Available online at www.sciencedirect.com

ScienceDirect

Fuzzy Sets and Systems 292 (2016) 85–97

FUZZY sets and systems

www.elsevier.com/locate/fss

Factorization of matrices with grades

Radim Belohlavek *, Vilem Vychodil

Dept. Computer Science, Palacký University, Czech Republic

Received 11 February 2014; received in revised form 22 January 2015; accepted 24 March 2015

Available online 1 April 2015

Dedicated to Francesc Esteva on the occasion of his 70th birthday

Abstract

We present an approach to decomposition and factor analysis of matrices with ordinal data. The matrix entries are grades to which objects represented by rows satisfy attributes represented by columns, e.g. grades to which an image is red, a product has a given feature, or a person performs well in a test. We assume that the grades are taken from bounded scales equipped with certain aggregation operators that are involved in the decompositions. Particular cases of the decompositions include the well-known Boolean matrix decomposition, and the sup-t-norm and inf-residuum decompositions. We consider the problem of decomposition of a given matrix into a product of two matrices with grades such that the number of factors, i.e. the inner dimension, be as small as possible. We observe that computing such decompositions is NP-hard and present a greedy approximation algorithm. Our algorithm is based on a geometric insight provided by a theorem identifying particular rectangular-shaped submatrices as optimal factors for the decompositions. These factors correspond to fixpoints of certain Galois connections associated with the input matrix, which are called formal concepts, and allow an easy interpretation of the decomposition. We present illustrative examples and experimental evaluation of the algorithm.

Keywords: Matrix decomposition; Ordinal data; Residuation; Galois connection; Algorithm

1. Introduction

1.1. Problem description

In traditional approaches to dimensionality reduction, such as factor analysis, a decomposition (factorization) of an object–variable matrix is sought into an object–factor matrix and a factor-variable matrix with the number of factors reasonably small. The factors are considered as new variables, hidden in the data and likely more fundamental than the original variables. Computing the factors and interpreting them is the central topic of this paper.

We consider decompositions of matrices I with a particular type of ordinal data. Namely, each entry I_{ij} of I represents a grade which the object corresponding to the ith row has, or is incident with, the attribute corresponding to

* Corresponding author.
E-mail addresses: radim.belohlavek@acm.org (R. Belohlavek), vychodil@acm.org (V. Vychodil).

http://dx.doi.org/10.1016/j.fss.2015.03.020
0165-0114/

the jth column. Examples of such data are results of questionnaires where respondents (rows) rate services, products, etc., according to various criteria (columns); results of performance evaluation of people (rows) by various tests (columns); or binary data in which case there are only two grades, 0 (no, failure) and 1 (yes, success). Our goal is to decompose an $n \times m$ object–attribute matrix I into a product

$$I = A \circ B \tag{1}$$

of an $n \times k$ object–factor matrix A and a $k \times m$ factor–attribute matrix B with the number k of factors as small as possible.

The scenario is thus similar to ordinary matrix decomposition problems but there are important differences. First, we assume that the entries of I, i.e. the grades, as well as the entries of A and B are taken from bounded scales L of grades, such as the real unit interval $L = [0, 1]$ or the Likert scale $L = \{1, \ldots, 5\}$ of degrees of satisfaction. Second, the matrix composition operation $\circ$ used in our decompositions is not the usual matrix product. Instead, we use a general product based on supremum-preserving aggregation operators introduced in [5,6], see also [12]. Two important, well-known [2,11] cases of this product are the sup-t-norm-product defined by

$$(A \circ B)_{ij} = \bigvee_{l=1}^{k} A_{il} \otimes B_{lj}, \tag{2}$$

and the inf-residuum-product (denoted also by $\triangleleft$) defined by

$$(A \circ B)_{ij} = \bigwedge_{l=1}^{k} A_{il} \to B_{lj}, \tag{3}$$

where $\otimes$ and $\to$ denote a (left-)continuous t-norm and its residuum [11,15], and $\bigvee$ and $\bigwedge$ denote the supremum and infimum. The ordinary Boolean matrix product is a particular case of the sup-t-norm product in which the scale L has 0 and 1 as the only grades and $a \otimes b = \min(a, b)$. It is to be emphasized that we attempt to treat graded incidence data in a way which is compatible with its semantics. This need has been recognized long ago in mathematical psychology, in particular in measurement theory [16]. For example, even if we represent the grades by numbers such as $0 \sim$ strongly disagree, $\frac{1}{4} \sim$ disagree, $\ldots$, $1 \sim$ strongly agree, addition, multiplication by real numbers, and linear combination of graded incidence data may not have natural meaning. Consequently, decomposition of a matrix I with grades into the ordinary matrix product of arbitrary real-valued matrices A and B may suffer from a difficulty to interpret A and B, see [20,28]. In this paper, we present an algorithm which is based on a theorem from [6] regarding the role of fixpoints of certain Galois connections associated with I as factors for decomposition of I. This is important both from the technical viewpoint, since due to [6] optimal decompositions may be obtained this way, and the knowledge discovery viewpoint, since the fixpoints, called formal concepts may naturally be interpreted. The algorithm runs in polynomial time and delivers suboptimal decompositions. This is a necessity because, as we show, computing optimal decompositions is an NP-hard optimization problem. In addition, we present an illustrative example demonstrating the usefulness of such decompositions, and an experimental evaluation of the algorithm.

1.2. Related work

Recently, new methods of matrix decomposition and dimensionality reduction have been developed. One aim is to have methods which are capable of discovering possibly non-linear relationships between the original space and the lower dimensional space [23,29]. Another is driven by the need to take into account constraints imposed by the semantics of the data. Examples include nonnegative matrix factorization, in which the matrices are constrained to those with nonnegative entries and which leads to additive parts-based discovery of features in data [17]. Another example, relevant to this paper, is Boolean matrix decomposition. Early work on this problem was done in [22,26]. Recent work on this topic includes [7,9,19,20,22]. As was mentioned above, Boolean matrix decomposition is a particular case of the problem considered in this paper. Note also that partly related to this paper are methods for decomposition of binary matrices into non-binary ones such as [18,24,25,27,31], see also [28] for further references.

2. Decomposition and factors

2.1. Decomposition and the factor model

As was mentioned above, we assume that the matrix entries contain elements from scales (grades) equipped with certain aggregation operators. In particular, we assume a general model of (1) in which the entries of A, B, and I

are elements of three complete lattices $\langle L_1, \leq_1\rangle$, $\langle L_2, \leq_2\rangle$, and $\langle L_3, \leq_3\rangle$. That is, we assume $A_{il} \in L_1$, $B_{lj} \in L_2$, and $I_{ij} \in L_3$. We denote the operations as usual, adding subscript i; for instance, the infima, suprema, the least, and the greatest element in L_2 are denoted by $\bigwedge_2$, $\bigvee_2$, 0_2, and 1_2, respectively. Moreover, we assume a supremum-preserving aggregation operation $\Box : L_1 \times L_2 \to L_3$, i.e. assume that for any $a, a_j \in L_1$ $(j \in J)$, $b, b_{j'} \in L_2$ $(j' \in J')$,

$$(\textstyle\bigvee_{1 j\in J} a_j) \Box b = \bigvee_{3 j\in J}(a_j \Box b) \quad \text{and} \quad a \Box (\bigvee_{2 j'\in J'} b_{j'}) = \bigvee_{3 j'\in J'}(a \Box b_{j'}). \tag{4}$$

We call the structure consisting of the $\langle L_i, \leq_i\rangle$s and $\Box$ a (*supremum-preserving*) *aggregation structure*, denote it by $\mathbf{L}$, and refer to [5] for further information.

