Symbolic Data Analysis and Formal Concept Analysis

Mehdi Kaytoue, Sergei O. Kuznetsov, Amedeo Napoli, Géraldine Polaillon

Symbolic Data Analysis and Formal Concept Analysis

Mehdi Kaytoue*, Sergei O. Kuznetsov**, Amedeo Napoli*, Géraldine Polaillon***

* LORIA – Campus Scientifique, B.P. 239 – Vandœuvre-lès-Nancy – France
Mehdi.Kaytoue@loria.fr;Amedeo.Napoli@loria.fr
** HSE – Pokrovskiy Bd. 11 – 109028 Moscow – Russia
skuznetsov@yandex.ru
*** E3S Supélec – 3 rue Joliot-Curie – 91192 Gif sur Yvette – France
Geraldine.Polaillon@supelec.fr

Abstract. Formal concept analysis (FCA) can be used for designing concept lattices from binary data for knowledge discovery purposes. Pattern structures in FCA are able to deal with complex data. In addition, this formalism provides a concise and an efficient algorithmic view of the formalism of symbolic data analysis (SDA).

1 Introduction

Many classification problems can be formalized by means of a *formal context*, a binary relation between an object set and an attribute set indicating whether an object has or does not have an attribute (see Ganter and Wille (1999)). According to the so-called *Galois connection*, one may classify within *formal concepts* a set of objects sharing a same maximal set of attributes, and vice-versa. Concepts are ordered in a lattice structure called *concept lattice* within the Formal Concept Analysis (FCA) framework. FCA can be used for a number of purposes like knowledge formalization and acquisition, ontology design, and data mining. To handle complex data in FCA, *pattern structures* have been proposed as a generalization of formal contexts to complex data (see Kuznetsov (2009); Kaytoue et al. (2011)). On the other hand, Symbolic Data Analysis (SDA, see Bock and Diday (2000)) aims at analyzing data such as numbers, intervals, sets of discrete values, etc. An object is described by a vector of values with each dimension corresponding to a variable, and each variable may be of different type. In Brito (1994); Brito and Polaillon (2005), the problem is addressed of building concept lattices by formalizing "symbolic objects" in SDA and properly defined Galois connections between these symbolic individuals and their descriptions. The links between the FCA and SDA approaches still remain unclear. Although both methods show the same behavior when working on the same data, the goal of this paper is to discuss how the SDA formalism for building concept lattices can be taken into account in FCA in a universal way, to facilitate comprehension and future extension (see also Agarwal et al. (2011)).

The paper is organized as follows. Section 2, 3 respectively present SDA, and pattern structures. Both approaches are compared and discussed in Section 4. Limited by space, we assume that the reader is familiar with FCA (see Ganter and Wille (1999)).

2 Symbolic Galois Lattices in Symbolic Data Analysis

	y_1	y_2
ω_1	[75, 80]	[1, 2]
ω_2	[60, 80]	[1, 1]
ω_3	[50, 70]	[2, 2]
ω_4	[72, 73]	[1, 2]

TAB. 1 –

The formalism of "Symbolic Data Analysis" was introduced and fully described (among others) in Brito (1994); Brito and Polaillon (2005). Due to place restrictions, we will not go into the details of SDA and we will briefly introduce some basic elements necessary for understanding this paper with the help of an example, see Table 1. Let $\Omega = \{\omega_1, \omega_2, \omega_3, \omega_4\}$ be a set of objects described by two variables y_1 with range $O_1 = \{[75, 80], [60, 80], [50, 70], [72, 73]\}$ and y_2 with range $O_2 = \{[1, 2], [1, 1], [2, 2]\}$. Then $(y_1 \subseteq [70, 80])$ can be considered as an "intensional description" –or elementary symbolic object– whose "extension" is the set $\{\omega_1, \omega_4\}$. Then an "assertion object" –that could be termed as a (generalized) symbolic object– is a conjunction of such elementary symbolic objects. For example, $d_1 = (y_1 \subseteq [60, 80]) \wedge (y_2 \subseteq [1, 2])$ describes the set $ext(d_1) = \{\omega_1, \omega_2, \omega_4\}$.

A partial ordering between description can be defined as follows: if d_1 and d_2 are two (generalized) intensional descriptions, then $d_1 \leq d_2 \Leftrightarrow ext(d_1) \subseteq ext(d_2)$. Further, Galois connections can be defined between $\wp(\Omega)$ and A depending on the choice of a "generalization operator" for building the upper bound of two assertions objects (see Brito (1994); Brito and Polaillon (2005)).

