Noname manuscript No.
(will be inserted by the editor)

Optimal decompositions of matrices with grades into binary and graded matrices

Eduard Bartl · Radim Belohlavek · Jan Konecny

Received: date / Accepted: date

Abstract We study the problem of decomposition of object-attribute matrices whose entries contain degrees to which objects have attributes. The degrees are taken from a bounded partially ordered scale. Examples of such matrices are binary matrices, matrices with entries from a finite chain, or matrices with entries from the unit interval $[0,1]$. We study the problem of decomposition of a given object-attribute matrix I with degrees into an object-factor matrix A with degrees and a binary factor-attribute matrix B, with the number of factors as small as possible. We present a theorem which shows that decompositions which use particular formal concepts of I as factors for the decomposition are optimal in that the number of factors involved is the smallest possible. We show that the problem of computing an optimal decomposition is NP-hard and present two heuristic algorithms for its solution along with their experimental evaluation. For the first algorithm, we provide its approximation ratio. Experiments indicate that the second algorithm, which is considerably faster than the first one, delivers decompositions whose quality is comparable to the decompositions delivered by the first algorithm. We also present an illustrative example demonstrating a factor analysis interpretation of the decomposition studied in this paper.

Keywords Matrix decomposition · Factor analysis · Formal concept analysis · Fuzzy logic

1 Introduction and Problem Setting

Problem description in brief This paper presents results on optimal decompositions of matrices with degrees. Examples of such matrices are binary (or Boolean) matrices, i.e. matrices which entries are 0 or 1. Other examples are matrices which contain numbers

Supported by grant No. P202/10/0262 of the Czech Science Foundation and by research plan MSM 6198959214.

Eduard Bartl, Radim Belohlavek, Jan Konecny
Dept. Computer Science, Palacky University, Olomouc
17. listopadu 12, CZ-771 46 Olomouc, Czech Republic
E-mail: eduard.bartl@upol.cz, radim.belohlavek@upol.cz, jan.konecny@upol.cz

from the unit interval $[0, 1]$ as their entries. In general we consider non-numerical matrices with entries from particular complete lattices L (binary matrices and matrices with entries from $[0, 1]$ are particular examples with $L = \{0, 1\}$ and $L = [0, 1]$, respectively).

We consider the following problem. Let L be a partially ordered scale bounded from below and above by 0 and 1 (details specified later). Given an $n \times m$ matrix I with entries from L (i.e. $I_{ij} \in L$), we want to decompose I into a product

$$I = A \circ B$$

of an $n \times k$ matrix A with entries from L (i.e. $A_{il} \in L$) and a $k \times m$ *binary matrix* B (i.e. $B_{lj} \in \{0, 1\}$) with k *as small as possible.* The composition operation $\circ$ which we consider is defined by

$$(A \circ B)_{ij} = \bigvee_{l=1}^{k} A_{il} \otimes B_{lj}, \tag{1}$$

where $\otimes$ is defined by $a \otimes 1 = a$ and $a \otimes 0 = 0$. Note that if $L = \{0, 1\}$ then $A \circ B$ is the well-known Boolean product of binary matrices. Note also that if we allow $A_{il} \in L$ and $B_{lj} \in L$ and if $\otimes$ is a t-norm then $\circ$ is the product of graded matrices well-known in fuzzy set theory, see e.g. [13], and that such decompositions were considered in [4, 7].

Factor analysis model For a decomposition $I = A \circ B$ given by (1), I_{ij} can be interpreted as the truth degree of the following proposition: there exists a factor l such that l applies to object i *and* l is associated with attribute j (j is a particular manifestation of l). This way, a decomposition $I = A \circ B$ provides us with a factor analysis model (see [1, 12, 14] for references on factor analysis): A relationship between objects and the original attributes given by I is described using a relationship between the objects and new variables, called factors, which is given by A, and a relationship between factors and the original attributes, which is given by B. Note that we require that B be binary, i.e. that the relationship between factors and attributes be a yes-or-no relationship. This feature distinguishes our approach from those which we considered earlier [4, 7]. The requirement of binarity naturally appears in several situations, as illustrated in Section 3.

Needless to say, one can consider decompositions $I = A \circ B$ given by (1), in which A is binary and B arbitrary. Obviously, using $I^T = B^T \circ A^T$, one can reduce this type of decomposition to the first type (A arbitrary, B binary).

Contribution of the paper We present a theorem regarding the above-mentioned decompositions of a given matrix I showing that decompositions which use particular formal concepts of I as factors are optimal in that they involve the least number of factors among all decompositions of I. Furthermore, we observe that the problem of computing optimal decompositions is NP-hard and present approximation algorithms for such decompositions along with their experimental evaluation.

Related and previous work The paper is a continuation of our previous work [4, 6, 7]. Note that the problem of decomposition of binary matrices and its factor analysis interpretation go back to [17, 18], see also [15] and [16, 20, 21] for recent approaches to decomposition of binary matrices and their applications in data analysis.

Preliminaries from fuzzy logic We use standard notions of fuzzy logic and fuzzy sets, see e.g. [2,11,13]. In particular, we use complete residuated lattices as structures of truth degrees and assume familiarity with the basic calculus of complete residuated lattices. Recall that a complete residuated lattice is an algebra $\mathbf{L} = \langle L, \wedge, \vee, \otimes, \rightarrow, 0, 1\rangle$ such that $\langle L, \wedge, \vee, 0, 1\rangle$ is a complete lattice, $\langle L, \otimes, 1\rangle$ is a commutative monoid, and $\otimes$ and $\rightarrow$ satisfy the adjointness condition, i.e. $a \otimes b \leq c$ if and only if $a \leq b \rightarrow c$. We assume familiarity with examples and basic properties of residuated lattices. As an example, for $L = [0, 1]$, $a \otimes b = \max(0, a + b - 1)$, $a \rightarrow b = \min(1, 1 - a + b)$, the algebra $\mathbf{L} = \langle [0, 1], \wedge, \vee, \otimes, \rightarrow, 0, 1\rangle$ is a complete residuated lattice (so-called standard Łukasiewicz algebra). An L-set in a universe set U is a mapping $A : U \rightarrow L$. The set of all L-sets in U is denoted by L^U.

From now on, we assume that $\mathbf{L} = \langle [0, 1], \wedge, \vee, \otimes, \rightarrow, 0, 1\rangle$ is a complete residuated lattice, mainly because of a relationship to our previous work [4]. At the end of Section 2, we comment on how this assumption may be weakened. At this point, note that for every complete residuated lattice $\mathbf{L}$, $a \otimes 1 = a$ and $a \otimes 0 = 0$, i.e. $\otimes$ is a suitable operation for the decomposition (1).

2 Optimal Decompositions

2.1 Composition as $\bigvee$-superposition of matrices

Observe first that $I = A \circ B$ for $n \times k$ and $k \times m$ matrices A (graded) and B (binary) means that I is a $\bigvee$-superposition of particular rectangular-shaped matrices.

