Claudio Carpineto, Sergei O. Kuznetsov, Amedeo Napoli (Eds.)

# FCAIR 2012 – Formal Concept Analysis Meets Information Retrieval

Workshop co-located with the 35th European Conference on Information Retrieval (ECIR 2013)
March 24, 2013, Moscow, Russia

**Volume Editors**

Claudio Carpineto
Fondazione Ugo Bordoni, Rome, Italy

Sergei O. Kuznetsov
School of Applied Mathematics and Information Science
National Research University Higher School of Economics, Moscow, Russia

Amedeo Napoli
LORIA (CNRS – Inria – Universite de Lorraine), Nancy, France

Printed in National Research University Higher School of Economics.

The proceedings are also published online on the CEUR-Workshop web site in a series with ISSN 1613-0073.

# Preface

Formal Concept Analysis (FCA) is a mathematically well-founded theory aimed at data analysis and classification, introduced and detailed in the book of Bernhard Ganter and Rudolf Wille, "Formal Concept Analysis", Springer 1999. The area came into being in the early 1980s and has since then spawned over 10000 scientific publications and a variety of practically deployed tools. FCA allows one to build from a data table with objects in rows and attributes in columns a taxonomic data structure called concept lattice, which can be used for many purposes, especially for Knowledge Discovery and Information Retrieval.

The "Formal Concept Analysis Meets Information Retrieval" (FCAIR) workshop collocated with the 35th European Conference on Information Retrieval (ECIR 2013) was intended, on the one hand, to attract researchers from FCA community to a broad discussion of FCA-based research on information retrieval, and, on the other hand, to promote ideas, models, and methods of FCA in the community of Information Retrieval.

This volume contains 11 contributions to FCAIR workshop (including 3 abstracts for invited talks and tutorial) held in Moscow, on March 24, 2013. All submissions were assessed by at least two reviewers from the program committee of the workshop to which we express our gratitude. We would also like to thank the co-organizers and sponsors of the FCAIR workshop: Russian Foundation for Basic Research, National Research University Higher School of Economics, and Yandex.

March 24, 2013

Claudio Carpineto
Sergei O. Kuznetsov
Amedeo Napoli

# Organization

## Workshop Co-Chairs

| | |
|---|---|
| Claudio Carpineto | Fondazione Ugo Bordoni, Rome, Italy |
| Sergei O. Kuznetsov | National Research University Higher School of Economics, Moscow, Russia |
| Amedeo Napoli | LORIA (CNRS – Inria – Universite de Lorraine), Nancy, France |

## Proceedings Chair

| | |
|---|---|
| Dmitry I. Ignatov | National Research University Higher School of Economics, Moscow, Russia |

## Program Committee

| | |
|---|---|
| Radim Belohlavek | Palacky University, Olomouc, Czech Republic |
| Peter Eklund | University of Wollongong, Australia |
| Sébastien Ferré | IRISA Rennes, France |
| Bernhard Ganter | Technische Universitaet Dresden, Germany |
| Andreas Hotho | University of Kassel, Germany |
| Robert Jaschke | Web Science, L3S Research Center Hannover, Germany |
| Dmitry I. Ignatov | Higher School of Economics, Moscow, Russia |
| Marianne Huchard | LIRMM Montpellier, France |
| Mehdi Kaytoue | LIRIS INSA, Lyon, France |
| Carlo Meghini | Istituto di Scienza e Tecnologie dell'Informazione, Pisa, Italy |
| Rokia Missaoui | University of Ottawa, Canada |
| Sergei A. Obiedkov | Higher Schools of Economics, Moscow, Russia |
| Uta Priss | Ostfalia University of Applied Sciences, Wolfenbüttel, Germany |
| Sebastian Rudolph | University of Karlsruhe, Germany |
| Baris Sertkaya | SAP Dresden, Germany |
| Petko Valtchev | Université du Québec à Montréal, Montréal, Canada |
| Francisco Valverde-Albacete | Universidad Carlos III de Madrid, Spain |

## Sponsoring Institutions

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

# Table of Contents

# FCA and IR: The Story So Far

Claudio Carpineto

Fondazione Ugo Bordoni, Rome
carpinet@fub.it

**Abstract.** The application of Formal Concept Analysis (FCA) to Information Retrieval (IR) is twenty-five years old. Over this period, a number of papers have explored the potentials of FCA for various information finding tasks while several system prototypes have been made available for experimentation and testing. In this talk we survey what has been achieved so far, discussing lessons and implications from a few successful case studies, including browsing of web search results, smooth integration of data driven and model driven search, exploratory document mining, and semantic text classification. We argue that, in spite of these good results, the impact of FCA on mainstream IR has been limited, due to theoretical and practical limitations. Nowadays, the tremendous increase in the richness and diversity of electronic data along with the inherent shortcomings of traditional search engines call for better IR techniques, opening up new opportunities for knowledge-intensive methods. Integration of FCA with existing search technology and new theoretical advances capable of extending the applicability of FCA to IR beyond the Boolean retrieval model are seen as key factors for the further development of this field.

# New Applications of Formal Concept Analysis: A Need for Original Pattern Domains

Jean-François Boulicaut

INSA Lyon, LIRIS CNRS UMR5205
F-69621 Villeurbanne cedex, France
Jean-Francois.Boulicaut@insa-lyon.fr

**Abstract.** We survey the results obtained by our research group (joint work with Jérémy Besson and Loïc Cerf, Kim-Ngan T. Nguyen, Marc Plantevit, and Céline Robardet) concerning the design of pattern domains to support knowledge discovery and information retrieval in arbitrary n-ary relations. Our contribution is related to Formal Concept Analysis and its recent developments in direction of, for instance, Triadic Concept Analysis. We focus on a real data mining perspective. It means that we need for both the design of scalable constraint-based mining algorithms and fault-tolerant approaches to support the discovery of relevant patterns from noisy data.

## 1 Extended abstract

The Formal Concept Analysis framework (FCA) has been studied for about three decades [13]. Given a binary relation, we may consider FCA as the computation and then the exploitation of a collection of closed patterns, the so-called formal concepts that are organized within a lattice structure. FCA supports knowledge discovery processes from such relations and many application domains have been considered. It includes applications to more or less simple information retrieval tasks (see, e.g., [5, 21]).

Nowadays, we have to face with more and more large but also structured types of data like, for instance, (collections) of graphs or information networks. New challenges have appeared such that pattern discovery methods have to be revisited. One important direction of research concerns the extension of FCA-based techniques for different types of data (e.g., numerical matrices, collections of strings). For instance, this is currently studied thanks to the concept of pattern structure [17, 16]. Also, Triadic Concept Analysis that concerns Boolean cube data analysis has been formalized in [18] and several algorithms have been proposed to discover patterns in such ternary relations (see, e.g., the computation of closed patterns [14, 15] or implications [12]). For instance, it can be applied to the discovery of conceptual structures in folksonomies that are ternary relations $Users \times Resources \times Tags$.

During the last decade, our research group[1] has been working on various evolutions of FCA where (a) datasets are arbitrary n-ary relations, (b) computed

[1] liris.cnrs.fr/equipes?id=46

patterns are not only closed but must also satisfy other user-defined primitive constraints, and (c) some fault-tolerance is provided.

Following the guidelines of inductive querying and constraint-based data mining [4, 11], we have been designing new pattern domains. The methodology is as follows.

Given a data type, we have to define pattern languages and measures that denote properties of patterns within the data. Then, we carefully design the primitive constraints that will be combined to support the declarative specification of both objective and subjective interestingness. Once declarative specifications are available - the so-called inductive queries - we must provide algorithms that compute the solution patterns. A major issue is to identify the constraint properties and the enumeration strategies that enable to compute correct and complete answers in practical cases. For this, generic algorithms can be designed: no specific combination of primitive constraint is expected but safe pruning theorems can be based on the constraint properties. Notice that it is generally possible to design more efficient ad-hoc algorithms when considering fixed forms of constraints.

In our 2008 survey [3], we were considering a constraint-based perspective on actionable formal concept mining from large binary relations. As a result, we were discussing the use of primitive constraints to compute more relevant formal concepts, for instance large-enough ones [2] but also some generalizations that provide fault-tolerance [1]. A few years later, it is now possible to discuss such issues in the enlarged setting of arbitrary n-ary relations. Therefore, we can consider (a) our generic algorithm that mines set patterns and exploits the large class of piecewise (anti-)monotonic constraints [7, 8], (b) its extension towards fault-tolerant pattern discovery by means of a correct and complete strategy [6] or an heuristic one [9]. We also studied a multidimensional association rule mining framework [19] that is based on closed pattern post-processing. Among others, promising though preliminary applications to dynamic relational graph analysis have been investigated [10, 20].

## References

1. J. Besson, R. Pensa, C. Robardet, and J.-F. Boulicaut. Constraint-based mining of fault-tolerant patterns from boolean data. In KDID'05 Revised Selected and Invited Papers, volume 3933 of LNCS, pages 55–71. Springer, 2005.
2. J. Besson, C. Robardet, J.-F. Boulicaut, and S. Rome. Constraint-based formal concept mining and its application to microarray data analysis. Intell. Data Anal., 9(1):59–82, 2005.
3. J.-F. Boulicaut and J. Besson. Actionability and formal concepts: A data mining perspective. In Proc. ICFCA, volume 4933 of LNCS, pages 14–31. Springer, 2008.
4. J.-F. Boulicaut, L. D. Raedt, and H. Mannila, editors. Constraint-Based Mining and Inductive Databases, volume 3848 of LNCS. Springer, 2005.
5. C. Carpineto and G. Romano. Using concept lattices for text retrieval and mining. In Proc. ICFCA, volume 3626 of LNCS, pages 161–179. Springer, 2005.
6. L. Cerf, J. Besson, K.-N. Nguyen, and J.-F. Boulicaut. Closed and noise-tolerant patterns in n-ary relations. Data Min. Knowl. Discov., 26(3):574–619, 2013.

7. L. Cerf, J. Besson, C. Robardet, and J.-F. Boulicaut. Data peeler: Contraint-based closed pattern mining in n-ary relations. In Proc. SIAM DM, pages 37–48, 2008.
8. L. Cerf, J. Besson, C. Robardet, and J.-F. Boulicaut. Closed patterns meet $n$-ary relations. ACM Transactions on KDD, 3(1), 2009.
9. L. Cerf, P.-N. Mougel, and J.-F. Boulicaut. Agglomerating local patterns hierarchically with ALPHA. In Proc. ACM CIKM, pages 1753–1756, 2009.
10. L. Cerf, T. B. N. Nguyen, and J.-F. Boulicaut. Mining constrained cross-graph cliques in dynamic networks. In Inductive Databases and Queries: Constraint-Based Data Mining, pages 201–230. Springer, 2010.
11. S. Dzeroski, B. Goethals, and P. Panov, editors. Inductive Databases and Queries: Constraint-Based Data Mining. Springer, 2010.
12. B. Ganter and S. A. Obiedkov. Implications in triadic formal contexts. In Proc. ICCS, volume 3127 of LNCS, pages 186–195. Springer, 2004.
13. B. Ganter, G. Stumme, and R. Wille, editors. Formal Concept Analysis, Foundations and Applications, volume 3626 of LNCS. springer, 2005.
14. R. Jaschke, A. Hotho, C. Schmitz, B. Ganter, and G. Stumme. TRIAS–an algorithm for mining iceberg tri-lattices. In Proc. IEEE ICDM, pages 907–911, 2006.
15. L. Ji, K.-L. Tan, and A. K. H. Tung. Mining frequent closed cubes in 3D data sets. In Proc. VLDB, pages 811–822, 2006.
16. M. Kaytoue, S. O. Kuznetsov, and A. Napoli. Revisiting numerical pattern mining with formal concept analysis. In Proc. IJCAI, pages 1342–1347, 2011.
17. S. O. Kuznetsov. Pattern structures for analyzing complex data. In Proc. RSFDGrC, volume 5908 of LNCS, pages 33–44. Springer, 2009.
18. F. Lehmann and R. Wille. A triadic approach to formal concept analysis. In Proc. ICCS, volume 954 of LNCS, pages 32–43. Springer, 1995.
19. K.-N. Nguyen, L. Cerf, M. Plantevit, and J.-F. Boulicaut. Multidimensional association rules in boolean tensors. In Proc. SIAM DM, pages 570–581, 2011.
20. K.-N. Nguyen, L. Cerf, M. Plantevit, and J.-F. Boulicaut. Discovering descriptive rules in relational dynamic graphs. Intell. Data Anal., 17(1):49–69, 2013.
21. J. Poelmans, D. I. Ignatov, S. Viaene, G. Dedene, and S. O. Kuznetsov. Text mining scientific papers: A survey on fca-based information retrieval research. In Proc. ICDM, volume 7377 of LNCS, pages 273–287. Springer, 2012.

# Key Phrase to Text Similarity, Clustering, and Interpretation in Hierarchical Ontologies

Boris Mirkin

Applied Mathematics and Informatics, National Research University Higher School of Economics Moscow
Computer Science and Information Systems, Birkbeck University of London
BMirkin@hse.ru

**Abstract.** Scoring similarity between key phrases and unstructured texts is an issue which is important in both information retrieval and text analysis. Researchers from the two fields use different scoring functions, although clear delineation between the two still is lacking. We use suffix tree based score expressing the average conditional probability of a symbol in a common substring. Usually, a domain taxonomy serves as the source of key-phrases. Given a set of entities, such as texts or projects or working groups, one can derive clusters of key-phrases using key-phrase-to-entity scores. The clusters represent common themes in the meaning of texts or in activities of working groups. To interpret them, the domain ontology should be used. If the ontology is a rooted tree, a lifting method is proposed to find the most parsimonious interpreting head subject(s), up to a few gaps and offshoots. Some applications and application issues are considered. The work is being conducted jointly with T. Fenner (London), S. Nascimento (Lisbon) and E. Chernyak (Moscow).

# Improving Text Retrieval Efficiency with Pattern Structures on Parse Thickets

Boris A. Galitsky[1,2], Dmitry Ilvovsky[2], Fedor Strok[2] and Sergei O. Kuznetsov[2]

[1] eBay Inc San Jose CA USA
[2] Higher School of Economics, Moscow Russia
{bgalitsky@hotmail.com; dilv_ru@yahoo.com; fdr.strok@gmail.com; skuznetsov@hse.ru}

**Abstract.** We develop a graph representation and learning technique for parse structures for paragraphs of text. We introduce Parse Thicket (PT) as a sum of syntactic parse trees augmented by a number of arcs for inter-sentence word-word relations such as co-reference and taxonomic relations. These arcs are also derived from other sources, including Speech Act and Rhetoric Structure theories. The operation of generalizing logical formulas is extended towards parse trees and then towards parse thickets to compute similarity between texts. We provide a detailed illustration of how PTs are built from parse trees, and generalized. The proposed approach is subject to preliminary evaluation in the product search domain of eBay.com, where user queries include product names, features and expressions for user needs, and query keywords occur in different sentences of an answer. We demonstrate that search relevance is improved by PT generalization.

**Keywords:** graph representation of text, learning syntactic parse tree, syntactic generalization, search relevance

## 1 Introduction

Parse trees have become a standard form of representing the linguistic structures of sentences. In this study we will attempt to represent a linguistic structure of a *paragraph of text* based on parse trees for each sentence of this paragraph. We will refer to the sum of parse trees plus a number of arcs for inter-sentence relations between nodes for words as Parse Thicket (PT). A PT is a graph which includes parse trees for each sentence, as well as additional arcs for inter-sentence relationship between parse tree nodes for words.

In this paper we will define the operation of *generalization of text paragraphs* to assess similarity between portions of text. Use of generalization for similarity assessment is inspired by structured approaches to machine learning versus unstructured, statistical where similarity is measured by a distance in feature space. Our intention is to extend the operation of least general generalization (unification of logic formula) towards structural representations of paragraph of texts. Hence we will define the operation of generalization on Parse Thickets and outline an algorithm for it.

This generalization operation is a base for number of text analysis application such as search, classification, categorization, and content generation [3]. *Generalization of text paragraphs* is based on the operation of generalization of two sentences, explored in our earlier studies [6,7,8]. In addition to learning generalizations of individual sentences, in this study we explore how the links between words in sentences other than syntactic ones can be used to compute similarity between texts. We will investigate how to formalize the theories of textual discourse such as Rhetoric Structure Theory [12] to improve the efficiency of text retrieval.

General pattern structures consist of objects with descriptions (called patterns) that allow a semilattice operation on them [9]. In our case, for paragraphs of text to serve such objects, they need to be represented by structures like parse thickets, which capture both syntactic level and discourse-level information about texts. Pattern structures arise naturally from ordered data, e.g., from labeled graphs ordered by graph morphisms. In our case labeled graphs are parse thickets, and morphisms are the mappings between their maximal common sub-graphs.

One of the first systems for the generation of conceptual graph representation of text is described in [18]. It uses a lexicon of canonical graphs that represent valid (possible) relations between concepts. These canonical graphs are then combined to build a conceptual graph representation of a sentence. Since then syntactic processing has dramatically improved, delivering reliable and efficient results.

[11] describes a system for constructing conceptual graph representation of text by using a combination of existing linguistic resources (VerbNet and WordNet). However, for practical applications these resources are rather limited, whereas syntactic level information such as syntactic parse trees is readily available. Moreover, building conceptual structure from individual sentences is not as reliable as building these structures from generalizations of two and more sentences.

In this study we attempt to approach conceptual graph level [15, 17] using pure syntactic information such as syntactic parse trees and applying learning to it to increase reliability and consistency of resultant semantic representation. The purpose of such automated procedure is to tackle information extraction and knowledge integration problems usually requiring deep natural language understanding [2] and cannot be solved at syntactic level.

Whereas machine learning of syntactic parse trees for individual sentences is an established area of research, the contribution of this paper is a structural approach to learning of syntactic information at the level of paragraphs. A number of studies applied machine learning to syntactic parse trees [1], convolution kernels being the most popular approach [10].

To represent the structure of a paragraph of text, given parse trees of its sentences, we introduce the notion of Parse Thicket (PT) as a union of parse trees. The union $G = G_1 \bigcup G_2$ of trees $G_1$ and $G_2$ with disjoint node sets $V_1$ and $V_2$ and edge sets $X_1$ and $X_2$ is the graph with $V = V_1 \bigcup V_2$ and $X = X_1 \bigcup X_2$.

## 2 Finding similarity between two paragraphs of text

We will compare the following approaches to assessing the similarity of text paragraphs:

- Baseline: bag-of-words approach, which computes the set of common keywords/n-grams and their frequencies.
- Pair-wise matching: we will apply syntactic generalization to each pair of sentences, and sum up the resultant commonalities. This technique has been developed in our previous work [3].
- Paragraph-paragraph match.

The first approach is most typical for industrial NLP applications today, and the second is the one of our previous studies. Kernel-based approach to parse tree similarities [20], as well as tree sequence kernel [19], being tuned to parse trees of individual sentences, also belongs to the second approach.

We intend to demonstrate the richness of the approach being proposed, and in the consecutive sections we will provide a step-by-step explanation. We will introduce a pair of short texts (articles) and compare the above three approaches. This example will go through the whole paper.

"Iran refuses to accept the UN proposal to end the dispute over work on nuclear weapons",
"UN nuclear watchdog passes a resolution condemning Iran for developing a second uranium enrichment site in secret",
"A recent IAEA report presented diagrams that suggested Iran was secretly working on nuclear weapons",
"Iran envoy says its nuclear development is for peaceful purpose, and the material evidence against it has been fabricated by the US",

$\wedge$

"UN passes a resolution condemning the work of Iran on nuclear weapons, in spite of Iran claims that its nuclear research is for peaceful purpose",
"Envoy of Iran to IAEA proceeds with the dispute over its nuclear program and develops an enrichment site in secret",
"Iran confirms that the evidence of its nuclear weapons program is fabricated by the US and proceeds with the second uranium enrichment site"

The list of common keywords gives a hint that both documents are on nuclear program of Iran, however it is hard to get more specific details

Iran, UN, proposal, dispute, nuclear, weapons, passes, resolution, developing, enrichment, site, secret, condemning, second, uranium

Pair-wise generalization gives a more accurate account on what is common between these texts: -+

[NN-work IN-* IN-on JJ-nuclear NNS-weapons ], [DT-the NN-dispute IN-over JJ-nuclear NNS-* ], [VBZ-passes DT-a NN-resolution ],
[VBG-condemning NNP-iran IN-* ],
[VBG-developing DT-* NN-enrichment NN-site IN-in NN-secret ]],
[DT-* JJ-second NN-uranium NN-enrichment NN-site ]],
[VBZ-is IN-for JJ-peaceful NN-purpose ],

[DT-the NN-evidence IN-* PRP-it ], [VBN-* VBN-fabricated IN-by DT-the NNP-us ]

Parse Thicket generalization gives the detailed similarity picture which looks more complete than the pair-wise sentence generalization result above:

[NN-Iran VBG-developing DT-* NN-enrichment NN-site IN-in NN-secret ]
[NN-generalization-<UN/nuclear watchdog> * VB-pass NN-resolution VBG condemning NN- Iran]
[NN-generalization-<Iran/envoy of Iran> Communicative_action DT-the NN-dispute IN-over JJ-nuclear NNS-*
[Communicative_action - NN-work IN-of NN-Iran IN-on JJ-nuclear NNS-weapons]
[NN-generalization <Iran/envoy to UN> Communicative_action NN-Iran NN-nuclear NN-* VBZ-is IN-for JJ-peaceful NN-purpose ],
Communicative_action - NN-generalize <work/develop> IN-of NN-Iran IN-on JJ-nuclear NNS-weapons]*
[NN-generalization <Iran/envoy to UN> Communicative_action NN-evidence IN-against NN Iran NN-nuclear VBN-fabricated IN-by DT-the NNP-us ]
*condemn^proceed* [*enrichment site*] <leads to> *suggest^condemn* [ *work Iran nuclear weapon* ]

One can feel that PT-based generalization closely approaches human performance in terms of finding similarities between texts. To obtain these results, we need to be capable of maintaining coreferences, apply the relationships between entities to our analysis *(subject vs relation-to-this subject)*, including relationships between verbs *(develop* is a partial case of *work).* We also need to be able to identify communicative actions and generalize them together with their subjects according to the specific patterns of speech act theory. Moreover, we need to maintain rhetoric structure relationship between sentences, to generalize at a higher level above sentences.

The focus of this paper will be to introduce parse thicket and their generalization as paragraph-level structured representation. It will be done with the help of the above example. Fig.1 and Fig.2 show the dependency-based parse trees for the above texts T1 and T2. Each tree node has labels as part-of-speech and its form (such as SG for 'single'); also, tree edges are labeled with the syntactic connection type (such as 'composite').

## 3 Introducing Parse Thickets

Is it possible to find more commonalities between these texts, treating parse trees at a higher level? For that we need to extend the syntactic relations between the nodes of the syntactic dependency parse trees towards more general text discourse relations.

Which relations can we add to the sum of parse trees to extend the match? Once we have such relations as "the same entity", "sub-entity", "super-entity" and anaphora, we can extend the notion of phrase to be matched between texts. Relations between the nodes of parse trees which are other than syntactic can merge phrases from different

sentences, or from a single sentence which are not syntactically connected. We will refer to such extended phrases as *thicket phrases.*

If we have to parse trees $P_1$ and $P_2$ of text $T_1$, and an arc for a relation $r$

$r$: $P_{1j} \rightarrow P_{2j}$ between the nodes $P_{1j}$ and $P_{2j}$, we can now match $...,P_{1,i-2}$, $P_{1,\,i-1}$, $P_{1,\,i}$, $P_{2,j}$, $P_{2,j+1}$, $P_{2,j+2}$, ... of $T_1$ against a chunk of a single sentence of merged chunks of multiple sentences from $T_2$.

### 3.1 Phrase-level generalization

Although the generalization is defined as maximum common sub-trees, its computation is based on matching phrases. To generalize a pair of sentences, we perform chunking and extract all noun, verb, prepositional and other types of phrases from each sentence. Then we perform generalization for each type of phrases, attempting to find a maximum common sub-phrase for each pair of phrases of the same type. The resultant phrase-level generalization can then be interpreted as paths in resultant common sub-trees [3].

Generalization of parse thickets, being a maximal common sub-graph (sub-parse thicket) can be computed at the level of phrases as well, as a structure containing a maximal common sub-phrases. However, the notion of phrases is extended now: *thicket phrases* can contain regular phrases from different sentences. The way these phrases are extracted and formed depend on the source of non-syntactic link between words in different sentences: thicket phrases are formed in a different way for communicative actions and RST relations. Notice that the set of regular phrases for a parse thicket is a sub-set of the set of thicket phrases (all phrases extracted for generalization). Because of this richer set of phrases for generalization, the parse thicket generalization is richer than the pair-wise thicket generalization, and can better tackle variety in phrasings and writing styles, as well as distribution of information through sentences.

### 3.2 Algorithm for forming thicket phrases for generalization

We will now outline the algorithm of forming thicket phrases. Most categories of thicket arcs will be illustrated below.

For each sentence $S$ in a paragraph $P$
  Form a list of previous sentences in a paragraph $S_{prev}$
  For each word in the current sentence:
    - If this word is a *pronoun*: find all nouns or noun phrases in the $S_{prev}$ which are
      * The same entities (via anaphora resolution)
    - If this word is a *noun*: find all nouns or noun phrases in the $S_{prev}$ which are
      * The same entities (via anaphora resolution)
      * Synonymous entity
      * Super entities
      * Sub and sibling entities

- If this word is a *verb*:
  * If it is a communicative action:
    Form the phrase for its subject $VBCA_{phrase}$, including its verb phrase Vph
    Find a preceding communicative action $VBCA_{phrase0}$ from $S_{prev}$ with its subject
    and form a thicket phrase [$VBCA_{phrase}$, $VBCA_{phrase0}$]
  * If it indicates RST relation
    Form the phrase for the pair of phrases which are the subjects [$VBRST_{phrase1}$, $VBRST_{phrase2}$], of this RST relation, $VBRST_{phrase1}$ belongs to $S_{prev}$.

Notice the three categories of the formed thicket phrases:

- Regular phrases;
- Thicket phrases;
- SpActT phrases;
- CA phrases.

Once we collected the thicket phrases for texts T1 and T2, we can do the generalization. When we generalize thicket phrases from various categories, the following constraints should be taken into account:

| | Regular phrases | Entity-based thicket phrases | RST-based thicket phrases | SpActT-based thicket phrases |
|---|---|---|---|---|
| Regular phrases | Obeying phrase type + | + | + | + |
| Entity-based thicket phrases | + | + | - | - |
| RST-based thicket phrases | | | + | - |
| SpActT-based thicket phrases | | | | + |

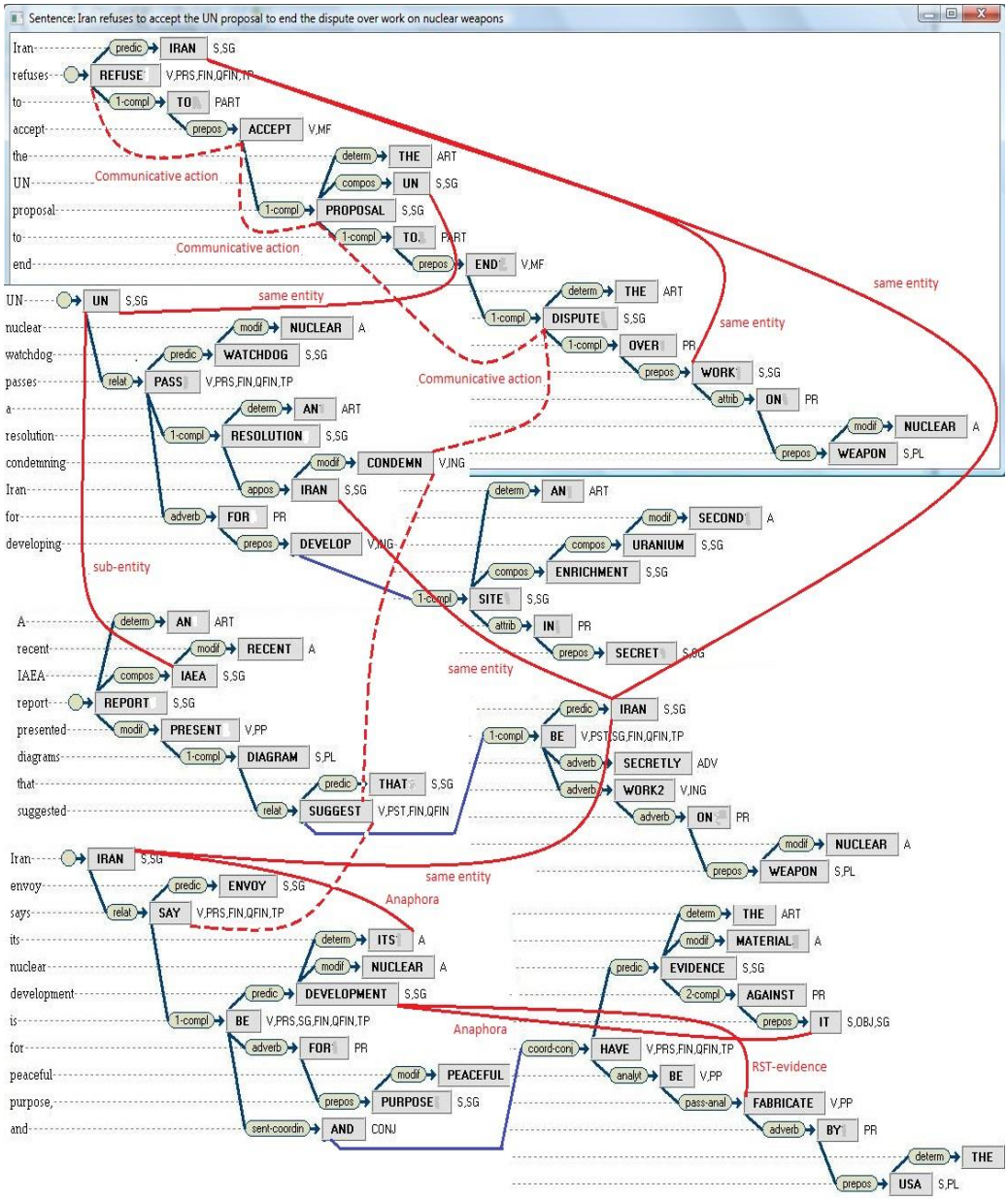



Fig. 1: Parse thicket for text T1.

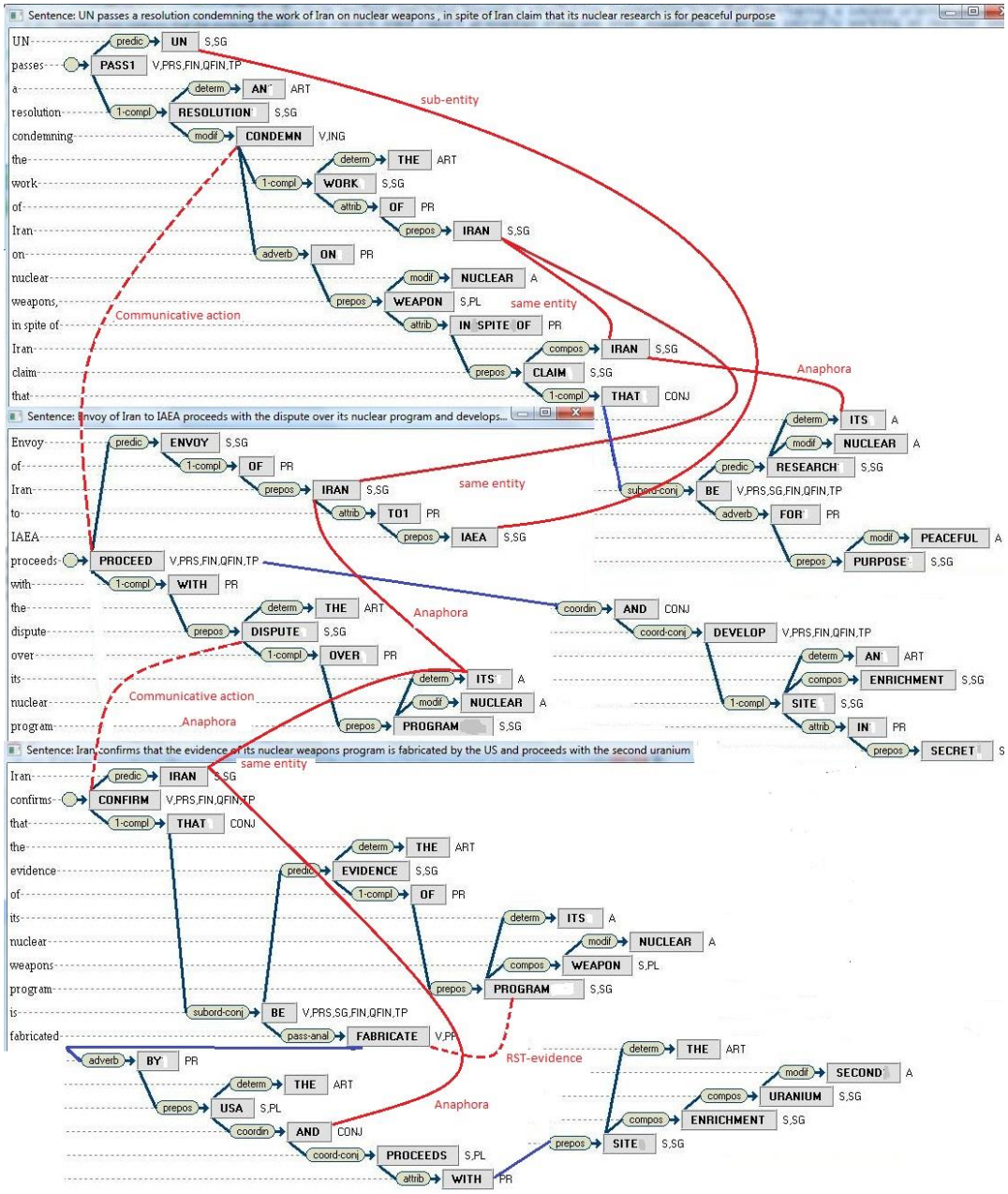



Fig. 2: Parse thicket for text T2.

### 3.3 Sentence-level generalization algorithm

Below we outline the algorithm on finding a maximal sub-phrase for a pair of phrases, applied to the sets of thicket phrases for T1 and T2.

1) Split parse trees for sentences into sub-trees which are phrases for each type: *verb, noun, prepositional* and others; these sub-trees are overlapping. The sub-trees are coded so that information about occurrence in the full tree is retained.
2) All sub-trees are grouped by phrase types.
3) Extending the list of phrases by adding equivalence transformations
4) Generalize each pair of sub-trees for both sentences for each phrase type.
5) For each pair of sub-trees yield an alignment, and then generalize each node for this alignment. For the obtained set of trees (generalization results), calculate the score.
6) For each pair of sub-trees for phrases, select the set of generalizations with highest score (least general).
7) Form the sets of generalizations for each phrase types whose elements are sets of generalizations for this type.
8) Filtering the list of generalization results: for the list of generalization for each phrase type, exclude more general elements from lists of generalization for given pair of phrases.

### 3.4 Arcs of parse thicket based on theories of discourse

We attempt to treat computationally, with a unified framework, two approaches to textual discourse:

- Rhetoric structure theory (RST, Mann et al 1992);
- Speech Act theory (SpActT, [16] 1969);

Although both these theories have psychological observation as foundations and are mostly of a non-computational nature, we will build a specific computational framework for them [4,5]. We will use these sources to find links between sentences to enhance indexing for search. For RST, we attempt to extract an RST relation, and form a thicket phrase around it, including a placeholder for RST relation itself [6]. For SpActT, we use a vocabulary of communicative actions to find their subjects [7], add respective arcs to PT, and form the respective set of thicket phrases.

### 3.5 Generalization based on RST arcs

Two connected clouds on the right of Fig.3 show the generalization instance based on RST relation “RCT-evidence”. This relation occurs between the phrases

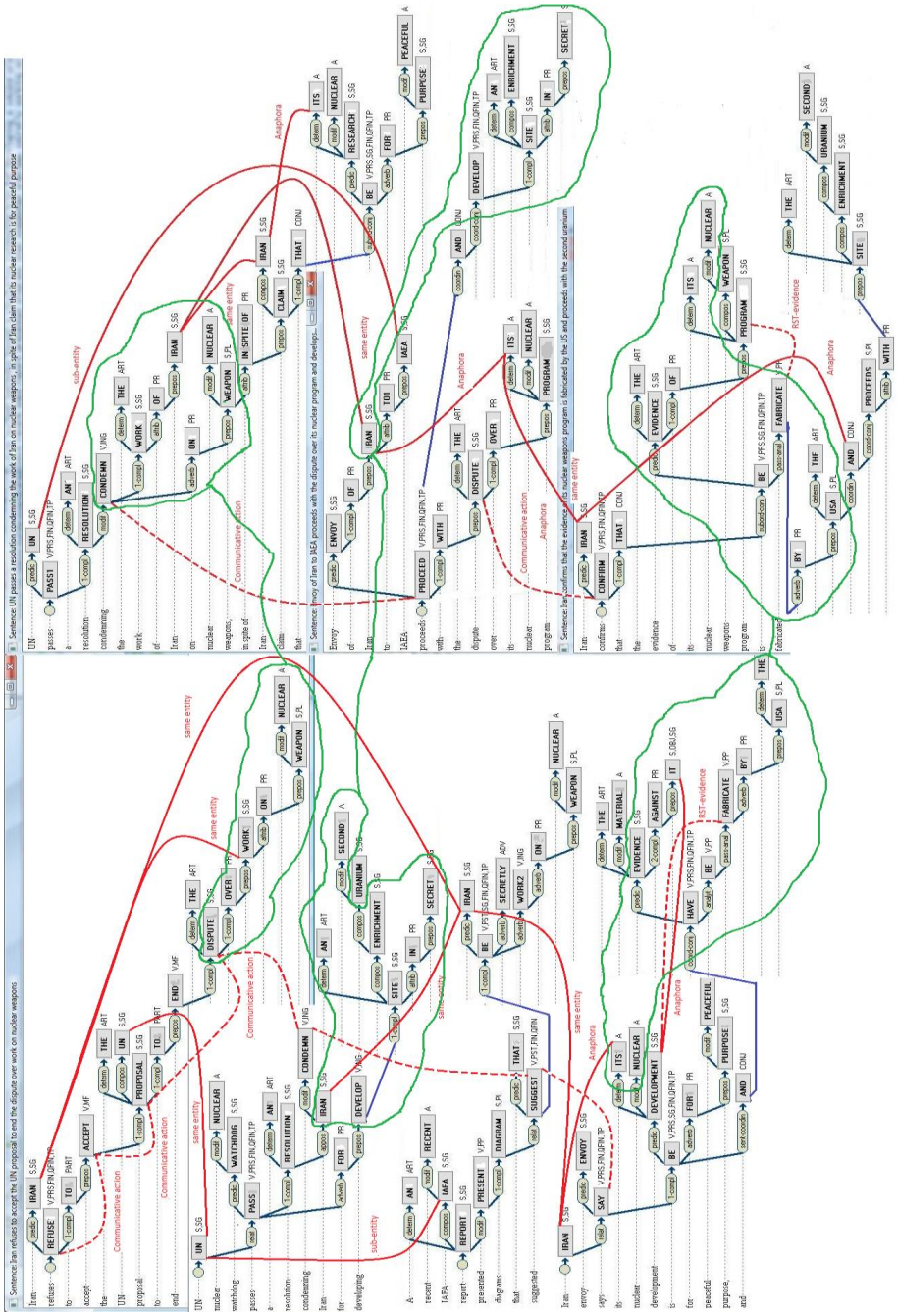


**Fig.3**: Three instances of matching between sub-PTs shown as connected clouds

*evidence-for-what [Iran's nuclear weapon program]* and *what-happens-with-evidence [Fabricated by USA]* on the right-bottom, and

*evidence-for-what [against Iran's nuclear development]* and *what-happens-with-evidence [Fabricated by the USA]* on the right-top.

Notice that in the latter case we need to merge (perform anaphora substitution) the phrase ' *its nuclear development'* with *'evidence against it'* to obtain *'evidence against its nuclear development'.* Notice the arc *it - development,* according to which this anaphora substitution occurred. *Evidence* is removed from the phrase because it is the indicator of RST relation, and we form the subject of this relation to match. Furthermore, we need another anaphora substitution *its- Iran* to obtain the final phrase.

As a result of generalizations of two RST relations of the same sort (evidence) we obtain

*Iran nuclear NNP – RST-evidence – fabricate by USA.*

Notice that we could not obtain this similarity expression by using sentence-level generalization.

Green clouds indicate the sub-PTs of $T_1$ and $T_2$ which are matched. We show three instances of PT generalization.

### 3.6 Generalization based on communicative action arcs

Communicative actions are used by text authors to indicate a structure of a dialogue or a conflict (Searle 1969). Hence analyzing the communicative actions' arcs of PT, one can find implicit similarities between texts. We can generalize:

1. one communicative actions from with its subject from T1 against another communicative action with its subject from T2 (communicative action arc is not used) ;
2. a pair of communicative actions with their subjects from T1 against another pair of communicative actions from T2 (communicative action arcs are used) .

In our example, we have the same communicative actions with subjects with low similarity:

*condemn* ['*Iran for developing second enrichment site in secret'*] vs *condemn* ['*the work of Iran on nuclear weapon*'] ,

or different communicative actions with similar subjects.

Looking on the left of Fig.3 one can observe two connected clouds: the two distinct communicative actions *dispute* and *condemn* have rather similar subjects: '*work on nuclear weapon'*. Generalizing two communicative actions with their subjects follows the rule: generalize communicative actions themselves, and 'attach' the result to generalization of their subjects as regular sub-tree generalization. Two communicative actions can always be generalized, which is not the case for their subjects: if their generalization result is empty, the generalization result of communicative actions with these subjects is empty too. The generalization result here for the case 1 above is:

*condemn^dispute [ work-Iran-on-nuclear-weapon]*.

Generalizing two different communicative actions is based on their attributes and is presented elsewhere [4].

T1 T2

*condemn [second uranium enrichment site ]* ↔ *proceed [develop an enrichment site in secret]*

↓ communicative action arcs ↓

*suggest [Iran is secretly working on nuclear weapon]* ↔ *condemn [the work of Iran on nuclear weapon]*

which results in

*condemn^proceed* [*enrichment site*] <leads to> *suggest^condemn* [ *work Iran nuclear weapon* ]

Notice that generalization

*condemn [second uranium enrichment site ]* ↔ *condemn [the work of Iran on nuclear weapon]*

↓ communicative action arcs ↓

*suggest [Iran is secretly working on nuclear weapon]* ↔ *proceed [develop an enrichment site in secret]*

gives zero result because the arguments of *condemn* from T1 and T2 are not very similar. Hence we generalize the subjects of communicative actions first before we generalize communicative actions themselves.

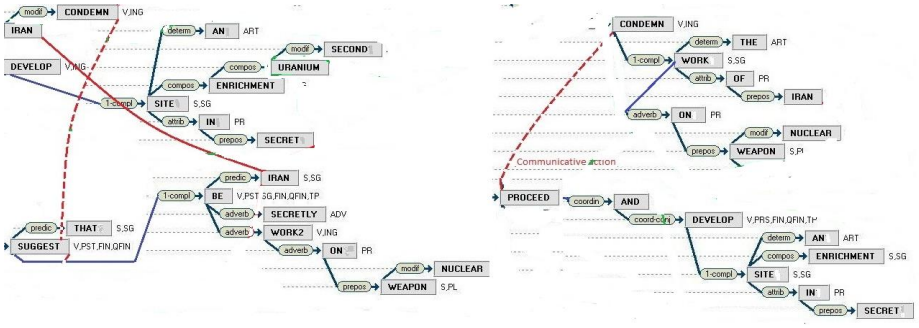



**Fig.4**: A fragment of PT showing the mapping for the pairs of communicative actions

## 4 Preliminary Evaluation of Parse Thicket generalization

Parse forests and their generalizations are important for domain-independent text relevance assessment. In our earlier studies we explored generalization of a PT against a single sentence: this is the case of question answering [4]. To find the best answers, we assess the similarity between the question and candidate answers represented as PTs [5] in the settings of eBay product search. In this section, we provide evaluation of these reduced cases of PT generalization, which can serve as a preliminary evaluation for the general case of PT generalization.

| Query | Answer | Relevancy of baseline Yahoo search, %, averaging over 20 searches | Relevancy of baseline Bing search, %, averaging over 20 searches | Relevancy of re-sorting by pair-wise sentence generalization, % averaging over 40 searches | Relevancy of re-sorting by forest generalization based on RST, % averaging over 20 searches | Relevancy of re-sorting by forest generalization based on | Relevancy of re-sorting by hybrid RST+SpActT forest generalization % averaging over 40 | Relevancy improvement for parse thicket approach, comp. to pair-wise generalization |
|---|---|---|---|---|---|---|---|---|
| 3-4 word phrases | 1 comp. sentence | 81.7 | 82.4 | 86.6 | 88.0 | 87.2 | 91.3 | 1.054 |
| | 2 sent | 79.2 | 79.9 | 82.6 | 86.2 | 84.9 | 89.7 | 1.086 |
| | 3 sent | 76.7 | 75.0 | 79.1 | 85.4 | 86.2 | 88.9 | 1.124 |
| | Average | 79.2 | 79.1 | 82.8 | 86.5 | 86.1 | 90.0 | 1.087 |
| 5-10 word phrases | 1 comp. sentence | 78.2 | 77.7 | 83.2 | 87.2 | 84.5 | 88.3 | 1.061 |
| | 2 sent | 76.3 | 75.8 | 80.3 | 82.4 | 83.2 | 87.9 | 1.095 |
| | 3 sent | 74.2 | 74.9 | 77.4 | 81.3 | 80.9 | 82.5 | 1.066 |
| | Average | 76.2 | 76.1 | 80.3 | 83.6 | 82.9 | 86.2 | 1.074 |
| 1 sentence | 1 comp. sent | 77.3 | 76.9 | 81.1 | 85.9 | 86.2 | 88.9 | 1.096 |
| | 2 sent | 74.5 | 73.8 | 78. | 82.5 | 83.1 | 86.3 | 1.101 |
| | 3 sent | 71.3 | 72.2 | 76.5 | 80.7 | 81.2 | 83.2 | 1.088 |
| | Average | 74.4 | 74.3 | 78.7 | 83.0 | 83.5 | 86.1 | 1.095 |
| 2 sentences | 1 comp. sent | 75.7 | 76.2 | 82.2 | 87.0 | 83.2 | 83.4 | 1.015 |
| | 2 sent | 73.1 | 71.0 | 76.8 | 82.4 | 81.9 | 82.1 | 1.069 |
| | 3 sent | 69.8 | 72.3 | 75.2 | 80.1 | 79.6 | 83.3 | 1.108 |
| | Average | 72.9 | 73.2 | 78.1 | 83.2 | 81.6 | 82.9 | 1.062 |
| 3 sentences | 1 sentence | 73.6 | 74.2 | 78.7 | 85.4 | 83.1 | 85.9 | 1.091 |
| | 2 sentences | 73.8 | 71.7 | 76.3 | 84.3 | 83.2 | 84.2 | 1.104 |
| | 3 sentences | 67.4 | 69.1 | 74.9 | 79.8 | 81.0 | 84.3 | 1.126 |
| | Average | 71.6 | 71.7 | 76.6 | 83.2 | 82.4 | 84.8 | 1.107 |
| Average for all Query and Answer type | | | | | | | | 1.085 |

**Table 1**: Evaluation results for search where answers occur in different sentences.

Discovering trivial (in terms of search relevance) links between different sequences (such as coreferences) is not as important for search as finding more implicit links provided by text discourse theories. We separately measure search relevance when PT is RST-based and SpActT-based. Since these theories are the main sources for establishing non-trivial links between sentences, we limit ourselves to measuring the contributions of these sources of links. Our hybrid approach includes both these sources for links. We consider all cases of questions (phrase, one, two, and three sentences) and all cases of search results occurrences (compound sentence, two, and three sentences) and measured how PT improved the search relevance, compared to original search results ranking averaged for Yahoo and Bing.

