AISTA 2004 in Cooperation with the IEEE Computer Society Proceedings / ISBN : 2-9599776-8-8
15-18 November, Kirchberg - Luxembourg

Fast factorization by similarity in formal concept analysis

Radim Bělohlávek, Jiří Dvořák, Jan Outrata
Dept. Computer Science, Palacký University, Tomkova 40, CZ-779 00, Olomouc, Czech Republic
Email: {radim.belohlavek, jiri.dvorak, jan.outrata}@upol.cz

Abstract— **Formal concept analysis aims at extracting a hierarchical structure (so-called concept lattice) of clusters (so-called formal concepts) from object-attribute data tables. An important problem in applications of formal concept analysis is a possibly large number of clusters extracted from data. Factorization is one of the methods being used to cope with the number of clusters. We present an algorithm for computing a factor lattice of a concept lattice from the data and a user-specified similarity threshold a. The factor lattice is smaller than the original concept lattice and its size depends on the similarity threshold. The elements of the factor lattice are collections of clusters which are pairwise similar in degree at least a. The presented algorithm computes the factor lattice directly from the data, without first computing the whole concept lattice and then computing the collections of clusters. We present theoretical insight and examples for demonstration.**

I. Problem Setting and Preliminaries

A. Problem Setting

Formal concept analysis (FCA) [9] is a method of exploratory data analysis which aims at extracting a hierarchical structure of clusters from object-attribute data tables. The clusters (A, B), called formal concepts, consist of a collection A (concept extent) of objects and a collection B (concept intent) of attributes. Formal concepts can be partially ordered by a natural subconcept-superconcept relation. The resulting partially ordered set, called a concept lattice, can be visualized by a labeled Hasse diagram. The extent-intent definition of formal concepts goes back to traditional Port-Royal logic. Alternatively, formal concepts can be thought of as maximal rectangles contained in object-attribute data table. The attributes can be binary (0/1), fuzzy (degrees e.g. in [0,1], then the clusters are fuzzy as well), or more general, see e.g. [9], [2], [5], [12].

FCA has been applied in various fields, for instance in software engineering, reengineering problems (redesign of hierarchical structures), text classification (analyzing e-mail collections, classification of library items), psychology (development of concepts by children), civil engineering (system for checking dependencies in regulations), classification and systematizing of heuristic methods, physiology (color perception), preprocessing of data for reduction; see [1], [8], [13], [15] and [9] for references and further applications.

A direct user comprehension and interpretation of the partially ordered set of clusters (concept lattice) may be difficult due to a possibly large number of clusters extracted from the data table. A way to go is to consider, instead of the whole concept lattice, its suitable factor lattice which can be considered a granularized version of the original concept lattice: Its elements are classes of clusters and the factor lattice is smaller. A method of factorization by a so-called compatible reflexive and symmetric relation (a tolerance) on the set of clusters was described in [9]. Interpreting the tolerance relation as similarity on clusters/concepts, the elements of the factor lattice are classes of pairwise similar clusters/concepts. The specification of the tolerance relation is, however, left to the user. In [2], a parametrized method of factorization for data with fuzzy attributes was presented: the tolerance relation is induced by a threshold (parameter of factorization) specified by a user. Using a suitable measure of similarity degree of clusters/concepts (see later), the method does the following. Given a threshold a (e.g. a number from $[0, 1]$), the elements of the factor lattice are similarity blocks determined by a (maximal collections of formal concepts which are pairwise similar in degree at least a). The smaller a, the smaller the factor lattice (i.e. the larger the reduction). In the extreme cases $a = 0$ or $a = 1$, the factor lattice degenerates and contains just one element (for $a = 0$) or is the same (up to an isomorphism) as the original concept lattice (for $a = 1$). For a between 0 and 1, the factor lattice provides a granular view on the original concept lattice (granules are the similarity blocks).

In order to compute the factor lattice (directly by definition), we have to compute the whole concept lattice (this can be done by an algorithm with a polynomial time delay, see [4]) and then compute all the similarity blocks, i.e. elements of the factor lattice (again, this can be accomplished by an algorithm with polynomial time delay, see later).

In this paper, we present a way to compute the factor lattice directly from data (input: data table and user-specified threshold a; output: factor lattice, i.e. similarity blocks given by a). The resulting algorithm is significantly faster than computing first the whole concept lattice and then computing the similarity blocks.

The paper is organized as follows. Section I-B is a brief survey of related work. Section I-C presents preliminaries on fuzzy sets and formal concept analysis of data with fuzzy attributes. In Section II we present our approach. Section III presents experiments and demonstrates the speed-up.

B. Related work

Due to the limited scope, we only briefly mention: A variety of methods to help to reduce/manage the size of a concept lattice is presented in [9]. Except of the above-mentioned factorization, there are several methods of decomposition described in [9]. A method based on selection (by user) of relevant attributes and dealing with only the corresponding part of a concept lattice is presented in [7]. In [6], the authors face the problem of a large concept lattice by computing and visualizing only its relevant local part which is used for browsing based on a user query.

C. Preliminaries

Fuzzy sets and fuzzy logic

We assume basic familiarity with fuzzy logic and fuzzy sets [11]. An element may belong to a fuzzy set in an intermediate degree not necessarily being 0 or 1. Formally, a fuzzy set A in a universe X is a mapping assigning to each $x \in X$ a truth degree $A(x) \in L$ where L is some partially ordered set of truth degrees containing at least 0 (full falsity) and 1 (full truth). L needs to be equipped with logical connectives, e.g. $\otimes$ (fuzzy conjunction), $\rightarrow$ (fuzzy implication), etc. L together with logical connectives forms a structure $\mathbf{L}$ of truth degrees. We assume that $\mathbf{L}$ forms a so-called residuated lattice in which arbitrary infima $\bigwedge$ and suprema $\bigvee$ exist.

