Pattern Structures and Concept Lattices for Data Mining and Knowledge Processing

Mehdi Kaytoue[1], Victor Codocedo[1], Aleksey Buzmakov[2], Jaume Baixeries[3], Sergei O. Kuznetsov[4], Amedeo Napoli[2]

[1] Université de Lyon. CNRS, INSA-Lyon, LIRIS. UMR5205, F-69621, France.
[2] LORIA (CNRS - Inria Nancy Grand Est - Université de Lorraine), B.P. 239, F-54506, Vandœuvre-lès-Nancy.
[3] Universitat Politècnica de Catalunya. 08032, Barcelona. Catalonia.
[4] National Research University Higher School of Economics (HSE), Kochnovski pr.3, Moscow 125319, Russia

Abstract. This article aims at presenting recent advances in Formal Concept Analysis (2010-2015), especially when the question is dealing with complex data (numbers, graphs, sequences, etc.) in domains such as databases (functional dependencies), data-mining (local pattern discovery), information retrieval and information fusion. As these advances are mainly published in artificial intelligence and FCA dedicated venues, a dissemination towards data mining and machine learning is worthwhile.

1 Pattern Structures in Formal Concept Analysis

Formal Concept Analysis (FCA) is a branch of applied lattice theory that appeared in the 1980's [11]. Starting from a binary relation between a set of objects and a set of attributes, formal concepts are built as maximal sets of objects in relation with maximal sets of attributes, by means of derivation operators forming a Galois connection. Concepts form a partially ordered set that represents the initial data as a hierarchy, called the concept lattice. This conceptual structure has proved to be useful in many fields, e.g. artificial intelligence, knowledge management, data-mining and machine learning, morphological mathematics, etc. In particular, several results and algorithms from itemset and association rule mining and rule-based classifiers were already characterized in terms of FCA [17, 20]. For example, the set of frequent closed itemsets is an order ideal of a concept lattice; association rules and functional dependencies can be characterized with the derivation operators; jumping patterns were defined as hypotheses, etc, not to mention efficient polynomial-delay algorithms for building all closed itemsets such as *CloseByOne* [18].

The goal of this communication is to present our recent advances in FCA over the period 2010–2015, especially when the question is dealing with complex data, thanks to the rich formalism of *pattern structures* [10]. This general approach translates FCA to any partially ordered data descriptions to deal elegantly with non binary, say *complex*, *heterogeneous* and *structured* data. Pattern structures also allow new ways of solving problems in several applications (see next section).

The key idea relies on defining so-called *similarity operators* which induce a semi-lattice on data descriptions. Several alternative attempts were made for defining such semi-lattices on sets of graphs and logical formulas (see, e.g., the works of Chaudron&Maille, Ferré&Ridoux, Polaillon&Brito cited in [16]). Formally, a pattern structure is a triple $(G, (D, \sqcap), \delta)$ where G is a set of objects, $(D, \sqcap)$ is a meet-semi-lattice of potential object descriptions and $\delta : G \longrightarrow D$ is a mapping associating each object with its description. Elements of D are called patterns and are ordered with a subsumption relation $\sqsubseteq$: $\forall c, d \in D$, $c \sqsubseteq d \iff c \sqcap d = c$. For any $A \subseteq G$ and $d \in (D, \sqcap)$, two derivation operators are defined: as $A^{\square} = \bigsqcap_{g \in A} \delta(g)$ and $d^{\square} = \{g \in G | d \sqsubseteq \delta(g)\}$. These operators form a Galois connection between $(\wp(G), \subseteq)$ and $(D, \sqcap)$. Pattern concepts of $(G, (D, \sqcap), \delta)$ are pairs of the form (A, d), $A \subseteq G$, $d \in (D, \sqcap)$, such that $A^{\square} = d$ and $A = d^{\square}$. For a pattern concept (A, d), d is a pattern intent and is the common description of all objects in A, the pattern extent. When partially ordered by $(A_1, d_1) \leq (A_2, d_2) \Leftrightarrow A_1 \subseteq A_2 \ \ (\Leftrightarrow d_2 \sqsubseteq d_1)$, the set of all concepts forms a complete lattice called pattern concept lattice.

