Background Knowledge in Formal Concept Analysis: Constraints via Closure Operators*

Radim Belohlavek
SUNY Binghamton, NY 13902, USA
Palacky University, Olomouc, Czech Republic
rbelohla@binghamton.edu

Vilem Vychodil
SUNY Binghamton, NY 13902, USA
Palacky University, Olomouc, Czech Republic
vychodil@binghamton.edu

ABSTRACT

The aim of this short paper is to present a general method of using background knowledge to impose constraints in conceptual clustering of object-attribute relational data. The proposed method uses the background knowledge to extract only particular clusters from the input data—those which are compatible with the background knowledge and thus satisfy the constraint. As a result, the method allows for extracting less clusters in a shorter time which are in addition more interesting. The paper presents the idea of constraints formalized by means of closure operators and introduces such constraints to a particular clustering technique, namely to formal concept analysis. Among the benefits of the presented approach are its versatility (the approach covers several examples studied before, e.g. extraction of closed frequent itemsets in generation of non-redundant association rules) and computational efficiency (polynomial time-delay algorithm for computing constrained clusters). Due to scope limitations, we present the main ideas only. Details will be available in a full version of this paper.

Categories and Subject Descriptors

I.2.3 [**Artificial Intelligence**]: Knowledge Representation Formalisms and Methods—*relation systems*; H.2.8 [**Database Management**]: Database Applications—*data mining*; I.5.3 [**Artificial Intelligence**]: Clustering; F.4.1 [**Mathematical Logic and Formal Languages**]: Mathematical Logic

General Terms

Algorithms, Theory, Human Factors

Keywords

formal concept analysis, background knowledge, constraints

*Supported by research plan MSM 6198959214.

SAC'10 March 22-26, 2010, Sierre, Switzerland.

FCA Notions We assume that the reader is familiar with formal concept analysis (FCA [1]): A *formal context* is denoted by $\langle X, Y, I\rangle$ (i.e., $I \subseteq X \times Y$, elements $x \in X$ and $y \in Y$ are called objects and attributes, respectively; $\langle x, y\rangle \in I$ indicates that object x has attribute y). For $A \subseteq X$, $A^{\uparrow} = \{y \in Y \mid \text{for each } x \in A : \langle x, y\rangle \in I\}$ (set of all attributes shared by all objects from A) and, dually, $B^{\downarrow} = \{x \in X \mid \text{for each } y \in B : \langle x, y\rangle \in I\}$. A *formal concept* in $\langle X, Y, I\rangle$ is a pair $\langle A, B\rangle$ of sets $A \subseteq X$ (extent) and $B \subseteq Y$ (intent) such that $A^{\uparrow} = B$ and $B^{\downarrow} = A$. A subconcept-superconcept relation is a partial order $\leq$ on the set $\mathcal{B}(X, Y, I) = \{\langle A, B\rangle \mid A^{\uparrow} = B, B^{\downarrow} = A\}$ of all formal concepts of $\langle X, Y, I\rangle$ defined by $\langle A_1, B_1\rangle \leq \langle A_2, B_2\rangle$ iff $A_1 \subseteq A_2$ (iff $B_2 \subseteq B_1$). A *concept lattice* of $\langle X, Y, I\rangle$ is the set $\mathcal{B}(X, Y, I)$ equipped with $\leq$ (it is indeed a complete lattice). Fig. 1 represents a formal context with food products as objects and food additives as attributes. The object-attribute relation $I \subseteq X \times Y$ represented by crosses $\times$ indicates whether a food product $x \in X$ does or does not contain an additive $y \in Y$. The corresponding concept lattice $\mathcal{B}(X, Y, I)$ is depicted in Fig. 2 (top left) using the usual labelled line diagram. For instance, the node with label 7 represents a formal concept $\langle A, B\rangle = \langle\{4, 7, 8, 9, 15, 21\}, \{a, c\}\rangle$. A concept lattice is the main output in FCA which is used for subsequent data analysis of data processing.