The composition $\circ$ in (1) is then defined by

$$(A \circ B)_{ij} = \textstyle\bigvee_{3 l=1}^{k} A_{il} \Box B_{lj}. \tag{5}$$

Particular cases of this model include the following.

(a) (See [5].) Let every $\langle L_i, \leq_i\rangle$ be the set $\{0, 1\}$ equipped with the natural ordering and put $a \Box b = \min(a, b)$. Then (5) becomes the well-known Boolean matrix product [14], i.e. $(A \circ B)_{ij} = \max_{l=1}^{k} \min(A_{il}, B_{lj})$.

(b) (See [5].) Let $\langle L, \wedge, \vee, \otimes, \to, 0, 1\rangle$ be a complete residuated lattice with a partial order $\leq$ [30]. That is, $\langle L, \leq\rangle$ is a complete lattice; $\otimes$ is a commutative and associative operation on L with a neutral element 1; $\otimes$ and $\to$ satisfy adjointness, i.e. $a \otimes b \leq c$ iff $a \leq b \to c$. Letting $L_i = L$, $\leq_i = \leq$ for $i = 1, 2, 3$, and $\Box$ be $\otimes$, we obtain an aggregation structure for which (5) becomes the sup-$\otimes$-product (2). Since the two-element Boolean algebra is a particular case of a residuated lattice, the present case includes (a) as a particular example.

(c) (See [5].) If we take $L_i = L$ $(i = 1, 2, 3)$ as in (b) but now, let $\leq_1 = \leq$, $\leq_2 = \leq_3 = \leq^{-1}$ (the inverse order), and let $\Box$ be $\to$, we obtain an aggregation structure again. Note that in this case, conditions (4) become

$$(\textstyle\bigvee_{j\in J} a_j) \to b = \bigwedge_{j\in J}(a_j \to b) \quad \text{and} \quad a \to (\bigwedge_{j\in J} b_j) = \bigwedge_{j\in J}(a \to b_j)$$

which are well-known properties of residuated lattices. In this case, (5) becomes the inf-residuum-product (3).

(d) Let L be a chain and $\Box : L \times L \to L$ an isotone operation. Take finite subsets $L_1, L_2 \subseteq L$ and put $L_3 = \{a_1 \Box a_2 \mid a_1 \in L_1, a_2 \in L_2\}$. Then L_1, L_2, L_3, and the restriction of $\Box$ to $L_1 \times L_2$ is an aggregation structure. Case (a) results in this way by putting $L = [0, 1]$, $L_1 = L_2 = \{0, 1\}$, and $\Box$ being an arbitrary t-norm. (One may easily extend this example from chains to complete lattices.)

Note that residuated lattices are the main structures used as structures of truth degrees in formal fuzzy logic [11, 13]: the elements $a \in L$ are interpreted as truth degrees, $\otimes$ and $\to$ are interpreted as (truth functions of) conjunction and implication. Important examples include those with $L = [0, 1]$ and $\otimes$ being a continuous t-norm, such as $a \otimes b = \min(a, b)$ (Gödel t-norm), $a \otimes b = a \cdot b$ (Goguen t-norm), and $a \otimes b = \max(0, a + b - 1)$ (Łukasiewicz t-norm); or L being a finite chain equipped with the restriction of Gödel t-norm, Łukasiewicz t-norm, or other suitable operation. We refer the reader for further examples of residuated lattices to [11,13,15].

Consider now the meaning of the factor model given by (1) and (5). The matrices A and B represent relationships between objects and factors, and between factors and the original attributes. We interpret A_{il} as the degree to which the factor l applies to the object i, i.e. the truth degree of the proposition "factor l applies to object i"; and B_{lj} as the degree to which the attribute j is a particular manifestation of the factor l, i.e. the truth degree of the proposition "attribute j is a manifestation of factor l". Consider first the particular case described in (b) above. Due to the basic principles of fuzzy logic, the truth degree $||\varphi\&\psi||$ (& stands for *and*) of the proposition $\varphi\&\psi$ is $||\varphi|| \otimes ||\psi||$ and the existential quantifier is modeled by $\bigvee$. Therefore, (5) implies that $I = A \circ B$ says: the degree I_{ij} to which the object i has the attribute j equals the degree of the proposition "there exists factor l such that l applies to i *and* j is a particular manifestation of l". In the general case, the situation is the same except a general aggregation possibly different from *and* is involved. As the nature of the relationship between objects and attributes via factors is traditionally of interest, it is worth noting that in our case, the attributes are expressed by means of factors in a non-linear manner, see [6] for details.

2.2. Factors for decomposition

We now recall a result from [6] saying that optimal decompositions of I may be attained by using as factors the fixpoints of certain Galois connections induced by the input matrix I, called formal concepts of I [4,10]. Denote by

L^U the set of all L-sets in a set U, i.e. the set of all mappings from U to L, and put $X = \{1, \ldots, n\}$ (objects) and $Y = \{1, \ldots, m\}$ (attributes).

A *formal concept* of I is any pair $\langle C, D\rangle$ consisting of $C \in L_1^X$ and $D \in L_2^Y$ for which $C^{\uparrow} = D$ and $D^{\downarrow} = C$ where the operators ${}^{\uparrow}: L_1^X \to L_2^Y$ and ${}^{\uparrow}: L_2^Y \to L_1^X$ are defined by

$$C^{\uparrow}(j) = \textstyle\bigwedge_{2 i\in X}(C(i) \circ_{\Box} I_{ij}) \quad \text{and} \quad D^{\downarrow}(i) = \textstyle\bigwedge_{1 j\in Y}(I_{ij} \,{}_{\Box}\!\circ\, D(j)), \tag{6}$$

where the operations $\circ_{\Box} : L_1 \times L_3 \to L_2$ and ${}_{\Box}\!\circ : L_3 \times L_2 \to L_1$ are the adjoints to $\Box$ defined by

$$a_1 \circ_{\Box} a_3 = \textstyle\bigvee_2 \{a_2 \mid a_1 \Box a_2 \leq_3 a_3\} \quad \text{and} \quad a_3 \,{}_{\Box}\!\circ\, a_2 = \textstyle\bigvee_1 \{a_1 \mid a_1 \Box a_2 \leq_3 a_3\}. \tag{7}$$

${}^{\uparrow}$ and ${}^{\downarrow}$ form Galois connections with additional properties regarding the adjoints [3]. The set

$$\mathcal{B}(X, Y, I) = \{\langle C, D\rangle \in L_1^X \times L_2^Y \mid C^{\uparrow} = D \text{ and } D^{\downarrow} = C\}$$

of all formal concepts of I is called the *concept lattice* of I and forms indeed a complete lattice when equipped with a natural ordering, see [4] for its structural characterization. Formal concepts are simple models of concepts in the sense of traditional, Port-Royal logic and have a natural interpretation. For a formal concept $\langle C, D\rangle$, C and D are called the extent and the intent of $\langle C, D\rangle$; the degrees $C(i)$ and $D(j)$ are interpreted as the degrees to which the concept applies to object i and attribute j, respectively.

