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

Optimal factorization of three-way binary data using triadic concepts

Radim Belohlavek · Cynthia Glodeanu · Vilem Vychodil

Received: date / Accepted: date

Abstract We present a new approach to factor analysis of three-way binary data, i.e. data described by a 3-dimensional binary matrix I, describing a relationship between objects, attributes, and conditions. The problem consists in finding a decomposition of I into three binary matrices, an object-factor matrix A, an attribute-factor matrix B, and a condition-factor matrix C, with the number of factors as small as possible. The scenario is similar to that of decomposition-based methods of analysis of three-way data but the difference consists in the composition operator and the constraint on A, B, and C to be binary. We show that triadic concepts of I, developed within formal concept analysis, provide us with optimal decompositions. We present an example demonstrating the usefulness of the decompositions. Since finding optimal decompositions is NP-hard, we propose a greedy algorithm for computing suboptimal decompositions and evaluate its performance.

Keywords Three-way binary data · Factorization · Triadic concept analysis · 3rd order tensor

Radim Belohlavek and Vilem Vychodil
Department of Computer Science, Palacky University, Olomouc
17. listopadu 12, CZ-771 46 Olomouc, Czech Republic
E-mail: {radim.belohlavek, vychodil}@acm.org

Cynthia Glodeanu
Institute of Algebra, Technical University, Dresden
TU Dresden, 01062 Dresden
E-mail: Cynthia_Vera.Glodeanu@mailbox.tu-dresden.de

1 Problem Description

Many methods of analysis of two-way data, i.e. data described by matrices, are based on various types of matrix decomposition. These include methods for binary data, see e.g. [1,4,12,15,19,20]. Recently, there has been a growing interest in three-way and generally N-way data, i.e. data represented by N-dimensional matrices. [7] provides an up-to-date survey with 244 references, see also [2,9,17]. Interestingly, decompositions of N-dimensional matrices go back as far as to the 1920s and have been studied in psychometrics since the 1940s (see [7] for historical account).

In this paper, we are concerned with decompositions of three-way binary data. Such data is represented by a 3-dimensional matrix which is denoted by I in this paper and whose entries, denoted I_{ijt}, are either 0 or 1. For illustrative purposes, the matrix entries are interpreted as follows:

$$I_{ijt} = \begin{cases} 1 & \text{if object } i \text{ has attribute } j \text{ under condition } t, \\ 0 & \text{if object } i \text{ does not have attribute } j \text{ under } t. \end{cases}$$

Such data tables have different names in different fields, e.g. three-way data in psychology, 3rd order tensor in geometry, or triadic context in formal concept analysis. We use the term 3-dimensional matrix.

It is well-known (see e.g. [14,20]) that applying decomposition methods to binary data that were designed for real-valued data distorts the meaning of the data and yields results that are difficult to interpret. On the other hand, decomposition methods based on the Boolean matrix product are interpreted in a straightforward way and are therefore preferable [14]. The decomposition involving the Boolean matrix product admits a natural generalization for the case of three-way binary data. Namely, one may look for a decomposition of a given $n \times m \times p$ matrix I with entries $I_{ijt} \in \{0,1\}$ into a product

$$I = \circ(A, B, C) \tag{1}$$

of an $n \times k$ object-factor matrix A with entries $A_{ik} \in \{0,1\}$, an $m \times k$ attribute-factor matrix B with entries $B_{jk} \in \{0,1\}$, and a $p \times k$ condition-factor matrix C with entries $C_{tk} \in \{0,1\}$, with $\circ$ defined by

$$\circ(A, B, C)_{ijt} = \max_{l=1}^{k} A_{il} \cdot B_{jl} \cdot C_{tl}. \tag{2}$$

It is easily seen that if $p = 1$, $\circ(A, B, C)$ may be identified with the Boolean matrix product of A and the transpose of B (put $C_{1,l} = 1$ for each l). Decomposition (2) has the following meaning: The object i has the attribute j under the condition t if and only if there exists a factor l such that l applies to i ($A_{il} = 1$), j is a particular manifestation of l ($B_{jl} = 1$), and t is one of the conditions under which l appears ($C_{tl} = 1$).

In this paper, we are interested in optimal decompositions of I, i.e. decompositions (2) with the number k of factors as small as possible. We call the smallest such k the *Schein rank* of I and denote it by $\rho(I)$. We show that triadic concepts of I [11,21] play a crucial role for optimal decompositions. Namely, they may be used as factors and yield optimal decompositions of I.

The decomposition problem is equivalent to finding a set cover of I. Therefore, the problem of computing optimal decompositions is NP-hard and propose an efficient greedy algorithm for computing suboptimal decompositions. We include an illustrative example and an experimental evaluation of the algorithm.