One can see (Table 1) that even the simplest cases of short query and a single compound sentence gives more than 5% improvement. PT-based relevance improvement stays within 7-9% as query complexity increases by a few keywords, and then increases to 9-11% as query becomes one-two sentences. For the same query complexity, naturally, search accuracy decreases when more sentences are required for answering this query. However, contribution of PTs does not vary significantly with the number of sentences the answer occurs in (two or three).

While single-sentence syntactic match gives 5.6% improvement [4] multi-sentences parse thickets provides 8.7% for the comparable query complexity (5.4% for single-sentence answer) and up to 10% for the cases with more complex answers. One can see that parse thicket improves single sentence syntactic generalization by at least 2%. On average through the cases of Table 1, parse thickets outperforms single sentence syntactic generalization by 6.7%, whereas RST on its own gives 4.6% and SpActT-4.0% improvement respectively. Hybrid RST + SpActT gives 2.1% improvement over RST-only and 2.7% over SpActT only. We conclude that RST links compliment SpActT links to properly establish relations between entities in sentences for the purpose of search.

## 5 Conclusions

In this study we introduced the notion of syntactic generalization to learn from parse trees for a pair of sentences, and extended it to learning parse thickets for two paragraphs. Parse thicket is intended to represent syntactic structure of text as well as a number of semantic relations for the purpose of indexing for search. To accomplish this, parse thicket includes relations between words in different sentences, such that these relations are essential to match queries with portions of texts to serve as an answer.

We considered the following sources of relations between words in sentences: coreferences, taxonomic relations such as sub-entity, partial case, predicate for subject etc., rhetoric structure relation and speech acts. We demonstrated that search relevance can be improved, if search results are subject to confirmation by parse thicket generalization, when answers occur in multiple sentences.

Traditionally, machine learning of linguistic structures is limited to keyword forms and frequencies. At the same time, most theories of discourse are not computa-

tional, they model a particular set of relations between consecutive states. In this work we attempted to achieve the best of both worlds: learn complete parse tree information augmented with an adjustment of discourse theory allowing computational treatment.

Graphs have been used extensively to formalize ranking of NL texts [13]. Graph-based ranking algorithms are a way of deciding the importance of a vertex withina graph, based on global information recursively drawn from the entire graph. The basic idea implemented by a graph-based ranking model is that of "voting": when one vertex links to another one, it is basically casting a vote for that other vertex. In the current papers graphs are used for representation of meaning rather than for ranking; the latter naturally appears based on the similarity score.

We believe this is a pioneering study in learning a union of parse trees. Instead of using linguistic information of individual sentences, we can now compute text similarity at the level of paragraphs. We plan to extend the functionality of the similarity component of OpenNLP [14] by the contribution of PT-based algorithms.

## References

1. Collins, M., and Duffy, N. 2002. Convolution kernels for natural language. In Proceedings of NIPS, 625–632.
2. Galitsky, B. Natural Language Question Answering System: Technique of Semantic Headers. Advanced Knowledge International, Australia (2003).
3. Galitsky, B., Josep Lluis de la Rosa, Gábor Dobrocsi. Inferring the semantic properties of sentences by mining syntactic parse trees. Data & Knowledge Engineering. Volume 81-82, November (2012) 21-45.
4. Galitsky, B., Daniel Usikov, Sergei O. Kuznetsov: Parse Thicket Representations for Answering Multi-sentence questions. 20th International Conference on Conceptual Structures, ICCS 2013 (2013).
5. Galitsky, B., G. Dobrocsi, J.L. de la Rosa, Kuznetsov, S.O.: From Generalization of Syntactic Parse Trees to Conceptual Graphs, in M. Croitoru, S. Ferré, D. Lukose (Eds.): Conceptual Structures: From Information to Intelligence, 18th International Conference on Conceptual Structures, ICCS 2010, Lecture Notes in Artificial Intelligence, vol. 6208, pp. 185-190.(2010)
6. Galitsky, B., Gabor Dobrocsi, Josep Lluís de la Rosa, Sergei O. Kuznetsov: Using Generalization of Syntactic Parse Trees for Taxonomy Capture on the Web. 19th International Conference on Conceptual Structures, ICCS 2011: 104-117 (2011).
7. Galitsky, B., Kuznetsov SO Learning communicative actions of conflicting human agents. J. Exp. Theor. Artif. Intell. 20(4): 277-317 (2008).
8. Galitsky,B., Machine Learning of Syntactic Parse Trees for Search and Classification of Text. Engineering Application of AI, http://dx.doi.org/10.1016/j.engappai.2012.09.017, (2012).

9. Ganter, B, Kuznetsov SO Pattern Structures and Their Projections. In: Conceptual Structures: Broadening the Base. Lecture Notes in Computer Science Volume 2120, 2001, pp 129-142.
10. Haussler, D. 1999. Convolution kernels on discrete structures.
11. Hensman, S. and Dunnion, J. Automatically building conceptual graphs using VerbNet and WordNet. International Symposium on Info and Comm. tech, Las Vegas, Nevada, June 16–18, 2004.
12. Marcu, D. (1997) 'From Discourse Structures to Text Summaries', in I. Mani and M.Maybury (eds) Proceedings of ACL Workshop on Intelligent Scalable Text Summarization, pp. 82–8, Madrid, Spain.
13. Mihalcea R., and Tarau P. 2004 TextRank: Bringing Order into Texts. Empirial Methods in NLP 2004.Punyakanok, V., Roth, D., & Yih, W. (2004). Mapping dependencies trees: an application to question answering. In: Proceedings of AI & Math, Florida, USA.
14. OpenNLP 2013. apache.org/opennlp/documentation/manual/opennlp.htm.
15. Polovina S. and John Heaton, "An Introduction to Conceptual Graphs," AI Expert, pp. 36-43, 1992.
16. Searle, John. 1969. Speech acts: An essay in the philosophy of language. Cambridge, England: Cambridge University.
17. Sowa JF, Eileen C. Way: Implementing a Semantic Interpreter Using Conceptual Graphs. IBM Journal of Research and Development 30(1): 57-69 (1986)
18. Sowa JF, Information Processing in Mind and Machine. Reading, MA: Addison-Wesley Publ., 1984.
19. Sun, J., Min Zhang, Chew Lim Tan. Tree Sequence Kernel for Natural Language. AAAI-25, 2011.
20. Zhang, M.; Che, W.; Zhou, G.; Aw, A.; Tan, C.; Liu, T.; and Li, S. 2008. Semantic role labeling using a grammar-driven convolution tree kernel. IEEE transactions on audio, speech, and language processing 16(7):1315–1329.

# Classification by Selecting Plausible Formal Concepts in a Concept Lattice

Madori Ikeda and Akihito Yamamoto

Graduate School of Informatics, Kyoto University, Kyoto 606-8501, Japan
m.ikeda@iip.ist.i.kyoto-u.ac.jp, akihiro@i.kyoto-u.ac.jp

**Abstract.** We propose a classification method using a concept lattice, and apply it to thesaurus extension. In natural language processing, solving a practical task by extending many thesauri with a corpus is time-consuming. The task can be represented as classifying a set of test data for each of many sets of training data. The method enables us to decrease the time-cost by avoiding feature selection, which is generally performed for each pair of a set of test data and a set of training data. More precisely, a concept lattice is generated from only a set of test data, and then each formal concept is given a score by using a set of training data. The score represents plausibleness as neighbors of an unknown object, and the unknown object classified into classes to which its neighbors belong. Therefore, once we make the lattice, we can classify test data for each set of training data by only scoring, which has a small computational cost. By experiments using practical thesauri and corpora, we show that our method classifies more accurately than $k$-nearest neighbor algorithm.

**Keywords:** Formal Concept, Concept Lattice, Classification, Thesaurus Extension

## 1 Introduction

In this paper, we propose a method for classifying data that generates a *concept lattice* and selects appropriate *formal concepts* in the lattice. The method enables us to avoid *feature selection* superficially in classification by securing storage enough to maintain both the selected concepts and redundant concepts. This contributes to saving time in solving practical problems concerning with a great variety of large data, e.g. *thesaurus extension*.

Classification can be divided into *feature selection* and classification with the selected features. Selecting features, which affects classification results, is very important, and many methods have been proposed for it [6, 14]. Generally, the selection is executed for a pair of a set of test data and a set of training data. Moreover, the selection is time-consuming when the size of these data is large or many noise are contained in them, as well as when test data are assumed to be classified into multi-classes. Therefore, classifying raw and large test data for each of many sets of training data can be costly from a computational point of view. Our method can overcome the problem. The method generates a concept lattice from a set of test data in advance by following the mathematical

definitions naïvely. When a set of training data is given, the method gives each formal concept a *score* by simple calculation using the training data. The score represents plausibleness of the concept as a set of *neighbors* of an unknown object to be classified. Then the method selects some of the concepts based on the score and finds the neighbors. Each unknown object is classified into classes to which its neighbors belong. The method is thus faster because it uses the same concept lattice for every set of training data without feature selection. In addition, we can easily maintain a concept lattice updated with novel test data using well known methods [4, 17, 19]. Storing such a lattice can be costly, since the lattice must store both concepts that end up being selected or not. We claim this disadvantage can be mitigated by the low cost of memory storage.

We apply our method to the problem of thesaurus extension. *Thesauri* are semantic dictionaries of *terms*, and many kinds of thesauri are available now. In almost of all thesauri, each terms have several semantic definitions, and the definitions of terms often vary from a thesaurus to another thesaurus. *Corpora* are also linguistic resources of another type that consist of sentences in a natural language, and recently some of them contain huge amount of sentences with many kinds of characteristics generated and attached to them by applying parsers and syntactic analyzers. *Thesaurus extension* can be regarded as classification and has been researched in the area of NLP (natural language processing) [1, 9, 18]. As classification, a thesaurus is a set of training data, and a corpus is a set of test data. Many proposed methods calculate similarities among terms by using features of them that are selected from the characteristics contained in a corpus. Then an unregistered term is put into the original thesaurus properly by finding registered terms similar to the unregistered term. This is a practical way of the extension because it is so easy to acquire many syntactic characteristics of terms for classifying unregistered terms semantically. However, many thesauri to be extended exist, and the selected features for a thesaurus might not be useful for another thesaurus. Moreover, these linguistic resources are often updated. These methods do not take the practical problems in NLP into account. Our method does not depend on feature selection, but generates a concept lattice without training data, and is robust to update. We apply thesaurus extension to two thesauri and two corpora that are freely available.

This paper is organized as follows. First, we introduce the definitions of formal concepts and concept lattices in the next section. In Section 3, we explain our classification method based on the definitions, and we compare the method with related works in Section 4. In Section 5, we define thesaurus extension in terms of classification, and show our experimental results. Conclusions are placed in Section 6.

## 2 Formal Concepts and Concept Lattices

We introduce the definitions of *formal concepts* and *concept lattices* according to [5, 7] with a running example.

**Table 1.** An example context $K_0 = (G_0, M_0, I_0)$

| | $m_1$ | $m_2$ | $m_3$ | $m_4$ | $m_5$ | $m_6$ | $m_7$ |
|---|---|---|---|---|---|---|---|
| $g_1$ | × | × | | | | | |
| $g_2$ | × | × | | × | | | |
| $g_3$ | × | × | | × | | | |
| $g_4$ | | × | | × | | × | |
| $g_5$ | | × | | | × | × | |
| $g_6$ | | × | | | × | × | |
| $g_7$ | | | × | | × | | × |

**Definition 1 ([7])** Let $G$ and $M$ be mutually disjoint finite sets, and $I \subseteq G \times M$. Each element of $G$ is called an *object*, and each element of $M$ is called an *attribute*, and $(g, m) \in I$ is read as the object $g$ has the attribute $m$. A triplet $(G, M, I)$ is called a *formal context* (context for short).

**Example 1** We introduce a context $K_0 = (G_0, M_0, I_0)$ as a running example where $G_0 = \{g_1, g_2, ..., g_7\}$, $M_0 = \{m_1, m_2, ..., m_7\}$, and every element of $I_0$ is represented with a cross in Table 1. For example, the object $g_4$ has the attributes $m_2$, $m_4$, and $m_6$.

**Definition 2 ([7])** For a context $(G, M, I)$, subsets $A \subseteq G$ and $B \subseteq M$, we define $A^I = \{\, m \in M \mid \forall g \in A, (g, m) \in I \,\}$, $B^I = \{\, g \in G \mid \forall m \in B, (g, m) \in I \,\}$. A *formal concept* of the context is a pair $(A, B)$ where $A^I = B$ and $A = B^I$.

**Definition 3 ([7])** For a formal concept $c = (A, B)$, $A$ and $B$ are called the *extent* and the *intent* respectively, and we let $\mathrm{Ex}(c) = A$ and $\mathrm{In}(c) = B$. For arbitrary formal concepts $c$ and $c'$, there is an order $c \leq c'$ iff $\mathrm{Ex}(c) \subseteq \mathrm{Ex}(c')$ (or equally $\mathrm{In}(c) \supseteq \mathrm{In}(c')$).

**Definition 4 ([7])** The set of all formal concepts of a context $K = (G, M, I)$ with the order $\leq$ is denoted by $\underline{\mathfrak{B}}(G, M, I)$ (for short, $\underline{\mathfrak{B}}(K)$) and is called the *concept lattice* of $K$. For a concept lattice, the least concept is called the *bottom* and is denoted by $\bot$, and the greatest concept is called the *top* and is denoted by $\top$.

**Example 2** The concept lattice $\underline{\mathfrak{B}}(K_0)$ of the context $K_0 = (G_0, M_0, I_0)$ given in Table 1 is shown in Figure 1. Each circle represents a formal concept $c \in \underline{\mathfrak{B}}(K_0)$ with $\mathrm{Ex}(c)$ and $\mathrm{In}(c)$ on its side. The gray concept and numbers called *scores* beside concepts are explained in Section 3. In the figure, each edge represents an order $\leq$ between two concepts, and the greater concept is drawn above, and transitional orders are omitted. In the lattice, the concept $c_1$ is the top and the concept $c_{11}$ is the bottom.

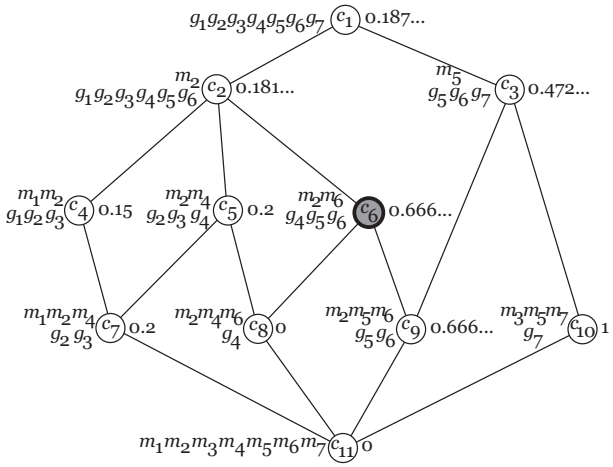



**Fig. 1.** A concept lattice $\underline{\mathfrak{B}}(K_0)$ with scores

**Definition 5 ([7])** For a concept lattice $\underline{\mathfrak{B}}(G, M, I)$, the formal concept $\gamma g = (\{ g \}^{II}, \{ g \}^{I})$ of an object $g \in G$ is called the *object concept*.

**Definition 6 ([7])** For every formal concept $c \in \underline{\mathfrak{B}}(K)$, the subset of formal concepts $\{ c' \in \underline{\mathfrak{B}}(K) \mid c' \geq c \}$ is denoted by $\uparrow c$ and called the *principal filter* generated by $c$.

**Definition 7 ([5])** Let $S$ be an ordered set and let $x, y \in S$. We say $x$ is *covered* by $y$ if $x < y$ and $x \leq z < y$ implies $x = z$.

**Definition 8** For a concept lattice $\underline{\mathfrak{B}}(K)$, a *path* is a string of formal concepts $c_0, c_1, ..., c_n$ satisfying that $c_i$ is covered by $c_{i+1}$ for every $i \in [0, n-1]$, and its *length* is $n$.

**Example 3** For the concept lattice $\underline{\mathfrak{B}}(K_0)$ shown in Figure 1, $\gamma g_4 = c_8$ and $\uparrow \gamma g_4 = \{ c_1, c_2, c_5, c_6, c_8 \}$. Length of the longest path from $\perp$ to $\top$ is four, and length of the shortest path from $\perp$ to $\top$ is three.

In addition, several algorithms have been proposed [4, 17, 19] in order to update a concept lattice $\underline{\mathfrak{B}}(K)$ when a new object or a new attribute is added to a context $K = (G, M, I)$, e.g. $K$ turns into $(G', M', I')$ where $G' \supset G$, $M' \supset M$, and $I' \supset I$. Thus we can easily modify a concept lattice by using these algorithms.

## 3 Classification with a Concept Lattice

We illustrate our classification method after formalizing classification problems.

**Table 2.** An example training set $\tau_0 = (T_0, \mathcal{L}_0)$

| $g \in T_0$ | $g_1$ | $g_2$ | $g_3$ | $g_5$ | $g_6$ | $g_7$ |
|---|---|---|---|---|---|---|
| $\mathcal{L}_0(g)$ | $\{ l_1, l_2 \}$ | $\{ l_2, l_3, l_4 \}$ | $\{ l_4, l_5, l_6 \}$ | $\{ l_1, l_6, l_7 \}$ | $\{ l_6, l_7 \}$ | $\{ l_1, l_7, l_8 \}$ |

**Definition 9** We let $L$ be a finite set that is disjoint with both $G$ and $M$, and every element of $L$ is called a *label*. We assume a function $\mathcal{L}_* : G \to 2^L$ as a target classification rule, and $\mathcal{L}_*(g)$ is called the set of *labels* of $g$.

Each label $l \in L$ represents a class. Note that, for every object $g \in G$, the value of $\mathcal{L}_*(g)$ might not be a singleton and might share some labels with the value of $\mathcal{L}_*(g')$ of another object $g' \in G$, i.e. $\mathcal{L}_*(g) \cap \mathcal{L}_*(g') \neq \emptyset$. Therefore this is a multi-class classification problem and is regarded as an extension of the binary-class classification problems [11–13].

**Definition 10** For a subset $T \subseteq G$ and a function $\mathcal{L} : T \to 2^L$ satisfying that $\mathcal{L}(g) = \mathcal{L}_*(g)$ if $g \in T$ , a pair $(T, \mathcal{L})$ is called a *training set*. For a training set $(T, \mathcal{L})$, every object $g \in G$ is called an *unknown object* if $g \notin T$, otherwise it is called a *known object*.

**Example 4** A training set $\tau_0 = (T_0, \mathcal{L}_0)$ where $T_0 = \{ g_1, g_2, g_3, g_5, g_6, g_7 \}$ and $\mathcal{L}_0 : T_0 \to 2^{\{ l_1, l_2, \ldots, l_8 \}}$ is shown in Table 2. The object $g_4$ is excluded from $G_0$ of the context $K_0 = (G_0, M_0, I_0)$ given in Example 1 and is unknown for $\tau_0$.

Classification problems can be defined as obtaining a function $\hat{\mathcal{L}} : G \to 2^L$, and a classification is *successful* when a function $\hat{\mathcal{L}}$ such that $\forall g \in G. \hat{\mathcal{L}}(g) = \mathcal{L}_*(g)$ is obtained from a given training set $(T, \mathcal{L})$.

Our method is designed for classifying only test data that can be expressed as a context $(G, M, I)$ and consists of the following steps.

1. constructing the concept lattice $\underline{\mathfrak{B}}(K)$ of a context $K = (G, M, I)$,
2. calculating *scores* of formal concepts using a given training set $\tau = (T, \mathcal{L})$,
3. finding the set of *neighbors* for each unknown object $u \in G \setminus T$ based on the scores, and
4. deciding a function $\hat{\mathcal{L}}$ by referring $\mathcal{L}(g)$ for every known object $g \in T$.

The first step is achieved by simply following the definitions of formal concepts.

In order to find the neighbors, formal concepts are given *scores*, and some of the scored concepts are extracted based on their score. The score of every formal concept is a real number calculated with a training set, and it changes for another training set.

**Definition 11** For every formal concept $c \in \underline{\mathfrak{B}}(K)$ and a training set $\tau = (T, \mathcal{L})$, we define $\mathrm{Ex}(c, \tau) = \mathrm{Ex}(c) \cap T$.

**Definition 12** For every formal concept $c \in \underline{\mathfrak{B}}(K)$ and a training set $\tau = (T, \mathcal{L})$, we define $\sigma(c, \tau)$ as a real number in $[0, 1]$ and call it the *score* of the concept $c$ under the training set $\tau$. The value $\sigma(c, \tau)$ is calculated as follows:

$$\sigma(c,\tau) = \begin{cases} 0 & \text{if } |\mathrm{Ex}(c,\tau)| = 0, \\ 1 & \text{if } |\mathrm{Ex}(c,\tau)| = 1, \text{ and} \\ \dfrac{\sum_{i=1}^{|\mathrm{Ex}(c,\tau)|-1} \sum_{j=i+1}^{|\mathrm{Ex}(c,\tau)|} \mathrm{sim}(g_i, g_j)}{\binom{|\mathrm{Ex}(c,\tau)|}{2}} & \text{otherwise,} \end{cases}$$

where $\mathrm{sim}(g_i, g_j) = \dfrac{|\mathcal{L}(g_i) \cap \mathcal{L}(g_j)|}{|\mathcal{L}(g_i) \cup \mathcal{L}(g_j)|}$.

The function sim calculates similarity between know objects $g_i$ and $g_j$, and the function $\sigma$ calculates the average of similarities among objects in $\mathrm{Ex}(c, \tau)$. The purpose of defining the score $\sigma(c, \tau)$ is to estimate similarity among all objects in $\mathrm{Ex}(c)$ that includes not only known objects but also unknown objects.

After scoring formal concepts, the set of *neighbors* is found for each unknown object based on the scores. The neighbors are known objects extracted from the extent of *plausible* concepts.

**Definition 13** For every object $g \in G$ in a concept lattice $\underline{\mathfrak{B}}(G, M, I)$ and a training set $\tau = (T, \mathcal{L})$, a formal concept $c \in \uparrow \gamma g$ is called *plausible* w.r.t. $g$ and $\tau$ if $\sigma(c, \tau) \geq \sigma(c', \tau)$ for any other concept $c' \in \uparrow \gamma g$ and $|\mathrm{Ex}(c)| \geq |\mathrm{Ex}(c'')|$ for any other concept $c'' \in \uparrow \gamma g$ such that $\sigma(c, \tau) = \sigma(c'', \tau)$. The set of plausible formal concepts w.r.t. $g$ and $\tau$ is denoted by $\mathrm{p}(g, \tau) \subseteq \uparrow \gamma g$.

We intend the score of a formal concept $c$ to represent similarities among objects in $\mathrm{Ex}(c)$. We therefore define objects in $\mathrm{Ex}(c) \ni u$ of the concept $c$ that has the highest score as neighbors of an unknown object $u$. However, it sometimes happens that some formal concepts have the highest score at the same time. In this case, we define a concept $c$ consisting of the largest size $\mathrm{Ex}(c)$ as a plausible concept. This is based on our policy that, among formal concepts that have the same score, the larger the size of $\mathrm{Ex}(c)$ is, the less the concept $c$ has noises in $\mathrm{Ex}(c)$.

**Definition 14** For every unknown object $u \in G \setminus T$ under a concept lattice $\underline{\mathfrak{B}}(G, M, I)$ and a training set $\tau = (T, \mathcal{L})$, a set $\mathrm{N}(u, \tau)$ of *neighbors* is defined as

$$\mathrm{N}(u,\tau) = \bigcup_{c \in \mathrm{p}(u,\tau)} \mathrm{Ex}(c,\tau).$$

At the last step, a function $\hat{\mathcal{L}}$ is constructed as

$$\hat{\mathcal{L}}(g) = \begin{cases} \bigcup_{g' \in \mathrm{N}(g,\tau)} \mathcal{L}(g') & \text{if } g \text{ is unknown for } \tau = (T, \mathcal{L}), \\ \mathcal{L}(g) & \text{otherwise.} \end{cases}$$

In this paper, we employed this simple definition although it could be defined by many ways.

**Example 5** Suppose that we obtained the context $K_0 = (G_0, M_0, I_0)$ given in Example 1 and constructed the concept lattice $\underline{\mathfrak{B}}(K_0)$ as shown in Figure 1 at the first step. Then, suppose that the training set $\tau_0 = (T_0, \mathcal{L}_0)$ shown in Example 4 was given. The score $\sigma(c, \tau_0)$ of every formal concept $c \in \underline{\mathfrak{B}}(K_0)$ under the training set $\tau_0$ can be calculated at the second step and is shown as the number beside each formal concept $c$ in Figure 1. Plausible formal concepts of the unknown object $g_4$ decided as the third step are represented as gray and bold circles in Figure 1. There is only one plausible concept $c_6$, and thus $\mathrm{N}(g_4, \tau_0) = \{ g_5, g_6 \}$. Finally, we can obtain a function $\hat{\mathcal{L}}_0$ at the last step, and $\hat{\mathcal{L}}_0(g_4) = \{ l_1, l_6, l_7 \}$.

## 4 Comparison with Related Works

We concern a task classifying objects in a context for each of many training sets, and we assume that training sets have multi-classes, and that an object might be classified into several classes. This is a practical task in NLP (natural language processing) that is extending many thesauri by using a corpus. Classification results required by every pair of the context and a training set must be different each other. Classification thus needs to be executed for each of the pairs, and the greater the number of the pairs is, the more solving the task is time-consuming. Our research is motivated to save the time required for the task, and our classification method can overcome the problem. The proposed method constructs a concept lattice from a context in advance, and then the method classifies unknown objects of a training set given later. The same concept lattice is used repeatedly in order to classify unknown objects of each training set, and each classification is performed by scoring formal concepts and finding neighbors of unknown objects. This is based on an idea that as many processes requiring no training sets as possible should be executed before training sets are given. Because of this idea, our method is different from some researches.

Learning models based on a concept lattice are proposed in [11–13]. In these researches, a hypothesis (a set of attributes) is constructed from positive and negative examples (objects) by using a formal concept lattice, and it is determined whether an unknown object is positive or not when the hypothesis is acquired. This is a binary-class classification problem, but unknown objects sometimes are not classified when hypotheses are not constructed appropriately in these approaches. Our method certainly classifies unknown objects into malt-classes by scoring formal concepts. In order to classifying data, some approaches generate decision trees from concept lattices [2, 12]. The decision tree is generated for a pair of a context and a training set. Our method however is not intended to generate decision trees because manipulating a concept lattice that is already constructed is not preferable in order to reduce the time for the task we concern.

As classification for a pair of a context and a training set, our method is similar to $k$-NN ($k$-nearest neighbor algorithm) [3] on the point that they find and refer neighbors of an unknown object in order to classify it. However, our method often decreases the number of the neighbors and does not need to be

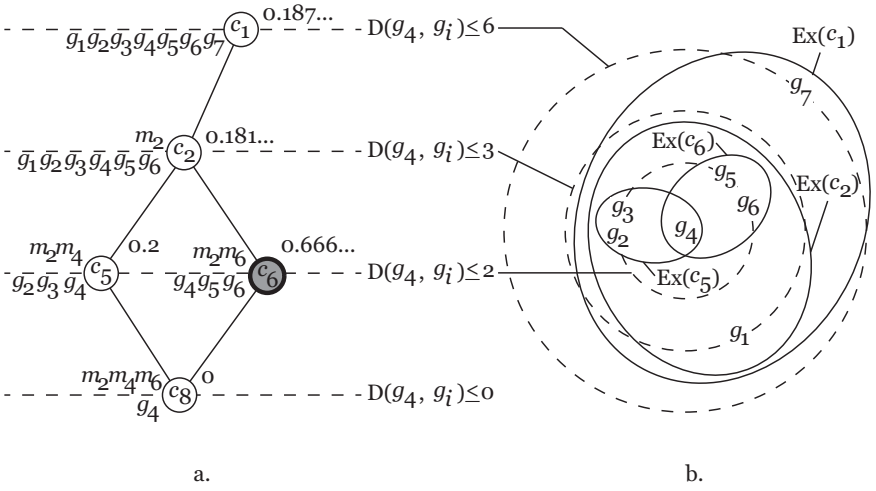



**Fig. 2.** Formal concepts $\uparrow \gamma g_4$ in the concept lattice $\underline{\mathfrak{B}}(K_0)$ with distance

given the number. In this section, we illustrate such differences between the two methods with an example, and we describe that the differences cause our method to classify more accurately.

In the illustration, we use the *symmetric difference* for measuring dissimilarity between two objects.

**Definition 15** For two objects $g, g' \in G$ of a context $(G, M, I)$, we define a distance $\mathrm{D}(g, g')$ between $g$ and $g'$ as

$$\mathrm{D}(g, g') = |\{ g \}^I \cup \{ g' \}^I| - |\{ g \}^I \cap \{ g' \}^I|.$$

This distance is also known as the *Hamming distance* between bit vectors in information theory. Figure 2.a shows a part of the concept lattice $\underline{\mathfrak{B}}(G_0, M_0, I_0)$ that is a set of formal concepts $\uparrow \gamma g_4$ with the order $\leq$, and Figure 2.b shows a space that every object $g \in G_0$ is placed according to the distance $\mathrm{D}(g_4, g)$ from the unknown object $g_4$. From these figures, we observe that each formal concept $c \in \uparrow \gamma g$, as the extent $\mathrm{Ex}(c)$, represents a set of objects located within a certain distance from an object $g$. It is also found that the greater a concept is, the more the concept includes many dissimilar objects, i.e. for every $g \in G$ and $c', c'' \in \uparrow \gamma g$, $\max(\{ \mathrm{D}(g, g') \mid g' \in \mathrm{Ex}(c') \}) \leq \max(\{ \mathrm{D}(g, g'') \mid g'' \in \mathrm{Ex}(c'') \})$ if $c' \leq c''$.

Suppose that we have the context $K_0 = (G_0, M_0, I_0)$ given in Example 1 and the training set $\tau_0 = (T_0, \mathcal{L}_0)$ given in Example 4, and that we have to find neighbors $\mathrm{N}(g_4, \tau_0)$ in order to complete a function $\hat{\mathcal{L}}_0$. As Figure 2.b shows, the known objects $g_2$, $g_3$, $g_5$, and $g_6$ are nearest to the unknown object $g_4$. In adopting $k$-NN, each of the four objects is equally a candidate of a neighbor of $g_4$.

When $k > 4$, we have to find $k-4$ more candidates that are less similar to $g_4$ than $g_2$, $g_3$, $g_5$, $g_6$, and such candidates might be noises and might decrease accuracy of obtained function $\hat{\mathcal{L}}_0$. Otherwise, we need to reject $4-k$ candidates from $g_2$, $g_3$, $g_5$, $g_6$ according to some policy. The rejection also affects the accuracy if values of the function $\mathcal{L}_0$ for the candidates are different. In this case, the values of $\mathcal{L}_0(g_2)$, $\mathcal{L}_0(g_3)$, $\mathcal{L}_0(g_5)$, and $\mathcal{L}_0(g_6)$ are mutually distinct, and the value of $\hat{\mathcal{L}}_0(g_4)$ varies depending on remaining candidates. Therefore, the fixed number $k$ of $k$-NN may lead accuracy of obtained function $\hat{\mathcal{L}}$ to be decreased. Generally, for the purpose of avoiding such problems, feature selection is repeated for each training set in advance.

By contrast, the nearest objects $g_2$, $g_3$, $g_5$, and $g_6$ are divided into two extents $\mathrm{Ex}(c_5) = \{ g_2, g_3, g_4 \}$ and $\mathrm{Ex}(c_6) = \{ g_4, g_5, g_6 \}$ in our method. The concepts $c_5$ and $c_6$ are discriminated by their scores, and $c_6$ is more plausible than $c_5$. Consequently, the objects $g_2, g_3 \in \mathrm{Ex}(c_5)$ are neighbors of the unknown object $g_4$. Therefore, the number of the neighbors is often less than one in $k$-NN. Moreover, the number is depends on the size of the extent of every plausible concept, so the number $k$ is not necessary in our method.

We claim that the process of our method, dividing candidates of neighbors into extents and discriminating them by scores, improves both the *precision* and the *recall* of an obtained function $\hat{\mathcal{L}}$.

**Definition 16** Under a target function $\mathcal{L}_*$ and an obtained function $\hat{\mathcal{L}}$, the *precision* $\mathrm{prec}(g)$ and the *recall* $\mathrm{rec}(g)$ for every object $g \in G$ is defined as

$$\mathrm{prec}(g) = \frac{|\hat{\mathcal{L}}(g) \cap \mathcal{L}_*(g)|}{|\hat{\mathcal{L}}(g)|}, \quad \mathrm{rec}(g) = \frac{|\hat{\mathcal{L}}(g) \cap \mathcal{L}_*(g)|}{|\mathcal{L}_*(g)|}.$$

Generally, a larger number $k$ of candidates of neighbors results in a lower precision and a higher recall. While $k$ is fixed for all unknown objects in $k$-NN, it is flexible for each unknown object in our method. More precisely, our method tries to make a precision higher by making $k$ for an unknown object less, but the method also tries to make a recall higher by making $k$ greater when it can keep a precision high. Thus, the two values are better than ones in $k$-NN. We confirm this assertion by experiments in the next section.

## 5 Thesaurus Extension and Experiments

We cast the problem of *thesaurus extension* as a classification problem to which we apply the method proposed in this paper. A *thesaurus* is a dictionary registering *terms* based on their *senses*. It is common to all thesauri available now that every registered term (known object) is sorted according to some senses (classes). Extending a thesaurus is putting an unregistered term (unknown object) on some proper positions corresponding to senses in the thesaurus. It is a classification problem defined in Section 3 when we regard a training set $\tau = (T, \mathcal{L})$ as an original thesaurus, $T$ as a set of registered terms, $\mathcal{L}(g)$ for every term $g \in T$ as a set of labels identifying its senses, and unknown objects as unregistered terms.

In this section, we compare our method with $k$-NN ($k$-nearest neighbor algorithm) [3] on the point of accuracy by experiments before illustrating resources used in the experiments.

### 5.1 Contexts and Training Sets for Experiments

We prepared two Japanese corpora, a case frame corpus published by Gengo-Shigen-Kyokai (which means "Language Resource Association" in English) [8] and the Google 4-gram [10], and two Japanese thesauri, Japanese WordNet 1.0 [15] and Bunruigoihyo [16]. We generated the set $G_1$ of 7,636 nouns contained by all of these linguistic resources. For our experiments, we constructed a context $K_1 = (G_1, M_1, I_1)$ from the case frame corpus and a context $K_2 = (G_1, M_2, I_2)$ from the Google 4-gram so that they satisfy $I_1 \cap I_2 = \emptyset$. Additionally, we construct the third context $K_3 = (G_1, M_3, I_3)$ where $M_3 = M_1 \cup M_2$ and $I_3 = I_1 \cup I_2$. We also constructed a pair $(G_1, \mathcal{L}_{1*})$ from Japanese WordNet 1.0, and $(G_1, \mathcal{L}_{2*})$ from Bunruigoihyo. We generated ten training sets from each pair in each experiment adopting 10-fold cross validation.

Table 3 shows the statistics for the three concept lattices of the three contexts. Numbers on the row beginning with "max $|\{g\}^{I_i}|$", "min $|\{g\}^{I_i}|$", and "mean $|\{g\}^{I_i}|$" respectively represent the maximum, the minimum, and the mean of the numbers of attributes that an object $g \in G_1$ has. Numbers on the row of "mean $|\mathrm{Ex}(c)|$" represent the mean of $|\mathrm{Ex}(c)|$ of every formal concept $c \in \underline{\mathfrak{B}}(K_i)$ for $i \in \{1, 2, 3\}$. Every numbers on the row beginning with "max height" and "min height" respectively indicate length of the longest and the shortest path from $\bot$ to $\top$. Observing the table, we find that every object has a few attributes in all of the three contexts, and that every formal concept $c \in \underline{\mathfrak{B}}(K_i)$ splits the objects into small groups. In other words, formal contexts constructed from the practical corpora are very sparse, and, for all concepts, not so many objects are needed to score it.

Table 4 shows the statistics for the pairs. Numbers on the row beginning with "max $|\mathcal{L}_{i*}(g)|$", "min $|\mathcal{L}_{i*}(g)|$", and "mean $|\mathcal{L}_{i*}(g)|$" respectively indicate the maximum, the minimum, and the mean of $|\mathcal{L}_{i*}(g)|$ of every object $g \in G_1$ for $i \in \{1, 2\}$. Note that, in both of the two thesauri, many terms have several senses, and many senses are represented by several terms, i.e. $|\mathcal{L}_{i_*}(g)| > 1$ and $\mathcal{L}_{i_*}(g) \cap \mathcal{L}_{i_*}(g') \neq \emptyset$ for many terms $g, g' \in G_1$. They have quite different definitions of terms. Moreover, we have to note that the two thesauri share no identifiers of senses, i.e. $\mathcal{L}_{1_*}(g) \cap \mathcal{L}_{2_*}(g) = \emptyset$ for every term $g \in G_1$. In the remains of this subsection, we describe contents of the contexts and the pairs in detail.

The case frame corpus, which is used for construct the context $K_1$, consists of *case frame structures* that are acquired from about 1.6 billion Japanese sentences on the Web by syntactic analysis. In Japanese, every predicate relates to some nouns with some case terms in a sentence, and such relations are called case frame structures. In $K_1 = (G_1, M_1, I_1)$, every element of $M_1$ is a pair of a predicate and a case term that the predicate relates to a noun in $G_1$ with the case term in the corpus, and every element of $I_1$ is such a relation between a noun in $G_1$ and

**Table 3.** Statistics for the concept lattices

| | $\underline{\mathfrak{B}}(K_1)$ | $\underline{\mathfrak{B}}(K_2)$ | $\underline{\mathfrak{B}}(K_3)$ |
|---|---|---|---|
| $\lvert G_1\rvert$ | 7,636 | 7,636 | 7,636 |
| $\lvert M_i\rvert$ | 19,313 | 7,135 | 26,448 |
| max $\lvert \{ g \}^{I_i} \rvert$ | 17 | 27 | 32 |
| min $\lvert \{ g \}^{I_i} \rvert$ | 1 | 1 | 2 |
| mean $\lvert \{ g \}^{I_i} \rvert$ | 3.85 | 4.70 | 8.55 |
| $\lvert\underline{\mathfrak{B}}(K_i)\rvert$ | 11,960 | 20,066 | 30,540 |
| mean $\lvert \mathrm{Ex}(c)\rvert$ | 2.55 | 6.04 | 4.89 |
| max height | 6 | 9 | 10 |
| min height | 2 | 2 | 2 |

**Table 4.** Statistics for the thesauri

| | $(G_1, \mathcal{L}_{1*})$ | $(G_1, \mathcal{L}_{2*})$ |
|---|---|---|
| $\lvert G_1\rvert$ | 7636 | 7636 |
| $\lvert \bigcup_{g\in G_1} \mathcal{L}_{i*}(g)\rvert$ | 9560 | 595 |
| max $\lvert \mathcal{L}_{i*}(g)\rvert$ | 19 | 9 |
| min $\lvert \mathcal{L}_{i*}(g)\rvert$ | 1 | 1 |
| mean $\lvert \mathcal{L}_{i*}(g)\rvert$ | 2.19 | 2.89 |

**Table 5.** A context from case frame structures

| | ⟨hoeru,ga⟩ | ⟨hoeru,ni⟩ |
|---|---|---|
| inu | × | |
| otoko | | × |

**Table 6.** A context from 4-grams

| | ga otoko ni | hoete | iru |
|---|---|---|---|
| inu | × × × | | |
| otoko | × | × | × |

a pair in $M_1$. For example, in a Japanese sentence "inu ga otoko ni hoete iru(in English, A dog is barking to a man)", the predicate "hoeru(bark)" relates to the noun "inu(dog)" with the case term "ga(be/do)" and also relates to the noun "otoko(man)" with the case term "ni(to)". These relations are represented as (inu,⟨hoeru,ga⟩) and (otoko,⟨hoeru,ni⟩) respectively and are shown in Table 5. Note that the corpus also has the frequency of each relation, and we used only relations satisfying $0.05 \leq (f/n) \leq 0.95$ where $f$ is the frequency of a relation and $n$ is the sum of the frequencies of relations including the same noun that the relation holds.

The Google 4-gram, which is used to construct the context $K_2$, is acquired from about 20 billion Japanese sentences on the Web. A Japanese sentence can be regarded as a string of POSs (part of speech) that are words included in the sentence, and a 4-gram is a string of POSs whose length is four. In $K_2 = (G_1, M_2, I_2)$, every element of $G_1$ is the first POS, and every element of $M_2$ is POS following the first in a sentence, and every element of $I_2$ is a relation of "following". For example, from the same sentence "inu ga otoko ni hoete iru" that is a string of six POSs, we can obtain two 4-grams starting with a noun, "inu ga otoko ni" and "otoko ni hoete iru", and the context shown in Table 6 is obtained. This corpus also has the frequency of each 4-gram, and, in order to construct $(G_1, M_2, I_2)$, we use only 4-grams satisfying the condition $0.05 \leq (f/n) \leq 0.95$ ($f$ is the frequency of a 4-gram and $n$ is the sum of the frequencies of 4-grams containing the same noun the 4-gram holds).

Japanese WordNet 1.0 is used to construct the pair $(G_1, \mathcal{L}_{1*})$, and we use values called *lemmas* in the thesaurus as terms. Bunruigoihyo is used to construct the pair $(G_1, \mathcal{L}_{2*})$, and we use values called *midashi-honntai* (entry) in

**Table 7.** Accuracies of obtained functions $\hat{\mathcal{L}}_1$ and $\hat{\mathcal{L}}_2$

| | | $(G_1, \mathcal{L}_{1*})$ | | $(G_1, \mathcal{L}_{2*})$ | |
|---|---|---|---|---|---|
| | method | precision | recall | precision | recall |
| $K_1 = (G_1, M_1, I_1)$ | our method | 0.039 | 0.274 | 0.164 | 0.533 |
| | 1-NN | 0.026 | 0.024 | 0.103 | 0.103 |
| | 5-NN | 0.007 | 0.036 | 0.031 | 0.150 |
| | 10-NN | 0.004 | 0.038 | 0.016 | 0.169 |
| $K_2 = (G_1, M_2, I_2)$ | our method | 0.007 | 0.079 | 0.028 | 0.248 |
| | 1-NN | 0.007 | 0.007 | 0.027 | 0.027 |
| | 5-NN | 0.002 | 0.013 | 0.014 | 0.070 |
| | 10-NN | 0.002 | 0.018 | 0.010 | 0.100 |
| $K_3 = (G_1, M_3, I_3)$ | our method | 0.030 | 0.072 | 0.132 | 0.250 |
| | 1-NN | 0.009 | 0.009 | 0.039 | 0.039 |
| | 5-NN | 0.004 | 0.018 | 0.017 | 0.085 |
| | 10-NN | 0.002 | 0.024 | 0.011 | 0.116 |

the thesaurus as terms and use values called *bunrui-bango* (class number) in it as senses.

### 5.2 Experimental Results

We carried out experiments to compare our method with $k$-NN ($k$-nearest neighbor algorithm).

In the experiments, both our method and other methods were executed with six combinations of the three concept lattices $\underline{\mathfrak{B}}(K_i)$ for $i \in \{1, 2, 3\}$ and the two pairs $(G_1, \mathcal{L}_{j*})$ for $j \in \{1, 2\}$. We adopted 10-fold cross validation on all combinations. On each combination, the set of objects $G_1$ was split into ten sets $U_l$ for $l \in \{1, 2, ..., 10\}$ that are almost equal about their size. The $l$-th classification was executed for a concept lattice $\underline{\mathfrak{B}}(K_i)$ and a training set $(G_1 \setminus U_l, \mathcal{L}_{j*})$. We used the precision and the recall in order to evaluate accuracy of both methods. The precision and the recall of a method are mutually defined as the means of precisions and recalls of the ten classifications, and the precision and the recall of the $l$-th classification are defined as the means of the values of the obtained function $\hat{\mathcal{L}}_j$ for unknown objects $U_l$. We have to note that our research is intended to solve a task classifying unknown objects in a contest for every one of many training sets, but each classification in the experiments was executed for a pair of a context and a training set.

We compare our method with $k$-NN for $k \in \{1, 5, 10\}$ and show the results in Table 7. In the table, every value is rounded off to the third decimal place. The results shows our method is better than $k$-NN in both the precision and the recall over the combinations. The results shows that we can conclude that this is due to the fact that our method is more flexible than the others on the numbers of neighbors.

## 6 Conclusions

We proposed a classification method that uses a concept lattice and introduces scores of concepts in order to find neighbors of each unknown object. The method tries to make a precision higher by making the number of the neighbors less, but the method also tries to make a recall higher by making the number greater when it can keep a precision high. Moreover, the method does not need feature selection by using the lattice and the scores. We intend the method to apply a task that classifies a set of test data for every one of many sets of training data. The task is practical in NLP, e.g. thesaurus extension By avoiding feature selection, the method has an advantage of time-cost in the task. We made sure that our method classifies unknown objects more accurately than $k$-NN (nearest neighbor algorithm) [3] by experiments applying both of the methods to thesaurus extension.

The classification results given in Figure 7 shows that the accuracy of our method is not enough although it is better than ones of the other method. The results also show that the accuracies vary over combinations of concept lattices and training sets. It is expected that the variation depends especially on the structure of a concept lattice because the structure affects directly what kind of object a formal concept contains in its extent. Therefore analyzing relations among structures of the lattices and classification results would be a future work for improvement in the accuracy.