The most applied set of truth values is the real interval $[0,1]$; with $a \wedge b = \min(a,b)$, $a \vee b = \max(a,b)$, and with three important pairs of fuzzy conjunction and fuzzy implication: Łukasiewicz ($a \otimes b = \max(a+b-1,0)$, $a \rightarrow b = \min(1-a+b,1)$), minimum ($a \otimes b = \min(a,b)$, $a \rightarrow b = 1$ if $a \leq b$ and $= b$ else), and product ($a \otimes b = a \cdot b$, $a \rightarrow b = 1$ if $a \leq b$ and $= b/a$ else). Another possibility is to take a finite chain $\{a_0 = 0, a_1, \ldots, a_n = 1\}$ ($a_0 < \cdots < a_n$) equipped with Łukasiewicz structure ($a_k \otimes a_l = a_{\max(k+l-n,0)}$, $a_k \rightarrow a_l = a_{\min(n-k+l,n)}$) or minimum ($a_k \otimes a_l = a_{\min(k,l)}$, $a_k \rightarrow a_l = a_n$ for $a_k \leq a_l$ and $a_k \rightarrow a_l = a_l$ otherwise).

The set of all fuzzy sets (or $\mathbf{L}$-sets) in X is denoted L^X. For fuzzy sets A, B in X we put $A \subseteq B$ (A is a subset of B) if for each $x \in X$ we have $A(x) \leq B(x)$. More generally, the degree $S(A,B)$ to which A is a subset of B is defined by $S(A,B) = \bigwedge_{x \in X} A(x) \rightarrow B(x)$. Then, $A \subseteq B$ means $S(A,B) = 1$.

Formal concept analysis of data with fuzzy attributes Let X and Y be sets of objects and attributes, respectively, I be a fuzzy relation between X and Y. That is, $I : X \times Y \rightarrow L$ assigns to each $x \in X$ and each $y \in Y$ a truth degree $I(x,y) \in L$ to which object x has attribute y. The triplet $\langle X, Y, I \rangle$ is called a formal fuzzy context (a data table with fuzzy attributes).

For fuzzy sets $A \in L^X$ and $B \in L^Y$, define fuzzy sets $A^{\uparrow} \in L^Y$ and $B^{\downarrow} \in L^X$ by

$$A^{\uparrow}(y) = \bigwedge_{x \in X} (A(x) \rightarrow I(x,y)) \tag{1}$$

and

$$B^{\downarrow}(x) = \bigwedge_{y \in Y} (B(y) \rightarrow I(x,y)). \tag{2}$$

Using basic rules of predicate fuzzy logic, $A^{\uparrow}(y)$ is the truth degree of the fact "y is shared by all objects from A" and $B^{\downarrow}(x)$ is the truth degree of the fact "x has all attributes from B". Putting

$$\mathcal{B}(X,Y,I) = \{\langle A, B \rangle \mid A^{\uparrow} = B,\ B^{\downarrow} = A\},$$

$\mathcal{B}(X,Y,I)$ is the set of all pairs $\langle A, B \rangle$ such that (a) A is the collection of all objects that have all the attributes of (the intent) B and (b) B is the collection of all attributes that are shared by all the objects of (the extent) A. Elements of $\mathcal{B}(X,Y,I)$ are called formal concepts of $\langle X,Y,I \rangle$ (interesting clusters in data); $\mathcal{B}(X,Y,I)$ is called the concept lattice given by $\langle X,Y,I \rangle$ (collection of all interesting clusters). Both the extent A and the intent B of a formal concept $\langle A, B \rangle$ are in general fuzzy sets and represent collections of objects and attributes which are covered by the concept.

Putting

$$\langle A_1, B_1 \rangle \leq \langle A_1, B_1 \rangle \text{ iff } A_1 \subseteq A_2 (\text{iff } B_1 \supseteq B_2) \tag{3}$$

for $\langle A_1, B_1 \rangle, \langle A_2, B_2 \rangle \in \mathcal{B}(X,Y,I)$, $\leq$ models the subconcept-superconcept hierarchy in $\mathcal{B}(X,Y,I)$. That is, being more general means to apply to a larger collection of objects and to cover a smaller collection of attributes.

The following is a part of characterization of the structure of fuzzy concept lattices, see [3].

Theorem 1: The set $\mathcal{B}(X,Y,I)$ is under $\leq$ a complete lattice where infima and suprema are given by

$$\bigwedge_{j \in J} \langle A_j, B_j \rangle = \langle \bigcap_{j \in J} A_j, (\bigcup_{j \in J} B_j)^{\downarrow\uparrow} \rangle \ , \tag{4}$$

$$\bigvee_{j \in J} \langle A_j, B_j \rangle = \langle (\bigcup_{j \in J} A_j)^{\uparrow\downarrow}, \bigcap_{j \in J} B_j \rangle \ . \tag{5}$$

II. Fast factorization by similarity

A. Factorization by similarity

In this section, we recall the parametrized method of factorization introduced in [2] to which we refer for details. Given a fuzzy context $\langle X, Y, I \rangle$, introduce a binary fuzzy relation $\approx$ on the set $\mathcal{B}(X,Y,I)$ of all formal concepts of $\langle X,Y,I \rangle$ by

$$(\langle A_1, B_1 \rangle \approx \langle A_2, B_2 \rangle) = \bigwedge_{x \in X} A_1(x) \leftrightarrow A_2(x)$$

for $\langle A_i, B_i \rangle \in \mathcal{B}(X,Y,I)$, $i = 1, 2$, where $\bigwedge$ is the infimum (minimum in most cases) and $\leftrightarrow$ is a connective of fuzzy equivalence defined by $a \leftrightarrow b = (a \rightarrow b) \wedge (b \rightarrow a)$. $(\langle A_1, B_1 \rangle \approx \langle A_2, B_2 \rangle)$ is called the degree of similarity of $\langle A_1, B_1 \rangle$ and $\langle A_2, B_2 \rangle$. It is easily seen that it is just the truth degree of "for each object x: x is covered by A_1 iff x is covered by A_2". It can be shown that

$$(\langle A_1, B_1 \rangle \approx \langle A_2, B_2 \rangle) = \bigwedge_{y \in Y} B_1(y) \leftrightarrow B_2(y).$$

Therefore, measuring similarity of formal concepts via extents A_i coincides with measuring similarity via intents B_i, corresponnding to the duality of extent/intent view.