Remark 1 With a slightly different interpretation behind, the decomposition (2) may alternatively be described by:

$$\circ(A,B,C)_{ijt} = \max_{l=1}^{k}(A*B)_{ijl}\cdot C_{tl}$$
$$\text{with } (A*B)_{ijl} = A_{il}\cdot B_{jl}.$$

That is, $A*B$ is an $n\times m\times k$ binary matrix whose layer l ($l=1,\dots,k$) results as the Boolean product of the l-th column $A_{\_l}$ of A and the l-th column $B_{\_l}$ of B. A layer l of $A*B$ describes the relationship between the objects and attributes of the l-th factor. Multiplying $A*B$ with matrix C that relates factors to conditions then results in matrix I describing the relationship between objects, attributes, and conditions. Another natural interpretation, in which we first consider the relationship between attributes and conditions for every factor, yields the following formula:

$$\circ(A,B,C)_{ijt} = \max_{l=1}^{k} A_{il}\cdot (B*C)_{jtl}$$
$$\text{with } (B*C)_{jtl} = B_{jl}\cdot C_{tl}.$$

2 Factorization Using Triadic Concepts

2.1 Triadic Concepts

This section provides the notions needed in what follows (see [11,21] for more information). A *triadic context* is a quadruple $\langle X_1,X_2,X_3,I\rangle$ where I is a ternary relation between X_1 (set of objects), X_2 (set of attributes), and X_3 (set of conditions), with $\langle x_1,x_2,x_3\rangle\in I$ being interpreted as "object x_1 has attribute x_2 under condition x_3". Let $\{i,j,k\}=\{1,2,3\}$. If $\langle x_1,x_2,x_3\rangle\in I$ we say that x_i, x_j, and x_k are related. For $C_k\subseteq X_k$, we define a dyadic context $\langle X_i,X_j,I^{ij}_{C_k}\rangle$ by $\langle x_i,x_j\rangle\in I^{ij}_{C_k}$ iff for each $x_k\in C_k$: x_i,x_j,x_k are related. The concept-forming operators induced by $\langle X_i,X_j,I^{ij}_{C_k}\rangle$ are denoted by ${}^{(i,j,C_k)}$. Therefore, for $C_i\subseteq X_i$ and $C_j\subseteq X_j$ we put

$$C_i^{(i,j,C_k)} = \{x_j\in X_j \mid \text{for each } (x_i,x_k)\in C_i\times C_k : x_i,x_j,x_k \text{ are related}\},$$
$$C_j^{(i,j,C_k)} = \{x_i\in X_i \mid \text{for each } (x_j,x_k)\in C_j\times C_k : x_i,x_j,x_k \text{ are related}\}.$$

A *triadic concept* of $\langle X_1,X_2,X_3,I\rangle$ is a triplet $\langle D_1,D_2,D_3\rangle$ of $D_1\subseteq X_1$, $D_2\subseteq X_2$, and $D_3\subseteq X_3$, such that $D_1=D_2^{(1,2,D_3)}$, $D_2=D_3^{(2,3,D_1)}$, and $D_3=D_1^{(3,1,D_2)}$. D_1, D_2, and D_3 are called the *extent*, *intent*, and *modus* of $\langle D_1,D_2,D_3\rangle$. In particular, for any $\{i,j,k\}=\{1,2,3\}$, $C_i\subseteq X_i$, and $C_k\subseteq X_k$, putting $D_j=C_i^{(i,j,C_k)}$, $D_i=D_j^{(i,j,C_k)}$, and $D_k=D_i^{(i,k,D_j)}$, we get a triadic concept $\mathfrak{b}_{ik}(C_i,C_k)=\langle D_1,D_2,D_3\rangle$, called the ik-join of C_i and C_k. $\mathfrak{b}_{ik}(C_i,C_k)$

has the smallest k-th component under all triadic concepts $\langle E_1, E_2, E_3\rangle$ with the largest j-th component satisfying $C_i \subseteq E_i$ and $C_k \subseteq E_k$. We denote $\mathfrak{b}_{ik}(\{x_i\}, \{x_k\})$ just by $\mathfrak{b}_{ik}(x_i, x_k)$. We say that a triadic concept $\langle D_1, D_2, D_3\rangle$ covers $\langle x_1, x_2, x_3\rangle$ if $x_1 \in X_1$, $x_2 \in X_2$, and $x_3 \in X_3$. The set of all triadic concepts of $\langle X_1, X_2, X_3, I\rangle$ is denoted by $\mathcal{T}(X_1, X_2, X_3, I)$ and forms a trilattice, called the *concept trilattice* of $\langle X_1, X_2, X_3, I\rangle$.

In what follows, we assume that $X_1 = \{1, \ldots, n\}$, $X_2 = \{1, \ldots, m\}$ and $X_3 = \{1, \ldots, p\}$, and, for convenience, identify ternary relations between X_1, X_2 and X_3 with $n \times m \times p$ binary matrices.

2.2 Triadic Concepts as Factors

In this section we show how triadic concepts of I may be used as factors for decomposition (2). Call a 3-dimensional binary matrix J a *cuboidal matrix* (shortly, a *cuboid*) if there exist an $n \times 1$ binary vector A, an $m \times 1$ binary vector B, and a $p \times 1$ binary vector C, such that $J = \circ(A, B, C)$. The following lemma explains the role of cuboids for decompositions (2).

Lemma 1 *$I = \circ(A, B, C)$ for an $n \times k$ matrix A, $m \times k$ matrix B, and $p \times k$ matrix C, iff I is a* max*-superposition of k cuboids $J_1, \ldots, J_k$, i.e.*

$$I = J_1 \max \cdots \max J_k.$$

In addition, for each $l = 1, \ldots, k$, $J_l = \circ(A_{\_l}, B_{\_l}, C_{\_l})$, i.e. each J_l is the product of the l-th columns of A, B, and C.

