with similar values ($\theta = 1$).

	$\mathtt{m}_1$	$\mathtt{m}_3$
$\mathtt{g}_2$	2	1
$\mathtt{g}_3$	2	1

Table 3: Constant column bicluster.

3 Biclustering using partition pattern structures

The pattern structure framework is an extension of FCA proposed to deal with complex data [5]. Partition pattern structures are an instance of the pattern structure framework proposed to mine functional dependencies among attributes of a database [1] dealing with set partitions. In the following, we provide the specifics of partition pattern structures where the main definitions are given in [5].

Let $\mathtt{G}$ be a set of objects, $\mathtt{M}$ a set of attributes and $\mathcal{M}$ a data-table of numerical values where $\mathcal{M}_{ij}$ contains the value of attribute (column) $\mathtt{m}_j \in \mathtt{M}$ in object (row) $\mathtt{g}_i \in \mathtt{G}$. A partition $\mathtt{d} = \{p_i\}$ of the set $\mathtt{G}$ can be formalized as a collection of components p_i such as:

$$\bigcup_{p_i \in \mathtt{d}} p_i = \mathtt{G} \qquad p_i \cap p_j = \emptyset \; ; (p_i, p_j \in \mathtt{d}, i \neq j)$$

Two partitions can be ordered by the *coarser-finer* relation where we say that a partition $\mathtt{d}_1 = \{p_i\}$ is a refinement of $\mathtt{d}_2 = \{p_j\}$ (or $\mathtt{d}_2$ is a coarsening of $\mathtt{d}_1$) iff $\forall\, p_i \in \mathtt{d}_1, \exists\, p_j \in \mathtt{d}_2,\; p_i \subseteq p_j$. We denote this as $\mathtt{d}_1 \sqsubseteq \mathtt{d}_2$ where $\mathtt{d}_1, \mathtt{d}_2 \in \mathtt{D}$ is the space of all partitions of the set $\mathtt{G}$.

Let us define the mapping function $\delta : \mathtt{M} \to \mathtt{D}$, which assigns to each attribute in $\mathtt{M}$ the partition it generates over the set of objects $\mathtt{G}$, as follows:

$$\delta(\mathtt{m}_j) = \{[\mathtt{g}_i]_{m_j} \mid \mathtt{g}_i \in \mathtt{G}\} \tag{1}$$

$$[\mathtt{g}_i]_{\mathtt{m}_j} = \{\mathtt{g}_k \in \mathtt{G} \mid \mathcal{M}_{ij} = \mathcal{M}_{kj}\} \tag{2}$$

Where $[\mathtt{g}_i]_{\mathtt{m}_j}$ is the equivalence class of $\mathtt{g}_i$ w.r.t. attribute $\mathtt{m}_j$, i.e. the set of rows in data-table $\mathcal{M}$ which have the same value in column $\mathtt{m}_j$ as row $\mathtt{g}_i$. Since the set of equivalence classes for a given attribute generates a partition over $\mathtt{G}$, it comes naturally that $\delta(\mathtt{m}_j) \in \mathtt{D}$ for any $\mathtt{m}_j \in \mathtt{M}$.

It is easy to show that the order in the space of object partitions $\mathtt{D}$ defines a complete lattice for which the similarity operator $\sqcap$ for any two partitions $\mathtt{d}_1, \mathtt{d}_2 \in \mathtt{D}$ is defined as follows:

$$\mathtt{d}_1 \sqcap \mathtt{d}_2 = \bigcup p_i \cap p_j \tag{3}$$

$$\mathtt{d}_1 \sqsubseteq \mathtt{d}_2 \iff \mathtt{d}_1 \sqcap \mathtt{d}_2 = \mathtt{d}_1 \tag{4}$$

Then, a *partition pattern structure* is determined by the triple $(\mathtt{M}, (\mathtt{D}, \sqcap), \delta)$ in which the following derivation operators for $\mathtt{B} \subseteq \mathtt{M}$ and $\mathtt{d} \in \mathtt{D}$ are defined:

$$\mathtt{B}^{\square} = \bigsqcap_{\mathtt{m} \in \mathtt{B}} \delta(\mathtt{m}) \tag{5}$$

$$\mathtt{d}^{\square} = \{\mathtt{m} \in \mathtt{M} \mid \mathtt{d} \sqsubseteq \delta(\mathtt{m})\} \tag{6}$$

Similarly to standard FCA, we have that $(\mathtt{B}, \mathtt{d})$ is a partition pattern concept (pp-concept) when $\mathtt{B}^{\square} = \mathtt{d}$ and $\mathtt{d}^{\square} = \mathtt{B}$ and that for two pp-concepts $(\mathtt{B}_1, \mathtt{d}_1)$ and $(\mathtt{B}_2, \mathtt{d}_2)$, the order between them is given by $(\mathtt{B}_1, \mathtt{d}_1) \leq (\mathtt{B}_2, \mathtt{d}_2) \iff (\mathtt{B}_1 \subseteq \mathtt{B}_2)$ or $(\mathtt{d}_2 \sqsubseteq \mathtt{d}_1)$. Pp-concepts determines biclusters as pairs $(p, \mathtt{B})$ where p is a component of the partition pattern $\mathtt{d}$. It should be noticed that to keep consistency with previous notation, we write biclusters as pairs $(p, \mathtt{B})$ (p represent rows and $\mathtt{B}$ represent columns), while pp-concepts are written inversely $(\mathtt{B}, \mathtt{d})$ ($\mathtt{B}$ is the extent and $\mathtt{d}$ is the intent of $(\mathtt{B}, \mathtt{d})$).

Proposition 1. *Let $(\mathtt{B}, \mathtt{d})$ be a pp-concept, then for any partition component $p \in \mathtt{d}$ each pair $(p, \mathtt{B})$ corresponds to a constant column value bicluster.*

The proof of this proposition is straightforward considering that each pair $(p, \mathtt{B})$ represents a submatrix the columns of which were selected using an equivalence relation, i.e. the values in the columns are the same.

We say that a bicluster $(p, \mathtt{B})$ is maximal iff adding an object to p or an attribute to $\mathtt{B}$ does not result in a bicluster, i.e. $(p \cup \{\mathtt{g}\}, \mathtt{B})$ and $(p, \mathtt{B} \cup \{\mathtt{m}\})$ are not biclusters. While pp-concepts are maximal (closed under $(\cdot)^{\square}$), biclusters corresponding to pairs $(p, \mathtt{B})$ are not always maximal. This is due to the fact that pp-concepts are maximal w.r.t. the partitions and not w.r.t. the individual components of those partitions. Nevertheless, maximal biclusters are still easy to identify.

Proposition 2. *Let* $(\mathtt{B}_1, \mathtt{d}_1), (\mathtt{B}_2, \mathtt{d}_2)$ *be two pp-concepts such as* $(\mathtt{B}_1, \mathtt{d}_1) \leq (\mathtt{B}_2, \mathtt{d}_2)$*. Let* $p \subseteq \mathtt{G}$ *be a component of a partition. If* $p \in \mathtt{d}_1$ *and* $p \notin \mathtt{d}_2$ *then the bicluster corresponding to* $(p, \mathtt{B}_1)$ *is maximal.*

Proof. Given definitions in Equations 2, 5 and 6, we have that for $(\mathtt{B}_1, \mathtt{d}_1)$ and for any $\mathtt{g}_i \in p$, the following is true:

$$p = \bigcap_{\mathtt{m}_j \in \mathtt{B}_1} \{\mathtt{g}_k \in \mathtt{G} \mid \mathcal{M}_{ij} = \mathcal{M}_{kj}\} \tag{7}$$

Consequently, for any other object $\mathtt{g}_h \in \mathtt{G}$, such as $\mathtt{g}_h \notin p$, we have $\mathcal{M}_{ij} \neq \mathcal{M}_{hj}$. Hence, the pair $(p + \{\mathtt{g}_h\}, \mathtt{B})$ cannot be a bicluster.

Let $\mathtt{B}_2 = \mathtt{B}_1 + \{\mathtt{m}_j\}$ for any $\mathtt{m}_j \in \mathtt{M}$, we show that $(p, \mathtt{B}_2)$ cannot be a cluster by contradiction. Let $(p, \mathtt{B}_2)$ be a bicluster. Then, there exists the pp-concept $(\mathtt{B}_2, \mathtt{B}_2^{\square})$ such as $p \in \mathtt{B}_2^{\square}$. If it does, then it is necessarily a direct super concept of $(\mathtt{B}_1, \mathtt{d}_1)$. However, this contradicts the definition $p \notin \mathtt{B}_2^{\square}$. $\square$

Supporting similar values: In general, it is not possible to support similar value biclusters as described in Section 2 using the partition pattern structures framework. This is due to the fact that the restriction $\mathcal{B}_{ij} \simeq_\theta \mathcal{B}_{kl} \iff |\mathcal{B}_{ij} - \mathcal{B}_{kl}| \leq \theta$ is not transitive and hence, it is not an equivalence but a tolerance relation [10] which do not necessarily generates partitions over the set of objects. However, the setting to support this scenario is only slightly different from the partition pattern structures framework. We do not provide its description for the sake of simplicity.

Nevertheless, through the use of interval of values we can get a close representation of similar value biclusters considering that two rows (objects) are in the same equivalence class if their values in a given column (attribute) is within a given interval (rather than being equal as described in Equation 2). For example, consider in Table 2 the intervals $[0, 1]$ and $[6, 7]$ for attribute m_4. We can see that it generates the partition $\{\mathtt{g}_1, \mathtt{g}_2\}, \{\mathtt{g}_3, \mathtt{g}_4\}$. We call these intervals "equivalence blocks", similarly as the "tolerance blocks" described in [10]. Equivalence blocks can be either pre-defined, allowing the user to include some background knowledge in the biclustering process, or calculated on-the-fly if a number of equivalence blocks γ is specified.

4 Experiments

4.1 Partition pattern concept lattice calculation

In order to calculate the partition pattern concept lattice for a given data-table we used the AddIntent algorithm as described in [16]. We applied AddIntent over a subset of the dataset called MovieLens 100k[1] of movie ratings containing 943 users and 50 movies (out of a total of 1682) using the predefined set of equivalence blocks $[1, 2][3, 3][4, 5]$. The dataset contains user ratings for movies

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

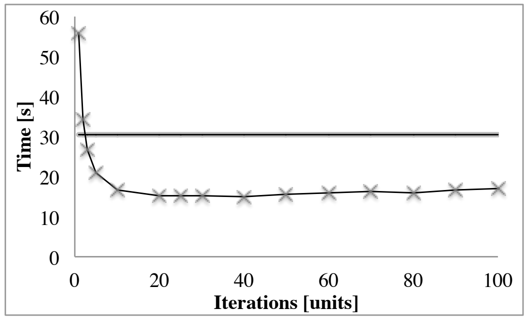


Fig. 1: AddIntent Iterations per prune vs Execution time

which range from 1 to 5. When information is not available, the matrix contains 0 which we disregard (we do not mine biclusters with columns equal to 0). The dataset contained 16532 similar column biclusters.

Empirical results showed that less than 20% of the pp-concepts within the pp-lattice actually hold a maximal bicluster. In order to improve the efficiency of AddIntent for biclustering purposes we have included a pruning step between a certain number of AddIntent iterations (each time a new intent is added to the lattice). The pruning step consists of removing from the lattice any concept that do not hold a maximal bicluster. Figure 1 shows experimental results in this regard. The graphic shows the execution time (y axis) taken by AddIntent to calculate the 16532 biclusters when a pruning step was included in a given number of iterations (x axis). The solid horizontal line represents the execution time without pruning (30.5 seconds). While initially, the execution time doubles the non-optimized version (for a lattice prune each AddIntent iteration), later the time quickly stabilizes around half the time the non-optimized version. Best time is found for 40 iterations (15 seconds).

The pruning affects the number of intent intersections performed by AddIntent. When the lattice is pruned, there are not as many intents to intersect as there were originally. However, pruning the lattice is an expensive task and adds overhead to the algorithm. The correct balance of this trade-off leads to dramatic improvements in the performance (twice in the experiments), however further experimentation in different numerical data-tables are needed to draw more conclusions regarding its setting.

4.2 Biclusters quality

A second experiment was performed over an example dataset provided with the system BicAt[2] containing 419 objects and 70 attributes. We measure the performance of our approach mining similar row biclusters compared with Cheng and Church's algorithm (CC) [3]. CC tries to find a determined number of biclusters

[2] http://www.tik.ee.ethz.ch/sop/bicat/

with a maximum threshold for the mean squared error δ. Results are shown in Table 4. Parameters for pp-lattice are number of equivalence blocks γ and minimal number of columns in the cluster σ. CC was executed as provided by BicAt and other parameters were left as system's default.

	Time [s]	Biclusters [Kunits]	Parameters	MSE Max	Max Size [cells]
PPL	451	901	γ=20, σ=10	0.016	209
PPL	27	36	γ=10, σ=30	0.032	372
PPL	306	390	γ=10, σ=25	0.037	442
PPL	3,404	4,471	γ=10, σ=20	0.041	462
PPL	253	314	γ=5, σ=50	0.259	1,173
CC	418	1	$\delta = 0.5$	3.2	17,752
CC	416	1	$\delta = 0.3$	2.81	17,752
CC	4,018	10	$\delta = 0.1$	4.92	17,752

Table 4: Comparison between CC and pp-lattice bicluster algorithm.

Results show a general better performance of our approach which is able to mine more than four million maximal biclusters from the dataset in less time than CC calculates only ten thousands. In terms of minimal squared error (MSE), our approach gets smaller scores which induces better quality biclusters. CC is able to find larger biclusters compared to our approach given the top-down strategy which implements. While larger biclusters can be found with our approach by decreasing the number of equivalent classes (γ), this is done at the cost of increasing the MSE as shown in Table 4. Compared to CC, our approach is better on finding many high quality and rather small biclusters inducing specialized associations among objects. CC is better at creating a global map of the entire data-table by finding larger biclusters.

5 Conclusions and research perspectives

In this work we have presented a novel technique for exhaustive similar row/column value biclustering based on FCA algorithms using partition pattern structures. We have shown the capabilities of the technique which is able to find a large number of high quality biclusters. Furthermore, biclusters are provided with an overlapping hierarchy based on a concept lattice structure. How to leverage current biclusters analysis techniques using the concept lattice is still a matter of research.

Partition pattern structures were initially proposed for functional dependencies mining [1] using association rules from pp-concepts. How these techniques may benefit from the current approach and the opposite, is an interesting subject which should be explored. Using other techniques of formal concept selection and filtering, and their associations with biclusters is another compelling aspect for a future work.

References

1. Jaume Baixeries, Mehdi Kaytoue, and Amedeo Napoli, 'Characterizing functional dependencies in formal concept analysis with pattern structures', *Annals of Mathematics and Artificial Intelligence*, (2014).
2. Jérémy Besson, Céline Robardet, Luc Raedt, and Jean-François Boulicaut, 'Mining bi-sets in numerical data', in *Knowledge Discovery in Inductive Databases*, (2007).
3. Yizong Cheng and George M. Church, 'Biclustering of expression data', in *Proceedings of the Eighth International Conference on Intelligent Systems for Molecular Biology*, (2000).
4. Adelaide Valente Freitas, Wassim Ayadi, Mourad Elloumi, Joséluis Oliveira, Joséluis Oliveira, and Jin-Kao Hao, *Survey on Biclustering of Gene Expression Data*, 2013.
5. Bernhard Ganter and Sergei O. Kuznetsov, 'Pattern Structures and their projections', *Conceptual Structures: Broadening the Base*, (2001).
6. Bernhard Ganter and Rudolf Wille, *Formal Concept Analysis*, mathematic edn., 1999.
7. Mehdi Kaytoue, Sergei O. Kuznetsov, Juraj Macko, and Amedeo Napoli, 'Biclustering meets triadic concept analysis', *Annals of Mathematics and Artificial Intelligence*, (2013).
8. Mehdi Kaytoue, Sergei O. Kuznetsov, and Amedeo Napoli, 'Biclustering numerical data in formal concept analysis', in *Formal Concept Analysis*, (2011).
9. Mehdi Kaytoue, Sergei O. Kuznetsov, and Amedeo Napoli, 'Revisiting numerical pattern mining with formal concept analysis', *Proceedings of the Twenty-Second international joint conference on Artificial Intelligence*, (November 2011).
10. Sergei O. Kuznetsov, 'Galois connections in data analysis: Contributions from the soviet era and modern russian research', in *Formal Concept Analysis*, volume 3626 of *Lecture Notes in Computer Science*, (2005).
11. Sergei O Kuznetsov and Sergei Obiedkov, 'Comparing Performance of Algorithms for Generating Concept Lattices', *Journal of Experimental and Theoretical Artificial Intelligence*, (2002).
12. Sara C. Madeira and Arlindo L. Oliveira, 'Biclustering algorithms for biological data analysis: A survey', *IEEE/ACM Trans. Comput. Biol. Bioinformatics*, (January 2004).
13. Sadaaki Miyamoto, 'Lattice-valued hierarchical clustering for analyzing information systems', in *Rough Sets and Current Trends in Computing*, volume 4259 of *Lecture Notes in Computer Science*, (2006).
14. Gaurav Pandey, Gowtham Atluri, Michael Steinbach, Chad L. Myers, and Vipin Kumar, 'An association analysis approach to biclustering', in *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, (2009).
15. Gianvito Pio, Michelangelo Ceci, Corrado Loglisci, Domenica D'Elia, and Donato Malerba, 'A novel biclustering algorithm for the discovery of meaningful biological correlations between mirnas and mrnas', *EMBnet.journal*, **18**(A), (2012).
16. Dean van der Merwe, Sergei Obiedkov, and Derrick Kourie, 'Addintent: A new incremental algorithm for constructing concept lattices', in *Concept Lattices*, (2004).

Towards an FCA-based Recommender System for Black-Box Optimization

Josefine Asmus[1,3], Daniel Borchmann[2], Ivo F. Sbalzarini[1,3], and Dirk Walther[2,3]

[1] MOSAIC Group, Center for Systems Biology Dresden (CSBD), Max Planck Institute of Molecular Cell Biology and Genetics, Dresden, Germany.
[2] TU Dresden, Theoretical Computer Science, Dresden, Germany.
[3] Center for Advancing Electronics Dresden (cfAED), Dresden, Germany.
{asmus,ivos}@mpi-cbg.de
{Daniel.Borchmann,Dirk.Walther}@tu-dresden.de

Abstract. Black-box optimization problems are of practical importance throughout science and engineering. Hundreds of algorithms and heuristics have been developed to solve them. However, none of them outperforms any other on all problems. The success of a particular heuristic is always relative to a class of problems. So far, these problem classes are elusive and it is not known what algorithm to use on a given problem. Here we describe the use of Formal Concept Analysis (FCA) to extract implications about problem classes and algorithm performance from databases of empirical benchmarks. We explain the idea in a small example and show that FCA produces meaningful implications. We further outline the use of attribute exploration to identify problem features that predict algorithm performance.

1 Introduction

Optimization problems are ubiquitous in science and engineering, including optimizing the parameters of a model [11], finding an optimal statistical estimator, or finding the best operating conditions for an electronic circuit or a biochemical network. An optimization problem is defined by a parameter space spanned by the variables to be optimized, and an objective function defined over that space. The goal is to find points in the parameter space where the objective function has an extremum. In a black-box problem, the function is not known, but it can be evaluated point-wise. The objective function is hence given as an oracle, which, given a point in parameter space as an input, returns the objective function value at that point. Most practical applications in science and engineering are black-box problems, where the objective function may comprise running a numerical computer simulation or performing a laboratory measurement in order to, e.g., determine how well an electronic circuit works.

Partially funded by the German Research Foundation (DFG) via the Cluster of Excellence 'Center for Advancing Electronics Dresden (cfAED)'. Supported by DFG Research Training Group 1763 (QuantLA).

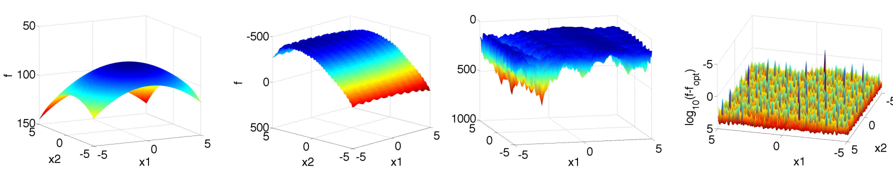
Owing to their practical importance, hundreds of algorithms and heuristics have been developed to (approximately) solve black-box optimization problems. The "No Free Lunch Theorem" [19] states that no algorithm performs better (i.e., finds the extremum in less iterations or finds a better extremum) than any other algorithm on all problems. Algorithm performance is hence relative to a problem or a class of problems. It is, however, unknown what these problem classes are, and how one should choose a "good" algorithm given a certain problem instance.

We aim at developing a recommender system for proposing efficient algorithms for a given black-box optimization problem. Formal Concept Analysis (FCA) [6] has previously been shown useful in recommender systems for rating systems and advertisement [4,8]. Here we present the idea of using attribute exploration in FCA to find discriminative attributes of empirical benchmark data of different algorithms tested on different problems. We show that FCA produces meaningful implications about problem features and algorithms that imply good performance. Some of the implications found confirm hypotheses that are commonly known in the optimization community, others are novel. We also outline the use of attribute exploration to discover sets of problem features that are particularly predictive for algorithm performance. This is work in progress and we are not presenting a final solution. We do, however, believe that FCA will provide a powerful tool on the way towards a generic recommender system and problem classification for black-box optimization.

2 Black-box Optimization Problems

We consider real-valued scalar black-box problems, modeled as an oracle function $f : \mathbb{R}^d \to \mathbb{R}$ mapping a d-tuple $\boldsymbol{x}$ of real numbers (problem parameters) to a scalar real number $f(\boldsymbol{x})$ (objective function value). The function f is not assumed to be explicitly known. A *black-box optimization problem* entails finding extremal points of f using only point-wise function evaluations. The iterative sequence of evaluations used to progress toward an extremum is called the *search strategy*.

The Black-Box Optimization Benchmark (BBOB) test suit is a standard collection of test functions used to empirically benchmark different optimization heuristics [7]. The suit contains 24 benchmark functions with known ground truth for parameter spaces of different dimensions. Figure 1 plots four example functions in two dimensions.



(a) Sphere (b) Rastrigin sep. (c) Weierstrass (d) Katsuura

Fig. 1. Four black-box optimization benchmark (BBOB) functions in two dimensions.

A *problem instance* is a triple (f, d, ϵ), where f is an instance of a black-box function, $d > 0$ is the parameter space dimension, and $\epsilon \in \mathbb{R}_{\geq 0}$ a tolerance.

Solving a problem instance (f, d, ϵ) means finding a d-tuple $\boldsymbol{x} \in \mathbb{R}^d$ such that $|f_{\min} - f(\boldsymbol{x})| < \epsilon$, where $f_{\min}$ is the global minimum of f. Note that the ground-truth global minimum $f_{\min}$ is known for the benchmark functions, but not in a practical problem instance.

3 An FCA-based Recommender System

We propose a system for recommending suitable algorithms for a given black-box optimization problem based on FCA. Similar recommender systems, albeit not based on FCA, have been suggested [3, 9, 15], and FCA has been applied to collaborative recommender systems for high-dimensional ratings [4] and in contextual advertising [8]. To the best of our knowledge, the use of FCA in a recommender system for black-box optimization has not been attempted so far.

3.1 Performance Measures

A run of an algorithm solving a problem instance is called *successful*. The performance of an algorithm a on a problem instance p is measured by the *expected running time* (ERT) [1,3]:

$$\mathsf{ERT}(a, p) = N^+(a, p) + \frac{1 - \pi^+(a, p)}{\pi^+(a, p)} N^-(a, p),$$

where $N^+(a, p)$ and $N^-(a, p)$ are the average numbers of function evaluations for successful and unsuccessful runs of a on p, respectively, and $\pi^+(a, p)$ is the ratio of the number of successful runs over the number of all runs of a on p.

The problem instances (f_j, d, ϵ), where $j \in \{1, ..., 24\}$, $d \in \{2, 3, 5, 10, 20, 40\}$, and $\epsilon \in E = \{10^3,\ 10^1,\ 10^{-1},\ 10^{-3},\ 10^{-5},\ 10^{-8}\}$, have been used to test over 112 algorithms; the dataset is available online.[1] Note that the algorithms treat the benchmark functions as black boxes, i.e., they only evaluate the functions, whereas the actual function remains hidden to the algorithm. The known global minimum is only used to evaluate the performance of the algorithms.