Given a truth degree $a \in L$ (a threshold specified by a user), consider the thresholded relation ${}^a\!\approx$ on $\mathcal{B}(X, Y, I)$ defined by

$$(\langle A_1, B_1\rangle, \langle A_2, B_2\rangle) \in {}^a\!\approx \quad \text{iff} \quad (\langle A_1, B_1\rangle \approx \langle A_2, B_2\rangle) \geq a.$$

That is, ${}^a\!\approx$ is the relation "being similar in degree at least a". ${}^a\!\approx$ is reflexive and symmetric, but need not be transitive (it is transitive if $a \otimes b = a \wedge b$ holds true in $\mathbf{L}$, i.e. if we use minimum as fuzzy conjunction). Call a subset B of $\mathcal{B}(X, Y, I)$ a ${}^a\!\approx$-block if it is a maximal subset of $\mathcal{B}(X, Y, I)$ such that each two formal concepts from B are similar in degree at least a (the notion of a ${}^a\!\approx$-block generalizes that of an equivalence class: if ${}^a\!\approx$ is an equivalence relation, ${}^a\!\approx$-blocks are exactly the equivalence classes). Denote by $\mathcal{B}(X, Y, I)/{}^a\!\approx$ the collection of all ${}^a\!\approx$-blocks.

It can be shown that ${}^a\!\approx$ are special intervals in the concept lattice $\mathcal{B}(X, Y, I)$. In detail, for a formal concept $\langle A, B\rangle \in \mathcal{B}(X, Y, I)$, put

$$\langle A, B\rangle_a := \bigwedge\{\langle A', B'\rangle \mid (\langle A, B\rangle, \langle A', B'\rangle) \in {}^a\!\approx\}$$

and

$$\langle A, B\rangle^a := \bigvee\{\langle A', B'\rangle \mid (\langle A, B\rangle, \langle A', B'\rangle) \in {}^a\!\approx\}.$$

That is $\langle A, B\rangle_a$ and $\langle A, B\rangle^a$ are the infimum and the supremum of the set of all formal concepts which are similar to $\langle A, B\rangle$ in degree at least a. $\ldots_a$ and $\ldots^a$ are important in description of ${}^a\!\approx$-blocks. Namely, we have the following.

Lemma 2: ${}^a\!\approx$-blocks are exactly intervals of $\mathcal{B}(X, Y, I)$ of the form $[\langle A, B\rangle_a, (\langle A, B\rangle_a)^a]$, i.e.

$$\mathcal{B}(X, Y, I)/{}^a\!\approx \; = \{[\langle A, B\rangle_a, (\langle A, B\rangle_a)^a] \mid \langle A, B\rangle \in \mathcal{B}(X, Y, I)\}.$$

Note that an interval with lower bound $\langle A_1, B_1\rangle$ and upper bound $\langle A_2, B_2\rangle$ is the subset

$$[\langle A_1, B_1\rangle, \langle A_2, B_2\rangle] = \{\langle A, B\rangle \mid \langle A_1, B_1\rangle \leq \langle A, B\rangle \leq \langle A_2, B_2\rangle\}.$$

Now, define a partial order $\preceq$ on blocks of $\mathcal{B}(X, Y, I)/{}^a\!\approx$ by

$$[c_1, c_2] \preceq [d_1, d_2] \text{ iff } \quad c_1 \leq d_1 (\text{iff } c_2 \leq d_2)$$

where $[c_1, c_2], [d_1, d_2] \in \mathcal{B}(X, Y, I)/{}^a\!\approx$, i.e. c_1, c_2, d_1, d_2 are suitable formal concepts from $\mathcal{B}(X, Y, I)$ and $c_i \leq d_i$ denotes that in $\mathcal{B}(X, Y, I)$, c_i is under (a subconcept of) d_i. Then we have

Theorem 3: $\mathcal{B}(X, Y, I)/{}^a\!\approx$ equipped with $\preceq$ is a partially ordered set which is a complete lattice, the so-called factor lattice of $\mathcal{B}(X, Y, I)$ by similarity $\approx$ and a threshold a.

Elements of $\mathcal{B}(X, Y, I)/{}^a\!\approx$ can be seen as similarity-based granules of formal concepts/clusters from $\mathcal{B}(X, Y, I)$. $\mathcal{B}(X, Y, I)/{}^a\!\approx$ thus provides a granular view on (the possibly large) $\mathcal{B}(X, Y, I)$. Note also that if ${}^a\!\approx$ is transitive then it is a congruence relation on $\mathcal{B}(X, Y, I)$ and $\mathcal{B}(X, Y, I)/{}^a\!\approx$ is the usual factor lattice modulo a congruence. For further details and properties of $\mathcal{B}(X, Y, I)/{}^a\!\approx$ we refer to [2].

B. Computing the factor lattice $\mathcal{B}(X, Y, I)/{}^a\!\approx$ directly from input data

In order to compute $\mathcal{B}(X, Y, I)/{}^a\!\approx$ using definition and Lemma 2, one has (1) to compute the whole concept lattice $\mathcal{B}(X, Y, I)$ and then (2) to compute ${}^a\!\approx$-blocks on $\mathcal{B}(X, Y, I)$, which can be quite demanding. We are going to propose a way to compute $\mathcal{B}(X, Y, I)/{}^a\!\approx$ directly from input data. It will turn out that our algorithm has a polynomial time delay (see [10]).