Proof Follows directly from definitions by easy calculation. □

As a result, to decompose I using a small number of factors, one needs to find a small number of cuboids in I which are full of 1s and cover all the entries of I with 1s. We say that a cuboid J is contained in I if $J_{ijt} \leq I_{ijt}$ for all i, j, t. As the following lemma shows, triadic concepts of I correspond to maximal cuboids contained in I.

Lemma 2 *$\langle D_1, D_2, D_3\rangle$ is a triadic concept of I iff $J = \circ(c(D_1), c(D_2), c(D_3))$ is a maximal cuboid contained in I. Here, $c(D_i)$ denotes the characteristic vector of D_i, i.e. $c(D_i)(x) = 1$ iff $x \in D_i$.*

Proof Follows from [21, Proposition 1] by a moment's reflection. □

We are going to use triadic concepts of I for decompositions of I in the following way. For a set

$$\mathcal{F} = \{\langle D_{11}, D_{12}, D_{13}\rangle, \ldots, \langle D_{k1}, D_{k2}, D_{k3}\rangle\}$$

of triadic concepts of I, we denote by $A_{\mathcal{F}}$ the $n \times k$ matrix in which the l-th column consists of the characteristic vector $c(D_{l1})$ of the extent D_{l1}, by $B_{\mathcal{F}}$ the $m \times k$ matrix in which the l-th column consists of the characteristic vector $c(D_{l2})$ of the intent D_{l2}, $C_{\mathcal{F}}$ the $p \times k$ matrix in which the l-th column consists of the characteristic vector $c(D_{l3})$ of the modus D_{l3} of $\langle D_{l1}, D_{l2}, D_{l3}\rangle$. That is,

$$(A_{\mathcal{F}})_{il} = \begin{cases} 1 & \text{if } i \in D_{l1}, \\ 0 & \text{if } i \notin D_{l1}, \end{cases} (B_{\mathcal{F}})_{jl} = \begin{cases} 1 & \text{if } j \in D_{l2}, \\ 0 & \text{if } j \notin D_{l2}, \end{cases} (C_{\mathcal{F}})_{tl} = \begin{cases} 1 \text{ if } t \in D_{l3}, \\ 0 \text{ if } t \notin D_{l3}. \end{cases}$$

If $I = \circ(A_{\mathcal{F}}, B_{\mathcal{F}}, C_{\mathcal{F}})$, $\mathcal{F}$ can be seen as a set of factors which fully explain the data. In such a case, we call the triadic concepts from $\mathcal{F}$ *factor concepts*. Given I, our aim is to find a small set $\mathcal{F}$ of factor concepts.

Using triadic concepts of I as factors is intuitively appealing because triadic concepts are simple models of human concepts according to traditional logic approach [11]. In fact, factors are often called "(hidden) concepts" in the ordinary factor analysis. In addition, the extents, intents, and modi of the concepts, i.e. columns of $A_{\mathcal{F}}$, $B_{\mathcal{F}}$, and $C_{\mathcal{F}}$, have a straightforward interpretation: they represent the objects, attributes, and conditions to which the factor concept applies (see Section 3 for particular examples).

The next theorem shows that triadic concepts are universal and optimal factors in that every 3-dimensional matrix can be factorized using triadic concepts and the factorizations that employ triadic concepts as factors are optimal.

Theorem 1 (optimality) *Let I be an $n \times m \times p$ binary matrix.*

(1) $\rho(I) \leq \min(nm, np, mp)$.

(2) *There exists $\mathcal{F} \subseteq \mathcal{T}(X_1, X_2, X_3, I)$ with $|\mathcal{F}| = \rho(I)$ for which*

$$I = \circ(A_{\mathcal{F}}, B_{\mathcal{F}}, C_{\mathcal{F}}).$$

Proof (1) Let $\mathcal{F} = \{\mathfrak{b}_{12}(x_i, x_j) \,|\, x_i \in X_i,\, x_j \in X_j\}$. Clearly, $|\mathcal{F}| \leq |X_i| \cdot |X_j| = nm$. Due to Lemma 2, the cuboids corresponding to $\mathfrak{b}_{12}(x_i, x_j) \in \mathcal{F}$ are contained in I, hence $\circ(A_{\mathcal{F}}, B_{\mathcal{F}}, C_{\mathcal{F}}) \leq I$ due to Lemma 1. On the other hand, if $I_{x_i x_j x_t} = 1$ then $\langle x_i, x_j, x_t\rangle \in \mathfrak{b}_{12}(x_i, x_j)$, i.e. $J_{x_i x_j x_t} = 1$ for the cuboid corresponding to $\mathfrak{b}_{12}(x_i, x_j)$, from which it follows that $\circ(A_{\mathcal{F}}, B_{\mathcal{F}}, C_{\mathcal{F}})_{x_i x_j x_t} = 1$. We proved $\rho(I) \leq mn$. In a similar way, one proves $\rho(I) \leq np$, and $\rho(I) \leq mp$, proving (1).