3 Pattern Concept Lattices

Pattern structures are introduced in Ganter and Kuznetsov (2001) in full compliance with FCA and can be thought of as a "generalization" of formal contexts to complex data from which a concept lattice can be built without any *a priori* scaling.

Formally, let G be a set of objects, $(D, \sqcap)$ be a semi-lattice of object descriptions, and $\delta : G \rightarrow D$ be a mapping. $(G, (D, \sqcap), \delta)$ is called a *pattern structure*. Elements of D are called *patterns* and are ordered by ordering relation $\sqsubseteq$, i.e. given $c, d \in D$, we have $c \sqsubseteq d \Longleftrightarrow c \sqcap d = c$. We use the operator $(.)^{\square}$ to derive the following images, where operator $(.)^{\square}$ corresponds to operator $(.)'$ in standard FCA:

$$A^{\square} = \bigsqcap_{g \in A} \delta(g), \text{ for any } A \subseteq G,$$
$$d^{\square} = \{g \in G \mid d \sqsubseteq \delta(g)\}, \text{ for any } d \in (D, \sqcap)$$

These operators form a Galois connection between $(\mathfrak{P}(G), \subseteq)$ and $(D, \sqsubseteq)$. $(.)^{\square\square}$ is a closure operator. *Pattern concepts* of $(G, (D, \sqcap), \delta)$ are pairs of the form (A, d), $A \subseteq G$, $d \in D$, such that $A^{\square} = d$ and $A = d^{\square}$, and d is called a *pattern intent* while A is a *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 pattern concepts forms a complete lattice called a *pattern concept lattice*. An example is given in the next section. Standard FCA algorithms need slight modification to compute the pattern concept lattice, see e.g. Ganter and Kuznetsov (2001); Kaytoue et al. (2011).

	y_1	y_2	y_3
g_1	[75,80]	[1,2]	{a,b}
g_2	[60,80]	[1,1]	{d,e}
g_3	[50,70]	[2,2]	{a,c}
g_4	[72,73]	[1,2]	{a}

TAB. 2 –

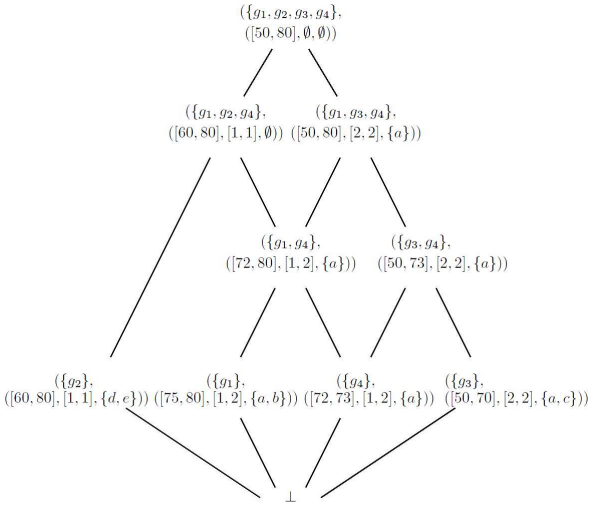

FIG. 1 – *Pattern concept lattice designed from Table 2.*

4 Symbolic Galois Lattices with Pattern Structures

SDA works on data tables where each column corresponds to a variable y_i. Pattern structures consider $(D, \sqcap)$ as corresponding to one variable in terms of SDA. Thus, given a set $Y = \{y_1, ..., y_p\}$ of p variables, we consider the direct product $(D, \sqcap) = (D_{y_1}, \sqcap_{y_1}) \times ... \times (D_{y_p}, \sqcap_{y_p})$ of all semi-lattices $(D_{y_i}, \sqcap_{y_i})$ for each $y_i \in Y$. $(D, \sqcap)$ is a semi-lattice itself containing all possible descriptions of objects and sets of objects, and corresponds to the set of possible intensional descriptions in SDA. The partial ordering $\sqsubseteq$ in $(D, \sqcap)$ is such that, for any $c, d \in D$, $c \sqcap d = c \iff c \sqsubseteq d$. Then a pattern $d \in D = (d_1, ..., d_p)$ is called a *pattern vector*. For any $c, d \in D$: $c \sqcap d = (c_1 \sqcap_{y_1} d_1, ..., c_p \sqcap_{y_p} d_p)$ and $c \sqsubseteq d \Leftrightarrow c_i \sqsubseteq_{y_i} d_i \; \forall i = 1...p$. A dimension i of a pattern vector corresponds to a variable y_i which may have a different type. For example, considering intervals, let us define define $\sqcap_{y_1}$ as interval convexification, i.e. with $a_1, b_1, a_2, b_2 \in \mathbb{R}$: $[a_1, b_1] \sqcap_y [a_2, b_2] = [min(a_1, a_2), max(b_1, b_2)]$ and $[a_1, b_1] \sqsubseteq_y [a_2, b_2] \Leftrightarrow [a_1, b_1] \supseteq [a_2, b_2]$. Based on this partial ordering of descriptions, the general Galois connection defined for pattern structures allows to compute pattern concepts and lattices from heterogeneous data.