For a set $\mathcal{F} = \{\langle C_1, D_1\rangle, \ldots, \langle C_k, D_k\rangle\}$ of formal concepts of I with a fixed order given by the indices, denote by $A_{\mathcal{F}}$ and $B_{\mathcal{F}}$ the $n \times k$ and $k \times m$ matrices defined by

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

If $I = A_{\mathcal{F}} \circ B_{\mathcal{F}}$, $\mathcal{F}$ can be seen as a set of factors fully explaining the data. In such a case, we call the formal concepts in $\mathcal{F}$ *factor concepts*. In this case, the factors have a natural, easy-to-understand meaning as is demonstrated in Section 4. The following theorem was proven in [6].

Theorem 1. *Let $I = A \circ B$ for an $n \times k$ matrix A and a $k \times m$ matrix B. Then there exists a set $\mathcal{F} \subseteq \mathcal{B}(X, Y, I)$ with $|\mathcal{F}| \leq k$ such that for the $n \times |\mathcal{F}|$ and $|\mathcal{F}| \times m$ matrices $A_{\mathcal{F}}$ and $B_{\mathcal{F}}$ we have $I = A_{\mathcal{F}} \circ B_{\mathcal{F}}$.*

The theorem implies that optimal decompositions of I may be obtained by using the formal concepts of I as factors. Consequently, one may restrict the search for factors to the set of formal concepts of I.

3. Algorithm and complexity of decompositions

Let us define our problem precisely. For a given (that is, constant for the problem) aggregation structure **L** consisting of complete lattices L_1, L_2, L_3, and an aggregation operation $\Box$, the problem we discuss is a minimization (optimization) problem [1] specified as follows:

PROBLEM:	DECOMP(**L**)
INPUT:	$n \times m$ matrix I with entries in L_3;
FEASIBLE SOLUTION:	$n \times k$ and $k \times m$ matrices A and B with entries in L_1 and L_2, respectively, for which $I = A \circ B$;
COST OF SOLUTION:	k.

Throughout this section, we assume that L_is are linearly ordered, i.e. $a \leq_i b$ or $b \leq_i a$ for any two degrees $a, b \in L_i$ ($i = 1, 2, 3$). (The general, non-linear case can be handled with no substantial difficulty but we prefer to keep things simple, particularly because of the practical importance of the linear case.) Moreover, we assume that the aggregation structures considered are *exhaustive* by which we mean that

for each $a_3 \in L_3$ there exist $a_1 \in L_1$ and $a_2 \in L_2$ such that $a_3 = a_1 \Box a_2$.

Clearly, the aggregation structures described in (a), (b), (c), and (d) in Section 2.1 are exhaustive. Exhaustivity is a natural condition to consider in our case:

Lemma 1. *An aggregation structure is exhaustive if and only if every matrix is decomposable, i.e. for every I there exist A and B such that $I = A \circ B$.*

Proof. Assume exhaustivity. For a given $n \times m$ matrix I, put $k = n \times m$, write indices $l = 1, \dots, k$ in the form $l = \langle i, j \rangle$ where $1 \leq i \leq n$, $1 \leq j \leq m$, and consider the $n \times k$ and $k \times m$ matrices A and B defined by

$$A_{i,\langle i',j'\rangle} = \begin{cases} a^{ij'} & \text{for } i = i' \\ 0_1 & \text{for } i \neq i' \end{cases} \quad \text{and} \quad B_{\langle i',j'\rangle,j} = \begin{cases} b^{i'j} & \text{for } j = j' \\ 0_2 & \text{for } j \neq j' \end{cases}$$

where $a^{ij} \in L_1$ and $b^{ij} \in L_2$ are any elements satisfying $a^{ij} \square b^{ij} = I_{ij}$ for every i, j (such elements exist due to exhaustivity). Since $a_1 \square 0_2 = 0_3$ and $0_1 \square a_2 = 0_3$ for every $a_1 \in L_1$ and $a_2 \in L_2$ [5], one easily verifies that $A \circ B = I$. Conversely, for any $a \in L_3$ consider the 1×1 matrix I for which $I_{11} = a$. The assumption implies that $I = A \circ B$ for some $1 \times k$ and $k \times 1$ matrices A and B with entries in L_1 and L_2, i.e. $a = \bigvee_{l=1}^{k}(A_{il} \square B_{lj})$. Due to the linearity assumption we obtain $a = A_{il} \square B_{lj}$ for some l, verifying exhaustivity. □

As indicated above, due to Theorem 1, we look for feasible solutions A and B in the form $A_{\mathcal{F}}$ and $B_{\mathcal{F}}$ for some $\mathcal{F} \subseteq \mathcal{B}(X, Y, I)$. Therefore, the algorithm we present in Section 3.1 computes a set $\mathcal{F}$ of formal concepts of I for which $A_{\mathcal{F}}$ and $B_{\mathcal{F}}$ is a good feasible solution. Our algorithm runs in polynomial time but produces only suboptimal solutions, i.e. $|\mathcal{F}| \geq \rho(I)$, where

$$\rho(I) = \min\{k \mid I = A \circ B \text{ for some } n \times k \text{ and } k \times m \text{ matrices } A \text{ and } B\}.$$

As is shown in Section 3.2, this is a necessity because the decomposition problem is an NP-hard optimization problem. That is, unless P = NP, there does not exist a polynomial time algorithm producing optimal solutions to the decomposition problem. We demonstrate experimentally in Section 4, however, that the quality of the solutions provided by our algorithm is reasonable.

In this section as well as in Section 4 we utilize the following "geometric" viewpoint. Every formal concept $\langle C_l, D_l \rangle \in \mathcal{F}$ induces a matrix $J_l = C_l \square D_l$ given by

$$(C_l \square D_l)_{ij} = C_l(i) \square D_l(j), \tag{8}$$

the rectangular matrix induced by $\langle C_l, D_l \rangle$ (in the Boolean case, the 1s in $C_l \square D_l$ form a rectangular area). Then $I = A_{\mathcal{F}} \circ B_{\mathcal{F}}$ means that

$$I_{ij} = (J_1)_{ij} \vee_3 \cdots \vee_3 (J_k)_{ij}, \tag{9}$$

i.e. I is the $\bigvee_3$-superposition of J_ls.