## References

1. Agirre, E., Ansa, O., Hovy, E., Martinez, D.: Enriching Very Large Ontologies Using the WWW. In: Proc. of the ECAI 2000 Workshop gOntology Learningh (2000)
2. Belohlavek, R., De Baets, B., Outrata, J., Vychodil, V.: Inducing Decision Trees via Concept Lattices. International Journal of General Systems, vol. 38, Num. 4, pp. 455–467 (2009)
3. Belur, V., D.: Nearest Neighbor (NN) Norms: NN Pattern Classification Techniques. IEEE Computer Society Press, Los Alamitos, CA (1991)
4. Choi, V., Huang, Y.: Faster Algorithms for Constructing a Galois Lattice, Enumerating All Maximal Bipartite Cliques and Closed Frequent Sets. In: SIAM Conference on Discrete Mathematics (2006)
5. Davey, B., A., Priestly, H., A.: Introduction to Lattice and Order. Cambridge University Press, Cambridge (2002)
6. Deng, H., Runger, G.: Feature Selection via Regularized Trees. In: Neural Networks (IJCNN), Proc. of the 2012 International Joint Conference on, pp. 1–8. IEEE (2012)
7. Ganter, B., Wille, R.: Formal Concept Analysis: Mathematical Foundations. Springer-Verlag New York, Inc., Secaucus, NJ (1999)
8. Gengo Shigen Kyokai, `http://www.gsk.or.jp`
9. Jun Wang, J., Ge, N.: Automatic Feature Thesaurus Enrichment: Extracting Generic Terms From Digital Gazetteer. In: Digital Libraries, 2006. JCDL '06. Proc. of the 6th ACM/IEEE-CS Joint Conference on, pp. 326–333. IEEE (2006)
10. Kudo, T., Kazawa, H.: Web Japanese N-gram Version 1, Gengo Shigen Kyokai

11. Kuznetsov, S., O.: Complexity of Learning in Concept Lattices from Positive and Negative Examples. Discrete Applied Mathematics, vol. 142, issues. 1–3, pp. 111–125 (2004)
12. Kuznetsov, S., O.: Machine Learning and Formal Concept Analysis. Lecture Notes in Computer Science, vol. 2961, pp. 287–312 (2004)
13. Kuznetsov, S., O.: Mathematical Aspects of Concept Analysis. Journal of Mathematical Sciences, vol. 80, no. 2, pp. 1654–1698 (1996)
14. Lopez, F., G., Torres, M., G., Melian, B., Perez, J., A., M., Moreno-Vega, J., M.: Solving Feature Subset Selection Problem by a Parallel Scatter Search. European Journal of Operational Research, vol. 169, no. 2, pp. 477–489 (2006)
15. Mok, S., W., H., Gao, H., E., Bond, F.: Using Wordnet to Predict Numeral Classifiers in Chinese and Japanese. In: Proc. of the 6th International Conference of the Global WordNet Association (GWC-2012), Matsue (2012)
16. National Institute for Japanese Language and Linguistics, `http://www.ninjal.ac.jp/archives/goihyo`
17. Soldano, H., Ventos, V., Champesme, M., Forge, D.: Incremental Construction of Alpha Lattices and Association Rules. In: Proc. of the 14th International Conference on Knowledge-based and Intelligent Information and Engineering Systems: Part II (KES'10), pp. 351–360. Springer, Heidelberg (2010)
18. Uramoto, N.: Positioning Unknown Words in a Thesaurus by Using Information Extracted from a Corpus. In: Proc. of the 16th Conference on Computational Linguistics (COLING '96), vol. 2, pp. 956–961. Association for Computational Linguistics, Stroudsburg, PA (1996)
19. Valtchev, P., Missaoui, R.: Building Concept (Galois) Lattices from Parts: Generalizing the Incremental Methods. In: Proc. of the ICCS'01, pp. 290–303. Springer, Heidelberg (2001)

# FCA-based Search for Duplicate objects in Ontologies

Dmitry A. Ilvovsky and Mikhail A. Klimushkin

National Research University Higher School of Economics, School of Applied Mathematics and Informational Science
11 Pokrovskiy boulevard, Moscow, Russia
dilv_ru@yahoo.com, klim.mikhail@gmail.com

**Abstract.** A new approach for detecting duplicates in ontology built on real redundant data is considered. This approach is based on transformation of initial ontology into a formal context and processing this context using methods of Formal Concept Analysis (FCA). As a part of a new method we also introduce a new index for measuring similarity between objects in formal concept. We study the new approach on randomly generated contexts and real ontology built for a collection of political news and documents.

## 1 Introduction

One of the most generic ways to represent structured data is an ontology [2]. A common way to build an ontology is its automatic or semi-automatic generation from unstructured data (usually text). The problem of this approach is data redundancy, because different sources of information often contain duplicated information. Detecting and elimination of redundancy directly at the ontology building (or extending) stage (for instance, via pairwise comparison of new objects with existing ones) is not very effective because such an approach significantly increases burden on the expert who makes final decisions. Moreover, in real world redundant data comes to ontology unevenly and it makes sense to eliminate redundancy not every time when an ontology renews but do it at longer intervals. The duration of intervals can be determined by features of a particular domain.

In this work we consider a new method for effective identification of redundancy in data represented by an ontology. This method can be either used as an automatic detector of a list of potential duplicate objects or as a recommendation system. Algorithm is realized via union of closed sets of objects and based on Formal Concept Analysis methods [1].

## 2 Basic definitions

Formal Concept Analysis (FCA) [1] is an applied branch of lattice theory. From data analysis point of view, methods used in Formal Concept Analysis belong

to methods of object-attribute clustering. FCA considers not clusters of objects without their attribute descriptions, but groups of objects and attributes strongly related with each other.

**Definition 1.** A formal context $\mathbb{K}$ *is defined as a triple* $(G, M, I)$*, where* $G$ *is called a set of* objects*,* $M$ *is called a set of* attributes*,* $I \subseteq G \times M$ *is a binary relation specifies which objects have which attributes.*

If $g \in G$, $m \in M$ and $gIm$, the object $g$ has the attribute $m$.

**Definition 2.** *The derivation operators* $(.)'$ *are defined for* $A \subseteq G$ *and* $B \subseteq M$ *as follows:*

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

**Definition 3.** Formal concept *of the context* $\mathbb{K} = (G, M, I)$ *is a pair* $(A, B)$*, where* $A \subseteq G$*,* $B \subseteq M$*,* $A' = B$ *and* $B' = A$*. The set* $A$ *is called the* extent*, and* $B$ *is called the* intent *of the concept* $(A, B)$*.*

**Definition 4.** *A concept* $(A, B)$ *is a subconcept of* $(C, D)$ *if* $A \subset C$ *(equivalently* $D \subset B$*). In this case* $(C, D)$ *is called a superconcept of* $(A, B)$*.*

The set of all concepts of $K$ ordered by subconcept-superconcept relation forms a lattice, which is called the *concept lattice* $\beta(\mathbb{K})$.

## 3 Problem statement

The problem solved in this paper is to search for duplicate objects in an ontology, i.e objects describing the same object in the real world. The original problem was posed by analysts of Avicomp company. Their main interest is to search for duplicate objects describing people and companies in an ontologies built by the automatic semantic processing flow of news texts. Currently, this problem in Avicomp is solved by methods based on the Hamming distance and a variety of additional heuristics.

The input to the algorithm takes an ontology constructed from text sources. An ontology contains objects of different classes. Objects can involve relationships, appropriate to their classes. The number of detected features and links between object can vary greatly. Some objects describe the same object in the real world.

At the output the algorithm should return lists of objects that have been detected as duplicates. The algorithm must have high precision because the returning of two different objects as duplicates is more critical error than not detecting some of the duplicates of the object.

## 4 A method of duplicates detection

The algorithm of duplicates detection consists of several stages:

1. Transformation of a source ontology to a multi-valued context.
2. Scaling multi-valued context.
3. Building the set of formal concepts of the context.
4. Building the set of potential duplicate objects.

### 4.1 Transformation of a source ontology to the multi-valued context

The source (instance of) ontology is transformed to the multi-valued context as follows:

1. A set of **context objects** is a set of objects $O$ of the ontology.
2. A set of **context attributes** is a set $M = L \cup C \cup R$, where:
   - $L$ is a set of all features defined by all ontology classes,
   - $C$ is a set of binary attributes, which characterize object classes,
   - $R$ is a set of binary attributes, which describe relations between ontology objects. Any relation $p(x, y)$ between ontology objects $x$ and $y$ generates two binary attributes in the context: $p(x, \_)$ and $p(\_, y)$. They correspond to the relations $p$ from $x$ and $p$ to $y$. An object $z$ has an attribute $p(\_, y)$ in context iff there exists a relation $p$ from $z$ to $y$ in the source ontology.
3. Each object is incident to the values of its source attributes. Also, each object gets a special value *null* for those attributes not incident to it or those whose incidence to the object is unknown. Also it gets binary attributes, corresponding to the object's class (and all of its superclasses) and relations between this object and other objects.

### 4.2 Scaling multi-valued context

The multi-valued context is converted to a binary context by means of *scaling* [1, 3]. Attribute sets $C$ and $R$ are binary. Attribute set $L$ is scaled depending on the type of attribute. As a rule, many attributes describe nonquantitative attributes of objects (e.g., a persons name, a company name, etc.). Moreover, many quantitative or numerical data are such that an approximated similarity by these attributes does not mean the similarity of objects. By way of example, if two company objects have attributes 2005 and 2006 as their year of establishment, the proximity (failure to match) of these attributes does not make us sure that the objects describe the same company; they more likely produce the opposite effect. For such attributes, only matching of their values makes sense and if the values are different then the distance between them does not matter. Such attributes are scaled by a *nominal scale*, i.e., its own binary attribute corresponds to each attribute value. In other attributes, some other scaling types are used such as:

- *Interval type*: the transformation of an attribute $A$ into a set of binary attributes of type $a \leq A < b$. In this case, the intervals $[a, b)$ can be both disjoint and overlapping.
- *Ordering*: an attribute $A$ is transformed into a set of binary attributes of $A > b$ type.
- Other scaling types which, in an experts opinion, can characterize the similarity of duplicate objects in the best way.

The experiments on the generated data and a real ontology use only nominal scaling; however, this does not restrain the generality of the proposed approach.

### 4.3 Building set of formal concepts of the context

There are several effective methods for building sets of formal concepts of the formal context. In this work the AddIntent [7] algorithm was used.

### 4.4 Formal concepts filtering

After building the set of formal concepts it is necessary to distinguish formal concepts having an extent which includes only duplicates of the particular object. Building special indices to filter concepts we have to consider all main features of these concepts. First, *index must increase if the number of attributes which are different for objects in a extent decreases all other things being equal.* To take into account this property we used the following index:

$$I_1(A, B) = \frac{|A||B|}{\sum_{g \in A} |\{g\}'|} \tag{1}$$

The second feature that an index must fulfill is — *it must increase when the number of common attributes is increasing, all other things being equal.* As a result we used an index that has got this feature:

$$I_2(A, B) = \sum_{m \in B} \frac{|A|}{|\{m\}'|} \tag{2}$$

The final index $DII$ is a combination of two indices described earlier. In our work we used the following combination variants:

1. Linear combination of indices:

$$DII_+ = k_1 I_1 + k_2 I_2 \tag{3}$$

2. Product of indices with power coefficients:

$$DII_* = I_1^{k_1} * I_2^{k_2} \tag{4}$$

Absolute values of the coefficients have an influence only to the value of a threshold and filtering quality depends on the coefficients correlation in the index formula. So we can specialize indices family without loss of the optimum and take 1 as a value for the one of the coefficients:

$$DII_+ = I_1 + kI_2, k > 0, \tag{5}$$

$$DII_* = I_1 * I_2^k, k > 0 \tag{6}$$

### 4.5 Forming the set of potential duplicate objects

The lists of objects that the algorithm returns as potential duplicates are obtained from those formal concept extents that have high value of index. The algorithm have two work modes: *automatic operation mode* and *semi-automatic operation mode with an expert assistance*.

In automatic mode the algorithm filters formal concepts by the threshold of the developed index. At this stage, various heuristics can be added that are hard to account for by means of the index. Then the algorithm prepares the lists of duplicate objects. We assume that the binary relation "be a duplicate" is transitive and it holds for objects in a selected formal concept. In this case to find lists of duplicates we should find connected components in the graph of objects with the "duplicate" relation.

In semi-automatic operation mode with an expert assistance the algorithm consequently offers expert estimate concepts in descending order of $DII$ values. The lists of duplicates are built as soon as an expert gives answers. Before asking an expert to estimate the concept the algorithm searches for all lists of duplicate objects with non-empty extent intersections with this object. If this extent is included in one of the lists then this concept is not offered to the expert. So the algorithm gets the list of objects corresponding to the current mark up made by the expert. Furthermore, the expert can stop the estimating process at each step and receive the formed lists of duplicate objects.

## 5 Experiments on artificially generated formal contexts

Basic experiments were conducted on artificially generated data with the duplicates known in advance in order to obtain statistical evaluation of the quality of the developed algorithm. Thus, this enables us to evaluate the quality of the method based on a large scope of input data and qualitatively compare it with the most widespread alternative approaches. Along with this, in the data generation, the features of ontology were taken into account to extrapolate the obtained results onto real data.

### 5.1 Input data generation

Various quality metrics on artificially generated contexts were used in order to evaluate the method quality. The generated formal contexts in this case have the properties of the contexts that were obtained from ontologies.

First, the generated contexts must contain a large number of objects and attributes. It is assumed that the objects will be measured in *tens of thousands*. The number of binary attributes is comparable to the number of objects, since many objects contain unique or rare attributes.

In this case, each object has a relatively small number of attributes. Their quantity does not exceed *several tens*. Therefore, the context is strongly scattered and despite its large dimension it has a relatively small number of formal concepts: from 5000 to 30000.

Second, the number of object attributes varies considerably and, as a rule, satisfies the Mandelbrot law, i.e., the number of attributes is in reverse proportion to the object range among the objects that are ordered by the number of their attributes.

The third property that is regarded in the context generation is an irregular distribution of attribute frequencies. Commonly, the attribute frequency is inversely proportional to its range in the sequence that is ordered by the frequency of the appearance of the attribute in the context objects. Upon generating unique objects, an input context was generated. An object in the context was generated for each object in the following way: each object attribute was added with a certain probability to the set of object attributes in the context. For some initial objects, several objects were thus generated. The obtained objects were taken as the duplicates of the same object.

### 5.2 Comparative analysis of the methods for detecting duplicates

As alternative methods we considered methods of pairwise comparison based on *Hamming distance* and *absolute similarity*. Also we considered the method which is similar to ours: the difference was in the use of *extensional stability* index [4, 8] instead of our index.

**The method based on the concept extensional stability** S. Kuznetsov was the first to introduce the formal concept stability in 1990 [4]. Later, in work [8], it was proposed to distinguish extensional and intentional stability. In our work, we deal with extensional stability since we assume that the objects that are considered as duplicates must be strongly related to a large number of attributes and have a small number of individual attributes. A formal concept they generate must be stable to the elimination of individual attributes.

The algorithm for searching for duplicates is similar to the basic method, viz. the most (extensionally) stable concepts are deleted from the set of formal concepts. Then it is assumed that the objects from the extent of the stable formal concept are the duplicates of the same object. The relationship R "to be duplicates" is built by the set of the chosen formal concepts. Then connectivity components for this relation are sought. The obtained components are given to the input as the lists of duplicate objects.

**The method based on absolute similarity** This method is based on the pairwise comparison of objects. Duplicate objects are assumed to have a large number of general attributes. Therefore, the number of general attributes serves the criterion of object similarity. The indicator that is based on this measure is the threshold of the quantity of general attributes.

The algorithm receives an incoming square similarity matrix $A : A[i][j] = k \Leftrightarrow$ the $i$th and $j$th objects have $k$ general binary attributes and the threshold $t(N)$.

The matrix $A$ and the threshold are used to build an adjacency matrix $B$ : $A[i][j] > t \Rightarrow B[i][j] = 1$.

The adjacency matrix (similarly to the ingress matrix) is symmetrical and describes the similarity relationship $R$. Proceeding from the fact that the relationship "to be a duplicate" is an equivalence relationship and possesses transitivity, its transitive closure $R^*$ is built using the obtained relationship $R$. The equivalence classes in $R^*$ correspond to the object groups that are the duplicates of the same object. The same result can be obtained by detecting all the connectivity components of the relationship $R$.

The asymptotic complexity of the algorithm by time is $O(n^2 * m)$, where n is the number of objects in the formal context and m is the number of attributes.

**The method based on Hamming distance** The algorithm for detecting duplicates is based on the pairwise comparison of objects. The Hamming distance serves as the metric of similarity. First, a square matrix of the distances between objects is built. Then, using the obtained matrix $A$ and a specified threshold $t(N)$ the matrix $B$ of the relationship $R$ "to be a duplicate" is built: $R : A[i][j] > t \Rightarrow B[i][j] = 1, (x_i, x_j) \in R$. The obtained relationship will be reflective and symmetrical. The connectivity components are sought by this relationship. The objects that enter the same connectivity component are considered as the duplicates of the same object.

The asymptotic complexity of the algorithm is similar to that of the algorithm based on absolute similarity, viz. $O(n^2 * m)$, where n is the number of objects in a formal context and m is the number of attributes.

### 5.3 The results

We used a few quality metrics for comparison of methods: recall, precision, average value of recall when precision is 100%, Mean Average Precision (MAP):

$$MAP(K) = \frac{\sum_{i=1}^{|K|} AveP(K_i)}{|K|} \tag{7}$$

$$AveP(k) = \frac{\sum_{c \in C_k} (P(c))}{|C_k|}, \tag{8}$$

where $K$ is set of contexts, $C_k$ is set of relevant formal concepts of the context $k$, $P(c)$ is number of the relevant concepts between all of the concepts having range (value of index) not lower than the concept $c$.

For the evaluation of the new method optimal coefficients for the index were primarily chosen. The coefficient was chosen using one of the generated contexts. The network on the positive real line was taken and the MAP index was maximized on it. Therefore, the coefficients for the used variants of the index $DII$ were obtained:

$$DII_+ = I_1 + 0.25I_2, \tag{9}$$

$$DII_* = I_1 * I_2^{0.18} \tag{10}$$

The algorithm with this index was compared with the alternative methods for searching for duplicates. In order to build the function of algorithm precision versus its recall, *several tens* of different thresholds were specified and then for each threshold, the recall and the precision were calculated.

The method based on extensional stability demonstrates good results at a high index threshold. At a threshold above 0.5 only formal concepts that have duplicates are chosen. At a threshold below 0.5, the algorithm precision drops on average to 10%, since a large number of formal concepts with stability 0.5 are one attribute concepts that do not characterize duplicate objects.

The algorithm for searching for duplicates using Hamming distance has shown relatively low results. The Hamming distance takes into consideration only distinctions in attributes rather than the quantity of general attributes

The algorithm based on absolute similarity proved to be the most efficient among the considered alternative algorithms. In most cases, a large number of common attributes in a pair of objects means that these objects are duplicates. The disadvantage of the index is that it disregards the distinctions between objects.

The algorithm based on the new index demonstrated better results than the alternatives considered. The main distinct feature of the new method is small decrease of precision (down to 90%) while recall increases up to 70%. The results for $DII_+$ and $DII_*$ are very similar. The difference is in that the behavior of $DII_*$ is less stable, viz., while sometimes making errors at a large threshold the algorithm did not make errors at a low threshold and detected 42% of the duplicates

**Table 1.** Max. recall with abs. precision

| Algorithm | Max. recall with 100% precision |
|---|---|
| Abs. similarity | 6.22% |
| Hamming distance | 0.56% |
| e-stability index | 22.44% |
| $DII_+$ index | 21.78% |
| $DII_*$ index | 9.49% |

## 6 Expirements on a real ontology

The ontology for tests was built by Avicomp. This ontology was built and extended automatically by semantic analysis of several political news sites. The OntosMiner [13] programming tool set was used. The ontology contains **12006**

**Table 2.** Mean Average Precision

| Algorithm | MAP |
|---|---|
| e-stability index | 0.4992 |
| $DII_+$index | 0.9352 |
| $DII_*$ index | 0.9382 |

**Table 3.** Optimal thresholds and search quality

| Algorithm | Threshold | Recall | Precision |
|---|---|---|---|
| Abs. similarity | 3.5 | 19.35% | 98.82% |
| Hamming distance | 0.5 | 34.37% | 86.32% |
| e-stability index | 0.5 | 22.44% | 100% |
| $DII_+$ index | 1.15 | 40.09% | 99.58% |
| $DII_*$ index | 0.9 | 31.8% | 99.55% |

objects of different classes. We used our algorithm for detecting duplicates with objects belonging to classes "Person" and "Company". The ontology contains **9821** such objects. Though we searched for duplicates only in two classes we used all classes and relations between objects and classes in ontology as attributes of objects in these classes.

A rather simple heuristical constraint was added in the algorithm based on the new index $DII$ (the $DII_+$ variant was used): we filtered out concepts which contained objects having different values of attribute Name or Last_name. The algorithm detected **905** group of objects. Group size ranged from 2 to 41 objects. The largest groups found by the algorithm described such people as Benjamin Netanyahu (41 objects), Julia Tymoshenko (35 objects), Vladimir Putin (34 objects), Dmitry Medvedev (33 objects), Steve Jobs (31 objects) etc. However, the main part of the detected groups contains 2 to 4 objects.

With experts assistance we estimated the precision of our algorithm. We could be sure that 98% of the detected groups consist of duplicates. Very often we can see groups, where attributes Name and Last_name are not common, but other attributes and relations let the algorithm place these objects in one group. For instance, the algorithm detected 7 objects, describing Ksenia Sobchak and having only 1 common ontology attribute but brought together because of same relations with other objects.

It is necessary to point out that attribute weights in index $I_2$ let algorithm detect large groups of objects describing Putin, Tymoshenko, Medvedev etc. The key feature of these objects is that all of them have a lot of attributes and relations that differ from other objects.

## 7 Conclusions

In this work a new algorithm for the detection of duplicate objects was introduced. The algorithm is based on methods of Formal Concepts Analysis. In particular a index for ranking formal concepts was proposed. The index allows one to select the set of concepts containing only duplicate objects with high accuracy. The proposed method was compared with other approaches to the solution of the problem of data duplication on randomly generated data and real ontology data. Experiments demonstrated the effectiveness of the new index. Further work will consist of estimating recall of the new method on a real ontology.

## Acknowledgments

The results of the project "Mathematical Models, Algorithms, and Software Tools for Intelligent Analysis of Structural and Textual Data", carried out within the framework of the Basic Research Program at the National Research University Higher School of Economics in 2012, are presented in this work.

## References

1. Ganter, B., Wille, R.: Formal Concept Analysis: Mathematical Foundations. In: Springer, 1999
2. Maedche, A., Zacharias, V.: Clustering Ontology-based Metadata in the Semantic Web. In: Proc. of 6th European Conference on Principles of Data Mining and Knowledge Discovery.—2002.—P. 348 - 360
3. Prediger S.: Logical scaling in formal concept analysis. In: ICCS, Lecture Notes in Computer Science.—1997.—Vol. 1257. Springer.—P. 332 - 341
4. Kuznetsov S.O.: Stability as an Estimate of the Degree of Substantiation of Hypotheses on the Basis of Operational Similarity. In: Nauchno-Tekhnicheskaya Informatsiya, Seriya 2, Vol. 24, No. 12, pp. 21-29, 1990
5. Kuznetsov, S.O.: On stability of a formal concept. In: Annals of Mathematics and Artificial Intelligence.—2007.—Vol. 49.—P. 101–115
6. Kuznetsov, S.O.: A Fast Algorithm for Computing All Intersections of Objects in a Finite Semi-Lattice. In: Automatic Documentation and Mathematical Linguistics 27(5), 11-21, 1993
7. Merwe, D., Obiedkov, S., Kourie, D.: AddIntent: a new incremental algorithm for constructing concept lattices. In: LNCS, Springer.—2004.—P. 205 - 206
8. Roth, C., Obiedkov, S., Kourie, D.: On Succinct Representation of Knowledge Community Taxonomies with Formal Concept Analysis. In: IJFCS (Intl Journal of Foundations of Computer Science).—2008.—P. 383 - 404
9. Rudolf Wille.: Restructuring lattice theory: an approach based on hierarchies of concepts. In: Ordered Sets: Dordrecht/Boston, Reidel. - 1982. P. 445-470
10. Kuznetsov, S.O.: Mathematical Aspects of Concept analysis. In: Journal of Mathematical Sciences, Vol. 80, Issue 2, P. 1654 - 1698, Springer.—1996
11. Kuznetsov, S., Obiedkov, S., Roth, C.: Reducing the representation complexity of lattice-based taxonomies. In: 15th Intl Conf on Conceptual Structures, ICCS 2007.—Sheffield, UK.—LNCS/LNAI. Vol. 4604. Springer.—2007

12. Klimushkin, M.A., Obiedkov, S.A., Roth, C.: Approaches to the selection of relevant concepts in the case of noisy data. In: 8th International Conference, ICFCA2010, Morocco.—Springer.—2010
13. http://www.ontos.com/?page_id=630

# A Database Browser based on Pattern Concepts

Jens Kötters and Heinz W. Schmidt†

†Computer Science & IT, RMIT University, Melbourne, Australia
(Heinz.Schmidt@rmit.edu.au)

**Abstract.** A Galois connection is stated between a knowledge base and queries over this knowledge. Queries are stated as conjunctions. Both the knowledge and queries are represented by certain graphs. This Galois connection gives rise to lattices of pattern concepts implicitly contained in the theory (all derivable facts) over the knowledge base.
The formal foundation for browsing such lattices and a realisation in terms of a prototype tool is outlined. Data types may be assigned to individual columns of tables in the database. A type assignment corresponds to an extension of the query language and incorporates additional knowledge into the process of concept creation. Type and derivation support in the tool may be provided by pluggable modules. In the examples in this paper, only the numeric type and concrete, stored relations are featured.

**Keywords:** Database Browsing, Pattern Concepts, Formal Concept Analysis, Knowledge Representation, Many-Sorted Logics

## 1 Introduction

The paper presents a prototype of a FCA browser for knowledge bases and its formal foundation. The browser allows interactive access to the content of a database. Via a command-line interface, concept lattices over relational data can be traversed. Each concept intent corresponds to a logical formula (or query) in one or more free variables, using relational expressions over function terms with variables where the functions range over primitive and user-defined data sorts.

Each extent is the corresponding table of results. There is one concept lattice for each set of free variables; the user can cross over into different lattices during navigation (thus changing variables in the result set). We will see that each concept lattice arises from a suitably defined pattern structure [5] (stretching the definition a bit), and pattern concepts have indeed been considered for the representation of logical formulas [5, p.129]. Further references and details of the approach can be found in [10], although many-sorted logic has not been considered there.

The formalisation of FCA navigation includes concrete (stored) and abstract (computed) relations derived by domain-specific conditional logical rules and/or relational algebra operations (SQL). Domain knowledge rules need to capture

– derived relations (e.g. computing relatives based on a network of parent-child relationships);

- domain-specific interpretations of object attributes, incl. common taxonomies and discretizations (e.g. age range of legal childhood, adulthood, retirement in social insurance databases etc.)
- representation invariants abstracting from syntactic and computational details of the representation incl. relational algebra, and independent of the specific database platform. While representations are typically realised imperatively, occassionally such invariants are required to manipulate and transform queries or tables for the purpose of navigation.

We call these rules *abstract*, in particular, because they are independent of specific sets of concrete relation tables, and hence remain invariant across different concrete databases for the given domain and also across updates of the same database. Although our browser prototype does not include an inference engine, such a module can be interfaced easily by storing the results of external reasoning steps as special tables accessible to the browser, on-the-fly. Here we focus on the connection with FCA lattices.

For the purpose of this paper, we interpret many-sorted logics in an algebraic-categorical framework – a view that has gained wide acceptance in the semantics of programming languages, abstract data types, knowledge representation and behaviour specification over several decades. It goes back to universal algebra [7] and work on formal specification and abstraction since the seventies (cf. e.g. [3, 4, 1]). In the interest of readability of the paper to a broad FCA audience, we limit ourselves to an overview and introduce notation only where necessary to be able to follow the core examples and algorithms presented in the paper. A complete formal exposition is beyond the scope of this paper and this conference.

The paper is organised as follows. In Section 2 we review relevant existing work on many-sorted structures and logics and summarise our notation; section 3 presents some technical advanced many-sorted structures that form the basis for our FCA-centric approach to patterns and queries; section 4 focuses on the browser, both in terms of the core algorithm and the user interface for navigation. Finally Section 5 provides some links to related work.

## 2 Many-Sorted Structures and Logics

In this section we briefly summarise basic notations and formalisation used in the rest of the paper. The algebraic-categorical view of *abstract data types* and data analysis has developed in line with model theory: syntax is captured in signatures limiting the construction of well-sorted terms and atomic formulae over algebras (data and functions) or structures (algebras plus relations) as models. Terms are sorted to represent data of primitive sorts abstractly, independent of a specific interpretation by a data domain. For example attributes, arithmetic or logical operators appearing in logical formulae or database queries may be sorted, as in the example below, where `Anne` is a constant of sort `person`, `age` is a numeric attribute of an `object` and `Parent` is a binary predicate on sort `person`. For flexible abstract many-sorted definitions we permit so-called order-

sorted models, i.e., where sorts are partially ordered. **bool** and **int** are assumed to be built in primitive sorts. **object** is a built-in maximal sort.

**sort int** < number, person < **object**
Anne,Bob,Chris,Dora,Emily: → person
_+_: number × number → number
_<_: number × number → **bool**
age: object → **int**
Parent: person × person
female,male: person

As we will see later, the sort order abstracts from a corresponding subset relationship between corresponding data domains. We also allow so-called 'mixfix' notation for function and predicate symbols as known from platforms realising algebraic-categorical forms of many-sorted type or logical specifications, such as OBJ3 [6], CASL [1], and ELAN [2, 9]. For instance, _+_ indicates the two argument positions for this binary infix operator '+'. **Signatures**. Formally,

**Parent**

| c0 | c1 |
|---|---|
| Anne | Bob |
| Anne | Chris |
| Bob | Dora |
| Bob | Emily |

**age**

| c0 | c1 |
|---|---|
| Anne | 59 |
| Bob | 31 |
| Chris | 27 |
| Dora | 7 |
| Emily | 3 |

**META**

| table | column | type |
|---|---|---|
| Parent | c0 | person |
| Parent | c1 | person |
| male | c0 | person |
| female | c0 | person |
| age | c0 | person |
| age | c1 | number |

**female**

| c0 |
|---|
| Anne |
| Dora |
| Emily |

**male**

| c0 |
|---|
| Bob |
| Chris |

**Fig. 1.** Database

a many-sorted signature is a triple $\Sigma = (S, F, P)$ where $S$ is a finite partial order (of elements called *sort symbols* or sorts for short), $F = (F_{u,s})_{u \in S^*, s \in S}$ is a pairwise disjoint family of sets of symbols (called *function symbols*) and $P = (P_u)_{u \in S^*}$ is a family of pairwise disjoint sets of symbol (called *predicate symbols*). For $f \in F_{u,s}$ (or $p \in P_u$) we set $dom(f) = u$ (or $dom(p) = u$, respectively) and $cod(f) = s$ (read *'domain'* and *'codomain'* respectively). As usual for abstract types and many-sorted logics, signature morphism remap sorts, function and predicate symbols preserving domains, codomains and sort order. We use $T_{\Sigma,s}$ to denote the set of well-formed terms of sort $s$ and $A_\Sigma$ the set of well-formed atoms $p(t_1, \ldots, t_n)$ for $p \in P_u, u = s_1 \cdots s_n, t_i \in T_{\Sigma,s_i} (1 \leq i \leq n)$.

**Elimination of junk**. Many-sorted approaches are interesting to us as they reduce the search space for inferencing and navigation: ill-sorted terms and formulae can be recognised efficiently, and in fact eliminated syntactically. This

reduces the search space significantly and eliminates massive amounts of so-called 'junk data' in terms of ill-sorted elements, in particular in queries and auxiliary formulae occuring in searches.

**Example database**. Before we formalise concrete models, let us look at a example database as a concrete model in terms of sets and tables.

A database table corresponds to (a) a relation interpreting a corresponding predicate symbol, or (b) a function from a set of columns (arguments) to a column (result), or (c) a set of attributes mapping the rows (object keys) to attributes, thus encoding functions similar not unlike (b). For example in Fig. 1, the table named `Parent` contains all pairs $(p, p')$ such that $Parent(p, p')$ is valid, while the table `age` maps persons to their ages. Column are representing attributes including selectors of components in tuples. For example $c0$ in the `age` table selects the `person` component of the table rows etc. Queries supported by the database are constrained by the signature and the logical connectives permitted in the structure of formulae outside the algebraic structure captured by the signatures. For concrete databases and in our prototype implementation, we assume the existence of a META table (see Fig. 1) which represents the signature information relevant for the tables in the database. The remaining signature (outside the data base) represents operators on data types in the tables or relations that can be computed from the data base using queries.

**Many-sorted structures**. Given a many-sorted signature $\Sigma$, a $\Sigma$-structure $\mathbf{D} = \langle (D_s)_{s\in S}; \mathcal{F}, \mathcal{R}\rangle$ has a family of carrier sets $D_s$ (aka domains) sorted and ordered by $S$ and families of sets of functions and relations compatible with the prescribed domains and codomains of functions and predicate symbols in $\Sigma$. The partial order of sorts is interpreted as subsorting: $D_s \subseteq D_t$ if $s \leq t$. Functions are total on their domains.[1] For readability in concrete examples we also denote $D_s$ by $s_D$ (the interpretation of sort $s$ in $\mathbf{D}$). Likewise, for $f \in F_{u,s}$ ($p \in P_u$) we denote the corresponding function in $\mathbf{D}$ by $f_D$ (or $p_D$ respectively). It is well-known that the term structure $\mathbf{T}_\Sigma := \langle (T_s)_{s\in S}; F, P\rangle$ with empty relations forms the free $\Sigma$-structure. Homomorphisms between $\Sigma$-structures are *weak*, i.e. preserve definedness but not necessarily undefinedness of relations. They are called *strong* if they also preserve undefinedness.

The formal structure underlying the database in Fig. 1 has for example:

$$\begin{aligned}
\texttt{person}_D &= \{\text{Anne}, \text{Bob}, \text{Chris}, \text{Dora}, \text{Emily}\},\\
\texttt{Anne}_D &= \text{Anne}, \ldots, \texttt{Emily}_D = \text{Emily},\\
\texttt{number}_D &= \{3, 7, 27, 31, 59, ...\},\\
\texttt{Parent}_D &= \{(\text{Anne}, \text{Bob}), (\text{Anne}, \text{Chris}), (\text{Bob}, \text{Dora}), (\text{Bob}, \text{Emily})\},\\
\texttt{age}_D &= \{\text{Anne} \mapsto 59, \text{Bob} \mapsto 31, \text{Chris} \mapsto 27, \text{Dora} \mapsto 7, \text{Emily} \mapsto 3\}
\end{aligned}$$

**Many-sorted logics**. For the rest of the paper, let $\Sigma$ be a fixed signature and $\mathbf{D}$ a $\Sigma$-structure. We assume each sort in $\Sigma$ includes a distinguished equality predicate $=_s$ with the obvious interpretation in $\mathbf{D}$. For sorted terms and formulae

[1] Note that the underlying algebra $\mathbf{D} = \langle D; \mathcal{F}\rangle$, with the unsorted carrier $D$ the union of the $D_s$, is partial, as is the underlying unsorted structure.

with variables we use a many-sorted family $(V_s)_{s\in S}$ of at most countably infinite and pairwise disjoint sets of variable symbols that are disjoint from function symbols in $\mathcal{F}$. We denote by $\Sigma(V)$ the extended signature that adds variables as constant function symbols to $\Sigma$. The $\Sigma(V)-term$ structure now contains all well-sorted terms with variables. $\Sigma$-formulae are built using well-sorted atoms $p(t,...)$ for predicate symbols $p$ in $\Sigma$, conjunction, implication and existential quantifiers over sorted variables constrained to prenex normal form (i.e. not occurring under conjunction or implication but ranging over the entire formula at hand). We denote by $Free(\phi)$ the free variables of a formula $\phi$, call $\phi$ *closed* iff $Free(\phi)=\emptyset$, denote by $At_\Sigma$ the set of all $\Sigma$-atoms, and by $Cl_\Sigma$ the set of $\Sigma$-formulae. Formulae are evaluated over a $\Sigma$-structure as usual, by recursively 'translating' function symbols into function application, predicate symbols into relations in $\mathbf{D}$ and interpreting conjunction, implication and existential quantification logically. We use $I_D$ to denote the corresponding interpretation of closed terms and fomulae and write $\mathbf{D}\models\phi$ to denote that $\phi$ is valid in the model $\mathbf{D}$. For terms or formulae with variables ($Free(\phi)=\{x_1,\ldots,x_n\}$) we write $I_D(\phi[a_1/x_1,\ldots,a_n/x_n])$ to denote the corresponding evaluation under the assignment of the $a_i$ to $x_i$.

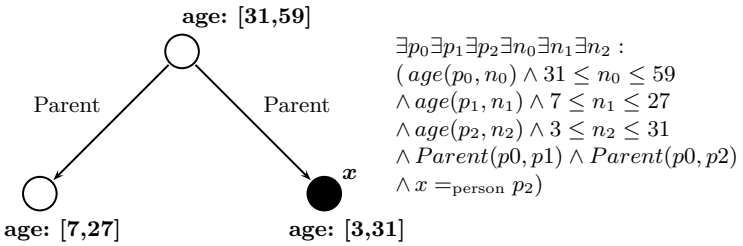



**Fig. 2.** Query graph for pattern formula

## 3 Patterns

For queries in particular, we are interested in formulae $\psi$ in the following prenex normal form,

$$\exists x_1 \ldots \exists x_m : \phi \tag{1}$$

where $\phi$ is a conjunction, $\phi \equiv (\phi_1 \wedge \cdots \wedge \phi_l)$. We interpret $\psi$ as the request to compute all possible consistent assignments to $Free(\psi)$ such that $\mathbf{D}\models\psi[a_1/x_1,\ldots,a_n/x_n]$. Without loss of generality, we assume that each free variable $y\in Free(\psi)$ has a single occurrence on the left-hand side of an equation $y=_s \ldots$. In the above sense, $\Sigma$-formulae of the form $\psi$ can be regarded as *patterns* that select matches in any $\Sigma$-structure. More precisely, the set of pattern matches $P_\phi := \{(a_1,\ldots,a_n)\in D_{s_1}\times\cdots\times D_{s_n} \mid \mathbf{D}\models\psi[a_i/x_1,\ldots,a_n/x_n]\}$ is well-defined. The patterns for a given set of free variables form a lattice by implication (set inclusion of their matches):

$$\psi\leq\psi' :\Leftrightarrow P_\psi\subseteq P_{\psi'} \tag{2}$$

where $\psi$ and $\psi'$ are two $\Sigma$-formulae of the form (1) above, s.t. $Free(\psi) = Free(\psi')$. We denote this lattice by $L_{\Sigma,u}$ (or $L_u$ for short when $\Sigma$ is fixed), where $x_i \in V_{s_i} (1 \leq i \leq n)$ and $u = s_1 \cdots s_n$. (Because formulae are equivalent under renaming of free variables, the sorts of variables only matter.) The top element $\top_{L_u}$ corresponds to the tautology $\exists y_1 \ldots \exists y_n : x_1 =_{s_1} y_1 \wedge \cdots \wedge x_n =_{s_n} y_n$. In particular single-sort patterns with $|Free(\psi)| = 1$ define subdomains of some $D_s$ (with $\{x\} = Free(\psi)$). Elements of $L_s$ represent *logical subdomains* of the given sort, i.e., subdomains expressible in logical formulae over $\Sigma$.

For the *structural* representation of (1) we use a tuple $(X, \nu, (\mathbf{G}, \kappa))$ determined as follows: For each sort $s \in S$, the free variables of sort $s$ occuring in (1) are collected in the set $X_s$, and $X := (X_s)_{s \in S}$. Correspondingly, the bound variables of sort $s$ are collected in the domain $G_s$ of the many-sorted structure $\mathbf{G}$. For each relation symbol $p \in P_u$, we have $\mathbf{G} \models p(y_1, \ldots, y_k)$ iff $p(y_1, \ldots, y_k)$ is an atom in $\phi$. For each $s \in S$, $\kappa_s$ is a mapping on $D_s$. For each $x \in D_s$, $\kappa_s(x)$ is a formula equivalent to the conjunction of all domain-specific conditions $\phi_i$ on $x$. In particular, $\kappa_s(x) := \top$ if no conditions on $x$ occur in (1). Finally, $\nu$ is a family of mappings $\nu_s : X_s \to D_s$, where $\nu_s(x)$ is the unique $v$ such that $x =_s v$ occurs in $\phi$. We call such a tuple a *windowed structure*, and the pair $(\mathbf{G}, \kappa)$ an *augmented structure*. For technical reasons, we allow arbitrary sets for the domains of $\mathbf{G}$. Figure 2 shows a formula and next to it the associated windowed structure, which may be drawn as a graph.

Entailment is formalized by homomorphisms. Their definition reflects the nestedness of structures. A homomorphism $f : (\mathbf{G_1}, \kappa_1) \to (\mathbf{G_2}, \kappa_2)$ of augmented structures is a homomorphism $f : \mathbf{G_1} \to \mathbf{G_2}$ of many-sorted structures such that $(\kappa_{\mathbf{D}})_s(f(v)) \Rightarrow \kappa_s(v)$ for all $s \in S$ and $v \in G_s$. A homomorphism $f : (X_1, \nu_1, \mathcal{G}_1) \to (X_2, \nu_2, \mathcal{G}_2)$ exists in the case $X_1 \subseteq X_2$ and is then a homomorphism $f : \mathcal{G}_1 \to \mathcal{G}_2$ which preserves free variables, that is $f \circ \nu_1 = \nu_2 \circ \iota_{X_2}^{X_1}$, where $\iota_{X_2}^{X_1}$ is the subset embedding.

Within the scope of this paper, we represent a knowledge base by an augmented structure $\Delta := (\mathbf{D}, \kappa_\Delta)$, where $\mathbf{D}$ is a $\Sigma$-structure representing a database and and $(\kappa_\Delta)_s(g)$ is the most specific equivalence class of formulae in $L_s$ characterising the object $g$. The solution set of a conjunctive query over $\Delta$, represented by a windowed structure $(X, \nu, \mathcal{G})$, is $\mathrm{Hom}(\mathcal{G}, \Delta) \circ \nu$. The solution set can be regarded a subset of $\mathrm{Hom}(X, \Delta)$, if we regard $X$ as a trivial augmented structure (the details are omitted). More generally, we define a *table* to be a pair $(X, \Lambda)$, where $X$ is a many-sorted family of variables and $\Lambda \subseteq \mathrm{Hom}(X, \Delta)$. By $\mathrm{Tab}(\Delta)$ we denote the set of all tables over $\Delta$. The order on $\mathrm{Tab}(\Delta)$ in which the infimum is the join is given by $(X_1, \Lambda_1) \leq (X_2, \Lambda_2) :\Longleftrightarrow X_2 \subseteq X_1 \wedge \Lambda_1 \circ \iota_{X_1}^{X_2} \subseteq \Lambda_2$.

Given many-sorted, augmented or windowed structures $S_1$ and $S_2$, we say that $S_1$ *generalizes* $S_2$ and denote this by $S_1 \lesssim S_2$, if a homomorphism $f : S_1 \to S_2$ exists. Generalization is a preorder, and we call $S_1$ and $S_2$ *hom-equivalent*, if $S_1 \lesssim S_2$ and $S_2 \lesssim S_1$. It is not difficult to see that the product $\prod_{i \in I} \mathbf{G_i}$ of a family $(\mathbf{G_i})_{i \in I}$ of many-sorted structures is an infimum in the generalization preorder(recall however that an infimum in a preorder is not unique). Infima of

augmented or windowed structures are realized by products:

$$\prod_{i\in I}(\mathbf{G_i}, \kappa_i) := (\prod_{i\in I}\mathbf{G_i}, \kappa, \quad \text{where } \kappa_s(v) := \bigwedge_{i\in I}(\kappa_i)_s(v) \quad ,$$

$$\prod_{i\in I}(X_i, \nu_i, \mathcal{G}_i) := (\bigcap_{i\in I}X_i, \langle\nu_i\rangle, \prod_{i\in I}\mathcal{G}_i, \quad \text{where } \langle\nu_i\rangle(x) := (\nu_i(x))_{i\in I} \quad .$$

**Galois Connection** The following operations define a Galois connection between $(\mathcal{W}, \lesssim)$ and $(\mathrm{Tab}(\Delta), \leq)$:

$$(X, \nu, \mathcal{G})' := (X, \mathrm{Hom}(\mathcal{G}, \Delta) \circ \nu), \tag{3}$$

$$(X, \Lambda)' := (X, \langle(\lambda)_{\lambda\in\Lambda}\rangle, \Delta^{\Lambda}) = \prod_{\lambda\in\Lambda}(X, \lambda, \Delta). \tag{4}$$

*Proof.* We only show operations are order-reversing, extensivity is easier to see. If $(X_1, \nu_1, \mathcal{G}_1) \lesssim (X_2, \nu_2, \mathcal{G}_2)$, then there is by definition $\varphi \in \mathrm{Hom}(\mathcal{G}_1, \mathcal{G}_2)$ with $\nu_2 \circ \iota_{X_2}^{X_1} = \varphi \circ \nu_1$. Thus $\mathrm{Hom}(\mathcal{G}_2, \Delta) \circ \nu_2 \circ \iota_{X_2}^{X_1} = \mathrm{Hom}(\mathcal{G}_2, \Delta) \circ \varphi \circ \nu_1 \subseteq \mathrm{Hom}(\mathcal{G}_1, \Delta)$. So $(\cdot)'$ in (3) is order-reversing.

Let $(X_1, \Lambda_1) \leq (X_2, \Lambda_2)$. Then for all $\lambda \in \Lambda_1$ we have $\lambda \circ \iota_{X_1}^{X_2} \in \Lambda_2$, and thus $\prod_{\lambda\in\Lambda_2}(X, \lambda, \Delta) \lesssim (X_2, \lambda \circ \iota_{X_1}^{X_2}, \Delta) \lesssim (X, \lambda, \Delta)$. So $\prod_{\lambda\in\Lambda_2}(X, \lambda, \Delta) \lesssim \prod_{\lambda\in\Lambda_1}(X, \lambda, \Delta)$, and $(\cdot)'$ in (4) is order-reversing. □

The Galois connection gives rise to the complete lattice $\mathfrak{L}_\Delta$, or to $\mathfrak{L}_\Delta[X]$ if restricted to queries $\phi$ with $Free(\phi) = \bigcup_{s\in S} X_s$:

$$\mathfrak{L}_\Delta := \{(T, W) \mid T \in \mathrm{Tab}(\Delta), W \in \mathcal{W}, T' = W, W' = T\} \tag{5}$$

$$\mathfrak{L}_\Delta[X] := \{(T, W) \in \mathfrak{L}_\Delta \mid \exists\Lambda : T = (X, \Lambda)\} \tag{6}$$

We hold that only formulas represented by connected patterns (as in Fig. 2) qualify as concept descriptions, and thus only components of powers of $\Delta$ qualify as concept intents (cf. (4)). The implemented algorithm is still immature and will therefore only briefly be considered in the next section.

| #(concepts) | DB relations | +constants | +numeric comparison |
|---|---|---|---|
| $x$:person | 9 | 12 | 18 |
| $x$,$y$:person | 26 | 59 | 85 |

**Table 1.** Number of concepts, depending on free variables and signature

## 4 Pattern Browser

### 4.1 Algorithm

In the order $\Delta^0, \Delta^1, \Delta^2, \ldots$, powers are computed and decomposed into their components, which are paired up with morphisms designating the subjects of the query (cf. $\langle(\lambda)_{\lambda\in\Lambda}\rangle$ in (4)), translated into SQL, paired up with result tables

returned by a MySQL server, and then compared (using the order on Tab($\Delta$)) to eliminate equivalent patterns and build the concept lattice(s). The algorithm terminates when a power $\Delta^k$ does not produce new patterns. Table 1 shows the number of generated concepts for queries in one and two free variables of type person, for different settings of query expressiveness. The "naive" algorithm did not terminate in reasonable time even for some of the small examples. This is due to combinatorial explosion of patterns in higher powers of $\Delta$ and hardness of query optimization. More efficient algorithms are expected to make use of the fact that graph nodes are tuples over $\Delta$.

```
00| DATABASE BROWSER (press 'h' for help)
01| Concept#0>top
02| Concept#0>intent
03| x : age(x,n0) AND 3<=n0<=59
04| Concept#0>specialize
05| Concept#1>intent
06| x : age(x,n0) AND female(x) AND 3<=n0<=59
07| Concept#1>extent
08|
09|   x
10| -----
11| Anne
12| Dora
13| Emily
14|
15| Concept#1>generalize
16| Concept#0>specialize
17| Concept#3>intent
18| x : age(p0,n0) AND age(p4,n4) AND age(x,n5) AND parent(p0,p4) AND parent(p0,x)
19| AND 31<=n0<=59 AND 7<=n4<=27 AND 3<=n5<=31
20|
21| Concept#3>specialize
22| Concept#14>intent
23| y,x : age(y,n0) AND age(p3,n3) AND age(x,n2) AND parent(y,p3) AND parent(y,x)
24| AND 31<=n0<=59 AND 7<=n3<=27 AND 3<=n2<=31
25|
26| Concept#14>extent
27|
28|  y  |   x
29| ---------
30| Anne| Bob
31| Anne| Chris
32| Bob | Dora
33| Bob | Emily
34|
35| Concept#14>specialize
36| ->Concept#46***
37| ->Concept#18
38|
39| ------------------------------------------------------------------------
40| y,x : age(y,n0) AND age(p3,n3) AND age(x,n4) AND parent(y,p3) AND
41| parent(y,x) AND 31<=n0<=59 AND 7<=n3<=27 AND 3<=n4<=27
```

**Fig. 3.** Browsing session

### 4.2 Interface

Figure 3 shows a browsing session, which starts in the top concept. Extent, intent, lower neighbors and upper neighbors of the current concept can each be shown by pressing a key. If a list of neighbors is shown, each concept in the list

can be highlighted and examined in the subwindow below the dashed line before it is selected (see Fig.3). The intent is shown as a formula; free variables are listed before the colon, all other variables are existence quantified. The concepts computed are the 103 concepts listed in the right column of Table 1.