Often, algorithms cannot solve problem instances for any tolerance value within a certain number of function evaluations; see Table 1. To obtain a performance measure also in these cases, we introduce a penalty of 10^6 for every tolerance level that the algorithm could not solve. This *penalized ERT* (pERT) of an algorithm a on an instance f of a benchmark function in d dimensions for a set E of tolerances is here defined as:

$$p\mathsf{ERT}(a, f, d, E) = \frac{\mathsf{ERT}(a, (f, d, \epsilon_{\text{best}}))}{d} + 10^6 |\{\epsilon \in E \mid \mathsf{ERT}(a, (f, d, \epsilon)) = \infty\}|,$$

where ϵ_{best} is the smallest value in E such that $\mathsf{ERT}(a, (f, d, \epsilon)) \neq \infty$.

3.2 Benchmark Data Used

Table 1 shows the performance values of four algorithms on the benchmark functions *Sphere* and *Rastrigin separable* in 10 dimensions. The algorithm *(1+1)-CMA-ES* implements an evolutionary strategy with covariance matrix adaptation [2]. We also include two variants of Particle Swarm Optimization (PSO):

[1] `http://coco.gforge.inria.fr`

Tolerance	10^3	10^1	10^{-1}	10^{-3}	10^{-5}	10^{-8}	pERT
	ERT						
Sphere							
(1+1)-CMA-ES	1	90	267	440	611	872	87.2
PSO	1	221	2 319	4 911	7 659	11 865	1 186.5
PSO-BFGS	1	68	74	74	74	75	7.5
Simplex	1	1 596	3 450	7 283	18 418	80 428	8 042.8
Rastrigin separable							
(1+1)-CMA-ES	4	1 456 464	∞	∞	∞	∞	4 145 646.4
PSO	4	259 992	6 532 067	6 533 720	6 537 337	6 542 104	654 210.4
PSO-BFGS	2	120 255	∞	∞	∞	∞	4 012 025.5
Simplex	1	∞	∞	∞	∞	∞	5 000 000.1

Table 1. ERT and pERT performance values of four algorithms on the functions Sphere and Rastrigin Separable in 10 dimensions.

standard *PSO* [5] and *PSO-BFGS*, a hybrid of PSO and the Broyden-Fletcher-Goldfarb-Shanno (BFGS) algorithm [18]. The fourth algorithm is the classical *Simplex* algorithm [16, 17]. Each algorithm can solve a problem instance for the Sphere function to any value of tolerance, whereas on the separable Rastrigin function this is the case only for PSO, but with higher ERT. The last column shows the pERT values that account for any failed tolerance level with a penalty.

Functions can be characterized by features. Several high-level features have been hypothesized to predict black-box optimization algorithm performance, such as multi-modality, global structure, and separability [10]. Table 2 lists the extent to which different BBOB test functions exhibit these features. *Multi-*

Function	multi-modality	global structure	separability
Sphere	none	none	high
Rastrigin separable	high	strong	high
Weierstrass	high	medium	none
Katsuura	high	none	none

Table 2. Three high-level features of four BBOB benchmark functions.

modality refers to the number of local minima of a function.

Global structure refers to the structure that emerges when only considering minima (in a minimization problem), i.e., the point set left after deleting all non-optimal points. *Separability* specifies whether a function is a Cartesian product of one-dimensional functions. For instance, the Sphere function (Fig. 1(a)) is unimodal and, therefore, has no global structure (only one point remains when restricting to minima) and is highly separable. The separable Rastrigin function (cf. Fig. 1(b)) is separable, highly multimodal, and has a strong global structure (the set of local minima forms a convex funnel).

3.3 Results

Different numerical measures have been suggested to estimate the features of a black-box function from sample evaluations. These include fitness distance correlations [13] and ICoFis, a method based on information content [14]. Furthermore, in [9] it is shown that in several cases it is possible to predict the *high-level* features of Table 2 from *low-level* features that can be computed from samples only. Based on these observations, we develop an FCA-based recommender system that suggests well-performing algorithms for a given black-box optimization problem by extracting implications from the BBOB database. The recommender system consists of three steps; cf. Algorithm 3.1. In Step 1, features of the black-box function that is given as an input are estimated by drawing samples from the function. In Step 2, these features are used to determine the features of algorithms that perform well on such functions. In Step 3, a specific algorithm is chosen that has these algorithm features. Determining good algorithm features

Algorithm 3.1 Recommender System

Input: unknown black-box function from which samples can be drawn
Output: algorithm which is suitable for optimizing input function
 Step 1: estimate features of input function
 Step 2: imply features of algorithms that do well on functions with such features
 Step 3: pick an algorithm that has these features

(Step 2) is done using *implications* that form the background knowledge of the recommender system. The premise of an implication contains algorithm features together with features of benchmark functions, whereas the conclusion consists of attributes corresponding to the performances of the algorithms when solving the functions. The implications are obtained from the BBOB database by FCA. More precisely, we build a formal context $\mathbb{K} = (G, M, I)$, where G is the set of *objects* consisting of all pairs of benchmark functions and algorithms, M is the set of *attributes* consisting of the features of algorithms and benchmark functions together with the attributes assigned to performance values, and I is the *incidence matrix* between the objects in G and the attributes in M. We obtain the implications from the context $\mathbb{K}$ by computing the canonical base (or another implicational base) of $\mathbb{K}$.

We illustrate our approach in Table 3. The objects are pairs of benchmark functions and algorithms, as introduced in Sec. 3.2. The attributes are features of algorithms, performance attributes P_i, for $i \in \{0, 5, 6, 7\}$,[2] and the function features (Table 2). As algorithm features we use the respective names of the algorithms together with the features *deterministic* and *hybrid* (stating that the algorithm is a combination of different search strategy). We consider Table 3 as a formal context and compute its canonical base to obtain the implications:

[2] To obtain a binary table, we partition the range of pERT-values into 10 intervals and introduce attribute names P_i for the intervals.

$$\{\ \mathsf{(1{+}1)\text{-}CMA\text{-}ES}, \mathsf{multimod\_high}\ \} \rightarrow \{\ \mathsf{P}_6\ \}$$
$$\{\ \mathsf{PSO}, \mathsf{sep\_high}\ \} \rightarrow \{\ \mathsf{P}_0\ \}$$
$$\{\ \mathsf{PSO}, \mathsf{sep\_none}, \mathsf{multimod\_high}\ \} \rightarrow \{\ \mathsf{P}_6\ \}$$
$$\{\ \mathsf{PSO\text{-}BFGS}, \mathsf{sep\_high}, \mathsf{globstruct\_strong}, \mathsf{multimod\_high}\ \} \rightarrow \{\ \mathsf{P}_6\ \}$$
$$\{\ \mathsf{sep\_high}, \mathsf{globstruct\_none}\ \} \rightarrow \{\ \mathsf{P}_0\ \}$$
$$\{\ \mathsf{sep\_none}, \mathsf{multimod\_high}, \mathsf{globstruct\_none}, \mathsf{Alg\_hybrid}\ \} \rightarrow \{\ \mathsf{P}_5\ \}$$
$$\{\ \mathsf{globstruct\_med}\ \} \rightarrow \{\ \mathsf{P}_6\ \}$$

Based on this background knowledge, the recommender system would, e.g., suggest to use the algorithm PSO for functions with high separability. If additionally the function does not exhibit a global structure (e.g., the function is unimodal), then any algorithm would do well. For functions with high multi-modality, no separability, and no global structure, the system would recommend a hybrid algorithm over (1+1)-CMA-ES, confirming previous observations [12].

We note that the quality of the recommendations made by the system depends on the "quality" of the implications constituting the background knowledge. To obtain better recommendations (Step 3), suitable features of the input function need to be efficiently computable (Step 1) and the implications need to form a basis of a sufficient benchmark set (Step 2).

4 Discovering Function Features

It is not clear that the function features currently used in the optimization community define meaningful problem classes with respect to expected algorithm performance. It is an open and non-trivial problem to discover new features for this purpose. We sketch a method that could assist an expert in finding new features using *attribute exploration* from FCA [6]. This goal-oriented approach may be more practical than an undirected synthesis of new features.

In attribute exploration, we start from an initial formal context $\mathbb{K}$ and a set $\mathcal{S}$ of valid implications of $\mathbb{K}$. Then, we compute new implications $A \rightarrow B$ such that $A \rightarrow B$ is valid in $\mathbb{K}$, but does not follow from $\mathcal{S}$, where A and B are sets of attributes. In case such an implication can be found, a (human) expert has to confirm or reject $A \rightarrow B$. In the former case, $A \rightarrow B$ is added to $\mathcal{S}$. If the expert rejects $A \rightarrow B$, s/he has to provide a counterexample that is then added to $\mathbb{K}$. If no new implications can be computed any more, the search terminates.

Consider the context $\mathbb{K}_{\text{features}} = (G, M, I)$, where G is the set of benchmark functions, M the set of function features, and the incidence relation I represents the fact that a function has a certain feature. Applying attribute exploration directly to $\mathbb{K}_{\text{features}}$ would not work, as we are seeking new attributes (features), and not new objects (benchmark functions). Since objects and attributes behave symmetrically in every formal context, however, we can apply attribute exploration to the dual context $\mathbb{K}^{-1}_{\text{features}} = (M, G, I^{-1})$.

Implications from $\mathbb{K}^{-1}_{\text{features}}$ are of the form $\{\ f_{i_1}, \ldots, f_{i_j}\ \} \rightarrow \{\ f_{i_{j+1}}\ \}$, $f_{i_k} \in G$. Presenting such implications to an expert means asking whether all features that

Testfunction	Algorithm	(1+1)-CMA-ES	PSO	PSO-BFGS	Simplex	no multimod.	high multimod.	no gl.struct.	med. gl.struct.	strong gl.struct	no separ.	high separ.	det. Alg.	hybrid Alg.	P_0	P_5	P_6	P_7
Sphere	(1+1)-CMA-ES	✓				✓		✓				✓			✓			
	PSO		✓			✓		✓				✓			✓			
	PSO-BFGS			✓		✓		✓				✓		✓	✓			
	Simplex				✓	✓		✓				✓	✓		✓			
Rastrigin sep.	(1+1)-CMA-ES	✓					✓			✓		✓					✓	
	PSO		✓				✓			✓		✓			✓			
	PSO-BFGS			✓			✓			✓		✓		✓			✓	
	Simplex				✓		✓			✓		✓	✓					✓
Weierstrass	(1+1)-CMA-ES	✓					✓		✓		✓						✓	
	PSO		✓				✓		✓		✓						✓	
	PSO-BFGS			✓			✓		✓		✓			✓			✓	
	Simplex				✓		✓		✓		✓		✓				✓	
Katsuura	(1+1)-CMA-ES	✓					✓	✓			✓						✓	
	PSO		✓				✓	✓			✓						✓	
	PSO-BFGS			✓			✓	✓			✓			✓		✓		
	Simplex				✓		✓	✓			✓		✓			✓		

Table 3. Example Context.

$f_{i_1}, \ldots, f_{i_j}$ have in common are also features of function $f_{i_{j+1}}$. A counterexample would then correspond to a new feature that $f_{i_1}, \ldots, f_{i_j}$ have in common, but that is not shared by $f_{i_{j+1}}$. These are the features we are looking for.

To illustrate the method, suppose that the context $\mathbb{K}_{\text{features}}$ contains the same functions as in Table 3, but the feature *multi-modality* as the only attribute. Then, using our approach, the expert would need to judge the implication $\{\text{Rastrigin}\} \rightarrow \{\text{Weierstrass}\}$, and s/he could reject the implication by providing a "new" feature *global structure strong*, which the separable Rastrigin function has, but the Weierstrass function has not (counterexample).

5 Conclusions

We have outlined the use of FCA in building a recommender system for black-box optimization. The implications resulting from the small, illustrative example presented here are meaningful in that they both confirm known facts and suggest new ones. Attribute exploration could moreover be a powerful tool for discovering new features that are more predictive of the expected performance.

The ideas presented here are work in progress. Ongoing and future work will address the computability/decidability of features on black-box functions, and evaluate the proposed FCA-based recommender system using the BBOB database. We will investigate novel function and algorithm features, and compare the results obtained by FCA with results from clustering and regression analysis.

References

1. A. Auger and N. Hansen. Performance evaluation of an advanced local search evolutionary algorithm. In *Proc. of CEC-2005*, pp. 1777–1784, 2005.
2. A. Auger and N. Hansen. Benchmarking the (1+1)-CMA-ES on the BBOB-2009 function testbed. In *Proc. of GECCO-2009*, pp. 2459–2466. ACM, 2009.
3. B. Bischl, O. Mersmann, H. Trautmann, and M. Preuss. Algorithm selection based on exploratory landscape analysis and cost-sensitive learning. In *Proc. of GECCO-12*, pp. 313–320. ACM, 2012.
4. P. du Boucher-Ryan and D. Bridge. Collaborative recommending using formal concept analysis. *Knowledge-Based Systems*, 19(5):309–315, 2006.
5. M. El-Abd and M. S. Kamel. Black-box optimization benchmarking for noiseless function testbed using particle swarm optimization. In *Proc. of GECCO-2009*, pp. 2269–2274. ACM, 2009.
6. B. Ganter and R. Wille. *Formal Concept Analysis: Mathematical Foundations*. Springer, 1999.
7. N. Hansen, S. Finck, R. Ros, A. Auger, et al. Real-parameter black-box optimization benchmarking 2009: Noiseless functions definitions. 2009.
8. D. I. Ignatov, S. O. Kuznetsov, and J. Poelmans. Concept-based biclustering for internet advertisement. In *ICDM Workshop, 2012*, pp. 123–130, 2012.
9. O. Mersmann, B. Bischl, H. Trautmann, M. Preuss, C. Weihs, and G. Rudolph. Exploratory landscape analysis. In *Proc. of GECCO-11*, pp. 829–836. ACM, 2011.
10. O. Mersmann, M. Preuss, and H. Trautmann. *Benchmarking evolutionary algorithms: Towards exploratory landscape analysis*. Springer, 2010.
11. C. L. Müller, R. Ramaswamy, and I. F. Sbalzarini. Global parameter identification of stochastic reaction networks from single trajectories. In *Advances in Systems Biology*, volume 736, chapter 28, pp. 477–498. Springer, 2012.
12. C. L. Müller and I. F. Sbalzarini. A tunable real-world multi-funnel benchmark problem for evolutionary optimization (and why parallel island models might remedy the failure of CMA-ES on it). In *Proc. of IJCCI-2009*, pp. 248–253, 2009.
13. C. L. Müller and I. F. Sbalzarini. Global characterization of the CEC 2005 fitness landscapes using fitness-distance analysis. In *Proc. EvoStar*, volume 6624 of *Leture Notes in Computer Science*, pp. 294–303. Springer, 2011.
14. M. Munoz, M. Kirley, and S. Halgamuge. Exploratory landscape analysis of continuous space optimization problems using information content. *IEEE Trans. Evol. Comput.*, 2014.
15. M. A. Muñoz, M. Kirley, and S. K. Halgamuge. A meta-learning prediction model of algorithm performance for continuous optimization problems. In *Proc. of PPSN-2012*, pp. 226–235. Springer, 2012.
16. J. A. Nelder and R. Mead. A simplex method for function minimization. *The Computer Journal*, 7(4):308–313, 1965.
17. L. Pál. Comparison of multistart global optimization algorithms on the BBOB noiseless testbed. In *Proc. of GECCO-2013*, pp. 1153–1160, 2013.
18. C. Voglis, G. S. Piperagkas, K. E. Parsopoulos, D. G. Papageorgiou, and I. E. Lagaris. Mempsode: comparing particle swarm optimization and differential evolution within a hybrid memetic global optimization framework. In *Proc. of GECCO-2012*, pp. 253–260. ACM, 2012.
19. D. H. Wolpert and W. G. Macready. No free lunch theorems for optimization. *IEEE Trans. Evol. Comput.*, 1(1):67–82, 1997.

Metric Generalization and Modification of Classification Algorithms Based on Formal Concept Analysis

Evgeny Kolmakov

Faculty of Computational Mathematics and Cybernetics, Lomonosov Moscow State University, Russia

Abstract. FCA-based classifiers can deal with nonbinary data representation in different ways: use it directly or binarize it. Those algorithms that binarize data use metric information from the initial feature space only as a result of scaling (feature binarization procedure). Metric approach in this area allows one significantly reducing classification refusals number and provides additional information which can be used for classifier training. In this paper we propose an approach which generalizes some of existing FCA classification methods and allows one to modify them. Unlike other algorithms, the proposed classifier model uses initial metric information together with order object-attribute dependencies.

Keywords: classification, pattern recognition, formal concept analysis

1 Introduction

Formal concept analysis (FCA) is a branch of applied lattice theory allowing one to formalize some machine learning models. It provides tools to solve various tasks in many domains of computer science, such as knowledge representation and management, data mining, including classification and clustering. There are many FCA-based classification algorithms known [6]. One of the particular features of FCA methods is that object $x \in \mathbb{X}$ is being described using binary attributes. However, in many cases attributes can be, e.g., real numbers, graphs, etc. There are classification methods using nonbinary representation directly, e.g., see these works on pattern structures [4], [5], but many classifiers use it only after scaling procedure. The scaling procedure is the transformation of the initial feature space $\mathcal{F}$ into the Boolean cube B^n. It leads to the significant loss of the metric information provided by $\mathcal{F}$ space. In this paper we propose generalizations and modifications of several FCA-based classifiers, which use scaling procedure, by introducing new classifier model on the basis of class estimates. It generalizes straight hypotheses-based algorithm [1] and both of GALOIS classification procedures [3]. We also define the pseudometric on arbitrary finite lattice, which is based on the ideas from Rulearner rules induction algorithm [2] and so has intelligible interpretation in terms of formal concepts and concept lattice.

In what follows we keep to standard lattice theory and FCA definitions. Therefore here we briefly describe some basic definitions, classifiers and introduce the notation which is used further. Let G and M be an arbitrary sets called the *set of objects* and the *set of attributes* respectively and $I \subseteq G \times M$ be a binary relation. The triple $\mathbb{K} = (G, M, I)$ is called a *formal context*. The following $(\cdot)'$ mappings define a *Galois connection* between 2^G and 2^M sets partially ordered by set-theoretic inclusion:

$$A' = \{m \in M \mid gIm \text{ for all } g \in A\}\,,\ B' = \{g \in G \mid gIm \text{ for all } m \in B\}\,.$$

A pair (A, B), such that $A \subseteq G, B \subseteq M$ and $A' = B, B' = A$ is called a *formal concept* of $\mathbb{K}$ with *formal extent* A and *formal intent* B. For object $g \in G$ we write g' instead of $\{g\}'$. Define the "projection" mappings $ext : (A, B) \mapsto A$ and $int : (A, B) \mapsto B$. Formal concepts of a given context $\mathbb{K}$ form a complete lattice denoted by $\mathfrak{B}(\mathbb{K})$. It is called the *concept lattice* of a context $\mathbb{K}$. Let $\langle L, \wedge, \vee\rangle$ be a lattice and $x \in L$. By $x^{\triangledown}$ ($x^{\vartriangle}$) we denote the order ideal (filter) generated by x. By $At(L), J(L)$ and $M(L)$ we denote the set of all atoms, join-irreducible and meet-irreducible elements of L respectively. The function $f\colon L \to \mathbb{R}$ is called *supermodular* if $f(x) + f(y) \leqslant f(x \vee y) + f(x \wedge y)$ for all $x, y \in L$.

A concept C is called *consistent* if all objects in $ext(C)$ belong to the same class. Both GALOIS classification procedures are described in [3]. GALOIS(1) calculates the similarity $\Gamma_C(x)$ between an object x and each consistent concept C, then x is assigned to the class corresponding to C with the highest value of $\Gamma_C(x)$. GALOIS(2) finds all consistent concepts C satisfying $int(C) \subseteq x'$, then x is assigned to the most numerous class in the previous set.

Let $\mathbb{K} = (G, M, I)$ be a context and $w \notin M$ be a target attribute. The input data for classification may be described by three contexts w.r.t. w: the *positive context* $\mathbb{K}_+ = (G_+, M, I_+)$, the *negative context* $\mathbb{K}_- = (G_-, M, I_-)$ and the *undefined context* $K_\tau = (G_\tau, M, I_\tau)$. G_-, G_+ and G_τ are sets of positive, negative and undefined objects respectively. $I_\epsilon \subseteq G_\epsilon \times M$, where $\epsilon \in \{-, +, \tau\}$ are binary relations that define structural attributes. Galois operators in these contexts are denoted by $(\cdot)^+, (\cdot)^-$, and $(\cdot)^\tau$ respectively. A formal concept of a positive context is called a *positive concept*. Negative and undefined concepts are defined similarly. If the intent B_+ of a positive concept (A_+, B_+) is not contained in the intent g^- of any negative example $g \in G_-$, then it is called a *positive hypothesis* with respect to the property w. A positive intent B_+ is called *falsified* if $B_+ \subseteq g^-$ for some negative example $g \in G_-$. Negative hypotheses are defined similarly. By "hypotheses-based classifier" we mean the classification procedure from [1], which can be described as follows. If unclassified object $g \in G_\tau$ contains a positive but no negative hypotheses, it is classified positively, similar for negative. If g does not contain any positive or negative hypothesis (insufficient data) or contains both positive and negative hypotheses (inconsistent data), then no classification happens.

2 Generalization and Modification of Algorithms

The common drawback of the FCA-based classifiers using binary features after scaling is that they forget the initial feature space metric structure. The main idea of this paper is to use this metric information together with order-theoretic relations between objects and attributes provided by a concept lattice. It is important that $\mathcal{F}$ and B^n spaces with additional structures (metric and formal context) are being used at the same time, providing more possibilities for classifier training methods.

2.1 Metric estimates

Denote by $\mathcal{H}_+$ and $\mathcal{H}_-$ the sets of concepts constructed from a training set, which intents are positive and negative hypotheses respectively. We assume that $(\mathcal{F}, \rho)$ is a metric space, and let $S(x, A)$ be the similarity measure (based on the metric from $\mathcal{F}$, see examples in Section 3) between an object x and a set of objects A. Let us define the estimates for positive and negative classes:

$$\Gamma_+(x) = \sum_{C \in \mathcal{H}_+} I(x, C) S(x, ext(C)), \quad \Gamma_-(x) = \sum_{C \in \mathcal{H}_-} I(x, C) S(x, ext(C)),$$

where $I(x, C) = [int(C) \subseteq x']$ and $[\cdot]$ is the indicator function. Then the classifier will have the following form: $a(x) = \text{sign } \Gamma(x) = \text{sign}(\Gamma_+(x) - \Gamma_-(x))$.

Proposition 1. *If hypotheses-based classifier correctly predicts class label for an object then* $a(x) = \text{sign } \Gamma(x)$ *does the same.*

In comparison with hypotheses-based classifier the number of classification refusals is reduced, but the total error rate can increase.

2.2 Analogy with algorithms based on estimate calculations

To calculate the estimates in the method above we use positive and negative hypotheses sets, i.e. special subsets of concept lattice. Such calculation of estimates can be generalized to an arbitrary concept lattice subsets somehow characterizing individual classes $y \in Y$.

Let $\mathcal{C}$ be the set of concepts which we call *the support concepts system.* Suppose that each concept from $\mathcal{C}$ characterizes only one class $y \in Y$, that is $\mathcal{C} = \bigsqcup_{y \in Y} \mathcal{C}_y$, where Y is the set of classes. Then define the estimate of object x for class y as follows:

$$\Gamma_y(x) = \sum_{C \in \mathcal{C}_y} S(x, C).$$

The classifier will have the following form: $a(x) = \arg\max_{y \in Y} \Gamma_y(x)$. The estimates of this type are similar to the estimates used in estimate calculations methods [7] and the sets $\mathcal{C}_y$ are the support sets analogues.