Definition 1 Let $K_1, K_2 \subseteq L$. An $n \times m$ matrix J with entries from L is called (K_1, K_2)-*rectangular* iff there exist L-sets C in $\{1, \ldots, n\}$ and D in $\{1, \ldots, m\}$ with $C(i) \in K_1$ and $D(j) \in K_2$ for all i and j such that $J = C \otimes D$, i.e.

$$J_{ij} = C(i) \otimes D(j) \tag{2}$$

for $1 \leq i \leq n$, $1 \leq j \leq m$.

In particular, we need $(L, \{0, 1\})$-rectangular matrices and call these just "rectangular". The term "rectangular" is inspired by the "shape" of such matrices. The following matrices are examples of $(\{0, 1\}, \{0, 1\})$-rectangular (J_1) and $([0, 1], \{0, 1\})$-rectangular (J_2) matrices:

$$J_1 = \begin{pmatrix} 0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0 \\ 0&0&\mathbf{1}&\mathbf{1}&\mathbf{1}&0&0 \\ 0&0&\mathbf{1}&\mathbf{1}&\mathbf{1}&0&0 \\ 0&0&\mathbf{1}&\mathbf{1}&\mathbf{1}&0&0 \\ 0&0&\mathbf{1}&\mathbf{1}&\mathbf{1}&0&0 \\ 0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0 \end{pmatrix}, \quad J_2 = \begin{pmatrix} 0.0&0.0&0.0&0.0&0.0&0.0&0.0 \\ 0.0&0.0&0.0&0.0&0.0&0.0&0.0 \\ 0.0&0.0&\mathbf{0.5}&\mathbf{0.5}&\mathbf{0.5}&0.0&0.0 \\ 0.0&0.0&\mathbf{1.0}&\mathbf{1.0}&\mathbf{1.0}&0.0&0.0 \\ 0.0&0.0&\mathbf{0.2}&\mathbf{0.2}&\mathbf{0.2}&0.0&0.0 \\ 0.0&0.0&\mathbf{1.0}&\mathbf{1.0}&\mathbf{1.0}&0.0&0.0 \\ 0.0&0.0&0.0&0.0&0.0&0.0&0.0 \\ 0.0&0.0&0.0&0.0&0.0&0.0&0.0 \end{pmatrix}.$$

In the above example, $J_1 = C \otimes D$ where C and D are characteristic functions of $\{3, 4, 5, 6\}$ and $\{3, 4, 5\}$, respectively; $J_2 = C \otimes D$ where $C(1) = C(2) = C(7) = C(8) = 0$, $C(3) = 0.5$, $C(4) = 1$, $C(5) = 0.2$, $C(6) = 1$, and $D(1) = D(2) = D(6) = D(7) = 0$, $D(3) = D(4) = D(5) = 1$.

The role of $(L, \{0, 1\})$-rectangular matrices is shown by the following lemma.

Lemma 1 *$I = A \circ B$ for $n \times k$ and $k \times m$ matrices A and B with $A_{il} \in L$ and $B_{lj} \in \{0,1\}$ iff I is a $\bigvee$-superposition of k $(L,\{0,1\})$-rectangular matrices, i.e. iff for some $(L,\{0,1\})$-rectangular matrices $J_1,\ldots,J_k$ we have*

$$I = J_1 \vee J_2 \vee \cdots \vee J_k.$$

Proof Denote by J_l the $\circ$-product $A_{\_l} \circ B_{l\_}$ of the l-th column $A_{\_l}$ of A and the l-th row $B_{l\_}$ of B, i.e. $(J_l)_{ij} = A_{il} \otimes B_{lj}$. $I = A \circ B$ means $I_{ij} = (A \circ B)_{ij}$, i.e. $I_{ij} = \bigvee_{l=1}^{k}(A_{il} \otimes B_{lj})$. Therefore, $I = J_1 \vee J_2 \vee \cdots \vee J_k$. Since B is a binary matrix, J_l are $(L,\{0,1\})$-rectangular matrices. □

Example 1 To illustrate the content of Lemma 1, consider $L = \{0, 0.1, \ldots, 0.9, 1\}$, $a \otimes b = \min(a,b)$, and the following decomposition $I = A \circ B$:

$$\begin{pmatrix} 0.3 & 1.0 & 0.0 & 0.0 & 0.0 & 0.3 \\ 1.0 & 1.0 & 0.0 & 0.0 & 1.0 & 1.0 \\ 1.0 & 0.9 & 1.0 & 1.0 & 0.0 & 0.8 \\ 1.0 & 0.2 & 0.0 & 0.0 & 1.0 & 0.2 \end{pmatrix} =$$
$$\begin{pmatrix} 0.3 & 0.0 & 0.0 & 1.0 \\ 1.0 & 0.0 & 1.0 & 0.7 \\ 0.8 & 1.0 & 0.0 & 0.9 \\ 0.2 & 0.0 & 1.0 & 0.0 \end{pmatrix} \circ \begin{pmatrix} 1 & 1 & 0 & 0 & 0 & 1 \\ 1 & 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \end{pmatrix}.$$

According to Lemma 1, I is a $\bigvee$-superposition of four matrices, J_1, J_2, J_3, J_4 where J_l is the $\circ$-product of the l-th column of A and the l-th row of B, i.e.

$$\begin{pmatrix} 0.3 & 1.0 & 0.0 & 0.0 & 0.0 & 0.3 \\ 1.0 & 1.0 & 0.0 & 0.0 & 1.0 & 1.0 \\ 1.0 & 0.9 & 1.0 & 1.0 & 0.0 & 0.8 \\ 1.0 & 0.2 & 0.0 & 0.0 & 1.0 & 0.2 \end{pmatrix} =$$
$$\begin{pmatrix} 0.3 & 0.3 & 0.0 & 0.0 & 0.0 & 0.3 \\ 1.0 & 1.0 & 0.0 & 0.0 & 0.0 & 1.0 \\ 0.8 & 0.8 & 0.0 & 0.0 & 0.0 & 0.8 \\ 0.2 & 0.2 & 0.0 & 0.0 & 0.0 & 0.2 \end{pmatrix} \vee \begin{pmatrix} 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 \\ 1.0 & 0.0 & 1.0 & 1.0 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 \end{pmatrix} \vee$$
$$\begin{pmatrix} 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 \\ 1.0 & 0.0 & 0.0 & 0.0 & 1.0 & 0.0 \\ 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 \\ 1.0 & 0.0 & 0.0 & 0.0 & 1.0 & 0.0 \end{pmatrix} \vee \begin{pmatrix} 0.0 & 1.0 & 0.0 & 0.0 & 0.0 & 0.0 \\ 0.0 & 0.7 & 0.0 & 0.0 & 0.0 & 0.0 \\ 0.0 & 0.9 & 0.0 & 0.0 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 \end{pmatrix}.$$

2.2 Formal concepts are optimal factors

Lemma 1 says that in order to find a decomposition $I = A \circ B$, we need to find a suitable set of $(L,\{0,1\})$-rectangular matrices J_l whose $\bigvee$-superposition gives I. We now describe decompositions of I which are optimal among all possible decompositions in that the number k of factors is the smallest possible. The decompositions use so-called crisply generated formal concepts of I [5].

Crisply generated formal concepts This section presents basic notions on formal concepts of data with fuzzy attributes, particularly on crisply generated formal concepts. The reader is referred, e.g., to [3,5] for details.