## 5 Related Work

The Galois connection between $\Sigma$-theories (sets of $\Sigma$-formulae closed under derivation) and categories of models (here $\Sigma$-structures) is folklore in model theory and celebrated in textbooks on algebraic and logical specification for data and behaviour. In this paper we use a more restricted connection for formal concept navigation on a knowledge base, focusing on queries (formulae) and their result sets (structures). The abstraction of the representation of data and knowledge bases in such theories renders access to a powerful mathematical tools. Details of a mapping from such data and knowledge bases to theories and structures can be found elsewhere.

The second author used a many-sorted theory construction [11] for expressively modeling typed formal concepts with a rich set of sorts and user-defined data types, including subsorts. However typed conceptual scaling was used for the relevant subsorts to associate a Galois connection with the resulting structures for each sort and to work directly on typed context tables. Patterns were not supported in that work.

If we extend the definition of pattern structure in [5] to categories of patterns (preordered by morphisms), then $(\bigcup_X \mathrm{Hom}(X,\Delta), (\mathcal{W}, \bigsqcap), \delta)$ is a pattern structure. A morphism $\lambda \in \mathrm{Hom}(X,\Delta)$, $X \subseteq V$, is essentially a partially defined variable assignment. Each such assignment is naturally identified with a windowed structure $\delta(\lambda) := (\mathrm{dom}(\lambda), \lambda, \mathrm{cod}(\lambda))$, where always $\mathrm{cod}(\lambda) = \Delta$. The Galois connection stated in [5] becomes

$$A^{\square} := \bigsqcap_{\lambda \in \Lambda} \delta(\lambda) \tag{7}$$

$$(X, \nu, \mathcal{G})^{\square} := \{\lambda \in \bigcup_X \mathrm{Hom}(X,\Delta) \mid \exists f : (X,\nu,\mathcal{G}) \to \delta(\lambda)\}. \tag{8}$$

The set $A$ of partial assignments corresponds to the table $(X, \Lambda)$ with $X := \bigcap_{\lambda \in A} \mathrm{dom}(\lambda)$ and $\Lambda := \{\lambda|_X \mid \lambda \in A\}$; the windowed structures $\Lambda'$ and $A^{\square}$ are hom-equivalent. However, the empty tables $(X, \emptyset)$,$X \neq V$, have no representation in this approach; in this, the produced lattice may differ from $\mathfrak{L}_\Delta$.

A relational context family can be defined as a pair $((\mathbb{K}_i)_{i\in I}, (R_j)_{j\in J})$, where each $\mathbb{K}_i =: (G_i, M_i, I_i)$ is a formal context and each $R_j$ is a binary relation on $G_{i_1} \times G_{i_2}$ for some $i_1, i_2 \in I$. The concept lattice for $\mathbb{K}_i$ is denoted by $\mathfrak{B}(\mathbb{K}_i)$. The relational context family corresponds to a many-sorted relational structure with sort set $I$ and family of relations $(R_j)_{j\in J}$. The concept lattices $\mathfrak{B}(\mathbb{K}_i)$, $i \in I$, correspond to the lattices of domain logical formulas. Relational Concept Analysis, as described in [8] and with existential scaling, produces for each sort $i$ a $\bigvee$-sublattice of $\mathfrak{C}_\Delta[X]$, where X contains one variable of sort $i$, which contains all concepts generated by finite, connected, acylic windowed graphs. This can be shown by induction over the steps of the algorithm given in [8].

## 6 Conclusion

This paper introduced a novel approach to model queries over relational data – both stored and computed – in terms of FCA lattices. In logical terms, abstract patterns are represented by certain many-sorted formulae with variables. The underlying implication lattices of the formulae and certain structures computed over the database form a Galois connection, as we showed, suitable for navigation of a solution space to the query. Changes in the query are translated into changes to the underlying lattice. Navigation thus includes intra-lattice and inter-lattice moves available to the user exploring domain knowledge over a database in terms of its concrete relation tables and tacit knowledge about these tables. A prototype browser was implemented to evaluate these concepts and was described in the paper.

## References

1. Baumeister, H., Bert, D.: Algebraic specification in casl. In: Frappier, M., Habrias, H. (eds.) Software Specification Methods, pp. 209–224. Formal Approaches to Computing and Information Technology FACIT, Springer London (2001), http://dx.doi.org/10.1007/978-1-4471-0701-9_12
2. Borovansky, P., Castro, C.: Cooperation of constraint solvers: Using the new process control facilities of elan. In: Proceedings of The Second International Workshop on Rewriting Logic and its Applications, RWLW'98. pp. 379–398 (1998)
3. Broy, M., Wirsing, M.: Partial abstract types. Acta Informatica 18(1), 47–64 (1982)
4. Cohn, A.G.: A more expressive formulation of many sorted logic. Journal of automated reasoning 3(2), 113–200 (1987)
5. Ganter, B., Kuznetsov, S.O.: Pattern structures and their projections. In: Delugach, H.S., Stumme, G. (eds.) Proceedings of ICCS 2001. LNCS, vol. 2120, pp. 129–142. Springer (2001)
6. Goguen, J., Kirchner, C., Kirchner, H., Mgrelis, A., Meseguer, J., Winkler, T.: An introduction to obj 3. In: Kaplan, S., Jouannaud, J.P. (eds.) Conditional Term Rewriting Systems, Lecture Notes in Computer Science, vol. 308, pp. 258–263. Springer Berlin Heidelberg (1988), http://dx.doi.org/10.1007/3-540-19242-5_22
7. Grätzer, G.: Universal algebra. Springer (2008)
8. Huchard, M., Hacene, M.R., Roume, C., Valtchev, P.: Relational concept discovery in structured datasets. Annals of Mathematics and Artificial Intelligence 49(1-4), 39–76 (2007)
9. Kirchner, H., Moreau, P.E.: Non-deterministic computations in elan. In: Fiadeiro, J. (ed.) Recent Trends in Algebraic Development Techniques, Lecture Notes in Computer Science, vol. 1589, pp. 168–183. Springer Berlin Heidelberg (1999), http://dx.doi.org/10.1007/3-540-48483-3_12
10. Kötters, J.: Concept lattices of a relational structure. In: Pfeiffer, H.D., Ignatov, D.I., Poelmans, J., Gadiraju, N. (eds.) Proceedings of ICCS 2013. LNCS, vol. 7735, pp. 301–310. Springer (2013)
11. Peake, I.D., Thomas, I., Schmidt, H.: Typed formal concept analysis. In: 7th International Conference on Formal Concept Analysis (ICFCA09). pp. 35–51. Springer (2009)

# An FCA-based Boolean Matrix Factorisation for Collaborative Filtering

Elena Nenova[2,1], Dmitry I. Ignatov[1], and Andrey V. Konstantinov[1]

[1] National Research University Higher School of Economics, Moscow
dignatov@hse.ru
http://www.hse.ru
[2] Imhonet, Moscow
http://imhonet.ru

**Abstract.** We propose a new approach for Collaborative filtering which is based on Boolean Matrix Factorisation (BMF) and Formal Concept Analysis. In a series of experiments on real data (Movielens dataset) we compare the approach with the SVD- and NMF-based algorithms in terms of Mean Average Error (MAE). One of the experimental consequences is that it is enough to have a binary-scaled rating data to obtain almost the same quality in terms of MAE by BMF than for the SVD-based algorithm in case of non-scaled data.

**Keywords:** Boolean Matrix Factorisation, Formal Concept Analysis, Singular Value Decomposition, Recommender Algorithms

## 1 Introduction

Recently Recommender Systems is one of the most popular subareas of Machine Learning. In fact, the recommender algorithms based on matrix factorisation techniques (MF) has become industry standard.

Among the most frequently used types of Matrix Factorisation we definitely should mention Singular Value Decomposition (SVD) [7] and its various modifications like Probabilistic Latent Semantic Analysis (PLSA) [14]. However, the existing similar techniques, for example, non-negative matrix factorisation (NMF) [16,13,9] and Boolean matrix factorisation (BMF) [2], seem to be less studied. An approach similar to matrix factorization is biclustering which was also successfully applied in recommender system domain [18,11]. For example, Formal Concept Analysis [8] can also be used as a biclustering technique and there are some of its applications in recommenders' algorithms [6,10].

The aim of this paper is to compare recommendation quality of some of the aforementioned techniques on real datasets and try to investigate the methods' interrelationship. It is especially interesting to conduct experiments on comparison of recommendations quality in case of an input matrix with numeric values and in case of a Boolean matrix in terms of Precision and Recall as well as MAE. Moreover, one of the useful properties of matrix factorisation is its ability to keep reliable recommendation quality even in case of dropping some

insufficient factors. For BMF this issue is experimentally investigated in section 4.

The novelty of the paper is defined by the fact that it is a first time when BMF based on Formal Concept Analysis [8] is investigated in the context of Recommender Systems.

The practical significance of the paper is determined by demands of the recommender systems' industry, that is to gain reliable quality in terms of Mean Average Error (MAE), Precision and Recall as well as time performance of the investigated method.

The rest of the paper consists of five sections. The second section is an introductory review of the existing MF-based recommender approaches. In the third section we describe our recommender algorithm which is based on Boolean matrix factorisation using closed sets of users and items (that is FCA). Section 4 contains methodology of our experiments and results of experimental comparison of different MF-based recommender algorithms by means of cross-validation in terms of MAE, Precision and Recall. The last section concludes the paper.

## 2 Introductory review of some matrix factorisation approaches

In this section we briefly describe different approaches to the decomposition of both real-valued and Boolean matrices. Among the methods of the SVD group we describe only SVD. We also discuss nonnegative matrix factorization (NMF) and Boolean matrix factorization (BMF).

### 2.1 Singular Value Decomposition (SVD)

Singular Value Decomposition (SVD) is a decomposition of a rectangular matrix $A \in \mathbb{R}^{m \times n} (m > n)$ into the product of three matrices

$$A = U \begin{pmatrix} \Sigma \\ 0 \end{pmatrix} V^T, \tag{1}$$

where $U \in \mathbb{R}^{m \times m}$ and $V \in \mathbb{R}^{n \times n}$ are orthogonal matrices, and $\Sigma \in \mathbb{R}^{n \times n}$ is a diagonal matrix such that $\Sigma = diag(\sigma_1, \ldots, \sigma_n)$ and $\sigma_1 \geq \sigma_2 \geq \ldots \geq \sigma_n \geq 0$. The columns of the matrix $U$ and $V$ are called singular vectors, and the numbers $\sigma_i$ are singular values [7].

In the context of recommendation systems rows of $U$ and $V$ can be interpreted as vectors of the user's and items's loyalty (attitude) to a certain topic (factor), and the corresponding singular values as the importance of the topic among the others. The main disadvantage is in the fact that the matrix may contain both positive and negative numbers; the last ones are difficult to interpret.

The advantage of SVD for recommendation systems is that this method allows to obtain the vector of its loyalty to certain topics for a new user without SVD decomposition of the whole matrix.

The evaluation of computational complexity of SVD according to [15] is $O(mn^2)$ floating-point operations if $m \geq n$ or more precisely $2mn^2 + 2n^3$.

Consider as an example the following table of movie ratings:

**Table 1.** Movie rates

| | The Artist | Ghost | Casablanca | Mamma Mia! | Dogma | Die Hard | Leon |
|---|---|---|---|---|---|---|---|
| User1 | 4 | 4 | 5 | 0 | 0 | 0 | 0 |
| User2 | 5 | 5 | 3 | 4 | 3 | 0 | 0 |
| User3 | 0 | 0 | 0 | 4 | 4 | 0 | 0 |
| User4 | 0 | 0 | 0 | 5 | 4 | 5 | 3 |
| User5 | 0 | 0 | 0 | 0 | 0 | 5 | 5 |
| User6 | 0 | 0 | 0 | 0 | 0 | 4 | 4 |

This table corresponds to the following matrix of ratings:

$$A = \begin{pmatrix} 4\,4\,5\,0\,0\,0\,0 \\ 5\,5\,3\,4\,3\,0\,0 \\ 0\,0\,0\,4\,4\,0\,0 \\ 0\,0\,0\,5\,4\,5\,3 \\ 0\,0\,0\,0\,0\,5\,5 \\ 0\,0\,0\,0\,0\,4\,4 \end{pmatrix}.$$

From the SVD matrix decomposition we get:

$$U = \begin{pmatrix} 0.31 & 0.48 & -0.49 & -0.64 & -0.06 & 0 \\ 0.58 & 0.50 & 0.03 & 0.63 & 0.06 & 0 \\ 0.29 & 0 & 0.57 & -0.23 & -0.72 & 0 \\ 0.57 & -0.37 & 0.31 & -0.30 & 0.57 & 0 \\ 0.29 & -0.47 & -0.43 & 0.15 & -0.28 & -0.62 \\ 0.23 & -0.37 & -0.35 & 0.12 & -0.22 & 0.78 \end{pmatrix},$$

$$\begin{pmatrix} \Sigma \\ 0 \end{pmatrix} = \begin{pmatrix} 12.62 & 0 & 0 & 0 & 0 & 0\,0 \\ 0 & 10.66 & 0 & 0 & 0 & 0\,0 \\ 0 & 0 & 7.29 & 0 & 0 & 0\,0 \\ 0 & 0 & 0 & 1.64 & 0 & 0\,0 \\ 0 & 0 & 0 & 0 & 0.95 & 0\,0 \\ 0 & 0 & 0 & 0 & 0 & 0\,0 \end{pmatrix},$$

$$V^T = \begin{pmatrix} 0.32 & 0.41 & -0.24 & 0.36 & 0.07 & 0.70 & 0.13 \\ 0.32 & 0.41 & -0.24 & 0.36 & 0.07 & -0.62 & -0.35 \\ 0.26 & 0.37 & -0.32 & -0.79 & -0.12 & -0.06 & 0.17 \\ 0.50 & 0.01 & 0.55 & 0.05 & 0.24 & -0.21 & 0.57 \\ 0.41 & 0.01 & 0.50 & -0.14 & -0.42 & 0.21 & -0.57 \\ 0.42 & -0.53 & -0.27 & -0.15 & 0.57 & 0.10 & -0.28 \\ 0.33 & -0.46 & -0.36 & 0.21 & -0.63 & -0.10 & 0.28 \end{pmatrix}.$$

It can be seen that the greatest weight have the first three singular values, which is confirmed by the calculations:

$$\frac{\sum_{i=1}^{3} \sigma_i^2}{\sum \sigma_i^2} \cdot 100\% \approx 99\%.$$

### 2.2 Non-negative matrix factorisation (NMF)

Non-negative Matrix Factorization (NMF) is a decomposition of non-negative matrix $V \in \mathbb{R}^{n \times m}$ for a given number $k$ into the product of two non-negative matrices $W \in \mathbb{R}^{n \times k}$ and $H \in \mathbb{R}^{k \times m}$ such that

$$V \approx WH. \tag{2}$$

NMF is widely used in such areas as finding the basis vectors for images, discovering molecular structures, etc. [16].

Consider the following matrix of ratings:

$$V = \begin{pmatrix} 4\,4\,5\,0\,0\,0\,0 \\ 5\,5\,3\,4\,3\,0\,0 \\ 0\,0\,0\,4\,4\,0\,0 \\ 0\,0\,0\,5\,4\,5\,3 \\ 0\,0\,0\,0\,0\,5\,5 \\ 0\,0\,0\,0\,0\,4\,4 \end{pmatrix}.$$

Its decomposition into the product of two non-negative matrices for $k = 3$ can be, for example, like this:

$$V = \begin{pmatrix} 2.34 & 0 & 0 \\ 2.32 & 1.11 & 0 \\ 0 & 1.28 & 0 \\ 0 & 1.46 & 1.23 \\ 0 & 0 & 1.60 \\ 0 & 0 & 1.28 \end{pmatrix} \cdot \begin{pmatrix} 1.89 & 1.89 & 1.71 & 0.06 & 0 & 0 & 0 \\ 0.13 & 0.13 & 0 & 3.31 & 2.84 & 0.27 & 0 \\ 0 & 0 & 0 & 0.03 & 0 & 3.27 & 2.93 \end{pmatrix}.$$

### 2.3 Boolean Matrix Factorisation (BMF) based on Formal Concept Analysis (FCA)

**Basic FCA definitions.** Formal Concept Analysis (FCA) is a branch of applied mathematics and it studies (formal) concepts and their hierarchy. The adjective "formal" indicates a strict mathematical definition of a pair of sets, called, the extent and intent. This formalisation is possible because the use of the algebraic lattice theory.

Definition 1. *Formal context* $K$ is a triple $(G, M, I)$, where $G$ is the set of *objects*, $M$ is the set of *attributes* , $I \subseteq G \times M$ is a binary relation.

The binary relation $I$ is interpreted as follows: for $g \in G$, $m \in M$ we write $gIm$ if the object $g$ has the attribute $m$.

For a formal context $\mathbb{K} = (G, M, I)$ and any $A \subseteq G$ and $B \subseteq M$ a pair of mappings is defined:

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

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

these mappings define Galois connection between partially ordered sets $(2^G, \subseteq)$ and $(2^M, \subseteq)$ on disjunctive union of $G$ and $M$. The set $A$ is called *closed set*, if $A'' = A$ [5].

Definition 2. A *formal concept* of the formal context $\mathbb{K} = (G, M, I)$ is a pair $(A, B)$, where $A \subseteq G$, $B \subseteq M$, $A' = B$ and $B' = A$. Set $A$ is called the *extent*, and $B$ is the *intent* of the formal concept $(A, B)$.

It is evident that extent and intent of any formal concept are closed sets.

The set of formal concepts of a context $\mathbb{K}$ is denoted by $\mathfrak{B}(G, M, I)$.

**Description of FCA-based BMF** Boolean matrix factorization (BMF) is a decomposition of the original matrix $I \in \{0,1\}^{n \times m}$, where $I_{ij} \in \{0,1\}$, into a Boolean matrix product $P \circ Q$ of binary matrices $P \in \{0,1\}^{n \times k}$ and $Q \in \{0,1\}^{k \times m}$ for the smallest possible number of $k$. We define boolean matrix product as follows:

$$(P \circ Q)_{ij} = \bigvee_{l=1}^{k} P_{il} \cdot Q_{lj},$$

where $\bigvee$ denotes disjunction, and $\cdot$ conjunction.

Matrix $I$ can be considered as a matrix of binary relations between set $X$ objects (users), and the set $Y$ attributes (items that users have evaluated). We assume that $xIy$ iff user $x$ estimated object $y$. The triple $(X, Y, I)$ is clearly composes a formal context.

Consider the set $\mathcal{F} \subseteq \mathcal{B}(X, Y, I)$, a subset of all formal concepts of context $(X, Y, I)$, and introduce the matrices $P_{\mathcal{F}}$ and $Q_{\mathcal{F}}$ :

$$(P_{\mathcal{F}})_{il} = \begin{cases} 1, i \in A_l, \\ 0, i \notin A_l, \end{cases} \quad (Q_{\mathcal{F}})_{lj} = \begin{cases} 1, j \in B_l, \\ 0, j \notin B_l. \end{cases}$$

We can consider the decomposition of the matrix $I$ into binary matrix product $P_{\mathcal{F}}$ and $Q_{\mathcal{F}}$ as described above. The following theorems are proved in [2]:

Theorem 1. (Universality of formal concepts as factors). For every $I$ there is $\mathcal{F} \subseteq \mathcal{B}(X, Y, I)$, such that $I = P_{\mathcal{F}} \circ Q_{\mathcal{F}}$.

Theorem 2. (Optimality of formal concepts as factors). Let $I = P \circ Q$ for $n \times k$ and $k \times m$ binary matrices $P$ and $Q$. Then there exists a $\mathcal{F} \subseteq \mathcal{B}(X, Y, I)$ of formal concepts of $I$ such that $|\mathcal{F}| \leq k$ and for the $n \times |F|$ and $|F| \times m$ binary matrices $P_{\mathcal{F}}$ and $Q_{\mathcal{F}}$ we have $I = P_{\mathcal{F}} \circ Q_{\mathcal{F}}$.

There are several algorithms for finding $P_{\mathcal{F}}$ and $Q_{\mathcal{F}}$ by calculating formal concepts based on these theorems [2].

The algorithm we use (Algoritm 2 from [2]) avoid the computation of all the possible formal concepts and therefore works much faster [2]. Time estimation of the calculation algorithm in the worst case yields $O(k|G||M|^3)$, where $k$ is the number of found factors, $|G|$ is the number of objects, $|M|$ this the number of attributes.

Transform the matrix of ratings described above, to a boolean matrix, as follows:

$$\begin{pmatrix} 4&4&5&0&0&0&0\\ 5&5&3&4&3&0&0\\ 0&0&0&4&4&0&0\\ 0&0&0&5&4&5&3\\ 0&0&0&0&0&5&5\\ 0&0&0&0&0&4&4 \end{pmatrix} \Rightarrow \begin{pmatrix} 1&1&1&0&0&0&0\\ 1&1&1&1&1&0&0\\ 0&0&0&1&1&0&0\\ 0&0&0&1&1&1&1\\ 0&0&0&0&0&1&1\\ 0&0&0&0&0&1&1 \end{pmatrix} = I.$$

The decomposition of the matrix $I$ into the Boolean product of $I = A_{\mathcal{F}} \circ B_{\mathcal{F}}$ is the following:

$$\begin{pmatrix} 1&1&1&0&0&0&0\\ 1&1&1&1&1&0&0\\ 0&0&0&1&1&0&0\\ 0&0&0&1&1&1&1\\ 0&0&0&0&0&1&1\\ 0&0&0&0&0&1&1 \end{pmatrix} = \begin{pmatrix} 1&0&0\\ 1&1&0\\ 0&1&0\\ 0&1&1\\ 0&0&1\\ 0&0&1 \end{pmatrix} \circ \begin{pmatrix} 1&1&1&0&0&0&0\\ 0&0&0&1&1&0&0\\ 0&0&0&0&0&1&1 \end{pmatrix}.$$

This example shows that the algorithm has identified three factors that significantly reduces the dimensionality of the data.

### 2.4 General scheme of user-based recommendations

Once the matrix of rates is factorized we need to learn how to compute recommendations for users and to evaluate whether a particular method handles well with this task.

For factorized matrices the already well-known algorithm based on the similarity of users can be applied, where for finding $K$ nearest neighbors we use not the original matrix of ratings $A \in \mathbb{R}^{m \times n}$, but the matrix $U \in \mathbb{R}^{m \times f}$, where $m$ is a number of users, and $f$ is a number of factors. After selection of $K$ users, which are the most similar to a given user, based on the factors that are peculiar to them, it is possible, based on collaborative filtering formulas to calculate the projected rates for a given user.

After formation of the recommendations the performance of the recommendation system can be estimated by measures such as Mean Absolute Error (MAE), Precision and Recall.

## 3 A recommender algorithm using FCA-based BMF

### 3.1 kNN-based algorithm

Collaborative recommender systems try to predict the utility of items for a particular user based on the items previously rated by other users.

Denote $u(c, s)$ the utility of item $s$ for user $c$. $u(c, s)$ is estimated based on the utilities $u(c_i, s)$ assigned to item $s$ by those users $c_i \in C$ who are "similar" to user $c$. For example, in a movie recommendation application, in order to recommend movies to user $c$, the collaborative recommender system finds the users that have similar tastes in movies with $c$ (rate the same movies similarly). Then, only the movies that are most liked by those similar users would be recommended.

Memory-based recommendation system, which are based on the previous history of the ratings, are one of the key classes of collaborative recommendation systems.

Memory-based algorithms make rating predictions based on the entire collection of previously rated items by the users. That is, the value of the unknown rating $r_{c,s}$ for user $c$ and item $s$ is usually computed as an aggregate of the ratings of some other (usually, the $K$ most similar) users for the same item $s$:

$$r_{c,s} = aggr_{c' \in \widehat{C}} r_{c,s},$$

where $\widehat{C}$ denotes the set of $K$ users that are the most similar to user $c$ , who have rated item $s$. For example, function $aggr$ may has the following form [1]

$$r_{c,s} = k \sum_{c' \in \widehat{C}} sim(c', c) \times r_{c',s},$$

where $k$ serves as a normalizing factor and selected as $k = 1/\sum_{c' \in \widehat{C}} sim(c, c')$.

The similarity measure between users $c$ and $c'$, $sim(c, c')$, is essentially a distance measure and is used as a weight, i.e., the more similar users $c$ and $c'$ are, the more weight rating $r_{c',s}$ will carry in the prediction of $r_{c,s}$.

The similarity between two users is based on their ratings of items that both users have rated. The two most popular approaches are *correlation* and *cosine-based*. One common strategy is to calculate all user similarities $sim(x, y)$ in advance and recalculate them only once in a while (since the network of peers usually does not change dramatically in a short time). Then, whenever the user asks for a recommendation, the ratings can be calculated on demand using precomputed similarities.

To apply this approach in case of FCA-based BMF recommender algorithm we simply consider as an input the user-factor matrices obtained after factorisation of the initial data.

### 3.2 Scaling

In order to move from a matrix of ratings to a Boolean matrix, and use the results of Boolean matrix factorization, scaling is required. It is well known that

scaling is a matter of expert interpretation of original data. In this paper, we use several variants of scaling and compare the results in terms of MAE.

1. $I_{ij} = 1$ if $R_{ij} > 0$, else $I_{ij} = 0$ (user $i$ rates item $j$).
2. $I_{ij} = 1$ if $R_{ij} > 1$, else $I_{ij} = 0$.
3. $I_{ij} = 1$ if $R_{ij} > 2$, else $I_{ij} = 0$.
4. $I_{ij} = 1$ if $R_{ij} > 3$, else $I_{ij} = 0$.

## 4 Experiments

To test our hypotheses and study the behavior of recommendations based on the factorization of a ratings matrix by different methods we used MovieLens data. We used the part of data, containing 100,000 ratings, while considered only users who have given over 20 ratings.

User ratings are split into two sets, a training set consisting of 80 000 ratings, and test set consisting of 20 000 ratings. Original data matrix is 943× 1682, where the number of rows is the number of users and the number of columns is the number of rated movies (each film has at least one vote).

### 4.1 The number of factors that cover $p$% of evaluations in an input data for SVD and BMF

The main purpose of matrix factorization is a reduction of matrices dimensionality. Therefore we examine how the number of factors varies depending on the method of factorization, and depending on $p$ % of the data that is covered by factorization. For BMF the coverage of a matrix is calculated as the ratio of the number of ratings covered by Boolean factorization to the total number of ratings.

$$\frac{|covered\_ratings|}{|all\_ratings|} \cdot 100\% \approx p_{BMF}\%, \tag{3}$$

For SVD we use the following formula:

$$\frac{\sum\limits_{i=1}^{K} \sigma_i^2}{\sum \sigma_i^2} \cdot 100\% \approx p_{SVD}\%, \tag{4}$$

where $K$ is the number of factors selected.

**Table 2.** Number of factors for SVD and BMF at different coverage level

| p% | 100% | 80% | 60% |
|---|---|---|---|
| SVD | 943 | 175 | 67 |
| BMF | 1302 | 402 | 223 |

### 4.2 MAE-based recommender quality comparison of SVD and BMF for various levels of evaluations coverage

The main purpose of matrix factorisation is a reduction of matrices dimensionality. As a result some part of the original data remains uncovered, so it was interesting to explore how the quality of recommendations changes based on different factorisations, depending on the proportion of the data covered by factors.

Two methods of matrix factorisation were considered: BMF and SVD. The fraction of data covered by factors for SVD was calculated as

$$p\% = \frac{\sum_{i=1}^{K} \sigma_i^2}{\sum \sigma_i^2} \cdot 100\%,$$

and for BMF as

$$p\% = \frac{|covered\_ratings|}{|all\_ratings|} \cdot 100\%.$$

To quality assessment we chose $MAE$.

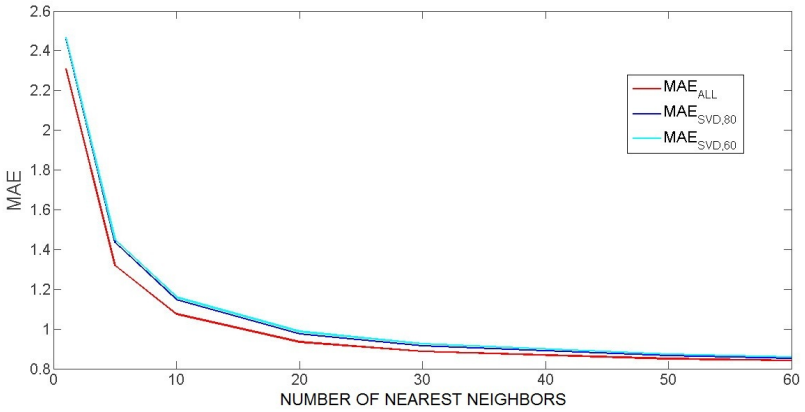



**Fig. 1.** MAE dependence on the percentage of the data covered by SVD-decomposition, and the number of nearest neighbors.

Fig. 1 shows that $MAE_{SVD60}$, calculated for the model based on 60% of factors, is not very different from $MAE_{SVD80}$, calculated for the model built for 80% factors. At the same time, for the recommendations based on a Boolean factorization covering 60% and 80% of the data respectively, it is clear that increasing the number of factors improves MAE, as shown in Fig. 2.

Table 3 shows that the MAE for recommendations built on a Boolean factorisation covering 80 % of the data for the number of neighbors less than 50 is better than the MAE for recommendations built on SVD factorization. It is also

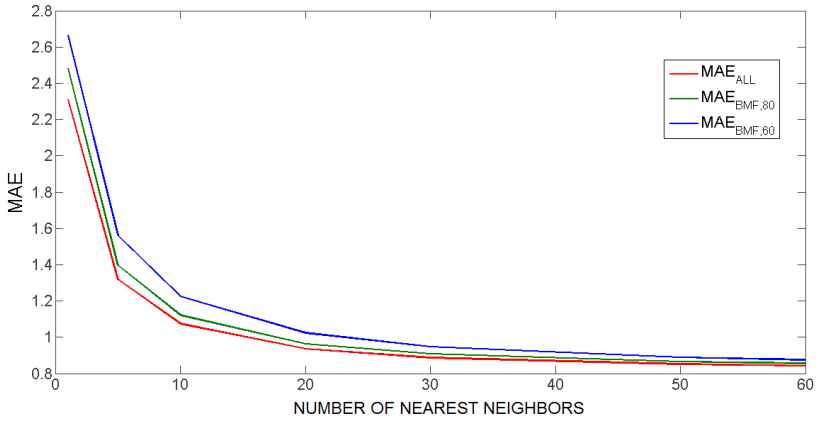



**Fig. 2.** MAE dependence on the percentage of the data covered by BMF-decomposition, and the number of nearest neighbors.

**Table 3.** MAE for SVD and BMF at 80% coverage level

| Number of neighbors | 1 | 5 | 10 | 20 | 30 | 50 | 60 |
|---|---|---|---|---|---|---|---|
| $MAE_{SVD80}$ | 2,4604 | 1.4355 | 1.1479 | 0.9750 | 0.9148 | 0.8652 | 0.8534 |
| $MAE_{BMF80}$ | 2.4813 | 1.3960 | 1.1215 | 0.9624 | 0.9093 | 0.8650 | 0.8552 |
| $MAE_{all}$ | 2.3091 | 1.3185 | 1.0744 | 0.9350 | 0.8864 | 0.8509 | 0.8410 |

easy to see that $MAE_{SVD80}$ and $MAE_{BMF80}$ are different from $MAE_{all}$ in no more than $1-7\%$.

### 4.3 Comparison of kNN-based approach and BMF by Precision and Recall

Besides comparison of algorithms using MAE other evaluation metrics can also be exploited, for example

$$Recall = \frac{|objects\_in\_recommendation \cap objects\_in\_test|}{|objects\_in\_test|},$$

$$Precision = \frac{|objects\_in\_recommendation \cap objects\_in\_test|}{|objects\_in\_recommendation|}$$

and

$$F1 = \frac{2 \cdot Recall \cdot Precision}{Recall + Precision}.$$

It is widely spread belief that the larger Recall, Precision and F1 are, the better is recommendation algorithm.

Figures 3, 4 and 5 show the dependence of relevant evaluation metrics on the percentage of the data covered by BMF-decomposition, and the number of nearest neighbors. The number of objects to recommend was chosen to be 20. The figures show that the recommendation based on the Boolean decomposition, is worse than recommendations built on the full matrix of ratings.

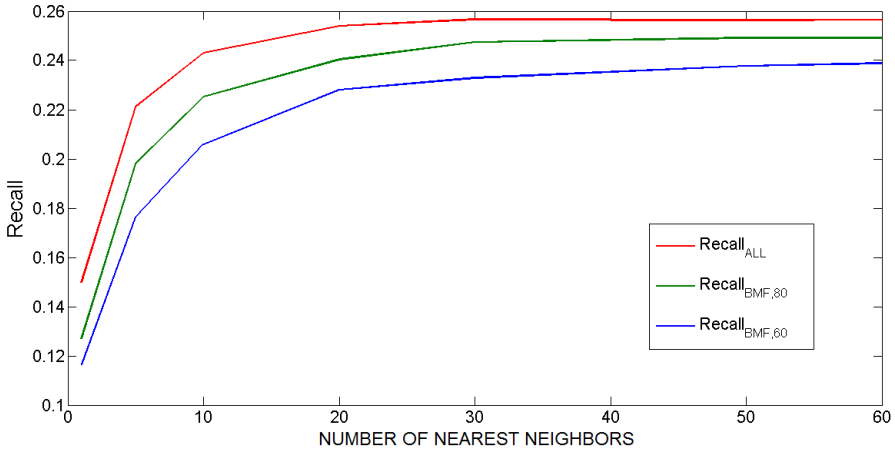



**Fig. 3.** *Recall dependence on the percentage of data covered by BMF-decomposition, and the number of nearest neighbors.*

### 4.4 Scaling influence on the recommendations quality for BMF in terms of MAE

Another thing that was interesting to examine was the impact of scaling described in 3.2 on the quality of recommendations. Four options of scaling were considered:

1. $I_{0,ij} = 1$ if $A_{ij} > 0$, else $I_{ij} = 0$ (user rates an item).
2. $I_{1,ij} = 1$ if $A_{ij} > 1$, else $I_{ij} = 0$.
3. $I_{2,ij} = 1$ if $A_{ij} > 2$, else $I_{ij} = 0$.
4. $I_{3,ij} = 1$ if $A_{ij} > 3$, else $I_{ij} = 0$.

The distribution of ratings in data is on Figure 6

For each of the boolean matrices we calculate its Boolean factorisation, covering 60 % and 80 % of the data. Then recommendations are calculated just like in 4.2. It can be seen that for both types of data coverage $MAE_1$ is almost the same as $MAE_0$, and $MAE_{2,3}$ is better than $MAE_0$.

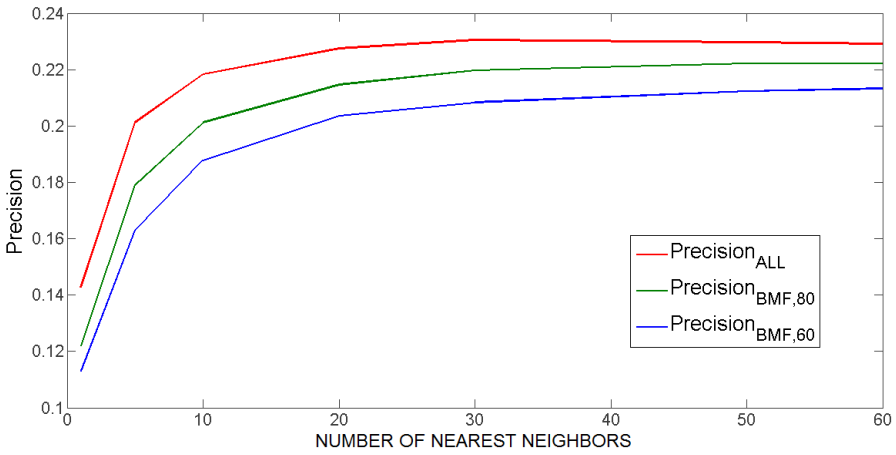



**Fig. 4.** *Precision dependence on the percentage of data covered by BMF-decomposition, and the number of nearest neighbors.*

### 4.5 Influence of data filtering on MAE for BMF kNN-based approach

Besides the ability to search for $K$ nearest neighbors not in the full matrix of ratings $A \in \mathbb{R}^{n \times m}$, but in the matrix $U \in \mathbb{R}^{m \times f}$, where $m$ is a number of users, and $f$ is a number of factors, Boolean matrix factorization can be used to data filtering. Because the algorithm returns as an output not only matrices users-factors and factors-objects, but also the ratings that were not used for factoring, we can try to search for users, similar to the user, on the matrix consisting only of ratings used for the factorization.

Just as before to find the nearest neighbors cosine measure is used, and the predicted ratings are calculated as the weighted sum of the ratings of nearest users. Figure 9 shows that the smaller the data we use for filtering the bigger is MAE. Figure 10 shows that the recommendations built on user-factor matrix, are better then recommendations, constructed on matrix of ratings filtered with boolean factorization.

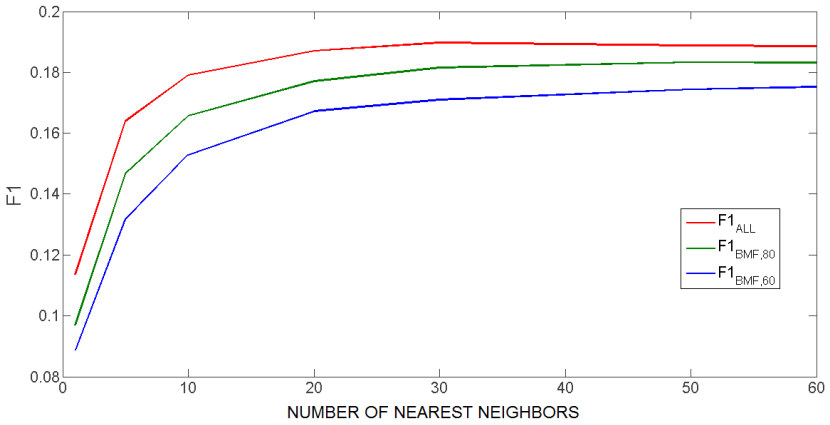



**Fig. 5.** *F1 dependence on the percentage of data covered by BMF-decomposition, and the number of nearest neighbors.*

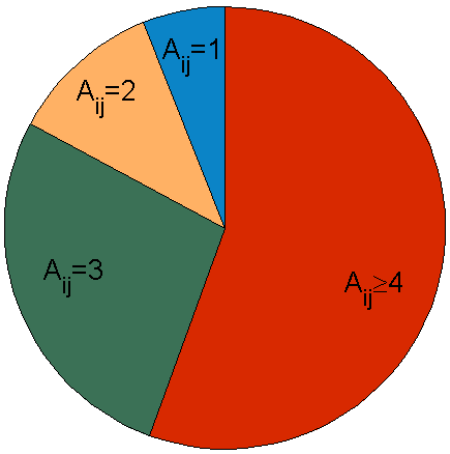



**Fig. 6.** *Ratings distribution in data.*

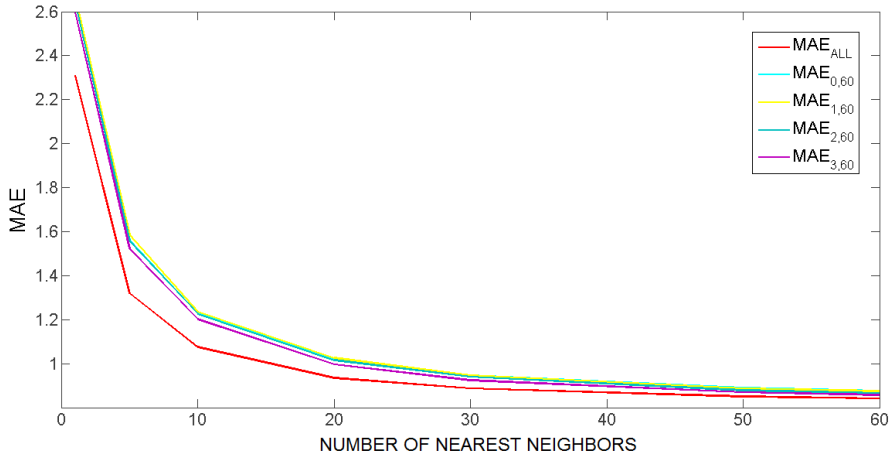



**Fig. 7.** MAE dependance on scaling and number of nearest neighbors for 60% coverage.

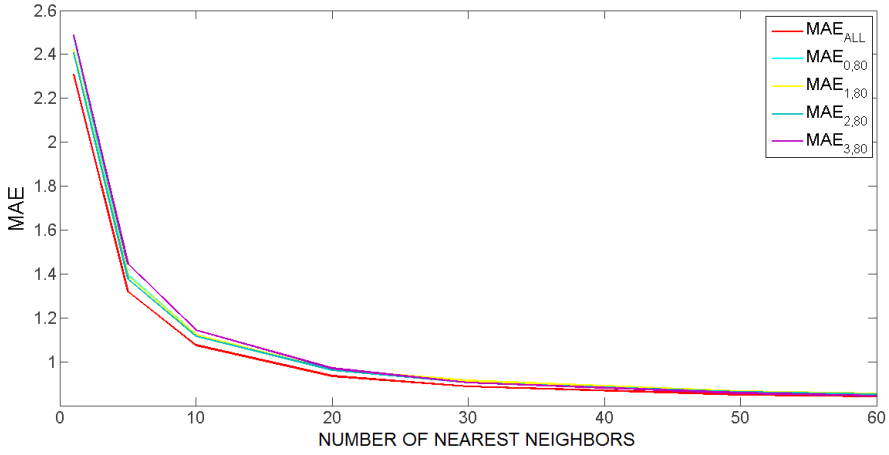



**Fig. 8.** MAE dependance on scaling and number of nearest neighbors for 80% coverage.

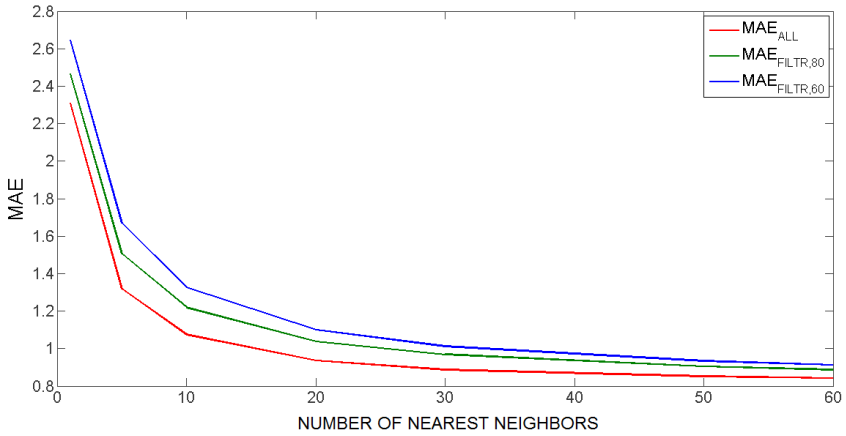



**Fig. 9.** *MAE dependance on percentage of covered with filtration data and the number of nearest neighbors.*

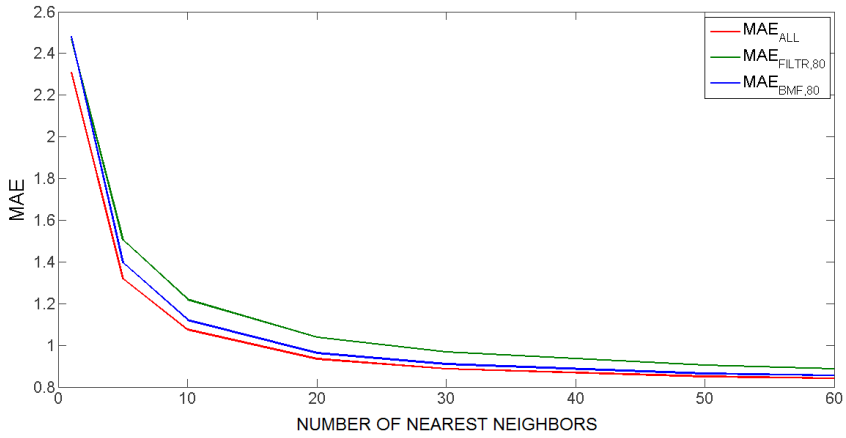



**Fig. 10.** *MAE dependance on data filtration algorithm and the number of nearest neighbors.*

## 5 Conclusion

In the paper we considered main methods of Matrix Factorisation which are suitable for Recommender Systems. Some of these methods were compared on real datasets. We investigated BMF behaviour as part of recommender algorithm. We also conducted several experiments on recommender quality comparison with numeric matrices, user-factor and factor-item matrices in terms of Recall, Precision and MAE. We showed that MAE of our BMF-based approach is not greater than MAE of SVD-based approach for the same number of factors on the real data. For methods that require the number of factors as an initial parameter in the user or item profile (e.g., NMF), we proposed the way of finding this number with FCA-based BMF. We also have investigated how data filtering, namely scaling, influences on recommendations' quality.

As a further research direction we would like to investigate the proposed approaches in case of graded and triadic data [3,4] and reveal whether there are some benefits for the algorithm's quality in usage least-squares data imputation techniques [19]. In the context of matrix factorisation we also would like to test our approach on the quality assessment of recommender algorithms that we performed on some basic algorithms (see bimodal cross-validation in [12]).

**Acknowledgments.** We would like to thank Radim Belohlavek, Vilem Vychodil and Sergei Kuznetsov for their comments, remarks and explicit and implicit help during the paper preparations. We also express our gratitude to Gulnaz Bagautdinova; she did her bachelor studies under the second author supervision on a similar topic and therefore contributed somehow to this paper.