Consider specific examples of support concepts system $\mathcal{C}$, similarity measure $S(x, C)$ and analyze corresponding classifiers:

1. $\mathcal{C} = \mathcal{H}_+ \bigsqcup \mathcal{H}_-$ are positive and negative hypotheses sets,
   $S(x, C) = [int(C) \subseteq x'] \hat{S}(x, ext(C))$, where $\hat{S}(x, ext(C))$ is the given similarity measure. The corresponding classifier was described above.
2. $\mathcal{C} = \bigsqcup_{y \in Y} \mathcal{C}_y$ is the consistent concepts set.
   If $S(x, C) = |\,(M \setminus int(C)) \cup x'|$ we get modified GALOIS(1) algorithm.
   If $S(x, C) = [int(C) \subseteq x']$ we get GALOIS(2) algorithm.

2.3 Analogy with metric classifiers

Let $\mathcal{C} = \{C_1, \ldots, C_n\}$ be the support concepts system. Suppose that there is the distance measure ρ in $\mathcal{F}$ space. Sort $\mathcal{C}$ in increasing order w.r.t. the values of the distance $\rho(x, C_i)$ between object x and concepts C_i:

$$\rho(x, C_x^{(1)}) \leqslant \rho(x, C_x^{(2)}) \leqslant \cdots \leqslant \rho(x, C_x^{(n)}),$$

where $C_x^{(i)}$ is the i-th neighbour of x among $\mathcal{C}$, $y_x^{(i)}$ is the class, characterized by $C_x^{(i)}$ concept. Define the estimate of object x for class y:

$$\Gamma_y(x) = \sum_{i=1}^{n} w_i(x)[y_x^{(i)} = y],$$

$w_i(x)$ is x i-th neighbour weight (positive function non-increasing w.r.t. i).

The defined estimates are completely analogous to the metric classifiers estimates, except that the neighbours here are not objects but support concepts.

Thus choosing the suitable weights $w_i(x)$ we get analogs of all known metric classifiers (kNN, Parzen window, potential functions and others), but in terms of concepts. For example:

- $w_i(x) = [i \leqslant k]$ is k nearest neighbours method;
- $w_i(x) = [i \leqslant k]w_i$ is k weighted nearest neighbour method (w_i depends only on the neighbour number);
- $w_i(x) = K(\frac{\rho(x, C_x^{(i)})}{h(x)})$ is Parzen window method ($K(z)$ is non-increasing positive-valued function defined on $[0, 1]$, $h(x)$ is the window width).

All the proposed methods are the generalizations of the existing methods and can be used for their modifications. They use both metric information from $\mathcal{F}$ and object-attribute dependencies provided by concept lattice. This allows to reduce the number of classification refusals and error rate.

2.4 Pseudometric on the set of concepts

Another approach which uses the notion of similarity in FCA algorithms is to define a distance function on the set of concepts. In Rulearner algorithm ([2]) the most important characteristics of concept lattice element u were the value of the function $\mathrm{cover}(u) = |J(L) \cap u^{\triangledown}|$ and $M(L) \cap u^{\vartriangle}$ set. The comparison of lattice elements is performed on the basis of these characteristics. In the case of reduced context, this ties up with a fact, that every concept is characterized by its extent (distinct objects correspond to join-irreducible elements) and intent (distinct features correspond to meet-irreducible elements). Thus, $\mathrm{cover}(u)$ corresponds to the number of objects from training set covered by the concept u, and $M(L) \cap u^{\vartriangle}$ corresponds to the attributes characterizing u. We use these observations to define the distance function on an arbitrary finite lattice. Due to the propositions dual to theorems 3.1 and 3.3 from [8], the following theorem holds.

Theorem 1. *Let $\langle L, \wedge, \vee\rangle$ be a lattice and $f\colon L \to \mathbb{R}$ is isotone and supermodular function, then $d_f(x,y) = f(x)+f(y)-2f(x\wedge y)$ defines a pseudometric on this lattice.*

Consider arbitrary finite lattice $\langle L, \wedge, \vee\rangle$, non-empty subset $D \subseteq L$ and a function $f\colon L \to \mathbb{Z}_+$, defined as follows: $f(x) = |D(x)|$, where $D(x) = D \cap x^\triangledown$.

Proposition 2. *The function $f(x)$ is isotone and supermodular.*

Proof. The isotone property of f follows from the following chain of implications:

$$x \leqslant y \Rightarrow x^\triangledown \subseteq y^\triangledown \Rightarrow D(x) \subseteq D(y) \Rightarrow f(x) = |D(x)| \leqslant |D(y)| = f(y).$$

To prove supermodularity consider the following:

$$f(x)+f(y) = |D(x)|+|D(y)| = |D(x)\cup D(y)|+|D(x)\cap D(y)| \leqslant f(x\vee y)+f(x\wedge y).$$

To prove the last inequality observe that $D(x) \cup D(y) \subseteq D(x \vee y)$ follows from the following inclusions:

$$x \leqslant x \vee y \Rightarrow D(x) \subseteq D(x \vee y), \quad y \leqslant x \vee y \Rightarrow D(y) \subseteq D(x \vee y).$$

The equality $D(x) \cap D(y) = D(x \wedge y)$ follows from $x^\triangledown \cap y^\triangledown = (x \wedge y)^\triangledown$. □

Thus, according to the theorem above, the function $f(x)$ induces the pseudometric $d_f(x,y)$ on the lattice, defined by the following equality:

$$d_f(x,y) = f(x) + f(y) - 2f(x \wedge y).$$

The value of $d_f(x,y)$ has simple interpretation.

Proposition 3. $d_f(x,y) = |D(x) \oplus D(y)|$, *where $A \oplus B = (A \setminus B) \cup (B \setminus A)$.*

Proof. From the proof of the proposition above we conclude that the equality $D(x) \cap D(y) = D(x \wedge y)$ holds. Consider the chain of equalities:

$$\begin{aligned}
f(x) + f(y) - 2f(x \wedge y) &= |D(x)| + |D(y)| - 2|D(x \wedge y)| = \\
&= |D(x)| + |D(y)| - 2|D(x) \cap D(y)| = \\
&= |D(x) \cup D(y)| + |D(x) \cap D(y)| - 2|D(x) \cap D(y)| = \\
&= |D(x) \cup D(y)| - |D(x) \cap D(y)| = |D(x) \oplus D(y)|.
\end{aligned}$$

□

Corollary 1. *If $\langle L, \wedge, \vee\rangle$ is a finite Boolean algebra and D is the set of all atoms of L, then $d_f(x,y)$ is exactly the Hamming distance.*

In order to compare formal concepts it is reasonable to choose $D = J(L)$ or $D = At(L)$. In terms of this pseudometric two concepts are the closer, the less object concepts are covered by only one of them. Moreover, the cover(u) and $d_f(x,y)$ functions are tied: cover$(u) = d_f(u, \bigwedge L)$. One of the drawbacks of the defined distance measure is that the number of elements from D covered by $x \wedge y$ is not taken into account. In some cases it may lead to inadequate distance estimates.

Possible modifications:

1. Various normalizations to take the number of elements into account, e.g.:

$$d(x, y) = \frac{|D(x) \oplus D(y)|}{|D(x) \cup D(y)|}.$$

2. Weighting elements of D, e.g. let $D = J(L)$ and w_e be the proportion of the hypotheses covering $e = (g'', g')$. Then $d(x, y)$ will have the following form:

$$d(x, y) = \sum_{e \in D(x) \oplus D(y)} w_e.$$

The distance between concepts can be applied to modify the classification algorithms mentioned above. For example, let an object x be classified with hypotheses-based algorithm. Suppose there are two positive hypotheses H_1^+, H_2^+ and two negative hypotheses H_1^-, H_2^- for the classification of x. In this case the algorithm refuses to classify x. Suppose we know the concept distances $d(H_1^+, H_2^+), d(H_1^-, H_2^-)$ and also $d(H_1^+, H_2^+) \gg d(H_1^-, H_2^-)$. Then it is natural to classify x as positive, because the distant concepts (in terms of the proposed measure) are less "correlated" (since they cover many distinct object concepts), and hence their answers are more significant. Distance between concepts can also be used for reducing the size of concepts system (used by classifier, e.g. consistent concepts) in order to improve generalization ability of classifier, reduce the overfitting and remove concepts based on noisy data.

3 Experiments

In this section the experimental results are presented. The algorithms have been tested on two data sets taken from UCI Machine Learning Repository [9]: SPECT and SPECTF Heart Data Set (training set consists of 80 objects, testing set consists of 187 objects, 22 binary attributes in SPECT, 44 real-valued attributes in SPECTF) and Liver Disorders Data Set (training set consists of 150 objects, testing set consists of 195 objects, 6 real-valued attributes, 30 binary attributes (after scaling)). Tested algorithms: GALOIS(1, 2), Rulearner, straight hypotheses-based algorithm, modified GALOIS(1) (described by the second example in Section 2.2), modified hypotheses-based algorithm with metric estimates (described in Section 2.1 with different similarity functions). Euclidian metric $\rho(x, y)$ was used in $\mathcal{F}$ space in both experiments. Similarity function: $S(x, C) = K(\rho(x, C), a)$, where $K(r, a)$ and $\rho(x, C)$ are one of the following functions:

$$K_1(r, a) = \frac{1}{1 + \exp(ar)}, \qquad K_2(r, a) = \frac{1}{r + a}.$$

$$\rho_1(x, C) = \inf_{c \in C} \rho(x, c), \quad \rho_2(x, C) = \frac{1}{|C|} \sum_{c \in C} \rho(x, c), \quad \rho_3(x, C) = \sup_{c \in C} \rho(x, c).$$

We introduce the following notation: ν_c is the proportion of classified objects, $\nu_r = 1 - \nu_c$ is the proportion of refused classifications, e_t is total error rate (including refusals), e_r is the error rate among classified objects.

Algorithm	ν_c	ν_r	e_t	e_r
GALOIS(1)	1	0	0.1604	0.1604
Modified GALOIS(1)	1	0	0.0856	0.0856
GALOIS(2)	1	0	0.0802	0.0802
Rulearner	0.7487	0.2513	0.2727	0.0286
Hypotheses-based	0.5936	0.4064	0.6150	0.1842
$K = K_1, a = 0.0125, \rho = \rho_1$	0.8021	0.1979	0.3155	0.1467
$K = K_1, a = 0.0125, \rho = \rho_2$	0.8021	0.1979	0.2888	0.1133
$K = K_1, a = 0.0125, \rho = \rho_3$	0.8021	0.1979	0.2834	0.1067
$K = K_1, a = 1, \rho = \rho_2$	0.7273	0.2727	0.3422	0.0956
$K = K_2, a = 1, \rho = \rho_1$	0.8021	0.1979	0.2941	0.1200
$K = K_2, a = 1, \rho = \rho_2$	0.8021	0.1979	0.3209	0.1533

Table 1. SPECT and SPECTF Heart Data Set. Experimental results.

Algorithm	ν_c	ν_r	e_t	e_r
GALOIS(1)	1	0	0.4605	0.4605
Modified GALOIS(1)	1	0	0.5590	0.5590
GALOIS(2)	1	0	0.4359	0.4359
Rulearner	0.9795	0.0205	0.4564	0.4450
Hypotheses-based	0.2923	0.7077	0.8256	0.4035
$K = K_1, a = 1, \rho = \rho_1$	0.8821	0.1179	0.5231	0.4593
$K = K_1, a = 0.01, \rho = \rho_2$	0.8974	0.1026	0.5436	0.4914
$K = K_1, a = 0.25, \rho = \rho_3$	0.8872	0.1128	0.5385	0.4798
$K = K_2, a = 200, \rho = \rho_1$	0.8974	0.1026	0.4769	0.4171
$K = K_2, a = 150, \rho = \rho_2$	0.8974	0.1026	0.4564	0.3943
$K = K_2, a = 150, \rho = \rho_3$	0.8974	0.1026	0.4667	0.4057

Table 2. Livers Disorder Data Set. Experimental results.

The aim of the experiments was to compare FCA classification methods and not to achieve low error rate in solving particular tasks. Hence we used simple scaling procedure: normalizing all attributes to $[0, 1]$ interval and then applying interval-based nominal scaling (the number of intervals was chosen to be 5). It explains high error rate of all classifiers in the second task. Individual scaling (e.g. scaling with floating-size intervals) for each task may significantly reduce error rate, but this work is not focused on this problem. From the results above we may conclude that for hypotheses-based algorithm modifications the number of refusals is substantially reduced together with total error rate e_t. Modified GALOIS(1) classifier improved GALOIS(1) on the first data set and disimproved it on the second. This may be due to the different nature of binary data description: in the first case 22 binary attributes were obtained from 44 real-valued using complex binarization procedure, while in the second one this procedure was very simple. The choice of $K(r,a)$ and $\rho(x,C)$ affects only e_t but not ν_r, hence their accurate selection may improve classification quality.

4 Conclusions

In this paper we have formally described and experimentally studied a new approach to classification which encompasses the usage both of metric information provided by the initial feature space and the order object-attribute dependencies. Also we have defined the pseudometric on arbitrary finite lattice, which has intelligeble interpretation in terms of concepts and hence can be used for comparing concepts in order to improve FCA classification methods. Further developments can be focused on studying of classifiers obtained from the proposed model by fixing the support concepts system $\mathcal{C}$ and the similarity measure $S(x,C)$ and on the possibilities of choosing such support concepts system that allows to construct only a part of concept lattice.

References

1. S.O. Kuznetsov: Complexity of Learning in Concept Lattices from Positive and Negative Examples. Discrete Applied Mathematics, 2004, No. 142(13), pp. 111-125.
2. M. Sahami: Learning classification Rules Using Lattices. N. Lavrac and S. Wrobel eds., pp. 343346, Proc ECML, Heraclion, Crete, Greece (April 1995).
3. C. Caprineto, G. Romano: GALOIS An order-theoretic approach to conceptual clustering. In proceedings of ICML93, pp. 3340, Amherst, USA (July 1993).
4. M. Kaytoue, S.O. Kuznetsov, A. Napoli, S. Duplessis, Mining gene expression data with pattern structures in formal concept analysis. Information Sciences, Volume 181, Issue 10, 15 May 2011, pp. 1989-2001, Information Science, 2011.
5. S.O. Kuznetsov, Scalable Knowledge Discovery in Complex Data with Pattern Structures. In: P. Maji, A. Ghosh, M.N. Murty, K. Ghosh, S.K. Pal, Eds., Proc. 5th International Conference Pattern Recognition and Machine Intelligence (PReMI'2013), Lecture Notes in Computer Science (Springer), Vol. 8251, pp. 30-41, 2013.
6. O. Prokasheva, A. Onishchenko, S. Gurov. Classification methods based on Formal Concept Analysis. FCAIR 2013 Formal Concept Analysis Meets Information Retrieval. Workshop co-located with the 35th European Conference on Information Retrieval (ECIR 2013). March 24, 2013, Moscow, Russia. National Research University Higher School of Economics, pp. 95-104. ISSN 1613-0073
7. Yu.I. Zhuravlev: An Algebraic Approach to Recognition and Classification Problems, in Problems of Cybernetics, Issue 33 (Nauka, Moscow, 1978), pp. 568 [In Russian].
8. Dan A. Simovici: Betweenness, Metrics and Entropies in Lattices. Proceedings of the 38th International Symposium on Multiple Valued Logic. 22-24 May, 2008, Dallas, TX, USA. IEEE Computer Society Washington, pp. 26-31. ISSN 0195-623X
9. Bache, K. & Lichman, M. (2013). UCI Machine Learning Repository [http://archive.ics.uci.edu/ml]. Irvine, CA: University of California, School of Information and Computer Science.

Concept Stability as a Tool for Pattern Selection

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

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

Abstract. Data mining aims at finding interesting patterns from datasets, where "interesting" means reflecting intrinsic dependencies in the domain of interest rather than just in the dataset. Concept stability is a popular relevancy measure in FCA but its behaviour have never been studied on various datasets. In this paper we propose an approach to study this behaviour. Our approach is based on a comparison of stability computation on datasets produced by the same general population. Experimental results of this paper show that high stability of a concept in one dataset suggests that concepts with the same intent in other dataset drawn from the population have also high stability. Moreover, experiments shows some asymptotic behaviour of stability in such kind of experiments when dataset size increases.

Keywords: formal concept analysis, stability, pattern selection, experiments

1 Introduction

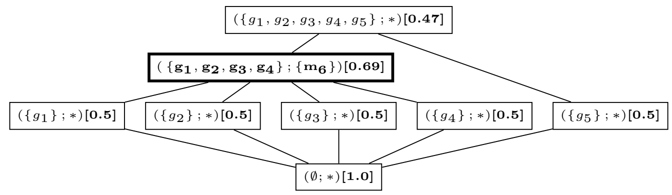
In data mining, many usefulness measures of patterns are introduced. For example, more than 30 statistical methods are enumerated and discussed in [1]. Such a high number of different approaches to pattern selection emphasizes the importance of the problem. In this paper we would like to focus on a measure which is introduced within Formal Concept Analysis (FCA). FCA is a mathematical formalism having many applications in data analysis [2]. Starting from the set of objects and the corresponding sets of attributes FCA tends to generalize the descriptions for any set of objects. Although this approach is less efficient than the statistical methods it is still feasible and ensures that no potentially interesting pattern is missed.

Within FCA there are several approaches for pattern selection. Two disjoint approaches can be distinguished. The first one is to introduce background knowledge into the procedure computing concepts [3–5]. These approaches allow one to find patterns which are likely to be useful for the current task. Although the number of resulting patterns can be significantly reduced, they are still numerous. The second approach can be applied in a composition with the first ones, ranking the resulting patterns w.r.t. a relevance measure.

The authors of [6] provide several measures for ranking concepts that stem from the algorithms possibly underlying human behavior. Stability is another

	m_1	m_2	m_3	m_4	m_5	m_6
g_1	x					x
g_2		x				x
g_3			x			x
g_4				x		x
g_5					x	

(a)



(b)

Fig. 1: A toy formal context (a) and the correspnoding concept lattice with stability indexes (b).

measure for ranking concepts, introduced in [7] and later revised in [8–10]. Several other methods are considered in [11], where it is shown that stability is more reliable in noisy data. For the moment, stability seems to be the most widely used usefulness measure around the FCA community. Thus, in this paper we are going to focus on stability. Although this measure is often used, there is neither a reliable comparison nor a deep research on its usefulness. Consequently, the goal of this paper is to evaluate the usefulness of stability. Here we experimentally prove that the stability for a pattern is coherent with the stability computed for the same pattern but w.r.t. a different dataset coming from the same population (the similarly distributed dataset).

The rest of the paper is organised as follows. Section 2 introduces definition of stability and discusses known stability estimates. In Section 3 experiments on relevancy of stability are discussed.

2 Stability of a formal concept

2.1 Formal concept analysis (FCA)

FCA [2] is a formalism for data analysis. FCA starts with a formal context and builds a set of formal concepts organized within a concept lattice. A formal context is a triple (G, M, I), where G is a set of objects, M is a set of attributes and I is a relation between G and M, $I \subseteq G \times M$. In Figure 1a, a formal context is shown. A Galois connection between G and M is defined as follows:

$$A' = \{m \in M \mid \forall g \in A, (g, m) \in I\}, \qquad A \subseteq G$$
$$B' = \{g \in G \mid \forall m \in B, (g, m) \in I\}, \qquad B \subseteq M$$

The Galois connection maps a set of objects to the maximal set of attributes shared by all objects and reciprocally. For example, $\{g_1, g_2\}' = \{m_6\}$, while $\{m_6\}' = \{g_1, g_2, g_3, g_4\}$.

Definition 1. *A formal concept is a pair (A, B), where A is a subset of objects, B is a subset of attributes, such that $A' = B$ and $A = B'$, where A is called the extent of the concept, and B is called the intent of the concept.*

For example, a pair $(\{g_1, g_2, g_3, g_4\}; \{m_6\})$ is a formal concept. Formal concepts can be partially ordered w.r.t. the extent inclusion (dually, intent inclusion). For example, $(\{g_3\}; \{m_3, m_6\}) \leq (\{g_1, g_2, g_3, g_4\}; \{m_6\})$. This partial order of concepts is shown in Figure 1b.

2.2 The definition of stability

Stability is an interestingness measure of a formal concept introduced in [7] and later revised in [8, 10].

Definition 2. *Given a concept c, concept stability Stab(c) is defined as*

$$Stab(c) := \frac{|\{s \in \wp(Ext(c)) \mid s' = Int(c)\}|}{2^{|Ext(c)|}} \tag{1}$$

i.e., the relative number of subsets of the concept extent (denoted by $Ext(c)$), whose description (i.e., the result of $(\cdot)'$) is equal to the concept intent (denoted by $Int(c)$) where $\wp(P)$ is the power set of P.

Example 1. Figure 1b shows the concept lattice of the context in Figure 1a, for simplicity some intents are not given. The extent of the highlighted concept c is $Ext(c) = \{g_1, g_2, g_3, g_4\}$, thus, its power set contains 2^4 elements. The descriptions of 5 subsets of $Ext(c)$ ($\{g_1\}, \ldots, \{g_4\}$ and $\emptyset$) are different from $Int(c) = \{m_6\}$, while all other subsets of $Ext(c)$ have a description equal to $\{m_6\}$. So, $Stab(c) = \frac{2^4-5}{2^4} = 0.69$.

Stability measures the dependence of a concept intent on objects of the concept extent. In [10] it is shown that stability of a concept c is the relative number of subcontexts where there exists the concept c with intent $Int(c)$. A stable concept can be found in many such subcontexts, and therefore is likely to be found in an unrelated context built from the population under study.

In some papers it is noticed that in large datasets most of the concepts tends to have stability close to 1 [12, 13]. Thus, in order to distinguish between them we use the following logarithmic stability:

$$LStab(c) = -\log_2(1 - Stab(c)) \tag{2}$$

Stability computation is #P-complete [7, 8]. In this paper we rely on the algorithm from [10], with a worst-case complexity of $O(L^2)$, where L is the size of the concept lattice. However, generally it is quite efficient on real data.

3 Experiment on relevancy of stability

Experiments on behaviour of stability are carried out on public datasets available from the UCI repository [14]. These datasets are shown in Table 1. With their different size and complexity, these datasets provide a rich experimental basis. Complexity here stands for the size of the concept lattice given the initial number

Table 1: Datasets used in the experiments. Column 'Shortcut' refers to the short name of the dataset used in the rest of the paper; 'Size' is the number of objects in the dataset; 'Max. Size' is the maximal number of objects in a random subset of the dataset the concept lattice can be computed for; 'Max. Lat. Size' is the size of the corresponding concept lattice; 'Lat. Time' is the time in seconds for computing this lattice; 'Stab. Time' is the time in seconds to compute stability for every concept in the maximal lattice.

Dataset	Shortcut	Size	Max. Size	Max. Lat. Size	Lat. Time	Stab. Time
Mushrooms[1]	Mush	8124	8124	$2.3 \cdot 10^5$	324	57
Plants[2]	Plants	34781	1000	$2 \cdot 10^6$	45	10^4
Chess[3]	Chess	3198	100	$2 \cdot 10^6$	30	$7.4 \cdot 10^3$
Solar Flare (II)[4]	Flare	1066	1066	2988	0	0
Nursery[5]	Nurs	12960	12960	$1.2 \cdot 10^5$	245	5

[1] `http://archive.ics.uci.edu/ml/datasets/Mushroom`

[2] `http://archive.ics.uci.edu/ml/machine-learning-databases/plants/`

[3] `http://archive.ics.uci.edu/ml/datasets/Chess+(King-Rook+vs.+King-Pawn)`

[4] `http://archive.ics.uci.edu/ml/datasets/Solar+Flare`

[5] `http://archive.ics.uci.edu/ml/datasets/Nursery`

of objects in the corresponding context. For example, `Chess` is the most complex dataset as for only 100 objects in the context there are already $2 \cdot 10^6$ of concepts in the concept lattice.