Pattern structures offer a concise way to define closed patterns. They also allow efficient polynomial-delay algorithms (modulo complexity of computing $\sqsubseteq$ and $\sqcap$) [18]. In presence of large datasets, they offer natural approximation tools (*projections*, detailed below) and achieve lazy classification [19].

Data heterogeneity. When D is the power set of a set of items I, $\sqcap$ and $\sqsubseteq$ are the set intersection and inclusion resp.: pattern intents are closed itemsets and we fall back in standard FCA settings. Originally, pattern structures were introduced to handle objects described by labeled graphs [18]. We developed the general approach in various ways for handling objects described by: numbers and intervals [16], partitions [4], sequences [5] and trees [22].

Data approximation. Pattern structure projections simplify computation and reduce the number of concepts [10]. For example, a set of labeled graphs can be projected as a set of k-chains [21], while intervals can be enlarged [12]. A projection ψ associates any pattern to a *more general* pattern covering more objects. A projection is $\sqcap$-preserving: $\forall c, d \in D, \psi(c \sqcap d) = \psi(c) \sqcap \psi(d)$. We studied how numerical data can be projected when a similarity relation between numbers (symmetric, reflexive but not transitive relation) is considered and showed that a projection can be performed as a pre-processing task. We also introduced a wider class of projections [6]: while projections can only modify object descriptions, *o-projections* modifies the semi-lattice of descriptions.

Data representation. For any pattern structure, a *representation* context can be built, which is a binary relation encoding the pattern structure. Concepts in both data representations are in 1-1-correspondence. We studied this aspect for several types of patterns, designing the transformation procedures and evaluating in which conditions one data representation prevails [15, 4]. We also showed that the bijection does not hold in general for minimal generators (qualified as *free* or *key* in pattern mining) [15]. The impact of projections on representation contexts are investigated with the new class of o-projections [6].

2 Applications

Database and functional dependencies. Characterizing and computing functional dependencies (FDs) are an important topic in database theory (see e.g. references in [4]). In FCA, Ganter & Wille proposed a first characterization of FDs as implications in a formal context (binary relation) obtained after a transformation of the initial data [11]. However, n^2 objects are created from the n initial tuples. To overcome this problem, we present a characterization of functional dependencies in terms of (partition) pattern structures that offers additional benefits for the computation of dependencies [4]. This method can be naturally generalized to other types of FDs (multi-valued and similarity dependencies [3]).

Pattern mining and biclustering. Biclustering aims at finding local patterns in numerical data tables. The motivation is to overcome the limitation of standard clustering techniques where distance functions using all the attributes may be ineffective and hard to interpret. Applications are numerous in biology, recommendation, etc. (see references in [13, 7]). In FCA, formal concepts are maximal rectangles of $True$ values in a binary data-table (modulo columns/rows permutations). Accordingly, concepts are binary biclusters with interesting properties: maximality (via a closure operator), overlapping and partial ordering of the local patterns. Such properties are key elements of a mathematical definition of numerical biclusters and the design of their enumeration algorithms. We highlight these links for several types of biclusters with interval [14] and partition pattern structures [7] and their representation contexts. Next investigations concern dimensionality: a bijection between n-clusters and $n+1$-concepts is proven [13].

Information retrieval. FCA has been used in a myriad of ways to support a wide variety of information retrieval (IR) techniques and applications [9]: the concept lattice represents concisely the document and the query space which can be used as an index for automatic retrieval. In the last years, the Boolean IR model (and consequently, FCA) has been considered as too limited for modern IR requirements, such as large datasets and complex document representations. Pattern structures have shown a great potential to reuse the body of work of FCA-based IR approaches by providing support to complex document representations, such as numerical and heterogeneous indexes [8]. In the context of semantic web, a noticeable application of this model is RDF data completion [1].

Information fusion for decision making. Merging information given by several sources (databases, experts...) into an interpretable and useful format is a tricky task. Fusion results may not be in suitable form for being used in decision analysis. This is due to the fact that information sources are heterogeneous, noisy and inconsistent. We investigated how FCA and pattern structures can be used in decision making when fusion is required: pattern concept lattices (based on intervals) provide an information fusion space where maximal subsets of information can be detected and support decision making [2].