We need some auxiliary results. For basic properties of fuzzy concept lattices we refer e.g. to [2], [3]. For a fuzzy set C in U and $a \in L$, the fuzzy sets $a \to C$ and $a \otimes C$ in U are defined by $(a \to C)(u) = a \to C(u)$ and $(a \otimes C)(u) = a \otimes C(u)$ for each $u \in U$. For fuzzy sets C, D in U, put $(C \approx D) = \bigwedge_{u \in U} C(u) \leftrightarrow D(u)$. Furthermore, we call a fuzzy set A in X an extent if there is a fuzzy set B in Y such that $\langle A, B\rangle \in \mathcal{B}(X, Y, I)$ (similarly, B is an intent if there is A with $\langle A, B\rangle \in \mathcal{B}(X, Y, I)$).

Lemma 4: If A is an extent then so is $a \to A$; similarly, if B is an intent then so is $a \to B$.

Proof: We prove the assertion for extents. Let A be an extent, i.e. $\langle A, B\rangle \in \mathcal{B}(X, Y, I)$ for some B. We have to show that $\langle a \to A, B'\rangle \in \mathcal{B}(X, Y, I)$. It is easy to see that it suffices to show that $a \to A = (a \to A)^{\uparrow\downarrow}$ (since then $\langle a \to A, (a \to A)^{\uparrow}\rangle$ is a formal concept). Since $a \to A \subseteq (a \to A)^{\uparrow\downarrow}$ is always the case, we have to show $(a \to A)^{\uparrow\downarrow} \subseteq a \to A$ which holds iff $(a \to A)^{\uparrow\downarrow}(x) \leq a \to A(x)$ for each $x \in X$. Using adjointness, the latter is equivalent to $a \leq (a \to A)^{\uparrow\downarrow}(x) \to A(x)$. Since $(a \to A)^{\uparrow\downarrow}(x) \to A(x) \geq \bigwedge_{x \in X}(a \to A)^{\uparrow\downarrow}(x) \leftrightarrow A(x) = ((a \to A)^{\uparrow\downarrow} \approx A)$, it suffices to show $a \leq ((a \to A)^{\uparrow\downarrow} \approx A)$ which can be shown using $(A_1 \approx A_2) \leq (A_1^{\uparrow} \approx A_2^{\uparrow}) \leq (A_1^{\uparrow\downarrow} \approx A_2^{\uparrow\downarrow})$ (see [2]) by a straightforward verification. ■

The next lemma shows that for a formal concept $\langle A, B\rangle$, $\langle A, B\rangle_a$ and $\langle A, B\rangle^a$ (defined as infimum and supremum of all formal concepts similar to $\langle A, B\rangle$ in degree at least a) can be computed from $\langle A, B\rangle$ directly.

Lemma 5: For $\langle A, B\rangle \in \mathcal{B}(X, Y, I)$, we have (a) $\langle A, B\rangle_a = \langle (a \otimes A)^{\uparrow\downarrow}, a \to B\rangle$ and (b) $\langle A, B\rangle^a = \langle (a \to A), (a \otimes B)^{\downarrow\uparrow}\rangle$.

Proof: Due to duality we verify only (a). First, one can show that (a1) $(a \otimes A)^{\uparrow\downarrow}$ is an extent of a formal concept $\langle (a \otimes A)^{\uparrow\downarrow}, D\rangle$ which is similar to $\langle A, B\rangle$ in degree at least a, and (a2) if $\langle C, F\rangle$ is a formal concept similar to $\langle A, B\rangle$ in degree at least a then $\langle (a \otimes A)^{\uparrow\downarrow}, D\rangle \leq \langle C, F\rangle$. It remains to show $D = a \to B$. To this end, it is sufficient to see that (a3) $a \to B$ is an intent of a concept c which is similar to $\langle A, B\rangle$ in degree at least a, and (a4) if $\langle C, F\rangle$ is a concept similar to $\langle A, B\rangle$ in degree at least a then $c \leq \langle C, F\rangle$. Indeed, from (a3) and (a4) it follows that $a \to B$ is the intent of the least formal concept similar to $\langle A, B\rangle$ in degree at least a, i.e. $a \to B = D$. We verify (a3) and (a4). (a3): By Lemma 4, $a \to B$ is an intent. Using adjointness we easily get $a \leq (B \approx a \to B) = (\langle A, B\rangle \approx c)$, proving (a3). (a4): We need to show

$F \subseteq a \to B$. Since $a \le (\langle A, B\rangle \approx \langle C, F\rangle) = (B \approx F)$, adjointness gives $a \otimes F \subseteq B$ and then $F \subseteq a \to B$, proving (a4). The proof is complete. ■

Thus we have $(\langle A, B\rangle_a)^a = \langle a \to (a \otimes A)^{\uparrow\downarrow}, (a \otimes (a \to B))^{\downarrow\uparrow}\rangle$.

Lemma 6: For $\langle A, B\rangle \in \mathcal{B}(X, Y, I)$ we have $\langle A, B\rangle_a = (((\langle A, B\rangle_a)^a)_a$.

Proof: First we show that for every formal concepts c and d we have (1) $c \le d$ implies $c_a \le d_a$, (2) $c \le d$ implies $c^a \le d^a$, (3) $c \le (c_a)^a$, (4) $c \ge (c^a)_a$. (1): Recall that $c_a = \bigwedge\{f \in \mathcal{B}(X, Y, I) \mid \langle c, f\rangle \in {}^a\!\approx\}$. We need to show that if $\langle d, f\rangle \in {}^a\!\approx$ then $c_a \le f$. Thus suppose $\langle d, f\rangle \in {}^a\!\approx$. From $\langle c, c\rangle \in {}^a\!\approx$ and from the fact that ${}^a\!\approx$ is a compatible tolerance on $\mathcal{B}(X, Y, I)$ we get $a \le \langle c \wedge d, c \wedge f\rangle = \langle c, c \wedge f\rangle$ and so $f \ge c \wedge f \ge c_a$, proving (1). (2) can be proved analogously. (3) and (4) are obvious.