When computing stability, one wants to know if the intent of a stable concept is a general characteristic rather than an artefact specific for a dataset. For that it is necessary to evaluate stability w.r.t. a test dataset different from the reference one. Reference and test datasets are two names of disjoint datasets on which the stability behaviour is evaluated. In order to do that the following scheme of experiment is developed:

1. Given a dataset $\mathbb{K}$ of size K objects, experiments are performed on dataset subsets whose size in terms of number of objects is N. This size is required to be at least half the size of K. For example, for a dataset of size $K = 10$ the size of it subset can be $N = 4$.
2. Two disjoint dataset subsets $\mathbb{K}_1$ and $\mathbb{K}_2$ of size N (in terms of objects) of dataset $\mathbb{K}$ are generated by sampling, e.g., $\mathbb{K}_1 = \{g_2, g_5, g_6, g_9\}$ and $K_2 = \{g_3, g_7, g_8, g_{10}\}$. Later, $\mathbb{K}_1$ is used as a reference dataset for computing stability, while $\mathbb{K}_2$ is a test dataset for evaluating stability computed in $\mathbb{K}_1$.
3. The corresponding sets of concepts $\mathcal{L}_1$ and $\mathcal{L}_2$ with their stability are built for both datasets $\mathbb{K}_1$ and $\mathbb{K}_2$.
4. The concepts with the same intents in $\mathcal{L}_1$ and $\mathcal{L}_2$ are declared as corresponding concepts.
5. Based on this list of corresponding concepts, a list of pairs $S = \{\langle X, Y \rangle, \dots\}$ is built, where X is the stability of the concept in $\mathcal{L}_1$ and Y is the stability of the corresponding concept in $\mathcal{L}_2$. If an intent exists only in one dataset, its stability is set to zero in the other dataset (following the definition of

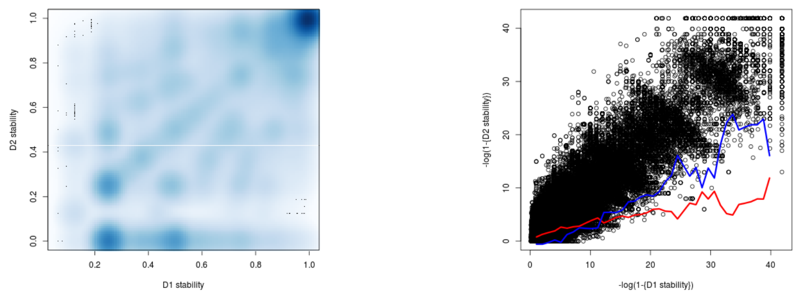


Fig. 2: Stability in the test dataset w.r.t the reference one in Mush4000 in (a) plane scale (b) logarithmic scale.

stability). Finally, the list $LS = \{\langle X_{\log}, Y_{\log}\rangle, \dots\}$ includes the stability pairs from S in logarithmic scale as stated by Eq. (2). We study here the sets S and LS.

The idea of evaluating stability computed on a reference dataset w.r.t. a test dataset comes from the supervised classification methods. Moreover, this idea is often used to evaluate statistical measures for pattern selection and can be found as a part of pattern selection algorithms with a good performance [15].

Sets of pairs S and LS can be drawn by matching every point $\langle X, Y\rangle$ to a point in a 2D-plot. The best case is $y = x$. It means that stability computed for dataset part $\mathbb{K}_1$ is exactly the same as stability computed for the dataset part $\mathbb{K}_2$. However, this is hardly the case in real-world experiments. For example, Figure 2a shows the corresponding diagram for the dataset `Mush4000`.[1] This figure also highlights the fact that many concepts have stability close to 1. However, when the logarithmic set LS is used, a blurred line $y = x$ can be perceived in Figure 2b. Moreover, selecting the concepts which are stable w.r.t. a high threshold in the reference dataset $\mathbb{K}_1$, the corresponding concepts in $\mathbb{K}_2$ are stable w.r.t. a lower threshold. Thus, we can conclude that stability is more tractable in the logarithmic scale, and, thus, we only consider this logarithmic scale in the rest of the paper.

3.1 Setting a stability threshold

In the previous subsection it is mentioned that concepts stable in the reference dataset are stable in the test dataset with a smaller threshold. *But what is "smaller"?* Imagine that in the reference dataset $\mathbb{K}_1$ we have the threshold θ_1, i.e., if $Stab(c) \geq \theta_1$ then c is stable, while in the $\mathbb{K}_2$ we have θ_2. Then, we want to know the threshold θ_1 such that at least 99% of stable concepts in $\mathbb{K}_1$

[1] From here, the name of a dataset followed by a number such as '*NameN*' refers to an experiment based on the dataset *Name* where $\mathbb{K}_1$ and $\mathbb{K}_2$ are of the size N.

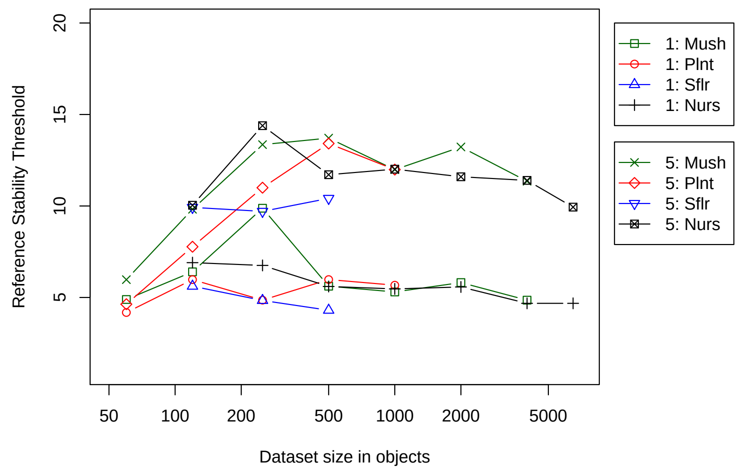


Fig. 3: Stability threshold in the reference dataset ensuring that 99% of concepts in the test datasets corresponding to stable concepts are stable with stability thresholds 1 or 5.

corresponds to stable concepts in $\mathbb{K}_2$. Figure 3 shows the reference threshold θ_1 (x-axis) w.r.t. the size of the datasets (y-axis) for $\theta_2 = 1$ and $\theta_2 = 5$. For example, the line '5: Mush' corresponds to the line of θ_1, where θ_2 is fixed to 5 w.r.t. to the size of the dataset built from dataset Mushrooms. The value $\theta_2 = 1$ means that any stable concept is just found in the test dataset, while $\theta_2 = 5$ requires that they are quite stable in the test dataset. We can see that for large datasets the stability threshold is independent of the dataset, while for small datasets the diversity is higher. We can see that the value of θ_1 should be set to 5–6 in order to ensure that 99% of stable concepts have corresponding concepts in another dataset.

3.2 Stability and ranking

Another way of using usefulness measures is pattern ranking. Thus, it is an interesting question if the order of patterns could be preserved by using stability. A way to study an order of an array ar is to compute its sorting rate r, i.e., the relative number of pairs in the array sorted in the ascending order: $r = 2 \cdot \frac{\{(i,j) | i<j \text{ and } ar_i \le ar_j\}}{|ar| \cdot (|ar|-1)}$. A sorting rate equal to 1 means that the array is in the ascending order, while 0 means that it is in the descending order; the value 0.5 means that there is no order at all. Figure 4 shows the sorting rate (SR) for different datasets, i.e., the sorting rate of concept stabilities in $\mathbb{K}_2$, ordered w.r.t. stabilities of the corresponding stable concepts in K_1. We can see that SR for all datasets is slowly increasing preserving nearly the same value along the stability threshold in $\mathbb{K}_1$. And, thus, concept stability can be used to rank concepts.

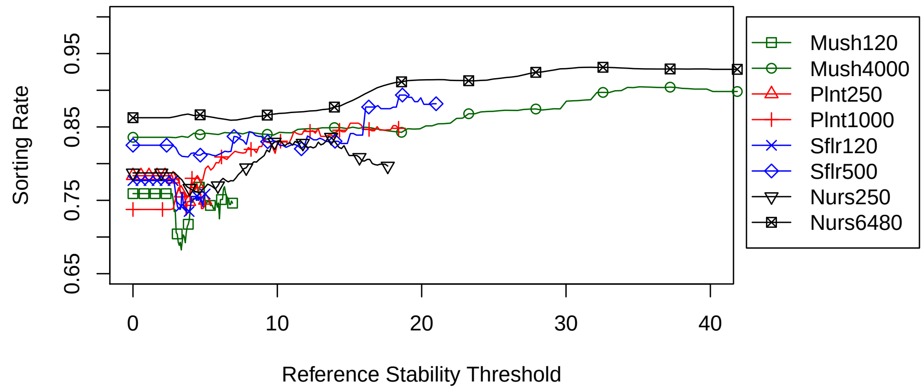


Fig. 4: Global sorting rate for different datasets.

4 Conclusion

In this paper we study concept stability as an efficient measure for pattern selection. It is shown that stability computed in the logarithmic scale is more convenient since it allows one to better distinguish stable concepts. Given a threshold of stability, patterns whose stability are above a threshold in a given dataset are likely to have stability above a smaller threshold in another dataset coming from the same distribution. However, independently of a dataset, as found experimentally, a concept should have logarithmic stability more than 5 in order to reflect any property of the population. We also show that stability is able to sort concepts in two independent datasets with nearly the same order by selecting concepts with stability above a certain threshold.

There are many future research directions. The found properties of stability suggest that interesting concepts can be found by resampling, i.e., analyzing many small parts of a large dataset, thus providing a key to an efficient processing of datasets with Formal Concept Analysis. The second important direction is to develop a methodology for comparison of stability and other known approaches for pattern selection.

Acknowledgements

This research was supported by the Basic Research Program at the National Research University Higher School of Economics (Moscow, Russia) and by the BioIntelligence project (France).

References

1. Masood, A., Soong, S.: Measuring Interestingness – Perspectives on Anomaly Detection. Computer Engineering and Intelligent Systems **4**(1) (2013) 29–40

2. Ganter, B., Wille, R.: Formal Concept Analysis: Mathematical Foundations. 1st edn. Springer (1999)
3. Bělohlávek, R., Vychodil, V.: Formal Concept Analysis with Constraints by Closure Operators. In Schärfe, H., Hitzler, P., Ohrstrom, P., eds.: Conceptual Structures: Inspiration and Application. Volume 4068 of Lecture Notes in Computer Science. Springer Berlin Heidelberg (2006) 131–143
4. Belohlavek, R., Vychodil, V.: Formal Concept Analysis With Background Knowledge: Attribute Priorities. IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews) **39**(4) (July 2009) 399–409
5. Buzmakov, A., Egho, E., Jay, N., Kuznetsov, S.O., Napoli, A., Raïssi, C.: On Projections of Sequential Pattern Structures (with an application on care trajectories). In: Proc. 10th International Conference on Concept Lattices and Their Applications. (2013) 199–208
6. Belohlavek, R., Trnecka, M.: Basic Level in Formal Concept Analysis: Interesting Concepts and Psychological Ramifications. In: Proceedings of the Twenty-Third International Joint Conference on Artificial Intelligence. IJCAI'13, AAAI Press (August 2013) 1233–1239
7. Kuznetsov, S.O.: Stability as an Estimate of the Degree of Substantiation of Hypotheses on the Basis of Operational Similarity. Automatic Documentation and Mathematical Linguistics (Nauch. Tekh. Inf. Ser. 2) **24**(6) (1990) 62–75
8. Kuznetsov, S.O.: On stability of a formal concept. Annals of Mathematics and Artificial Intelligence **49**(1-4) (2007) 101–115
9. Kuznetsov, S., Obiedkov, S., Roth, C.: Reducing the Representation Complexity of Lattice-Based Taxonomies. In Priss, U., Polovina, S., Hill, R., eds.: Conceptual Structures: Knowledge Architectures for Smart Applications. Volume 4604 of Lecture Notes in Computer Science. Springer Berlin Heidelberg (2007) 241–254
10. Roth, C., Obiedkov, S., Kourie, D.G.: On succinct representation of knowledge community taxonomies with formal concept analysis. International Journal of Foundations of Computer Science **19**(02) (April 2008) 383–404
11. Klimushkin, M., Obiedkov, S.A., Roth, C.: Approaches to the Selection of Relevant Concepts in the Case of Noisy Data. In: Proc. of the 8th International Conference on Formal Concept Analysis. ICFCA'10, Springer (2010) 255–266
12. Jay, N., Kohler, F., Napoli, A.: Analysis of Social Communities with Iceberg and Stability-Based Concept Lattices. In Medina, R., Obiedkov, S., eds.: Formal Concept Analysis. Volume 4933 of Lecture Notes in Computer Science. Springer Berlin Heidelberg (2008) 258–272
13. Buzmakov, A., Kuznetsov, S.O., Napoli, A.: Scalable Estimates of Concept Stability. In Glodeanu, C.V., Kaytoue, M., Sacarea, C., eds.: Formal Concept Analysis. Volume 8478 of Lecture Notes in Computer Science. Springer Berlin Heidelberg (2014) 157–172
14. Frank, A., Asuncion, A.: UCI Machine Learning Repository [http://archive.ics.uci.edu/ml]. University of California, Irvine, School of Information and Computer Sciences (2010)
15. Webb, G.I.: Discovering Significant Patterns. Machine Learning **68**(1) (2007) 1–33

About Universality and Flexibility of FCA-based Software Tools

A.A. Neznanov, A.A. Parinov

National Research University Higher School of Economics,
20 Myasnitskaya Ulitsa, Moscow, 101000, Russia
ANeznanov@hse.ru, AParinov@hse.ru

Abstract. There is a big gap between variety of applications of Formal Concept Analysis (FCA) methods and general-purpose software implementations. We discuss history of FCA-based software tools, which is relatively short, main problems in advancing of such tools, and development ideas. We present Formal Concept Analysis Research Toolbox (FCART) as an integrated environment for knowledge and data engineers with a set of research tools based on Formal Concept Analysis. FCART helps us to illustrate methodological and technological problems of FCA approach to data analysis and knowledge extraction.

Keywords: Formal Concept Analysis, Knowledge Extraction, Data Mining, Software.

1 Introduction

Formal Concept Analysis (FCA) [1] is a mature group of mathematical models and a well foundation for methods of data mining and knowledge extraction. There is a huge amount of publications about many aspects of FCA using in different application fields.

Why are there no universal and flexible FCA based tools have implemented in data mining software? We can assume that last ten years is an enough time to implement such tools not only in the most popular big analytic software but also in separate instruments are tightly integrated with other data access tools. Is there a problem in methodology of current "FCA data mining" approach?

Most popular FCA tools are small utilities purposed for final drawing of relatively small (tens to hundred concepts) formal concept lattice and calculate its properties. The other tools are very experimental and separately implement one or several mathematical models.

Can we have exchangeable set of tools for iterative building a formal context from raw data, for working with big lattices, for deep analysis of properties of interesting concepts in good user interface?

2 Problems and solutions

The FCA community started a discussion of the universal FCA workflow several years ago. There were a number of approaches to extending FCA to richer descriptions and at the same time to extending scalability of FCA-based methods proposed:

1. Pattern structures [2, 3, 4, 5] for dealing with complex input objects (such objects may have, for example, graph representation).
2. Various scaling techniques [6, 7, 8] for dealing with many-valued contexts.
3. Similarity measures on concepts [9, 10].
4. Various concepts indices for important concepts selection [11, 12, 13].
5. Alpha lattices [14] (and other coarser variants of concept lattice).
6. Relational concept analysis [15].
7. Attribute exploration [16].
8. Approaches for fragmentary lattice visualization (from iceberg to embedded lattices).
9. "Fuzzy FCA" approach in form of biclustering methods [17] and other techniques.

Near the middle of the last decade there were very successful implementations of transforming a small context into a small line diagram and calculate implications and association rules. The most well-known open source projects are ConExp [18], Conexp-clj [19], Galicia [20], Tockit [21], ToscanaJ [22], FCAStone [23], Lattice Miner [24], OpenFCA [25], Coron [26], Cubist [27]. These tools have many advantages. However, they suffer from the lack of rich data preprocessing, the abilities to communicate with various data sources, session management, reproducibility of computational experiments. It prevents researchers from using these programs for analyzing complex big data without different additional third party tools.

The goal of our efforts is an integration of ideas, methods, and algorithms, which are mentioned all above, in one environment. It is not only a programing task. At first, it is a challenge facing methodologists of the FCA community. The most significant problems:

1. Sense of FCA approaches in AI tasks.
2. Performance of basic FCA algorithms.
3. Logical complexity of raw data preprocessing task.
4. Scaling of many-valued contexts.
5. Interactive work with FCA artifacts.

Now we want to discuss some aspects of integration problems. In a previous article [28], we have described the stages of development of a software system for information retrieval and knowledge extraction from various data sources (textual data, structured databases, etc.). Formal Concept Analysis Research Toolbox (FCART) was designed especially for the analysis of semi structured and unstructured (including textual) data. In this article, we describe FCART as an extensible toolset for checking different integration ideas. As an example of current development activities, the main FCA workflow will be demonstrated.

3 Methodology

FCART was originated from DOD-DMS platform which creation was inspired by the CORDIET methodology (abbreviation of Concept Relation Discovery and Innovation Enabling Technology) [29] developed by J. Poelmans at K.U. Leuven and P. Elzinga at the Amsterdam-Amstelland police. The methodology allows analyst to obtain new knowledge from data in an iterative ontology-driven process. In FCART we concentrated on main FCA workflow and tried to support high level of extensibility. FCART is based on several basic principles that aim to solve main integration problems:

1. Iterative process of data analysis using adjustable data queries and interactive *analytic artifacts* (such as concept lattice, clusters, etc.).
2. Separation between processes of *data querying* (from various data sources), *data preprocessing* (of locally saved immutable snapshots), *data analysis* (in interactive visualizers of immutable analytic artifacts), and *results presentation* (in report editor).
3. Extendibility at three levels: customizing settings of data access components, query builders, solvers and visualizers; writing scripts (macros); developing components (add-ins).
4. Explicit definition of all analytic artifacts and their types. It provides consistent parameters handling, links between artifacts for end-user and allows one to check the integrity of session.
5. Realization of integrated performance estimation tools and system log.
6. Integrated documentation of software tools and methods of data analysis.

The core of the system supports knowledge discovery techniques, based on Formal Concept Analysis, clustering, multimodal clustering, pattern structures and the others. From the analyst point of view, basic *FCA workflow* in FCART has four stages. On each stage, a user has the ability to import/export every artifact or add it to a report.

1. Filling Local Data Storage (LDS) of FCART from various external SQL, XML or JSON-like data sources (querying external source described by External Data Query Description – EDQD). EDQD can be produced by some External Data Browser.
2. Loading a data snapshot from local storage into current analytic session (snapshot is described by Snapshot Profile). Data snapshot is a data table with annotated structured and text attributes, loaded in the system by accessing LDS.
3. Transforming the snapshot to a binary context (transformation described by Scaling Query).
4. Building and visualizing formal concept lattice and other artifacts based on the binary context in a scope of analytic session.

FCART has been already successfully applied to analyzing data in medicine, criminalistics, and trend detection.

4 Technology

We use Microsoft and Embarcadero programming environments and different programming languages (C++, C#, Delphi, Python and other). Native executable of FCART client (the core of the system) is compatible with Microsoft Windows 2000 and later and has no additional dependences. Scripting is an important feature of FCART. Scripts can do generating and transforming artifacts, drawing, and building reports. For scripting we use Delphi Web Script and Python languages. Integrated script editor with access to *artifacts API* allows quick implementation of experimental algorithms for generating, transforming, and visualizing artifacts.

Current version of FCART consists of the following components.

- Core component includes
  - multiple-document user interface of research environment with session manager and extensions manager,
  - snapshot profiles editor (SHPE),
  - snapshot query editor (SHQE),
  - query rules database (RDB),
  - session database (SDB),
  - main part of report builder.
- Local Data Storage (LDS) for preprocessed data.
- Internal solvers and visualizers of artifacts.
- Additional plugins, scripts and report templates.

FCART Local Data Storage (LDS) plays important role in effectiveness of whole data analysis process because all data from external data storages, session data and intermediate analytic artifacts saves in LDS. All interaction between user and external data storages is carried out through the LDS. There are many data storages, which contain petabytes of collected data. For analyzing such Big Data an analyst cannot use any of the software tools mentioned above. FCART provides different scenarios for working with local and external data storages. In the previous release of FCART an analyst could work with quite small external data storage because all data from external storage is converted into JSON files and is saved into LDS. In the current release, we have improved strategies of working with external data. Now analyst can choose between loading all data from external data storage to LDS and accessing external data by chunks using paging mechanism. FCART-analyst should specify the way of accessing external data at the property page of the generator.

All interaction between a client program and LDS goes through the web-service. The client constructs http-request to the web-service. The http-request to the web service is constructed from two parts: prefix part and command part. Prefix part contains domain name and local path (e.g. http://zeus.hse.ru/lds/). The command part describes what LDS has to do and represents some function of web-service API. Using web-service commands FCART client can query data from external data storages in uniform and efficient way.

5 Interactive part of main FCA workflow

The most interesting part for analyst is an interactive work with artifacts in multiple-document user interface of FCART client.

We will illustrate our vision with real examples of querying data and working with big lattices. FCART supports interactive browsing of concept lattices with more than 20000 concepts (see Fig. 1) with fine adjustment of drawing. User may feel some interaction delays but system have special instruments to visualize and navigate big lattices: scaling of lattice element sizes, parents-children navigator, filter-ideal selection, separation of focused concepts from other concepts. We have tested several drawing techniques for visualizing fragments of big lattices and prepared a collection of additional drawing scripts for rendering focused concept neighborhood, selected concepts, "important" concepts, and filter-ideal with different sorting algorithms for lattice levels.

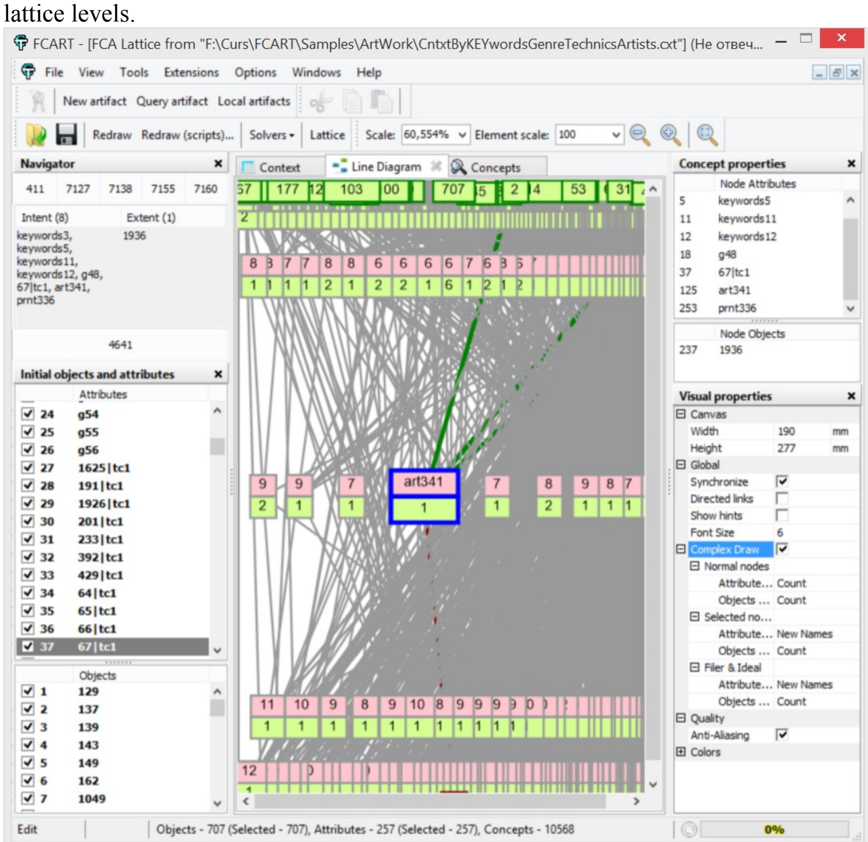


Fig. 1. Lattice browser with focused concept separation

User also can build linked sublattices and calculate standard association rules, implication basis, etc. All artifacts can be commented, annotated and appended to one of reports in a current session.

FCART also supports working with concepts in a form of table with sorting and grouping abilities (see Fig. 2). Concept properties in the last column of this table (indices [30], similarity measures, etc.) can be calculated by scripts.