Let $X = \{1,\ldots,n\}$ and $Y = \{1,\ldots,m\}$ be sets (of objects and attributes, respectively), I be an $n \times m$ matrix with entries from a support set L of a complete residuated lattice $\mathbf{L}$. The degree $I_{xy} \in L$ is interpreted as a degree to which object x has attribute y. Consider the operators ${}^{\uparrow} : L^X \to L^Y$ and ${}^{\downarrow} : L^Y \to L^X$ defined by

$$C^{\uparrow}(y) = \textstyle\bigwedge_{x\in X}(C(x) \to I_{xy}), \qquad D^{\downarrow}(x) = \textstyle\bigwedge_{y\in Y}(D(y) \to I_{xy}),$$

where $\to$ is the residuum of the complete residuated lattice $\mathbf{L}$. That is, $^{\uparrow}$ assigns an L-set $C^{\uparrow}$ in Y to a given L-set C in X, and $^{\downarrow}$ assigns an L-set $D^{\downarrow}$ in X to a given L-set D in Y. $C^{\uparrow}(y)$ can verbally be described as the degree to which y is shared by every object from C (note that according to basic principles of first-order fuzzy logic, see [11], $C^{\uparrow}(y)$ is just the truth degree of "for each object $x \in X$: if x is from C then x has attribute y"). Likewise, $D^{\downarrow}(x)$ is the degree to which x shares all attributes from D ($D^{\downarrow}(x)$ is the truth degree of "for each attribute $y \in Y$: if y is from D then x has attribute y"). If $L = \{0,1\}$, $^{\uparrow} : L^X \to L^Y$ and $^{\downarrow} : L^Y \to L^X$ coincide with the well-known concept-forming operators of the basic setting of formal concept analysis [8,10]. $^{\uparrow}$ and $^{\downarrow}$ form a fuzzy Galois connection and the compound operators $^{\uparrow\downarrow}$ and $^{\downarrow\uparrow}$ form particular closure operators on X and Y [2]. A pair $\langle C, D\rangle$ consisting of an L-set C in X and an L-set D in Y is called a formal concept of I if $C^{\uparrow} = D$ and $D^{\downarrow} = C$. C and D are called the extent and intent of $\langle C, D\rangle$, respectively. The set of all formal concepts of I is denoted by $\mathcal{B}(X,Y,I)$. With a partial order $\leq$ defined by

$$\langle C_1, D_1\rangle \leq \langle C_2, D_2\rangle \text{ iff } C_1 \subseteq C_2 \text{ (iff } D_2 \subseteq D_1)$$

for $\langle C_1, D_1\rangle, \langle C_2, D_2\rangle \in \mathcal{B}(X,Y,I)$, $\mathcal{B}(X,Y,I)$ happens to be a complete lattice, so-called concept lattice of I [2,3]. Note that $C_1 \subseteq C_2$ means that C_1 is contained in C_2, i.e. for each $x \in X$, $C_1(x) \leq C_2(x)$. For $L = \{0,1\}$, $\mathcal{B}(X,Y,I)$ coincides with the ordinary concept lattice [10,23].

Note that an L-set $D \in L^Y$ is called crisp if $D(y) \in \{0,1\}$ for every $y \in Y$, i.e. if in fact $D \in \{0,1\}^Y$. In [5], the following notion was introduced.

Definition 2 A formal concept $\langle C, D\rangle \in \mathcal{B}(X,Y,I)$ is called *crisply generated* if there is a crisp L-set $E \in \{0,1\}^Y$, i.e. for each $y \in Y$: $E(y) = 0$ or $E(y) = 1$, such that $C = E^{\downarrow}$ (and thus $D = E^{\downarrow\uparrow}$).

Let $\mathcal{B}_c(X,Y,I)$ denote the collection of all crisply generated formal concepts of I, i.e.

$$\mathcal{B}_c(X,Y,I) = \{\langle C, D\rangle \in \mathcal{B}(X,Y,I) \mid \text{ there exists } E \in \{0,1\}^Y : C = E^{\downarrow}\}.$$

The structure of $\mathcal{B}_c(X,Y,I)$ as well as an algorithm for computing $\mathcal{B}_c(X,Y,I)$ are presented in [5]. We need the following characterization of crisply generated formal concepts. For L-sets $C_1, C_2 \in L^X$ and $D_1, D_2 \in L^Y$, we put $\langle C_1, D_1\rangle \trianglelefteq \langle C_2, D_2\rangle$ if for each $x \in X$, $y \in Y$ we have $C_1(x) \leq C_2(x)$ and $D_1(y) \leq D_2(y)$.

Lemma 2 ([5]) *$\langle C, D\rangle$ is a crisply generated formal concept iff $\langle C, D\rangle$ is maximal (w.r.t. $\trianglelefteq$) such that (1) the rectangular matrix J defined by $J_{xy} = C(x) \otimes D(y)$ is contained in I (i.e. $J_{xy} \leq I_{xy}$ for all x, y) and (2) $C(x) = \bigwedge_{D(y)=1} I_{xy}$.*

Remark 1 Note that condition (2) of Lemma 2 means that for the crisp L-set $D_c \in \{0,1\}^Y$ corresponding to the 1-cut of D, i.e.

$$D_c(y) = \begin{cases} 1 \text{ if } D(y) = 1, \\ 0 \text{ if } D(y) < 1, \end{cases} \tag{3}$$

we have $C = D_c^{\downarrow}$.

Matrices $A_{\mathcal{F}}$ and $B_{\mathcal{F}}$ For convenience, we identify $1 \times p$ vectors containing entries from L with L-sets in $\{1, \ldots, p\}$ (the l-th coordinate of the vector = the degree to which l belongs to the L-set). Given a set

$$\mathcal{F} = \{\langle C_1, D_1\rangle, \ldots, \langle C_k, D_k\rangle\}$$

of L-sets C_l and D_l in $\{1, \ldots, n\}$ and $\{1, \ldots, m\}$, respectively, with values from L, define $n \times k$ and $k \times m$ matrices $A_{\mathcal{F}}$ and $B_{\mathcal{F}}$ by

$$(A_{\mathcal{F}})_{il} = (C_l)(i) \quad \text{and} \quad (B_{\mathcal{F}})_{lj} = (D_l)(j).$$

That is, the l-th column of $A_{\mathcal{F}}$ is the transpose of the vector corresponding to C_l and the l-th row of $B_{\mathcal{F}}$ is the vector corresponding to D_l.

For $\mathcal{F} \subseteq \mathcal{B}(X, Y, I)$, denote

$$\mathcal{F}_c = \{\langle C, D_c\rangle \mid \langle C, D\rangle \in \mathcal{F}\}.$$

Note that D_c is defined by (3). We are going to show that sets $\mathcal{F}_c$ corresponding to sets $\mathcal{F}$ of crisply generated formal concepts are fundamental for decompositions (1).