Let $c = \langle A, B\rangle$. By (3), $c \le (c_a)^a$ and so $c_a \le ((c_a)^a)_a$ by (1). Applying (4) to c_a we get $c_a \ge ((c_a)^a)_a$, proving $c_a = ((c_a)^a)_a$. ■

By Lemma 6, if a ${}^a\!\approx$-block $[c_1, c_2]$ is generated by $\langle A, B\rangle \in \mathcal{B}(X, Y, I)$, i.e. $c_1 = \langle A, B\rangle_a$, $c_2 = (\langle A, B\rangle_a)^a$, then it is also generated by c_2, i.e. $c_1 = (c_2)_a$ and $c_2 = ((c_2)_a)^a$.

Therefore, ${}^a\!\approx$-blocks $[c_1, c_2]$ are uniquely given by their suprema c_2. Moreover, since each formal concept $c_2 = \langle A, B\rangle$ is uniquely given by A (namely, $B = A^\uparrow$), ${}^a\!\approx$-blocks are uniquely given by extents of their suprema. Therefore, denote the set of all extents of suprema of ${}^a\!\approx$-blocks by $\mathrm{ESB}(a)$, i.e.

$$\mathrm{ESB}(a) = \{A \in L^X \mid \langle A, B\rangle \in \mathcal{B}(X, Y, I) \text{ and } [\langle A, B\rangle_a, \langle A, B\rangle] \in \mathcal{B}(X, Y, I)/{}^a\!\approx\}.$$

We are going to present the main result. Let $C : A \to C(A)$ be a mapping (assigning a fuzzy set $C(A)$ in X to a fuzzy set A in X). A fixed point of C is any fuzzy set A in X such that $A = C(A)$. Let $\mathrm{fix}(C)$ denote the set of all fixed points of C, i.e.

$$\mathrm{fix}(C) = \{A \in L^X \mid A = C(A)\}.$$

Recall (see e.g. [4]) that C is called a fuzzy closure operator in X if

$$A \subseteq C(A) \tag{6}$$
$$S(A_1, A_2) \le S(C(A_1), C(A_2)) \tag{7}$$
$$C(A) = C(C(A)) \tag{8}$$

for any $A, A_1, A_2 \in L^X$.

Theorem 7: Given input data $\langle X, Y, I\rangle$ and a threshold $a \in L$, a mapping C_a sending a fuzzy set A in X to a fuzzy set $a \to (a \otimes A)^{\uparrow\downarrow}$ in X is a fuzzy closure operator in X for which $\mathrm{fix}(C_a) = \mathrm{ESB}(a)$.

Therefore, A is a fixed point of C_a if and only if A is the extent of some formal concept c_2 which is the supremum of some ${}^a\!\approx$-block $[c_1, c_2] \in \mathcal{B}(X, Y, I)/{}^a\!\approx$.

Proof: First, one can verify that C_a is a fuzzy closure operator (we omit the proof and refer to a forthcoming paper).

Second, we verify $\mathrm{fix}(C_a) = \mathrm{ESB}(a)$. Let $A \in \mathrm{fix}(C_a)$. Then $A = C_a(A) = a \to (a \otimes A)^{\uparrow\downarrow}$. By Lemma 5, $[\langle A, A^\uparrow\rangle_a, \langle A, A^\uparrow\rangle]$ is a ${}^a\!\approx$-block and so A is the extent of a supremum of a block, i.e. $A \in \mathrm{ESB}(a)$. Conversely, let $A \in \mathrm{ESB}(a)$. Then $[\langle A, A^\uparrow\rangle_a, \langle A, A^\uparrow\rangle]$ is an ${}^a\!\approx$-block and so $(\langle A, A^\uparrow\rangle_a)^a = \langle A, A^\uparrow\rangle$. Lemma 5 now gives $A = a \to (a \otimes A)^{\uparrow\downarrow}$, i.e. $A = C_a(A)$ verifying $A \in \mathrm{fix}(C_a)$. ■

Remark 1: Suppose we can compute $\mathrm{fix}(C_a)$ (we will se later how to do it). By Theorem 7 and the above considerations, going through $\mathrm{fix}(C_a)$ and computing for each $A \in \mathrm{fix}(C_a)$ the corresponding $[\langle A, A^\uparrow\rangle_a, \langle A, A^\uparrow\rangle] = [\langle (a \otimes A)^{\uparrow\downarrow}, a \to A^\uparrow\rangle, \langle A, A^\uparrow\rangle]$ generates all ${}^a\!\approx$-blocks of $\mathcal{B}(X, Y, I)/{}^a\!\approx$.

Remark 2: Strictly speaking, proceeding the just-described way, we do not generate the ${}^a\!\approx$-blocks $[c_1, c_2] \in \mathcal{B}(X, Y, I)/{}^a\!\approx$, i.e. we do not generate ${}^a\!\approx$-blocks $[c_1, c_2]$ as collections of formal concepts $[c_1, c_2] = \{\langle A, B\rangle \mid c_1 \le \langle A, B\rangle \le c_2\}$. For us, generating a ${}^a\!\approx$-block $[c_1, c_2]$ means generating the boundary formal concepts $c_1, c_2 \in \mathcal{B}(X, Y, I)$. This is, however, in acordance with the purpose of the factorization of $\mathcal{B}(X, Y, I)$: We are looking for a granular view which is more concise than $\mathcal{B}(X, Y, I)$ itself.