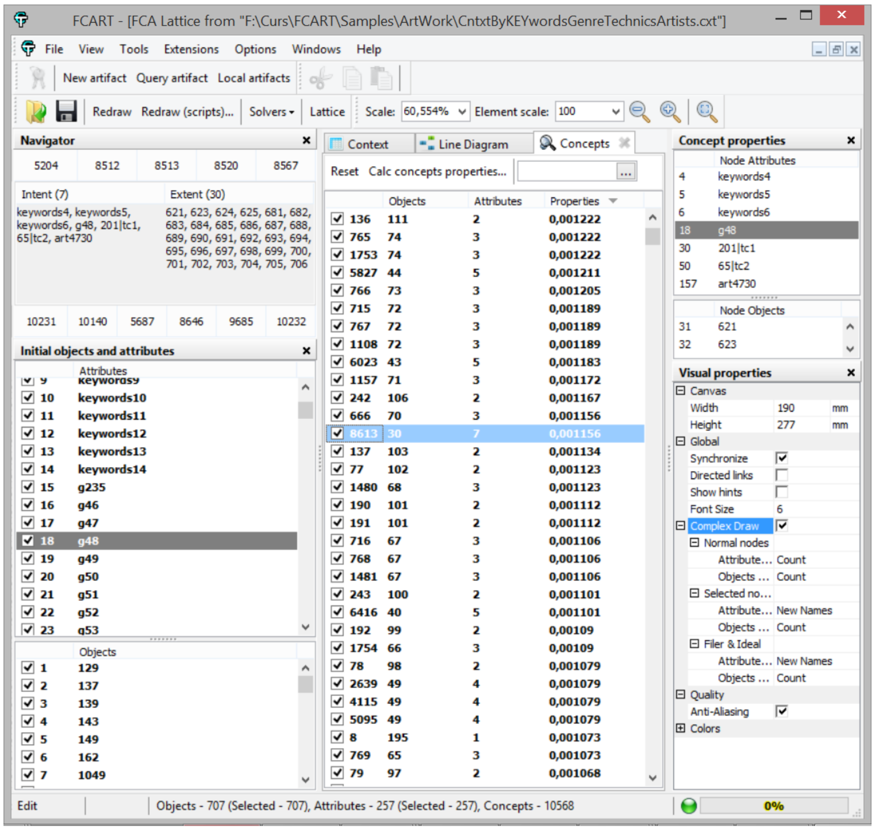


Fig. 2. List of all formal concepts, sorted by value of concept stability index

6 Conclusion and future work

In this paper, we have discussed important questions for all researchers in FCA field about implementation of essential software tools. We will try to suggest solutions for some of those problems in our system. The version 0.9.4 of FCART client (http://ami.hse.ru/issa/Proj_FCART) and version 0.2 of LDS Web-service (http://zeus2.hse.ru) have been introduced this spring.

We have tested preprocessing in the current release with Amazon movie reviews dataset (7.8 million of documents – it is big enough for check standard FCA limitations) [31]. This dataset was loaded to FCART LDS and transformed into many-valued contexts, binary contexts, lattices and other artifacts. The main goal of the current release is to develop architecture, which can work with really big datasets. For now, FCART is being tested on other collections of CSV, SQL, XML, and JSON data with unstructured text fields.

The release of the version 1.0 of FCART is planned for August 2014. We will discuss this version at the seminar and touch on flexibility, extensibility, and overall performance issues. We are opened for ideas to improve methodology and various aspects of implementation.

Acknowledgements

This work was carried out by the authors within the project "Mathematical Models, Algorithms, and Software Tools for Intelligent Analysis of Structural and Textual Data" supported by the Basic Research Program of the National Research University Higher School of Economics.

References

1. Ganter, B., Wille R. Formal Concept Analysis: Mathematical Foundations, Springer, 1999.
2. Ganter, B., Kuznetsov, S.O. Pattern Structures and Their Projections. Proc. 9th International Conference on Conceptual Structures (ICCS-2001), 2001, pp. 129-142.
3. Kuznetsov, S.O. Pattern Structures for Analyzing Complex Data // Proc. of 12th International conference on Rough Sets, Fuzzy Sets, Data Mining and Granular Computing (RSFDGrC-2009), 2009, pp. 33-44.
4. Kuznetsov, S.O. Fitting Pattern Structures to Knowledge Discovery in Big Data. Proc. of International Conference on Formal Concept Analysis (ICFCA'13), 2013, pp. 254-266.
5. Buzmakov, A., Egho, E., Jay, N., Kuznetsov, S.O., Napoli, F., Chedy Raïssi: On Projections of Sequential Pattern Structures (with an Application on Care Trajectories) // Proc. of conference on Concept Lattices and their Applications (CLA-2013), 2013: pp. 199-208.
6. Ganter, B., Stahl, J., Wille, R. Conceptual Measurement and many-valued contexts. In Gaul, W., Schader, M. Classification as a Tool of Research. Elsevier, 1986, pp. 169-176.
7. Prediger, S. Logical Scaling in Formal Concept Analysis // Proc. of ICCS'97, LNAI 1257, 1997, pp. 332-341.
8. Stumme, G. Hierarchies of Conceptual Scales // Proc.Workshop on Knowledge Acquisition, Modeling and Management KAW'99, 2, 1999, pp 78-95.
9. Ceja , J.M.O., Arenas, A.G. Concept similarity measures the understanding between two agents. Natural Language Processing and Information Systems, Lecture Notes in Computer Science, Volume 3136, 2004, pp. 182-194.
10. Alqadah, F., Bhatnagar, R. Similarity measures in formal concept analysis. Annals of Mathematics and Artificial Intelligence, Volume 61, Issue 3, 2011, pp. 245-256.
11. Kuznetsov, S.O. On stability of a formal concept. Annals of Mathematics and Artificial Intelligence, Vol. 49, 2007, pp. 101-115.

12. Ilvovsky, D.A., Klimushkin, M.A. FCA-based Search for Duplicate Objects in Ontologies, Proceedings of the Workshop Formal Concept Analysis Meets Information Retrieval, Vol. 977. CEUR Workshop Proceeding, 2013.
13. Belohlávek, R., Trnecka, M. Basic Level in Formal Concept Analysis: Interesting Concepts and Psychological Ramifications // Proc. 23th International Joint Conference on Artificial Intelligence, 2013, pp. 1233-1239.
14. Soldano, H., Ventos, V., Champesme, M., Forge, D. Incremental construction of Alpha lattices and association rules // Proc. 14th International Conference, 2010, pp. 351-360.
15. Hacene, M.R., Huchard, M., Napoli, A., Valtchev, P. Relational concept analysis: mining concept lattices from multi-relational data. Ann. Math. Artif. Intell. 67(1), 2013, pp. 81-108.
16. Ganter, B. Attribute exploration with background knowledge. Theoretical Computer Science, 217(2), 1999, pp. 215-233.
17. Ignatov, D.I., Kuznetsov, S.O., Poelmans, J. Concept-Based Biclustering for Internet Advertisement // ICDM Workshop, IEEE Computer Society, 2012, pp. 123-130.
18. Yevtushenko, S.A. System of data analysis "Concept Explorer" (In Russian) // Proceedings of the 7th national conference on Artificial Intelligence KII-2000, pp. 127-134, Russia, 2000.
19. Conexp-clj (http://daniel.kxpq.de/math/conexp-clj/)
20. Valtchev, P., Grosser, D., Roume, C. Hacene, M.R. GALICIA: an open platform for lattices, in Using Conceptual Structures // Contributions to the 11th Intl. Conference on Conceptual Structures (ICCS'03), pp. 241-254, Shaker Verlag, 2003.
21. Tockit: Framework for Conceptual Knowledge Processing (http://www.tockit.org)
22. Becker, P., Hereth, J., Stumme, G. ToscanaJ: An Open Source Tool for Qualitative Data Analysis // Proc. Workshop FCAKDD of the 15th European Conference on Artificial Intelligence (ECAI-2002). Lyon, France, 2002.
23. Priss, U. FcaStone - FCA file format conversion and interoperability software, Conceptual Structures Tool Interoperability Workshop (CS-TIW), 2008.
24. Lahcen, B., Kwuida, L. Lattice Miner: A Tool for Concept Lattice Construction and Exploration // Suplementary Proceeding of International Conference on Formal Concept Analysis (ICFCA'10), 2010.
25. Borza, P.V., Sabou, O., Sacarea, C. OpenFCA, an open source formal concept analysis toolbox // Proc. of IEEE International Conference on Automation Quality and Testing Robotics (AQTR), 2010, pp. 1-5.
26. Szathmary, L. The Coron Data Mining Platform (http://coron.loria.fr)
27. Cubist Project (http://www.cubist-project.eu)
28. Neznanov A.A., Ilvovsky D.A., Kuznetsov S.O. FCART: A New FCA-based System for Data Analysis and Knowledge Discovery // Contributions to the 11th International Conference on Formal Concept Analysis, 2013. pp. 31-44.
29. Poelmans, J., Elzinga, P, Neznanov, A.A., Viaene S., Kuznetsov, S.O., Ignatov, D., Dedene, G. Concept Relation Discovery and Innovation Enabling Technology (CORDIET) // CEUR Workshop proceedings Vol. 757, Concept Discovery in Unstructured Data, 2011.
30. Neznanov, A., Ilvovsky, D., Parinov, A. Advancing FCA Workflow in FCART System for Knowledge Discovery in Quantitative Data // 2nd International Conference on Information Technology and Quantitative Management (ITQM-2014), Procedia Computer Science, 31, 2014, pp. 201-210.
31. Web data: Amazon movie reviews (http://snap.stanford.edu/data/web-Movies.html)

PRCA - A Parallel Relational Concept Analysis Framework

Ines Moosdorf[1], Adrian Paschke[1], Alexandru Todor[1], Jens Dietrich[2], Hans W. Guesgen[2]

[1] Freie Universitaet Berlin, Germany
{moosdorf, paschke, todor} AT inf.fu-berlin.de
[2] Massey University, New Zealand
{j.b.dietrich, h.w.guesgen} AT massey.ac.nz

Abstract. Relational Concept Analysis (RCA) extends standard Formal Concept Analysis (FCA) by taking relations between objects into account. The scalability of RCA learning on top of huge amounts of sensor data is a challenge for applications such as smart home system monitoring in ambient assisted living environments. One possible approach to improve scalability is to exploit the capabilities of modern parallel computing architectures such as multi-core CPUs. In this paper, we propose PRCA (Parallel Relational Concept Analysis), a novel framework for parallel relational concept learning. [1]

1 Introduction

In the next few years, the world population will be ageing dramatically: the percentage of people over 65 will grow to more than 25% and average life expectancy will increase to 75. This will have a particular impact on the health care systems since there will not be enough health care workers to adequately attend to all elderly people. Especially elderly people who suffer from cognitive impairment are known to remain independent for longer when living in their own home. Despite their cognitive shortfalls these people are still able to perform everyday activities like washing, grooming and eating. These activities are called Activities of Daily Living (ADLs) and it has been demonstrated that they will be retained for a longer period if the elderly people remain in their familiar environment. [1] The application scenario of the research described in this paper is in the field of smart home systems that support elderly cognitive impaired people to stay independently in their own houses as long as possible with just minimal support from health care services. A smart home system monitors inhabitants with unobtrusive sensors, identifies particular behaviors and notifies health workers if an abnormal behavior, such as taking medication in the middle of the night, occurs. Abnormal behaviour detection is a core feature of smart home systems.

[1] This colloborative work between the Massey University Smart Environments (MUSE) group and der Corporate Semantic Web (CSW) group at Freie Universitaet Berlin has been partially supported by the Royal Society of New Zealand and the BMBF in the User Guided Semantic Content Enrichment project and the "InnoProfile-Transfer Corporate Smart Content" project

Concepts of normal behaviour are learned from positive and negative training data. New behaviours are classified using the concepts of normal behaviours.

Formal concept analysis (FCA) is a simple yet powerful and elegant representational concept learning mechanism introduced in [2]. The explicit separation between intention and extension makes FCA an ideal platform for symbolic machine learning: training data represents concept extensions from which concept intentions are being inferred that can later be used as classifiers.

Relational Concept Analysis (RCA) was first proposed in [3]. It extends standard FCA by taking relations between objects into account.

Given the amount of data that has to be processed by modern applications, the scalability of learning is of particular concern. One approach to tackle this problem is to take advantage of parallel computing platforms (multicore CPUs, GPUs, cloud computing), and to parallelise learning algorithms. In this paper, we present PRCA (Parallel Relational Concept Analysis), a novel framework for parallel concept learning. PRCA is based on RCA [3, 6, 4] in order to improve the expressiveness of pure FCA, and uses multicore CPUs to improve scalability. We evaluate the accuracy of the learning algorithm and the performance gains on a set of benchmark data sets widely used in description logic learning, and compare results with existing description logic learners (DLLearner [2], PARCEL [3]). The results indicate that on most data sets, PRCA outperforms DL-based learners. PRCA also provides a wide range of configuration options that can be used to implement project specific heuristics.

2 Related Work

Our research is mainly based on the mathematical foundations of FCA as described in [2]. Standard FCA is restricted to data sets thata are either already represented as binary relations or that can be easily transformed into such a representation using method such as conceptual scaling [2]. We are not interested in "pure" FCA-based learning, but in learning from data sets that also contain binary relations between objects. These data sets cannot be transformed via conceptual scaling and hence cannot be processed by standard FCA algorithms.

Huchard et al. have proposed Relational Concept Analysis (RCA) [3], a method that extends FCA for the purpose of taking relations between objects into account. PRCA is based on ideas from the relational data model, relational scaling and iterative relational property generation. RCA aims to generate complete lattices of data sets. This leads to scalability issues since the size of the concept lattices grows rapidly with the number of relationships between contexts. Our idea differs from RCA in that we do not focus on complete lattice creation but on building lattices of selected properties that are good for dividing positive from negative examples. Furthermore, in PRCA we try to address the scalability problem by using concurrent computing. In [5], Kuznetsov proposes the symbolic

[2] http://aksw.org/Projects/DLLearner.html
[3] http://code.google.com/p/parcel-2013/

machine learning method JSM in terms of FCA learning from positive and negative examples. This method consists of two parts, learning hypotheses from positive and negative examples and a classification of undetermined examples by the learned hypotheses. This method is adapted and employed in the PRCA framework. Our hypotheses generation algorithm differs from the approach presented in [5], in that it is not based on two separate formal contexts for positive and negative examples, but on a combined formal context which is processed in parallel. When a concept extension contains only positive examples, then its intention is regarded as positive hypothesis. When a concept extension contains only negative examples, then its intention is regarded as negative hypothesis. The difference between the RCA and PRCA approach is that after creating the formal context PRCA generates all possible combinations in the relational scaling step, instead of only relations to concepts as in RCA. In PRCA the relational properties are combinations of relational and basic information of the relational context. As a result PRCA creates more relational properties then RCA, because the concepts that already pre-group the data are not used. Although this seems to be a disadvantage on the first glance, it is necessary for the parallelization of the algorithm. Otherwise, there would be step dependency as in RCA.

3 Parallel Relational Concept Analysis Framework

The main approach of the Parallel Relational Concept Analysis (PRCA) framework is the parallelization of the scaling step of object-object relations to relational properties and the integration of basic and relational properties into one concept lattice. The aim is to learn positive and negative hypotheses. PRCA does not generate a complete lattice with all relational properties, but only finds suitable relational properties that are good for dividing the positive from the negative examples. Therefore, the PRCA framework iteratively generates new relational properties from the relational information given in the data set and combines them with the basic properties in one lattice until sufficient hypotheses are found.

3.1 Basic Steps

Figure 1 depicts the basic steps of the PRCA framework. The input of the framework are *relational contexts*. We define a relational context C as a pair (K, R) consisting of a set of formal contexts $K = \{K_i\}$, whereby each context $K_i = (O_i, P_i, I_i)$ has objects O_i, properties P_i and a relationship I_i between these objects and properties; and object-object relations $R = \{R_i\}$, with $R_i \subseteq O_1^i \times O_2^i$, associating objects from two contexts.

Each basic relation R_j has a source (relational) context C_i and a target (relational) context C_k, both source and target can be identical. The main (relational) context is a learning problem with multiple contexts. One context is the main context. This is the context that contains the positive, negative (and unknown) labelled objects of the learning problem.

The learning algorithm consists of several steps.

In the *Generation of Relations* step, relations are generated based on the basic relations and properties of the relational contexts. The generator yields basic relations as well as new *composite relations*. A composite relation is the result of composing two relations or one relation and an additional post condition. Different generation operators like joins, intersections and conditional joins exist. For instance, the relation join is defined as $R_{j.k} := R_j.R_k$. Applying a postcondition creates a new relation by applying filters based on properties in the target context.

In the following *Relational Scaling* step, these relations are then scaled to *relational properties*. Different Scaling operators exist: existential, universal and cardinality restricted. There are also different scaling directions: left and right direction (has/is).

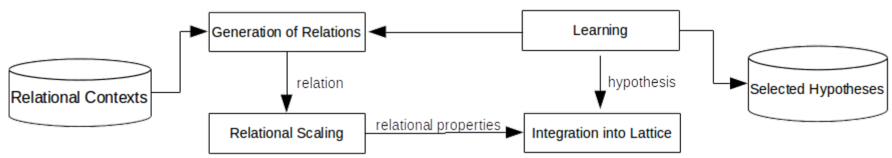


Fig. 1. The basic steps of the Parallel Relational Concept Analysis (PRCA) framework

In the *Integration into Lattice* step, the relational properties are integrated with the basic properties into one lattice to check for new positive hypotheses, i.e. intentions of concepts that contain only positive examples)=, All new formal concept intentions being hypotheses are selected and stored.

In the *Learning* step, it is checked if all positive and negative examples are covered by at least one positive, or negative hypothesis, respectively. If all positive examples of the main context are covered by at least one hypothesis, the best hypotheses are selected and returned as result of the learning process.

3.2 Components of the PRCA Framework

Figure 2 shows the realization of the basic steps of the PRCA framework. Input data are one or more *relational contexts*. In addition to formal contexts and the set of basic relations, each relational context contains a set for storing relational properties that are scaled during relational scaling step. *Uncovered positive examples* is an agenda containing all positive examples. When a new positive hypothesis is found , the examples covered by this hypothesis will be removed from the agenda. *Hypotheses* is a container shared by all parallel running workers to collect the learned hypotheses. The *global property pool* contains all basic properties and relational properties which are relevant for building the lattice. The parallel running workers scale new relational properties and add them to the global property pool. The *relation generator* generates new relations using the operations described above.

The steps *Relational Scaling* and *Integration into Lattice* are realized by parallel running *worker* threads. In each *working step*, the worker requests a new relation from the relation generator, scales new relational properties and integrates them into one lattice with the basic properties and previously scaled properties.

Step by step the lattice is extended by new formal concepts. When a new formal concept covers only positive examples the worker updates the agenda of *uncovered positive examples* and stores the intention of the concept as new hypothesis in its *local hypotheses pool*.

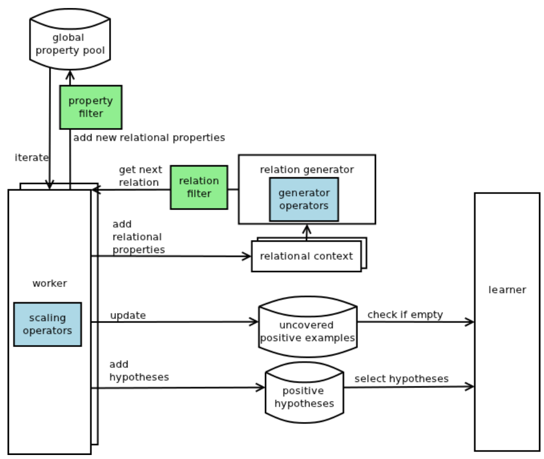


Fig. 2. Components and Configuration possibilities of the PRCA framework

The worker adds the new relational properties to the *global property pool* and to the *relational property pool* of the respective relational context.

The *Learning* step is done by the *learner*. The learning is finished when all examples have been removed from the agenda, i.e., all examples are covered. Then, each worker adds its found hypotheses to the global hypotheses pool. Then, the learner selects the best hypotheses to create the result of the learning process. A set cover algorithm is used for this purpose. By default, we use a simple greedy algorithm that selects hypotheses covering the most (not yet covered) examples. The use of other algorithms is possible as well.

3.3 Configuration

The purpose of the framework is to provide a tool for developing and evaluating different strategies of RCA learning. The framework offers several variability points and configuration options that can be combined. In particular, this includes operators to compose relations, filters and set coverage algorithms to select hypotheses.

Scaling operators define the type of the relational scaling. Multiple scaling operators can be defined at the same time. Each worker applies all the defined scaling operators to the given relation.

Property filter: The workers add their newly scaled relational properties to the global property pool. However, not all relational properties are relevant. To reduce the number of irrelevant properties, the configured *property filter* controls which properties are added to the global property pool.

A *relation filter* filters the generated relations. When the relation generator is requested for a next relation it will only return relations that pass the filter.

4 Evaluation

4.1 Methodology

The evaluation is conducted with a ten 10 fold cross validation. The evaluation metrics are:

Learning Time: duration from context to selected hypotheses.

The hypotheses learned by the prototypes are used to classify unknown examples. To determine the quality of the learned hypotheses their *correctness, completeness* and *accuracy* are measured. Therefore a set of positive and negative labelled examples is used. Each example of the data set is classified by the learned hypotheses. The results of the classification are compared to the original labels of the examples. *Correctness* determines the ability of the learned hypotheses to classify negative examples as negative. *Completeness* determines the ability of the learned hypotheses to classify positive examples as positive. *Accuracy* combines correctness and completeness. It determines the ability of the learned hypotheses to classify undetermined examples correctly.

$$correctness = \frac{|negative\ examples\ classified\ as\ negative|}{|all\ negative\ examples|}$$

$$completeness = \frac{|positive\ examples\ classified\ as\ positive|}{|all\ positive\ examples|}$$

$$accuracy =$$

$$\frac{|negative\ examples\ classified\ as\ negative|\ +\ |positive\ examples\ classified\ as\ positive|}{|all\ examples|}$$

Definition length: A further quality property of the learned hypotheses is their length. A shorter hypothesis is regarded as better than a longer one describing the same objects.

- property length: To compute the length of a property the containing relations, properties and scaling operators are counted, e.g.,
  - female = 1
  - exists has sibling (exists has child (female)) = 5
- hypotheses length: The hypothesis is the conjunction of all its properties. The *hypothesis length* is influenced by the *number of properties per hypothesis* and the *length of the properties*. It is the sum of the length of all its properties plus n-1 "`AND`s" between n properties. For example, the hypothesis {`female, old`} consists of two hypotheses with length one. Its *hypothesis length* is three (`female AND old`).

– definition length:The *definition length* is the sum of the length of all its hypotheses plus the n-1 `OR`s between n hypotheses. For example, in the learning problem Uncle of the Family data set the learned definition consists of two hypotheses: {`male, exists has sibling.child`} `OR`
{`male, exists has married.sibling.child`}.
It has a definition length of twelve (4 + 1 *(AND)* + 1 (OR) + 5 + 1 *(AND)*)

4.2 Data sets

Data sets used for evaluation are the family data set: machine learning data set from DL learner repository [4] and the straight data set: a (randomly) generated data set.

Data Sets	**Examples**	**Positive**	**Negative**	**Relational Contexts**	**Basic Properties**	**Relations**
Uncle (Family)	202	38	38	1	2	4
Cousin (Family)	202	71	71	1	2	4
Aunt (Family)	202	41	41	1	2	4
Grandson (Family)	202	30	30	1	2	4
Grand mother (Family)	202	17	16	1	2	4
Straight200	200	100	100	2	Deck 0, Card 17	Deck 1, Card 3
Straight800	800	400	400	2	Deck 0, Card 17	Deck 1, Card 3

Table 1. Summary of the data sets used for evaluation.

4.3 Evaluation Results

– First experiments (family benchmark): configuration PRCA I: minimal configuration to solve family problems: existential scaling (left), join, all-filter
– Second experiments (family benchmark): configuration PRCA II: more expressive configuration: join, both post conditional join, existential scaling (left), universal scaling (left)
– Third experiments (poker benchmark): configuration PRCA III: trade-off high accuracy and short definition length: all-filter, intersection, join, bloom relation filter, existential scaling (left)
– Fourth experiments (poker benchmark): configuration PRCA IV: trade-off learning time: 80% uncovered positive filter, intersection, join, bloom relation filter, existential scaling (left)

Family data sets:

– The learning time is faster with minimal configuration (PRCA I) than with more expressive configuration.