Theorem 1 (universality) *For every I with entries from L there is $\mathcal{F} \subseteq \mathcal{B}_c(X, Y, I)$ such that $I = A_{\mathcal{F}_c} \circ B_{\mathcal{F}_c}$, i.e. I is a product of $A_{\mathcal{F}_c}$ with entries from L and $B_{\mathcal{F}_c}$ with entries from $\{0, 1\}$.*

Proof Denote for $l \in \{1, \ldots, m\}$, $\langle C_l, D_l\rangle = \langle \{1/l\}^{\downarrow}, \{1/l\}^{\downarrow\uparrow}\rangle$. Here, $\{1/l\}$ is a singleton in $\{1, \ldots, m\}$, i.e. an L-set defined by $\{1/l\}(l) = 1$ and $\{1/l\}(j) = 0$ for $j \neq l$. $\langle C_l, D_l\rangle$ are particular crisply generated formal concepts from $\mathcal{B}(X, Y, I)$. Furthermore, we have

$$I_{ij} = \textstyle\bigvee_{l=1}^{m} C_l(i) \otimes D_l(j),$$

see [2]. Putting $\mathcal{F} = \{\langle C_l, D_l\rangle \mid l = 1, \ldots, m\}$, we get $I = A_{\mathcal{F}_c} \circ B_{\mathcal{F}_c}$. Indeed,

$$\begin{aligned} I_{ij} = I_{ij} \otimes 1 = \{1/j\}^{\downarrow}(i) \otimes (\{1/j\}^{\downarrow\uparrow})_c(j) = C_j(i) \otimes (D_j)_c(j) \leq \\ \leq C_j(i) \otimes D_j(j) \leq \textstyle\bigvee_{l=1}^{m} C_l(i) \otimes D_l(j) = I_{ij}, \end{aligned}$$

completing the proof. □

However, Theorem 1 and its proof yield only $|\mathcal{F}| = m$, i.e. the number $k = |\mathcal{F}|$ of factors equals the number m of attributes. In general, better decompositions may exist, i.e. those with $k < m$. The next theorem shows that the decompositions which use crisply generated formal concepts of I as factors are optimal among all decompositions of I.

Theorem 2 (optimality) *Let $I = A \circ B$ for $n \times k$ and $k \times m$ matrices A and B with $A_{il} \in L$, $B_{lj} \in \{0, 1\}$. Then there exists a set $\mathcal{F} \subseteq \mathcal{B}_c(X, Y, I)$ of crisply generated formal concepts of I such that for $\mathcal{F}_c$ we have*

$$|\mathcal{F}_c| \leq k$$

and for the $n \times |\mathcal{F}_c|$ and $|\mathcal{F}_c| \times m$ matrices $A_{\mathcal{F}_c}$ with entries from L and $B_{\mathcal{F}_c}$ with entries from $\{0, 1\}$ we have

$$I = A_{\mathcal{F}_c} \circ B_{\mathcal{F}_c}.$$

Proof Let $I = A \circ B$ for an $n \times k$ matrix A with entries from L and a $k \times m$ matrix B with entries from $\{0,1\}$. Consider the corresponding rectangular matrices $J_1, \ldots, J_k$ of which I is a $\bigvee$-superposition according to Lemma 1. Denoting now the L-sets in $\{1,\ldots,n\}$ and $\{1,\ldots,m\}$ corresponding to the l-th column of A and the l-th row of B by G_l and H_l, respectively, we have $J_l = G_l \otimes H_l$. We have $G_l \otimes H_l \subseteq I$ and one can check using adjointness that also $H_l^{\downarrow} \otimes H_l \subseteq I$, i.e. $H_l^{\downarrow} \otimes H_l$ is contained in I. As $H_l = (H_l)_c$, the pair $\langle H_l^{\downarrow}, H_l\rangle$ satisfies condition (2) of Lemma 2 (see also Remark 1). Therefore, $\langle H_l^{\downarrow}, H_l\rangle$ is contained in some maximal (w.r.t. $\trianglelefteq$ defined in the paragraph preceding Lemma 2) $\langle C_l, D_l\rangle$ which is contained in I and satisfies (2) of Lemma 2. According to Lemma 2, such $\langle C_l, D_l\rangle$ is a crisply generated formal concept of I. In particular, we have $C_l \otimes D_l \subseteq I$. Therefore, for $\mathcal{F} = \{\langle C_1, D_1\rangle, \ldots, \langle C_k, D_k\rangle\}$ we have $|\mathcal{F}_c| \leq k$. Because $(H_l)_j \in \{0,1\}$ and because $H_l \subseteq D_l$, we get $H_l \subseteq (D_l)_c$, cf. (3). We thus get,

$$I = A \circ B = \bigvee_{l=1}^{k} G_l \otimes H_l \subseteq \bigvee_{l=1}^{k} H_l^{\downarrow} \otimes H_l \subseteq$$

$$\subseteq \bigvee_{l=1}^{k} C_l \otimes (D_l)_c = A_{\mathcal{F}_c} \circ B_{\mathcal{F}_c} \subseteq \bigvee_{l=1}^{k} C_l \otimes D_l = A_{\mathcal{F}} \circ B_{\mathcal{F}} \subseteq I,$$

i.e. $A_{\mathcal{F}_c} \circ B_{\mathcal{F}_c} = I$, finishing the proof. □

Remark 2 (a) Note that using the notation from the proof of Theorem 2, two distinct $\langle G_l, H_l\rangle$'s may be contained in a single $\langle C_l, D_l\rangle$, i.e. for $\langle G_{l_1}, H_{l_1}\rangle \neq \langle G_{l_2}, H_{l_2}\rangle$ we may have $\langle C_{l_1}, D_{l_1}\rangle = \langle C_{l_2}, D_{l_2}\rangle$. As a consequence, we may have $|\mathcal{F}_c| < k$.

(b) Note that it follows from the proof of Theorem 2 that if $\mathcal{F} \subseteq \mathcal{B}_c(X,Y,I)$ is a set of crisply generated formal concepts of I, then $I = A_{\mathcal{F}} \circ B_{\mathcal{F}}$ if and only if $I = A_{\mathcal{F}_c} \circ B_{\mathcal{F}_c}$.

In the rest of this section, we show that optimal decompositions $I = A \circ B$ in which B is a binary matrix and which use crisply generated formal concepts as factors do not depend on the choice of logical connectives on the scale L of truth degrees. This is indicated by the fact that if A is a matrix with degrees from L and B a binary matrix, then $A \circ B$ defined by (1) does not depend on the choice of multiplication $\otimes$ because $a \otimes 0 = 0$ and $a \otimes 1 = a$ for every multiplication of a residuated lattice. As a result of this observation, we could have presented the results in this paper in a different way, not referring to the notion of residuum. However, our way of presenting makes the connection to our previous results [4,7] on decomposition of graded matrices transparent.

Let $\mathbf{L}_1 = \langle L, \wedge, \vee, \otimes_1, \rightarrow_1\rangle$ and $\mathbf{L}_2 = \langle L, \wedge, \vee, \otimes_2, \rightarrow_2\rangle$ be two complete residuated lattices, i.e. $\mathbf{L}_1$ and $\mathbf{L}_2$ have a common support L. Denote by $\mathcal{B}_c^{\mathbf{L}_i}(X,Y,I)$ the set of all crisply generated formal concepts using $\mathbf{L}_i$ as the structure of truth degrees. Since $A \circ B$ does not depend on the multiplication, it follows immediately from Theorem 2 that if $\mathcal{F}_1$ is the smallest subset of $\mathcal{B}_c^{\mathbf{L}_1}(X,Y,I)$ for which $I = A_{\mathcal{F}_{1c}} \circ B_{\mathcal{F}_{1c}}$, then $k = |\mathcal{F}_1|$ is also the size of the smallest subset $\mathcal{F}_2$ of $\mathcal{B}_c^{\mathbf{L}_2}(X,Y,I)$ for which $I = A_{\mathcal{F}_{2c}} \circ B_{\mathcal{F}_{2c}}$. In fact, as the next theorem shows, optimal decompositions using crisply generated formal concepts do not depend on the choice of the logical connectives on the set L of truth degrees.

