Conceptual Visualization and Navigation Methods for Polyadic Formal Concept Analysis[*]

Diana Troancă
Babeş-Bolyai University Cluj-Napoca
dianat@cs.ubbcluj.ro
Thesis Advisors: Prof. Sebastian Rudolph, sebastian.rudolph@tu-dresden.de, TU Dresden
Prof. Florian-Mircea Boian, florin@cs.ubbcluj.ro, Babeş-Bolyai University Cluj-Napoca

1 Motivation and Problem Description

Conceptual knowledge is closely related to a deeper understanding of existing facts and relationships, but also to the argumentation and communication of why something happens in a particular way. Formal Concept Analysis (FCA) is the core of Conceptual Knowledge Processing. It emerged from applied mathematics and quickly developed into a powerful framework for knowledge representation. It is based on a set-theoretical semantics and provides a rich amount of mathematical instruments for representation, acquiring, retrieval, discovery and further processing of knowledge.

FCA was introduced in the dyadic setting [Ganter and Wille, 1999] and extended to a triadic [Lehmann and Wille, 1995] and eventually n-adic setting [Voutsadakis, 2002]. Intuitively, dyadic datasets can be understood as objects related to attributes and, in addition, to conditions for the triadic case. FCA defines concepts as maximal clusters of data in which all elements are mutually interrelated. A common problem for n-adic FCA is concept visualization and navigation.

The goal of my thesis is to find visualization and navigation paradigms that can be applied to higher-dimensional datasets. Therefore, we study the triadic case and propose several visualization and navigational approaches. Furthermore, we evaluate these approaches, study their generalizations and extend them, where possible, to n-ary formal contexts.

2 Preliminaries

A short introduction to the FCA notions is necessary in order to fully understand the presented topic.

Definition 1. *Let $n \geq 2$ be a natural number. An* n-context *is an $(n+1)$-tuple $\mathbb{K} := (K_1, K_2, \dots, K_n, Y)$, where $K_1, K_2, \dots, K_n$ are sets and Y is an n-ary relation $Y \subseteq K_1 \times K_2 \times \cdots \times K_n$.*

Definition 2. *The* n-concepts *of an n-context $(K_1, \dots, K_n, Y)$ are exactly the n-tuples that satisfy $A_1 \times \cdots \times A_n \subseteq Y$ and which are maximal with respect to component-wise set inclusion. $(A_1, \dots, A_n)$ is called a* proper *n-concept if $A_1, \dots, A_n$ are all non-empty.*

Example 1. *Finite triadic contexts can be represented as multiple cross-tables labeled with condition names, while rows are being labeled with object names and columns with attribute names. In the following example we consider a triadic context where the object set consists of authors of scientific papers, the attribute set contains conference names/journal names while the conditions are the publication years (the data has been selected from dblp database).*

2014	Corr	ICC	PIMRC	HICSS
Rumpe	×			
Alouni	×	×	×	

2015	Corr	ICC	PIMRC	HICSS
Rumpe	×			×
Alouni	×	×		

Figure 1: DBLP data: author, conference/journal, year

There are exactly six triconcepts of this context, i.e., maximal 3D cuboids full of incidences:

- $(\{Rumpe, Alouni\}, \{Corr\}, \{2014, 2015\})$,
- $(\{Alouni\}, \{Corr, ICC, PIMRC\}, \{2014\})$,
- $(\{Alouni\}, \{Corr, ICC\}, \{2014, 2015\})$,
- $(\{Rumpe\}, \{Corr, HICSS\}, \{2015\})$,
- $(\emptyset, \{Corr, ICC, PIMRC, HICSS\}, \{2014, 2015\})$ *and*
- $(\{Rumpe, Alouni\}, \{Corr, ICC, PIMRC, HICSS\}, \emptyset)$

The first four of these triconcepts are proper.

3 Progress to Date

3.1 Visualization of Triadic Contexts

In our previous work we applied triadic FCA to study the navigational patterns of students in an e-learning environment [Dragoş *et al.*, 2014a; 2014b]. The purpose is to find methods to enhance e-learning systems and to make predictions of user patterns. The problem that arised during this study is that the triconcepts contain a large amount of data, so displaying the components is not helpful. The approach used in these works is to define some dyadic projections and visualize the dyadic data obtained with a tool called Circos that uses a circular layout.

[*] My research was supported by a scholarship from DAAD.

3.2 A Navigation Paradigm based on Reachability Clusters

Next, we address the problem of navigation and consider a paradigm based on a neighborhood notion arising from appropriately defined projections [Rudolph *et al.*, 2015b]. The navigation starts locally, from a chosen triconcept and the first step is to fix one perspective, i.e. one of the three dimensions. The next step of the navigation must be chosen from the so-called directly reachable concepts, which can be computed with a dyadic projection once a perspective is chosen. After selecting one of the directly reachable triconcepts, one may change the perspective and move towards a different group of reachable triconcepts. Because of the local character, this navigation allows to explore large datasets. The corresponding reachability relation defined as a transitive closure of the direct reachability, gives rise to equivalence classes called clusters. These clusters are structured by a partial order. As a consequence, we present in the paper a first exploration strategy based on the previously introduced notions.