Let us turn to the problem of generating $\mathrm{fix}(C_a)$. To this end, we can use the algorithm for generating all formal concepts of a given fuzzy context described in [4]. Indeed, the algorithm described in [4] generates extents of all formal concepts from $\mathcal{B}(X, Y, I)$. Now, the extents of formal concepts are exactly the fixed points of a fuzzy closure operator C defined by $C(A) = A^{\uparrow\downarrow}$. Furthermore, as one can check, as the algorithm uses only properties of fuzzy closure operators, it is in fact an algorithm for generating the set of fixed points of a fuzzy closure operator. Adapting the algorithm for our situation and taking in account Remark 1, we get the following algorithm for computing ${}^a\!\approx$-blocks $[c_1, c_2]$, i.e. elements of $\mathcal{B}(X, Y, I)/{}^a\!\approx$:

Suppose $X = \{1, 2, \dots, n\}$; $L = \{0 = a_1 < a_2 < \cdots < a_k = 1\}$ (the assumption that L is linearly ordered is in fact not essential). For $i, r \in \{1, \dots, n\}$, $j, s \in \{1, \dots, k\}$ we put

$$(i, j) \le (r, s) \quad \text{iff} \quad i < r \quad \text{or} \quad i = r, a_j \ge a_s.$$

In the following, we will freely refer to a_i just by i, thus not distinguish between $X \times L$ and $\{1, \dots, n\} \times \{1, \dots, k\}$, i.e. we denote $(i, a_j) \in X \times L$ also simply by (i, j).

For $A \in L^X$, $(i, j) \in X \times L$, put

$$A \oplus (i, j) := C_a((A \cap \{1, 2, \dots, i-1\}) \cup \{{}^{a_j}\!/i\}).$$

Here, $A \cap \{1, 2, \dots, i-1\}$ is the intersection of a fuzzy set A and the ordinary set $\{1, 2, \dots, i-1\}$, i.e. $(A \cap \{1, 2, \dots, i-1\})(x) = A(x)$ for $x < i$ and $(A \cap \{1, 2, \dots, i-1\})(x) = 0$ otherwise. Furthermore, for $A, C \in L^X$, put

$$A <_{(i,j)} C \text{ iff } A \cap \{1, \dots, i-1\} = C \cap \{1, \dots, i-1\} \text{ and } A(i) < C(i) = a_j.$$

Finally,

$$A < C \quad \text{iff} \quad A <_{(i,j)} C \quad \text{for some } (i, j).$$

The algorithm is based on the following theorem (see [4]).

Theorem 8: The least fixed point A^+ which is greater (w.r.t. $<$) than a given $A \in L^X$ is given by

$$A^+ = A \oplus (i, j)$$

where (i, j) is the greatest one with $A <_{(i,j)} A \oplus (i, j)$.

The algorithm for generating ${}^a\!\approx$-blocks follows.

INPUT: $\langle X, Y, I\rangle$ (data table with fuzzy attributes), $a \in L$ (similarity threshold)
OUTPUT: $\mathcal{B}(X, Y, I)/{}^a\!\approx$ (${}^a\!\approx$-blocks $[c_1, c_2]$)

```
/* Algorithm */
A := ∅
while A ≠ X do
  A := A⁺
  store([⟨(a ⊗ A)↑↓, a → A↑⟩, ⟨A, A↑⟩])
```

As argued in [4], generating $\text{fix}(C_a)$ has polynomial time delay complexity (i.e., given a fixed point, the next one is generated in time polynomial in terms of size of the input $\langle X, Y, I\rangle$ [10]). Since generating a ${}^a\!\approx$-block $[\langle (a \otimes A)^{\uparrow\downarrow}, a \rightarrow A^{\uparrow}\rangle, \langle A, A^{\uparrow}\rangle]$ from A takes a polynomial time, our algorithm is of polynomial time delay complexity as well.

III. Examples and Experiments

Due to the limited scope, we demonstrate our algorithm on a data table (fuzzy context) from Tab. I for which we consider various parameters a (threshold) and some characteristics for comparison. The data table contains countries (objects from X) and some of their economic characteristics (attributes from Y). The original values of the characteristics are scaled to interval $[0, 1]$ so that the characteristics can be considered as fuzzy attributes. Tab. II summarizes the effect of our algorithm and some related characteristics when using Łukasiewicz fuzzy logical connectives. The whole concept lattice $\mathcal{B}(X, Y, I)$ contains 774 formal concepts, computing $\mathcal{B}(X, Y, I)$ using the polynomial time delay algorithm from [4] takes 2292ms. The columns correspond to different threshold values $a = 0.2, 0.4, 0.6, 0.8$. Entries "size $|\mathcal{B}(X, Y, I)/{}^a\!\approx|$" contain the number of ${}^a\!\approx$-blocks; "naive algorithm (ms)" contain the time in ms for computing $\mathcal{B}(X, Y, I)/{}^a\!\approx$ by first generating $\mathcal{B}(X, Y, I)$ and subsequently generating the ${}^a\!\approx$-blocks by producing $[\langle A, B\rangle_a, (\langle A, B\rangle_a)^a]$; "our algorithm (ms)" contain the time in ms for computing $\mathcal{B}(X, Y, I)/{}^a\!\approx$ by our algorithm; "reduction $|\mathcal{B}(X, Y, I)/{}^a\!\approx|/|\mathcal{B}(X, Y, I)|$" contain the reduction factors of the size of the concept lattice; "time reduction" contain "our algorithm (ms)" divided by "naive algorithm (ms)" (1/"time reduction" is thus the speedup). Fig. 1 and Fig. 2 contain graphs depicting reduction $|\mathcal{B}(X, Y, I)/{}^a\!\approx|/|\mathcal{B}(X, Y, I)|$ and time reduction from Tab. II.

The example demonstrates that smaller thresholds lead to larger reduction (in time and size of the concept lattice). Furthermore, we can see that the time needed for computing the factor lattice $\mathcal{B}(X, Y, I)/{}^a\!\approx$ is smaller than time for computing the original concept lattice $\mathcal{B}(X, Y, I)$

Tab. III, Fig. 3, and Fig. 4 show the same characteriztics when using the minimum-based fuzzy logical operations.