Theorem 3 *Let $I = A_{\mathcal{F}_{1c}} \circ B_{\mathcal{F}_{1c}}$ for $\mathcal{F}_1 \subseteq \mathcal{B}_c^{\mathbf{L}_1}(X,Y,I)$. Then there exists $\mathcal{F}_2 \subseteq \mathcal{B}_c^{\mathbf{L}_2}(X,Y,I)$ such that $A_{\mathcal{F}_{1c}} = A_{\mathcal{F}_{2c}}$ and $B_{\mathcal{F}_{1c}} = B_{\mathcal{F}_{2c}}$.*

Proof For every $l = 1, \ldots, |\mathcal{F}_1|$, the l-th column of $A_{\mathcal{F}_{1c}}$ and the l-th row of $B_{\mathcal{F}_{1c}}$ coincide with an extent C_l^1 and the 1-cut $(D_l^1)_c$ of an intent D_l^1 of a crisply generated formal concept $\langle C_l^1, D_l^1 \rangle \in \mathcal{B}_c^{\mathbf{L}_1}(X, Y, I)$, respectively. It follows from [5, Lemma 3] that there exists $\langle C_l^2, D_l^2 \rangle \in \mathcal{B}_c^{\mathbf{L}_2}(X, Y, I)$ such that $C_l^1 = C_l^2$ and $(D_l^1)_c = (D_l^2)_c$. Therefore, for $\mathcal{F}_2 = \{\langle C_l^2, D_l^2 \rangle \mid \langle C_l^1, D_l^1 \rangle \in \mathcal{F}_1\}$ we have $A_{\mathcal{F}_{1c}} = A_{\mathcal{F}_{2c}}$ and $B_{\mathcal{F}_{1c}} = B_{\mathcal{F}_{2c}}$. □

3 Illustrative Example

In this section, we present an illustrative example regarding decompositions of a matrix with grades into a matrix with grades and a binary matrix.

In our example, we consider n users, m permissions, and a user-to-permission assignment. The assignment can be represented by an $n \times m$ matrix I with entries from a scale $L = \{0, r, a, w, 1\}$, with 0 representing "no permission", r, a and w representing "permission to read", "permission to append" and "permission to write", respectively, and 1 representing "full permission" (that means, anyone who has this permission is also allowed to change metadata associated with files). We define a partial order on L such that 0 is the least element, 1 is the greatest one, r and a are incomparable, and w covers r and a, see Fig. 1.

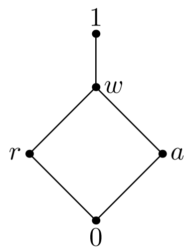


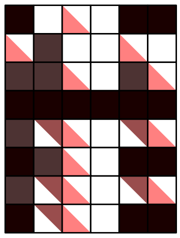
Fig. 1 Partial order on the scale of permissions

Furthermore, we need to define the operation $\otimes$ of multiplication on L. We put $x \otimes y = x \wedge y$, for all $x, y \in L$. The residuum is then determined by $\otimes$ (due to adjointness, see Section 1) and is defined by $x \rightarrow y = \bigvee\{z \mid x \otimes z \leq y\}$.

We want to decompose I into a product of an $n \times k$ matrix A and a $k \times m$ matrix B where A and B represent a user-to-role and a role-to-permission relationship, respectively. Therefore, the factors we want to discover are to be interpreted as roles, such as "system administrator", "standard user", or the like. Naturally, we expect A to be a binary matrix (i.e. $A_{il} \in \{0, 1\}$), assigning roles to users (a user has a given role or not), whereas B contains degrees (i.e. $B_{lj} \in L$). In order to be consistent with the previous sections, A should contain degrees and B should be binary. Therefore, we use well-known fact that $I = A \circ B$ is equivalent to $I^{-1} = B^{-1} \circ A^{-1}$, and decompose I^{-1} instead of I.

As a particular example, we consider 8 users and 6 file-types in a computer system (for instance, "text documents", "database files", "executable files", "system files", "html files", "archive files"). The user-to-permission relationship is described in the table thereunder. The data can be visualized using a rectangular grid, where , , , , and represent permissions 0, r, a, w, and 1, respectively:

	type_1	type_2	type_3	type_4	type_5	type_6
Alice	1	0	r	0	1	1
Bob	r	w	0	0	r	0
Charles	w	w	r	0	w	r
David	1	1	1	1	1	1
Eve	w	a	r	0	a	r
Frank	1	w	r	0	1	1
George	w	a	r	0	a	r
Henry	1	a	r	0	1	1



Our aim is to decompose the corresponding matrix

$$I^{-1} = \begin{pmatrix} 1 & r & w & 1 & w & 1 & w & 1 \\ 0 & w & w & 1 & a & w & a & a \\ r & 0 & r & 1 & r & r & r & r \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & r & w & 1 & a & 1 & a & 1 \\ 1 & 0 & r & 1 & r & 1 & r & 1 \end{pmatrix}.$$

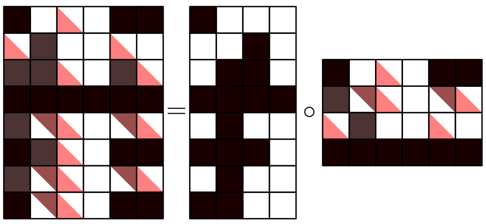
Consider the following decomposition:

$$I^{-1} = \begin{pmatrix} 1 & w & r & 1 \\ 0 & a & w & 1 \\ r & r & 0 & 1 \\ 0 & 0 & 0 & 1 \\ 1 & a & r & 1 \\ 1 & r & 0 & 1 \end{pmatrix} \circ \begin{pmatrix} 1 & 0 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \end{pmatrix},$$

i.e.,

$$I = A \circ B = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 \end{pmatrix} \circ \begin{pmatrix} 1 & 0 & r & 0 & 1 & 1 \\ w & a & r & 0 & a & r \\ r & w & 0 & 0 & r & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 \end{pmatrix}.$$

This decomposition can be displayed as:



We obtained a 8×4 binary matrix A describing a user-to-role assignment and a 4×6 matrix B describing a role-to-permission assignment, i.e. we obtained 4 factors: role_1, role_2, role_3, and role_4. As can be seen from matrix B, role_1 (corresponding to the first row of matrix B) may be interpreted as "webmaster", role_2 (the second row of B) as "standard user", role_3 (the third row of B) as a "user possessing permission to database files" (for instance, accountant), and role_4 (the last row of B) as "system administrator".