One of the useful techniques used to optimize visualization and navigation in dyadic contexts are the clarification and reduction processes. In order to support our proposed navigation paradigm, we extend these notions to triadic formal contexts [Rudolph *et al.*, 2015c]. The intuition behind is to eliminate elements from the context without loss of information. In the clarification phase one looks for elements of a set having the same behavior, i.e. that are in relation with the same elements of the other two sets. Those elements can be clarified (unified as one element), hence reducing the size of the context. The reduction process consists of the removal of elements which can be written as combinations of other elements of the same set. Both techniques help reducing the size of the context while keeping its structure for further analysis.

3.3 Interactive Concept Navigation using Membership Constraints

The second navigational approach [Rudolph *et al.*, 2015a] is more user-oriented and has an interactive character. We give the user the freedom to choose what elements to include and exclude from the components of the formal concept. We formalize the user's restrictions as membership constraints.

Definition 3. *An* n-adic membership constraint *on an* n*-context* $\mathbb{K} = (K_1, \ldots, K_n, R)$ *is a* $2n$*-tuple* $\mathbb{C} = (K_1^+, K_1^-, \ldots, K_n^+, K_n^-)$ *with* $K_i^+ \subseteq K_i$ *called* required sets *and* $K_i^- \subseteq K_i$ *called* forbidden sets.

An n*-concept* $(A_1, \ldots, A_n)$ *of* $\mathbb{K}$ *is said to* satisfy *such a membership constraint if* $K_i^+ \subseteq A_i$ *and* $K_i^- \cap A_i = \emptyset$ *hold for all* $i \in \{1, \ldots, n\}$.

After each step, i.e. when the user adds a new restriction, there is a propagation phase, during which we compute all additional restrictions required for ensuring that the user will always get to a proper n-concept at the end of the navigation.

We consider this approach to be more intuitive and user-friendly than the previous one, especially for users who are not very familiar with FCA. Hence, we intesively analyzed implementation methods for this navigation framework. We came up with two different strategies for the implementation of the navigation tool. The first strategy[1] relies on Answer Set Programming [Gebser *et al.*, 2012] and extends the ASP encoding of the membership constraint satisfiability problem [Rudolph *et al.*, 2015a]. The advantage of this navigation tool is that it can be easily extended to n-ary contexts, for any n, by the model of the already implemented cases $n \in \{2, 3, 4\}$. The second strategy[2] is a brute force implementation and uses an exhaustive search in the whole formal concept space. In order to use this tool one must precompute the formal concepts with an existing tool, like Trias[3] for the triadic case. Both implementations were described in more detail, as well as evaluated and compared, in a paper currently under review.

4 Future Plan

For future work, we plan to implement the navigation paradigm based on clusters as well as the brute-force approach for the n-adic case and run usability studies on all the proposed navigation paradigms.

References

[Dragoş *et al.*, 2014a] S. Dragoş, D. Haliţă, C. Săcărea, and D. Troancă. Applying triadic FCA in studying web usage behaviors. In R. Buchmann, C.V. Kifor, and J. Yu, editors, *KSEM 2014*, volume 8793 of *LNCS*, pages 73–80. Springer, 2014.

[Dragoş *et al.*, 2014b] S. Dragoş, D. Haliţă, C. Săcărea, and D. Troancă. An fca grounded study of user dynamics through log exploration. *Studia Univ. Babes-Bolyai, Ser. Informatica*, 2(LIX):82–97, 2014.

[Ganter and Wille, 1999] B. Ganter and R. Wille. *Formal concept analysis - mathematical foundations*. Springer, 1999.

[Gebser *et al.*, 2012] M. Gebser, R. Kaminski, B. Kaufmann, and T. Schaub. *Answer Set Solving in Practice*. Synthesis Lectures on Artificial Intelligence and Machine Learning. Morgan and Claypool Publishers, 2012.

[Lehmann and Wille, 1995] F. Lehmann and R. Wille. A triadic approach to formal concept analysis. In G. Ellis, R. Levinson, W. Rich, and J.F. Sowa, editors, *ICCS 1995*, volume 954 of *LNCS*, pages 32–43. Springer, 1995.

[Rudolph *et al.*, 2015a] S. Rudolph, C. Săcărea, and D. Troancă. Membership constraints in formal concept analysis. In Q. Yang and M. Wooldridge, editors, *IJCAI 2015*, pages 3186–3192. AAAI Press, 2015.

[Rudolph *et al.*, 2015b] S. Rudolph, C. Săcărea, and D. Troancă. Towards a navigation paradigm for triadic concepts. In J. Baixeries, C. Săcărea, and M. Ojeda-Aciego, editors, *ICFCA 2015*, volume 9113 of *LNCS*, pages 252–267. Springer, 2015.

[Rudolph *et al.*, 2015c] S. Rudolph, C. Săcărea, and Diana Troancă. Reduction in triadic data sets. In S.O. Kuznetsov, A. Napoli, and S. Rudolph, editors, *FCA4AI 2015*, volume 1430 of *CEUR Workshop Proc.*, pages 55–62. CEUR-WS.org, 2015.

[Voutsadakis, 2002] G. Voutsadakis. Polyadic concept analysis. *Order*, 19(3):295–304, 2002.

[1] https://sourceforge.net/projects/asp-concept-navigation

[2] https://sourceforge.net/projects/brute-force-concept-navigation

[3] https://github.com/rjoberon/trias-algorithm