(2) We have to show that if $I = \circ(A, B, C)$ for $n \times k$, $m \times k$, and $p \times k$ binary matrices A, B, and C, there exists a set $\mathcal{F} \subseteq \mathcal{T}(X_1, X_2, X_3, I)$ with $|\mathcal{F}| \leq k$ such that for $A_{\mathcal{F}}$, $B_{\mathcal{F}}$, and $C_{\mathcal{F}}$ we have $I = \circ(A_{\mathcal{F}}, B_{\mathcal{F}}, C_{\mathcal{F}})$. For every $l = 1, \ldots, k$, consider the cuboid $J_l = \circ(A_{\_l}, B_{\_l}, C_{\_l})$. Since J_l is contained in I, there exists a maximal cuboid J'_l that is contained in I and contains J_l. Due to Lemma 2, the triplet $\langle D^l_1, D^l_2, D^l_3\rangle$ corresponding to J'_l is a triadic concept of I. Put $\mathcal{F} = \{\langle D^l_1, D^l_2, D^l_3\rangle \,|\, l = 1, \ldots, k\}$. Then the columns $(A_{\mathcal{F}})_{\_l}$, $(B_{\mathcal{F}})_{\_l}$, and $(C_{\mathcal{F}})_{\_l}$ are the characteristic vectors of D^l_1 , D^l_2 and D^l_3, and therefore $(A_{\mathcal{F}})_{\_l} \leq A_{\_l}$, $(B_{\mathcal{F}})_{\_l} \leq B_{\_l}$, and $(C_{\mathcal{F}})_{\_l} \leq C_{\_l}$. Therefore,

$$I_{ijt} = \circ(A, B, C)_{ijt} = \textstyle\bigvee_{l=1}^{k} \circ(A_{\_l}, B_{\_l}, C_{\_l})_{ijt} \leq$$
$$\leq \textstyle\bigvee_{l=1}^{k} \circ((A_{\mathcal{F}})_{\_l}, (B_{\mathcal{F}})_{\_l}, (C_{\mathcal{F}})_{\_l})_{ijt} \leq I_{ijt}.$$

Since $|\mathcal{F}| \leq k$, the proof of (2) is finished. □

Theorem 1 means that when looking for factors for decompositions of I, one may restrict the search to triadic concepts. Theorem 1 is the basis for the decomposition algorithm presented in Section 4.

We now describe mandatory factors of I, i.e. triadic concepts that have to be present in every factorization of I. In the dyadic case, mandatory factors of a dyadic relation I are exactly the dyadic concepts of I that are both object concepts and attribute concepts [1]. In the triadic case, the concepts of $\langle X_1, X_2, X_3, I\rangle$ that can be seen as analogous to object and attribute concepts are $\mathfrak{b}_{ij}(a_i, a_j)$ for $\{i, j\} = \{1, 2\}$, $\mathfrak{b}_{jk}(b_j, b_k)$ for $\{j, k\} = \{2, 3\}$, and $\mathfrak{b}_{ik}(c_i, c_k)$ for $\{i, k\} = \{1, 3\}$ where $a_1, c_1 \in X_1$, $a_2, b_2 \in X_2$, $b_3, c_3 \in X_3$.

Lemma 3 *Let $\langle X_1, X_2, X_3, I\rangle$ be a triadic context, $a_1, c_1 \in X_1$, $a_2, b_2 \in X_2$, and $b_3, c_3 \in X_3$. If there exist $\{i_a, j_a\} = \{1, 2\}$, $\{i_b, j_b\} = \{2, 3\}$, and $\{i_c, j_c\} = \{1, 3\}$ for which*

$$\mathfrak{b}_{i_a j_a}(a_{i_a}, a_{j_a}) = \mathfrak{b}_{i_b j_b}(b_{i_b}, b_{j_b}) = \mathfrak{b}_{i_c j_c}(c_{i_c}, c_{j_c}),$$

then the concept $\langle D_1, D_2, D_3\rangle$ described equivalently by any of the formulas $\mathfrak{b}_{i_x j_x}(x_{i_x}, x_{j_x})$ is the only triadic concept for which

$$\{a_1, c_1\} \times \{a_2, b_2\} \times \{b_3, c_3\} \subseteq D_1 \times D_2 \times D_3. \tag{3}$$

Proof (3) follows from the fact that the p-th and q-th component of $\mathfrak{b}_{pq}(x_p, x_q)$ contains x_p and x_q, respectively. Let $\langle C_1, C_2, C_3\rangle \in \mathcal{T}(X_1, X_2, X_3, I)$ satisfy $\{a_1, c_1\} \times \{a_2, b_2\} \times \{b_3, c_3\} \subseteq C_1 \times C_2 \times C_3$. Then $\{a_{i_a}\} \subseteq C_{i_a}$ and $\{a_{j_a}\} \subseteq C_{j_a}$. Since D_3 is the third component of $\mathfrak{b}_{i_a j_a}(a_{i_a}, a_{j_a})$, the properties of the concept-derivation operators yield

$$D_3 = \{a_{i_a}\}^{(i_a, 3, \{a_{j_a}\})} \supseteq C_{i_a}^{(i_a, 3, C_{j_a})} = C_3.$$

In a similar way, one proves $D_1 \supseteq C_1$ and $D_2 \supseteq C_2$. Since $\langle C_1, C_2, C_3\rangle$ is a triadic concept, [21, Proposition 1] implies $\langle C_1, C_2, C_3\rangle = \langle D_1, D_2, D_3\rangle$. □

A triadic concept $\langle D_1, D_2, D_3\rangle \in \mathcal{T}(X_1, X_2, X_3, I)$ is called *mandatory* if $\langle D_1, D_2, D_3\rangle \in \mathcal{F}$ for every $\mathcal{F} \subseteq \mathcal{T}(X_1, X_2, X_3, I)$ for which $I = \circ(A_{\mathcal{F}}, B_{\mathcal{F}}, C_{\mathcal{F}})$. The following lemma is essential in describing mandatory concepts.