References

1. M. Alam, A. Buzmakov, V. Codocedo, and A. Napoli. An approach for Improving RDF Data with Formal Concept Analysis. *Int. Joint Conf. on Artif. Intell.*, 2015.
2. Z. Assaghir, A. Napoli, M. Kaytoue, D. Dubois, and H. Prade. Numerical information fusion: Lattice of answers with supporting arguments. In *Int. Conf. on Tools with Artificial Intelligence (ICTAI)*, pages 621–628. IEEE, 2011.
3. J. Baixeries, M. Kaytoue, and A. Napoli. Computing similarity dependencies with pattern structures. In *Int. Conf. on Concept Lattices and Their Applications (CLA)*, CEUR 1062, pages 33–44, 2013.
4. J. Baixeries, M. Kaytoue, and A. Napoli. Characterizing functional dependencies in formal concept analysis with pattern structures. *Ann. Math. Artif. Intell.*, 72(1-2):129–149, 2014.
5. A. Buzmakov, E. Egho, N. Jay, S. Kuznetsov, A. Napoli, and C. Raïssi. On Mining Complex Sequential Data by Means of FCA and Pattern Structures. *International Journal of General Systems*, 2015.
6. A. Buzmakov, S. Kuznetsov, and A. Napoli. Revisiting Pattern Structure Projections. In *Formal Concept Analysis (ICFCA)*, LNAI 9113, 2015.
7. V. Codocedo and A. Napoli. Lattice-based biclustering using partition pattern structures. In *European Conf. on Artificial Intelligence (ECAI)*, 2014.
8. V. Codocedo and A. Napoli. A proposition for combining pattern structures and relational concept analysis. In *Formal Concept Analysis*, LNCS 8478, 2014.
9. V. Codocedo and A. Napoli. Formal concept analysis and information retrieval - a survey. In *Int. Conf. on Formal Concept Analysis (ICFCA)*, 2015. Accepted.
10. B. Ganter and S. O. Kuznetsov. Pattern structures and their projections. In *Int. Conf. on Conceptual Structures (ICCS)*, LNCS 2120, pages 129–142, 2001.
11. B. Ganter and R. Wille. *Formal Concept Analysis*. Springer, Berlin, 1999.
12. M. Kaytoue, Z. Assaghir, A. Napoli, and S. O. Kuznetsov. Embedding tolerance relations in fca: an application in information fusion. In *CIKM*. ACM, 2010.
13. M. Kaytoue, S. O. Kuznetsov, J. Macko, and A. Napoli. Biclustering meets triadic concept analysis. *Ann. Math. Artif. Intell.*, 70(1-2):55–79, 2014.
14. M. Kaytoue, S. O. Kuznetsov, and A. Napoli. Biclustering numerical data in formal concept analysis. In *Formal Concept Analysis (ICFCA)*, LNCS 6628, 2011.
15. M. Kaytoue, S. O. Kuznetsov, and A. Napoli. Revisiting numerical pattern mining with formal concept analysis. In *Int. Joint Conf. on Art. Intell.(IJCAI)*, 2011.
16. M. Kaytoue, S. O. Kuznetsov, A. Napoli, and S. Duplessis. Mining gene expression data with pattern structures in formal concept analysis. *Inf. Sci.*, 181(10), 2011.
17. S. Kuznetsov. Galois Connections in Data Analysis: Contributions from the Soviet Era and Modern Russian Research. In *Formal Concept Analysis (ICFCA)*, LNCS 3626, pages 196–225. Springer, 2005.
18. S. O. Kuznetsov. Learning of simple conceptual graphs from positive and negative examples. In *PKDD*, LNCS 1704, pages 384–391. Springer, 1999.
19. S. O. Kuznetsov. Fitting pattern structures to knowledge discovery in big data. In *Formal Concept Analysis (ICFCA)*, LNCS 7880, pages 254–266. Springer, 2013.
20. S. O. Kuznetsov and J. Poelmans. Knowledge representation and processing with formal concept analysis. *Wiley Interdisc. Rew.: Data Mining and Knowledge Discovery*, 3(3):200–215, 2013.
21. S. O. Kuznetsov and M. V. Samokhin. Learning closed sets of labeled graphs for chemical applications. In *ILP*, LNCS 3625, pages 190–208. Springer, 2005.
22. A. Leeuwenberg, A. Buzmakov, Y. Toussaint, and A. Napoli. Exploring Pattern Structures of Syntactic Trees for Relation Extraction. In *ICFCA*, LNAI 9113, 2015.