According to matrix A, we assign roles to users by:

Alice	- role_1
Bob	- role_3
Charles	- role_2, role_3
David	- all roles,
Eve	- role_2
Frank	- role_1, role_2, role_3
George	- role_2
Henry	- role_1, role_2

4 Hardness of Decompositions and Approximation Algorithms

In this section, we show that the problem of computing optimal decompositions (1) is provably hard. Then, we propose two approximation algorithms and present results of an experimental evaluation of the algorithms.

4.1 Hardness of Decompositions

The decomposition problem studied in this paper is an optimization problem: Given an $n \times m$ matrix I with entries from L, find an $n \times k$ matrix A with entries from L and a $k \times m$ matrix B with entries from $\{0, 1\}$ such that $I = A \circ B$ and such that the number k of factors is the smallest possible. In accordance to [22, p. 347], we call such an optimization problem NP-hard if the corresponding decision problem is NP-hard. Note that the decision problem corresponding to our problem is: Given I with entries from L and a positive integer k, can I be decomposed into $I = A \circ B$ for an $n \times k$ matrix A with entries from L and a $k \times m$ matrix B with entries from $\{0, 1\}$? We assume that elements from L can be reasonably encoded by strings of length polynomial in n and m. The following theorem shows that computing optimal decompositions is hard.

Theorem 4 *The above problem of deciding whether for a given I and k there exists a decomposition $I = A \circ B$ with k factors is NP-complete. Hence, the optimization problem of computing a decomposition $I = A \circ B$ with the number k of factors as small as possible is NP-hard.*

Proof It suffices to prove that the decision problem, call it DECOMP(L), is NP-complete. Clearly, the problem is in NP. The NP-hardness of the decision problem follows from the NP-hardness of the set basis problem [19]. It is well known [17, 18], see also [6], that the set basis problem can equivalently be formulated as the following problem of decomposition of binary matrices DECOMP($\{0, 1\}$): Given an $n \times m$ binary matrix J and a positive integer k, do there exist an $n \times k$ binary matrix M and a $k \times m$ binary matrix B such that J equals the Boolean matrix product of M and N? Now, DECOMP($\{0, 1\}$) is reducible in polynomial time to DECOMP(L). Namely, since every residuum $\rightarrow$ on L coincides with the truth function of classical implication on values from $\{0, 1\}$, a direct calculation shows that the crisply generated formal concepts from $\mathcal{B}_c^{\mathbf{L}}(X, Y, J)$ coincide with the ordinary formal concepts of J. Therefore, [6] and Theorem 2 imply that the smallest number of factors needed for a decomposition of J into binary matrices A and B equals the smallest number of factors needed for a decomposition of J into A with degrees from L and a binary matrix B. $\square$

4.2 Approximation Algorithms

As a consequence of Theorem 4, in order to compute optimal decompositions of I, we need to resort to approximation algorithms. In the following, we propose two greedy approximation algorithms inspired by [6].

The algorithms start with an empty set $\mathcal{F}$ of crisply generated formal concepts of $\mathcal{B}_c(X,Y,I)$. In every step, the algorithms check whether the matrix $I_{\mathcal{F}_c}$ which results as a product of $A_{\mathcal{F}_c}$ and $B_{\mathcal{F}_c}$, i.e.

$$I_{\mathcal{F}_c} = A_{\mathcal{F}_c} \circ B_{\mathcal{F}_c},$$

equals I. While $I \neq I_{\mathcal{F}_c}$, the algorithms use a greedy strategy and generate a crisply generated concept $\langle C,D\rangle \in \mathcal{B}_c(X,Y,I)$ such that the number of matrix entries for which I and $I_{(\mathcal{F}\cup\{\langle C,D\rangle\})_c}$ coincide is large.

For the greedy selection of $\langle C,D\rangle \in \mathcal{B}_c(X,Y,I)$, Algorithm 1 first computes the set $\mathcal{B}_c(X,Y,I)$ of all crisply generated formal concepts using the method described in [5]. The greedy strategy used in Algorithm 1 selects $\langle C,D\rangle \in \mathcal{B}_c(X,Y,I)$ which maximizes the number of entries for which I and $I_{(\mathcal{F}\cup\{\langle C,D\rangle\})_c}$ coincide. This strategy is inspired by the well-known approximation algorithm for a set covering problem described e.g. [9,22]. In fact, the set $\mathcal{U}$ to be covered is $\mathcal{U} = \{\langle i,j\rangle \mid I_{ij} > 0\}$ and the collection $\mathcal{S}$ of subsets of $\mathcal{U}$ is $\mathcal{S} = \{\mathcal{S}_{\langle C,D\rangle} \subseteq \mathcal{U} \mid \langle C,D\rangle \in \mathcal{B}_c(X,Y,I)\}$ with $\mathcal{S}_{\langle C,D\rangle} = \{\langle i,j\rangle \mid C(i) \otimes D_c(j) > 0\}$. From this point of view, Algorithm 1 follows the greedy strategy of the well-known approximation algorithm to select the smallest subset $\mathcal{C}$ of $\mathcal{S}$ for which $\bigcup\mathcal{C} = \mathcal{U}$. As a result, Algorithm 1 achieves the approximation ratio of $\ln(|\{\langle i,j\rangle \mid I_{ij} > 0\}|) + 1$, see [9,22].

Algorithm 1 Find Factors

Input: I (matrix with entries from L)
Output: $\mathcal{F}$ ($\mathcal{F} \subseteq \mathcal{B}_c(X,Y,I)$ for which $I = A_{\mathcal{F}_c} \circ B_{\mathcal{F}_c}$)
set $\mathcal{F} = \emptyset$
compute $\mathcal{S} = \mathcal{B}_c(X,Y,I)$ using NextClosure from [5]
while $I \neq I_{\mathcal{F}_c}$ **do**
  select $\langle C,D\rangle \in \mathcal{S}$ which maximizes the number of entries for which I and $I_{(\mathcal{F}\cup\{\langle C,D\rangle\})_c}$ coincide
  remove $\langle C,D\rangle$ from $\mathcal{S}$
  add $\langle C,D\rangle$ to $\mathcal{F}$
end while

While Algorithm 1 is provided with its approximation ratio, its drawback consists in that it needs to compute $\mathcal{B}_c(X,Y,I)$. As $\mathcal{B}_c(X,Y,I)$ may be large, computing $\mathcal{B}_c(X,Y,I)$ slows Algorithm 1 down. Algorithm 2 overcomes this drawback by computing the next $\langle C,D\rangle$ to be added to $\mathcal{F}$ "locally" as follows. $\langle C,D\rangle$ is constructed iteratively. In the beginning, $\langle C,D\rangle = \langle \emptyset^{\downarrow}, \emptyset^{\downarrow\uparrow}\rangle$. In every step of the iteration, we add to D the best column j for which $\langle (D \cup \{\,{}^{1}\!/j\})^{\downarrow}, (D \cup \{\,{}^{1}\!/j\})^{\downarrow\uparrow}\rangle$ is better than $\langle C,D\rangle$, i.e. for which I coincides for more entries with $I_{(\mathcal{F}\cup\{\langle (D\cup\{\,{}^{1}\!/j\})^{\downarrow}, (D\cup\{\,{}^{1}\!/j\})^{\downarrow\uparrow}\rangle\})_c}$ than with $I_{(\mathcal{F}\cup\{\langle C,D\rangle\})_c}$. If no such column j exists, the iteration ends with $\langle C,D\rangle$ to be added to $\mathcal{F}$. In the pseudocode of Algorithm 2, we use $|D \oplus j|$ to denote the number of entries for which I and $I_{(\mathcal{F}\cup\{\langle (D\cup\{\,{}^{1}\!/j\})^{\downarrow}, (D\cup\{\,{}^{1}\!/j\})^{\downarrow\uparrow}\rangle\})_c}$ coincide, i.e. the number of