Lemma 4 *For a concept $\mathfrak{d} \in \mathcal{T}(X_1, X_2, X_3, I)$ and $x_1 \in X_1$, $x_2 \in X_2$ and $x_3 \in X_3$, the following conditions are equivalent:*

(i) *$\mathfrak{d}$ is the only concept of I that covers $\langle x_1, x_2, x_3\rangle$.*

(ii) *There exists $\{i, j, k\} = \{1, 2, 3\}$ such that*

$$\mathfrak{d} = \mathfrak{b}_{ij}(x_i, x_j) = \mathfrak{b}_{jk}(x_j, x_k) = \mathfrak{b}_{ik}(x_i, x_k). \tag{4}$$

(iii) *For every $\{i, j, k\} = \{1, 2, 3\}$ we have*

$$\mathfrak{d} = \mathfrak{b}_{ij}(x_i, x_j) = \mathfrak{b}_{jk}(x_j, x_k) = \mathfrak{b}_{ik}(x_i, x_k).$$

Proof "(i)⇒(ii)": If $\mathfrak{d}$ covers $\langle x_1, x_2, x_3\rangle$, then x_1, x_2, and x_3 are related. Therefore, $\mathfrak{b}_{ij}(x_i, x_j)$, $\mathfrak{b}_{jk}(x_j, x_k)$, and $\mathfrak{b}_{ik}(x_i, x_k)$ all cover $\langle x_1, x_2, x_3\rangle$. Since $\mathfrak{d}$ is the only concept covering $\langle x_1, x_2, x_3\rangle$, (4) follows.

"(ii)⇒(i)": Follows from Lemma 3.

"(ii)⇒(iii)": Let (4) for some $\{i, j, k\} = \{1, 2, 3\}$. $\mathfrak{b}_{ji}(x_j, x_i)$ has the same k-th component as $\mathfrak{b}_{ij}(x_i, x_j)$ and therefore contains x_k. Since the i-th and j-th components of $\mathfrak{b}_{ji}(x_j, x_i)$ contain x_i and x_j, $\mathfrak{b}_{ji}(x_j, x_i)$ covers $\langle x_1, x_2, x_3\rangle$. Because (ii) implies (i), $\mathfrak{d}$ is the only concept that covers $\langle x_1, x_2, x_3\rangle$, whence $\mathfrak{d} = \mathfrak{b}_{ji}(x_j, x_i)$. The same way one proves that $\mathfrak{d} = \mathfrak{b}_{kj}(x_k, x_j)$ and $\mathfrak{d} = \mathfrak{b}_{ki}(x_k, x_i)$, proving (iii).

"(iii)⇒(ii)": Trivial. □

Theorem 2 (mandatory factors) *A concept $\mathfrak{d} \in \mathcal{T}(X_1, X_2, X_3, I)$ is mandatory if and only if there exist $x_1 \in X_1$, $x_2 \in X_2$, and $x_3 \in X_3$, that satisfy* (ii)*, or, equivalently* (iii)*, of Lemma 4.*

Proof Clearly, $\mathfrak{d}$ is mandatory if and only if there exist $x_1 \in X_1$, $x_2 \in X_2$, and $x_3 \in X_3$ such that $\mathfrak{d}$ is the only concept that covers x_1, x_2, and x_3. The claim thus follows from Lemma 4. □

Remark 2 A claim such as (iii) in Lemma 4 does not hold in Lemma 3. Namely, consider $I = \{a_1, c_1\} \times \{a_2, b_2\} \times \{b_3, c_3\} \cup \{a_1\} \times \{x_2\} \times \{b_3, c_3\}$ with x_2 distinct from both a_2 and b_2. $\mathfrak{d} = \langle\{a_1, c_1\}, \{a_2, b_2\}, \{b_3, c_3\}\rangle$ is the only concept covering $\{a_1, c_1\} \times \{a_2, b_2\} \times \{b_3, c_3\}$. Clearly, $\mathfrak{d} = \mathfrak{b}_{12}(a_1, a_2) = \mathfrak{b}_{23}(b_2, b_3) = \mathfrak{b}_{13}(c_1, c_3)$. However, $\mathfrak{b}_{21}(a_2, a_1) = \langle\{a_1\}, \{a_1, b_2, x_2\}, \{b_3, c_3\}\rangle \neq \mathfrak{d}$.

2.3 Transformation Between the Attributes and Conditions Space and Factors Space

Given an $n \times m \times p$ binary matrix I and its decomposition $I = \circ(A, B, C)$ where $A = A_{\mathcal{F}}, B = B_{\mathcal{F}}$, and $C = C_{\mathcal{F}}$ for some k-element set $\mathcal{F}$ of factor concepts, one is naturally interested in how one can transform a description of a given object in terms of attributes and conditions to a description of the same object in terms of factors. That is, one asks for transformations between the attribute×condition space $\{0, 1\}^{m\times p}$ and the factor space $\{0, 1\}^k$. For the dyadic case, such transformations are described in [1] and were utilized in [16] for improving classification of binary data.