3.1. Algorithm

Due to the linearity assumption, (9) implies that $I = A_{\mathcal{F}} \circ B_{\mathcal{F}}$ if and only if for each $\langle i, j \rangle \in \{1, \dots, n\} \times \{1, \dots, m\}$ there exists $\langle C_l, D_l \rangle \in \mathcal{F}$ for which

$$I_{ij} = C_l(i) \square D_l(j). \tag{10}$$

In case of (10), we say that $\langle C_l, D_l \rangle$ *covers* $\langle i, j \rangle$. This allows us to see that the problem of finding a set $\mathcal{F}$ of formal concepts of I for which $I = A_{\mathcal{F}} \circ B_{\mathcal{F}}$ can be reformulated as the problem of finding $\mathcal{F}$ such that every pair from the set

$$\mathcal{U} = \{\langle i, j \rangle \mid I_{ij} \neq 0_3\} \tag{11}$$

is covered by some $\langle C_l, D_l \rangle \in \mathcal{F}$. Since $C_l(i) \square D_l(j) \leq_3 I_{ij}$ is always the case [6], we need not worry about "overcovering". We now see that every instance of the decomposition problem may be rephrased as an instance of the well-known set cover problem, see e.g. [1] in which the set $\mathcal{U}$ is to be covered by the sets in

$$\{\{\langle i, j \rangle\,;\ I_{ij} \leq_3 C(i) \square D(j)\} \mid \langle C, D \rangle \in \mathcal{B}(X, Y, I)\}.$$

Accordingly, one can use the well-known greedy approximation algorithm [1] for solving set cover to select a set $\mathcal{F}$ for formal concepts for which $I = A_{\mathcal{F}} \circ B_{\mathcal{F}}$. However, this is costly from the computational point of view. Namely, one would need to compute the possibly rather large set $\mathcal{B}(X, Y, I)$ first and, worse, repeatedly iterate over this set in the greedy set cover algorithm.

Instead, we propose a different greedy algorithm. The idea is to supply promising candidate factor concepts *on demand* during the factorization procedure, as opposed to computing all candidate factor concepts beforehand. The algorithm generates the promising candidate factor concepts by looking for promising columns. A technical property which we utilize is the fact that for each formal concept $\langle C, D\rangle$,

$$D = \bigcup_{j \in Y} \{{}^{D(j)}/j\}^{\downarrow\uparrow},$$

i.e. each intent D is a union of intents $\{{}^{D(j)}/j\}^{\downarrow\uparrow}$ [4]. Here, $\{{}^{D(j)}/j\}$ denotes the L_2-set in Y defined by

$$\{{}^{D(j)}/j\}(j') = \begin{cases} D(j) & \text{if } j' = j, \\ 0_2 & \text{if } j' \neq j, \end{cases}$$

and $\bigcup$ denotes the union based on $\bigvee_2$, i.e. $(D_1 \cup D_2)(j) = D_1(j) \vee_2 D_2(j)$ for any $D_1, D_2 \in L_2^Y$. As a consequence, we may construct any formal concept by adding sequentially $\{{}^{a}/j\}^{\downarrow\uparrow}$ to the empty set of attributes. Our algorithm follows a greedy approach that makes us select $j \in Y$ and a degree $a \in L_2$ that maximize the size of

$$D \oplus_a j = \{\langle k, l\rangle \in \mathcal{U} \mid D^{+\downarrow}(k) \,\square\, D^{+\downarrow\uparrow}(l) \geq_3 I_{kl}\}, \tag{12}$$

where $D^+ = D \cup \{{}^{a}/j\}$ and $\mathcal{U}$ denotes the set of $\langle i, j\rangle$ for which the corresponding entry I_{ij} is not covered yet. At the start, $\mathcal{U}$ is initialized according to (11). As the algorithm proceeds, $\mathcal{U}$ gets updated by removing from it the pairs $\langle i, j\rangle$ which have been covered by the selected formal concept $\langle C, D\rangle$. Note that $|D \oplus_a j|$ is the number of entries of I which are covered by formal concept $\langle D^{+\downarrow}, D^{+\downarrow\uparrow}\rangle$, i.e. by the concept generated by D^+. Therefore, instead of going through all possible formal concepts and selecting the factors from them, we just go through columns and degrees and add them repeatedly as to maximize $|D \oplus_a j|$, until such addition is impossible. The resulting algorithm is summarized below.

FIND-FACTORS(I)
1 $\mathcal{U} \leftarrow \{\langle i, j\rangle \mid I_{ij} \neq 0_3\}$
2 $\mathcal{F} \leftarrow \emptyset$
3 **while** $\mathcal{U} \neq \emptyset$
4   **do** $D \leftarrow \emptyset$
5       $V \leftarrow 0$
6   **select** $\langle j, a\rangle$ **that maximizes** $|D \oplus_a j|$
7   **while** $|D \oplus_a j| > V$
8   **do** $V \leftarrow |D \oplus_a j|$
9             $D \leftarrow (D \cup \{{}^{a}/j\})^{\downarrow\uparrow}$
10   **select** $\langle j, a\rangle$ **that maximizes** $|D \oplus_a j|$
11       $C \leftarrow D^{\downarrow}$
12       $\mathcal{F} \leftarrow \mathcal{F} \cup \{\langle C, D\rangle\}$
13   **for** $\langle i, j\rangle \in \mathcal{U}$
14   **do if** $I_{ij} \leq_3 C(i) \,\square\, D(j)$
15   **then**
16                   $\mathcal{U} \leftarrow \mathcal{U} \setminus \{\langle i, j\rangle\}$
17 **return** $\mathcal{F}$

The main loop of the algorithm (lines 3–16) is executed until all the non-zero entries of I are covered by at least one factor in $\mathcal{F}$. The code between lines 4 and 10 constructs an intent by adding the most promising columns. After such an intent D is found, we construct the corresponding factor concept and add it to $\mathcal{F}$. The loop between lines 13 and 16 ensures that all matrix entries covered by the last factor are removed from $\mathcal{U}$. Obviously, the algorithm is sound and finishes after finitely many steps, polynomial in terms of n and m, with a set $\mathcal{F}$ of factor concepts.

3.2. Complexity of finding optimal decompositions

As mentioned above, there is no guarantee that our algorithm finds an optimal decomposition, i.e. the one with $k = \rho(I)$. The following theorem shows that, unless P = NP, no polynomial time algorithm computing optimal decompositions exists.

In fact, the theorem applies to a broad class of aggregation structures satisfying

$$a_1 \Box a_2 = 1_3 \quad \text{implies} \quad a_2 = 1_2 \tag{13}$$

for every $a_1 \in L_1$ and $a_2 \in L_2$. One easily checks that (13) is satisfied by any of the aggregation structures in (a), (b), and (c) in Section 2.1, but is not satisfied by (d) in general (see Remark 1). Moreover, note that (13) is a weakened form of a natural condition requiring that an aggregation of two degrees, a_1 and a_2, yields the top degree only if both a_1 and a_2 are top degrees. In the following theorem we therefore assume that **L** is an exhaustive aggregation structure satisfying (13).

Theorem 2. DECOMP(**L**) *is NP-hard.*

Proof. By definition of NP-hardness of optimization problems, we need to show that the corresponding decision problem is NP-complete. The decision problem, which we denote by Π in what follows, is to decide for a given I and positive integer k whether there exists a decomposition $I = A \circ B$ with the inner dimension k or smaller. Now, Π is NP-complete because the decision version of the set basis problem, which is known to be NP-complete [26], is reducible to it. The decision version of the set basis problem is: Given a collection $S = \{S_1, \ldots, S_n\}$ of sets $S_i \subseteq \{1, \ldots, m\}$ and a positive integer k, is there a collection $P = \{P_1, \ldots, P_k\}$ of subsets $P_l \subseteq \{1, \ldots, m\}$ such that for every S_i there is a subset $Q_i \subseteq \{P_1, \ldots, P_k\}$ for which $\bigcup Q_i = S_i$ (i.e., the union of all sets from Q_i is equal to S_i)? This problem is readily reducible to Π: Given S, define an $n \times m$ matrix I by $I_{ij} = 1_3$ if $j \in S_i$ and $I_{ij} = 0_3$ if $j \notin S_i$.