Problem and Main Idea On of the main challenges in FCA is how to handle the usually large number of formal concepts in a concept lattice. The idea put forward in this paper is based on utilizing a background knowledge which a user may have about the input data. It is often the case that the user has some additional information (background knowledge) about the input data $\langle X, Y, I\rangle$ which lets him conclude that certain formal concepts from $\mathcal{B}(X, Y, I)$ are relevant (interesting, or compatible with the background knowledge) while others are not. For instance, the user may be interested only in formal concepts which contain a sufficiently large number of objects, or which include certain prescribed attributes. In such a case, the user is in fact not interested only in a possibly small part of the concept lattice $\mathcal{B}(X, Y, I)$ rather then in the whole $\mathcal{B}(X, Y, I)$. The challenge consists in devising an appropriate method for treating user background knowledge. We put forward the idea that several types of user background knowledge may be represented by closure operators. Below, we outline a formal treatment, results, and provide illustrative examples. Related work will be reported in the full version of this paper.

Formal Treatment, Results, Examples Recall that a closure operator in Y is a mapping $C : 2^Y \to 2^Y$ sat-

		E322 a	E330 b	E440 c	E471 d	E476 e	E500 f
dia muesli	1	×	×		×		×
cherry muesli	2	×	×		×	×	×
chocolate muesli	3				×		×
strawberry yogurt	4	×	×	×			
hazelnut wafers	5	×			×	×	×
lemon soda	6		×	×			
stracciatella yogurt	7	×		×			
chocolate soy bar	8	×		×		×	
milky way bar	9	×	×	×			×
assorted chocolates	10	×	×			×	
choc-ice	11	×			×		
cranberry muesli	12	×	×				×
margarine 1	13		×		×	×	
enriched margarine	14		×		×		
gingerbread	15	×	×	×		×	×
margarine 2	16	×	×				
margarine 3	17	×	×		×	×	
margarine 4	18	×	×		×		
hazelnut chocolate	19	×				×	
raspberry jelly	20			×			
raisin chocolate	21	×	×	×		×	
cinnamon cookies	22						×
wafers	23	×					×
vegetable broth	24				×		
chocolate wafers	25	×				×	×
chicken broth	26		×				
dia ginger cookies	27	×					

Figure 1: Formal context. Legend: objects = food products, attributes = food additives (E322: lecithin; E330: citric acid; E440: pectin; E471: mono- and diglycerides; E476: polyglycerol polyricinoleate; E500: sodium carbonates).

isfying $B \subseteq C(B)$; $B_1 \subseteq B_2$ implies $C(B_1) \subseteq C(B_2)$; $C(C(B)) = C(B)$. In our approach, a user background knowledge is represented by a closure operator C in Y and a formal concept $\langle A, B\rangle$ is considered "interesting" (compatible with background knowledge) if the intent B is C-closed, i.e. $B = C(B)$. Examples of C are provided below.

Definition 1. Let C be a closure operator in Y. A set $B \subseteq Y$ is called a C-interesting if $B = C(B)$. We put

$$\mathcal{B}_C(X, Y, I) = \{\langle A, B\rangle \in \mathcal{B}(X, Y, I) \,|\, B = C(B)\}.$$

$\langle A, B\rangle \in \mathcal{B}_C(X, Y, I)$ are called C-concepts (A and B are called C-extents and C-intents). $\mathrm{Ext}_C(X, Y, I)$ and $\mathrm{Int}_C(X, Y, I)$ denote the set of all C-extents and C-intents, respectively. ■

THEOREM 1. *$\mathcal{B}_C(X, Y, I)$ is a complete $\bigvee$-sublattice of $\mathcal{B}(X, Y, I)$. Conversely, every complete $\bigvee$-sublattice of $\mathcal{B}(X, Y, I)$ is of the form $\mathcal{B}_C(X, Y, I)$ for some C.* □