[4] `http://sourceforge.net/p/dl-learner/code/HEAD/tree/trunk/examples/family/`

	Learning Time (ms)	Testing Accuracy (%)	Definition Length	No. of hypotheses
Family data set - Aunt				
PRCA I	24.5 ± 0.9	100 ± 0	12 ± 0	2 ± 0
PRCA II	59.9 ± 8.5	100 ± 0	15 ± 0	2 ± 0
DL Learner	134.3 ± 29.8	99.5 ± 0.6	21.9 ± 2	2 ± 0
Family data set - Grandgrandmother				
PRCA I	24.3 ± 1.3	100 ± 0	6 ± 0	1 ± 0
PRCA II	58.4 ± 6.2	99 ± 2.4	13.1 ± 2	1.6 ± 0.2
DL Learner	68 ± 4.4	82.3 ± 6.3	26 ± 3	1.7 ± 0.3
Family data set - Grandson				
PRCA I	19.1 ± 0.5	100 ± 0	5 ± 0	1 ± 0
PRCA II	19.8 ± 1.1	99.6 ± 0.8	6 ± 0.1	1 ± 0
DL Learner	23.8 ± 4.5	99.4 ± 0.7	10.1 ± 2.4	1.1 ± 0.2
Family data set - Uncle				
PRCA I	25.5 ± 1.1	99.3 ± 0.8	12 ± 0	2 ± 0
PRCA II	65.2 ± 11.3	99 ± 0.8	16.7 ± 0.3	2 ± 0
DL Learner	140.2 ± 13.9	97.9 ± 1.7	23.5 ± 3.7	2.1 ± 0.2
Family data set - Cousin				
PRCA I	37.8 ± 1.5	100 ± 0	10 ± 0	2 ± 0
PRCA II	1727.2 ± 489	99.7 ± 0.5	17.8 ± 3	2.1 ± 0.2
DL Learner	346 ± 24.7	99.3 ± 0.8	23.3 ± 7	2.1 ± 0.1
Poker data set - Straight 200				
PRCA III	2196.5 ± 131.8	100 ± 0	10 ± 0	1 ± 0
PRCA IV	1061.9 ± 55.5	99.1 ± 0.1	26.3 ± 1.5	1 ± 0
DL Learner	1596.6 ± 30.7	73.8 ± 2.5	466.2 ± 19.4	16.7 ± 0.6
Poker data set - Straight 800				
PRCA III	9504.3 ± 510.4	100± 0	10 ± 0	1 ± 0
PRCA IV	2609.5 ± 144.8	99.95 ± 06	33.8 ± 0.2	1 ± 0
DL Learner	runs out of memory	-	-	-

Table 2. Experiment result summary: PRCA and DL-Learner with ParCEL-Ex on several Family and Straight learning problems. The values are the averages and standard deviations of ten 10-fold cross-validations.

- The additional generator operators lead to the generation of more irrelevant relations, that need to be scaled and integrated into the lattice. Furthermore, the additional scaling operators and the less restrictive property filter lead to more properties that need to be integrated into the lattice as well. Hence bigger lattices are generated. The generation and scaling of more irrelevant relations and the generation of lattices with more concepts increases the learning time when PRCA is run with a more expressive configuration.
- PRCA achieves high testing accuracy on all learning problems, but not 100% because the data sets are small and the learned definitions are over fitted to the training data set.
- A general problem of FCA (for our purpose) is that it generates most specific descriptions instead of most general ones. According to the definition of*formal concept* a concept consists of **all** properties common to all objects in the concept extension (because FCA is based on closure operator). (this happens on small data sets, but may happen on noisy data sets as well)
- PRCA with minimal configuration outperforms DL Learner regarding learning time while achieving the same testing accuracy values.
- definition length: the definitions of the DL Learner tend to be a bit longer than those of PRCA because DL Learner combines partial definitions and counter partial definitions, e.g., one partial definition for the uncle learning problem is `not female and exists sibling.exists child.top`.

- DL Learner and PRCA find the same number of partial definitions/hypotheses (hypotheses in PRCA correspond to partial definitions in DL Learner).

Straight data set:

- The DL-Learner has troubles with learning these problems. On Straight200 its testing accuracy is only 73.9% which is useless for practical application and on Straight800 it runs out of memory. The definition length value and analysis of the result files reveal that ParCEL-Ex learns specific partial definitions whereas PRCA learns one generic hypothesis and achieves high accuracy (99-100%) in all test runs.
- The configurations for the straight data sets are trade-offs between learning time.
- With PRCA III the hypothesis with minimal length is learned and 100% testing accuracy is gained, e.g.,
  `exists has [[card+[card+[card.nextRank+card].nextRank].nextRank].nextRank+card]`
- However, learning times are large on both data sets: more than 2 seconds on Straight200 and more than 9 seconds on Straight800.
- With the weaker *80% uncovered positives filter* learning time is reduced on both data sets: the learning duration of the Staight200 learning problem becomes two times faster and the learning duration of Straight800 becomes 3.6 times faster. The trade-offs are that the testing accuracy gets less (but is still more than 99%) and the definition length becomes 2.6 times longer on Straight200 and 3.4 times longer on Straight800.
- The definition length values and result file analysis reveal that when the filter gets weaker the hypotheses consist of more properties. These properties are shorter than in the PRCA III hypothesis, but describe the straight only partially. For instance, the hypothesis describes a sequence of four cards of sequential rank and three cards with two of sequential rank and a third of the “next-next-next” rank. This leads to worse testing accuracy. The reason is the small training data set: the hypothesis covers all positive examples and not any negative example.

In summary,

- the benchmark shows that PRCA outperforms DL Learner on the used data sets (when run with appropriate configuration). It is faster with always higher accuracy.
- the experiments revealed the general problem of FCA. It generates most specific descriptions instead of most general ones. We tried to reduce the irrelevant properties by property filters. However, hypotheses still contain irrelevant properties. Further work may investigate property reduction during the final hypotheses selection in the learner.
- PRCA find short definitions because DL Learner (ParCEL-Ex) combines counter partial definitions.
- for generalizing the results evaluations on bigger data sets (more examples, more relations, more properties, more complex definition, noisy data) need to be conducted

– the slowing down on more expressive configurations and on learning problems with long hypotheses indicates that the relation generator needs to be improved, e.g., by atm breadth first search → heuristic based search, relation filter mechanism for filtering duplicate relations
– for improving lattice creation (the more properties are in the pool the bigger the lattice the more time is needed to create the lattice) we apply distributed lattice creation.

5 Discussion

Our application scenario is in learning positive concepts of normality for detecting abnormal (i.e. negative) behaviours in the context of smart monitoring environments in ambient assisted living. Our event data sets don't only contain object-property relations but also more complex information relating objects to other objects which have properties. We therefore extended standard FCA-based learning on the basis of RCA for parallel learning on top of data with object-object relations. Due to the amount of data, high scalability of the learning method is relevant and the proposed parallel learner addresses this problem. The proposed approach is configurable and extensible which allows us to further study and evaluate relational concept analysis in different parallel configurations. We conducted experiments that have shown that we can handle data sets with one or multiple relational contexts and that PRCA outperforms DL Learner on the used data sets: PRCA finds a solution on straight data sets, where DL Learner doesn't find a correct solution. On data sets where both find solutions PRCA learns the definitions faster and achieves similar results for testing accuracy.

References

1. R.S. Bucks and J. Haworth. Bristol activities of daily living scale: a critical evaluation. *Expert Rev Neurother*, 2(5):669–76, 2002.
2. B. Ganter and R. Wille. *Formal Concept Analysis: Mathematical Foundations*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 1st edition, 1997.
3. M. Huchard, A. Napoli, M.R. Hacene, and P. Valtchev. Mining Description Logics Concepts With Relational Concept Analysis. In P. Bertrand P. Brito, G. Cucumel and F. de Carvalho, editors, *Selected Contributions in Data Analysis and Classification*, Studies in Classification, Data Analysis, and Knowledge Organization, pages 259–270. Springer Berlin Heidelberg, August 2007.
4. M. Huchard, A. Napoli, and P. Valtchev M.R. Hacene. Relational concept analysis - a gentle introduction. http://hal.archives-ouvertes.fr/docs/00/61/62/75/PDF/rca_icfca2011.pdf, May 2011.
5. S.O. Kuznetsov. Machine learning on the basis of formal concept analysis. *Automation and Remote Control*, 62(10):15431564, 2001.
6. Mohamed Rouane-Hacene, Marianne Huchard, Amedeo Napoli, and Petko Valtchev. Relational concept analysis: mining concept lattices from multi-relational data. *Annals of Mathematics and Artificial Intelligence*, 67(1):81–108, 2013.

Concept Building with Non-Hierarchical Relations Extracted from Text – Comparing a Purely Syntactical Approach to a Semantic one

Sílvia Maria Wanderley Moraes, Vera Lúcia Strube de Lima and Luis Otávio Furquim

Pontifícia Universidade Católica do Rio Grande do Sul (PUCRS)
Faculdade de Informática
Av. Ipiranga, 6681 Prédio 32, Porto Alegre, RS, Brazil
silvia.moraes@pucrs.br, vera.strube@pucrs.br, luisfurquim@gmail.com

Abstract. In this paper, we introduce Semantic Lattice (SemLat), a method that allows the construction of concept lattices from lexical-semantic information extracted from PropBank-style labelled texts. We apply SemLat to Tourism and Finances domain texts from Wikicorpus 1.0 through case studies that are examined in detail. We compare conceptual structures generated by SemLat, that makes use of semantic relations, to structures generated from purely syntactic relations. We intrinsically evaluate the structures using a semantic-similarity based structural measure. We also analyse, in a qualitative approach, the contribution of semantic roles in concept formation. We claim that conceptual structures generated by SemLat produce richer concepts as they provide intentional descriptions that are more informative, from a semantic point of view.

Keywords: Formal Concept Analysis, Semantic Role, Concept Lattices.

1 Introduction

Conceptual structures such as terminologies, thesauri, taxonomies and ontologies are important resources for information systems. Since building and maintaining such structures is costly, automatic and semi-automatic approaches have been proposed to minimize the effort of extracting concepts and semantic relations from texts. We are interested in exploring the potential of the semantic roles in the learning of conceptual structures. A semantic role expresses the meaning of an argument in a situation described by the verb in a sentence. With the use of semantic roles, we can identify, for example, the agentive entity of an action, even if it appears in diverse syntactical positions through the text. In this paper, we present the Semantic Lattice (SemLat) - a simple method to generate concept lattices from semantic relations extracted from texts, exploring the benefits of Formal Concept Analysis (FCA) as a conceptual clustering method. We intrinsically evaluate the conceptual lattices built, using a structural measure based on semantic similarity. We qualitatively analyse the contribution of semantic roles in the formation of concepts. Results show that conceptual structures created by SemLat generate richer concepts, as they provide intentional descriptions that are more informative, from a semantic point of view.

This paper is organized in 6 sections. In Section 2 we study related work. Section 3 shortly introduces semantic roles and FCA. Section 4 briefly describes the SemLat method. Section 5 presents the studies concerning SemLat and Section 6 brings our conclusions.

2 Related Work

The idea of combining the FCA method with semantic roles is not new. Kamphuis and Sarboin [1] propose to represent a sentence in natural language, associating FCA to semantic roles. They deal with two types of linguistic relations: minor (nouns to adjectives and adverbs) and major (verbs to nouns). Differently from that work, we extract relations from linguistically tagged texts namely the major ones. Rudolf Wille [2] also presents examples of FCA structures combined with semantic roles. He combines conceptual graphs with FCA structures, aiming the formalization of useful logic to representation and processing. There are no comments, in his work, on the processing of information present in the conceptual graphs, so we understand that neither the construction of these graphs nor their mapping into FCA structures, were performed automatically. Our study deals with the automatic extraction of information from texts (to generate representation structures) and we analyse the limits of our approach. The FCA method was aheady combined with semantic roles, as in [3], where efforts turn to the linguistic analysis as a purpose for representing FrameNet through concept lattices. Distinct from our work, the authors do not use FCA as a support to build ontological structures from texts. Instead, we use textual information and PropBank annotation to identify the roles. Although the approaches in [1,2,3] seemed promising at the time they have been proposed, they were little explored probably due to the difficulties with the text annotation process, since the appearance of automatic semantic role annotators is more recent. Even with thorough literature review, we did not find, to date, studies that explore the use of semantic roles in conjunction with the FCA method to support construction of ontological structures from texts. We address this issue in our research.

3 Semantic Roles and FCA

Semantics roles are “roles within the situation described by a sentence” [4]. Although there is no consensus on a single list of semantic roles, some are widely accepted [5] such as: Agent, Patient, Instrument, Theme, Source and Destination. The barrier regarding the definition of roles has been circumvented by assigning numerical labels (A0, A1, A2, ...) to the arguments of the verbs. This is the case for PropBank[1] corpus, which has been extensively used to train semantic role taggers for the English language. The F-EXT-WS tool used to tag

[1] http://www.cis.upenn.edu/~ace

the corpora in the present study, also adopts these labels [6]. For the English language, it provides Part-of-Speech (POS) tagging, syntactical annotation and semantic roles tagging. F-EXT-WS uses the tags defined for PennTreeBank [2].

FCA was introduced by Rudolf Wille in the 80's as a method for data analysis [2]. A key element in FCA is the formal context, characterized by the triple (G, M, I), where: G is the set of domain entities, called formal objects; M consists of the features of these entities, their formal attributes; and I is the binary relation on $G \times M$, called the incidence relation, which associates a formal object to its attributes. The formal concepts are built from the formal context. A formal concept is determined by the pair (O, A) if and only if $O \subset G$ and $A \subset M$. Once the concepts have already been defined, the concept lattice is created [7].

4 The SemLat Method

The SemLat method is the result of several studies, including Relational Concept Analysis [8], in the interest of how to include semantic roles in lattices [9,10]. SemLat comprises 3 stages shown in Fig. 1. The SemLat input is a corpus annotated with lexical-semantic information, lemmatized. From this corpus we create the conceptual structure.

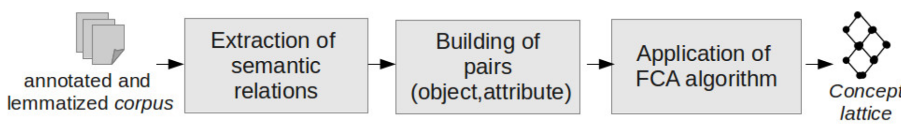


Fig. 1. SemLat stages

The 'Extraction of semantics relations' stage consists of the building of tuples containing, for a certain verb, its arguments and the semantic roles associated with these arguments. Aiming to build a conceptual structure, relevant noun phrases are extracted from the arguments. The steps to build tuples are:

1. To analyze the sentences, identifying and extracting verbs and respective arguments and associated semantic roles.
2. To identify the noun phrases in the verb arguments discarding those formed by proper nouns (as we have not included an instance level in the ontological structure).
3. To form tuples, using information extracted from sentences in steps 1 and 2. Each tuple must contain noun phrases and their correspondent semantic roles. Tuples are in the following format: (np_1,sr_1,np_2,sr_2) where np_i and sr_i correspond, respectively, to the noun phrase and its semantic role.

Let's consider the following sentence from PropBank: "The financial-services company will pay 0.82 share for each Williams share." After annotating (Fig. 2) the sentence with the use of F-EXT-WS, we are able to extract necessary lexical-semantic information from this sentence and complete the tuple (company, A0, share, A1).

[2] http://www.cis.upenn.edu/~treebank/

(A0 (DT The) (NNS financial-services) (NN company)) (MD will) (V (VB pay)) (A1 (CD 0.82) (NN share) (IN for) (DT each) (NNP Williams) (NN share)).

Fig. 2. Sentence annotated with F-EXT-WS

The second stage aims to produce the object-attribute pairs that will give origin to the FCA formal context. From each tuple (np_1,sr_1,np_2,sr_2) extracted from the texts, two object-attribute pairs are created: (np_1,sr_1_of_np_2) and (np_2,sr_2_of_np_1). So, from the tuple (company, A0, share, A1) the pairs (company, A0_of_share) and (share, A1_of_company) are created. Frequently, A0 corresponds to Agent and A1 to Patient. With the use of semantic roles, we can better determine the relationship between the nouns: company is an agent of share, and share is a patient of company. As many pairs can be generated, in order to avoid an excessively sparse formal context, we group concepts, as described in [9]. The pairs created, the formal context can be built. SemLat's last stage consists of the generation of the conceptual structure (Fig. 3). In order to accomplish this task, FCA algorithms, such as Bordat [11], can be used. Another alternative is to use a specific tool to generate lattices such as Concept Expert[3] 1.3 .

5 Studies concerning SEMLAT

We compare structures built with the SemLat method (Fig. 3b) to those built with FCA exclusively based on the syntactic relations between verbs and their arguments, as proposed by Cimiano in [12] (Fig. 3a). In order to accomplish this task, we use Wikicorpus[4] 1.0 comprising Wikipedia texts. We randomly took from Wikicorpus 322 texts of the Finances domain and 284 texts of the Tourism domain. These subsets were named correspondingly, WikiFinance and WikiTourism. Both corpora were annotated with lexical-semantic information using F-EXT-WS. We lemmatized nouns present in the identified noun phrases with TreeTagger[5]. To analyse the contribution brought with the semantic roles in the formal concepts formation, we outlined two case studies:

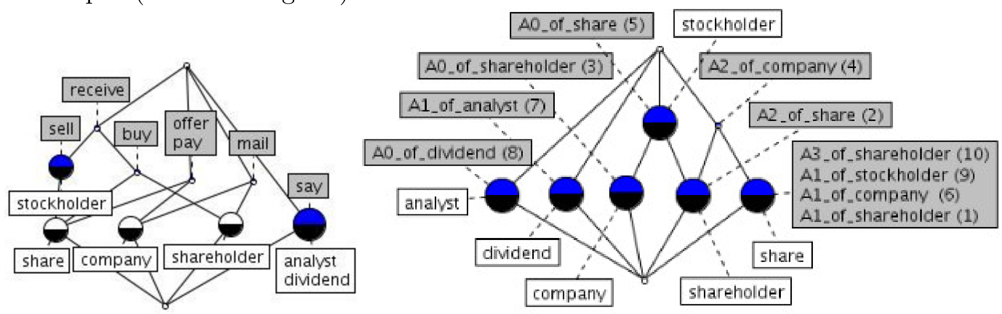
- case (np, v): describes syntactical relations of the type verb-argument.
- case (np, sr_of_np): describes semantic relations obtained with SemLat.

With these two studies and using WikiFinance and WikiTourism corpora, we produced four conceptual structures to be examined (only relations with a minimum frequency of 2 were considered):

- TourismFCA (np, v): from case (np, v) for the WikiTourism corpus.
- TourismFCA (np, sr_of_np): from case (np, sr_of_np) for the WikiTourism corpus.

[3] http://sourceforge.net/projects/conexp
[4] http://nlp.lsi.upc.edu/wikicorpus/
[5] http://www.ims.uni-stuttgart.de/projekte/corplex/TreeTagger/

– FinanceFCA (np, v): from case (np, v) for the WikiFinance corpus. A subset of this structure is shown in Fig. 3a.
– FinanceFCA (np, sr_of_np): from case (np, sr_of_np) for the WikiFinance corpus (subset in Fig. 3b).



(a) (b)

Fig. 3. Syntactic and semantic lattices

Table 1 contains information on these structures, including number of objects, attributes, concepts and edges. We notice that case (np, sr_of_np) has in average 4 times more attributes than case (np, v). This number of attributes was already expected, since in case (np, sr_of_np) the attributes are much more specific. This specificity increases in around 7% of the concepts in the Finances domain and in approximately 30% in the Tourism domain. This fact may be related to the scope of the texts in each domain.

Table 1. Information on the conceptual structures generated

FCA	case	#objects	#attributes	#concepts	#edges
Finance	(np,v)	631	237	760	2055
	(np,sr_of_np)	631	1018	819	1919
Tourism	(np,v)	383	121	239	529
	(np,sr_of_np)	383	535	343	633

In a subjective and shallow analysis, we perceive the Tourism domain texts are more restricted than the Finances ones. While Tourism texts mostly approach subjects related to attractions, texts from Finances include descriptions on the key terms in the domain. In the following sections we study the contribution of semantic roles in the formation of concepts.

5.1 Lexical cohesion

Although extensively studied, the evaluation of conceptual structures is still an issue to be further investigated. When we evaluate FCA-based structures, difficulties increase due to the fact that this investigation is recent. We found two measures ideally appliable to this evaluation [13,14] both comparing FCA structures regarding the objects and the formal attributes of their concepts. As the formal concepts generated from the case studies were not equivalently configured (they had different attributes), we could not apply these measures satisfactorily. So we focused our analysis on formal objects. The evaluation of these lattices was based on the structural Semantic Similarity Measure (SSM) [15]. SSM indicates how close are the concepts that match (exactly or partially) the search terms in an ontology. In the present study, SSM became a sort of lexical cohesion measure, as it was applied to the objects of each formal concept from the FCA structure. Typically, synonymy, hypernymy and meronymy are considered, when calculating cohesion. In order to obtain such cohesion value, as recommended in [15], we used in the SSM estimation the measure defined by Wu and Palmer [16] which takes semantic relations from an ontological structure to calculate the semantic distance between words. Equation (1) indicates the average lexical cohesion among the N concepts in a FCA structure, regarding a conceptual structure E.

$$SSM_E = \frac{1}{N} \sum_{i=1}^{N} ssm_i \qquad (1)$$

As detailed in Equation (2), ssm_i computes the similarity in the set of objects G of a concept i in a FCA structure, using Wu e Palmer (wup) measure. In case the cardinality of G is 1, ssm_i is zero.

$$ssm_i = \begin{cases} \dfrac{1}{|G_i|} \displaystyle\sum_{j=1}^{|G_i|-1} \sum_{k=j+1}^{|G_i|} wup_E(o_j, o_k) & for\ |G_i| > 1\ and\ o_j, o_k \in G_i \\ 0 & o/w \end{cases} \qquad (2)$$

Besides WordNet[6], we applied SSM over domain ontologies: LSDIS_Finance[7] and Finance[8] for the Finances domain, and Travel[9] and TGPROTON[10] for the Tourism domain. Although the extension and richness in WordNet relations, these relations are mostly general and do not refer to a specific domain. We believe that the measure proposed by Wu and Palmer [16], applied to the WordNet structure, might not fully capture the expected semantic relations so producing less expressive values. Besides, even if domain ontologies have a more concise concepts set (regarding its domain), it is more frequent to find n-gram labelled concepts ($n > 1$) as for the present studies. So, it is possible to assert that the relations among concepts are domain relations. These points may conduct to more significant results, from a semantic point of view, when concerning the quality of the clusters of concepts. Table 2 shows the results obtained from the application of SSM. In this table, W, F, L, TG and T correspond to the lexical resources used: WordNet, LSDIS_Finance, Finance, TGPROTON and Travel, respectively. As we imagined, SSM showed a low cohesion for both domains when using WordNet. As we expected, the domain ontologies have a cohesion

[6] http://wordnet.princeton.edu/
[7] http://lsdis.cs.uga.edu/projects/meteor-s/wsdl-s/ontologies/LSDIS_Finance.owl
[8] http://www.fadyart.com/ontologies/data/Finance.owl
[9] http://protege.cim3.net/file/pub/ontologies/travel/travel.owl
[10] http://goodoldai.org/ns/tgproton.owl

distinct from that found in the texts we used. In this case, cohesion values were low because less than 10% of the objects in concepts from a lattice were present in the ontologies. For the Tourism domain, we believe the variety of the texts was the main reason for the low matching. The absence of non-hierarchical relations in the selected ontologies caused some difficulties to the evaluation as well. As the semantic roles should express non-hierarchical relations, the cohesion of these relations were not computed in the evaluation results. As a next step we performed a qualitative analysis.