## References

1. Adomavicius, G., Tuzhilin, A.: Toward the Next Generation of Recommender Systems: A Survey of the State-of-the-Art and Possible Extensions. IEEE Transactions on Knowledge and data engineering, vol.17 (6) (2005)
2. Belohlavek, R., Vychodil, V.: Discovery of optimal factors in binary data via a novel method of matrix decomposition, Journal of Computer and System Scinces, 76 (2010)
3. Belohlavek, R., Osicka, P.: Triadic concept lattices of data with graded attributes. Int. J. General Systems 41(2), 93-108 (2012)
4. Belohlavek, R.: Optimal decompositions of matrices with entries from residuated lattices. J. Log. Comput. 22(6), 1405-1425 (2012)
5. Birkhoff, G.: Lattice Theory, eleventh printing, Harvard University, Cambridge, MA (2011)
6. du Boucher-Ryan, P., Bridge, D.G. : Collaborative Recommending using Formal Concept Analysis. Knowl.-Based Syst. 19(5), 309-315 (2006)
7. Elden, L.: Matrix Methods in Data Mining and Pattern Recognition, Society for Industrial and Applied Mathematics (2007)
8. Ganter, B., and Wille, R.: Formal Concept Analysis: Mathematical Foundations, Springer (1999)

9. Gaussier, E., Goutte, C.: Relation between PLSA and NMF and Implications. In: SIGIR'05: Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval. New York, NY, USA. ACM, pp. 601-602 (2005)
10. Ignatov, D.I., Kuznetsov, S.O.: Concept-based Recommendations for Internet Advertisement. In.: Proc. of The Sixth International Conference Concept Lattices and Their Applications (CLA'08), Radim Belohlavek, Sergei O. Kuznetsov (Eds.): CLA 2008, Palacky University, Olomouc, pp. 157166 (2008)
11. Ignatov, D.I., Kuznetsov, S.O., Poelmans, P.: Concept-Based Biclustering for Internet Advertisement. ICDM Workshops 2013, pp. 123-130 (2013)
12. Ignatov, D.I., Poelmans, J., Dedene, G., Viaene, S.: A New Cross-Validation Technique to Evaluate Quality of Recommender Systems. PerMIn 2012, pp. 195-202. (2012)
13. Lee, D.D., Seung, H.S.: Algorithms for Non-Negative matrix factorization, advances in Neural Information Processing Systems 13: Proceedings of the 2000 Conference. MIT Press. pp. 556562. (2000)
14. Leksin, A.V.: Probabilistic models in client environment analysis, PhD Thesis (2011) (In Russian)
15. Lloyd N. Trefethen and David Bau. Numerical Linear Algebra, 3rd edition, SIAM (1997)
16. Lin, Ch.-J: Projected Gradient Methods for Non-negative Matrix Factorization, Neural computation 19 (10), pp. 2756–2779 (2007)
17. Mirkin, B.G.: Core Concepts in Data Analysis: Summarization, Correlation, Visualization (2010)
18. Symeonidis,P., Nanopoulos, A., Papadopoulos, A., Manolopoulos, Ya.: Nearest-biclusters collaborative filtering based on constant and coherent values. Inf. Retr. 11(1): 51-75 (2008)
19. Wasito, I., Mirkin, B.: Nearest neighbours in least-squares data imputation algorithms with different missing patterns. Computational Statistics & Data Analysis 50(4), 926-949 (2006)
20. Zhou, G., Cichocki,A., Xie, Sh.: Fast Nonnegative Matrix/Tensor Factorization Based on Low-Rank Approximation. IEEE Transactions on Signal Processing 60(6), pp. 2928-2940 (2012)

# Information Retrieval and

# Knowledge Discovery with FCART

A.A. Neznanov, S.O. Kuznetsov

National Research University Higher School of Economics,
Pokrovskiy bd., 11, 109028, Moscow, Russia
ANeznanov@hse.ru, SKuznetsov@hse.ru

**Abstract.** We describe FCART software system, a universal integrated environment for knowledge and data engineers with a set of research tools based on Formal Concept Analysis. The system is intended for knowledge discovery from big dynamic data collections, including text collections. FCART allows the user to load structured and unstructured data (texts and various meta-information) from heterogeneous data sources, build data snapshots, compose queries, generate and visualize concept lattices, clusters, attribute dependencies, and other useful analytical artifacts. Full preprocessing scenario is considered.

**Keywords:** Data Analysis, Knowledge Extraction, Text Mining, Formal Concept Analysis

## 1 Introduction

We introduce a new software system for information retrieval and knowledge discovery from various data sources (textual data, structured databases, etc.). Formal Concept Analysis Research Toolbox (FCART) was designed especially for the analysis of unstructured (textual) data. The core of the system supports knowledge discovery techniques, including those based on Formal Concept Analysis [1], clustering [2], multimodal clustering [2, 3], pattern structures [4, 5] and others. In case studies we applied FCART for analyzing data in medicine, criminalistics, and trend detection.

FCART is based on DOD-DMS (The Dynamic Ontology-driven Data Mining System) software platform. In case studies we applied DOD-DMS for analyzing data in the fields of medical informatics and trends detection. The core of the system complements a traditional knowledge extraction process with methods of clustering, multimodal clustering, Formal Concept Analysis, Hidden Markov chains, pattern structures and others.

Currently, there are several well-known open source FCA-based tools, such as ConExp [6], Conexp-clj [7], Galicia [8], Tockit [9], ToscanaJ [10], FCAStone [11], Lattice Miner [12], OpenFCA [13], Coron [14]. These tools have many advantages. However, they cannot completely satisfy the growing demands of the scientific com-

munity. One of the common drawbacks of these systems is poor data preprocessing. It prevents researchers from using the programs for analyzing complex big data without additional third party preprocessing tools.

For example, Coron has some tools for filtering objects and attributes, merging and transforming contexts (http://coron.wikidot.com/pre:filterdb), but Coron does not provide flexible tools for importing external data.

## 2 Methodology

The DOD-DMS is a universal and extensible software platform intended for building data mining and knowledge discovery tools for various application fields. The creation of this platform was inspired by the CORDIET methodology (abbreviation of Concept Relation Discovery and Innovation Enabling Technology) [15] developed by J. Poelmans at K.U. Leuven and P. Elzinga at the Amsterdam-Amstelland police. The methodology allows one to obtain new knowledge from data in an iterative ontology-driven process. The software is based on modern methods and algorithms of data analysis, technologies for processing big data collections, data visualization, reporting, and interactive processing techniques. It implements several basic principles:

1. Iterative process of data analysis using ontology-driven queries and interactive artifacts (such as concept lattice, clusters, etc.).
2. Separation of 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 on three levels: customizing settings of data access components, query builders, solvers and visualizers; writing scripts (macros); developing components (add-ins).
4. Explicit definition of analytic artifacts and their types. It allows one to check the integrity of session data and provides links between artifacts for end-user.
5. Realization of integrated performance estimation tools.
6. Integrated documentation of software tools and methods of data analysis.

FCART uses all these principles, but does not have an ontology editor and does not support the full C-K cycle. The current version consists of the following components.

- Core component including
  - multidocument user interface of research environment with session manager,
  - snapshot profiles editor (SHPE),
  - snapshot query editor (SHQE),
  - query rules database (RDB),
  - session database (SDB),
  - main part of report builder;
- Local XML-storage for preprocessed data;
- Internal solvers and visualizers;
- Additional plugins and scripts.

## 3 Current software properties and future work

Now we introduce version 0.8 of DOD-DMS as a local Windows application and version 0.4 as a distributed Web-based application. Those versions use local XML-storage for accumulating snapshots and integrated research environment with snapshot profiles editor, query builder, ontology editor, and some set of solvers (artifact builders) and visualizers (artifact browsers). The main solvers for this time can produce clusters, biclusters, concept lattice, sublattices, association rules, and implications, calculate stability indexes, similarity measures for contexts and concepts, etc. The set of solvers, visualizers, and scripts specifies a subject field of DOD-DMS edition.

We use Microsoft and Embarcadero programing environments and different programing languages (C++, C#, Delphi, Python and others). For scripting we use Delphi Web Script [16] and Python [17].

## 4 Data preprocessing in FCART

### 4.1 Obtaining initial artifacts

There are several ways to obtain a binary context, the basic FCA artifact:

- Load from ready data files of supported formats like CXT or CSV,
- Generate by plugin or script,
- Query from data snapshots.

Loading contexts from ready data files is supported by most FCA-tools. The most interesting way to obtain a context is querying from snapshots. Let us look to all steps needed to convert external data into some set of objects with binary attributes.

### 4.2 Access to external data sources and generating snapshots

Local storage of FCART can be filled from various data sources. System supports SQL, XML and JSON sources, so it can load data from most databases and Web-services.

Data snapshot (or snapshot) is a data table with structured and text attributes, loaded in the local storage by accessing external data sources. Snapshot is described by a set of metadata: snapshot profile, local path, link to external data source, time stamp, author, and comment. FCART provides one with a *snapshot profile editor* (SHPE). *Profile* consists of definitions of fields. Each element of a snapshot is a record: array of values of fields. Each field is defined by the following main properties:

- Id (identifier of field)
- Path (path in initial XML or JSON – may be empty)
- Name (user-friendly name of field)
- Group (for visual grouping of fields)

- Comment
- Data type (Boolean / Integer / Float / Text / Binary / DateTime)
- Is Unstructured? (field can be interpreted as unstructured text)
- Is Multivalued? (for sets / arrays)
- Type of multivalued presentation (delimited content / same path / path in form of "name + number")

Consider the following example of XML file:

```
<?xml version="1.0" encoding="utf-8"?>
<Data>
    // ...
  <Genre>Lounge</Genre>
  <Genre>Easy listening</Genre>
    // ...
</Data>
```

In this example field "Genre" is multivalued and have multivalued presentation type "same path" (Path = "<Data>/<Genre>"). But in other source we can have type "name + number" (Path = "<Data>/<Genre%d>"):

```
<Data>
    // ...
  <Genre01>Lounge</Genre01>
  <Genre02>Easy listening</Genre02>
    // ...
</Data>
```

Unstructured field definition additionally contains the following properties:

- Language (main language of text)
- SW (list of stop words)
- Stemmer (not required now because we use snowball stemmer from Lucene).

It is very useful for dealing with dynamic data collections, including texts in natural language, and helps to query full-text data more effectively. There is a sample of unstructured and multivalued field description in JSON format:

```
{       "Id": "02",
        "Path": "object\author",
        "Caption": "Artwork Creators",
        "Group": "Common",
        "Comment": "The sequence of authors",
        "DataType": "Text",
        "Unstruct": {
                "Is": true,
                "Language": "English",
                "StopWords": [ ],
                "Stemmer": "Snowball"       },
        "MV": {  "Is": true,
                "MVType": "Vector",
                "MVRepresentation": "NameNumber",
                "MVFormat": "author%d"    }
```

}

Fig. 1 shows a variant of snapshot profile editor (SHPE) for XML filtering. The left pane “XML Structure” displays a sample of an XML-document from a dataset. A user can select any element from the document, add it to profile as a new field and set properties of the field.

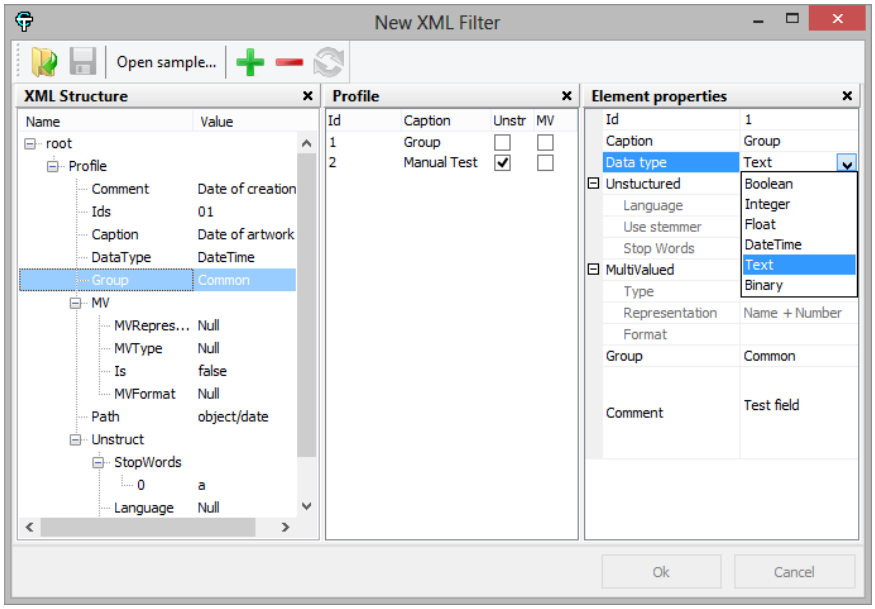



**Fig. 1.** Profile editor (profile by example)

### 4.3 Queries to snapshots and constructing binary contexts

The system has query language for transforming snapshots into binary formal context. This language describes so-called rules. Main rule types are the following:

- *Simple rule* generates one attribute from atomic fields of a snapshot. This rule type has syntax very similar to SQL WHERE clause
- *Scaling rule* generates several attributes from atomic fields based on nominal or ordinal scale
- *Text mining rule* generates one attribute from unstructured text fields.
- *Multivalued rule* generates one or many attributes from multivalued field (arrays and sets)
- *Compound rule* merges rules of all types into a single rule. This rule uses standard logical operations and brackets to combine elements.

We have also implemented additional rule types: *Temporal rules* are used for manipulating date and time intervals and *Filters* are used for removing objects with their intents from contexts.

In most cases, it is not necessary to write a query from scratch. One can select some entities in rules DB (RDB) and automatically generate a query. It is possible because the RDB is aware of dependencies between rules. Each rule type has XML presentation, so every query (or full RDB) can be imported and exported as an XML-file.

The following XML file is a sample of the scaling rule:

```
<scale name="Age" ScaleType="Order" DataType="Integer"
Ends="Open" id="t34">
  <Offset1>8</Offset1>
  <Offset2>16</Offset2>
  <Offset3>35</Offset3>
  <Offset4>60</Offset4>
</scale>
```

The application of this rule to snapshot generates 5 binary attributes: “Age < 8”, “8 <= Age < 16”, …, “60 <= Age”.

FCART uses Lucene full text search engine [18] to index the content of unstructured text fields in snapshots. The resulting index is later used to validate quickly whether the text mining or compound rule returns true or false.

## 5 Interactive visualization of concept lattice

The *concept lattice visualizer* is an example of interactive visualizer. It can be used to browse the collection of objects with binary attributes given as a result of query to snapshot (with structured and text attributes). The user can select and deselect objects and attributes and the lattice diagram is modified accordingly. The user can click on a concept. In a special window the screen shows names of objects in the extent and names of attributes in the intent. Names of objects and attributes are linked with initial snapshot records and fields. If the user clicks on the name of an object or an attribute, the content of the object or attribute description is shown in a special window according to the snapshot profile.

Fig. 2 demonstrates the result of building a sublattice from a concept lattice. The multi-document interface allows us to inspect several artifacts, so a sublattice will be opened in a new window.

The user can customize settings of lattice browsing in various ways. The user can specify whether nodes corresponding to concepts show numbers of all (or only new) objects and all (or only new) attributes in extent and intent respectively, or names of all (or only new) objects and all (or only new) attributes. Separate settings can be specified for the selected concept, concepts in the order filter, and the remainder of the lattice. The visual appearance can be changed: zooming, coloring, and other tools are available.

Right clicking on the name of an attribute user can choose several options: one can build a sublattice containing only objects with selected attribute; build a sublattice containing only objects without selected attribute; or find the highest concept with a selected attribute. Right clicking on the name of an object allows one the same actions.

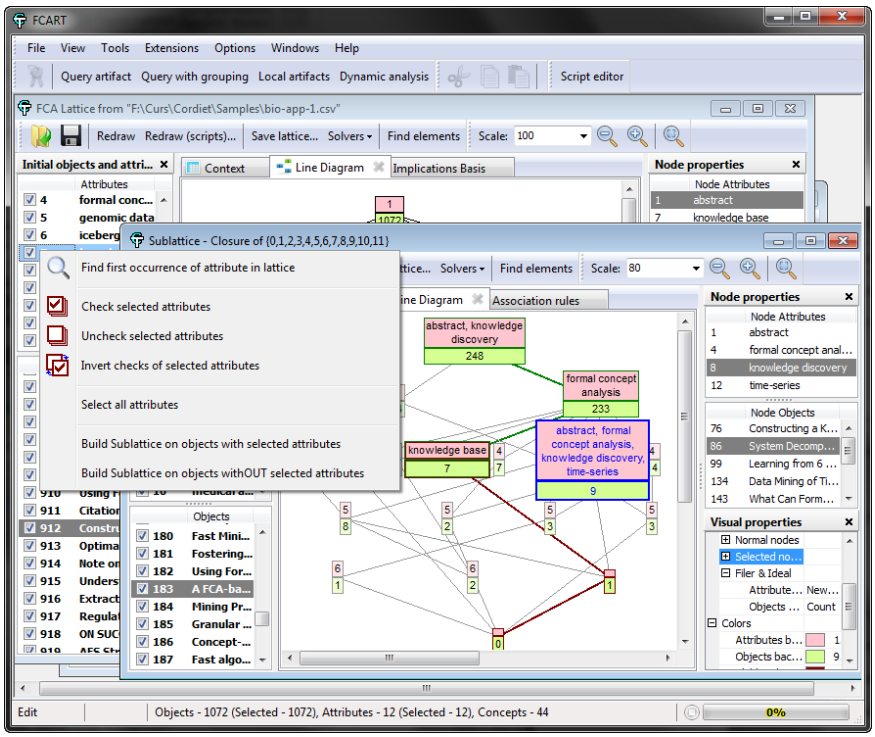



**Fig. 2.** Concept lattice visualizer

If we built a formal context using a query to a snapshot then we can simply look for a definition of each attribute (in form of a query rule from RDB) and a source of each object (in form of XML- or JSON-file) in left part of the visualizer window. If a filter rule is present in query then we can see comparison between sets of objects in the context and records in the snapshot.

Linking analytic artifacts with snapshots are very important for subsequent analysis of the same data collection. Researcher can simply interpret results of the analysis by viewing initial pieces of data.

## 6 Conclusion and future work

FCART is a powerful environment being in active development. The next major release of the local version 0.8 is planned for March 2013 and after that the system will be freely available to the FCA community. In this article we considered in details the powerful preprocessing tools of the system.

We intend to improve methodology, extend the set of solvers, optimize some algorithms, and use the proposed system for solving various knowledge discovery problems. We already have tested new solvers based on concept stability [19, 20] and other indices [21]. In the preprocessing queue we will try to simplify writing queries to external data sources by introducing SQL- and XML-explorer of databases and web-services.

## 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. Mirkin, B. Mathematical Classification and Clustering, Springer, 1996.
3. Ignatov, D.I., Kuznetsov, S.O., Magizov, R.A., Zhukov, L.E. From Triconcepts to Triclusters. Proc. of 13th International Conference on rough sets, fuzzy sets, data mining and granular computing (RSFDGrC-2011), LNCS/LNAI Volume 6743/2011, Springer (2011), pp. 257-264.
4. Ganter, B., Kuznetsov, S.O. Pattern Structures and Their Projections. Proc. of 9th International Conference on Conceptual Structures (ICCS-2001), 2001, pp. 129-142.
5. Kuznetsov, S.O. Pattern Structures for Analyzing Complex Data. Proc. of 12th International conference on Rough Sets, Fuzzy Sets, Data Mining and Granular Computing, Conference (RSFDGrC-2009), 2009, pp. 33-44.
6. Yevtushenko, S.A. System of data analysis "Concept Explorer". (In Russian). Proceedings of the 7th national conference on Artificial Intelligence KII-2000, p. 127-134, Russia, 2000.
7. Conexp-clj (http://daniel.kxpq.de/math/conexp-clj/)
8. Valtchev, P., Grosser, D., Roume, C. Mohamed Rouane Hacene. 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.
9. Tockit: Framework for Conceptual Knowledge Processing (http://www.tockit.org)
10. 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.

11. Priss, U. FcaStone - FCA file format conversion and interoperability software, Conceptual Structures Tool Interoperability Workshop (CS-TIW), 2008.
12. Lahcen, B., Kwuida, L. Lattice Miner: A Tool for Concept Lattice Construction and Exploration. In Suplementary Proceeding of International Conference on Formal concept analysis (ICFCA'10), 2010.
13. 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.
14. Szathmary, L., Kaytoue, M., Marcuola, F., Napoli, A., The Coron Data Mining Platform (http://coron.loria.fr)
15. Poelmans, J., Elzinga, P., Neznanov, 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.
16. Grange, E. DelphiWebScript Project (http://delphitools.info/dwscript)
17. Python Programming Language – Official Website (http://www.python.org)
18. Apache Lucene (http://lucene.apache.org)
19. Kuznetsov, S.O.: Stability as an Estimate of the Degree of Substantiation of Hypotheses on the Basis of Operational Similarity. In: Nauchno-Tekhnicheskaya Informatsiya, Ser. 2, Vol. 24, No. 12, pp. 21-29, 1990.
20. Kuznetsov, S.O., Obiedkov, S.A. and Roth, C., Reducing the Representation Complexity of Lattice-Based Taxonomies. In: U. Priss, S. Polovina, R. Hill, Eds., Proc. 15th International Conference on Conceptual Structures (ICCS 2007), Lecture Notes in Artificial Intelligence (Springer), Vol. 4604, pp. 241-254, 2007.
21. Klimushkin, M.A., Obiedkov, S.A., Roth, C.: Approaches to the Selection of Relevant Concepts in the Case of Noisy Data // 8th International Conference on Formal Concept Analysis (ICFCA 2010), pp. 255-266, 2010.

# Retrieval of Criminal Trajectories with an FCA-based Approach

Jonas Poelmans, Paul Elzinga[3], Guido Dedene[1,2]

[1]KU Leuven, Faculty of Business and Economics, Naamsestraat 69,
3000 Leuven, Belgium
[2]Universiteit van Amsterdam Business School, Roetersstraat 11
1018 WB Amsterdam, The Netherlands
[3]Amsterdam-Amstelland Police, James Wattstraat 84,
1000 CG Amsterdam, The Netherlands

Jonas.Poelmans@gmail.com
Paul.Elzinga@amsterdam.politie.nl
Guido.Dedene@econ.kuleuven.be

**Abstract.** In this paper we briefly discuss the possibilities of Formal Concept Analysis for gaining insight in large amounts of unstructured police reports. We present a generic human centred knowledge discovery approach and showcase promising results obtained during empirical validation. The first case study focusses on distilling indicators for identifying domestic violence from 4814 reports with the aim of better recognizing new incoming cases. In the second case study we used FCA in combination with Temporal Concept Analysis to identify and investigate human trafficking suspects extracted from 266157 short observational reports. The third case study we present in this paper describes our application of FCA for identifying radicalising subjects from 166577 observational police reports. Finally, we conclude our paper with the case study on pedophile chat conversation analysis and the CORDIET data mining system.

**Keywords.** Formal Concept Analysis, Security informatics, Human trafficking, Terrorism, Pedophiles, Domestic violence

## 1 Introduction

During the joint Knowledge Discovery in Databases project, the Katholieke Universiteit Leuven and the Amsterdam-Amstelland Police Department have developed new special investigations techniques for gaining insight in police databases. These methods have been empirically validated and their application resulted in new actionable knowledge which helps police forces to better cope with domestic violence, human trafficking, terrorism and pedophile related data.

The implementation of the Intelligence-led policing management paradigm by the Amsterdam-Amstelland Police Department has led to an annual increase of suspicious activity reports filed in the police databases. These reports contain observations made

by police officers on the street during police patrols and were entered as unstructured text in these databases. Until now this massive amount of information was barely used to obtain actionable knowledge which may help improve the way of working by the police. The main goal of this joint research project was to develop a system which can be operationally used to extract useful knowledge from large collections of unstructured information. The methods which were developed aimed at recognizing (new) potential suspects and victims better and faster as before. In this paper we describe in detail the four major projects which were undertaken during the past five years, namely domestic violence, human trafficking (sexual exploitation), terrorism (Muslim radicalization) and pedophile chat conversations. During this investigation a knowledge discovery suite was developed, Concept Relation Discovery and Innovation Enabling Technology (CORDIET). At the basis of this knowledge discovery suite is the C-K design theory developed in Hatchuel et al. (2004) which contains four major phases and transition steps each of them focusing on an essential aspect of exploring existing and discovering and applying new knowledge. The investigator plays an important role during the knowledge discovery process. In the first step he has to assess and decide which information should be used to create the visual data analysis artifacts. During the next step multiple facilities are provided to ease the exploration of the data. Subsequently the acquired knowledge is returned to the action environment where police officers should decide where and how to act. This way of working is a corner stone for police forces who want to actively pursue an intelligent led policing approach.

## 2 Domestic violence

The first project started in 2007 and aimed at developing new methods to automatically detect domestic violence cases within the police databases (Poelmans et al. 2010a). The technique Formal Concept Analysis (Wille 1982, Ganter et al 1999, Poelmans et al. 2010b, 2012b) which can be used to analyze data by means of concept lattices, is used to interactively elicit the underlying concepts of the domestic violence phenomenon (van Dijk 1997).

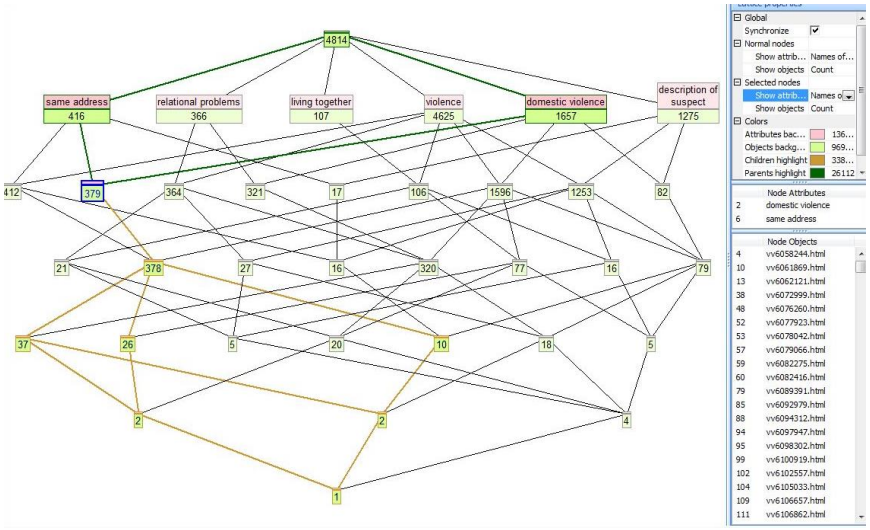



**Fig. 1.** Analyzing statements made by victims of a violent incident

The domestic violence definition which was employed by the Amsterdam-Amstelland police was as follows (Keus et al. 2000): "*Domestic violence can be characterized as serious acts of violence committed by someone in the domestic sphere of the victim. Violence includes all forms of physical assault. The domestic sphere includes all partners, ex-partners, family members, relatives and family friends of the victim. The notion of family friend includes persons that have a friendly relationship with the victim and (regularly) meet with the victim in his/her home.*" To identify domestic violence in police reports we make use of indicators which consist of words, phrases and / or logical formulas to compose compound attributes. The open source tool Lucene was initially used to index the unstructured textual reports using these attributes. The concept lattice visualization where reports are objects and indicators are attributes made it possible to iteratively identify valuable new knowledge. The lattice in Figure 1 contains 4814 police reports of which 1657 were labeled as domestic violence by police officers.

With CORDIET (see section 6 for details), the user can visually represent the underlying concepts in the data, gain insight in the complexity of the domain under investigation and zoom in on interesting concepts. For example we clicked on the node with 379 reports where suspect and victim lived on the same address and labeled as domestic violence by officers. Domain experts assumed that a situation where perpetrator and victim live at the same address is always a case of domestic violence, since these persons are probably family members, however this turned out not to be true. Analysis of the reports with attribute "same address" and not labeled as domestic violence revealed borderline cases such as violence in prisons, violence between a caretaker and inhabitant of an old folks home, etc. After multiple iterations of identi-

fying new concepts, composing new indicators and creating concept lattices we were able to refine the definition of domestic violence. Each of the cases were presented to the steering board of the domestic violence policy resulting in an improved definition of domestic violence and an improved handling of domestic violence cases. This investigation also resulted in a new automated case labelling system which is currently used to automatically label statements made by a victim to the police as domestic or non domestic violence (Poelmans et al. 2009, 2011a, Elzinga et al. 2009). At this moment the Amsterdam-Amstelland Police Department is using this system in combination with the national case triage system Trueblue.

## 3 Human trafficking

The next project focused on applying the knowledge exploration technique Formal Concept Analysis to detect (new) potential suspects and victims in suspicious activity reports and create a visual profile for each of them. The first application domain was human trafficking with a focus on sexual exploitation of the victims, a frequently occurring crime where the willingness of the victims to report is very low (Poelmans et al. 2011b, Hughes 2000).

After composing a set of early warning indicators and identifying potential suspects and victims, a detailed lattice profile of the suspect can be generated which shows the date of observation, the indicators observed and the contacts he or she had with other involved persons. In figure 2 the real names are replaced by arbitrary numbers and a number of indicators have been omitted for reasons of readability (the lattice was built using Concept Explorer). The persons (f = female and m = male) in the bottom of the figure are the most interesting potential suspects or victims because the lower a person appears in a lattice, the more indicators he or she has. For each of these persons a separate analysis can be made.

A selection of one of the men in the left bottom of figure 2 results in the concept lattice diagram in figure 3. In this figure the time stamps corresponding to each of the observations relevant for this person, together with the indicators and other persons mentioned are shown. The variant of formal concept analysis which makes use of temporal information is called temporal concept analysis (Wolff 2005). The lattice diagram shows that person D (4th left below) might be responsible for logistics, because he is driving in an expensive car ("dure auto"), and where the occupants show behavior of avoiding the police ("geen politie"). The man H (who appears in the extent of all concepts) is the possible pimp, who forced to work the possible victim woman S (1st upper right) in prostitution ("prostitutie" and "dwang"). Based on this diagram the corresponding reports can be collected and as soon as the investigators find sufficient indications a document based on section 273f of the Code of Criminal Law can be composed. This is a document that precedes any further criminal investigation against the man H.

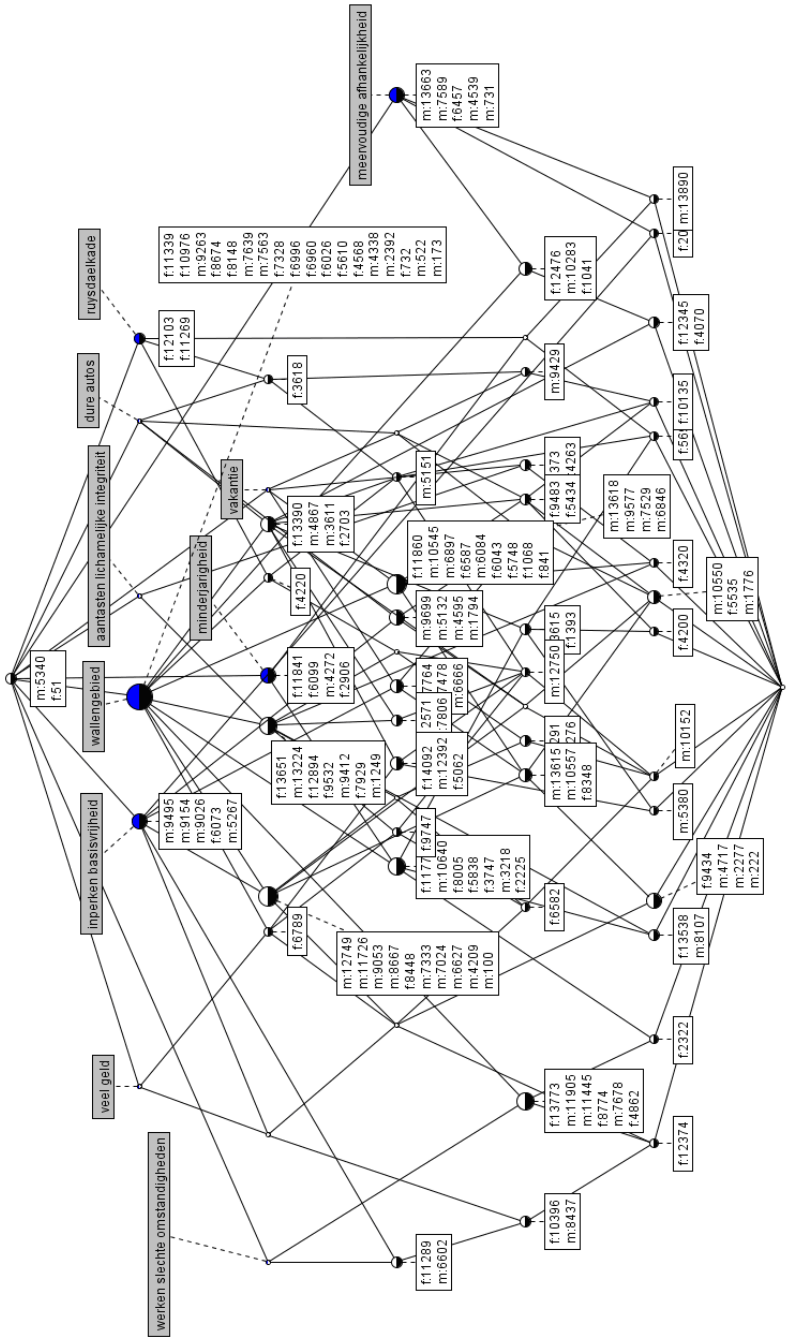



**Fig. 2.** Lattice of potential suspects and victims of human trafficking

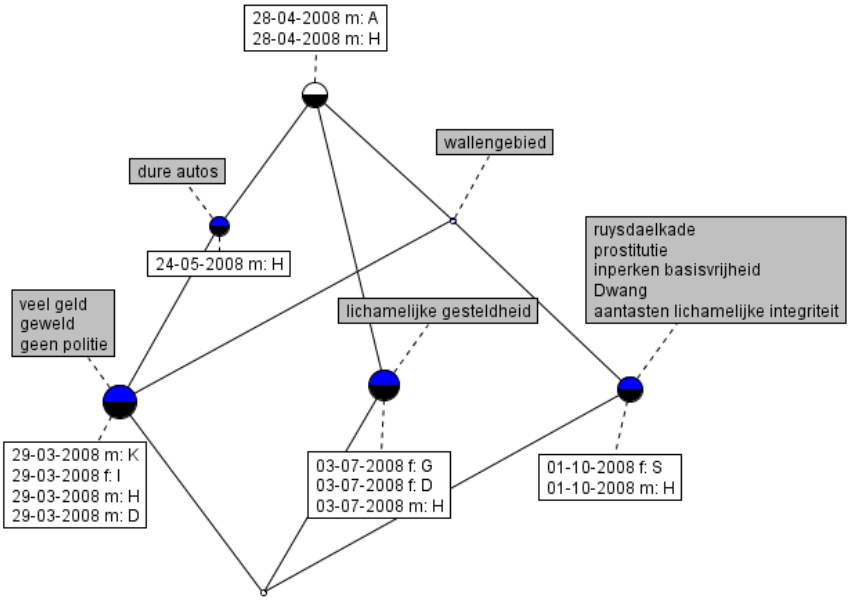



**Fig. 3.** Analysis of social network of suspect

## 4 Terrorism

During the third project we cooperated with the project team "Kennis in Modellen" (KiM, Knowledge in Models) from the National Police Service Agency in the Netherlands (KLPD). We combined formal concept analysis with the KiM model of Muslim radicalization to actively identifying potential terrorism suspects from suspicious activity reports (Elzinga et al. 2010, AIVD 2006). According to this model, a potential suspect goes through four stages of radicalization. The KiM project team has developed a set of 35 indicators based on interviews with experts on Muslim radicalism using which a person can be positioned in a certain phase. Together with the KLPD we intensively looked for characterizing words and combinations of words for each of these indicators. The difference with the previous models is that the KiM model added an extra dimension in terms of the number of different indicators which a person must have to be assigned to a radicalization phase.

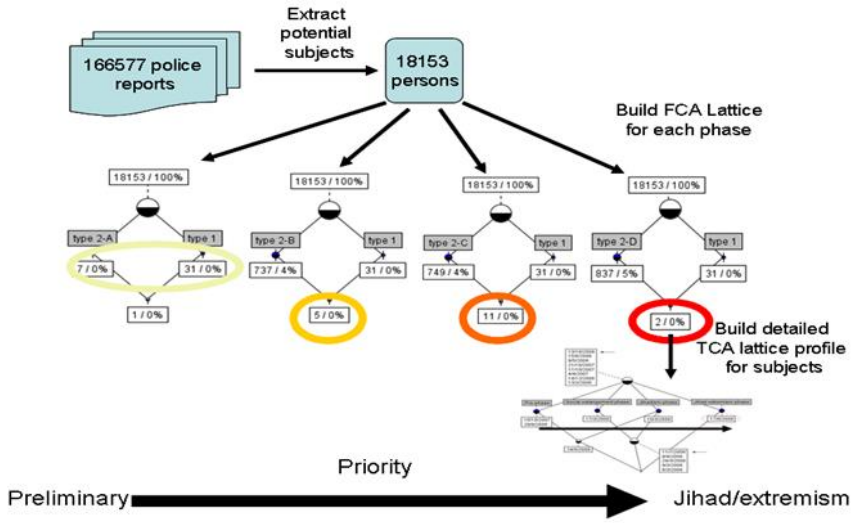



**Fig. 4.** The process model of extracting and profiling potential jihadists

The analysis was performed on the set of suspicious activity reports filed in the BVH database system of the Amsterdam-Amstelland Police Department during the years 2006, 2007 and 2008 resulting in 166,577 reports. From this set of observations 18,153 persons were extracted who meet at least one of the 35 indicators. From these 18.153 persons 38 persons were extracted who can be assigned to the 1st phase of radicalization, the preliminary phase ("voorfase"). Further analysis revealed that 19 were correctly identified, 3 of these persons were previously unknown by the Amsterdam-Amstelland Police Department, but known by the KLPD. From the 19 persons, 2 persons were found who met the minimal conditions of the jihad/extremism phase. For each of these persons a profile was made containing all indicators that were observed over time.

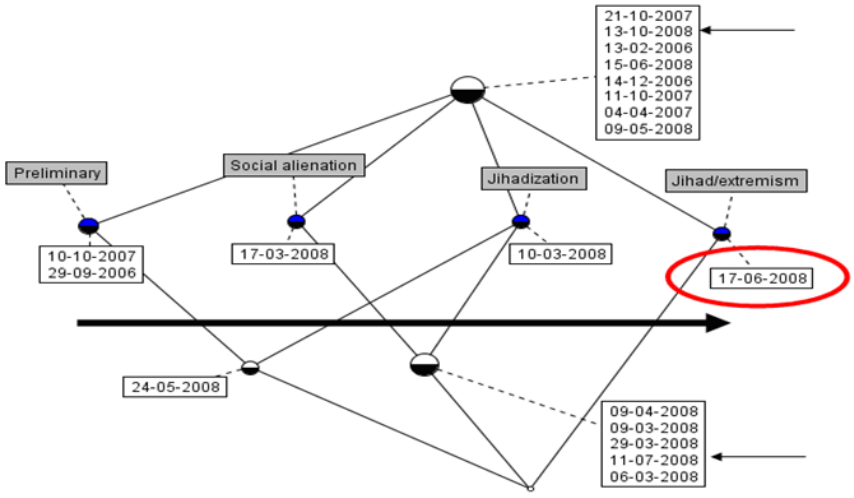



**Fig. 5.** Temporal lattice for subject C.

From the lattice diagram in figure 5 can be concluded that the person has reached the jihad/extremism phase on June 17, 2008 and has been observed by police officers two times afterwards (the arrows in the upper right and lower right of the figure) on July 11, 2008, and October 12, 2008.

## 5 Pedophile chat conversations

Chat conversations can be very long and time-consuming to read. A system which helps officers quickly identify those conversations posing a threat to a child's safety and understand what has been talked about may significantly speed up and improve the efficiency of their work (Elzinga et al. 2012).

Because original chat data collected by the Dutch police force organizations is restricted by law, results may not be made public. To demonstrate our FCA based method we use the chat data collected by a public American organization, Perverted Justice, which actively searches for pedophiles on the internet. We downloaded 533 chat files, i.e. one for each of the 533 different suspects. The victims in all chat files are adults playing the role of a young girl or boy in the age from 12 to 14. All these adults are members of the Perverted Justice organization and are trained to act as a youngster. The adults playing the victim try to lure the suspect by playing his or her role as good as possible. The behavior of the victims cannot be representative for young girls or boys, but the behavior of the suspects is realistic since they really believe to have contact with a young girl or boy and act in that way.

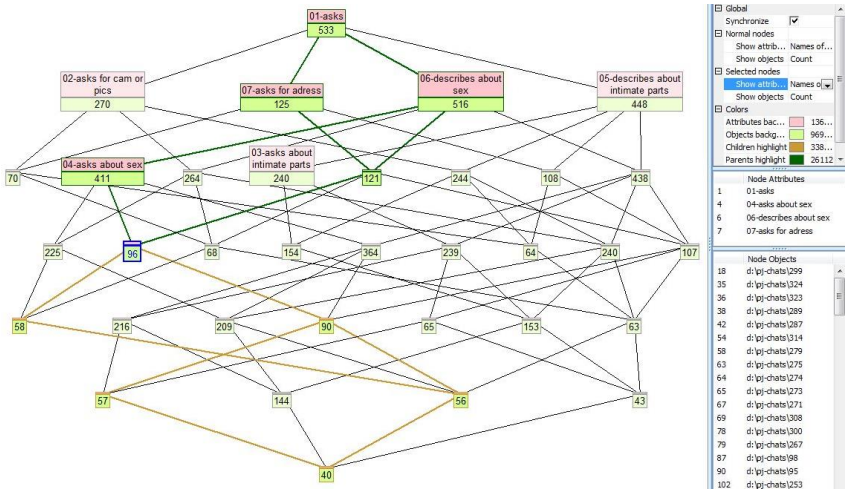



**Fig. 6.** Analyzing chat conversations of pedophiles with members of the perverted justice organization who pretend to be a young child

The lattice in Figure 8 shows how a set of 533 chat conversations was analyzed with FCA. We defined 7 term clusters containing keywords which were used by pedophiles in their chat conversations. We numbered these 7 attributes according to the severity of the threat to the child's safety. We clicked on a concept with 96 conversations in the extent and attributes "asks", "asks about sex", "describes about sex" and "asks for address". In the "node objects" pane the user can click on the name of a conversation to display its contents.

Figure 7 shows a transition diagram of the chat conversation 451 which was selected based on the line diagram shown in Figure 6. We explain Figure 7 intuitively in this paper, readers interested in the mathematical definitions are referred to Elzinga et al. (2012). Figure 7 is constructed by restricting the data table to the rows where chat log = 451, which are the 22 rows from 300 to 321. The chat time runs in these rows from 0 to 21. The many-valued attribute 'state' has in row 300, that is at time 0, the value '2' which means that the conversation 451 is in the state '2 Compliments'; in the next row 301, at time 1, the conversation 451 is in the state '5 Cam and pics'. This transition is graphically represented in Figure 7 by the arrow from the object concept of 300 to the object concept of 301. Clearly, the direction of the arrow is induced from the fact that time 0 is the predecessor of time 1 (in the natural ordering of integers).

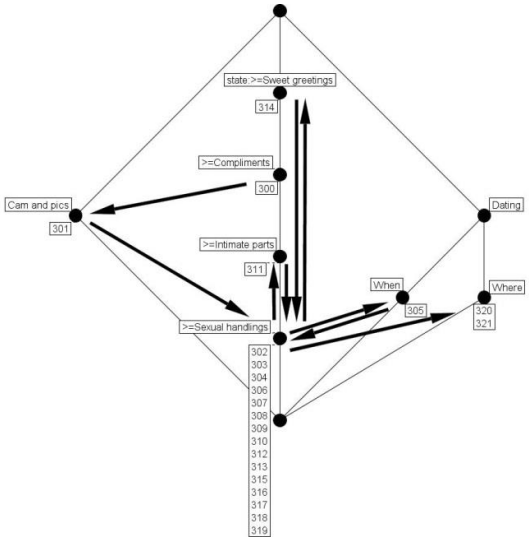



**Fig. 7.** A transition diagram for chat conversation 451

In Elzinga et al. (2012) we describe in detail how we selected chats from such a concept lattice and analyze them in detail with temporal relational semantic systems.

## 6 CORDIET

More and more companies have large amounts of unstructured data, often in textual form available. The few analytical tools that focus on this problem area offer insufficient functionality for the specific needs of many of these organizations. As part of the research work in the doctoral research of Jonas Poelmans the development of the data analysis suite Concept Relation Discovery and Innovation Enabling Technology (CORDIET, Elzinga 2011, Poelmans et al. 2012) was started in September 2010 in cooperation with the Moscow Higher School of Economics. Elzinga et al (2009) developed the first prototype where the strength of our approach with concept lattices and other visualization techniques such as Emergent Self Organizing Maps (ESOM) is demonstrated for the detection of individuals with radicalizing behavior. This toolset allows to carry out much faster and more detailed data analysis to distil relevant persons from police data.

## 7 Conclusions

The four projects which are carried out as part of the research chair show the potential of the knowledge exploration technique formal concept analysis. Especially the intuitively interpretable visual representation was found to be of great importance for in-

formation specialists within the police force on all levels, strategic, tactic and operational. This visualization did not only allow to explore the data interactively, but also to explore and define the underlying concepts of the investigation areas. New concepts, anomalies, confusing situations and faulty labeled cases were discovered, but also not previously known subjects were found who might be involved in human trafficking or terroristic activities. The temporal variant of formal concept analysis proved to be very useful for profiling suspects and their evolution over time. Never before unstructured information sources were retrieved in such a way that new insights, new suspects and victims became visible. That's why formal concept analysis will become an important instrument in the nearby future for information specialists within the police and will be an essential contribution to the formation of Intelligence within the Dutch police.

Among the future developents are applications of FCA-based biclustering (Ignatov et al. (2012)) and triclustering techniques (Ignatov et al. (2011)) in the Criminal Investagations domain.