Finally, we demonstrate the effects on an example of data table from Tab. IV with a finer distribution of thresholds, $a = 0.1, 0.2, \ldots, 0.9$. Using Łukasiewicz fuzzy logical operations, the characteristics are the same as for the above example and are depicted in Fig. 5 and Fig. 6.

TABLE I
Data table (fuzzy context).

	1	2	3	4	5	6	7
1 Czech	0.4	0.4	0.6	0.2	0.2	0.4	0.2
2 Hungary	0.4	1.0	0.4	0.0	0.0	0.4	0.2
3 Poland	0.2	1.0	1.0	0.0	0.0	0.0	0.0
4 Slovakia	0.2	0.6	1.0	0.0	0.2	0.2	0.2
5 Austria	1.0	0.0	0.2	0.2	0.2	1.0	1.0
6 France	1.0	0.0	0.6	0.4	0.4	0.6	0.6
7 Italy	1.0	0.2	0.6	0.0	0.2	0.6	0.4
8 Geramny	1.0	0.0	0.6	0.2	0.2	1.0	0.6
9 UK	1.0	0.2	0.4	0.0	0.2	0.6	0.6
10 Japan	1.0	0.0	0.4	0.2	0.2	0.4	0.2
11 Canada	1.0	0.2	0.4	1.0	1.0	1.0	1.0
12 USA	1.0	0.2	0.4	1.0	1.0	0.2	0.4

attributes: 1 - Gross Domestic Product per capita (USD), 2 - Consumer Price Index (1995=100) , 3 - Unemployment Rate (percent - ILO), 4 - Production of electricity per capita (kWh), 5 - Energy consumption per capita (GJ), 6 - Export per capita (USD), 7 - Import per capita (USD)

TABLE II
Łukasiewicz fuzzy logical connectives, $\mathcal{B}(X, Y, I)$ of data from Tab. I: $|\mathcal{B}(X, Y, I)| = 774$, time for computing $\mathcal{B}(X, Y, I)$ = 2292 ms; table entries for thresholds $a = 0.2, 0.4, 0.6, 0.8$.

	0.2	0.4	0.6	0.8
size $\lvert\mathcal{B}(X, Y, I)/{}^a\!\approx\rvert$	8	57	193	423
naive algorithm (ms)	8995	9463	8573	9646
our algorithm (ms)	23	214	383	1517
reduction $\lvert\mathcal{B}(X, Y, I)/{}^a\!\approx\rvert/\lvert\mathcal{B}(X, Y, I)\rvert$	0.010	0.073	0.249	0.546
time reduction	0.002	0.022	0.044	0.157

Acknowledgment

Radim Bělohlávek gratefully acknowledges support by grant No. 201/02/P076 of the Grant Agency of the Czech Republic.

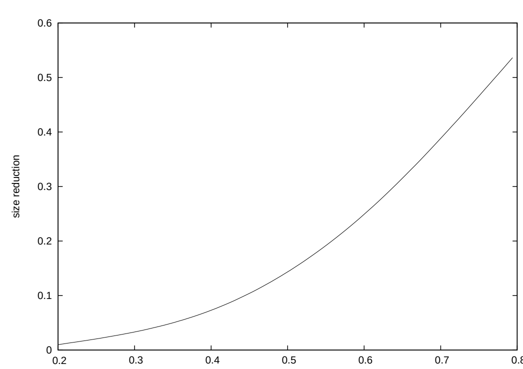

Fig. 1. Reduction $|\mathcal{B}(X, Y, I)/{}^a\!\approx|/|\mathcal{B}(X, Y, I)|$ from Tab. II.

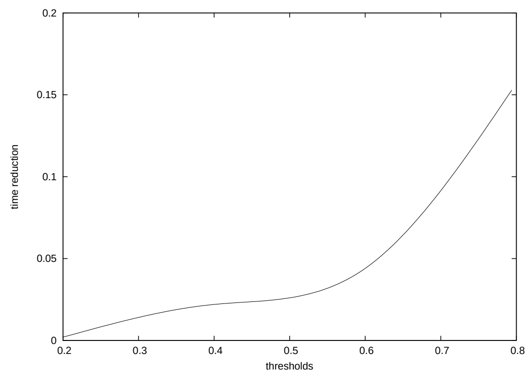

Fig. 2. Time reduction from Tab. II.

TABLE III

MINIMUM-BASED FUZZY LOGICAL CONNECTIVES, $\mathcal{B}(X, Y, I)$ OF DATA FROM TAB. I: $|\mathcal{B}(X, Y, I)| = 304$, TIME FOR COMPUTING $\mathcal{B}(X, Y, I) = 341$ MS; TABLE ENTRIES FOR THRESHOLDS $a = 0.2, 0.4, 0.6, 0.8$.

	0.2	0.4	0.6	0.8
size $\lvert\mathcal{B}(X, Y, I)/^a\!\approx\rvert$	8	64	194	304
naive algorithm (ms)	1830	1634	3787	4440
our algorithm (ms)	23	106	431	1568
reduction $\lvert\mathcal{B}(X, Y, I)/^a\!\approx\rvert / \lvert\mathcal{B}(X, Y, I)\rvert$	0.026	0.210	0.638	1.000
time reduction	0.012	0.064	0.113	0.353

REFERENCES

[1] G. Ammons, D. Mandelin, R. Bodik, J. R. Larus. Debugging temporal specifications with concept analysis. In *Proc. ACM SIGPLAN'03 Conference on Programming Language Design and Implementation*, pages 182–195, San Diego, CA, June 2003.

[2] R Bělohlávek. Similarity relations in concept lattices. *J. Logic Comput.* 10(6):823–845, 2000.