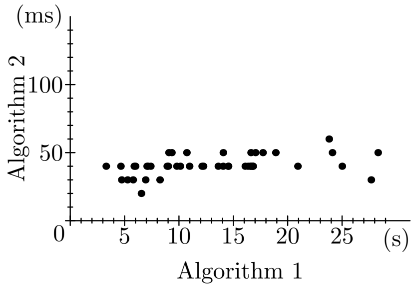


Fig. 2 Computation time needed for matrix decomposition by Algorithm 2 against Algorithm 1

pairs $\langle i, j' \rangle$ for which $I_{ij'} = \Big(I_{\mathcal{F}_c} \vee (D \cup \{\,{}^{1}/j\})^{\downarrow} \otimes (D \cup \{\,{}^{1}/j\})_c^{\downarrow\uparrow}\Big)_{ij'}$. Note that Algorithms 1 and 2 contain just the pseudocodes and that we omit the details regarding their efficient implementation.

Algorithm 2 Find Factors

```
Input: I (matrix with entries from L)
Output: F (F ⊆ B_c(X, Y, I) for which I = A_{F_c} ∘ B_{F_c})
set F = ∅
while I ≠ I_{F_c} do
   set D to ∅
   set V to 0
   while there is j such that D(j) < 1 and |D ⊕ j| > V do
      select j with D(j) < 1 which maximizes |D ⊕ j|
      set D to (D ∪ {1/j})^{↓↑}
      set V to |D ⊕ j|
   end while
   set C to D^↓
   add ⟨C, D⟩ to F
end while
```

An interesting feature is that not only is Algorithm 2 much faster than Algorithm 1, but also the decompositions delivered by Algorithm 2 and by Algorithm 1 are comparable in terms of the number of factors. This is shown in detail below.

The graph in Fig. 2 demonstrates the speedup achieved by Algorithm 2 compared to Algorithm 1. The graph compares the time needed for computing decompositions of randomly generated 25×25 matrices over a five-element chain L by Algorithm 1 vs. Algorithm 2. A higher dispersion in the case of Algorithm 1 is due to the possibly rather varying sizes of crisply generated concept lattices associated with various input matrices, which need to be computed by Algorithm 1.

Table 1 illustrates the quality of decompositions delivered by Algorithm 1 compared to Algorithm 2 in terms of the number of factors computed. The numbers were obtained by decomposing 1000 randomly generated 30×30 matrices. Every such matrix was generated as a product of a $30 \times k$ matrix with degrees and a $k \times 30$ binary matrix. Hence, the number k of factors (the upper bound of the smallest number of factors, in fact) was known. As the table illustrates, both Algorithm 1 and Algorithm 2 tend to find the same number of factors.

Table 1 Numbers of factors computed by Algorithm 1 and Algorithm 2 (number of factors ± standard deviation)

k	Algorithm 1 no. computed factors	Algorithm 2 no. computed factors
2	2.01±0.01	2.00±0.00
3	3.03±0.03	3.03±0.03
4	4.07±0.06	4.06±0.06
5	5.16±0.15	5.13±0.13
6	6.29±0.28	6.26±0.29
7	7.47±0.49	7.44±0.44
8	8.76±0.68	8.75±0.80
9	10.14±1.07	10.29±1.38
10	11.74±1.62	12.00±2.34
11	13.56±2.38	13.91±3.12
12	15.31±3.16	16.14±3.89
13	17.21±3.22	18.34±4.87
14	19.24±3.58	20.49±4.95
15	21.08±3.64	22.59±5.78

Table 2(a) demonstrates how Algorithm 2 performs in computing approximate decompositions. For a prescribed percentage $p\%$ (such as 88% demonstrated by the second column of the tables), we stopped Algorithm 2 after the set $\mathcal{F}$ of computed factors "explained" at least $p\%$ of the input data, i.e. after at least $p\%$ of the entries of I and $I_{\mathcal{F}_c} = A_{\mathcal{F}_c} \circ B_{\mathcal{F}_c}$ were equal. The number of factors in $\mathcal{F}$ is then considered as the number of factors explaining $p\%$ of the data. This is demonstrated in Table 2(a). Again, we used 1000 randomly generated matrices with known upper bounds of the numbers k of factors.

Table 2(b) demonstrates the same type of experiment with a different way of assessment of approximate equality of matrices I and $I_{\mathcal{F}_c}$ in which we considered higher truth degrees more important. Particularly, we used weights of truth degrees in the assessment of approximate equality of matrices and used higher weights for higher truth degrees.

References

1. Bartholomew, D. J., Knott M.: Latent Variable Models and Factor Analysis, 2nd Ed., London, Arnold (1999)
2. Belohlavek R.: Fuzzy Relational Systems: Foundations and Principles. Kluwer, Academic/Plenum Publishers, New York (2002)
3. Belohlavek R.: Concept lattices and order in fuzzy logic. Annals of Pure and Applied Logic 128(1–3), 277–298 (2004)
4. Belohlavek R.: Optimal decompositions of matrices with grades. Proc. IEEE Intelligent Systems 2008, Varna, Bulgaria, pp. 15-2–15-7, ISBN 978-1-4244-1740-7 (2008) (extended version submitted to J. Logic and Computation).
5. Belohlavek R., Sklenar V., Zacpal J.: Crisply generated fuzzy concepts. Lecture Notes in Artificial Intelligence 3403, 268–283 (2005)
6. Belohlavek R., Vychodil V.: Discovery of optimal factors in Boolean factor analysis via novel method of binary matrix decomposition. J. Computer and System Sciences 76(1), 3–20 (2010)
7. Belohlavek R., Vychodil V.: Factor analysis of incidence data via novel decomposition of matrices. Lecture Notes in Artificial Intelligence 5548, 83–97 (2009)
8. Carpineto C., Romano G.: Concept Data Analysis. Theory and Applications. J. Wiley (2004)