In the attribute×condition space, the object $i \in X_1$ is represented by the i-th *(object) dyadic cut* of I, which is the $m \times p$ binary matrix $I_{i\_\_}$ corresponding to the dyadic context $\langle X_2, X_3, I^{23}_{\{i\}}\rangle$ (see Section 2.1). In the factor space, i is represented by the i-th row $A_{i\_}$ of A. Consider the transformations $g : \{0, 1\}^{m\times p} \to \{0, 1\}^k$ and $h : \{0, 1\}^k \to \{0, 1\}^{m\times p}$ defined for $P \in \{0, 1\}^{m\times p}$ and $Q \in \{0, 1\}^k$ by

$$(g(P))_l = \bigwedge_{j=1}^{m} \bigwedge_{t=1}^{p} (B_{jl} \cdot C_{tl} \to P_{jt}) \tag{5}$$

$$(h(Q))_{jt} = \bigvee_{l=1}^{k} (Q_l \cdot B_{jl} \cdot C_{tl}) \tag{6}$$

for $l \in \{1, \dots, k\}, j \in \{1, \dots, m\}$ and $t \in \{1, \dots, p\}$. Here, $\to$, $\cdot$, $\bigwedge$, and $\bigvee$ denote the truth function of classical implication, the usual product, minimum, and maximum, respectively. If P represents a description of an object i in terms of attributes and conditions and Q a description in terms of factors then given the interpretation of A, B, and C following (2), g and h have the following meaning: (5) says that a factor l applies to i if and only if object i has every attribute j under every condition t such that j is a manifestation of l and t is one of the conditions under which l appears; (6) says that object i has attribute j under condition t if there exists a factor l such that l applies to i, j is a manifestation of l, and t is one of the conditions under which l appears. The next theorem shows that g and h can be considered as appropriate transformations between the attribute×condition and factor spaces.

Theorem 3 *For $i \in \{1, \dots n\}$:*

$$g(I_{i\_\_}) = A_{i\_} \text{ and } h(A_{i\_}) = I_{i\_\_}.$$

That is, g maps the object dyadic cuts of I to the rows of A and vice versa, h maps the rows of A to the object dyadic cuts of I.

Proof $h(A_{i\_}) = I_{i\_\_}$ follows directly from $I = \circ(A, B, C)$. Since $A = A_{\mathcal{F}}, B = B_{\mathcal{F}}$ and $C = C_{\mathcal{F}}$ the l-th columns of A, B and C coincide with the characteristic vectors $c(D_{l1}), c(D_{l2})$, and $c(D_{l3})$ of the extent D_{l1}, intent D_{l2}, and modus D_{l3} of a triadic concept $\langle D_{l1}, D_{l2}, D_{l3}\rangle \in \mathcal{F}$, respectively.

$$\begin{aligned}(g(I_{i\_\_}))_l &= \textstyle\bigwedge_{j=1}^{m} \bigwedge_{t=1}^{p} (B_{jl} \cdot C_{tl} \to (I_{i\_\_})_{jt}) = \\ &= \textstyle\bigwedge_{j=1}^{m} \bigwedge_{t=1}^{p} ((c(D_{l2}))_j \cdot (c(D_{l3}))_t \to I_{ijt}) = \\ &= c(D_{l2}^{(1,2,D_{l3})})_i = c(D_{l1})_i = A_{il}.\end{aligned}$$

□

The following theorem, which can easily be checked from definitions, shows that g and h form an isotone Galois connection.

Theorem 4 *For $P, P' \in \{0,1\}^{m\times p}$ and $Q, Q' \in \{0,1\}^k$:*

$$P \le P' \Rightarrow g(P) \le g(P'), \tag{7}$$

$$Q \le Q' \Rightarrow h(Q) \le h(Q'), \tag{8}$$

$$h(g(P)) \le P, \tag{9}$$

$$Q \le g(h(Q)). \tag{10}$$

Conditions (7)–(10) represent natural requirements. Namely, (7) says that the more attributes under more conditions an object has, the more factors apply, while (8) asserts the converse relationship. This is in accordance with the factor model described in the paragraph following (2). Consequently, for an object, having "more attributes" and "more conditions" is positively correlated with having "more factors". (9) means that common attributes in common modi associated to all the factors which apply to a given object are contained in the collection of all attributes in the modi possessed by that object. The meaning of (10) is analogous.

A geometry behind the transformations is described by the following assertion. For $P \in \{0,1\}^{m\times p}$ and $Q \in \{0,1\}^k$, put

$$\begin{aligned}g^{-1}(Q) &= \{P \in \{0,1\}^{m\times p} \mid g(P) = Q\},\\ h^{-1}(P) &= \{Q \in \{0,1\}^k \mid h(Q) = P\}.\end{aligned}$$

Recall that $S \subseteq \{0,1\}^s$ is called *convex* if $V \in S$ whenever $U \leq V \leq W$ for some $U, W \in S$.

Theorem 5 (1) $g^{-1}(Q)$ *is a convex partially ordered subspace of the attribute and condition space and* $h(Q)$ *is the least element of* $g^{-1}(Q)$.
(2) $h^{-1}(P)$ *is a convex partially ordered subspace of the factor space and* $g(P)$ *is the largest element of* $h^{-1}(P)$.