## References


1. AIVD (2006), Violent jihad in the Netherlands, current trends in the Islamist terrorist threat. https://www.aivd.nl/aspx/download.aspx?file=/contents/pages/65582/jihad2006en.pdf
2. Elzinga, P., Poelmans, J., Viaene, S., Dedene, G. (2009), Detecting Domestic Violence, showcasing a knowledge browser based on Formal Concept Analysis and Emerging Self Organizing Maps. 11th International Conference on Enterprise Information Systems, Milan 6-10 may 2009.
3. Elzinga, P., Poelmans, J., Viaene, S., Dedene, G., Morsing, S. (2010) Terrorist threat assessment with Formal Concept Analysis. Proc. IEEE International Conference on Intelligence and Security Informatics. May 23-26, 2010 Vancouver, Canada, pp.77-82.
4. Elzinga, P. (2011) Formalizing the concepts of crimes and criminals. PhD dissertation University of Amsterdam.
5. Elzinga, P., Wolff, K.E., Poelmans, J., Viaene, S., Dedene, G. (2012) Analyzing chat conversations of pedosexuals with temporal relational semantic systems. F. Domenach et al. (eds.): Contributions to 10th International Conference on Formal Concept Analysis, Leuven, Belgium, 6 – 10 May 2012. ISBN 978-9-08-140995-7, 82-97.
6. Ganter, B., Wille, R. (1999), Formal Concept Analysis: Mathematical Foundations. Springer, Heidelberg.
7. Hatchuel, A., Weil, B., Le Masson, P (2004) Building innovation capabilities. The development of Design-Oriented Organizations: In Hage, J.T. (Ed), Innovation, Learning and Macro-institutional Change: Patterns of knowledge changes.
8. Hughes, D.M. (2000), 'The "Natasha" Trade: The Transnational Shadow Market of Trafficking in Women,' *Journal of International Affairs*, Spring, 53, no. 2.
9. Ignatov, D.I., Kuznetsov, S.O., Magizov, R.A., and Zhukov, L.E. (2011) From Triconcepts to Triclusters. In: Proc. of 13th International Conference on Rough Sets, Fuzzy Sets, Data Mining And Granular Computing, Kuznetsov et al. (Eds.): RSFDGrC 2011, LNCS/LNAI Volume 6743/2011, Springer-Verlag Berlin Heidelberg, 257-264

10. Ignatov, D.I., Kuznetsov, S.O., Poelmans, P. (2012) Concept-Based Biclustering for Internet Advertisement. ICDM Workshops 2012, 123-130
11. Keus, R., Kruijff, M.S. (2000) Huiselijk geweld, draaiboek voor de aanpak. Directie Preventie, Jeugd en Sanctiebeleid van de Nederlandse justitie.
12. Poelmans, J., Elzinga, P., Neznanov, A., Dedene, G., Viaene, S., Kuznetsov, S. (2012) Human-Centered Text Mining: a New Software System. P. Perner (Ed.): Lecture Notes in Artificial Intelligence 7377, 258–272, 12th Industrial Conference on Data Mining. Springer
13. Poelmans, J., Elzinga, P., Viaene, S., Dedene, G. (2009). A case of using formal concept analysis in combination with emergent self organizing maps for detecting domestic violence. In : Lecture Notes in Artificial Intelligence, Vol. 5633(XI), (Perner, P. (Eds.)). Industrial conference on data mining ICDM 2009. Leipzig (Germany), 20-22 July 2009 (pp. 402 p.).
14. Poelmans, J., Elzinga, P., Viaene, S., Dedene, G. (2010a) Curbing domestic violence: Instantiating C-K theory with Formal Concept Analysis and Emergent Self Organizing Maps. Intelligent Systems in Accounting, Finance and Management 17, (3-4) 167-191. Wiley and Sons, Ltd..
15. Poelmans, J., Elzinga, P., Viaene, S., Dedene, G. (2010b), Formal Concept Analysis in knowledge discovery: a survey. Lecture Notes in Computer Science, 6208, 139-153, 18th international conference on conceptual structures (ICCS 2010): from information to intelligence. 26 - 30 July, Kuching, Sarawak, Malaysia. Springer.
16. Poelmans, J., Elzinga, P., Viaene, S., Dedene, G. (2011a) Formally Analyzing the Concepts of Domestic Violence, Expert Systems with Applications 38, (4) 3116-3130. Elsevier Ltd.
17. Poelmans, J., Elzinga, P., Viaene, S., Dedene, G., Kuznetsov, S. (2011b) A concept discovery approach for fighting human trafficking and forced prostitution. Lecture Notes in Computer Science, 6828, 201-214, 19th International conference on conceptual structures, July 25-29, Derby, England. Springer.
18. Poelmans, J., Ignatov, I., Viaene, S., Dedene, G., Kuznetsov, S. (2012b) Text mining scientific papers: a survey on FCA-based information retrieval research. P. Perner (Ed.): Lecture Notes in Artificial Intelligence 7377, 273–287, 12th Industrial Conference on Data Mining, July 13-20, Berlin, Germany. Springer
19. T. van Dijk, Huiselijk geweld, aard, omvang en hulpverlening (Ministerie van Justitie, Dienst Preventie, Jeugd-bescherming en Reclassering, oktober 1997).
20. Wille, R. (1982), Restructuring lattice theory: an approach based on hierarchies of concepts, I. Rival (ed.). Ordered sets. Reidel, Dordrecht-Boston, 445-470.
21. Wolff, K.E. (2005) States, transitions and life tracks in Temporal Concept Analysis. In: B. Ganter et al. (Eds.): Formal Concept Analysis, LNAI 3626, pp. 127-148. Springer, Heidelberg.

# Classification Methods Based on Formal Concept Analysis

Olga Prokasheva, Alina Onishchenko, and Sergey Gurov

Faculty of Computational Mathematics and Cybernetics, Moscow State University

**Abstract.** *Formal Concept Analysis (FCA) provides mathematical models for many domains of computer science, such as classification, categorization, text mining, knowledge management, software development, bioinformatics, etc. These models are based on the mathematical properties of concept lattices. The complexity of generating a concept lattice puts a constraint to the applicability of software systems. In this paper we report on some attempts to evaluate simple FCA-based classification algorithms. We present an experimental study of several benchmark datasets using FCA-based approaches. We discuss difficulties we encountered and make some suggestions concerning concept-based classification algorithms.*

**Keywords:** Classification, pattern recognition, data mining, formal concept analysis, biclustering

## 1 Introduction

Supervised classification consists in building a classifier from a set of examples labeled by their classes or precedents (learning step) and then predicting the class of new examples by using the generated classifiers (classification step). Document classification is a sub-field of information retrieval. Documents may be classified according to their subjects or according to other attributes (such as document type, author, year of publication, etc.). Mostly, document classification algorithms are based on supervised classification. Algorithms of this kind can be used for text mining, automatical spam-filtering, language identification, genre classification, text mining. Some modern data mining methods can be naturally described in terms of lattices of closed sets, i.e., concept lattices [1], also called Galois lattices. An important feature of FCA-based classification methods do not make any assumptions regarding statistical models of a dataset. Biclustering [9, 10] is an approach related to FCA: it proposes models and methods alternative to classical clustering approaches, being based on object similarity expressed by common sets of attributes. There are several FCA-based models for data analysis and knowledge processing, including classification based on learning from positive and negative examples [1, 2].

In our previous work [15] the efficiency of a simple FCA-based binary classification algorithm was investigated. We tested this method on different problems

with numerical data and found some difficulties in its application. The main purpose of this paper is to investigate critical areas of the FCA method for better understanding of its features. Several advices for developers are also provided. We test hypothesis-based classification algorithm and our modified FCA-based method on 8 benchmarks. We describe our experiments and compare the performance of FCA-based algorithms with that of SVM-classification [16].

## 2 Definitions

**Formal Concept Analysis.** In what follows we keep to standard FCA definitions from [1]. Let $G$ and $M$ be sets, called set of objects and set of attributes, respectively. Let $I \subseteq G \times M$ be a binary relation. The triple $K = (G, M, I)$ is called a *formal context*. For arbitrary $A \subseteq G$ and $B \subseteq M$ the mapping $(.)'$ is defined as follows:

$$A' = \{m \in M \mid \forall g \in A(gIm)\};\ B' = \{g \in G \mid \forall m \in B(gIm)\}. \qquad (1)$$

This pair of mappings defines a Galois connection between the sets $2^G$ and $2^M$ partially ordered by the set-theoretic inclusion. Double application of the operation $(\cdot)'$ is a closure operator on the union of the sets $2^G$ and $2^M$. Let a context $K$ be given. A pair of subsets $(A, B)$, such that $A \subseteq G$, $B \subseteq M$, $A' = B$, and $B' = A$ is called a *formal concept* of $K$ with *formal extent* $A$ and *formal intent* $B$. The extent and the intent of a formal concept are closed sets.

**FCA in learning and classification.** Here we keep to definitions from [2] and [3]. Let $K = (G, M, I)$ be a context and $w \notin M$ a target attribute. In FCA terms, the input data for classification may be described by three contexts w.r.t. $w$: the positive context $K_+ = (G_+, M, I_+)$, the negative context $K_- = (G_-, M, I_-)$ , and the undefined context $K_\tau = (G_\tau, M, I_\tau)$ [2]. $G_-, G_+$ and $G_\tau$ 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. Operators $(\cdot)'$ in these contexts are denoted by $A^+, A^-, A^\tau$, respectively. For short we write $g', g'', g^+, g^-, g^\tau$ instead of $\{g\}', \{g\}'', \{g\}^+, \{g\}^-, \{g\}^\tau$, respectively. A formal concept of a positive context is called a *positive concept*. Negative and undefined concepts, as well as extents and intents of the contexts $K_-$ and $K_\tau$, are defined similarly. A positive formal intent $B_+$ of $(A_+, B_+) \in K_+$ is called *a positive or (+) — prehypothesis* if is not the formal intent of any negative concept, and it is called *a positive or (minimal) (+) — hypothesis* if it is not a subset of the intent $g^-$ for some elementary concept $(g, g^-)$ for a negative example $g$; otherwise it is called a false (+)-generalization.

Negative (or (-) —) prehypotheses, hypotheses, and false generalizations are defined similarly. The definitions imply that a hypothesis is also a prehypothesis. Hypotheses are used to classify undefined examples from the set $G^\tau$. If unclassified object $g^\tau$ contains a positive but no negative hypothesis, it is classified positively, similar for negative. No classification happens if the formal intent $g^\tau$ does not contain any subsets of either positive or negative hypotheses (insufficient data) or contains both a positive and a negative hypothesis (inconsistent data).

**Biclustering.** The particular case of biclustering [10–12] we have considered is a development of the FCA-based classification method. Using FCA methods, we can construct a hierarchical structure of biclusters that reflects the taxonomy of data. *Density* of bicluster $(A, B)$ of the formal context $K = (G, M, I)$ is defined as $\rho(A, B) = |I \cap \{A \times B\}|/(|A| \cdot |B|)$. Specify some value $\rho_{min} \in [0, 1]$. The bicluster $(A, B)$ is called dense if $\rho(A, B) \geq \rho_{min}$. *Stability index* $\sigma$ of a concept $(A, B)$ is given by $\sigma(A, B) = |\mathcal{C}(A, B)|/2^{|A|}$, where $\mathcal{C}(A, B)$ is the union of the sets $C \subseteq A$ such that $C = B'$ [13, 21]. Biclusters, as well as dense and stable formal concepts (i.e., concepts having stability above a fixed threshold), are used to generate hypotheses for clustering problems [13].

## 3 Basic Classification Algorithms

Several FCA-based classification methods are known [19, 15]: GRAND [31, 17], LEGAL [26], GALOIS [25], RULELEARNER [24], CIBLe [30], CLNN&CLNB [27], NAVIGALA[28], CITREC [29, 17] and classification method based on hypotheses [8, 7, 2, 3]. There are several categories of FCA-based classification methods:

1. **Hypothesis-based classification** using the general principle described in Section 2.
2. **Concept lattice based classification.** A concept lattice can be seen as a search-space in which one can easily pass from a level to another one. The navigation can e.g. start from the top concept with the least intent. Then one can progress concept by concept by taking new attributes and reducing the set of objects. Many systems use lattice-based classification, such as GRAND [31, 17], RULEARNER [24], GALOIS [25], NAVIGALA [28] and CITREC [29, 17]. The common constraint of these systems is the exponential algorithmic complexity of generating a lattice. For this reason, some systems search in a subset of the set of all concepts.
3. **Classification based on Galois sub-hierarchies.** Systems like CLNN& CLNB [27], LEGAL [26] and CIBLe [30] build Galois sub-hierarchy (ordered set of object and attribute concepts), which drastically reduces algorithmic complexity.
4. **Cover-based classification.** A concept cover is a part of the lattice containing only pertinent concepts. The construction of a cover concept is based on heuristic algorithms which reduce the complexity of learning. The concepts are extracted one by one. Each concept is given by a local optimization of a measure function that defines pertinent concepts. IPR (Induction of Product Rules) [32] was the first method generating a concept cover. Each pertinent concept induced by IPR is given by a local optimization of entropy function. The sets of pertinent generated concepts are sorted from the more pertinent to the less pertinent and each pertinent concept with the associated class gives a classification rule.

## 4 Classification Experiments with Benchmarks

### 4.1 A Hypothesis-Based Algorithm

The method for constructing concept-based hypotheses described above inspired the following binary classification algorithm [15]. The main steps of the algorithm are as follows:

1. **Data binarization.** The situation where the attributes are non-binary, but a classification method is designed for binary data brings up the problem of attribute binarization, or scaling. This problem is very difficult and a lot of papers are devoted to it. Scaling problem arises also when we use FCA for object classification. For specific tasks scaling is usually carried out empirically by repeatedly solving the problem of classification on precedents. It is clear, however, that in a couple of "scaling–recognition method" the determining factor is exactly scaling. Indeed, in the event of its successful application a 'good' transformation of the feature space will be obtained and almost any recognition algorithm will show good results in that space. So that the problem of scaling is a nonspecific for FCA-recognition methods and the current level of development of these methods unable to point the best technique of scaling focused on their use. That is why our work is not focused on this problem and we use a simple scaling, which, we believe, allowed more clearly to identify the features of FCA-classification methods. Hence, we just normalized all attributes to [0,1] interval and than applied interval-based nominal scaling. The number of intervals is fixed and equals 10. The size of intervals is also fixed and equals 0.1.
2. **Hypothesis generation and classification.** Algorithm searches common attributes for all objects from the first class (second class), which are not observed for any objects from the second class (first class). Obtained sets of attributes (hypothesis) are used to classify undefined objects.

The algorithm has been tested on numerical benchmarks. The data for the first four problems is taken from the UCI Machine Learning Repository[1]. Problem 5 (Two Norm) involving the separation of two normal 20-dimensional distributions is taken from the University of Toronto site[2]; the CART classification algorithm [22] produced for this problem an error rate of 22.1% with a training sample of 300 precedents, which is almost a factor of 10 higher than the theoretical minimum for the ideal classifier — the Fisher discriminant function. Problems 6 (Lung Cancer), 7 (Cirrhosis), and 8 (Cloud Seeding) are taken from the StatLib site[3]. The considered problems come from different specific research areas. For example, in the Liver Disorders problem, the objects are datasets obtained from the tests of six patients. The training sample consists of 345 precedents divided into positive and negative classes with respect to the

[1] http://archive.ics.uci.edu/ml
[2] http://www.cs.toronto.edu/~delve/data/twonorm/desc.html
[3] http://lib.stat.cmu.edu/datasets, pages /veteran, /pbc, and /cloud, respectively

target attribute "presence of liver disorder". The experiment results obtained for various problems by using leave-one-out cross-validation are presented below (Table 1). In the table heading, $n$ is the number of attributes, $l$ is the number of objects (the size of the training sample),$err$ is the classification error rate and $l_c$ is the number of classified objects ( $l - l_c$ = the number of failed classifications).

| Problem | n | l | $l_c$ | err |
|---|---|---|---|---|
| 1. Liver Disorders | 6 | 345 | 20 | 15.00% |
| 2. Glass identification | 9 | 146 | 25 | 20.00% |
| 3. Wine | 13 | 130 | 47 | 08.50% |
| 4. Wine quality | 11 | 310 | 51 | 09.80% |
| 5. Two norm | 20 | 354 | 109 | 07.30% |
| 6. Lung cancer | 8 | 137 | 9 | 11.10% |
| 7. Cirrhosis | 19 | 276 | 29 | 34.48% |
| 8. Cloud-seeding | 5 | 108 | 6 | 50.00% |

**Table 1.** Experimental results

The algorithm was updated with the following modifications in definitions of hypothesis and classification:

1. Hypothesis modification: attributes observed for "almost" all objects of the particular class were added to the hypothesis. It was ensured that the ratio of objects which did not comply with the hypothesis in the same class did not exceed the value $P$ (a new algorithm parameter) and obviously there was no guarantee that the hypothesis is not contained in descriptions of objects of the opposite class.
2. Introduction of an inter-object metric and modification of the classification procedure: the "distance" between objects increases as they reveal difference in a larger number of coordinates. We compute the distance of the object being classified by positive and negative hypotheses and normalize it by the number of attributes (or 1s in binary represenation) in each hypothesis. The object is classified to the nearest class in contexts of the metric defined above.
3. Attribute weighting: an attribute is assigned a weight which increases with the number of 1s in the corresponding column.

The modified algorithm was applied to the considered problems. The experimental results for $P = 0.2$ are given in Table 2.

### 4.2 Classification Using Biclustering

Biclustering can be used for classification upon data scaling. For this purpose we select informative objects which are included in biclusters with density greater than threshold $\rho_{min}$. Hypotheses are generated using these objects. This approach avoids noise effects during learning step [15]. The difference between

| Problem | n | l | $l_c$ | err |
|---|---|---|---|---|
| 1. Liver Disorders | 6 | 345 | 79 | 39.2% |
| 2. Glass identification | 9 | 146 | 64 | 25.00% |
| 3. Wine | 13 | 130 | 87 | 14.9% |
| 4. Wine quality | 11 | 310 | 142 | 17.60% |
| 5. Two norm | 20 | 354 | 224 | 15.10% |
| 6. Lung cancer | 8 | 137 | 36 | 36.10% |
| 7. Cirrhosis | 19 | 276 | 136 | 33.30% |
| 8. Cloud-seeding | 5 | 108 | 37 | 43.20% |

**Table 2.** Experimental results for the modified algorithm

proposed algorithm and simple FCA algorithm resides only in the second step: hypotheses are now generated using only informative objects selected by biclustering. The method has two adjustable parameters: the bicluster density $\rho_{min}$ and the ratio $P$ of objects which do not satisfy classical hypotheses. The parameter $\rho_{min}$ affects the generation of hypotheses. If its value is too small, hypothesis generation is tainted by noisy attributes and outliers. If its value is too large, the hypothesis will have to meet excessively stringent requirements. It may be efficient to use a range of values for $\rho_{min}$ and thus focus on the main objects, skipping the marginal ones. This method has been tested, but it failed to produce a significant improvement of the classification performance, which will be later explained by the specific features of the particular problem.

The parameter $P$ affects the ratio of objects which do not satisfy the hypotheses of the same class. When the parameter $P$ is close to zero, hypotheses are generated in accordance with the classical definitions: they include only the attributes that are observed for all the objects of the given class. The difficulty is that hypotheses may become "non-representative" for the given class. If the parameter $P$ is taken too large, the hypotheses will require that the control object has a large number of attributes, which again may impose an excessively stringent requirement on hypotheses. In a certain sense, this is the well-known overfitting effect often observed in pattern recognition.

The experimental results assessed by leave-one-out cross-validation are presented in Table 3 and Table 4. In the table heading, $n$ is the number of attributes, $l$ is the number of objects (the size of the training sample). The columns present the solution results obtained with the algorithm parameters (the threshold $\rho$ and the proportion $P$ ) optimized by two criteria: the classification error rate $err$ (Table 3) and the number of classified objects $l_c$ ( $l - l_c$ = the number of failed classifications) (Table 4). The local optimization of the algorithm parameters was carried out by the GaussSeidel method, their optimal values $\rho_{min}$ and $P^*$ are shown together with $err$.

According to the experimental results, the lower is the error rate, the smaller is the number of classified objects. We can construct an algorithm with zero error rate, but the ratio of classified objects will be also small. We apply such an algorithm for all considered objects. The results are shown in Table 5, where

| Problem | n | l | $l_c$ | err | $\rho_{min}$ | $P^*$ |
|---|---|---|---|---|---|---|
| 1. Liver Disorders | 6 | 345 | 22 | 13.6% | 0.30 | 0.01 |
| 2. Glass identification | 9 | 146 | 28 | 10.00% | 0.15 | 0.05 |
| 3. Wine | 13 | 130 | 76 | 02.00% | 0.25 | 0.05 |
| 4. Wine quality | 11 | 310 | 83 | 08.40% | 0.25 | 0.05 |
| 5. Two norm | 20 | 354 | 206 | 12.10% | 0.15 | 0.15 |
| 6. Lung cancer | 8 | 137 | 18 | 05.50% | 0.01 | 0.01 |
| 7. Cirrhosis | 19 | 276 | 33 | 21.00% | 0.05 | 0.05 |
| 8. Cloud-seeding | 5 | 108 | 7 | 28.00% | 0.15 | 0.05 |

**Table 3.** Experimental results. Classification error rate is optimized.

| Problem | n | l | $l_c$ | err | $\rho_{min}$ | $P^*$ |
|---|---|---|---|---|---|---|
| 1. Liver Disorders | 6 | 345 | 79 | 29.1% | 0.30 | 0.20 |
| 2. Glass identification | 9 | 146 | 59 | 16.90% | 0.30 | 0.20 |
| 3. Wine | 13 | 130 | 85 | 08.20% | 0.30 | 0.20 |
| 4. Wine quality | 11 | 310 | 141 | 13.50% | 0.30 | 0.20 |
| 5. Two norm | 20 | 354 | 233 | 15.20% | 0.30 | 0.20 |
| 6. Lung cancer | 8 | 137 | 98 | 25.50% | 0.05 | 0.05 |
| 7. Cirrhosis | 19 | 276 | 83 | 37.79% | 0.30 | 0.20 |
| 8. Cloud-seeding | 5 | 108 | 20 | 30.00% | 0.15 | 0.15 |

**Table 4.** Experimental results. The number of classified objects is optimized.

$l_c$ is the number of classified objects (the ratio is in brackets). The efficiency of FCA-based algorithm was compared with that of classical SVM-algorithm [16]. Each dataset was divided into training sample (80% of objects) and test sample (20% of objects). Table 5 $SVMerr$ shows the error rate of the SVM-algorithm, $SVMerr\ on\ l_c$ is the error rate on objects which were classified by the rigorous FCA-based method. Zero error rate was attained with classical hypotheses

| Problem | $l$ | $l_c$ | $\rho_{min}$ | $P^*$ | $SVMerr$ | $SVMerr\ on\ l_c$ |
|---|---|---|---|---|---|---|
| 1. Liver Disorders | 345 | 18 (5.2%) | 0.15 | 0 | 34.78% | 22.2% |
| 2. Glass identification | 146 | 22 (15%) | 0.15 | 0 | 31.03% | 4.55% |
| 3. Wine | 130 | 45 (35%) | 0.25 | 0 | 7.69% | 2.22% |
| 4. Wine quality | 130 | 49 (5.8%) | 0.1 | 0 | 35.48% | 6.12% |
| 5. Two norm | 354 | 103 (29%) | 0.03 | 0 | 3.85% | 0% |
| 6. Lung cancer | 137 | 9 (6.50%) | 0.01 | 0 | 40.74% | 0% |
| 7. Cirrhosis | 276 | 24 (9.00%) | 0.05 | 0 | 9.8% | 12.5% |
| 8. Cloud-seeding | 108 | 5 (4.6%) | 0.15 | 0 | 40.91% | 25.00% |

**Table 5.** Experimental results with zero error rate.

($P$=0) from objects with low density ($\rho_{min} \leq 0.25$). This rigorous algorithm can be applied to problems with high error costs. It is more likely to refuse classification than make wrong decisions.

## 5 Conclusions

FCA provides a convenient tool for formalizing symbolic machine learning and classification models. We studied hypothesis-based classification in different areas without special modifications for each dataset, using a simple binarization (scaling) of numerical data. Our results suggest the following conclusions:

1. Application of biclustering with parameter optimization made a very slight improvement in the quality of classification compared to the updated FCA algorithm (only by 3% in problem 1).
2. In all cases there was an unacceptably high rate of classification failures.
3. In all cases there was an unacceptably high error rate.
4. Attempts to fine-tune the algorithm parameters with the objective of reducing the failure rate were generally accompanied by increasing in the number of errors, although in some cases (problem 8) the error rate increased only slightly; the number of classifiable objects in these cases increased substantially (problem 6).
5. The classical FCA-based algorithm can produce accurate classification, but it refuses to classify the majority of test sample.

The analysis of the hypotheses generated with various parameter values and different optimization criteria has shown that hypotheses of different classes are often included in one another. We can naturally assume that if the classes show less tendency to diffuse into one another, biclustering and the classical FCA method would produce more impressive results. The relative location of classes is improved in pattern recognition theory by methods that involve transformation of the attribute space. In these cases, data compactification methods may be effectively applied [14]. We can reasonably assume that another scaling algorithms with floating-size intervals and interval-length optimization may improve classification results compared to those we have obtained with the simplest scaling. Our analysis of FCA-based classification provides the following conclusions:

1. For the chosen universal scaling procedure the classification results are far from being optimal. Individual scaling for each problem may improve classification quality.
2. FCA-based classification methods without modification and/or thorough preprocessing of data are usable only for preliminary classification.
3. A well-known idea for the modification of the direct FCA approach is to develop hypothesis generation methods. It is useful to allow for the specific features of the particular subject area and to fine-tune hypotheses and the algorithms by using e.g. parameters $\rho_{min}$ , P*, $\sigma_{min}$.
4. It is also possible to develop and apply sharper classification rules, e.g. by weighting objects, attributes, hypotheses, etc.
5. A promising approach is to use FCA-based methods to transform the attribute space, in particular using data compactness estimates.
6. concept-based methods are appropriate for classification problems with high error costs, e.g. in medical, security and military applications.

An important step in many data classification problems is the selection of a suitable similarity measure. We decided to investigate how different similarity metric described in [20, 23] affects the quality of hypothesis-based classification method. The majority of pattern recognition methods use the metric information about objects: methods based on distances, potential functions, dividing surface, the algebraic approach, etc. In these methods the amount of information about classes either is fairly used at all. The strength of FCA-based classification problems is in the identification and use of these particular data, but the metric information about the feature space is lost. Thus, in pattern recognition the classic discriminant methods and method based on FCA are at opposite poles w.r.t. metrics. In the FCA the metric information appears in a weak form as the result of scaling, which accounts for "distances" between attributes. It seems that the success in the development of FCA-based recognition methods will be related to the introduction of information about metric properties of feature spaces. Our future work will be focused on developing and applying sharper classification rules with modifications of similarity measure.

## References

1. B. Ganter, R. Wille: Formal Concept Analysis: Mathematical Foundations. Springer, Berlin/Heidelberg (1999).
2. S.O. Kuznetsov: Mathematical aspects of concept analysis. In: Journal of Mathematical Science, Vol. 80, Issue 2, pp. 1654–1698, (1996).
3. S.O. Kuznetsov: Complexity of Learning in Concept Lattices from Positive and Negative Examples. Discrete Applied Mathematics, 2004, No. 142(1–3), pp. 111-125.
4. G. Birkhoff: Lattice Theory. AMS, Providence, 3rd edition (1967).
5. O. Ore: Theory of Graphs. American Mathematical Society, Providence (1962).
6. S.I. Gurov: Ordered Sets and Universal Algebra [in Russian], MGU, Moscow (2004).
7. V.K. Finn: The Synthesis of Cognitive Procedures and the Problem of Induction. Autom. Doc. Math. Linguist., 43, 149-195 (2009).
8. V.K. Finn: On machine-oriented formalization of plausible reasoning in the style of F. Bacon and D.S. Mill [in Russian], Semiotika i Informatika, 20, 35–101 (1983).
9. S.C. Madeira and A.L. Oliveira: Biclustering Algorithms for Biological Data Analysis: A Survey. IEEE/ACM Trans. Comput. Biol. Bioinformatics 1, 24–45 (2004).
10. B.G. Mirkin: Mathematical Classification and Clustering. Kluwer (1996).
11. D.I. Ignatov, S.O. Kuznetsov: Frequent Itemset Mining for Clustering Near Duplicate Web Documents. In: Proc. 17th Int. Conf. on Conceptual Structures (ICCS 2009), LNAI (Springer), Vol. 5662, 185–200, 2009.
12. D.I. Ignatov, S.O. Kuznetsov, R.A. Magizov and L.E. Zhukov: From Triconcepts to Triclusters. In: Proc. 13th Int. Conf. on Rough Sets, Fuzzy Sets, Data Mining, and Granular Computing (RSFDGrC 2011), LNCS (Springer), Vol. 6743, 257–264, 2011.
13. S.O. Kuznetsov: Stability as an Estimate of the Degree of Substantiation of Hypotheses on the Basis of Operational Similarity. In: Nauchno-Tekhnicheskaya Informatsiya, Seriya 2, Vol. 24, No. 12, pp. 21–29, 1990.
14. S.I. Gurov, N.S. Dolotova, I.N. Fatkhutdinov: Noncompact recognition problems. Circuit design according to E. Gilbert. In: Spectral and Evolution Problems: Proc.

17 th Crimean Autumn Mathematical School-Symposium, Simferopol, Crimean Scientific Center of Ukrainian Academy of Sciences, 17, pp. 37–44 (2007).
15. A.A. Onishchenko, S.I. Gurov: Classification based on Formal Concept Analysis and Biclustering: possibilities of the approach. In: Computational Mathematics and Modeling, Vol. 23, No. 3, July, pp. 329–336 (2012).
16. C. Cortes, V. Vapnik: Support–vector networks. In: Machine Learning, September 1995, Volume 20, Issue 3, pp. 273–297.
17. N. Meddouri, M. Meddouri: Classification Methods based on Formal Concept Analysis, — CLA 2008 (Posters), pp. 9–16, Palacky University, Olomouc, (2008).
18. S. I. Gurov: Boolean Algebras, Ordered Sets, Lattices: Definitions, Properties, Examples [in Russian], KRASAND, Moscow (2012).
19. S.O. Kuznetsov: Machine learning on the basis of formal concept analysis Automation and Remote Control 62 (10),pp. 1543–1564.
20. F. Alqadah, R. Bhatnagar: Similarity measures in formal concept analysis, Annals of Mathematics and Artificial Intelligence, 61(3), 245–256 (2011).
21. S.O. Kuznetsov: On Stability of a Formal Concept. Annals of Mathematics and Artificial Intelligence, Vol. 49, pp.101–115, 2007.
22. L. Breiman, J.H. Friedman, R.A. Olsen, and C.J. Stone: Classification and Regression Trees, Wadsworth Int. Group, Belmont, CA (1984).
23. F. Dau, J. Ducrou, P.W. Eklund: Concept Similarity and Related Categories in SearchSleuth. ICCS 2008: pp. 255–268
24. M. Sahami: Learning classification Rules Using Lattices. N. Lavrac and S. Wrobel eds., pp. 343–346, Proc ECML, Heraclion, Crete, Greece (Avril 1995).
25. C. Caprineto, G. Romano: GALOIS An order-theoretic approach to conceptual clustering. In proceedings of ICML93, pp. 33–40, Amherst, USA (July 1993).
26. P. Njiwoua, E.M. Nguifo: Forwarding the choice of bias LEGAL-F Using Feature Selection to Reduce the complexity of LEGAL. In Proceedings of BENELEARN-97, ILK and INFOLAB, Tilburg University, the Netherlands, pp. 89–98, 1997.
27. Zhipeng Xie, Wynne Hsu, Zongtian Liu, Mong Li Lee: Concept Lattice based Composite Classifiers for high Predictability. Artificial Intelligence, vol. 139, pp. 253–267, Wollongong, Australia (2002).
28. S. Guillas, K. Bertet, J-M. Ogier: Extension of Bordats algorithm for attributes. Concept Lattices and Their Applications: CLA07, Montpellier, France (2007).
29. B. Douar, C. Latiri, Y. Slimani: Approche hybride de classification supervisee base de treillis de Galois: application la reconnaissance de visages. In: Extraction et Gestion des Connaissances (EGC08), 309–320, Nice, France (2008).
30. P. Njiwoua, Mephu Nguifo E.: Ameliorer lapprentissage partir dinstances grace linduction de concepts: le systeme CIBLe, Revue dIntelligence Artificielle (RIA), vol. 13, 2, 1999, pp. 413–440, Hermes Science.
31. E.M. Nguifo, P. Njiwoua: Treillis de concepts et classification supervise. Technique et Science Informatiques: TSI, Volume 24, Issue 4, pp. 449–488 (2005).
32. M. Maddouri: Towards a machine learning approach based on incremental concept formation. Intelligent Data Analysis, Volume 8, Issue 3, pp. 267–280 (2004).

# Debugging Programs using Formal Concept Analysis

Artem Revenko[1,2]

[1] Technische Universität Dresden
Zellescher Weg 12-14, 01069 Dresden, Germany
[2] National Research University Higher School of Economics
Pokrovskiy bd. 11, 109028 Moscow, Russia
artem_viktorovich.revenko@mailbox.tu-dresden.de

**Abstract.** The classification of possible errors in object intents is given and some possibilities of exploring them are discussed. Two techniques for finding some types of errors in new object intents are introduced. After comparing the better technique is developed further in order to guarantee the absence of certain errors given enough information. Based on this technique an approach for debugging source code is presented and discussed. It is shown that the new approach yields bug hypothesis in a strict logical form. Using the new approach it is possible to come closer to debugging programs on a logical level not checking executions line by line. An example of applying the new approach is presented.

**Keywords:** formal context analysis, implication, debugging

## 1 Introduction

In this work we present a new approach to debug programs. This work was inspired by the Delta Debugger project [13] where authors discuss the possibilities of automatic debugging, namely isolation of failure-inducing inputs. However, when it comes to finding actual causes of the failure it is still not possible to automatically explain the failure logically. In some cases the nearest neighbour technique yields good results, but usually near-probabilistic criteria like coverage or chi-square are used [2]. In this work we use recent advance in Formal Concept Analysis in an attempt to find logical dependencies between fails and successful runs of a program. Several studies were performed to discover the possibilities of using Formal Concept Analysis in software development. For example, in [11] and [6] authors use Formal Concept Analysis for building class hierarchies. In [8] FCA is used to determine dependencies on program trace. Authors reveal causal dependencies and even are able to find "likely invariants" of program in special cases. However, they do not consider the possibility of debugging. However, to our best knowledge there are no works about applying Formal Concept Analysis to program debugging.

In this paper we first introduce a new technique for finding errors in new object intents. This technique was first introduced in our previous work; we partly

repeat the results and refer to our previous work for more details. In this paper we recall two different approaches for revealing errors in new object intents: one based on computing the implication system of the context and another one based on computing the closures of the subsets of the new object intent. Since computing closures may be performed much faster we improve and generalize this approach and finally obtain a procedure for finding all possible errors of the considered types. We also provide experimental results to compare two approaches. After that we present a new approach of debugging based on the discussed above technique of finding errors in data. An example of debugging is provided.

All sets and contexts we consider in this paper are assumed to be finite.

## 2 Main Definitions

Let $G$ and $M$ be sets. Let $I \subseteq G \times M$ be a binary relation between $G$ and $M$. Triple $\mathbb{K} := (G, M, I)$ is called a *(formal) context*.
The set $G$ is called a set of *objects*. The set $M$ is called a set of *attributes*.

Consider mappings $\varphi\colon 2^G \to 2^M$ and $\psi\colon 2^M \to 2^G$: $\varphi(X) := \{m \in M \mid gIm \text{ for all } g \in X\}$, $\psi(A) := \{g \in G \mid gIm \text{ for all } m \in A\}$. For any $X_1, X_2 \subseteq G$, $A_1, A_2 \subseteq M$ one has

1. $X_1 \subseteq X_2 \Rightarrow \varphi(X_2) \subseteq \varphi(X_1)$
2. $A_1 \subseteq A_2 \Rightarrow \psi(A_2) \subseteq \psi(A_1)$
3. $X_1 \subseteq \psi\varphi(X_1)$ and $A_1 \subseteq \varphi\psi(A_1)$

Mappings $\varphi$ and $\psi$ define a *Galois connection* between $(2^G, \subseteq)$ and $(2^M, \subseteq)$, i.e. $\varphi(X) \subseteq A \Leftrightarrow \psi(A) \subseteq X$. Usually, instead of $\varphi$ and $\psi$ a single notation $(\cdot)'$ is used. $(\cdot)'$ is sometimes called a *derivation operator*. For $X \subseteq G$ the set $X'$ is called the *intent* of $X$ and is denoted $\mathrm{int}(X)$. Similarly, for $A \subseteq M$ the set $A'$ is called the *extent* of $A$ and is denoted $\mathrm{ext}(A)$.
Let $Z \subseteq M$ or $Z \subseteq G$. $(Z)''$ is called the *closure* of $Z$ in $\mathbb{K}$. Applying Properties 1 and 2 consequently one gets the *monotonicity* property: for any $Z_1, Z_2 \subseteq G$ or $Z_1, Z_2 \subseteq M$ one has $Z_1 \subseteq Z_2 \Rightarrow Z_1'' \subseteq Z_2''$.
Let $m \in M, X \subseteq G$, then $\overline{m}$ is called a *negated* attribute. $\overline{m} \in X'$ whenever no $x \in X$ satisfies $xIm$. Let $A \subseteq M$; $\overline{A} \subseteq X'$ iff all $m \in A$ satisfy $\overline{m} \in X'$.

An *implication* of $\mathbb{K} := (G, M, I)$ is defined as a pair $(A, B)$, written $A \to B$, where $A, B \subseteq M$. $A$ is called the *premise*, $B$ is called the *conclusion* of the implication $A \to B$. The implication $A \to B$ is *respected by a set of attributes* $N$ if $A \nsubseteq N$ or $B \subseteq N$. The implication $A \to B$ holds (is valid) in $\mathbb{K}$ if it is respected by all $g'$, $g \in G$, i.e. every object, that has all the attributes from $A$, also has all the attributes from $B$. Implications satisfy *Armstrong rules*:

$$\frac{}{A \to A} \quad , \quad \frac{A \to B}{A \cup C \to B} \quad , \quad \frac{A \to B, B \cup C \to D}{A \cup C \to D}$$

A *support* of an implication in context $\mathbb{K}$ is the set of all objects of $\mathbb{K}$, whose intents contain the premise and the conclusion of the implication. A *unit implications* is defined as an implication with only one attribute in the conclusion,

i.e. $A \to b$, where $A \subseteq M,\ b \in M$. Every implication $A \to B$ can be regarded as the set of unit implications $\{A \to b \mid b \in B\}$. One can always observe only unit implications without loss of generality.

An *implication basis* of a context $\mathbb{K}$ is defined as a set $\mathfrak{L}$ of implications of $\mathbb{K}$, from which any valid implication for $\mathbb{K}$ can be deduced by the Armstrong rules and none of the proper subsets of $\mathfrak{L}$ has this property.
A minimal implication basis is an implication basis minimal in the number of implications. A minimal implication basis was defined in [7] and is known as the *canonical implication basis*. In paper [4] the premises of implications from the canonical base were characterized in terms of pseudo-intents. A subset of attributes $P \subseteq M$ is called a *pseudo-intent*, if $P \neq P''$ and for every pseudo-intent $Q$ such that $Q \subset P$, one has $Q'' \subset P$. The canonical implication basis looks as follows: $\{P \to (P'' \setminus P) \mid P$ - pseudo-intent$\}$.

We say that an object $g$ is *reducible* in a context $\mathbb{K} := (G, M, I)$ iff $\exists X \subseteq G : g' = \bigcap_{j \in X} j'$.

## 3 Types of Errors

In this section we use the idea of *data domain dependency*. Usually objects and attributes of a context represent entities. Dependencies may hold on attributes of such entities. However, such dependencies may not be implications of a context as a result of an error in object intents. Thereby, data domain dependencies are such rules that hold on data represented by objects in a context, but may erroneously be not valid implications of a context.

In this work we consider only dependencies that do not have negations of attributes in premises. As mentioned above there is no need to specially observe non-unit implications. Consider possible types of such dependencies ($A \subseteq M,\ b, c \in M$):

1. $A \to b$
2. $A \to \overline{b}$
3. $A \to b \vee c$
4. $A \to \Phi$, where $\Phi$ is any logical formula not considered above, for example, $\Phi = a \vee (b \wedge \overline{c})$

The types 1 and 2 are most simple and common dependencies. In this work we try to find the algorithm to reveal these two types of dependencies and find corresponding errors.

## 4 Finding Errors

Below we assume that we are given a context (possibly empty) with correct data and a number of new object intents that may contain errors. This data is taken from some data domain and we may ask an expert whose answers are always

correct. However, we should ask as few questions as possible.

We introduce two different approaches to finding errors. The first one is based on inspecting the canonical basis of a context. When adding a new object to the context one may find all implications from the canonical basis of the context such that the implications are not respected by the intent of the new object. These implications are then output as questions to an expert in form of unit implications. If at least one of these implications is accepted, the object intent is erroneous. Since the canonical basis is the most compact (in the number of implications) representation of all valid implications of a context, it is guaranteed that the minimal number of questions is asked and no valid dependencies of Type 1 are left out.
Although this approach allows one to reveal all dependencies of Type 1, there are several issues. The problem of producing the canonical basis with known algorithms is intractable. Recent theoretical results suggest that the canonical base can hardly be computed with better worst-case complexity than that of the existing approaches ([3], [1]). One can use other bases (for example, see progress in computing proper premises [10]), but the algorithms known so far are still too costly and non-minimal bases do not guarantee that the expert is asked the minimal sufficient number of questions.

However, since we are only interested in implications corresponding to an object, it may be not necessary to compute a whole implication basis. Here is the second approach. Let $A \subseteq M$ be the intent of the new object not yet added to the context. $m \in A''$ iff $\forall g \in G : A \subseteq g' \Rightarrow m \in g'$, in other words, $A''$ contains the attributes common to all object intents containing $A$. The set of unit implications $\{A \to b \mid b \in A'' \setminus A\}$ can then be shown to the expert. If all implications are rejected, no attributes are forgotten in the new object intent. Otherwise, the object is erroneous. This approach allows one to find errors of Type 1.

## 5 Improvements

Obviously, applying the derivation operator two times is a much easier task than computing the canonical basis, and can be performed in polynomial time. However, the following case is possible. Let $A \subseteq M$ be the intent of the new object such that $\nexists g \in G : A \subseteq g'$. In this case $A'' = M$ and the implication $A \to A'' \setminus A$ has empty support. This may indicate an error of Type 2, because the object intent contains a combination of attributes impossible in the data domain, but the object may be correct as well. An expert could be asked if the combination of attributes in the object intent is consistent in the data domain. For such a question the information already input in the context is not used. More than that, this question is not sufficient to reveal an error of Type 1.

**Proposition 1.** *Let* $\mathbb{K} = (G, M, I), A \subseteq M$. *The set*

$$\mathcal{I}_A = \{B \to d \mid B \in \mathcal{MC}_A, d \in B'' \setminus A \cup \overline{A \setminus B}\},$$

*where* $\mathcal{MC}_A = \{B \in \mathcal{C}_A | \nexists C \in \mathcal{C}_A : B \subset C\}$ *and* $\mathcal{C}_A = \{A \cap g' \mid g \in G\}$, *is the set of all unit implications (or their non-trivial consequences with some attributes added in the premise) of Types 1 and 2 such that implications are valid in* $\mathbb{K}$, *not respected by* $A$, *and have not empty support.*

Proposition 1 allows one to find an algorithm for computing the set of questions to an expert revealing possible errors of Types 1 and 2. The pseudocode is pretty straightforward and is not shown here for the sake of compactness.
Since computing the closure of a subset of attributes takes $O(|G| \times |M|)$ time in the worst case, and we need to compute respective closures for every object in the context, the time complexity of the whole algorithm is $O(|G|^2 \times |M|)$.
We may now conclude that we are able to find possibly broken dependencies of two most common types in new objects. However, this does not always indicate broken real dependency, as we not always have enough information already input in our context. That is why we may only develop a hypothesis and ask an expert if it holds.
For more details, example, and proof of Proposition 1, please, refer to [9].

## 6 Debugging

### 6.1 Context Preparation

Normally debugging starts with a failure report. Such a report contains the input on which the program failed. By this we mean that our program was not able to output the expected result or did not finish at all. This implicitly defines "goal" function which is capable of determining either a program run was successful or not. We could imagine a case where we do not have any successful inputs, i.e. those inputs which were processed successfully by the program. However, it does not seem reasonable. In a such a case the best option seems to rewrite the code or look for obvious mistakes. Modern techniques of software development suggests running tests even before writing code itself; unless the tests are passed code is not considered finished. Therefore, successful inputs are at least those contained in the test suites.
As discussed in the beginning of this paper the problem of finding appropriate inputs was considered by different authors. This problem is indeed of essential importance for debugging. However, we do not aim at solving it. Instead we assume that inputs are already found (using user reports, random generator, or something else), processed (it is better if inputs are minimized, however, not necessary), and are at hands. We focus on processing the program runs on given inputs.
Our approach consists in the following. We construct two contexts: first with successful runs as objects, second with failed runs. In both cases attributes are

the lines of the code (conveniently presented via line numbers). We put a cross if during processing of the input the program has covered the corresponding line. So in both cases we record the information about covered lines during processing of the inputs. After contexts are ready we treat all the objects from the context with failed runs as new objects and try to find errors as described in the previous sections. Expected output is an implication $A \to B$. The interpretation is as follows; in successful runs whenever lines numbers $A$ are covered, lines numbers $B$ are covered as well. For some reason in the inspected failed run this is not the case. Debugging consists now in finding this reason. This is not absolutely automatic debugging, however, we receive some more clues and may find a bug without checking the written code line by line. More than that, this approach is strict, that is we say that it always happens, not with any probability. And it corresponds to the real situation: the bug *is* there, not with any probability.

### 6.2 Example

Consider the following function written in Python (example taken from [12]):

Listing 1.1: remove_html_markup [12]

```
1  def remove_html_markup(s):
2      tag   = False
3      quote = False
4      out   = ""
5      for c in s:
6          if (c == '<' and
7              not quote):
8              tag = True
9          elif (c == '>' and
10                 not quote):
11             tag = False
12         elif (c == '"' or
13                 c == "'" and
14                 tag):
15             quote = not quote
16         elif not tag:
17             out = out + c
18     return out
```

The goal of the function, as follows from its name, is to remove html markup from the input, no matter if it occurs inside or outside quotes. Therefore, we may formulate our goal as: no `<` in output. Such a formulation does not allow us to catch all the bugs (check input `"foo"` in contexts below), but it suffices for our purposes.

The function works as follows. After initialisation we have four "if" cases. The first and the second one checks if we have encountered a tag symbol outside of quotes. If so, the value of "tag" is changed. The third one checks if we have

encountered a quote symbol inside tag. This is important for not closing a tag if the closing symbol happens to be in one of the parameters (see inputs). If so, the value of "quote" is changed. The last "if" adds the current character to the output if we are outside the tag.

We consider the following set of inputs: `foo`, `<b>foo</b>`, `"<b>foo</b>"`, `"<b>a</b>"`, `"<b></b>"`, `"<>"`, `"foo"`, `'foo'`, `<em>foo</em>`, `""`, `<"">`, `<p>`, `<a href="">foo</a>`

Using the given outputs we obtain two contexts:

| Context with successful inputs | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| `foo` | | × | × | × | × | × | | | × | | | × | × | | | × | × | × |
| `<b>foo</b>` | | × | × | × | × | × | × | × | × | × | × | × | × | | | × | × | × |
| `"foo"` | | × | × | × | × | × | | | × | | | × | × | | × | × | × | × |
| `'foo'` | | × | × | × | × | × | | | × | | | × | × | × | | × | × | × |
| `<em>foo</em>` | | × | × | × | × | × | × | × | × | × | × | × | × | | | × | × | × |
| `<a href="">foo</a>` | | × | × | × | × | × | × | × | × | × | × | × | × | | × | × | × | × |
| `""` | | × | × | × | × | × | | | × | | | × | | | × | | | × |
| `<"">` | | × | × | × | × | × | × | × | × | × | × | × | | | × | | | × |
| `<p>` | | × | × | × | × | × | × | × | × | × | × | × | × | | | × | | × |

| Context with failed inputs | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| `"<b>foo</b>"` | | × | × | × | × | × | × | | × | × | | × | × | | × | × | × | × |
| `"<b>a</b>"` | | × | × | × | × | × | × | | × | × | | × | × | | × | × | × | × |
| `"<b></b>"` | | × | × | × | × | × | × | | × | × | | × | × | | × | × | × | × |
| `"<>"` | | × | × | × | × | × | × | | × | × | | × | × | | × | × | × | × |

Fig. 1: Contexts with failed and successful runs

It is easy to notice that the only difference between failed inputs as context objects is their names.
Adding any of failed inputs to the first context yields the following implication:

$$7, 13, 15 \rightarrow 8, 11$$

What is essentially said is if we happened to cover lines 7, 13, 15, we should have also had been inside tag (lines 8 and 11). Given some thought and attention we realize that this is absolutely true, because it is not clear how we could reach line 15 without having "tag"$= true$, as this condition is checked in line 14.
In Python as well as in many other languages logical operation "and" has a higher priority as "or", so condition of the third "if" `(c == '"' or c == "'" and tag)` is implicitly transformed in `(c == '"' or (c == "'" and tag))`, that is why on lines 12 and 13 brackets are forgotten. After debugging the condition should look as follows: `((c == '"' or c == "'") and tag)` and the program runs correctly.