The example in Table 2 can be represented as a pattern structure $(G, (D, \sqcap), \delta)$ where $G = \{g_1, g_2, g_3, g_4\}$ and $\delta(g_1) = ([75, 80], [1, 2], \{a, b\})$. Descriptions contain two interval-valued variables: y_1 where ordering is based on interval intersection, y_2 where ordering is based on interval convexification, and one categorical multi-valued variable y_3 where ordering is based on inclusion. For example, $\{g_1, g_3\}^{\square\square} = ([50, 80], [2, 2], \{a\})$ and $\{g_1, g_3\}^{\square\square} = \{g_1, g_3, g_4\}$. Hence, $(\{g_1, g_3, g_4\}, ([50, 80], [2, 2], \{a\})$ is a pattern concept of $(G, (D, \sqcap), \delta)$.

The links with SDA formalism are natural but the algorithmic machinery is not the same at all: algorithms building pattern structures are very efficient and can easily build the SDA lattices (see Kaytoue et al. (2011)), but the converse is not true. Moreover, pattern structures consider object descriptions in their original form and propose any kind of partial ordering

between descriptions (compare with intersection and union, the actual two types of partial ordering in SDA).

5 Conclusion

Pattern structures allow to directly consider complex data, avoiding to represent descriptions as symbolic/assertion objects. One general Galois connection is sufficient to consider several data-types, hence it is not required to define a new Galois connection for different data-types and description generalization operations (with union and intersection in SDA). Indeed, the main core of pattern structures lies in defining an appropriate semi-lattice operation inducing a partial order of descriptions. This is rather simple with numerical and categorical data as illustrated in this paper.

Avoiding discretization and loss of information, generally leads to a great amount of concepts. In SDA, it is shown how to reduce concept lattices to simpler hierarchies with reduction techniques based on quality criteria defined in SDA, but this requires to work with a concept lattice already computed, which can be bottleneck for very large databases. On the other hand, pattern structures propose to project object descriptions to "simpler ones" before computation, allowing to reduce the number of concepts. This gives interesting perspectives of research to consider well studied SDA quality criteria within pattern structures.

References

Agarwal, P., M. Kaytoue, S. O. Kuznetsov, A. Napoli, and G. Polaillon (2011). Symbolic Galois Lattices with Pattern Structures. In *Proceedings of RSFDGrC-2011*, LNAI 6743, pp. 191–198. Springer.

Bock, H.-H. and E. Diday (Eds.) (2000). *Analysis of Symbolic Data*. Springer.

Brito, P. (1994). Order structure of symbolic assertion objects. *IEEE Transactions on Knowledge and Data Engineering 6*(5), 830–834.

Brito, P. and G. Polaillon (2005). Structuring probabilistic data by Galois lattices. *Mathématiques et sciences humaines 169*, 77–104.

Ganter, B. and S. O. Kuznetsov (2001). Pattern Structures and Their Projections. In *Proceedings of ICCS-2001*, LNCS 2120, pp. 129–142. Springer.

Ganter, B. and R. Wille (1999). *Formal Concept Analysis*. Springer.

Kaytoue, M., S. O. Kuznetsov, A. Napoli, and S. Duplessis (2011). Mining Gene Expression Data with Pattern Structures in Formal Concept Analysis. *Information Science 181*(10), 1989–2001.

Kuznetsov, S. O. (2009). Pattern Structures for Analyzing Complex Data. In *Proceedings of RSFDGrC-2009*, LNAI 5908, pp. 33–44. Springer.

Résumé

L'analyse formelle de concepts (FCA) est utilisée pour construire des treillis de concepts à partir de tables de données binaires pour des besoins de découverte de connaissances. Les structures de patrons en FCA sont capables de prendre en compte des données complexes et de plus fournissent une vue concise et algorithmique efficace sur le formalisme des objets symboliques (SDA).