To verify that the reduction works, let us observe that if P_l and Q_l represent a solution to S and k, i.e. $|P| \le k$, the $n \times |P|$ and $|P| \times m$ matrices A and B defined by $B_{lj} = 1_2$ if $j \in P_l$ and $B_{lj} = 0_2$ if $j \notin P_l$, and $A_{il} = 1_1$ if $P_l \in Q_i$ and $A_{il} = 0_1$ if $P_l \notin Q_i$, represent a solution to Π. To check this fact, note that in every aggregation structure we have $0_1 \Box 1_2 = 1_1 \Box 0_2 = 0_3$ [5] and observe that since **L** is exhaustive, we have $1_1 \Box 1_2 = 1_3$. Namely, exhaustivity implies the existence of a_1 and a_2 with $1_3 = a_1 \Box a_2$ and the monotony of $\Box$, which is a consequence of (4), yields $a_1 \Box a_2 \le 1_1 \Box 1_2$, hence $1_3 = 1_1 \Box 1_2$. One now easily verifies that $I = A \circ B$.

Conversely, let A and B be $n \times K$ and $K \times m$ matrices which represent a solution to Π, i.e. $I = A \circ B$ and $K \le k$. Let a be the smallest entry of A for which $a \Box 1_2 = 1_3$. Define $P_l = \{j \in \{1, \ldots, m\} \mid B_{lj} = 1_2\}$, $l = 1, \ldots, K$, and $Q_i = \{P_l \mid A_{il} \ge_1 a\}$, $i = 1, \ldots, n$. We need to check that $S_i = \bigcup Q_i$ for every i. We have $j \in S_i$ iff $I_{ij} = 1_3$ iff $(A \circ B)_{ij} = 1$ iff $A_{il} \Box B_{lj} = 1_3$ for some l. Now, (13) and the definition of a, and the isotony of $\Box$ implies that the last condition is equivalent to the fact that $A_{lj} \ge_1 a$ and $B_{lj} = 1_2$ for some l which is equivalent to $P_l \in Q_i$ and $j \in P_l$ for some l, i.e. to $j \in \bigcup Q_i$. □

Remark 1. Note that $1_1 \Box 1_2 = 1_3$, which is a consequence of exhaustivity used in the proof, need not hold in every aggregation structure [5]. Note also that if **L** does not satisfy (13), the reduction may not work. As an example, consider the aggregation structure where both L_1 and L_2 are the three-element chain $0 < a < 1$, L_3 is the two element chain $0 < 1$, and $\Box$ is given by

□	0	a	1
0	0	0	0
a	0	0	1
1	0	1	1

.

This aggregation structure is a particular example of (d) in Section 2.1, is exhaustive but does not satisfy (13) because $1 \Box a = 1$. Then the instance of Π given by $I = \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix}$, and $k = 1$ is positive since

$$I = \begin{pmatrix} 1 \\ a \end{pmatrix} \circ \begin{pmatrix} a & 1 \end{pmatrix},$$

Table 1
2004 Olympic Games decathlon.
Scores of Top 5 Athletes

	10	*lj*	*sp*	*hj*	40	11	*di*	*pv*	*ja*	15
Sebrle	894	1020	873	915	892	968	844	910	897	680
Clay	989	1050	804	859	852	958	873	880	885	668
Karpov	975	1012	847	887	968	978	905	790	671	692
Macey	885	927	835	944	863	903	836	731	715	775
Warners	947	995	758	776	911	973	741	880	669	693

Incidence data table with graded attributes

	10	*lj*	*sp*	*hj*	40	11	*di*	*pv*	*ja*	15
Sebrle	0.50	1.00	1.00	1.00	0.75	1.00	0.75	0.75	1.00	0.75
Clay	1.00	1.00	0.75	0.75	0.50	1.00	0.75	0.50	1.00	0.50
Karpov	1.00	1.00	0.75	0.75	1.00	1.00	1.00	0.25	0.25	0.75
Macey	0.50	0.50	0.75	1.00	0.75	0.50	0.75	0.25	0.50	1.00
Warners	0.75	0.75	0.50	0.50	0.75	1.00	0.25	0.50	0.25	0.75

Legend: 10—100 meters sprint race; *lj*—long jump; *sp*—shot put; *hj*—high jump; 40—400 meters sprint race; 11—110 meters hurdles; *di*—discus throw; *pv*—pole vault; *ja*—javelin throw; 15—1500 meters run.

but the corresponding instance of the decision version of the set basis problem, which is given by $S = \{\{1, 2\}, \{2\}\}$, $k = 1$, is negative because, obviously, no singleton $P = \{P_1\}$ is a basis of S.

4. Examples and experiments

In Section 4.1, we examine in detail a factor analysis of 2004 Olympic Decathlon data. We include this example to illustrate that the algorithm developed in this paper can be used to obtain reasonable factors from data with grades. In Section 4.2, we present an experimental evaluation of our algorithm.

4.1. Decathlon data

Grades of ordinal scales are conveniently represented by numbers. In such case, we assume these numbers are normalized and taken from the unit interval [0, 1]. As an example, the Likert scale is represented by $L = \{0, \frac{1}{4}, \frac{1}{2}, \frac{3}{4}, 1\}$. Due to the well-known Miller's 7 ± 2 phenomenon [21], one might argue that in order to support comprehension by users, one should restrict to small scales.

In this section, we explore factors explaining the athletes' performance in the 2004 Olympic Games decathlon. Table 1 (top) contains the results of top five athletes in this event in points obtained using the IAAF Scoring Tables for Combined Events. Note that the IAAF Scoring Tables provide us with a scaling function assigning the scale values to athletes.

We first transform the data from Table 1 (top) to a five-element scale

$$L = \{0.00, 0.25, 0.50, 0.75, 1.00\} \tag{14}$$

by a natural transformation and rounding. Namely, for every discipline, we first take the lowest and highest scores achieved among all athletes who finished the event, see Table 2. We then perform a linear transform of values from [lowest, highest] to [0, 1]. Thus, for *lj* (long jump), we obtain

$$f_{lj}(x) = \frac{x - 723}{(1050 - 723)} = \frac{x - 723}{(1050 - 723)} = \frac{x - 723}{327} \in [0, 1] \tag{15}$$

and round this value to the closest value in the discrete scale (14). This way, we transform Table 1 (top) to Table 1 (bottom). As a consequence, the factors then have a simple reading. Namely, the grades to which a factor applies to an athlete can be described in natural language as "not at all", "little bit", "half", "quite", "fully", or the like.

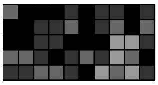
Using shades of gray to represent grades, the matrix I corresponding to Table 1 (bottom) can be visualized as:

Table 2
Lowest and highest scores in the 2004 Olympic Games decathlon.

	10	*lj*	*sp*	*hj*	40	11	*di*	*pv*	*ja*	15
lowest	782	723	672	670	673	803	661	673	598	466
highest	989	1050	873	944	968	978	905	1035	897	791

Table 3
Factor concepts.