## 7 Conclusion

A technique for finding errors of two types in new object intents is presented. As opposed to finding the canonical basis of the context the proposed algorithm terminates much faster. Based on this technique an approach for debugging source code is presented. This approach is capable of finding strict dependencies between lines of source code covered in successful and failed runs. The output is a logical expression which allows to debug the source code using the logic of the program. This may get us one step closer to automated debugging.

## References

1. Mikhail Babin and Sergei O. Kuznetsov. Recognizing pseudo-intent is conp-complete. *Proc. 7th International Conference on Concept Lattices and Their Applications, University of Sevilla*, pages 294–301, 2010.
2. Holger Cleve and Andreas Zeller. Locating causes of program failures. In Gruia-Catalin Roman, William G. Griswold, and Bashar Nuseibeh, editors, *ICSE*, pages 342–351. ACM, 2005.
3. Felix Distel and Barış Sertkaya. On the complexity of enumerating pseudo-intents. *Discrete Applied Mathematics*, 159(6):450–466, 2011.
4. Bernhard Ganter. Two basic algorithms in concept analysis. *Preprint-Nr. 831*, 1984.
5. Bernhard Ganter, Gerd Stumme, and Rudolf Wille, editors. *Formal Concept Analysis, Foundations and Applications*, volume 3626 of *Lecture Notes in Computer Science*. Springer, 2005.
6. Robert Godin and Petko Valtchev. Formal concept analysis-based class hierarchy design in object-oriented software development. In Ganter et al. [5], pages 304–323.
7. J.-L. Guigues and V. Duquenne. Familles minimales d'implications informatives résultant d'un tableau de données binaires. *Math. Sci. Hum*, 24(95):5–18, 1986.
8. John L. Pfaltz. Using concept lattices to uncover causal dependencies in software. In *Proc. Int. Conf. on Formal Concept Analysis, Springer LNAI 3874*, pages 233–247, 2006.
9. Artem Revenko and Sergei O. Kuznetsov. Finding errors in new object intents. In *CLA 2012*, pages 151–162, 2012.
10. Uwe Ryssel, Felix Distel, and Daniel Borchmann. Fast computation of proper premises. In Amedeo Napoli and Vilem Vychodil, editors, *International Conference on Concept Lattices and Their Applications*, pages 101–113. INRIA Nancy – Grand Est and LORIA, 2011.
11. Gregor Snelting and Frank Tip. Reengineering class hierarchies using concept analysis. *SIGSOFT Softw. Eng. Notes*, 23(6):99–110, November 1998.
12. Andreas Zeller. Software debugging course. https://www.udacity.com/course/cs259.
13. Andreas Zeller and Ralf Hildebrandt. Simplifying and isolating failure-inducing input. *IEEE Trans. Software Eng.*, 28(2):183–200, 2002.

# Systems vs. Methods: an Analysis of the Affordances of Formal Concept Analysis for Information Retrieval*

Francisco J. Valverde-Albacete[1],* and Carmen Peláez-Moreno[2]

[1] Departamento de Lenguajes y Sistemas Informáticos
Univ. Nacional de Educación a Distancia, c/ Juan del Rosal, 16. 28040 Madrid, Spain
fva@lsi.uned.es
[2] Departamento de Teoría de la Señal y de las Comunicaciones
Universidad Carlos III de Madrid, 28911 Leganés, Spain
carmen@tsc.uc3m.es

**Abstract.** We review previous work using Formal Concept Analysis (FCA) to build Information Retrieval (IR) applications seeking a wider adoption of the FCA paradigm in IR. We conclude that although a number of systems have been built with such paradigm (FCA *in* IR), the most effective contribution would be to help establish IR on firmer grounds (FCA *for* IR). Since such an approach is only incipient, we contribute to the general discussion by discussing affordances and challenges of FCA for IR.

## 1 Introduction

Modern Information Retrieval (IR) is a wide field with several different concerns pulling in different directions. Under the competitive task evaluation paradigm [29], IR strives to solve *tasks* using any of a variety of *models*, mostly by *Machine Learning* techniques [41]. A glimpse at the main types of IR models can be found in [27], reproduced here as Fig. 1.

Perhaps the simplest and best known task is that of *ad hoc retrieval*, where a corpus of documents is searched with a number of topics (Sec. 2), but certainly the most prevalent task is the familiar *Web retrieval*. They are also typical instances of *batch* and *interactive retrieval tasks*, respectively.

The Formal Concept Analysis (FCA)[3] community has been *implementing* Information Retrieval (IR) systems for well over 25 years, starting with [21]. Yet few mainstream IR practitioner confess to understanding the bases of FCA, a testimony of the scarce impact of the former in mainstream IR.

* FJVA has been partially supported by EU FP7 project LiMoSINe (contract 288024). CPM has been partially supported by the Spanish Government-Comisión Interministerial de Ciencia y Tecnología project TEC2011-26807 for this paper.

[3] This paper is targeted at FCA practitioners, so the reader is expected to be acquainted with the principal results of FCA. For analogue papers targeted at IR practitioners see [35, 42].

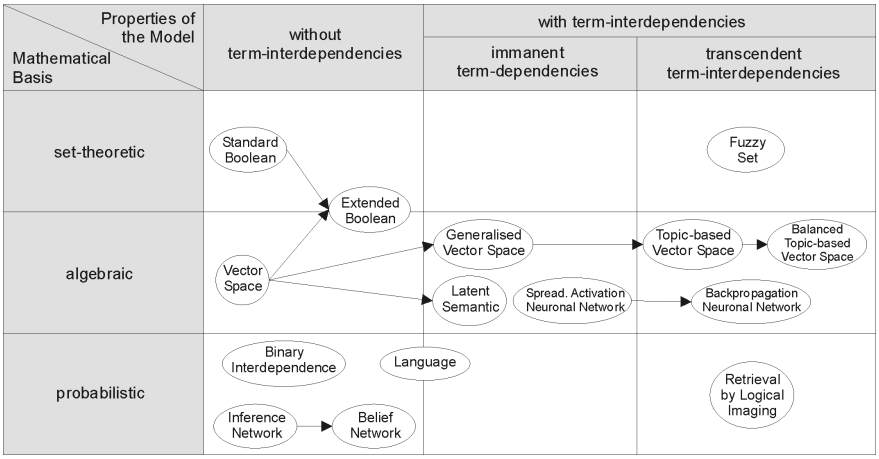



Fig. 1: A schematics showing the main types of Information Retrieval models (after [27])

To the best of our knowledge only two information retrieval-incepted books have realised the potential of FCA for IR: [45] and [11]. On the one hand, Van Rijsbergen briefly notes down that the Boolean Retrieval Model is captured in terms of Galois connections between documents and features (terms) [45, p. 37], although he includes there the inverse index on terms and documents which may best be conceived in terms of a Galois adjunction[42]. On the other hand, Dominich makes a very cursory review of the state-of-the-art up until 2008 [11]. He notes down the work of [37] on faceted information retrieval and that of [6] on browsing Web retrieval results with concept lattices, and the disjunctive approximation to boolean retrieval of [32]. Curiously, the data-driven nature of FCA is downplayed in this work.

In the FCA camp, the broadest review is still [6] but [5, 23, 37, 38] have narrower foci. Notice that both [11, 37] review work in lattice-based IR systems prior to the groundbreaking [21], but pre-FCA emphasis is in *designing* the lattice instead of *obtaining* it from the relevance relation: the *data driven* quality of FCA is missing in this early work, e.g. [33].

We believe that part of the explanation for this divide may be that only the most simple, basic tasks in IR—and using the oldest IR models—have been successfully tackled with FCA techniques. After all, IR in some 60+ years has developed its own set of techniques, methods for research and testing and is practised by, probably, the most thriving community in ICT. It is only natural that FCA can only be considered as a subsidiary discipline to such endeavour. Or not?

In this paper we want to put forward the distinction between FCA *in* IR and FCA *for* IR, that is *implementing IR systems with FCA* vs. *augmenting IR with the methods and ideas of FCA*. We claim that most of the work so far has been

in *FCA in IR* and the time is ripe to expound on a *FCA for IR*, that is a theory of the *affordances* and *challenges* of using FCA to solve IR tasks, already started in [6]. Here we use affordances in the sense of [34], to refer to "the actionable properties between the world and an actor", that is, the 'world' of FCA and the 'actor' that is an IR practitioner.

This paper is about raising awareness of these two conceptions of the role of FCA vis-à-vis IR. For this purpose, we introduce in Sec. 2 a prototypical information retrieval task to make explicit what types of problems an IR practitioner comes up with. In Sec. 3 we review to what extent FCA actually solves such problems by supplying a set of *affordances* of FCA *for* IR. Finally, we discuss in Sec. 4 what are further challenges that FCA has to solve for a wider adoption in a number of data-intensive application domains, including IR.

## 2 A prototypical information retrieval task

To guide our exposition we will discuss the *ad-hoc retrieval task*, that is, the task where the IR system is expected to produce the documents relevant to an arbitrary user need as expressed in a one-off, user-initiated query [29, p. 3]. Although Web retrieval is perhaps the prevalent IR task at present, ad-hoc retrieval is the best studied one and it admits many different models. In the following, we expand the *modelling* of this task propounded in [42] as a script to discuss affordances and challenges in using FCA for IR tasks.

**A model for batch ad-hoc tasks** To fix notation, we adapt the formal model put forward by Fuhr [17] reproduced in Fig. 2—although we interpret the signs there differently—and we let $\overline{Q}$, $\overline{D}$, and $\overline{R}$ respectively stand for a *set of information needs* for a querying user, a set of *information-bearing percepts* and a *psychological capability* whereby a particular user is going to judge the relevance of the information percepts for her information needs.

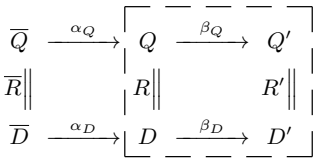



Fig. 2: An adaptation of the conceptual model of Fuhr [17] with the concepts dealt with in this paper highlighted.

Figure 2 highlights the data and models we will address in this paper: let $Q$, $D$ and $R$ be the outcome of as many instantiation processes of the above-mentioned information needs, information supplies and relevance judgments, respectively. We will call them *queries*, *documents* and *relevance judgments* and assume that

the relevance judgment representations adopt the form of a relevance relation, $R \subseteq D \times Q$ . Finally, let $Q'$, $D'$, $R'$ be the *query representations*, *document representations* and the *relevance judgments in representation space* respectively, so that $R' \subseteq D' \times Q'$ .

Whereas Fuhr's model considers queries, documents and judgements to be *inside* the information retrieval system, we consider them *both inside and outside*, since they are more properly conceived as *(multimedia) recordings* of the psychological entities and processes considered above. They have an immanent existence independent of the system yet are related to them by their representations. However, representations arise when we try to approximate the information content of queries and documents inside an IR system, hence they are sometimes called *surrogates or surrogate representations (for their records).*

Although the model posits four maps between the above-introduced domains, for practical reasons it is common to concentrate on only two

**A query representation process,** $\beta_Q : Q \to Q'$, mapping from queries to query representation suitable for processing in a particular information retrieval system.

**A document representation process,** $\beta_D : D \to D'$, mapping from documents to document representations.

Therefore, we limit ourselves to the domains, mappings and sets enclosed by the square in Fig. 2, the recording- and representation-related domains.

**Assessment** The *ideal IR system* $S_{D,Q}(R) =< \varrho_R >$ would consist in a *relevance function* $\varrho_R$ describing relevant documents where $\varrho_R(q_i)$ is the set of documents *relevant to query* $q_i$ as dictated by the ideal relevance relation $R$. But in the process of building an IR system we may incur modelling errors, approximation, etc., whence we accept that the actual relation implemented will be the approximated relevance $\hat{R} \neq R$ for the *implemented IR system* $S_{D,Q}(\hat{R}) =< \varrho_{\hat{R}} >$ . Its *retrieval function* may only return $\varrho_{\hat{R}}(q_i)$ the set of documents *retrieved for the same query as dictated by the approximate relevance* $\hat{R}$,

$$\begin{aligned} \varrho_R : Q \to 2^D & \qquad \varrho_{\hat{R}} : Q \to 2^D \\ q_i \mapsto \varrho_R(q_i) = \{d_j \in D | d_j R q_i\} & \qquad q_i \mapsto \varrho_{\hat{R}}(q_i) = \{d_j \in D | d_j \hat{R} q_i\} \ . \end{aligned} \tag{1}$$

The batch retrieval task can be subjected to the so-called "Cranfield model of Information Retrieval system evaluation" [31], where a set of document records, or *collection*, $D_T \subseteq D$, a set of sampled query records, *topics*, $Q_T \subseteq Q$, and a set of relevance judgments involving documents and query records, $R_T \subseteq D_T \times Q_T$ are known. Assessing the quality of $S_{D,Q}(\hat{R})$ means, essentially, comparing $R$ and $\hat{R}$ : For a given query $q$, the system would retrieve documents $\varrho_{\hat{R}}(q)$ whereas the relevant documents are given by the prescribed relevance as $\varrho_R(q)$ . Therefore the retrieved relevant documents for each query $q \in Q$ would be

$\varrho_R(q) \cap \varrho_{\hat{R}}(q)$, and we would have *precision* $P_{\hat{R}}$ and *recall* $R_{\hat{R}}$—or any measure derived therefrom—as

$$P_{\hat{R}}(q) = \frac{|\varrho_R(q) \cap \varrho_{\hat{R}}(q)|}{|\varrho_{\hat{R}}(q)|} \qquad R_{\hat{R}}(q) = \frac{|\varrho_R(q) \cap \varrho_{\hat{R}}(q)|}{|\varrho_R(q)|} \ . \tag{2}$$

**A decomposition of the problem.** We believe it is convenient to conceptually decompose the synthesis of $S_{D,Q}(\hat{R})$ into the following problems[cfr. 6, §. 4]:

*Problem 1 (Representation).* Given different spaces of queries $Q$ and their representations $Q'$ find a mapping $\beta_Q$ between them. Do likewise for documents $D$, their representations $D'$ and a *surjective* mapping $\beta_D$ between them.

*Problem 2 (Generalization).* Given local information about the relevance relation $R$ in the form of a training subset $R'_T = D'_T \times Q'_T$, extend/generalise such information to $\hat{R}' \subseteq D' \times Q'$.

*Problem 3 (Surrogate implementation).* Given domains of documents $D$ and queries $Q$ (whether they be descriptions or representations), a querying hypothesis and an estimated relevance relation $\hat{R}$, build an information retrieval system that faithfully implements the prescribed relevance[4].

Once solved these problems we can build the retrieval set as

$$\varrho_{\hat{R}}(q) = \beta_D^{-1}[\varrho_{\hat{R}'}(\beta_Q[q])] \tag{3}$$

where we have taken the precaution of making all of the functions apply over *sets* rather than singletons.

*Problem 4 (Post-retrieval interaction).* Given the answer set to a query $\varrho_{\hat{R}}(q)$ present it to the user in an effective manner.

Note that in standard IR engineering practice the steps of retrieving document representations and then finding their original document are often aggregated by means of an inverted index. Also, (3) is often complemented with *retrieval status value* for each result, a number stating the *degree of relevance* of each retrieved document to the query.

## 3 Affordances of FCA for IR

This list is going to be informally structured as a sort of proof: first we state what we consider the affordances of FCA for IR and then we explain the reasoning behind our assertion.

**Affordance 1 (Solving problem 3 in the conjunctive Boolean Model).** *FCA implements the (conjunctive) Boolean Keyword model.*

[4] We use here $\hat{R}$ as a variable ranging over possible relation values, not necessarily the optimal one.

Suppose that there exists a set of keywords $T$ [5], queries are represented as keywords $Q' \equiv T$, documents are represented as set of keywords $D' \equiv 2^T$, and estimated relevance $\hat{R}'$ is defined by means of the inclusion relation $d'\hat{R}'q' \Leftrightarrow d' \ni q'$ . The retrieval function is easy to write $\varrho_{\hat{R}'}(\{t\}) = \{d \in D \mid q \in d\}$, but what are we to expect when supplying several queries, that is, several keywords?

To implement *conjunctive querying* we produce the intersection of the result sets, that is, for $B = \{q_i\}_{i\in I}$ we have

$$\varrho^1_{\hat{R}'}(\{q_i\}_{i\in I}) = \{d \in D \mid \forall i \in I, q_i \in d\} = \cap_{i\in I}\{d \in D \mid q_i \in d\} \ .$$

In that case, the more keywords a query has the less documents the retrieval function returns, that is, $q'_1 \subseteq q'_2$ implies $\varrho(q'_1) \supseteq \varrho(q'_2)$. Then we realise that this retrieval function is the *query polar* $\varrho^1_{R'}(q)$ of the Galois Connection in Fig. 3.(a)

$$\varrho^1_{\hat{R}'} : 2^Q \to 2^D \qquad \varrho^2_R : 2^Q \to 2^D$$
$$\varrho^1_{\hat{R}'}(B) = \{d'_i \in D' | \forall q' \in B, d'_i\hat{R}'q'\} \qquad \varrho^2_R(B) = \{d'_i \in D' | \exists q' \in B, d'_i\hat{R}q'\}$$

$$\iota^1_{\hat{R}'} : 2^{D'} \to 2^{Q'} \qquad \iota^2_{\hat{R}'} : 2^{D'} \to 2^{Q'}$$
$$\iota^1_{\hat{R}'}(A) = \{q' \in Q' \mid \forall d' \in A, d'R'q'\} \qquad \iota^2_{\hat{R}'}(A) = \{q' \in Q' \mid d'R'q' \Rightarrow d' \in A\}$$

(a) Galois connection (b) Galois adjunction

Fig. 3: Galois connection and adjunction between two powersets of terms that implement the conjunctive and disjunctive models of Boolean retrieval, respectively.

This is one of the contributions of [21], the first paper to use *FCA in IR*, that is, to build a Galois connections that implements an IR system, to the best of our knowledge. Most of the work in FCA in IR uses this model [3, 9, 13, 37], with the notable exception of the work starting with [32], who define relevance in a way that leads to the disjunctive model of Fig. 3.(b). In this case, $\varrho^2_{\hat{R}'}(B) = \{d_i \in D | \exists q \in B, d_i R q\}$ , but, since there are some tricks to representing this in a concept lattice [44], the authors of [32] develop a browsing model of their own.

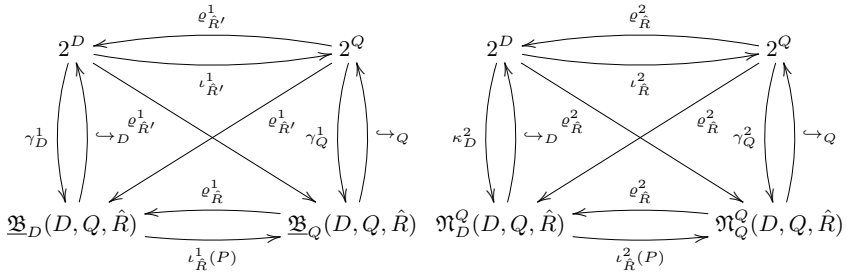
**Affordance 2.** *FCA implements query term expansion*

In fact, the Galois connection has "another half", the *document polar*. Let $A \subseteq D'$ be a set of documents. Then the set of queries for which *all* those documents are relevant is $\iota^1_{\hat{R}'}(A) = \{q' \in Q' \mid \forall d' \in A, d'R'q'\}$ . Actually retrieval sets come in pairs called *formal concepts*[6]. In our example, a formal

[5] This is sometimes called the *bag-of-keywords* model of documents.

[6] In [21] they were originally called "complete pairs".

concept $(A, B)$ is a pair of a set of documents $A$ and a set of queries $B$ so that all the documents in $A$ are relevant to all the queries in $B$, and dually, $\left(A = \varrho^1_{\hat{R}'}(B), B = \iota^1_{\hat{R}'}(A)\right)$ . These pairs come from the properties of the polars in the Galois connection, as described in Fig. 4: the composition of the polars are extensive, idempotent operators, that is, *closure operators*.




(a) The concept lattice $\underline{\mathfrak{B}}(D, Q, \hat{R})$ and two closure operators $\gamma^1_D$ and $\gamma^1_Q$ on documents and queries.

(b) The lattice of neighbourhood of queries $\mathfrak{N}^Q(D, Q, \hat{R})$, an interior operator $\kappa^2_D$ on documents and a closure operator $\gamma^2_Q$ on queries.

Fig. 4: Galois connections describing conjunctive (left) and disjunctive (right) boolean retrieval.

Note that for a set of queries $B \in Q$, $\gamma^1_{\hat{R}'}(B) \geq B$ hence querying through formal concepts expands the query sets in a data-dependent manner. This was noted cursorily in [21] but is thoroughly explained in [6, Chap. 3] whose authors have contributed the most to this line of work.

**Affordance 3.** *FCA provides for integrated browsing and querying.*

As previously noted, query submission in a concept lattice-based IR system is just an application of the query polar, which obtains the concept whose extent is the retrieval set, and whose intent is the extended query. This acts as a querying mechanism.

On the other hand, formal concepts have a natural order based in the inclusion order of extents or the dual inclusion order of intents, $(A_1, B_1) \leq (A_2, B_2) \Leftrightarrow A_1 \subseteq A_2 \Leftrightarrow B_1 \supseteq B_2$ . Furthermore, the Fundamental Theorem of Concept Lattices asserts that this order between concepts is a complete lattice [20, p. 20], representable as an *order diagram*.

Godin et al. [21, 22] put forth the idea that lower and upper neighbours as well as parallel concepts define *a topology for browsing in a (concept) lattice* (see Fig. 5 ). Consider a *concept in focus* $C$,

– Below it lie its lower covers, those concepts with more stringent (higher cardinality) query sets.
– Above it lie its upper covers, those which have less stringent (lower cardinality) query sets.
– To each side of the concept in focus stand those *sibling concepts* sharing parents (and descendants) with it. They have incompatible query sets (inconsistent with the focus concept intent).

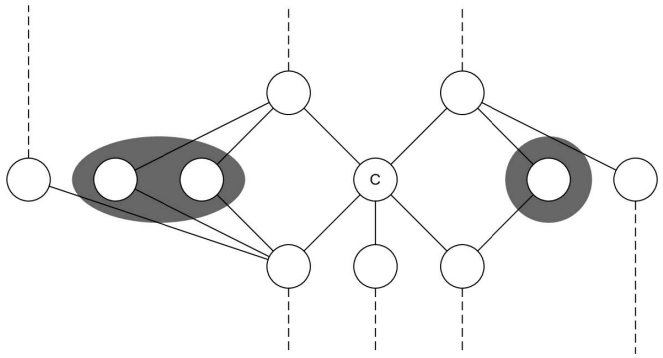



Fig. 5: Schematic representation of a concept $C$ in focus in a concept lattice with upper and lower neighbours, from [12] rather than [21]. Sibling nodes are within shaded areas.

Although previous work had noted the interest of lattices for navigation, to the extent of our knowledge, Godin et al. were the first to tie the modification of queries (and therefore retrieval sets) to navigation in a systematic manner. For in-detail reviews of this affordance in the context of Personal Information Systems see [12, 13].

**Affordance 4.** *FCA provides visualization schemes for the document-query lattice at different scales.*

The scales we refer to in this affordance are those related to the visual and informational complexity of the lattice. Complexity scales are in other contexts termed the *micro-, meso-*, *macro-* and *mega-scales.*

The local neighbourhood of a formal concept illustrated in Fig. 5 was posited in [21, 22] and developed in a number of works [12]. It is a micro-visualization device depicting the part of the lattice surrounding a particular concept in focus whether incorporating a *fisheye view* [5, 21] or not [12].

On the other hand, the order diagram of the concept lattice acts as a meso-scale visualization technique. Similarly, visualizing only the concepts that lie below—or above—a focus concept produces visualization devices of comparable complexity and can be considered meso-scale visualizations. Here we consider the

mapping of the downset of the focus as a tree as in [4]. Furthermore, the use of *attribute and object projections* on the whole lattice, *reduced labelling* and *nested line diagrams* [20, 39] are all tools that help us balance displayed information vs. visual complexity allowing us to display complex lattices at the mesoscale.

For those cases when these complexity-reducing strategies are not sufficient, very little work has been done on observing lattices at the macro-scale—let alone the mega-scale—sacrificing concept and local structure readability for the quick glimpse of emerging features like height, width, overall shape, concept density, etc. For an illustration of such problems, see the lattices in [24]. Recently, [14] have proposed a technique to embed any concept lattice onto a boolean lattice of similar complexity which acts as a representation space disposing of a lot of information: its usability is, as yet, unassessed.

**Affordance 5 (Solving problem 4).** *FCA provides retrieval-set navigation.*

This niche application of FCA is perhaps the best-know to the IR community [25, § 10.7]. It is a natural consequence of treating the retrieval set as a subcorpus (of snippets, possibly) and using FCA to establish ordering relations between them as induced by their terms. Perhaps the first to propose this use of concept lattices is [3], and it is thoroughly explained in [6], usability studies included. Systems implementing also a post-retrieval visualization of Web retrieval searches or Meta-searches can be found in [8–10, 26].

**Affordance 6.** *FCA captures naturally occurring (immanent) term dependencies*

If terms were independent, then concept lattices, at least from the perspective of terms, should be boolean: all possible combinations of terms would arise as intents, but this is never the case. Since the inception of the first FCA in IR systems it was noticed that particular groupings of terms occur naturally in documents and this is reflected in the system of intents. Of course, this dovetails into the Automatic Expansion of Queries mentioned in Affordance 2: modelling term dependencies is how automatic query expansion is catered to. In terms of IR models, this means that the model implemented by FCA is actually in the empty square in Fig. 1. Carpineto and Romano [7] have investigated this issue heavily both from the point of view of IR and from that of FCA [see 6, §3.1 for a rather extensive review].

**Affordance 7.** *FCA scaling implements faceted search & navigation.*

Sometimes certain sets of attributes have different multiple possible values and/or special relationships between those values—such as hierarchies—and it is interesting for navigation purposes to see the collection of documents through the prism of those relations. This is called *faceted information retrieval*.

In FCA, discrete multi-valued attributes or otherwise-related attributes may be rendered in a data-dependent fashion by means of the process of *scaling attributes* [20]. But the effectiveness of this process depends extraordinarily on the experience of the expert user doing the encoding of attributes.

Although faceted navigation is explicitly mentioned in [21], it seems that FaIR was the first actual implementation using FCA [11, 37], albeit for a restricted application, thesaurus exploration. A review of faceted boolean IR can be found in [13]—as applied to Personal Information Systems—with an emphasis on usability, visualisation and navigation.

An alternative to scaling is *logical concept analysis, LCA* where any logical formula may be used to characterize intents [16], and it has been used to build a Personal Information Retrieval system for photos based in metadata [15]. Note that LCA is a *proper generalization* of FCA.

Although a number of other topics suggest themselves for this review—such as Semantic Filesystems [15, 30] or the duality of Information Pull & Push—to put them in context would demand more space than we have at our disposal.

## 4 Discussion: challenges of IR for FCA

**Dealing with redundancy and noise in data.** As in other subfields of machine learning and pattern recognition, functions $\beta_Q$ and $\beta_D$ of Fig. 2 can be thought of as functions that reduce unnecesary redundancy and noise.

For instance, when dealing with text we should be aware that natural language is widely-acknowledged to be *extremely redundant*: many words, expressions, constructions, etc. convey the same ideas and essentially make the complexity of the system grow. Furthermore, if words are considered terms for IR, every single word encountered when tokenizing a text *invokes* all of the senses conventionally assigned to it in a language. Since it is these senses that are purported to mediate the actual relation between the terms and documents, serendipity may reinforce not just the *originally intended* sense but also some *unrelated senses* due to surrounding context. This is a manifestation of *noise*, e.g. undesired content. And these problems can only be compounded by the ubiquity of synonymy and polysemy in Natural Language.

On top of the excess complexity incurred by redundancy, it is well-known that FCA is very sensitive to the spureous absence or presence of crosses in the incidence relation between documents and terms: the addition or deletion of any such incidences may as much as double or halve the number of concepts in the lattice[20]. If FCA is to succeed in dealing with such problems it has to devise methods to cope with this kind of noise at the incidence relation level.

**Big data, supervised operation and training.** The main challenge for FCA to be of any help to IR is *scalability*. Perhaps the maximum reported size for FCAinIR systems is some thousands of documents [39], while it is customary for present-day IR systems to have millions of documents. There is no easy way to overcome this inherent limitation for concept lattices: building them is just too costly in time and space[28].

One way to address the complexity of Big Data would be to assume the data-driven paradigm of Machine Learning or Pattern Recognitition [41]. However, FCA is an *unsupervised* machine learning technique: all of the information in the

lattice stems from the information in the documents, the terms and the incidence relation between them. But the solution to Problem 2 seems to entail a supervised procedure whereby the training topic judgements can be used to improve unseen topics. At present, relevance in the boolean case is dictated *a priori* and there is no room for such supervision, only for post-retrieval assessment.

Unless this mismatch is addressed, machine learning-inspired techniques will still outperform FCA or address tasks which FCA simply cannot attempt.

**Catering to more complex IR models.** The history of IR seems to be an account of progressively complex modelling of textual data. From the boolean bag-of-word models, conceived as boolean vectors, it is easy to take a conceptual jump towards softer weighting schemes in the Vector Space Model. From constant-dimension vectors in the Vector Space Model, it is an easy jump to probability-weighted formal series, that is *(generative) language models*. Similarly, from vector description in non-orthogonal systems of generators it is easy to conceive an orthonormal basis wherein to represent vectors, which is the essence of Latent Semantic Indexing, and so on. All such conceptual leaps are steps in a process of continual algebraization of the underlying models that entail better modelling or learning capabilities in IR.

Such a process has barely begun in FCA with the so-called *generalizations* of FCA, [1, 2, 43]. Nevertheless, coincidences can be seen in all such evolutions: it seems that the basis for any possible generalization of FCA is the theory of residuated semirings [36], while many of the models in IR have semiring-based costs (probabilities, log-probabilities, etc.)

In a similar tone, most of the implementations of FCA in IR deal with the conjunctive querying case, with the previously noted exception of [32], which implements a sort of disjunctive model. If FCA wants to embrace all possible "conceptualization modes" for queries, it needs to standardize and use habitually the whole gamut of Galois connections available [44].

**A concluding note...** On the one hand, the FCA community has an increasing collective expertise in the development of IR applications (FCA in IR) in different domains and tasks, but has achieved only limited impact in IR proper, for the reasons explained above among others.

On the other hand, FCA has strong theoretical foundations that can help IR understand better its own models and basic assumptions (FCA for IR). Yet FCA would very much profit by the assessment-oriented approach to task-solving now prevalent in the field of IR. It would seem FCA only needs to embrace the new generalization efforts outgrowing from the dynamic flourishing of FCA these past 15 years to do so.

At the risk of being too poetical, since IR is highly empirical (and in the quest for firmer theoretical grounding) and FCA highly theoretical yet completely data-driven (but still needs to come to terms with task-realities) there is still hope for a middle ground/sweet spot where someday the twain may meet.

# Bibliography


[1] R. Belohlavek. Fuzzy Galois connections. *Math. Logic Quarterly*, Jan. 1999.

[2] A. Burusco and R. Fuentes-González. The study of the L-fuzzy concept lattice. *Mathware & soft computing*, Jan. 1994.

[3] C. Carpineto and G. Romano. ULYSSES: A lattice-based multiple interaction strategy retrieval interface. In *Human-Computer Interaction, 5th International Conference, EWHCI'95*, pages 91–104, Berlin, Heidelberg, 1995. Human-Computer Interaction, 5th International Conference, EWHCI'95.

[4] C. Carpineto and G. Romano. Exploiting the potential of concept lattices for information retrieval with CREDO. *Journal of Universal Computer Science*, 10:8, 2004.

[5] C. Carpineto and G. Romano. *Concept Data Analysis*. Theory and Applications. John Wiley & Sons, Ltd, Chichester, UK, Sept. 2005.

[6] C. Carpineto and G. Romano. Using concept lattices for text retrieval and mining. In *Formal Concept Analysis*, pages 161–179. Springer Verlag, 2005.

[7] C. Carpineto and G. Romano. A Survey of Automatic Query Expansion in Information Retrieval. *Journal of the ACM*, 44(1):1–50, Jan. 2012.

[8] C. Carpineto, A. Della Pietra, S. Mizzaro, and G. Romano. Mobile clustering engine. In M. Lalmas, editor, *Advances in Information Retrieval*, pages 155–166. Springer, 2006.

[9] J. M. Cigarrán, J. Gonzalo, A. Peñas, and F. Verdejo. Browsing search results via Formal Concept Analysis: Automatic selection of attributes. In *Concept Lattices*, volume LNAI 2961, pages 311–331. Springer, 2004.

[10] J. M. Cigarrán, A. Peñas, J. Gonzalo, and F. Verdejo. Automatic selection of noun phrases as document descriptors in an FCA-based information retrieval system. In Ganter and Godin [18], pages 49–63.

[11] S. Dominich. *The Modern Algebra of Information Retrieval*. The Information Retrieval Series. Springer, 2008.

[12] J. Ducrou and P. Eklund. SearchSleuth: The conceptual neighbourhood of an web query. In *Proceedings of the 5th International Conference on Concept Lattices and their Applications, (CLA07)*, pages 253–263, Montpellier, France, Oct. 2007.

[13] J. Ducrou and P. W. Eklund. Faceted document navigation. In P. Hitzler and H. Schärfe, editors, *Conceptual Structures in Practice*. Chapman & Hall / CRC Press, 2009.

[14] J. M. Fernández-Calabozo, C. Peláez-Moreno, and F. Valverde-Albacete. WebGeneKFCA: an On-line Conceptual Analysis Tool for Genomic Expression Data. In L. Szathmary and U. Priss, editors, *Concept Lattices and Applications (CLA 2012)*, pages 345–350, Oct. 2012.

[15] S. Ferré. CAMELIS: Organizing and Browsing a Personal Photo Collection with a Logical Information System. *Proceedings of the 5th International Conference on Concept Lattices and their Applications, (CLA07)*, pages 112–123, Sept. 2007.

[16] S. Ferré and O. Ridoux. Introduction to logical information systems. *Information Processing and Management*, 40:383–419, Jan. 2004.

[17] N. Fuhr. Probabilistic models or information retrieval. *The Computer Journal*, 35(3):243–255, 1992.

[18] B. Ganter and R. Godin, editors. *Proceedings of the 3nd International Conference on Formal Concept Analysis, (ICFCA2005)*, number 3403 in LNAI, Berlin, Heidelberg, February 2005. Springer.

[19] B. Ganter and L. Kwida, editors. *Supplementary Proceedings of the 4th International Conference on Formal Concept Analysis, (ICFCA2006)*, February 2006. Verlag Allgemeine Wissenschaft.

[20] B. Ganter and R. Wille. *Formal Concept Analysis: Mathematical Foundations*. Springer, Berlin, Heidelberg, 1999.

[21] R. Godin, E. Saunders, and J. Gecsei. Lattice model of browsable data spaces. *Information Sciences*, 40:89–116, 1986.

[22] R. Godin, J. Gecsel, and C. Pichet. Design of a browsing interface for information retrieval. In *Proceedings of the 12th International Conference on Research and Development in Information Retrieval (ACM SIGIR '89)*, pages 32–39, Cambridge, MA, 1989. ACM.

[23] R. Godin, G. Mineau, R. Missaoui, and H. Mili. Méthodes de classification conceptuelle basées sur les treillis de Galois et applications. *Revue d'intelligence artificielle*, 9(2):105–137, 1995.

[24] T. Hannan and A. Pogel. Spring-based lattice drawing highlighting conceptual similarity. *Formal Concept Analysis*, pages 264–279, 2006.

[25] M. Hearst. *Search User Interfaces*. Cambridge University Press, 2009. ISBN 9780521113793.

[26] B. Koester. FooCA: Enhancing Google information research by means of Formal Concept Analysis. In Ganter and Kwida [19], pages 1–17.

[27] D. Kuropka. *Modelle zur Repräsentation natürlichsprachlicher Dokumente. Ontologie-basiertes Information-Filtering und -Retrieval mit relationalen Datenbanken*. Advances in Information Systems and Management Science. logos-verlag.de, 2004.

[28] S. O. Kuznetsov and S. A. Obiedkov. Comparing performance of algorithms for generating concept lattices. *Journal Of Experimental & Theoretical Artificial Intelligence*, 14(2-3):189–216, Apr. 2002.

[29] C. D. Manning, P. Raghavan, and H. Schütze. *Introduction to Information Retrieval*. Cambridge University Press, 2008.

[30] B. Martin. Formal concept analysis and semantic file systems. In P. Eklund, editor, *Concept Lattices*, pages 88–95. Springer Berling Heidelberg, 2004.

[31] C. T. Meadow, B. R. Boyce, and D. H. Kraft. *Text Information Retrieval Systems*. Library and Information Sciences. Academic Press, San Diego and San Francisco, second edition, 2000.

[32] N. Messai, M. D. Devignes, A. Napoli, and M. Smail-Tabbone. BR-Explorer: An FCA-based algorithm for Information Retrieval. In *Proceedings of the 4th International Conference on Concept Lattices and their Applications, CLA '06*, 2006.

[33] C. Mooers. A mathematical theory of language symbols in retrieval. In *Proceedings of the International Conference on Scientific Information*, Washington, D.C., 1958.

[34] D. A. Norman. Affordance, conventions, and design. *Journal of the ACM*, 6(3):38–43, May 1999.

[35] R. Pedraza-Jiménez, F. J. Valverde-Albacete, and A. Navia-Vázquez. A generalisation of fuzzy concept lattices for the analysis of web retrieval tasks. In *Proceedings of the International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems, (IPMU'06)*, July 2006.

[36] S. Pollandt. *Fuzzy-Begriffe. Formale Begriffsanalyse unscharfer Daten.* Springer, Heidelberg, 1997.

[37] U. Priss. Lattice-based information retrieval. *Knowledge Organization*, 27 (3):132–142, 2000.

[38] U. Priss. Formal concept analysis in information science. In B. Cronin, editor, *Annual Review of Information Science and Technology (ARIST)*, pages 521–543. Information Today, Inc., Jan. 2006.

[39] T. Rock and R. Wille. Ein TOSCANA-Erkundungssystem zur Literatursuche. In Stumme and Wille [40], pages 239–253.

[40] G. Stumme and R. Wille, editors. *Begriffliche Wissensverarbeitung; Methoden und Anwendungen*, Heidelberg, 2000. Springer.

[41] S. Theodoridis and K. Koutroumbas. *Pattern Recognition.* Academic Press, third edition, 2006.

[42] F. J. Valverde-Albacete. Combining soft and hard techniques for the analysis of batch retrieval tasks. In E. Herrera-Viedma, G. Pasi, and F. Crestani, editors, *Soft Computing for Information Retrieval on the Web. Models and Applications*, volume 197 of *Studies in Fuzziness and Soft Computing*, pages 239–258. Springer, 2006. ISSN 1434-9922 (print edition). ISSN (electronic edition) 1860-0808, ISBN-10 3-540-31588-8, ISBN-13 978-3-540-31588-9.

[43] F. J. Valverde-Albacete and C. Peláez-Moreno. Towards a generalisation of formal concept analysis for data mining purposes. *Concept Lattices. Proceedings of the International Conference on Formal Concept Analysis (ICFCA 06)*, LNAI 3874:161—176, Dec 2006.

[44] F. J. Valverde-Albacete and C. Peláez-Moreno. Extending conceptualisation modes for generalised Formal Concept Analysis. *Information Sciences*, 181: 1888–1909, May 2011.

[45] C. van Rijsbergen. *The Geometry of Information Retrieval.* Cambridge University Press, Cambridge, New York,Melbourne, Madrid and Cape Town, 2004.

# A Markov Chain Approach to Random Generation of Formal Concepts

Dmitry V. Vinogradov[1,2]

[1] All-Russia Institute for Scientific and Technical Information (VINITI), Intelligent Information Systems Laboratory, Moscow 125190, Russia
vin@viniti.ru
[2] Russian State University for Humanities, Intelligent Robotics Laboratory, Moscow 125993, Russia
http://isdwiki.rsuh.ru/index.php/User:Vinogradov.Dmitry

**Abstract.** A Monte Carlo approach using Markov Chains for random generation of concepts of a finite context is proposed. An algorithm similar to CbO is used. We discuss three Markov chains: non-monotonic, monotonic, and coupling ones. The coupling algorithm terminates with probability 1. These algorithms can be used for solving various information retrieval tasks in large datasets.

**Keywords:** formal context, formal concept, Markov chain, termination with probability 1

## 1 Introduction

In many natural problems of information retrieval the piece of information which we are looking for is not contained in few documents. The query generates a huge amount of relevant documents and the task is to generates a cluster of related documents together with the set of common terms describing its common meaning. There are many choices for such cluster. So, the user has to look for plausible answers to his query.

JSM-method is a logical device to provide plausible reasoning for generation, verification and falsification of such clusters and for explanation of whole collection of all relevant documents by means of accepted clusters. The first variant of JSM method was presented by Prof. V.K. Finn in 1983 [1] (in Russian). FCA corresponds to the generation ("induction") step of JSM method. See [5] for details. This correspondence allows to use FCA algorithms in JSM-method and vice versa. For example, the well-known algorithm "Close-by-One" (CbO) was initially introduced by S.O. Kuznetsov in [4] for JSM-method and later translated into FCA framework. The state of art for JSM-method is represented in [2].

In our opinion, the main drawback of 'old-fashioned' JSM-method is the computational complexity of JSM algorithms, especially for the induction step. Paper [7] presents a results of comparison between various (partially improved by the survey's authors) variants of famous deterministic algorithms of FCA.

Paper [6] provides theoretical bounds on computational complexities of various JSM tasks.

The development of JSM-method has resulted in intelligent systems of JSM type that were applied in various domains such as sociology, pharmacology, medicine, information retrieval, etc. In practice there were situations when a JSM system generates more than 10,000 formal concepts (JSM similarities) from a context with about 100 objects. In our opinion the importance of all generated concepts is doubtful, because when experts manually select important JSM causes they reject majority of generated JSM similarities.

In this paper we propose Monte Carlo algorithms using Markov Chain approach for random generation of concepts of a finite context. In other words, we replace the lattice of all concepts by small number of its random elements.

## 2 Background

### 2.1 Basic definitions and facts of FCA

Here we recall some basic definitions and facts of Formal Concept Analysis (FCA) [3].

A **(finite) context** is a triple $(G, M, I)$ where $G$ and $M$ are finite sets and $I \subseteq G \times M$. The elements of $G$ and $M$ are called **objects** and **attributes**, respectively. As usual, we write $gIm$ instead of $\langle g, m\rangle \in I$ to denote that object $g$ has attribute $m$.

For $A \subseteq G$ and $B \subseteq M$, define

$$A' = \{m \in M | \forall g \in A(gIm)\}, \tag{1}$$

$$B' = \{g \in G | \forall m \in B(gIm)\}; \tag{2}$$

so $A'$ is the set of attributes common to all the objects in $A$ and $B'$ is the set of objects possesing all the attributes in $B$. The maps $(\cdot)' : A \mapsto A'$ and $(\cdot)' : B \mapsto B'$ are called **derivation operators** (**polars**) of the context $(G, M, I)$.

A **concept** of the context $(G, M, I)$ is defined to be a pair $(A, B)$, where $A \subseteq G$, $B \subseteq M$, $A' = B$, and $B' = A$. The first component $A$ of the concept $(A, B)$ is called the **extent** of the concept, and the second component $B$ is called its **intent**. The set of all concepts of the context $(G, M, I)$ is denoted by $\mathbf{B}(G, M, I)$.

*Example 1 (Boolean cube with n atoms).* Consider the context $(G, M, I)$, where $G = \{g_1, \ldots, g_n\}$, $M = \{m_1, \ldots, m_n\}$, and

$$g_j I m_k \Leftrightarrow j \neq k. \tag{3}$$

Then $(\{g_{j_1}, \ldots, g_{j_k}\})' = M \backslash \{m_{j_1}, \ldots, m_{j_k}\}$, $(\{m_{j_1}, \ldots, m_{j_k}\})' = G \backslash \{g_{j_1}, \ldots, g_{j_k}\}$, $A'' = A$ for all $A \subseteq G$, and $B'' = B$ for all $B \subseteq M$. Hence $\mathbf{B}(G, M, I)$ has element $(\{g_{j_1}, \ldots, g_{j_k}\}, M \setminus \{m_{j_1}, \ldots, m_{j_k}\})$ for every $\{g_{j_1}, \ldots, g_{j_k}\} \subseteq G$.

Let $(G, M, I)$ be a context. For concepts $(A_1, B_1)$ and $(A_2, B_2)$ in $\mathbf{B}(G, M, I)$ we write $(A_1, B_1) \leq (A_2, B_2)$, if $A_1 \subseteq A_2$. The relation $\leq$ is a **partial order** on $\mathbf{B}(G, M, I)$.

A subset $A \subseteq G$ is the extent of some concept if and only if $A'' = A$ in which case the unique concept of which $A$ is the extent is $(A, A')$. Similarly, a subset $B$ of $M$ is the intent of some concept if and only if $B'' = B$ and then the unique concept with intent $B$ is $(B', B)$.

It is easy to check that $A_1 \subseteq A_2$ implies $A_1' \supseteq A_2'$ and for concepts $(A_1, A_1')$ and $(A_2, A_2')$ reverse implication is valid too, because $A_1 = A_1'' \subseteq A_2'' = A_2$. Hence, for $(A_1, B_1)$ and $(A_2, B_2)$ in $\mathbf{B}(G, M, I)$

$$(A_1, B_1) \leq (A_2, B_2) \Leftrightarrow A_1 \subseteq A_2 \Leftrightarrow B_2 \subseteq B_1. \tag{4}$$

Fix a context $(G, M, I)$. In the following, let $J$ be an index set. We assume that $A_j \subseteq G$ and $B_j \subseteq M$, for all $j \in J$.

**Lemma 1.** *[3] Assume that $(G, M, I)$ is a context and let $A \subseteq G$, $B \subseteq M$ and $A_j \subseteq G$ and $B_j \subseteq M$, for all $j \in J$. Then*

$$A \subseteq A'' \quad \textit{and} \quad B \subseteq B'', \tag{5}$$

$$A_1 \subseteq A_2 \Rightarrow A_1' \supseteq A_2' \quad \textit{and} \quad B_1 \subseteq B_2 \Rightarrow B_1' \supseteq B_2', \tag{6}$$

$$A' = A''' \quad \textit{and} \quad B' = B''', \tag{7}$$

$$(\bigcup_{j \in J} A_j)' = \bigcap_{j \in J} A_j' \quad \textit{and} \quad (\bigcup_{j \in J} B_j)' = \bigcap_{j \in J} B_j', \tag{8}$$

$$A \subseteq B' \;\Leftrightarrow\; A' \supseteq B. \tag{9}$$

**Proposition 1.** *[3] Let $(G, M, I)$ be a context. Then $(\boldsymbol{B}(G, M, I), \leq)$ is a lattice with join and meet given by*

$$\bigvee_{j \in J} (A_j, B_j) = ((\bigcup_{j \in J} A_j)'', \bigcap_{j \in J} B_j), \tag{10}$$

$$\bigwedge_{j \in J} (A_j, B_j) = (\bigcap_{j \in J} A_j, (\bigcup_{j \in J} B_j)''); \tag{11}$$

**Corollary 1.** *For context $(G, M, I)$ the lattice $(\boldsymbol{B}(G, M, I), \leq)$ has $(M', M)$ as the bottom element and $(G, G')$ as the top element. In other words, for all $(A, B) \in \boldsymbol{B}(G, M, I)$ the following inequalities hold:*

$$(M', M) \leq (A, B) \leq (G, G'). \tag{12}$$

### 2.2 The "Close-by-One" operations: definition and properties

With the help of expressions for the infinum and the supremum operations in $\mathbf{B}(G, M, I)$ given by Proposition 1 we can introduce local steps of our Markov chains:

**Definition 1.** *For $(A, B) \in \boldsymbol{B}(G, M, I)$, $g \in G$, and $m \in M$ define*

$$CbO((A, B), g) = ((A \cup \{g\})'', B \cap \{g\}'), \tag{13}$$

$$CbO((A, B), m) = (A \cap \{m\}', (B \cup \{m\})''). \tag{14}$$

*so $CbO((A, B), g)$ is equal to $(A, B) \vee (\{g\}'', \{g\}')$ and $CbO((A, B), m)$ is equal to $(A, B) \wedge (\{m\}', \{m\}'')$.*

We call these operations CbO because the first one is used in Close-by-One (CbO) Algorithm to generate all the elements of $\mathbf{B}(G, M, I)$, see [4] for details.