$\mathcal{B}_C(X, Y, I)$ may be computed directly (without computing the whole $\mathcal{B}(X, Y, I)$) using existing algorithms for computing fixpoints of closure operators [2] due to the following result (details postponed to full version). For any $B \subseteq Y$

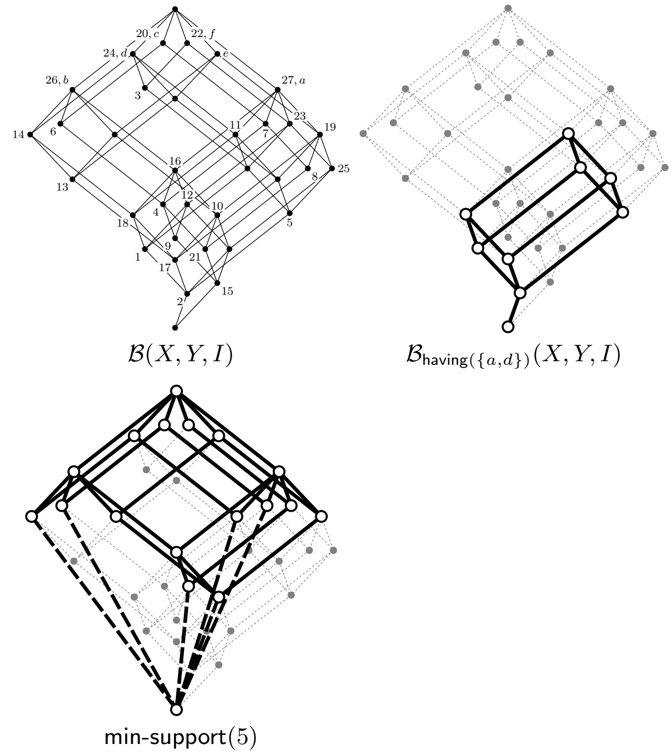

Figure 2: Concept lattices of data from Fig. 1.

define sets $B_j \subseteq Y$ ($j \in \mathbb{N}_0$) by

$$B_j = \begin{cases} B, & \text{if } j = 0, \\ C({B_{j-1}}^{\downarrow\uparrow}), & \text{if } j \geq 1. \end{cases}$$

Define an operator $cl\colon 2^Y \to 2^Y$ by

$$cl(B) = \textstyle\bigcup_{j=0}^{\infty} B_j.$$

THEOREM 2. *Let $\langle X, Y, I\rangle$ be a formal context, C be a closure operator in Y. Then cl is a closure operator such that* $\mathrm{fix}(cl) = \mathrm{Int}_C(X, Y, I)$. □

Required Attributes: $\mathsf{having}(Z)$ C defined by $C(B) = B \cup Z$ is a closure operator. C-interesting concepts then need to contain all attributes from Z. For instance, $C = \mathsf{having}(\{a, d\})$, $\mathcal{B}_C(X, Y, I)$ consists of concepts representing "products containing lecithin and mono- and diglycerides", see Fig. 2 (top-left).
Required Minimal Support: $\mathsf{min\text{-}support}(s)$ For a non-negative integer s (support), C defined by $C(B) = B$ if $|B^{\downarrow}| \geq s$ and $C(B) = Y$ otherwise, is a closure operator. Extents of C-interesting concepts contain at least s objects (note that the extents are just the closed frequent itemsets [3]). The case of $C = \mathsf{min\text{-}support}(5)$ is depicted in Fig. 2 (top-right).

1. REFERENCES

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

[2] Kuznetsov S., Obiedkov S.: Comparing performance of algorithms for generating concept lattices. *J. Exp. Theor. Artif. Int.*, **14**(2002), 189–216.

[3] Zaki M. J.: Mining non-redundant association rules. *Data Mining and Knowledge Discovery* **9**(2004), 223–248.