F_i	Extent	Intent
F_1	{0.5/S, C, K, 0.5/M, 0.75/W}	{10, lj, 0.75/sp, 0.75/hj, 0.5/40, 11, 0.5/di, 0.25/pv, 0.25/ja, 0.5/15}
F_2	{S, 0.75/C, 0.25/K, 0.5/M, 0.25/W}	{0.5/10, lj, sp, hj, 0.75/40, 11, 0.75/di, 0.75/pv, ja, 0.75/15}
F_3	{0.75/S, 0.5/C, 0.75/K, M, 0.5/W}	{0.5/10, 0.5/lj, 0.75/sp, hj, 0.75/40, 0.5/11, 0.75/di, 0.25/pv, 0.5/ja, 15}
F_4	{S, 0.75/C, 0.75/K, 0.5/M, W}	{0.5/10, 0.75/lj, 0.5/sp, 0.5/hj, 0.75/40, 11, 0.25/di, 0.5/pv, 0.25/ja, 0.75/15}
F_5	{0.75/S, 0.75/C, K, 0.75/M, 0.25/W}	{0.75/10, 0.75/lj, 0.75/sp, 0.75/hj, 0.75/40, 0.75/11, di, 0.25/pv, 0.25/ja, 0.75/15}
F_6	{0.75/S, 0.5/C, K, 0.75/M, 0.75/W}	{0.75/10, 0.75/lj, 0.75/sp, 0.75/hj, 40, 0.75/11, 0.5/di, 0.25/pv, 0.25/ja, 0.75/15}
F_7	{S, C, 0.25/K, 0.5/M, 0.25/W}	{0.5/10, lj, 0.75/sp, 0.75/hj, 0.5/40, 11, 0.75/di, 0.5/pv, ja, 0.5/15}⟩



In this example, we used the aggregation structure in which $L_i = L$ $(i = 1, 2, 3)$ and $\Box$ is the restriction of the Łukasiewicz t-norm (see Section 2.1) on L. Note that this is a particular example of (d) in Section 2.1.

The algorithm described in Section 3.1 found a set $\mathcal{F} = \{F_l = \langle C_l, D_l\rangle \mid l = 1, \ldots, 7\}$ of seven formal concepts which factorize I, i.e. for which $I = A_{\mathcal{F}} \circ B_{\mathcal{F}}$. These factor concepts are shown in Table 3 in the order in which they were produced by the algorithm. In addition, Fig. 1 shows the corresponding rectangular matrices $C_l \Box D_l$, cf. (8).

The matrices $A_{\mathcal{F}}$ and $B_{\mathcal{F}}$ are

$$A_{\mathcal{F}} = \begin{pmatrix} 0.50 & 1.00 & 0.75 & 1.00 & 0.75 & 0.75 & 1.00 \\ 1.00 & 0.75 & 0.50 & 0.75 & 0.75 & 0.50 & 1.00 \\ 1.00 & 0.25 & 0.75 & 0.75 & 1.00 & 1.00 & 0.25 \\ 0.50 & 0.50 & 1.00 & 0.50 & 0.75 & 0.75 & 0.50 \\ 0.75 & 0.25 & 0.50 & 1.00 & 0.25 & 0.75 & 0.25 \end{pmatrix},$$

$$B_{\mathcal{F}} = \begin{pmatrix} 1.00 & 1.00 & 0.75 & 0.75 & 0.50 & 1.00 & 0.50 & 0.25 & 0.25 & 0.50 \\ 0.50 & 1.00 & 1.00 & 1.00 & 0.75 & 1.00 & 0.75 & 0.75 & 1.00 & 0.75 \\ 0.50 & 0.50 & 0.75 & 1.00 & 0.75 & 0.50 & 0.75 & 0.25 & 0.50 & 1.00 \\ 0.50 & 0.75 & 0.50 & 0.50 & 0.75 & 1.00 & 0.25 & 0.50 & 0.25 & 0.75 \\ 0.75 & 0.75 & 0.75 & 0.75 & 0.75 & 0.75 & 1.00 & 0.25 & 0.25 & 0.75 \\ 0.75 & 0.75 & 0.75 & 0.75 & 1.00 & 0.75 & 0.50 & 0.25 & 0.25 & 0.75 \\ 0.50 & 1.00 & 0.75 & 0.75 & 0.50 & 1.00 & 0.75 & 0.50 & 1.00 & 0.50 \end{pmatrix}$$

and the decomposition may be visualized as:

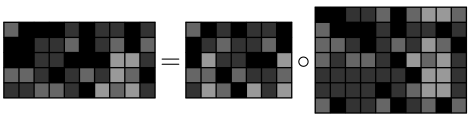


Fig. 2 demonstrates what portion of the data matrix I is explained using the first l factors in $\mathcal{F}$, i.e. using the sets $\mathcal{F}_l = \{F_1, \ldots, F_l\}$, $l = 1, \ldots, 7$. The first matrix labeled by 46% shows $A_{\mathcal{F}_1} \circ B_{\mathcal{F}_1}$ for $\mathcal{F}_1$ consisting of the first factor F_1 only. That is, the matrix is just the rectangular matrix $C_1 \Box D_1$ corresponding to $F_1 = \langle C_1, D_1\rangle$, cf. Fig. 1. As we can see, this matrix approximates I from below, in that $(A_{\mathcal{F}_1} \circ B_{\mathcal{F}_1})_{ij} \leq I_{ij}$ for all entries. The label 46% indicates that 46% of the entries of $A_{\mathcal{F}_1} \circ B_{\mathcal{F}_1}$ and I are equal. In this sense, the first factor explains 46% of the data. Note however, that several of the $54\% = 100\% - 46\%$ of the other entries of $A_{\mathcal{F}_1} \circ B_{\mathcal{F}_1}$ are close to the corresponding entries of I.

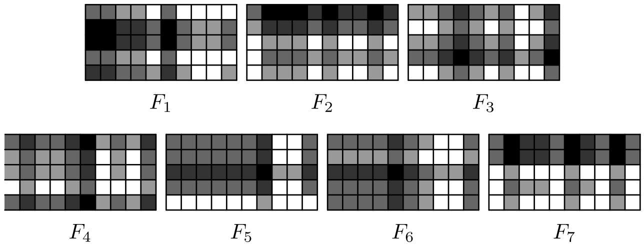


Fig. 1. Factor concepts as rectangular patterns.

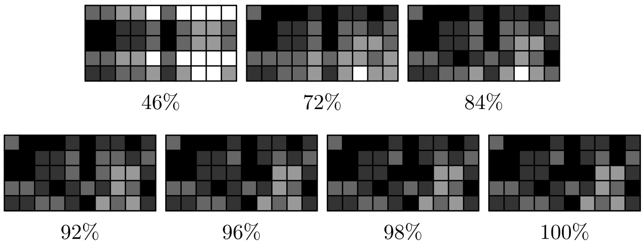


Fig. 2. $\bigvee$-superposition of factor concepts.