**Lemma 2.** *Assume that $(G, M, I)$ is a context and let $(A, B) \in \boldsymbol{B}(G, M, I)$, $g \in G$, and $m \in M$. Then*

$$g \in A \Rightarrow CbO((A, B), g) = (A, B), \tag{15}$$

$$m \in B \Rightarrow CbO((A, B), m) = (A, B), \tag{16}$$

$$g \notin A \Rightarrow (A, B) < CbO((A, B), g), \tag{17}$$

$$m \notin B \Rightarrow CbO((A, B), m) < (A, B). \tag{18}$$

*Proof.* If $g \notin A$ then $A \subset A \cup \{g\} \subseteq (A \cup \{g\})''$ by (5). By definition of the order between concepts this inclusion and (13) imply (17). Relation (18) is proved in the same way, the rest is obvious.

**Lemma 3.** *Assume that $(G, M, I)$ is a context and let $(A_1, B_1), (A_2, B_2) \in \boldsymbol{B}(G, M, I)$, $g \in G$, and $m \in M$. Then*

$$(A_1, B_1) \leq (A_2, B_2) \Rightarrow CbO((A_1, B_1), g) \leq CbO((A_2, B_2), g), \tag{19}$$

$$(A_1, B_1) \leq (A_2, B_2) \Rightarrow CbO((A_1, B_1), m) \leq CbO((A_2, B_2), m). \tag{20}$$

*Proof.* If $A_1 \subseteq A_2$ then $A_1 \cup \{g\} \subseteq A_2 \cup \{g\}$. Hence (6) implies $(A_2 \cup \{g\})' \subseteq (A_1 \cup \{g\})'$. Second part of (6) implies $(A_1 \cup \{g\})'' \subseteq (A_2 \cup \{g\})''$. By definition of the order between concepts this is (19). Relation (20) is proved in the same way by using (4).

## 3 Markov Chain Algorithms

Now we represent Markov chain algorithms for random generation of formal concepts.

**Data**: context $(G, M, I)$, external function $CbO(\ ,\ )$
**Result**: random concept $(A, B) \in \mathbf{B}(G, M, I)$
$t := 0$; $(A, B) := (M', M)$;
**while** $(t < T)$ **do**
  select random element $x \in (G \setminus A) \sqcup (M \setminus B)$;
  $(A, B) := CbO((A, B), x)$;
  $t := t + 1$;
**end**

**Algorithm 1:** Non-monotonic Markov chain

*Example 2 (Random walk on Boolean cube).* Consider the context $(G, M, I)$ of Example 1. Then Non-monotonic Markov chain corresponds to Random Walk on Boolean Cube of all the concepts in $\mathbf{B}(G, M, I)$.

**Definition 2.** *An **(order) ideal** of partially odered set (poset) $(S, \leq)$ is a subset $J$ of $S$ such that*

$$\forall s \in S \ \forall r \in J[s \leq r \Rightarrow s \in J]. \tag{21}$$

*A Markov chain $S_t$ with values into poset $(S, \leq)$ is called **monotonic** if for every pair of start states $a \leq b$ $(a, b \in S)$ and every order ideal $J \subseteq S$*

$$\mathrm{P}[S_1 \in J | S_0 = a] \geq \mathrm{P}[S_1 \in J | S_0 = b]. \tag{22}$$

**Proposition 2.** *There exists the context $(G, M, I)$ such that Non-monotonic Markov chain for $(\boldsymbol{B}(G, M, I), \leq)$ isn't monotonic one.*

See [8] for the proof of Proposition 2. The following Markov chain is always monotonic one.

**Data**: context $(G, M, I)$, external function $CbO(\ ,\ )$
**Result**: random concept $(A_T, B_T) \in \mathbf{B}(G, M, I)$
$t := 0$; $X := G \sqcup M$; $(A, B) := (M', M)$;
**while** $(t < T)$ **do**
  select random element $x \in X$;
  $(A, B) := CbO((A, B), x)$;
  $t := t + 1$;
**end**

**Algorithm 2:** Monotonic Markov chain

**Proposition 3.** *For every context $(G, M, I)$ the Monotonic Markov chain for $(\boldsymbol{B}(G, M, I), \leq)$ is monotonic one.*

The proof of Proposition 3 can be found in [8].The Monotonic Markov chain algorithm has another advantage: random selection of elements of the static set $X := G \sqcup M$. However both previous algorithms have common drawback: the unknown value $T$ for the termination time of the calculation. In the Monte Carlo Markov Chain (MCMC) theory this value corresponds to the **mixing time** of the chain. For some special Markov chains (for instance, for the chain of Example 2) the mixing time is estimated by sophisticated methods. In general case, it is an open problem. The following algorithm has not this problem at all.

**Data**: context $(G, M, I)$, external function $CbO(\ ,\ )$
**Result**: random concept $(A, B) \in \mathbf{B}(G, M, I)$
$X := G \sqcup M$; $(A, B) := (M', M)$; $(C, D) = (G, G')$;
**while** $((A \neq C) \vee (B \neq D))$ **do**
  select random element $x \in X$;
  $(A, B) := CbO((A, B), x)$; $\quad (C, D) := CbO((C, D), x)$;
**end**

**Algorithm 3:** Coupling Markov chain

Intermediate value of quadruple $(A, B) \leq (C, D)$ on step $t$ corresponds to Markov chain state $Y_t$. The $A_t$, $B_t$, $C_t$ and $D_t$ are first, second, third, and fourth components, respectively.

**Definition 3.** *A **coupling length** for context $(G, M, I)$ is defined by*

$$L = \min(\mid G \mid, \mid M \mid). \tag{23}$$

*A **choice probability** of fixed object or attribute in context $(G, M, I)$ is equal to*

$$p = \frac{1}{\mid G \mid + \mid M \mid}. \tag{24}$$

**Lemma 4.** *If $\mid G \mid < \mid M \mid$ then for every integer $r$ and every pair of start states $(A, B) \leq (C, D)$ $((A, B), (C, D) \in \boldsymbol{B}(G, M, I))$*

$$\mathrm{P}[A_r = A_{r+L} = C_{r+L} \& B_r = B_{r+L} = D_{r+L} | Y_r = (A, B) \leq (C, D)] \geq p^L. \tag{25}$$

*If $\mid G \mid \geq \mid M \mid$ then for every integer $r$ and every pair of start states $(A, B) \leq (C, D)$ $((A, B), (C, D) \in \boldsymbol{B}(G, M, I))$*

$$\mathrm{P}[A_{r+L} = C_{r+L} = C_r \& B_{r+L} = D_{r+L} = D_r | Y_r = (A, B) \leq (C, D)] \geq p^L. \tag{26}$$

*Proof.* For coupling to $(A, B) \leq (A, B)$ it suffices to get an element from $A \setminus C$, but $\mid A \setminus C \mid \leq \mid G \mid = L$. Relation (26) is proved in a similar way. □

**Theorem 1.** *The coupling Markov chain has the probability of coupling (termination) before $n$ steps with limit 1 when $n \to \infty$.*

*Proof.* Lemma 3 implies that $(A_{(k-1)\cdot L}, B_{(k-1)\cdot L}) \leq (C_{(k-1)\cdot L}, D_{(k-1)\cdot L})$. Let $r = (k-1) \cdot L$. Then Lemma 4 implies that $\mathrm{P}[A_{k\cdot L} \neq C_{k\cdot L} \vee B_{k\cdot L} \neq D_{k\cdot L} | Y_r = (A_r, B_r) \leq (C_r, D_r)] \leq (1 - p^L)$. After $k$ independent repetitions we have $\mathrm{P}[A_{k\cdot L} \neq C_{k\cdot L} \vee B_{k\cdot L} \neq D_{k\cdot L} | Y_0 = (M', M) \leq (G, G')] \leq (1 - p^L)^k$. But when $k \to \infty$ we have $(1 - p^L)^k \to 0$. □

## Conclusions

In this paper we have described a Monte Carlo approach using Markov Chains for random generation of concepts of a finite context. The basic steps of proposed Markov chains are similar to ones of algorithm CbO. We discuss three Markov chains: non-monotonic, monotonic, and coupling ones. The coupling algorithm terminates with probability 1.

## Acknowledgements.

The author would like to thank Prof. Victor K. Finn and Tatyana A. Volkova for helpful discussions. The research was supported by Russian Foundation for Basic Research (project 11-07-00618a) and Presidium of the Russian Academy of Science (Fundamental Research Program 2012-2013).

## References

1. V.K. Finn, About Machine-Oriented Formalization of Plausible Reasonings in F. Beckon-J.S. Mill Style. Semiotika I Informatika, 20, 35–101 (1983)
2. V.K. Finn, The Synthesis of Cognitive Procedures and the Problem of Induction. Autom. Doc. Math. Linguist., 43, 149–195 (2009)
3. B. Ganter, R. Wille, Formal Concept Analysis: Mathematical Foundations, Springer-Verlag, 1999
4. S.O. Kuznetsov, A Fast Algorithm for Computing All Intersections of Objects in a Finite Semi-Lattice, Automatic Documentation and Mathematical Linguistics, vol.27, no.5, 11-21, 1993.
5. S.O. Kuznetsov, Mathematical aspects of concept analysis. Journal of Mathematical Science, Vol. 80, Issue 2, pp. 1654-1698, 1996.
6. S.O. Kuznetsov, Complexity of Learning in Concept Lattices from Positive and Negative Examples. Discrete Applied Mathematics, 2004, no. 142(1-3), pp. 111-125.
7. S.O. Kuznetsov and S.A. Obiedkov, Comparing Performance of Algorithms for Generating Concept Lattices. Journal of Experimental and Theoretical Artificial Intelligence, Vol. 14, no. 2-3, pp. 189-216, 2002.
8. D.V. Vinogradov, Random generation of hypotheses in the JSM method using simple Markov chains. Autom. Doc. Math. Linguist., 46, 221–228 (2012)

# Situation Assessment Using Results of Objects Parameters Measurements Analyses in IGIS

Andrey Pankin[1], Alexander Vodyaho[2], Nataly Zhukova[1]

[1]Saint-Petersburg Institute for Informatics and Automation of the Russian Academy of Sciences, Research laboratory of object-oriented geo-information systems, Russia

[2]Saint-Petersburg Electrotechnical University, Russia

aivodyaho@mail.ru, {pankin, gna}@oogis.ru

**Abstract.** The paper describes method for situations assessment based on retrieving information about similar earlier observed conditions. Situation is a set of qualitative and quantitative characteristics that describe states of interrelated objects. Object states are defined by measurement parameters. A method for situation assessment is based on calculation of aggregated indices and their comparison was developed. For calculating aggregated indices it is proposed to use an algorithm for alphabetic description of time series that provide convenient means for their comparison. For situations retrieval it is suggested to use FCA methods. As a case study the results of ocean data analyses for calculating temperature and salinity parameters of water area are presented.

**Keywords:** situation assessment, measurements analyses, summary indicators

## 1 Introduction

Nowadays Intelligent Geographic Information Systems (IGIS) are widely used for solving different functional tasks. IGIS incorporates GIS interface as well as various methods of artificial intelligence intended for solving certain intricate problems including problem of decision making support. Decision support systems in IGIS are aimed to provide end users with complex information about the solved problem as well as with reasonable alternative decisions in real time with a pictorial rendition of this information to let it be easily perceived and used.

One of the important tasks, that is solved in decision making support systems, is situation assessment and awareness. Situation assesment is aimed to make situations understandable by users. Situation awareness is the perception of the elements in the environment within a volume of time and space, the comprehension of their meaning, and the projection of their status in the near future [1]. Situation assessment represents analysis of available information in order to get validated estimations of current system state and probable direction and dynamic of its changing. In this context term "system" can have wide interpretation: it can be used to describe dynamic technical or

environmental objects, set of interacting objects, analyzed phenomena, entities, or environments.

Problem of situation assessment can be decomposed into two subtasks. The first task is situation recognition and the second is making decision about system state. For situation recognition an approach based on comparing situations to ones that were earlier observed is widely used. To provide this knowledge base of situations a description is formed. Decision about system state is based on knowledge about recognized situations. Various directions of situation development can be considered using modeling tools or expert systems.

By now one of the key means for storing information about systems actual states are measurements instruments that provide information about system parameters in real time. Using these measurements ability to analyze the dynamic of a state evaluation and control a system state is provided. There are several problems related with measurements processing. First problem is great volume of data that has to be processed in limited time. Second problem is that measurements have rather bad quality; they are not coordinated in time and space and are implemented as non-stationary time series. Consequently, highly specialized methods have to be used for measurements processing. Third problem is a necessity to represent measurements as a set of complex characteristics, so that they can be used in methods of situation assessment.

In the paper an approach to situation assessment based on comparison of situations is extended for using measurements of system parameters as one of important information sources and approach to retrieval situations using formal concept analyses (FCA) methods is proposed. In the second section general description of the method for situation assessment based on measures analyses in presented. Following sections provide detailed description of algorithms used in the general method. An algorithm of alphabetic representation of time series given in section 3 is aimed to represent time series in a form that provides easy mechanism for comparing parameters. In section 4 the algorithm of identification of information valuable parameters that allow ranging parameters according to information values is considered. In section 5 algorithm for objects aggregated indicators calculating that takes into account values of parameters measurements is presented. Algorithms for building and comparing graphs that describe situations in terms of objects and their relations are discussed in section 6. In section 7 application of Formal Concept Analysis methods for revealing earlier observed distinguishable situations are considered. As a case study task of ocean parameters estimation using measurements provided by floating hydrographic buoys is described.

## 2 Main definitions

FCA is a well-established technique in mathematics that is widely used for solving various tasks of intelligent data analyses. Standard FCA definitions are introduced in [2, 3]. Given a formal context $K = (G, M, I)$, where $G$ is called a set of objects, M is called a set of attributes, and the binary relation $I \subseteq G \times M$ specifies which ob-

jects have which attribute, the derivation operators $(\cdot)^I$ are defined for $A \subseteq G$ and $B \subseteq M$ as follows:

$$A^I = \{m \in M \mid \forall g \in A : g \text{ Im}\};$$
$$B^I = \{m \in M \mid \forall m \in B : g \text{ Im}\}.$$

$A^I$ is the set of attributes common to all objects of $A$ and $B^I$ is the set of objects that share attributes of $B$. For simplicity operator $(\cdot)'$ is used instead of $(\cdot)^I$. The double application of $(\cdot)'$ is a closure operator; it is extensive, idempotent, and monotonous. Therefore, sets $A''$ and $B''$ are closed sets.

A formal concept of the context $(G, M, I)$ is a pair $(A, B)$, where $(A \subseteq G)$, $(B \subseteq M)$, $A = B'$, and $B = A'$. In this case $A = A''$ and $B = B''$. The set $A$ is called the extent and $B$ is called the intent of the concept $(A, B)$. In categorical terms a formal concept is defined by its objects $A$ or its attributes $B$.

A concept $(A, B)$ is a subconcept of $(C, D)$ and $(C, D)$ is a superconcept of $(A, B)$ if $(A \subseteq C)$ (equivalently, $(D \subseteq B)$). For $(A, B)$ and $(C, D)$ relations $\geq$, $\leq$, $<$, and $>$ are defined and written as usual. $(A, B)$ is a lower neighbor of $(C, D)$ (notation is (A, B) $\prec (C, D)$) and $(C, D)$ is an upper neighbor of (A, B) (notation is $(C, D) \succ (A, B)$) if $(A, B) < (C, D)$ and there is no $(E, F) : (A, B) < (E, F) < (C, D)$. The set of all concepts ordered by $<$ forms a concept lattice of the context K, that is denoted by $B(K)$. The relation $\prec$ defines edges in the covering graph of $B(K)$.

For building lattices while solving task of situations analyses formal context as a set of objects $G$ situations are considered, $M$ is a set of situations characteristics, $I$ is an incidence relation between these sets. Each situation s is characterized by a set of relevant objects $E = \{O_i\}_{i=1}^N$ and relations between objects $R = \{r_{i,j}\}_{i,j=1}^N$, where $N$ is a total number of objects. For each object a set of parameters $e = \{P_i\}_{i=1}^M$ that describes objects state is defined.

## 3 General description of method for situation assessment

Situation assessment is based on comparing current conditions with the previously observed ones. Situation involves objects that can be technical or natural and relation between them. Relations are described for pairs of objects, for each relation its type is defined. All types are to be described a priori in a vocabulary of subject domain. An object state is characterized with a set of parameters; values of parameters are measured using various measurement instruments and are represented as time series.

When solving problem of situation assessment it is necessary to provide an effective and efficient mechanism for situation comparing to retrieve similar ones. For

comparing two situations it is necessary to compare list of objects and their states and relations between objects. Objects and relations between objects are reasonable to represent as a graph, where vertexes of the graph are objects and edges of the graph are relations.

For comparing two graphs a wide range of methods is developed. The most convenient algorithm for similar situations retrieval is based on graph edit distance. The main idea of this algorithm is to define difference between graphs using a set of editing operations that are necessary for transforming one graph to the other. This method is tolerant to errors and provides inexact graph matching. Algorithms for graph edit distance calculation are described in [4]. When edges of graph are compared the result is binary – if the relations that corresponds to the edges are equal then result of comparison is '1' else the result is '0'. For comparing vertexes it is necessary to compare objects associated with them. As each object is characterized with a set of parameters to build object description it is necessary to solve two problems –to describe each time series of parameters measurements in such a way that descriptions can be easily compared and to define how to calculate aggregate characteristic of objects using formed descriptions.

For describing parameters measurements it is proposed to use alphabetic representation of time series. To build alphabetic representation method based on Symbolic Aggregate Approximation (SAX) [5] is used. Strings that are composed with SAX-based algorithms can be compared using string Edit Distance that is used in algorithms of string inexact comparing [6].

For solving the second task Aggregated Indices Randomization Method (AIRM) [7] can be applied that is targeting complex objects subjected to multi-criteria estimation under uncertainty. The essence of application of AIRM consists in an aggregation of single characteristics into one complex characteristic that is used for comparing objects. One of the key tasks that is to be solved before ARIM method can be applied is to define weights for objects parameters that are considered as indicators. Taking into account that parameters are characterized with measurement time series for evaluation of time series information value a set of statistical characteristics is used [8].

General description of proposed method for situation assessment is given in Fig.1

---

**Input data**. Data base of graphs describing earlier observed situations, description of estimated situation, that includes $E = \{O_i\}_{i=1}^{N}$ is a set of objects, $R = \{r_{i,j}\}_{i,j=1}^{N}$ is a set of objects relations, $O = \{P_i\}_{i=1}^{M}$ is set of objects parameters, $P = \{(t_i, x_i)\}_{i=1}^{H}$ is a time series of parameters measurements, where $H$ is a total number of measurements.

**Output data**. $S = \{(s_i, q_i)\}_{i=1}^{T}$ is a set of situations $s$ that are similar to a defined situation $s^d$ with similarity degree $q$.

**Algorithm description**

*A. Building description of estimated situation*

*Step A1* **build** symbolic representation of objects parameters measurements $\hat{C} = f_{symb}(P)$

*Step A2* Building descriptions of objects

**calculate** weights of parameters $W = \{w_i\}_{i=}^{M}$ according to information value

**calculate** estimations of aggregated indices for objects $\tilde{Q} = \{\tilde{Q}_i\}_{i=1}^{N}$

*B. Building graph for situation description*

*Step B1* Defining graph vertexes $G_V$ using formalized descriptions of objects

*Step B2* Defining graph edges $G_D$ using formalized descriptions of objects relations

*C. Situation estimation*

*Step C1*. Reveling similar graphs of situation description in data base

*Step C2*. Ranging graphs according to degree of similarity $S = \{(s_i, q_i)\}_{i=1}^{T}$

Fig.1 General description of method for situation assessment

## 4 Algorithm of alphabetic representation of time series

Proposed algorithm of alphabetic representation is based on algorithm of Symbolic Aggregate Approximation (SAX) described in [5]. In SAX for building symbolic representation of time series approach based on application of Piecewise Aggregate Approximation (PAA) is used. According to the algorithm time series are presented as a sequence of segments using window of defined length. For each segment a set of defined statistical characteristics are estimated. PAA can be considered as an attempt to represent a time series in a form of windows line combination. The description of the algorithm is given in Fig. 2. PAA representation of time series is converted into symbolic representation. In SAX it is assumed that analyzed time series have normal distribution, but measurements time series very often doesn't satisfy this criterion. In [9] the description of modification of SAX for time series with various distributions is proposed. The modified procedure assumes, at first, estimation of measurements values interval. To avoid usage of values that contain noise and outliers for determining border median values of $K$, minimum and maximum values are used. Second, interval of values are split into equal intervals, each part corresponds to one level. Segments, which characteristics correspond to one interval, are the segments of one level and they are described using same symbol from a priori defined alphabet (Fig. 3).

**Input data**. $P = p_1, ..., p_H$ is an initial time series, where $H$ is a number of segments.

**Output data**. $\overline{C} = \bar{c}_1, ..., \bar{c}_z$ is aPAA representation of time series.

**Algorithm description**

*Step 1*. **calculate** length of one segment $l = \frac{H}{z}$

*Step 2* **for** ( $i = 1 ... z$ )

$\bar{c}_i = f(\{c_j\}_{j=l(j-1)+1}^{l-i})$ , where $f$ is a function of calculating segment statistical characteristics

Fig.2 Algorithm for building PAA representation of time series

**Input data**. $P = p_1, ..., p_H$ is an initial time series, where $H$ is a number of segments, $A = a_1, ..., a_k$ is an alphabet for time series symbolic representation, $B = \beta_1, ..., \beta_{k-1}$ are levels of time series representation.

**Output data**. $\hat{C} = \hat{c}_1, ..., \hat{c}_z$ is asymbolic representation of time series.

**Algorithm description**

*Step 1.* **calculate** $\overline{C}$ using algorithm for PAA representation of time series

*Step 2.* **calculate** range of time series characteristic values [ $V_l$ , $V_h$ ], where $V_l$ - low border, $V_h$ - high border

*Step 3.* **calculate** range of characteristics values for each level $\beta$

*Step 4.* **for (** $i = 1 ... w$ **)**

**define** alphabet symbol $\hat{c}_i = a_j \Leftrightarrow \beta_{j-1} \le \overline{c}_j < \beta_j$

*Step 5.* **concatenate** symbols $\hat{C} = \{\hat{c}_i\}_{i=1}^{z}$

Fig.3 Algorithm for building time series symbolic representation

By now many algorithms that allow to deal with strings, in particular, algorithms of inexact string comparison, based on calculation of Edit Distance are developed. Algorithms of string comparison are applied for qualitative evaluation of time series similarity.

## 5 Algorithm of information valuable parameters identification

Each object is described by a set of various parameters. Degree of information value of each parameter differs and it is necessary to take it into account when two objects are compared. The degree of parameter information value is used to range parameters in algorithm of calculating objects aggregated indices.

The proposed algorithm of calculating degree of parameter information value is based on using a set of statistical characteristics. Depending on objects characteristics different measures for time series described in [8] can be calculated. Most often the following measures are used: mean, median, variance, standard deviation, interquartile distance, skewness and kurtosis. The algorithm of ranging parameters is based on the idea that most informative are measures that have maximum difference for different objects. So mean distances between measures of parameters time series are calculated and according to them parameters are ranged and preliminary weight coefficients are defined. The proposed algorithm is given in Fig. 4.

**Input data.** $E = \{O_i\}_{i=1}^{N}$ is set of objects, $O = \{P_i\}_{i=1}^{M}$ is a set of measured objects parameters, where $M$ is a total number of objects parameters, $G = \{g_i\}_{i=1}^{U}$ is a list of time series measures, $U$ is a total number of measures.

**Output data.** $\overline{P}$ is a sorted set of parameters, $W = \{w_i\}_{i=1}^{M}$ is a list of preliminary weight coefficients for parameters.

**Algorithm description**

*A. Calculating measures for parameters time series*

*Step A1*. **for each** parameter $(i = 1 \ldots M)$

**for each** object $(j = 1 \ldots N)$

**calculate** measures $s_{ij} = (s_{ij}^1, \ldots, s_{ij}^U)$

**calculate** mean distance $\bar{s}_i = \frac{1}{N} \sum_{k=1}^{M} \sum_{l=1}^{M} \sqrt{(s_{ik} - s_{il})^2}$

*B. Ranging parameters*

*Step B1* **for** $(i = 1 \ldots M)$

**define** preliminary weights $w_i = \bar{s}_i$

*Step B2* **sort** parameters according to preliminary weights $\overline{P} = sort(\{P_i\}_{i=1}^{M})$

Fig.4 Algorithm for ranging parameters

## 6 Algorithm for objects aggregated indicators calculating

To calculate objects aggregated indicators based on set of parameters it is proposed to use indices randomization method. ARIM is used to solve tasks of multiple criteria decision making on the base of poor-quality input information. The main advantage of AIRM is its ability to cope with non-numeric (ordinal), non-exact (interval) and non-complete information. When solving user's tasks information about objects parameters is often incomplete as parameters due to different reasons can't be gathered. Calculated in section 5 preliminary weights of parameters provide approximate estimation of parameters information value and therefor can't be used directly for calculating objects aggregated indicators. Preliminary parameters weights are used to range parameters and thus provide ordinal information about parameters. This information can be effectively used in AIRM.

In ARIM three key steps are executed: i) building vector of single indicators; ii) defining aggregative function; iii) defining weighs coefficients.

Main features of ARIM application for calculating objects indicators using measurements are the following:

1. Results of symbolic representation of time series of parameters measurements build according to algorithm described in section 3 are considered as list of objects characteristics.

2. Single indicators for objects are functions of objects characteristics. They are defined as normalizing power functions of degree one. When characteristic values increase functions also increase.

3. An aggregative indicator is a synthesized function that characterizes each object in general. It depends on weight coefficients and is represented in a form of linear convolution of single indicators functions and weight coefficients.

4. As information $I$ about objects parameters weights is incomplete, weight-vector $w = (w_1, \ldots, w_m)$ is ambiguously determined. In ARIM this vector is determined with accuracy to within a set $w(I)$ of all admissible weight-vectors. An uncer-

tain choice of a weight-vector from set $w(I)$ is modeled by a random choice of an element of the set according to the concept of Bayesian randomization. Such randomization produces a random weight-vector $w(I) = (w_1(I), \ldots, w_m(I))$, which is uniformly distributed on the set $w(I)$. Set $w(I)$ is reduced using ordinal and interval information. Mathematical expectation of random weight coefficient $w_i(I)$ may be used as a numerical estimation of particular indicator $q_i$ significance. Then randomized weight-vector can be defined as $\tilde{w}(I) = (\tilde{w}_1(I), \ldots, \tilde{w}_M(I))$. The precision of this estimation is measured by standard deviation of the corresponding random variable.

The algorithm for objects summary indicators calculating is given in Fig.5.

---

**Input data**. $E = \{O_i\}_{i=1}^{N}$ is a set of objects, $O = \{\hat{C}_i\}_{i=1}^{M}$ is a set of symbolic representation of measured objects parameters, where $M$ is a total number of objects parameters, $\hat{C} = \{\hat{c}_i\}_{i=1}^{z}$ is a symbolic representation of parameter, where $z$ is a length of symbolic representation.

**Output data**. $\tilde{Q} = \{\tilde{Q}_i\}_{i=1}^{N}$ are estimations of objects aggregated indicators.

**Algorithm description**

*Step 1*. **for each** object $(1, \ldots, N)$

**define** $\hat{C} = (\hat{c}_1, \ldots, \hat{c}_M)$ as set of initial characteristics

**calculate** vector of single indicators $q = (q_1, \ldots, q_M)$,

$$q_j = q_j(\hat{c}_j) = \begin{cases} 0, & \hat{c}_j \le MIN_j, \\ \left( \dfrac{\hat{c}_j - MIN_j}{MAX_j - MIN_j} \right), & MIN_j < \hat{c}_j \le MAX_j, \\ 1, & \hat{c}_j > MAX_j; \end{cases}$$

$MIN_j$ and $MAX_j$ - minimum and maximum values of characteristic

**calculate** randomized weight-vector for characteristics $\tilde{w}_i = (\tilde{w}_1, \ldots, \tilde{w}_M)$

**calculate** aggregated indicator

$$\tilde{Q}(q; I) = Q(q; \tilde{w}(I)) = Q(q_1, \ldots, q_m; \tilde{w}_1(I), \ldots, \tilde{w}_m(I)) = \sum_{i=1}^{m} q_i \tilde{w}_i(I)$$

**calculate** estimation of aggregated indicator

$\overline{Q}(I) = E\tilde{Q}(I)$

*Step 2*. **form** vector of aggregated indicators estimations $\tilde{Q} = (\tilde{Q}_1, \ldots, \tilde{Q}_N)$

---

Fig.5 Algorithm for objects aggregated indicators calculating

## 7 Algorithms for building and comparing situation graphs

A situation graph contains information about objects, a set of characteristic that are sufficient for objects description, and relations between objects. Building a situation graph assumes following main steps: i) making a list of objects that are significant for situation description; ii) defining set of objects characteristics; iii) defining

set of admissible relations between objects; iv) building structure of the graph. All tasks are enumerated but the last one is solved by experts manually. A set of objects characteristics contains aggregated characteristics of measured parameters that are defined in section 5 and it may also contain one or several additional characteristics. Usually, as additional characteristics, time and earth coordinates of parameters measurements are considered. The algorithm for building situation graph is given in Fig. 6.

---

**Input data**. $E = \{O_i\}_{i=1}^{N}$ is a set of objects, $R = \{r_{i,j}\}_{i,j=1}^{N}$ is a set of objects relations, $F = \{f_i\}_{i=1}^{Y}$ is a set of object characteristics, where $Y$ is a total number of object characteristics.

**Output data**. $G = < G_V, G_D >$ is a situation graph, $G_V$ are graph vertexes and $G_D$ aregraph edges.

**Algorithm description**

*Step 1*. **define** empty graph $G_V \leftarrow [\,],\ G_D \leftarrow [\,]$

*Step 2*. **create** vertexes from objects $G_V \leftarrow E$

*Step 3*. **create** edges for related objects $G_D \leftarrow R$

*Step 4* **for each** vertex $v_i \in G_V \quad (i = 1, ..., N)$

**define** attributes $A_{vi} = a_v(O_i)$ according to characteristics of object $O_i$

*Step 5*. **for each** edge $d_{i,j} = d(O_i, O_j)$, $i = 1, ..., N$, $j = 1, ..., N$

if ( $d_i$ **exists**)

**define** attributes $A_{di} = a_d(r_{i,j})$ according to defined relation between objects $O_i, O_j$

---

Fig.6 Algorithm for building situation graphs

Widely used methods for comparing graphs are based on calculation of graph edit distance [1, 4]. The main idea of these methods is to find minimum number of graph editing operations (edit path) that will allow the transformation one compared graph to another. Edit distance $d$ for graphs $G_1$ and $G_2$ can be defined as: $d(G_1, G_2) = \min\limits_{(e_1, ..., e_k) \in Y(G_1, G_2)} \sum_{i=1}^{k} c(e_i)$, where Y are all possible edit paths, $e$ is a graph editing operation, $c(e)$ is the cost of operation $e$. The key advantage of these methods is their flexibility as methods are able to deal with any graphs and any types of vertex and edge attributes. The standard set of graphs operations include following operations: adding, removing and modifying elements.

The described group of methods allows finding optimal solution, but is complicated from computational point of view. Due to this fact if situation description contains considerable number of object and relations, it is proposed to use suboptimal methods for graph comparison [10, 11]. According to these methods graph is decomposed into a set of sub graphs. Each sub graph contains one vertex and edges that are related to the vertex. The task of comparing two graphs is substituted by the task of comparing sets of sub graphs.

The alternative approach for suboptimal graph comparing is based on using Hungarian method [12, 13]. It assumes searching optimal matching of vertexes and their local structure using approximation of graph Edit Distance.

In case if a priori knowledge about objects and their relations for different types of situations is available, complexity of comparing graphs methods can be significantly reduced.

## 8 Algorithm for revealing situations using FCA

The approach for situations assessment based on building and comparing graphs supposes that a data base of situations is created a priori. The task of creation of a universal mechanism for distinguishable situations retrieval is highly complicated as situations are often rather similar; they have a number of equal characteristics, relations and involved objects. Since there are many situations and each situation is described by huge volume of heterogeneous data it is proposed to use Formal Concept Analysis methods [2] for revealing equal and different features of situations, interconnected situations, and groups of similar situations.

To build lattices formal context $K$ is defined using a set of defined situations and their characteristics. Characteristics can be binary, quantitative or qualitative. Binary characteristics can be used directly for building a context. Qualitative characteristics can be considered as a set of adjusted characteristics, where each of characteristics values correspond to one adjusted characteristic. For representation of quantitative characteristics in binary form nominal scales can be used. This approach is rather flexible as it allows user to modify scales manually. It is also possible to build lattices using multivalued contexts that are defined as $K = (G, M, W, I)$, where $W$ is a set of situations characteristics values, $I$ is a ternary relation, $I \subset G \times M \times W \times I$, where process of scaling is automated. Approaches for building lattices using multivalued contexts are described in [14, 15].

The algorithm for revealing situations using FCA supposes executing of three main stages. The first stage assumes building formal context for representation situations and their characteristics. As objects of formal context a preliminary list of situations defined by experts is used. A list of context features contains set of three characteristics for each involved subject domain object. A set of used characteristics is equal to the set that is used for building graphs. Each object is characterized by i) its name or id, ii) its location in space and, if necessary, in time and iii) aggregated indicators. All characteristics are represented in binary form. For building nominal scale for aggregated indicators, ranges for values are defined using entropy based methods, in particular, Gini [16] evaluation measure. At the second stage FCA methods are applied to build concept lattice [17]. At the third stage formal concepts are analyzed by experts that modify the preliminary list of situations and, in separate cases, the list of features using obtained results. The algorithm for revealing situation using FCA is given in Fig. 7.

**Input data.** $E = \{O_i\}_{i=1}^{N}$ isa set of objects, $F = \{f_i\}_{i=1}^{Y}$ is a set of object characteristics, where $Y$ is a total number of object characteristics.

**Output data**. $S = \{s_i\}_{i=1}^{K}$ is a set of situations, where $K$ is a number of revealed situations.

**Algorithm description**

*Step 1* **define** preliminary list of situations $S = \{s_i\}_{i=1}^{K}$

*Step 2* **calculate** characteristics of situation

**for each** situation $s_i \in S$

**for each** object $e_j \in E$ involved in $s_i$

**calculate** object characteristics $F_{ij}$

**convert** characteristics to binary form $F_{ij} \to F_{ij}^{B}$

*Step 3* **build** formal context $K$

**define** formal objects $G \leftarrow S$

**define** formal objects features $M \leftarrow \{E, F^{B}\}$

**define** relations $I$

*Step 4* **build** lattice

*Step 5* **improve** set of situations $S$

Fig.7 Algorithm for revealing situations using FCA

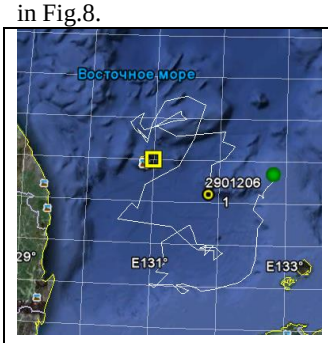
## 9 Case study

The proposed approach for situation assessment was used for solving task of providing operational information about ocean temperature and salinity parameters for hydroacoustics calculations that use sound speed of water area as one of parameters. Regular grids of parameters values are usually used as a source for information about water area state. Performing processing and analysis of available oceanographic data in order to build regular data grids includes two main steps: data verification and data regularization. The main purpose of data verification step is systematic storage, analysis and processing of data in order to prepare it for solving problem of building data grids [18, 19]. The main objective of regularization stage is to build a regular grid using methods of objective analyses and estimate the accuracy of gridded data [20]. Regular grids are usually updated and provided to end-users twice a year. It is possible to organize grid recalculation each time new measurements are acquired in systems that include components for oceanographic data processing. Algorithms for grids recalculation assumes that the whole grid is processed. The recalculation takes much time, besides new data is processed equally to historical data, though it is much more important for estimation of actual water area parameters.

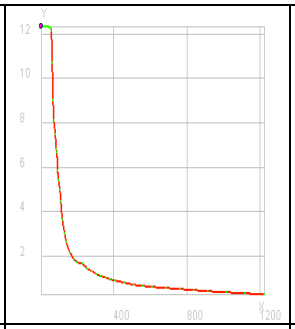
The experiments on operational estimation of water area parameters were made using measurements received from Argo float drifts [21]. The objective of Argo program is to operate and manage a set of floats distributed in all oceans. An Argo float drifts for a number of years in the ocean. It continuously performs measurement cycles. Each cycle lasts about 10 days and can be divided into 4 phases: a descent from

surface to a defined pressure (e.g., 1500 decibars), a subsurface drift (e.g., 10 days), an ascending profile with measurements (e.g., pressure, temperature, salinity), a surface drift with data transmission to a communication satellite.

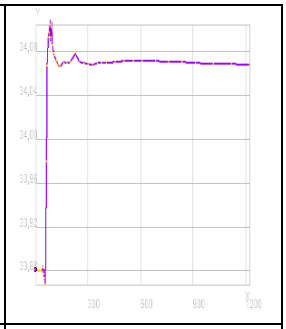
An example of Argo float trajectory, temperature and salinity profiles are given in Fig.8.




a) trajectory



b) temperature profile



c) salinity profile

Fig.8 An example of an Argo float trajectory and profiles with measurements

For operational estimation of ocean parameters each Argo buoy was considered as a system that was characterized by trajectory and a set of profiles with measurements. Each point where data transmission was fulfilled was defined as objects. For neighboring objects according to the trajectory relations were set. List of possible relations contained two types of relations: 'measured before', 'measured after'. Each object was characterized by a vector of characteristics listed in table 1 and by a vector of measured parameters. The parameters were described in the form presented in table 2.

Table 1 Objects characteristics

| *Name* | *Definition* | *Comment* |
|---|---|---|
| PLATFORM_ NUMBER | char PLATFORM_NUMBER(N_PROF, STRING8);<br>PLATFORM_NUMBER:long_name = "Float unique identifier";<br>PLATFORM_NUMBER:conventions = "WMO float identifier : A9IIIII";<br>PLATFORM_NUMBER:_FillValue = " "; | WMO float identifier. WMO is the World Meteorological Organization. This platform number is unique. Example : 6900045 |
| JULD | double JULD(N_PROF);<br>JULD:long_name = "Julian day (UTC) of the station relative to REFERENCE_DATE_TIME";<br>JULD:units = "days since 1950-01-01 00:00:00 UTC";<br>JULD:conventions = "Relative julian days with decimal part (as parts of day)";<br>JULD:_FillValue = 999999.; | Julian day of the profile1. The integer part represents the day, the decimal part represents the time of the profile. Date and time are in universal time coordinates.<br>Example : 18833.8013889885 : July 25 2001 19:14:00 |
| LATITUDE | double LATITUDE(N_PROF);<br>LATITUDE:long_name = "Latitude of the station, best estimate";<br>LATITUDE:units = "degree_north";<br>LATITUDE:_FillValue = 99999.;<br>LATITUDE:valid_min = -90.;<br>LATITUDE:valid_max = 90.; | Latitude of the profile. Unit : degree north.<br>Example : 44.4991 : 44° 29' 56.76'' N |
| LONGITUDE | double LONGITUDE(N_PROF); | Longitude of the profile. Unit : |

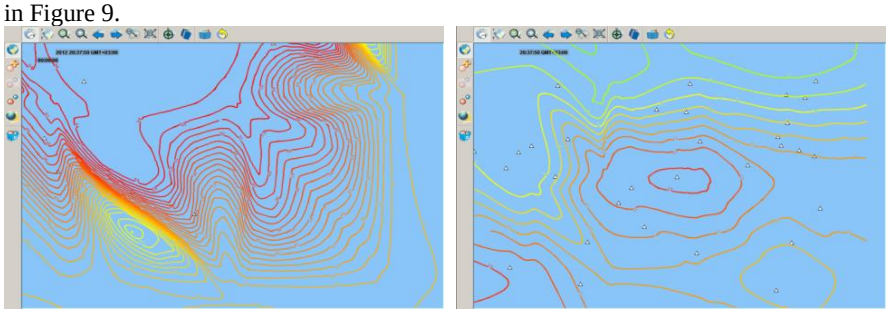
| | | |
|---|---|---|
| | LONGITUDE:long_name = "Longitude of the station, best estimate";<br>LONGITUDE:units = "degree_east";<br>LONGITUDE:_FillValue = 99999.;<br>LONGITUDE:valid_min = -180.;<br>LONGITUDE:valid_max = 180.; | degree east.<br>Example : 16.7222 : 16° 43' 19.92'' E |

Table 2 The description of parameters measurements

| *Name* | *Definition* | *Comment* |
|---|---|---|
| <PARAM> | float <PARAM>(N_PROF, N_LEVELS);<br><PARAM>:long_name = "<X>";<br><PARAM>:_FillValue = <X>;<br><PARAM>:units = "<X>";<br><PARAM>:valid_min = <X>;<br><PARAM>:valid_max = <X>;<br><PARAM>:comment = "<X>";<br><PARAM>:resolution = <X>; | <PARAM> contains the original values of a parameter<br><X> format of values representation |

To provide end-users with actual information based on results of new measurements, regular grids were rebuilt for the region where new data was received. Identification of ocean regions borders can be made manually by experts of subject domain or using algorithms of cluster analyzes. Algorithms for building gridded data were extended by a preliminary step that assumed assessment of observable situation. Buoys with similar or partly similar trajectories that have close measurements values were found using algorithms for building and comparing situation graphs. Depending on distances between the analyzed and similar situations weight coefficients were assigned to measurements. The highest values were assigned to newly received measurements. When rebuilding grid weight of measurements are considered. It allows calculating ocean parameters estimations based on new data and take into account tendencies that were observed in similar situations. As not all grid is rebuild, but only region of interest, processing is executed enough fast to meet users requirements.

Examples of results of ocean data processing using proposed approach are given in Figure 9.



a) Temperature b) Salinity

**Fig. 9.**Measuring facilities and ocean parameters regular grids

The evaluation of the results was carried out by comparing measurements from a test set that contained 5000 temperature and salinity values for various depths measured by instruments and calculated values for the same parameters at the points with the same coordinates. The result of the comparison showed that the accuracy of calcu-

lated parameters values has increased up to 5% in some regions and in average in about 2-3%.

## 10 Conclusion

The application of the proposed method for situation assessment allows to take into account results of objects parameters measurements received from different sources. Recognition of situations and revealing similar situations provides possibility to obtain additional information about observed situation including tendencies and dynamics of its development. The approach to describe and compare situations using graphs provides high speed of calculations. Thus, we can say that the presented method can solve all problems considered in the paper.

Our future research is connected with developing algorithms that will allow using information about dependencies between parameters and their mutual influence.

## References

1. Endsley, M.: Toward a Theory of Situation Awareness in Dynamic Systems. Human Factors, vol. 37, no 1, 32 – 64 (995)
2. Ganter, B., Wille, R.: Formal Concept Analysis: Mathematical Foundations, Springer (1999)
3. Kuznetsov, S.O.: Mathematical aspects of concept analysis. Journal of Mathematical Science, vol. 80, issue 2, 1654-1698 (1996)
4. Bunke, H., Allermann, G.: Inexact graph matching for structural pattern recognition. Pattern Recognition Letters, no. 1, 245–253 (1983)
5. Lin, J., Keogh, E., Wei L., Lonardi, S.: Experiencing SAX: a novel symbolic representation of time series. Data Mining and knowledge discovery, vol. 15, no. 2, 107‑144 (2007)
6. Lin, J., Keogh, E., Lonardi, S., Chiu, B.: A Symbolic Representation of Time Series with Implications for Streaming Algorithms. In: 8th ACM SIGMOD Workshop on Research Issues in Data Mining and Knowledge Discovery, pp. 2–11. ACM Press (2003)
7. Fedotov, Y., Hovanov, N. Complex Production Systems' Performance Measurement: Methods of Estimation Aggregate Indices. Discussion Paper #25(R)–2006. Institute of Management, Saint Petersburg State University, St. Petersburg (2006)
8. Kugiumtzis, D., Tsimpiris, A. Measures of Analysis of Time Series (MATS):A MATLAB Toolkit for Computation of Multiple Measures on Time Series Data Bases. Journal of Statistical Software, vol. 33. issue 5, 1-30 (2010)
9. Sokolov, I. S.: Method for building graph model for group telemetric signal. In: Scientific session of National Research Nuclear University MEPhI, pp. 77‑78. MEPhI, Moscow (2011)
10. Eshera, M., Fu, K.: A graph distance measure for image analysis. IEEE Transactions on Systems, Man, and Cybernetics (Part B), vol. 14, no. 3, 398–408 (1984)
11. Eshera M., Fu, K.: A similarity measure between attributed relational graphs for image analysis. In: 7th International Conference on Pattern Recognition, pp. 75–77. Springer, Heidelberg (1984)

12. Riesen, K., Neuhaus, M., Bunke, H.: Bipartite graph matching for computing the edit distance of graphs. In: 6th International Workshop on Graph Based Representations in Pattern Recognition. LNCS, vol. 2726, pp. 1–12. Springer, Heidelberg (2006)
13. Riesen, K., Bunke, H.: Approximate graph edit distance computation by means of bipartite graph matching. Image and Vision Computing, vol. 27, no. 7, 950–959 (2009)
14. Wille, R.: Conceptual structures of multicontexts. In: ICCS, Eklund, P. W., Ellis G., Mann, G. (eds.). Lecture Notes in Computer Science Series, vol. 1115, pp. 23-39. Springer, Heidelberg (1996)
15. Ganter, B., Kuznetsov, S. O.: Pattern structures and their projections. In: ICCS, Delugach H. S., Stumme G., (eds). Lecture Notes in Computer Science Series, vol. 2120, pp. 129-142. Springer, Heidelberg (2001)
16. Witten, I., Frank, E., Hall, M.: Data Mining: Practical Machine Learning Tools and Techniques. Morgan Kaufmann, third edition, San Francisco (2011)
17. Kuznetsov, S.O., Obiedkov, S.A.: Comparing Performance of Algorithms for Generating Concept Lattices. Journal of Experimental and Theoretical Artificial Intelligence, vol. 14, no. 2-3, 189-216 (2002)
18. Korablev, A. A., Pnushkov, A. V., Smirnov, A. V.: Compilation of the oceanographic database for the Nordic Seas. Journal of Arctic and Antarctic Research Institute, vol. 447, 85-108 (2007)
19. Boyer, T., Levitus, S., Garcia, H., Locarnini, R. A., Stephens, C., Antonov, J.: Objective analyses of annual, seasonal, and monthly temperature and salinity for the World Ocean on a 0.25° grid. Int. J. Climatol., vol. 25, 931–945 (2005)
20. Zhuang, S. Y., Fu, W. W., She, J.: A pre-operational three Dimensional variational data assimilation system in the North/Baltic Sea. Ocean Sci., vol. 7, 771-781 (2011)
21. http:// www.argo.ucsd.edu/

# Author Index