Table 2. SSM application results

	Finance			Tourism		
case	**SSM**$_w$	**SSM**$_F$	**SSM**$_L$	**SSM**$_w$	**SSM**$_{TG}$	**SSM**$_T$
(np,v)	0.33	0.33	0.27	0.18	0.05	0.02
(np,sr_of_np)	0.20	0.21	0.16	0.09	0.01	0.01

5.2 Qualitative Analysis

In this section we address, from a qualitative perspective, the importance of semantic roles in the formation of the formal concepts. Features inherent to semantic roles may help distinguish, classify and, essentially, better associate the elements extracted from texts. To illustrate this analysis, we used a subset from FinanceFCA (*np*, *sr*_of_*np*) and FinanceFCA (*np*, *v*) lattices. These subsets are those presented in Fig. 3. We perceived that the semantic roles caused the generation of an extra concept. The nouns *analyst* and *dividend* were not clustered in a same concept. However, the relation between them was not lost. In the structure obtained from case (*np*, *sr*_of_*np*) from Fig. 3b, transversal relations appear as attributes. The object *analyst* is defined as A0_of_dividend, meaning that it is the Agent of *dividend*. And the object *dividend* is A1_of_analyst, its patient. In both cases, the structures produce a concept for "share". In case (*np*, *sr*_of_*np*) we get to more clearly interpret the relation between *share* and the other elements of the domain. We can notice that share is usually patient (A1) of *stockholder*, *company* e *shareholder*. The *stockholder* concept showed to be a superconcept in both structures but, in case (*np*, *sr*_of_*np*), *share* was not its subconcept. This relation was expressed in the attributes. In case (*np* ,*sr*_of_*np*), *stockholder* as well as *company* and *shareholder*, its subconcepts, are agents (A0) of *share*. From this analysis we noticed that, even if the semantic roles make the concepts more specific, they are much more informative than the verbs. The concepts generated from case (*np*, *sr*_of_*np*) are semantically richer, from an intentional point of view, than those from case (*np*,*v*).

6 Conclusions

In this paper we depicted the SemLat method, which allows to build concepts based on semantic roles, using FCA as a conceptual clustering method. We then investigated the contribution of SemLat in the formation of concepts. From a structural and lexical point of view, it is still difficult to objectively evaluate the contribution of semantic roles in the building of formal concepts. The cohesion computed by SSM for the Tourism and Finances domains was inconclusive. From a qualitative point of view, we perceived semantically richer formal concepts. The inclusion of semantic roles in the formal attributes improved the intentional description of concepts. We are interested in the extrinsic evaluation of the concept lattices generated by SemLat. Presently, we are analysing the contribution of these structures in the text categorization task. Work on more appropriate methods for the evaluation of ontological structures is also important for future directions of the present study.

References

1. Kamphuis, V., Sarbo, J.: Natural language concept analysis. In: NeMLaP3/CoNLL98, ACL, Yellow Book (1998) 205–214
2. Wille, R.: Conceptual graphs and formal concept analysis. In: ICCS '97, London, UK, Springer-Verlag (1997) 290–303
3. Valverde-Albacete, F.J., Peláez-Moreno, C.: Galois connections between semimodules and applications in data mining. In: ICFCA. (2007) 181–196
4. Yule, G.: The Study of Language. Cambridge University Press, Cambridge (1996)
5. Jurafsky, D., Martin, J.H.: Speech and Language Processing: An Introduction to Natural Language Processing, Computational Lingustics, and Speech Recognition. Artificial Intelligence. Prentice Hall (2008)
6. Fernandes, E.R., Milidiu, R.L., Santos, C.N.: Portuguese language processing service. In: 18th International World Wide Web Conference. (2009)
7. Stumme, G., Darmstadt, T.H., Mathematik, F.: Exploration tools in formal concept analysis. In: Proc. OSDA 95. Studies in Classification, Data Analysis, and Knowledge Organization 8, Springer (1995) 31–44
8. Priss, U.: Relational Concept Analysis: Semantic Structures in Dictionaries and Lexical Databases. PhD thesis, Darmstadt, Aachen (1998)
9. Moraes, S.M.W., Lima, V.L.S.: Combining formal concept analysis and semantic information for building ontological structures from texts : an exploratory study. In: LREC'12, Istanbul, Turkey (2012)
10. Moraes, S.M.W.: Construção de Estruturas Ontológicas a partir de textos: um estudo baseado no método Formal Concept Analysis e em papéis semânticos. PhD thesis, Faculdade de Informática, PUCRS, Brazil (2012)
11. Fu, H., Nguifo, E.M.: Lattice algorithms for data mining. In: Revue ISI (Ingénierie des Systèmes d'Information). Volume 9., Edition Hermes (2004) 109–132
12. Cimiano, P., Hotho, A., Staab, S.: Learning concept hierarchies from text corpora using Formal Concept Analysis. JAIR **24** (2005) 305–339
13. Alqadash, F., Bhatnagar, R.: "Similarity Measures in Formal Concept Analysis". Annals of Mathematics and Artificial Intelligence **61** (2011) 245–256
14. Formica, A.: Concept similarity in formal concept analysis: An information content approach. Knowledge-Based Systems **21** (2008) 80–87
15. Alani, H., Brewster, C.: Metrics for ranking ontologies. In: Proceedings of the 4th EON2006 at the 15th WWW 2006, Edinburgh, Scotland (2006) 24–30
16. Wu, Z., Palmer, M.: Verbs semantics and lexical selection. In: ACL '94, Stroudsburg, PA, USA, Association for Computational Linguistics (1994) 133–138

What can FCA do for database linkkey extraction? (problem paper)

Manuel Atencia[1,2], Jérôme David[1,2], and Jérôme Euzenat[2,1]

[1] Univ. Grenoble Alpes & [2] INRIA,
Grenoble, France
http://exmo.inria.fr

Abstract. Links between heterogeneous data sets may be found by using a generalisation of keys in databases, called linkkeys, which apply across data sets. This paper considers the question of characterising such keys in terms of formal concept analysis. This question is natural because the space of candidate keys is an ordered structure obtained by reduction of the space of keys and that of data set partitions. Classical techniques for generating functional dependencies in formal concept analysis indeed apply for finding candidate keys. They can be adapted in order to find database candidate linkkeys. The question of their extensibility to the RDF context would be worth investigating.

We aim at finding correspondences between properties of two RDF datasets which allows for identifying items denoting the same individuals. This is particularly useful when dealing with linked data [8] for finding equality links between data sets.

Because the RDF setting raises many additional problems, we restrict ourselves here to databases. The problem is illustrated by the two (small) book relations of Table 1 (from [5], p.116). We would like to characterise a way to identify items on the same line while not (wrongly) identifying any other pair of items.

bookstore relation					library relation				
id	firstname	title	lastname	lang	year	author	orig	translator	wid
id	fn	tt	ln	lg	y	a	o	t	w
					1845	Poe	Raven	Baudelaire	a1
					1845	Poe	Raven	Mallarmé	a2
3	E.	Gold bug	Poe	en	1843	Poe	Gold Bug	Baudelaire	b
4	T.	On murder	Quincey	en	1827	Quincey	On murder	Schwob	c
5	T.	Kant	Quincey	en	1827	Quincey	Kant	Schwob	d
6	T.	Confessions	Quincey	en	1822	Quincey	Confessions	Musset	e
7	J.-J.	Confessions	Rousseau	fr					
8	T.	Confessions	Aquinus	fr					

Table 1. Two relations with, on the same lines, those tuples that represents the same individual (the line after attribute names are abbreviations).

For that purpose, we have defined linkkeys [5, 2] and we would like to formulate the linkkey extraction problem in the framework of formal concept analysis [6].

We first present this problem in the context of database candidate key extraction where one looks for sets of attributes and the sets of equality statements that they generate. We formulate this problem as the computation of a concept lattice. Then we turn to an adaptation of linkkeys to databases and show that the previous technique cannot be used for extracting the expected linkkeys. Instead we propose an adaptation.

1 Candidate keys in databases

A relation $\mathcal{D} = \langle \mathcal{A}, \mathcal{T} \rangle$ is a set of tuples $\mathcal{T}$ characterised by a set of attributes $\mathcal{A}$. A key is a subset of the attributes whose values identify a unique tuple.

Definition (key) Given a database relation $\mathcal{D} = \langle \mathcal{A}, \mathcal{T} \rangle$, a key is a subset of the attributes $K \subseteq \mathcal{A}$, such that $\forall t, t' \in \mathcal{T}, (\forall p \in K, p(t) = p(t')) \Rightarrow t \approx t'$.

Classically, keys are defined from functional dependencies. A set of attributes A is functionally dependent from another K, if equality of the attributes of K determines equality for the attributes of A. If the equality between tuples is the same thing as the equality for all attribute values, then a key is simply those sets of attributes of which $\mathcal{A}$ is functionally dependent.

However, we have not used the equality between tuple ($=$) but a particular $\approx$ relation. The reason is that we do not want to find keys for the database with $=$, but with an unknown relation $\approx$ which is to be discovered.

The statements $t \approx t'$ are those equality statements that are generated by the key. The $\approx$ relation must contain $=$ ($t = t' \Rightarrow t \approx t'$) and be an equivalence relation (this is by definition if it is the smallest relation satisfying the key).

From a key K of a relation $\langle \mathcal{A}, \mathcal{T} \rangle$, it is easy to obtain these statements through the function $\gamma : 2^{\mathcal{A}} \rightarrow 2^{\mathcal{T} \times \mathcal{T}}$ such that $\gamma(K) = \{t \approx t' | \forall p \in K, p(t) = p(t')\}$. γ is anti-monotonic ($\forall K, K' \subseteq \mathcal{A}, K \subseteq K' \Rightarrow \gamma(K) \supseteq \gamma(K')$).

We define *candidate key extraction* as the task of finding the minimal sets of attributes which generate a partition of the set of tuples.

Definition (candidate key) Given a database relation $\mathcal{D}$, a candidate key is a key such that none of its proper subsets generate the same partition. $\kappa(\mathcal{D})$ is the set of candidate keys.

Those candidate keys which generate the $singletons(\mathcal{T})$ partition are called normal candidate keys and their set noted $\hat{\kappa}(\mathcal{D}) = \{K \in \kappa(\mathcal{D}) | \forall (t \approx t') \in \gamma(K), t = t'\}$.

The problem of candidate key extraction is formulated in the following way:

Problem: Given a database relation $\mathcal{D}$, find $\kappa(\mathcal{D})$.

This problem is usually not considered in databases. Either keys are given and used for finding equivalent tuples and reducing the table, or the table is assumed without redundancy and keys are extracted. In this latter case, the problem is the extraction of normal candidate keys.

Using lattices is common place for extracting functional dependencies [9, 4] and the link to extract functional dependencies with formal concept analysis has already been considered [6] and further refined [10, 3].

In fact, this link can be fully exploited for extracting candidate keys instead of finding functional dependencies.

It consists of defining[1] a formal context $enc(\langle \mathcal{A}, \mathcal{T} \rangle) = \langle \mathfrak{P}_2(\mathcal{T}), \mathcal{A}, I \rangle$ such that: $\forall p \in \mathcal{A}, \forall \langle t, t' \rangle \in \mathfrak{P}_2(\mathcal{T})$,

$$\langle t, t' \rangle I p \text{ iff } p(t) = p(t')$$

The (formal) concepts of this encoding, that we denote by the set $FCA(enc(\mathcal{D}))$, associate a set of attributes to a set of pairs of tuples. These pairs of tuples are tuples that cannot be distinguished by the values of the attributes, i.e., our $\approx$ assertions. The candidate keys are the minimal elements of the intent which generate exactly the corresponding partition[2]. $\kappa(\mathcal{D}) = \bigcup_{c \in FCA(enc(\mathcal{D}))} \mu_{\subseteq} \{K \subseteq intent(c) | \gamma(K) = \gamma(intent(c))\}$.

For any key $K \in \kappa(\mathcal{D})$, $\gamma(c)$ is the reflexive, transitive and symmetric closure of the extent of its concept.

If this method is applied to the data sources of Table 1, the result is displayed in Figure 1.

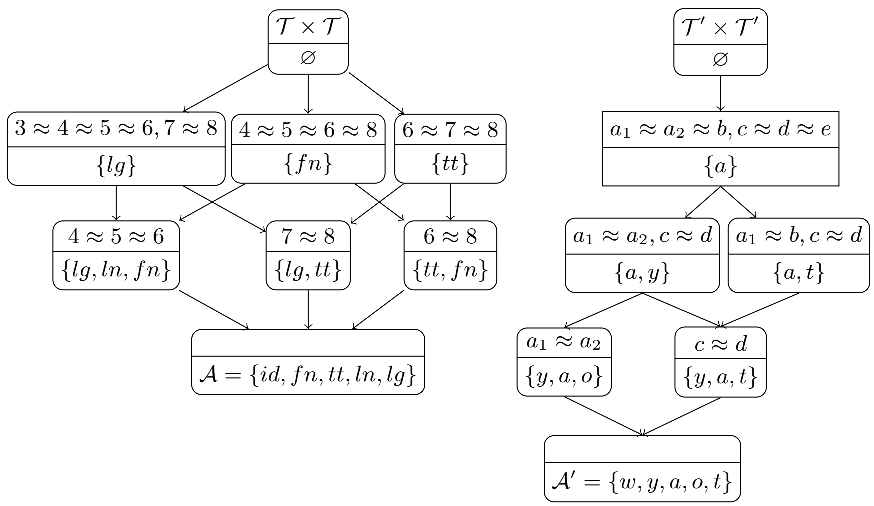


Fig. 1. Example of the key concept lattice for the bookstore (left) and library (right) relations.

As presented in Table 2, this may generate several candidate keys for the same concept ($\{o, t\}$ and $\{w\}$ for the maximal partition in the library dataset; in the bookstore dataset, the concept of extent $4 \approx 5 \approx 6$ has two candidate keys $\{ln\}$ and $\{lg, fn\}$ and the maximal partition has three candidate keys $\{id\}$, $\{tt, ln\}$ and $\{tt, fn, lg\}$).

This answers positively to our first question: it is possible to extract keys, i.e., generating $\kappa(\mathcal{D})$ from data with some help from formal concept analysis.

[1] For an arbitraty total strict order $<$ on $\mathcal{T}$, $\mathfrak{P}_2(\mathcal{T}) = \{\langle t, t' \rangle \in \mathcal{T}^2 \mid t < t'\}$.

[2] $\mu_R E = \{X \in E | \forall X' \in E, \neg(X' R X))\}$.

intent	potential keys	candidate keys	extent
$\{id, fn, tt, ln, lg\}$	id. . . ,	$\{id\}, \{tt, ln\}, \{tt, fn, lg\}$	$\varnothing$
$\{fn, ln, lg\}$	fnlnlg, fnlg, fnln, lgln, ln	$\{ln\}, \{fn, lg\}$	$\{4 \approx 5 \approx 6\}$
$\{tt, lg\}$	ttlg	$\{tt, lg\}$	$\{7 \approx 8\}$
$\{fn, tt\}$	fntt	$\{fn, tt\}$	$\{6 \approx 8\}$
$\{lg\}$	lg	$\{lg\}$	$\{3 \approx 4 \approx 5 \approx 6, 7 \approx 8\}$
$\{fn\}$	fn	$\{fn\}$	$\{4 \approx 5 \approx 6 \approx 8\}$
$\{tt\}$	tt	$\{tt\}$	$\{6 \approx 7 \approx 8\}$
$\varnothing$	$\varnothing$	$\varnothing$	$\mathcal{T} \times \mathcal{T}$
$\{w, y, a, t, o\}$	w, . . . ot, yot, aot, yaot, wyaot	$\{o, t\}, \{w\}$	$\varnothing$
$\{y, a, o\}$	o, ao, at, yo, yao	$\{o\}$	$\{a_1 \approx a_2\}$
$\{y, a, t\}$	yt, yat	$\{y, t\}$	$\{c \approx d\}$
$\{y, a\}$	y, ya	$\{y\}$	$\{a_1 \approx a_2, c \approx d\}$
$\{a, t\}$	t	$\{t\}$	$\{a_1 \approx b, c \approx d\}$
$\{a\}$	a	$\{a\}$	$\{a_1 \approx a_2 \approx b, c \approx d \approx e\}$
$\varnothing$	$\varnothing$	$\varnothing$	$\mathcal{T}' \times \mathcal{T}'$

Table 2. The list of concepts extracted from the bookstore (top) and library (bottom) relations. All intent should be completed by their subsets.

2 Database linkkey extraction

Consider that, instead of one relation, we are faced with two relations from two different databases which may contain tuples corresponding to the same individual.

We assume that candidate attribute pairs are already available through an alignment A which expresses equivalences between attributes of both relations. In this example, $A = \{\langle \mathsf{lastname}, \mathsf{author}\rangle, \langle \mathsf{title}, \mathsf{orig}\rangle, \langle \mathsf{id}, \mathsf{wid}\rangle\}$. Our goal is to find those which will identify the same individuals (tuples) in both databases.

2.1 Linkkeys for databases

Linkkeys [5] have been introduced for generating equality, a.k.a. sameAs, links between RDF datasets. We present a simplified notion of linkkey which is defined over relations.
Definition (Linkkey) Given two relations $\mathcal{D} = \langle \mathcal{A}, \mathcal{T}\rangle$ and $\mathcal{D}' = \langle \mathcal{A}', \mathcal{T}'\rangle$ and an alignment $A \subseteq \mathcal{A} \times \mathcal{A}'$. $LK \subseteq A$ is a linkkey between $\mathcal{D}$ and $\mathcal{D}'$ iff $\forall t, t' \in \mathcal{T}, \mathcal{T}'; (\forall \langle p, p'\rangle \in LK, p(t) = p(t')) \Rightarrow t \approx t'$. The set of linkkeys between $\mathcal{D}$ and $\mathcal{D}'$ with respect to A is denoted $\kappa_A(\mathcal{D}, \mathcal{D}')$.

This definition may be rendered independent from A by assuming $A = \mathcal{A} \times \mathcal{A}'$, so any attribute of one relation may be matched to any other.

2.2 Strong linkkey extraction

One way to deal with this problem is to start with keys: either candidate keys or normal candidate keys. For that purpose, we define $\kappa(\mathcal{D})/A$ as the operation which replaces, in all candidate keys of $\mathcal{D}$, each occurrence of an attribute in a correspondence of A by this correspondence[3].

A first kind of linkkeys that may be extracted are those which are normal candidate keys in their respective relations. They are called strong linkkeys and may be obtained by selecting normal candidate keys that contain only attributes mentioned in the alignment (replacing the attribute by the correspondence) and to intersect them, i.e., $\hat{\kappa}(\mathcal{D})/A \cap \hat{\kappa}(\mathcal{D}'\rangle)/A$. Strong linkkeys have the advantage of identifying tuples matching across relations without generating any links within the initial relations.

In the example of Table 1, there is one such strong linkkey: $\{\langle \mathsf{id}, \mathsf{wid}\rangle\}$. Indeed, the normal candidate keys for the bookstore relation are $\{\mathsf{id}\}$, $\{\mathsf{title}, \mathsf{lastname}\}$, or $\{\mathsf{title}, \mathsf{firstname}, \mathsf{lang}\}$ and, for the library relation they are $\{\mathsf{wid}\}$ and $\{\mathsf{orig}, \mathsf{translator}\}$. Since, translator has no equivalent in the bookstore relation (through A), only $\{\langle \mathsf{id}, \mathsf{wid}\rangle\}$ can be used. Unfortunately, it does not identify any equality statement as this happens very often with databases surrogates (this may have been worse if both relations used integers as identifiers: identifying false positives).

This scheme may be relaxed by trying to extract linkkeys from all candidate keys. In this way one would simply use $\kappa(\mathcal{D})/A \cap \kappa(\mathcal{D}')/A$. In our example, this does not bring further linkkeys.

2.3 Candidate linkkey extraction

The technique proposed above, does indeed generate linkkeys, but does not generate all of them: linkkeys may rely on sets of attributes which are not candidate keys. Indeed, one interesting linkkey for the relations above is $\{\langle \mathsf{lastname}, \mathsf{author}\rangle, \langle \mathsf{title}, \mathsf{orig}\rangle\}$.

Surprisingly, it does not use a normal candidate key of the library relation and not even a candidate key of the bookstore relation as $\{\mathsf{author}, \mathsf{orig}\}$ generates the same links as $\{\mathsf{orig}\}$ in this relation. However, when applied to the elements of $\mathcal{T} \times \mathcal{T}'$ this linkkey generates non ambiguous links, i.e., links which do not entail new links within a relation (this would have been different if a tuple $\langle \mathsf{year} = 1822, \mathsf{author} = \mathsf{Quincey}, \mathsf{orig} = \mathsf{Confessions}, \mathsf{translator} = \mathsf{Baudelaire}\rangle$ were present in the library relation).

Such linkkeys may be found by the same type of technique as before. It consists of defining a formal context $enc(\langle \mathcal{A}, \mathcal{T}\rangle, \langle \mathcal{A}', \mathcal{T}'\rangle, A) = \langle \mathcal{T} \times \mathcal{T}', A, I\rangle$ such that: $\forall \langle p, p'\rangle \in A, \forall \langle t, t'\rangle \in \mathcal{T} \times \mathcal{T}'$,

$$\langle t, t'\rangle I \langle p, p'\rangle \text{ iff } p(t) = p'(t')$$

γ is redefined to deal with subsets of alignments and generate $\approx$ assertions on $\mathcal{T} \cup \mathcal{T}'$. But, in order for $\approx$ to remain an equivalence relation it will be necessary to close $\approx$ on $\mathcal{T} \cup \mathcal{T}'$ and not only on $\mathcal{T} \times \mathcal{T}'$. Indeed if two tuples of $\mathcal{T}$ are found equal to a tuple of $\mathcal{T}'$, then by transitivity, they should be equal as well.

Again, candidate linkkeys are the minimal elements of the intent which generate exactly the corresponding set of links. $\kappa_A(\mathcal{D}, \mathcal{D}') = \bigcup_{c \in FCA(enc(\mathcal{D}, \mathcal{D}', A))} \mu_{\subseteq}\{K \subseteq intent(c) | \gamma(K) = \gamma(intent(c))\}$.

[3] This assumes that the alignment is one-to-one. This assumption is necessary for this subsection of the paper only.

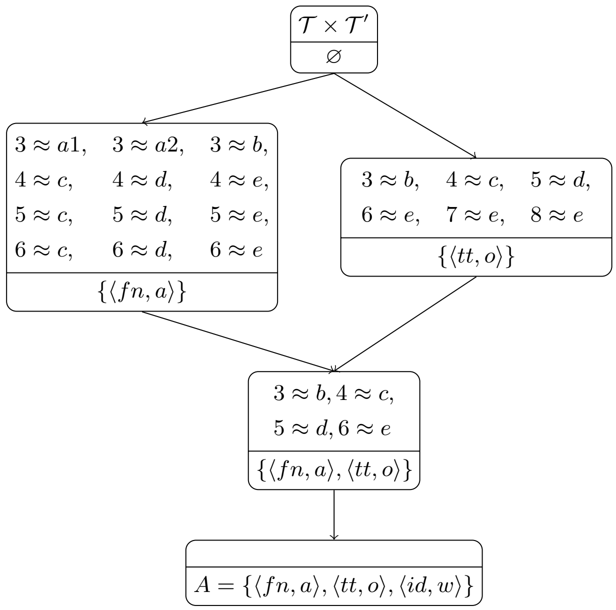


Fig. 2. Linkkey concept lattice for the relations of Table 1.

This technique, applied to the example of Table 1, generates the lattice of Figure 2. It can be argued that the candidate linkkeys $\{\langle fn, a\rangle, \langle tt, o\rangle\}$ and $\{\langle id, w\rangle\}$ are better than the others because they do not generate other statements within the relations. Indeed, $\{\langle tt, o\rangle\}$ generates $6 \approx 7 \approx 8$, and $\{\langle fn, a\rangle\}$ generates $a_1 \approx a_2 \approx b$, $c \approx d \approx e$ and $4 \approx 5 \approx 6$.

3 Conlusion and further work

We introduced, in the context of the relational model, the notions of candidate keys and linkkeys and we discussed potential links with formal concept analysis. These are only a few elements of a wider program. Problems were expressed in the relational framework because they are simpler. Our ambition is to provide an integrated way to generate links across RDF data sets using keys and it may be worth investigating if the proposed formal concept analysis framework can be extended to full RDF data interlinking.

Plunging this in the context of RDF requires further developments:

- considering that values do not have to be syntactically equal but may be found equal with respect to some theory: this may be a simple set of equality statement ("étudiant"="student") or may depend on RDF Schemas or OWL ontologies;
- considering several tables depending on each others together (this is related to Relational Concept Analysis [7] and could use the notion of foreign keys);

– considering that RDF attributes are not functional and hence yield a more general type of keys [1].