[3] R Bělohlávek. Concept lattices and order in fuzzy logic. *Annals of Pure and Applied Logic* (to appear, 22 pp.).

[4] Bělohlávek R.: Getting maximal rectangular submatrices from [0,1]-valued object-attribute tables: algorithms for fuzzy concept lattices (submitted). Preliminary version appeared in *Proc. Fourth Int. Conf. on Recent Advances in Soft Computing*. Nottingham, United Kingdom, 12-13 December, 2002, pp. 200–205.

[5] A. Burusco, R. Fuentes-Gonzáles. The study of the L-fuzzy concept lattice. *Mathware & Soft Computing*, 3:209–218, 1994.

[6] C. Carpineto, R. Romano. A lattice conceptual clustering system and its application to browsing retrieval. *Machine Learning* 24:95–122, 1996.

[7] R. Cole, P. Eklund. Scalability in formal context analysis: a case study using medical texts. *Computational Intelligence* 15:11–27, 1999.

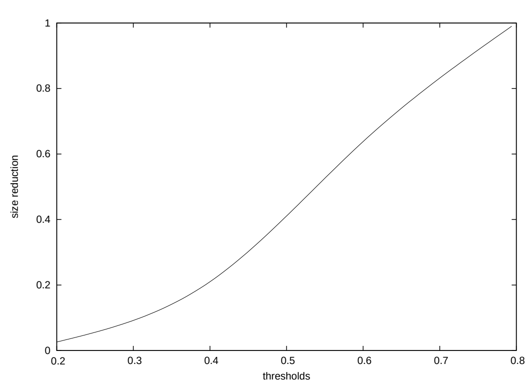

Fig. 3. Reduction $|\mathcal{B}(X, Y, I)/^a\!\approx| / |\mathcal{B}(X, Y, I)|$ from Tab. III.

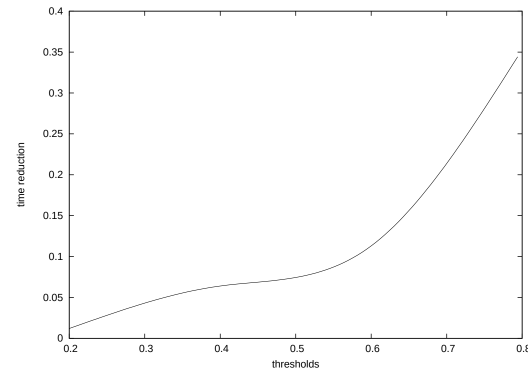

Fig. 4. Time reduction from Tab. III.

TABLE IV

DATA TABLE (FUZZY CONTEXT).

	1	2	3	4	5
	1.0	0.8	0.2	0.3	0.5
	0.8	1.0	0.2	0.6	0.9
	0.2	0.3	0.2	0.3	0.4
	0.4	0.7	0.1	0.2	0.3
	1.0	0.9	0.3	0.2	0.4

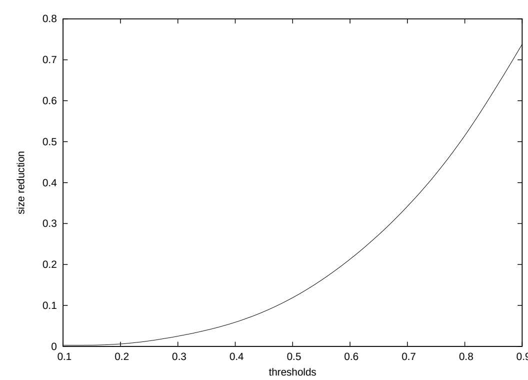

Fig. 5. Reduction $|\mathcal{B}(X, Y, I)/^a\!\approx| / |\mathcal{B}(X, Y, I)|$ from Tab. IV.

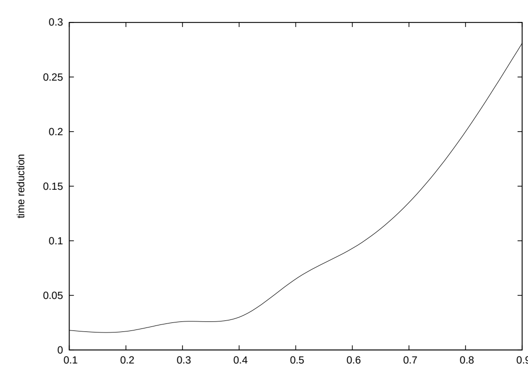

Fig. 6. Time reduction from Tab. IV.

[8] U. Dekel, Y. Gill. Visualizing class interfaces with formal concept analysis. In *OOPSLA'03*, pages 288–289, Anaheim, CA, October 2003.

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

[10] D. S. Johnson, M. Yannakakis, C. H. Papadimitrou. On generating all maximal independent sets. *Inf. Processing Letters* 15:129–133, 1988.

[11] G. .J. Klir, B. Yuan. *Fuzzy Sets and Fuzzy Logic. Theory and Applications.* Prentice Hall, Upper Saddle River, NJ, 1995.

[12] S. Pollandt. *Fuzzy Begriffe.* Springer, Berlin, 1997.

[13] G. Snelting, F. Tip. Understanding class hierarchies using concept analysis. *ACM Trans. Program. Lang. Syst.* 22(3):540–582, May 2000.

[14] G. Stumme, R. Wille, U. Wille. Conceptual knowledge discovery in databases using formal concept analysis methods. In J. M. Zytkow, M. Quafofou (Eds.). *Principles of Data Mining and Knowledge Discovery.* LNAI 1510, pages 450–458, Springer, Heidelberg, 1998.

[15] P. Valtchev, R. Missaoui, R. Godin, M. Meridji. Generating frequent itemsets incrementally: two novel approaches based on Galois lattice theory. *J. Exp. Theor. Artif. Intelligence* 14(2/3):115–142, 2002.