The second matrix in Fig. 2, with label 72%, shows $A_{\mathcal{F}_2} \circ B_{\mathcal{F}_2}$ for $\mathcal{F}_2$ consisting of F_1 and F_2. That is, the matrix demonstrates what portion of the data matrix I is explained by the first two factors. Again, $A_{\mathcal{F}_2} \circ B_{\mathcal{F}_2}$ approximates I from below and 72% of the entries of $A_{\mathcal{F}_2} \circ B_{\mathcal{F}_2}$ and I coincide now. Note again that even for the remaining 28% of entries, $A_{\mathcal{F}_2} \circ B_{\mathcal{F}_2}$ provides a reasonable approximation of I, as can be seen by comparing $A_{\mathcal{F}_2} \circ B_{\mathcal{F}_2}$ and I.

Similarly, the matrices labeled by 84%, 92%, 96%, 98%, and 100% represent $A_{\mathcal{F}_l} \circ B_{\mathcal{F}_l}$ for $l = 3, 4, 5, 6, 7$. We can conclude from the visual inspection of the matrices that already the two or three first factors explain the data reasonably well.

Let us now focus on the interpretation of the factors. Fig. 1 is helpful as it shows the clusters corresponding to the factor concepts which draw together the athletes and their performances in the events.

Factor F_1: Manifestations of this factor with grade 1 are 100 m, long jump, 110 m hurdles. This factor can be interpreted as the ability to run fast for short distances (speed). Note that this factor applies particularly to Clay and Karpov which is well known in the world of decathlon. Factor F_2: Manifestations of this factor with grade 1 are long jump, shot put, high jump, 110 m hurdles, javelin. F_2 can be interpreted as the ability to apply very high force in a very short term (explosiveness). F_2 applies particularly to Sebrle, and then to Clay, who are known for this ability. Factor F_3: Manifestations with grade 1 are high jump and 1500 m. This factor is typical for lighter, not very muscular athletes (too much muscles prevent jumping high and running long distances). Macey, who is evidently that type among decathletes (196 cm and 98 kg) is the athlete to whom the factor applies to degree 1. These are the most important factors behind data matrix I. Note that these factors were considered natural by an experienced decathlon coach, according to whom F_2 is typical of the Czech school of decathlon.

4.2. Experimental evaluation

In this section, we present experiments with exact and approximate factorization of selected publicly-available datasets and randomly generated matrices. First, we observed how close is the number of factors found by the algorithm FINDFACTORS to a known number of factors in synthetic matrices. We were generating 20×20 matrices according to various distributions of 5 grades. For the aggregation structures, we used $L_i = \{0, \frac{1}{4}, \frac{1}{2}, \frac{3}{4}, 1\}$ ($i = 1, 2, 3$) and the restriction of the Łukasiewicz or minimum t-norm $\otimes$ (see Section 2.1) on $\{0, \frac{1}{4}, \frac{1}{2}, \frac{3}{4}, 1\}$ for $\square$. The matrices were generated by multiplying $m \times k$ and $k \times n$ matrices. We then ran FINDFACTORS the algorithm to find $\mathcal{F}$ and observed how close is the number $|\mathcal{F}|$ of factors to k. The results are depicted in Table 4. The rows of Table 4 correspond to numbers $k = 5, 7, \ldots, 15$ denoting the known number of factors. For each k, we computed the average number of factors produced by the algorithm in 2000 k-factorizable matrices. The average values are written in the

Table 4
Exact factorizability.

k	Łukasiewicz ⊗ no. of factors	minimum ⊗ no. of factors
5	5.205 ± 0.460	6.202 ± 1.037
7	7.717 ± 0.878	10.050 ± 1.444
9	10.644 ± 1.316	13.379 ± 1.676
11	13.640 ± 1.615	15.698 ± 1.753
13	16.423 ± 1.879	17.477 ± 1.787
15	18.601 ± 2.016	18.721 ± 1.863

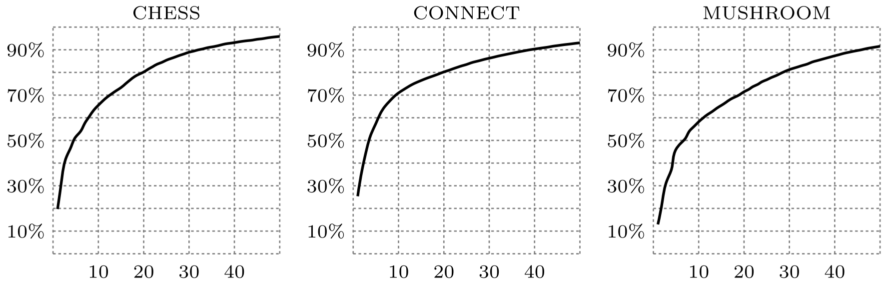


Fig. 3. Approximate factorization of Boolean matrices by first 50 factors.

form of "average number of factors ± standard deviation". We also observed that on average, the choice of a t-norm is not essential and all t-norms give approximately the same results.

As mentioned above, factorization of Boolean data is a special case of our setting, see example (a) in Section 2.1. Then, $\circ$ coincides with the Boolean matrix multiplication and the decomposition problem becomes the problem of decomposition of Boolean matrices. We performed experiments with our algorithm in this particular case with three large binary data sets (binary matrices) from the Frequent Itemset Mining Dataset Repository.[1] In particular, we considered the CHESS (3196×75 binary matrix), CONNECT (67557×129 binary matrix), and MUSHROOM (8124×119 binary matrix) data sets. The results are shown in Fig. 3. The x-axes correspond to the number of factors (from 1 up to 50 factors were observed) and the y-axes are percentages of data explained by the factors. For example, we can see that the first 10 factors of CHESS explain more than 70% of the data, i.e. $A_{\mathcal{F}} \circ B_{\mathcal{F}}$ covers more than 70% of the nonzero entries of CHESS for $|\mathcal{F}| = 10$. In all the three cases, we can see a tendency that a relatively small number of factors (compared to the number of attributes in the datasets) cover a significant part of the data.

A similar tendency can also be observed for graded incidence data. For instance, we have the algorithm for factor analysis of the FOREST FIRES [8] dataset from the UCI Machine Learning Repository.[2] In its original form, the dataset contains real values. We transformed it to a matrix with grades representing the relationship between spatial coordinates within the Montesinho park map (rows) and 50 different groups of environmental and climate conditions (columns). The matrix entries are degrees in $L = \{\frac{n}{100} \mid n \text{ is integer}, 0 \le n \le 100\}$ interpreted as degrees to which there has been a large area of burnt forest in the sector of the map under the environmental conditions. The aggregation structure is the one given by $L_i = L$ ($i = 1, 2, 3$) and the restriction of the Łukasiewicz t-norm on L. Factor analysis of data in this form can help reveal factors which contribute to forests burns in the park. The exact factorization revealed 46 factors. As in case of the Boolean datasets, a relatively small number of factors explain large portions of the data. For instance, more than 50% of the data is covered by the first 10 factors and more than 80% of the data is covered by the first 23 factors, see Fig. 4.

[1] http://fimi.cs.helsinki.fi/data/.
[2] http://archive.ics.uci.edu/ml/.

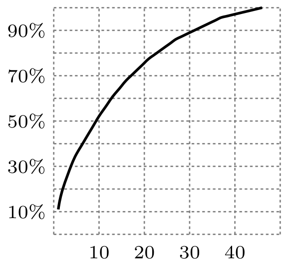


Fig. 4. Factorization of graded incidence matrix FOREST FIRES.