k	no. computed factors 88%	no. computed factors 90%	no. computed factors 92%	no. computed factors 94%	no. computed factors 96%	no. computed factors 98%	no. computed factors 100%
2	2.00±0.00	2.00±0.01	2.00±0.00	2.00±0.01	2.01±0.01	2.00±0.01	2.01±0.01
3	2.98±0.03	3.00±0.03	3.01±0.02	3.02±0.03	3.02±0.02	3.02±0.02	3.02±0.02
4	3.79±0.17	3.91±0.09	3.99±0.06	4.02±0.04	4.05±0.05	4.05±0.05	4.05±0.05
5	4.43±0.25	4.68±0.22	4.89±0.13	5.01±0.09	5.06±0.07	5.09±0.09	5.10±0.10
6	5.18±0.17	5.40±0.25	5.63±0.26	5.90±0.19	6.09±0.14	6.16±0.16	6.23±0.22
7	6.04±0.20	6.20±0.25	6.47±0.30	6.80±0.32	7.09±0.21	7.26±0.27	7.44±0.44
8	6.82±0.31	7.03±0.28	7.33±0.33	7.65±0.39	8.07±0.36	8.47±0.53	8.78±0.77
9	7.53±0.39	7.87±0.41	8.21±0.39	8.60±0.49	9.06±0.56	9.63±0.64	10.29±1.37
10	8.26±0.48	8.68±0.48	9.09±0.53	9.64±0.67	10.13±0.70	10.89±0.98	11.90±2.02
11	9.01±0.55	9.45±0.62	9.97±0.71	10.59±0.86	11.38±1.06	12.25±1.28	13.90±3.19
12	9.77±0.75	10.24±0.80	10.86±0.87	11.57±1.07	12.48±1.34	13.60±1.71	16.14±4.11
13	10.50±0.87	11.03±0.90	11.72±1.12	12.65±1.30	13.68±1.52	14.98±2.17	18.25±4.35
14	11.10±1.00	11.82±1.11	12.60±1.20	13.52±1.40	14.77±1.82	16.49±2.37	20.49±5.38
15	11.64±1.04	12.42±1.15	13.24±1.43	14.42±1.59	15.82±2.05	17.82±2.87	22.54±5.03
16	12.13±1.25	12.93±1.45	13.93±1.55	15.12±1.85	16.79±2.28	19.04±2.84	24.24±5.27
17	12.55±1.34	13.37±1.46	14.43±1.66	15.76±1.93	17.43±2.44	20.00±2.86	25.63±4.93
18	12.75±1.39	13.78±1.59	14.79±1.99	16.23±2.09	18.02±2.50	20.61±3.18	26.76±5.06
19	12.96±1.46	13.93±1.79	15.12±1.89	16.58±2.32	18.39±2.73	21.09±3.45	27.50±5.48

(a) using original method of comparison matrices

k	no. computed factors 88%	no. computed factors 90%	no. computed factors 92%	no. computed factors 94%	no. computed factors 96%	no. computed factors 98%	no. computed factors 100%
2	2.00±0.01	2.00±0.01	2.00±0.01	2.01±0.01	2.01±0.01	2.01±0.01	2.01±0.01
3	2.98±0.03	3.01±0.03	3.02±0.03	3.02±0.03	3.03±0.03	3.03±0.03	3.02±0.02
4	3.80±0.17	3.92±0.09	4.00±0.07	4.04±0.05	4.07±0.07	4.06±0.07	4.06±0.06
5	4.45±0.25	4.70±0.23	4.90±0.14	5.02±0.09	5.09±0.10	5.12±0.11	5.14±0.14
6	5.21±0.19	5.41±0.26	5.65±0.28	5.92±0.20	6.12±0.16	6.22±0.23	6.26±0.26
7	6.05±0.22	6.24±0.28	6.51±0.33	6.83±0.35	7.13±0.25	7.35±0.34	7.52±0.50
8	6.85±0.34	7.08±0.33	7.38±0.38	7.74±0.43	8.11±0.39	8.55±0.56	8.80±0.83
9	7.60±0.44	7.90±0.41	8.28±0.42	8.66±0.55	9.14±0.62	9.65±0.61	10.34±1.30
10	8.29±0.49	8.70±0.48	9.14±0.56	9.66±0.68	10.18±0.86	10.98±1.01	11.93±2.06
11	9.04±0.60	9.49±0.60	10.00±0.70	10.62±0.84	11.42±0.98	12.25±1.20	13.82±3.01
12	9.75±0.72	10.24±0.77	10.87±0.87	11.59±1.09	12.47±1.24	13.60±1.58	15.89±3.59
13	10.40±0.79	10.99±0.85	11.66±1.03	12.54±1.13	13.55±1.44	14.98±2.06	17.94±4.28
14	10.97±0.94	11.72±1.05	12.48±1.13	13.42±1.30	14.64±1.71	16.21±2.10	20.03±4.84
15	11.49±0.94	12.29±1.07	13.10±1.37	14.23±1.55	15.65±1.95	17.57±2.40	22.04±4.74
16	11.97±1.11	12.71±1.23	13.72±1.36	14.88±1.71	16.51±1.92	18.64±2.36	23.54±4.46
17	12.29±1.16	13.12±1.37	14.22±1.36	15.46±1.82	17.12±2.14	19.55±2.54	24.92±4.72
18	12.50±1.30	13.43±1.33	14.46±1.66	15.93±1.80	17.68±2.23	20.09±2.63	26.10±4.57
19	12.69±1.23	13.64±1.58	14.75±1.74	16.25±2.19	18.06±2.30	20.65±3.11	26.77±4.67

(b) using alternative method of comparison matrices

Table 2 Number of computed factors

9. Cormen T. H., Leiserson C. E., Rivest R. L., Stein C.: Introduction to Algorithms, 2nd Ed. MIT Press (2001)
10. Ganter B., Wille R.: Formal Concept Analysis. Mathematical Foundations. Springer, Berlin (1999)
11. Hájek P.: Metamathematics of Fuzzy Logic. Kluwer, Dordrecht (1998)
12. Harman H. H.: Modern Factor Analysis, 2nd Ed. The Univ. Chicago Press, Chicago (1970)
13. Klir G. J., Yuan B.: Fuzzy Sets and Fuzzy Logic. Theory and Applications. Prentice-Hall (1995)
14. McDonald R. P.: Factor Analysis and Related Methods. Lawrence Erlbaum Associates, Inc. (1985)
15. Mickey M. R., Mundle P., Engelman L.: Boolean factor analysis. In: W.J. Dixon (Ed.), BMDP Statistical Software, University of California Press, Berkeley, CA, 538–545 (1983)
16. Miettinen P., Mielikäinen T., Gionis A., Das G., Mannila H.: The discrete basis problem. Lecture Notes in Artificial Intelligence 4213, pp. 335–346 (2006)
17. Nau D. S.: Specificity covering: immunological and other applications, computational complexity and other mathematical properties, and a computer program. A. M. Thesis, Technical Report CS–1976–7, Computer Sci.Dept., Duke Univ., Durham, N. C. (1976)
18. Nau D. S., Markowsky G., Woodbury M. A., Amos D. B.: A mathematical analysis of human leukocyte antigen serology. Math. Biosciences 40, 243–270 (1978)
19. Stockmeyer L. J.: The set basis problem is NP-complete. IBM Research Report RC5431, Yorktown Heights, NY (1975)
20. Tatti N., Mielikäinen T., Gionis A., Mannila H.: What is the dimension of your binary data? In: The 2006 IEEE Conference on Data Mining (ICDM 2006), IEEE Computer Society, 603–612 (2006)
21. Vaidya J., Atluri V., Guo Q.: The Role Mining Problem: Finding a Minimal Descriptive Set of Roles. ACM Symposium on Access Control Models and Technologies, June, 175–184 (2007)
22. Vazirani V. V.: Approximation Algorithms. Springer, Berlin (2003)
23. Wille R.: Restructuring lattice theory: an approach based on hierarchies of concepts. In: I. Rival (Ed.): Ordered Sets, 445–470, Reidel, Dordrecht-Boston (1982)