Once this is integrated within a common theoretical framework, a full solution requires work before and after running formal concept analysis:

– Before, it is necessary to use ontology/database matching [5] and to proceed to value normalisation.
– After, it is necessary to select among these potential or candidate keys those which are the more accurate [2].

Acknowledgements

This work has been partially supported by the ANR projects Qualinca (12-CORD-0012 for Manuel Atencia), and Lindicle (12-IS02-0002 for all three authors), and by grant TIN2011-28084 (for Manuel Atencia and Jérôme David) of the Ministry of Science and Innovation of Spain, co-funded by the European Regional Development Fund (ERDF).

Thanks to the anonymous reviewers for helping clarifying the paper.

References

1. Manuel Atencia, Michel Chein, Madalina Croitoru, Michel Chein, Jérôme David, Michel Leclère, Nathalie Pernelle, Fatiha Saïs, François Scharffe, and Danai Symeonidou. Defining key semantics for the RDF datasets: experiments and evaluations. In *Proc. 21st International Conference on Conceptual Structures (ICCS), Iasi (RO)*, pages 65–78, 2014.
2. Manuel Atencia, Jérôme David, and Jérôme Euzenat. Data interlinking through robust linkkey extraction. In *Proc. 21st european conference on artificial intelligence (ECAI), Praha (CZ)*, 2014.
3. Jaume Baixeries, Mehdi Kaytoue, and Amedeo Napoli. Characterizing functional dependencies in formal concept analysis with pattern structures. *Annals of mathematics and artificial intelligence*, 2014. To appear.
4. János Demetrovics, Leonid Libkin, and Ilya Muchnik. Functional dependencies in relational databases: A lattice point of view. *Discrete Applied Mathematics*, 40(2):155–185, 1992.
5. Jérôme Euzenat and Pavel Shvaiko. *Ontology matching*. Springer, Heidelberg (DE), 2nd edition, 2013.
6. Bernhard Ganter and Rudolf Wille. *Formal concept analysis: mathematical foundations*. Springer, Berlin, New York, Paris, 1999.
7. Mohamed Rouane Hacene, Marianne Huchard, Amedeo Napoli, and Petko Valtchev. Relational concept analysis: mining concept lattices from multi-relational data. *Annals of Mathematics and Artificial Intelligence*, 67(1):81–108, 2013.
8. Tom Heath and Christian Bizer. *Linked Data: Evolving the Web into a Global Data Space*. Morgan & Claypool, 2011.
9. Mark Levene. A lattice view of functional dependencies in incomplete relations. *Acta cybernetica*, 12(2):181–207, 1995.
10. Stéphane Lopes, Jean-Marc Petit, and Lotfi Lakhal. Functional and approximate dependency mining: database and FCA points of view. *Journal of Experimental & Theoretical Artificial Intelligence*, 14(2-3):93–114, 2002.

Lattice-Based View Access: A way to Create Views over SPARQL Query for Knowledge Discovery

Mehwish Alam and Amedeo Napoli

LORIA (CNRS – Inria Nancy Grand Est – Université de Lorraine)
BP 239, Vandoeuvre-lès-Nancy, F-54506, France
{mehwish.alam,amedeo.napoli}

Abstract. The data published in the form of RDF resources is increasing day by day. This mode of data sharing facilitates the exchange of information across the domains. Although it provides easier ways in the use of data such as through SPARQL queries. These queries over semantic web data usually produce list of tuples as answers which may be huge in number or may require further manipulation so that it can be understood and interpreted. Accordingly, this paper introduces a new clause `View By` in the SPARQL query for creating semantic views over the raw SPARQL query answers. This approach namely, Lattice-Based View Access (LBVA), is a framework based on Formal Concept Analysis (FCA). It provides a classification of the answers of SPARQL queries based on a concept lattice, that can be navigated for retrieving or mining specific patterns in query results w.r.t. user constraints. In this way, the concept lattice can be considered as a materialized view of the data resulting from a SPARQL query.

Keywords: Formal Concept Analysis, SPARQL Query Views, Lattice-Based Views, SPARQL, Classification.

Introduction

A considerable amount of Semantic Web (SW) data is already available on the web. Thus many agents are looking for more and more data present in the form of ontologies and RDF triples. Linked Open Data (LOD) [2] is a huge source of RDF resources interlinked with each other to form a cloud. SPARQL queries are used in order to make these data usable by domain experts and software agents. Sometimes queries are executed which may generate huge amount of results giving rise to the problem of *information overload* [4]. A typical example is the answers retrieved by search engines, which may mix between several meanings of one keyword. In case of huge results, many results are navigated to find the interesting links, which may be overwhelming without any navigation tool. Same is the case with the answers obtained by SPARQL queries, which may be huge in number and it may be harder to extract the most interesting patterns. This problem of information overload raises new challenges for data modeling and analysis and calls for improving data access, information retrieval and knowledge discovery w.r.t web querying.

In order to deal with the problem of *information overload*, this paper proposes a new approach based on Formal Concept Analysis (FCA [5]), which provides a lattice-based classification of the results obtained by SPARQL queries w.r.t user constraints. This new framework, namely Lattice Based View Access (LBVA), allows the classification of SPARQL query results into a concept lattice, referred to as views, for data analysis, knowledge discovery and information retrieval purposes. Based on one SPARQL query several views can be generated from different perspectives. In addition, LBVA allows for navigation over SPARQL query results. Here after, we describe how the *views* (a view corresponds to a concept lattice) can be designed from a SPARQL query and the result which is returned. Moreover, the analysis and the interpretation of the views is totally supported by the concept lattice. In case of large data only a part of the lattice can be considered for the analysis using the technique of iceberg lattices.

The intuition of classifying results obtained by querying LOD is inspired by web clustering engines [3] such as Carrot2[1]. The general idea behind web clustering engines is to group the results obtained by query posed by the user based on the different meanings of the terms related to a query. Such systems deal with unstructured textual data on web. However, there are some studies conducted to deal with structured RDF data. In [4], the authors target the problem of managing large amounts of results obtained by conjunctive queries with the help of subsumption hierarchy present in the knowledge base. On the other hand, lattice-based views provide classification based on the formal concepts and a partially ordered organization of the results. It also opens possibilities for navigation or information retrieval by traversing the concept lattice and for data analysis by allowing the extraction of association rules from the lattice. Additionally, unlike [4], LBVA also deals with data that has no schema (which is often the case with linked data).

The concept lattice provides a well founded structure on which a series of interpretations can be carried out. This framework is general and does not depend on any particular domain and may be used in addition with external resources, e.g. domain knowledge.

The paper is structured as follows: Section 1 gives a brief overview of Linked Open Data and gives the motivating example. Section 2 defines LBVA and gives the overall architecture of the framework. Section 3 briefly described the experimentation setting. Finally, Section 4 concludes the paper.

1 Linked Open Data

Linked Open Data (LOD) [2] is the way of publishing structured data which helps in the connection between several resources through their schema. LOD

[1] http://project.carrot2.org/index.html

represents data in the form of RDF graphs. Given a set of URIs U, blank nodes B and literals L, an RDF triple is represented as $t = (s, p, o) \in (\mathbf{U} \cup \mathbf{B}) \times \mathbf{U} \times (\mathbf{U} \cup \mathbf{B} \cup \mathbf{L})$, where s is a subject, p is a predicate and o is an object. A finite set of RDF triples is called as RDF Graph $\mathcal{G}$ such that $\mathcal{G} = (V, E)$, where V is a set of vertices and E is a set of labeled edges and $\mathcal{G} \in \mathbf{G}$, such that $\mathbf{G} = (\mathbf{U} \cup \mathbf{B}) \times \mathbf{U} \times (\mathbf{U} \cup \mathbf{B} \cup \mathbf{L})$. Each pair of vertices connected through a labeled edge keeps the information of a statement. Each statement is represented as $\langle subject, predicate, object \rangle$ referred to as an RDF Triple. V includes *subject* and *object* while E includes the *predicate*.

SPARQL[2] is the standard query language for RDF. In the current work we will focus more on the type of queries whose output performs value selection over the variables matching the patterns (queries containing *SELECT* clause). Now let us assume that there exists a set of variables V disjoint from U in the above definition of RDF, then $(U \cup V) \times (U \cup V) \times (U \cup V)$ is a graph pattern called a triple pattern. If a variable $?X \in V$ and $?X = c$ then $c \in U$. Given U, V and a triple pattern t a mapping $\mu(t)$ would be the triple obtained by replacing variables in t with U. $[\![.]\!]_G$ takes an expression of patterns and returns a set of mappings. Given a mapping $\mu : V \rightarrow U$ and a set of variables $W \subseteq V$, μ is represented as $\mu_{|W}$, which is described as a mapping such that $dom(\mu_{|W}) = dom(\mu) \cap W$ and $\mu_{|W}(?X) = \mu(?X)$ for every $?X \in dom(\mu) \cap W$. Finally, the SPARQL SELECT query is defined as follows:

Definition 1. *A SPARQL SELECT query is a tuple (W, P), where P is a graph pattern and W is a set of variables such that $W \subseteq var(P)$. The answer of (W, P) over an RDF graph G, denoted by $[\![(W, P)]\!]_G$, is the set of mappings:*

$$[\![(W, P)]\!]_G = \{\mu_{|W} | \mu \in [\![P]\!]_G\}$$

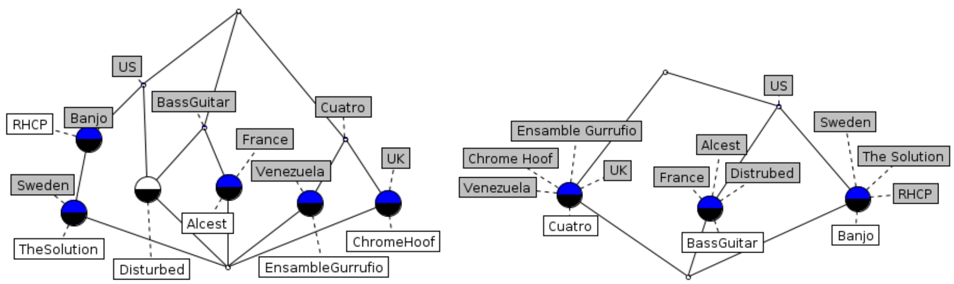
In the above definition $var(P)$ is the set of variables in pattern P and variables W in SELECT clause of SPARQL query[3]. Further details on the formalization and foundations of RDF databases are discussed in [1].

Example 1. Consider a query *all the bands which play different stringed instruments along with their origin.* This example will continue in the rest of this paper. Let us name this query `Q`, then `Q` can not be answered by standard search engines as it generates a separate list of bands and stringed instruments requiring multiple resources to be integrated. However, `Q` can be answered by SPARQL queries over LOD. For example, let us consider the SPARQL query in Listing 1.1 over DBpedia[4]. DBpedia is the central hub of LOD which extracts data from Wikipedia and makes it available in the structured format.

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

[3] In the rest of the paper we denote W as V to avoid overlap between the attribute values W in many-valued context.

[4] http://dbpedia.org/sparql



(a) Classes of Bands w.r.t. Musical Instruments and Countries, e.g., the concept on the top right corner with the attribute `Cuatro` contains all the bands which play Cuatro.

(b) Classes of Musical Instruments w.r.t Bands and their Origin

Fig. 1: Concept Lattices w.r.t Musical Instrument's and Band's Perspective.

Listing 1.1: SPARQL Query Q

```
SELECT ?band ?instrument ?origin WHERE {
  ?band rdf:type dbpedia-owl:Band.
  ?band dbpprop:origin ?origin.
  ?band dbpedia-owl:bandMember ?member .
  ?member dbpedia-owl:instrument ?instrument .
  ?instrument dcterms:subject Category:String_instruments .
GROUP BY ?instrument ?origin
```

The above SPARQL query returns a list of bands along with the instruments they play and their origin as an answer. An excerpt of the answers is shown below:

`dbpedia`[5]`:RHCP`[6] `dbpedia:Banjo dbpedia:US`

`dbpedia:Disturbed dbpedia:Bass_Guitar dbpedia:US,`

`dbpedia:The_Solution dbpedia:Banjo dbpedia:Sweden.`

In case of too many origins `GROUP BY` clause will lead to many small groups which would be hard for the user to observe with respect to origin or instrument, failing in the task of grouping. A classification technique can be used for navigation or interpretation. For example, Figure 1(a) shows a concept lattice for a small part of query answers. Here we can see classes such as the concept which contains all the bands which play `Cuatro`. If the search is more specified then the origin of each of the bands can also be retrieved. It is possible to retrieve bands which play `Cuatro` and are from `UK`, here `Chrome Hoof` is the band which plays `Cuatro` in the current small example. On the other hand, Figure 1(b) shows a concept lattice where musical instruments are classified with respect to bands and their origin, giving a totally different perspective over the same set of answers.

[5] http://dbpedia.org/resource/
[6] Red Hot Chilli Peppers

2 Lattice-Based View Access

In this paper, we propose an approach called Lattice-Based View Access which generates a concept lattice referred to as *view*. This view provides users with classification, navigation and analysis capabilities over these results. In the scenario of LOD, query processing procedure can not be controlled, so, in our algorithm we do not process the SPARQL query. The views are defined over RDF data by processing the set of tuples returned by the SPARQL query.

2.1 SPARQL Queries with Classification Capabilities

The idea of introducing a `VIEW BY` clause is to provide classification of the results and add a knowledge discovery aspect to the results w.r.t the variables appearing in `VIEW BY` clause. For example, we have a SPARQL SELECT query `Q = SELECT ?X ?Y ?Z WHERE {pattern P} VIEW BY ?X` then the set of variables $V = \{?X, ?Y, ?Z\}$. According to the definition 1 the answer of the tuple (V, P) is represented as $[\![(\{?X, ?Y, ?Z\}, P)]\!] = \mu_i$ where $i \in \{1, \ldots, k\}$ and k is the number of mappings obtained for the query `Q`. Here, $dom(\mu_i) = \{?X, ?Y, ?Z\}$ which means that $\mu(?X) = X_i$, $\mu(?Y) = Y_i$ and $\mu(?Z) = Z_i$. Finally, a complete set of mappings can be given as $\{\{?X \rightarrow X_i, ?Y \rightarrow Y_i, ?Z \rightarrow Z_i\}\}$.

Now, variables appearing in the `VIEW BY` clause are referred to as object variable[7] and is denoted as Ov such that $Ov \in V$. In the current scenario $Ov = \{?X\}$. The remaining variables are referred to as attribute variables and are denoted as Av where $Av \in V$ such that $Ov \cup Av = V$ and $Ov \cap Av = \emptyset$.

Example 2. An alternate query for the query in Listing 1.1 with the `VIEW BY` clause can be given as:

```
SELECT ?band ?instrument ?origin WHERE {
   ?band rdf:type dbpedia-owl:Band.
   ?band dbpprop:origin ?origin.
   ?band dbpedia-owl:bandMember ?member .
   ?member dbpedia-owl:instrument ?instrument .
   ?instrument dcterms:subject dbpedia8:Category:String_instruments.}
VIEW BY ?band
```

Here, V={`?band, ?instrument, ?origin`} then the evaluation of the SELECT query $[\![(\{?band, ?instrument, ?origin\}, P)]\!]$ will generate the mappings shown in Table 1. Accordingly, $dom(\mu_i) = \{?band, ?instrument, ?origin\}$. Here, $\mu_1(?band) = RHCP$, $\mu_1(?instrument) = Banjo$ and $\mu_1(?origin) = US$. In the current example, we have, $Ov = \{?band\}$ because it appears in the `VIEW BY` clause and $Av = \{?instrument, ?origin\}$. Figure 1a shows the generated view when $Ov = \{?band\}$ and in Figure 1b, we have; $Ov = \{?instrument\}$.

[7] The object here refers to the object in FCA.

[8] http://dbpedia.org/resource/

	?band	?instrument	?origin
μ_1	RHCP	Banjo	US
μ_2	Disturbed	Bass_Guitar	US
$\vdots$	$\vdots$	$\vdots$	$\vdots$

Table 1: Generated Mappings for SPARQL Query `Q`

2.2 Designing a Formal Context (G, M, W, I)

The results obtained by the query are in the form of set of tuples, which are then organized as a many-valued context. If $Ov = \{?X\}$ then $\mu(?X) = \{X_i\}_{i \in \{1,\ldots,k\}}$, where X_i denote the values obtained for the object variable and the corresponding mapping is given as $\{\{?X \to X_i\}\}$. Finally, $G = \mu(?X) = \{X_i\}_{i \in \{1,\ldots,k\}}$. Let $Av = \{?Y, ?Z\}$ then $M = Av$ and the attribute values $W = \{\mu(?Y), \mu(?Z)\} = \{\{Y_i\}, \{Z_i\}\}_{i \in \{1,\ldots,k\}}$. The corresponding mapping for attribute variables are $\{\{?Y \to Y_i, ?Z \to Z_i\}\}$ Consider an object value $g_i \in G$ and an attribute value $w_i \in W$ then we have $(g_i, \text{“}?Y'', w_i) \in I$ iff $?X(g_i) = w_i$, i.e., the value of g_i for attribute $?Y$ is w_i, $i \in \{1, \ldots, k\}$ as we have k values for $?Y$.

Obtaining Binary Context (G, M, I): Afterwards, a conceptual scaling used for binarizing the many-valued context, in the form of (G, M, I). Finally, we have $G = \{X_i\}_{i \in \{1,\ldots,k\}}$, $M = \{Y_i\} \cup \{Z_i\}$ where $i \in \{1, \ldots, k\}$ for object variable $Ov = \{?X\}$.

Example 3. In the example $Ov = \{band\}$, $Av = \{instrument, origin\}$. The answers obtained by this query are organized into a many-valued context as follows: the distinct values of the object variable $?band$ are kept as a set of objects, so $G = \{RHCP,\ Disturbed, \ldots\}$, attribute variables provide $M = \{instrument, origin\}$, $W_1 = \{Banjo,\ BassGuitar, \ldots\}$ and $W_2 = \{US,\ UK,\ France \ldots\}$ in a many-valued context. The obtained many-valued context is shown in Table 2. Following the above defined procedure a many-valued context is conceptually scaled to obtain a binary context shown in Table 3. The corresponding concept lattice is shown in Figure 1(a).

Band	Instrument	Origin
RHCP	Banjo	US
Disturbed	Bass Guitar	US
Alcest	Bass Guitar	France
The Solution	Banjo	Sweden, US
Chrome Hoof	Cuatro	UK
Ensamble Gurrufio	Cuatro	Venezuela

Table 2: Many-Valued Context representing the answer tuple (X_i, Y_i, Z_i).

Band	Banjo	Bass Guitar	Cuatro	US	Sweden	UK	France	Venezuela
RHCP	×			×				
Disturbed		×		×				
Alcestr		×					×	
The Solution	×			×	×			
Chrome Hoof			×			×		
Ensamble Gurrufio			×					×

Table 3: Formal Context $\mathcal{K}_{DBpedia}$.

2.3 Building a Concept Lattice

Once the context is designed, the concept lattice can be built using an FCA algorithm. This step is straight forward as soon as the context is provided. In the current implementation we use AddIntent [8] which is an incremental concept lattice construction algorithm. In case of large data iceberg lattices can be considered [6]. The use of `VIEW BY` clause activates the process of LBVA, which transforms the SPARQL query answers (tuples) to a formal context $\mathcal{K}_{answers}$ through which a concept lattice is obtained which is referred to as a *Lattice-Based View*. A view on SPARQL query in section 1, i.e, a concept lattice corresponding to Table 3 is shown in Figure 1a. At the end of this step the concept lattice is built and the interpretation step can be considered.

2.4 Interpretation Operations over a Concept Lattice

Navigation Operation and Knowledge Discovery: The obtained concept lattice can be navigated for searching and accessing particular LOD elements. It is possible to drill down from general to specific concepts according to some constraints. For example, in order to search for bands in `US` playing `Banjo`, the concept lattice in Figure 1(a) is explored levelwise. First the broader concept contains all the bands from `US`, `RHCP, The Solution, Disturbed`. Then, the children concepts contain more specific concepts with the instruments `Banjo` and `Bass Guitar`. According to the initial constraint, the attribute concept of `Banjo` can be selected returning two objects namely `RHCP, The Solution`. Next, to check which instruments are played in music originating from `US`, another concept lattice can be explored, where objects correspond to instruments shown in Figure 1(b). The results in this case is the set of objects `Bass Guitar, Banjo`.

FCA provides a powerful means for data analysis and knowledge discovery. Iceberg lattices provide the top most part of the lattice filtering out only general concepts. The concept lattice is still explored levelwise depending on a given threshold. Then, only concepts whose extent is sufficiently large are explored, i.e., the support of a concept corresponds to the cardinal of the extent. If further specific concepts are required the support threshold of the iceberg lattices can be lowered and the resulting concept lattice can be explored levelwise.

Another way of interpreting the data is provided by Duquenne-Guigues basis of implications which takes into account a minimal set of implications which represent all the association rules that can be generated for a given formal context. For example, $\mathcal{DG}$-basis of implications according to the formal context in Table 3 state that all the bands which play `Banjo` are from `US` (rule: `Banjo` $\rightarrow$ `US`). Moreover, the rule `Venezuela` $\rightarrow$ `Cuatro` suggests that all the bands from `Venezuela` play `Cuatro`. This rule states that Cuatro is widely used in the folk music of Venezuela.

3 Experimentation

Several experiments were conducted on real datasets. These datasets include DBpedia, Yago [7], which is a knowledge base automatically extracted from

Wikipedia (infoboxes, categories), Wordnet and Geonames. The experiment was also tested on the biomedical data such as Sider[9] and Drugbank[10]. Sider keeps the information about the medicines along with their side effects. Drugbank keeps the detailed information about the drugs such as drug category and target proteins. These experiments provide qualitative and quantitative evaluation to our approach. These experiments are not discussed in the current paper due to lack of space. However, the software, a detailed technical report along with the visualization of the SPARQL query views can be accessed online[11].

4 Conclusion and Discussion

In LBVA, we introduce a classification framework for the set of tuples obtained as a result of SPARQL queries over LOD. We introduce a classification framework based on FCA for organizing a view, i.e., the set of tuples resulting from a SPARQL query. In this way, the view is organized as a concept lattice that can be navigated where information retrieval and knowledge discovery can be performed. For future work, we are interested in working with several *object variable* allowing to deal with more complex relations, with the help of Relational Concept Analysis (RCA). In addition, here only binary contexts are taken into account. It is possible to go beyond this limitation in using another variation of FCA which is the formalism of pattern structures.

References

1. Marcelo Arenas, Claudio Gutierrez, and Jorge Pérez. Foundations of rdf databases. In *Reasoning Web*, pages 158–204, 2009.
2. Christian Bizer, Tom Heath, and Tim Berners-Lee. Linked data - the story so far. *Int. J. Semantic Web Inf. Syst.*, 5(3):1–22, 2009.
3. Claudio Carpineto, Stanislaw Osiński, Giovanni Romano, and Dawid Weiss. A survey of web clustering engines. *ACM Comput. Surv.*, 41(3):17:1–17:38, July 2009.
4. Claudia d'Amato, Nicola Fanizzi, and Agnieszka Lawrynowicz. Categorize by: Deductive aggregation of semantic web query results. In *ESWC (1)*, 2010.
5. Bernhard Ganter and Rudolf Wille. *Formal Concept Analysis: Mathematical Foundations.* Springer, Berlin/Heidelberg, 1999.
6. G. Stumme, R. Taouil, Y. Bastide, and L. Lakhal. Conceptual clustering with iceberg concept lattices. In *Proc. GI-Fachgruppentreffen Maschinelles Lernen (FGML'01)*, 2001.
7. Fabian M. Suchanek, Gjergji Kasneci, and Gerhard Weikum. Yago: A core of semantic knowledge. In *Proceedings of the 16th International Conference on World Wide Web*, WWW '07, pages 697–706, New York, NY, USA, 2007. ACM.
8. Dean van der Merwe, Sergei A. Obiedkov, and Derrick G. Kourie. Addintent: A new incremental algorithm for constructing concept lattices. In *ICFCA*, 2004.

[9] http://sideeffects.embl.de/
[10] http://www.drugbank.ca/
[11] http://webloria.loria.fr/~alammehw/lbva/