5. Conclusions

We presented an approach to decomposition and factor analysis of matrices with grades, i.e. of a particular form of ordinal data. The factors in this approach correspond to fixpoints of certain Galois connections associated with the input matrix, which are known as formal concepts. The approach is justified by a theorem according to which optimal decompositions are attained by using formal concepts as factors. The relationship between the factors and original attributes is a non-linear one. An advantageous feature of the model is a transparent way of treating the grades which results in good interpretability of factors. We observed that the decomposition problem is NP-hard as an optimization problem. We proposed a greedy algorithm for computing suboptimal decompositions and provided results of experiments demonstrating its behavior. Furthermore, we presented a detailed example which demonstrates that the method yields interesting factors from data.

Future research shall include the following topics. First, a comparison, both theoretical and experimental, to other methods of matrix decompositions, in particular to the methods emphasizing good interpretability, such as non-negative matrix factorization [17]. Second, an investigation of approximate decompositions of I, i.e. decompositions to A and B for which $A \circ B$ is approximately equal to I with respect to a reasonable notion of approximate equality. Third, development of further theoretical insight focusing particularly on reducing further the space of factors to which the search for factors can be restricted. Fourth, explore further the applications of the decompositions studied in this paper, particularly in areas such as psychology, sports data, or customer surveys, where ordinal data is abundant.

Acknowledgements

R. Belohlavek acknowledges support by grant No. P202/10/0262, and during revision by No. 15-17899S of the Czech Science Foundation. V. Vychodil acknowledges support by the ESF project No. CZ.1.07/2.3.00/20.0059, the project is co-financed by the European Social Fund and the state budget of the Czech Republic. This paper is an extended version of the authors' paper "Factor analysis of incidence data via novel decomposition of matrices," Proc. ICFCA 2009, 83–97.

References

[1] G. Ausiello, et al., Complexity and Approximation. Combinatorial Optimization Problems and Their Approximability Properties, Springer, 2003.
[2] W. Bandler, L.J. Kohout, Semantics of implication operators and fuzzy relational products, Int. J. Man-Mach. Stud. 12 (1980) 89–116.
[3] R. Belohlavek, Fuzzy Galois connections, Math. Log. Q. 45 (4) (1999) 497–504.
[4] R. Belohlavek, Concept lattices and order in fuzzy logic, Ann. Pure Appl. Log. 128 (1–3) (2004) 277–298.
[5] R. Belohlavek, Sup-t-norm and inf-residuum are one type of relational product: unifying framework and consequences, Fuzzy Sets Syst. 197 (2012) 45–58.
[6] R. Belohlavek, Optimal decompositions of matrices with entries from residuated lattices, J. Log. Comput. 22 (6) (2012) 1405–1425.
[7] R. Belohlavek, V. Vychodil, Discovery of optimal factors in binary data via a novel method of matrix decomposition, J. Comput. Syst. Sci. 76 (1) (2010) 3–20.
[8] P. Cortez, A. Morais, A data mining approach to predict forest fires using meteorological data, in: J. Neves, M.F. Santos, J. Machado (Eds.), New Trends in Artificial Intelligence, Proc. 13th EPIA 2007, Guimaraes, Portugal, 2007, pp. 512–523.
[9] A.A. Frolov, D. Húsek, I.P. Muraviev, P.A. Polyakov, Boolean factor analysis by Hopfield-like autoassociative memory, IEEE Trans. Neural Netw. 18 (3) (2007) 698–707.
[10] B. Ganter, R. Wille, Formal Concept Analysis. Mathematical Foundations, Springer, Berlin, 1999.

[11] S. Gottwald, A Treatise on Many-Valued Logic, Studies in Logic and Computation, vol. 9, Research Studies Press, Baldock, Hertfordshire, England, 2001.
[12] M. Grabisch, J.-L. Marichal, R. Mesiar, E. Pap, Aggregation Functions, Cambridge University Press, 2009.
[13] P. Hájek, Metamathematics of Fuzzy Logic, Kluwer, Dordrecht, 1998.
[14] K.H. Kim, Boolean Matrix Theory and Applications, Marcel Dekker, Inc., New York, 1982.
[15] E.P. Klement, R. Mesiar, E. Pap, Triangular Norms, Kluwer, Dordrecht, 2000.
[16] H.H. Krantz, R.D. Luce, P. Suppes, A. Tversky, Foundations of Measurement. Vol. I (Additive and Polynomial Representations), Vol. II (Geometric, Threshold, and Probabilistic Representations), Vol. III (Representations, Axiomatization, and Invariance), Dover Edition, 2007.
[17] D. Lee, H. Seung, Learning the parts of objects by non-negative matrix factorization, Nature 401 (1999) 788–791.
[18] J.D. Leeuw, Principal component analysis of binary data. Application to roll-call analysis [Online]. Available at http://gifi.stat.ucla.edu, 2003.
[19] M.R. Mickey, P. Mundle, L. Engelman, Boolean factor analysis, in: W.J. Dixon (Ed.), BMDP Statistical Software Manual, vol. 2, University of California Press, Berkeley, CA, 1990, pp. 849–860.
[20] P. Miettinen, T. Mielikäinen, A. Gionis, G. Das, H. Mannila, The discrete basis problem, IEEE Trans. Knowl. Data Eng. 20 (10) (2008) 1348–1362.
[21] G.A. Miller, The magical number seven, plus or minus two: some limits on our capacity for processing information, Psychol. Rev. 63 (1956) 81–97.
[22] D.S. Nau, G. Markowsky, M.A. Woodbury, D.B. Amos, A mathematical analysis of human leukocyte antigen serology, Math. Biosci. 40 (1978) 243–270.
[23] S.T. Roweis, L.K. Saul, Nonlinear dimensionality reduction by locally linear embedding, Science 290 (2000) 2323–2326.
[24] Sajama, A. Orlitsky, Semi-parametric exponential family PCA, in: L.K. Saul, Y. Weiss, L. Bottou (Eds.), Advances in Neural Information Processing, NIPS, Cambridge, MA, 2005, pp. 1177–1184.
[25] A. Schein, L. Saul, L. Ungar, A generalized linear model for principal component analysis of binary data, in: Proc. Int. Workshop on Artificial Intelligence and Statistics, 2003, pp. 14–21.
[26] L.J. Stockmeyer, The set basis problem is NP-complete, IBM Research Report RC5431, Yorktown Heights, NY, 1975.
[27] F. Tang, H. Tao, Binary principal component analysis, in: Proc. British Machine Vision Conference, 2006, pp. 377–386, 2006.
[28] N. Tatti, T. Mielikäinen, A. Gionis, H. Mannila, What is the dimension of your binary data?, in: The 2006 IEEE Conference on Data Mining, ICDM 2006, IEEE Computer Society, 2006, pp. 603–612.
[29] J.B. Tenenbaum, V. de Silva, J.C. Langford, A global geometric framework for nonlinear dimensionality reduction, Science 290 (2000) 2319–2323.
[30] M. Ward, R.P. Dilworth, Residuated lattices, Trans. Am. Math. Soc. 45 (1939) 335–354.
[31] Z. Zivkovic, J. Verbeek, Transformation invariant component analysis for binary images, in: 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, CVPR'06, vol. 1, 2006, pp. 254–259.