Proof By standard application of the properties of isotone Galois connections. □

According to Theorem 5, the space $\{0,1\}^{m \times p}$ of attributes and conditions and the space $\{0,1\}^k$ of factors are partitioned into an equal number of convex subsets. The subsets of the space of attributes and conditions have least elements and the subsets of the space of factors have greatest elements. g maps every element of any convex subset of the space of attributes and conditions to the greatest element of the corresponding subset of the factor space, whereas h maps every element of some convex subset of the space of factors to the least element of the corresponding convex subset of the space of attributes and conditions.

3 Illustrative Example

In this section, we present an illustrative example of factorization. We consider input data containing information about students and their performance in various courses. Such data is frequently obtained from student evaluation and recommendation systems that are used during the process of admission to universities. Factor analysis of this type of data can help reveal important factors describing skills of students under various conditions.

Our model data is represented by a triadic context $\langle X_1, X_2, X_3, I \rangle$ where $X_1 = \{\mathsf{a}, \mathsf{b}, \ldots, \mathsf{h}\}$ is a set of students (objects); $X_2 = \{\mathsf{co}, \mathsf{cr}, \mathsf{di}, \mathsf{fo}, \mathsf{in}, \mathsf{mo}\}$ is a set of student qualities (attributes): communicative, creative, diligent, focused, independent, motivated; and $X_3 = \{\mathsf{AL}, \mathsf{CA}, \mathsf{CI}, \mathsf{DA}, \mathsf{NE}\}$ is a set of courses passed by the students (conditions): algorithms, calculus, circuits, databases, and networking. The fact that x_1 is related with x_2 under x_3 is interpreted so that "student x_1 showed quality x_2 in course x_3". We consider I given by the table in Fig. 1.

The rows of the table correspond to students, the columns correspond to attributes under the various conditions (courses). Then the triadic context

	AL						CA						CI						DA						NE					
	co	cr	di	fo	in	mo	co	cr	di	fo	in	mo	co	cr	di	fo	in	mo	co	cr	di	fo	in	mo	co	cr	di	fo	in	mo
a	1	1	1	1	1	1	0	0	1	1	0	1	1	1	0	0	1	1	0	0	1	1	0	1	1	1	0	0	1	1
b	1	1	0	0	1	1	0	0	0	0	0	0	1	1	0	0	1	1	1	1	0	0	0	0	1	1	0	0	1	1
c	1	1	1	1	0	1	0	0	1	1	0	1	1	1	0	0	0	0	1	1	1	1	0	1	1	1	0	0	0	0
d	1	1	1	1	1	1	0	0	1	1	0	1	1	1	0	0	1	1	0	0	1	1	0	1	1	1	0	0	1	1
e	1	1	0	0	1	1	0	0	0	0	0	0	1	1	0	0	1	1	1	1	0	0	0	0	1	1	0	0	1	1
f	1	1	1	1	1	1	0	0	1	1	0	1	1	1	0	0	1	1	1	1	1	1	0	1	1	1	0	0	1	1
g	1	1	0	0	1	1	0	0	0	0	0	0	1	1	0	0	1	1	0	0	0	0	0	0	1	1	0	0	1	1
h	0	0	1	1	0	1	0	0	1	1	0	1	0	0	0	0	0	0	0	0	1	1	0	1	0	0	0	0	0	0

Fig. 1 Triadic context

$\langle X_1, X_2, X_3, I\rangle$ contains 14 triadic concepts $D_1, \ldots, D_{14}$:

$$
\begin{aligned}
D_1 &= \langle \emptyset, \{\mathsf{co}, \mathsf{cr}, \mathsf{di}, \mathsf{fo}, \mathsf{in}, \mathsf{mo}\}, \{\mathsf{AL}, \mathsf{CA}, \mathsf{CI}, \mathsf{DA}, \mathsf{NE}\}\rangle,\\
D_2 &= \langle \{\mathsf{f}\}, \{\mathsf{co}, \mathsf{cr}, \mathsf{mo}\}, \{\mathsf{AL}, \mathsf{CI}, \mathsf{DA}, \mathsf{NE}\}\rangle,\\
D_3 &= \langle \{\mathsf{c}, \mathsf{f}\}, \{\mathsf{co}, \mathsf{cr}, \mathsf{di}, \mathsf{fo}, \mathsf{mo}\}, \{\mathsf{AL}, \mathsf{DA}\}\rangle,\\
D_4 &= \langle \{\mathsf{b}, \mathsf{c}, \mathsf{e}, \mathsf{f}\}, \{\mathsf{co}, \mathsf{cr}\}, \{\mathsf{AL}, \mathsf{CI}, \mathsf{DA}, \mathsf{NE}\}\rangle,\\
D_5 &= \langle \{\mathsf{a}, \mathsf{d}, \mathsf{f}\}, \{\mathsf{mo}\}, \{\mathsf{AL}, \mathsf{CA}, \mathsf{CI}, \mathsf{DA}, \mathsf{NE}\}\rangle,\\
D_6 &= \langle \{\mathsf{a}, \mathsf{d}, \mathsf{f}\}, \{\mathsf{co}, \mathsf{cr}, \mathsf{di}, \mathsf{fo}, \mathsf{in}, \mathsf{mo}\}, \{\mathsf{AL}\}\rangle,\\
D_7 &= \langle \{\mathsf{a}, \mathsf{c}, \mathsf{d}, \mathsf{f}\}, \{\mathsf{co}, \mathsf{cr}, \mathsf{di}, \mathsf{fo}, \mathsf{mo}\}, \{\mathsf{AL}\}\rangle,\\
D_8 &= \langle \{\mathsf{a}, \mathsf{c}, \mathsf{d}, \mathsf{f}, \mathsf{h}\}, \{\mathsf{di}, \mathsf{fo}, \mathsf{mo}\}, \{\mathsf{AL}, \mathsf{CA}, \mathsf{DA}\}\rangle,\\
D_9 &= \langle \{\mathsf{a}, \mathsf{b}, \mathsf{d}, \mathsf{e}, \mathsf{f}, \mathsf{g}\}, \{\mathsf{co}, \mathsf{cr}, \mathsf{in}, \mathsf{mo}\}, \{\mathsf{AL}, \mathsf{CI}, \mathsf{NE}\}\rangle,\\
D_{10} &= \langle \{\mathsf{a}, \mathsf{b}, \mathsf{c}, \mathsf{d}, \mathsf{e}, \mathsf{f}, \mathsf{g}\}, \{\mathsf{co}, \mathsf{cr}\}, \{\mathsf{AL}, \mathsf{CI}, \mathsf{NE}\}\rangle,\\
D_{11} &= \langle \{\mathsf{a}, \mathsf{b}, \mathsf{c}, \mathsf{d}, \mathsf{e}, \mathsf{f}, \mathsf{g}\}, \{\mathsf{co}, \mathsf{cr}, \mathsf{mo}\}, \{\mathsf{AL}\}\rangle,\\
D_{12} &= \langle \{\mathsf{a}, \mathsf{b}, \mathsf{c}, \mathsf{d}, \mathsf{e}, \mathsf{f}, \mathsf{g}, \mathsf{h}\}, \emptyset, \{\mathsf{AL}, \mathsf{CA}, \mathsf{CI}, \mathsf{DA}, \mathsf{NE}\}\rangle,\\
D_{13} &= \langle \{\mathsf{a}, \mathsf{b}, \mathsf{c}, \mathsf{d}, \mathsf{e}, \mathsf{f}, \mathsf{g}, \mathsf{h}\}, \{\mathsf{mo}\}, \{\mathsf{AL}\}\rangle,\\
D_{14} &= \langle \{\mathsf{a}, \mathsf{b}, \mathsf{c}, \mathsf{d}, \mathsf{e}, \mathsf{f}, \mathsf{g}, \mathsf{h}\}, \{\mathsf{co}, \mathsf{cr}, \mathsf{di}, \mathsf{fo}, \mathsf{in}, \mathsf{mo}\}, \emptyset\rangle.
\end{aligned}
$$

Interestingly, there exists a three-element set $\mathcal{F}$ of factor concepts, consisting of $F_1 = D_8$, $F_2 = D_4$, and $F_3 = D_9$. Using $\mathcal{F} = \{F_1, F_2, F_3\}$, we obtain the following 8×3 object-factor matrix $A_\mathcal{F}$, 6×3 attribute-factor matrix $B_\mathcal{F}$, and 5×3 condition-factor matrix $C_\mathcal{F}$:

$$
A_\mathcal{F} = \begin{pmatrix} 1\,0\,1\\ 0\,1\,1\\ 1\,1\,0\\ 1\,0\,1\\ 0\,1\,1\\ 1\,1\,1\\ 0\,0\,1\\ 1\,0\,0 \end{pmatrix}, \qquad
B_\mathcal{F} = \begin{pmatrix} 0\,1\,1\\ 0\,1\,1\\ 1\,0\,0\\ 1\,0\,0\\ 0\,0\,1\\ 1\,0\,1 \end{pmatrix}, \qquad
C_\mathcal{F} = \begin{pmatrix} 1\,1\,1\\ 1\,0\,0\\ 0\,1\,1\\ 1\,1\,0\\ 0\,1\,1 \end{pmatrix},
$$

where the columns correspond to (characteristic vectors of) factors F_1, F_2, and F_3. One can check that $I = \circ(A_\mathcal{F}, B_\mathcal{F}, C_\mathcal{F})$, i.e., I decomposes into three (two-dimensional) matrices using three factors. Note that the meaning of the factors can be seen from the extents, intents, and modi of the factor concepts. For instance, F_1 applies to students $\mathsf{a}, \mathsf{c}, \mathsf{d}, \mathsf{f}, \mathsf{h}$ (encoded by the first column 10110101 of $A_\mathcal{F}$) who are diligent, focused, and motivated (encoded by the first column 001101 in $B_\mathcal{F}$) in algorithms, calculus, and databases (encoded by the first column 11010 in $C_\mathcal{F}$). This suggests that F_1 can be interpreted as "having good background in theory / formal methods". F_2 applies to students who are communicative and creative in algorithms, circuits, databases, and networking. This may indicate interests and skills in "practical subjects". Finally, F_3 can be interpreted as a factor close to "self-confidence" because it is manifested by being communicative, creative, independent, and motivated. As a result, by finding the factors set $\mathcal{F} = \{F_1, F_2, F_3\}$, we have explained the structure of the input matrix I using three factors which describe the abilities of student applicants in terms of their skills in various subjects.

Let us recall that the factor concepts $\mathcal{F} = \{F_1, F_2, F_3\}$ can be seen as maximal cuboids in I. Indeed, I itself can be depicted as a three-dimensional box where the axes correspond to students, their qualities, and courses. Figure 2

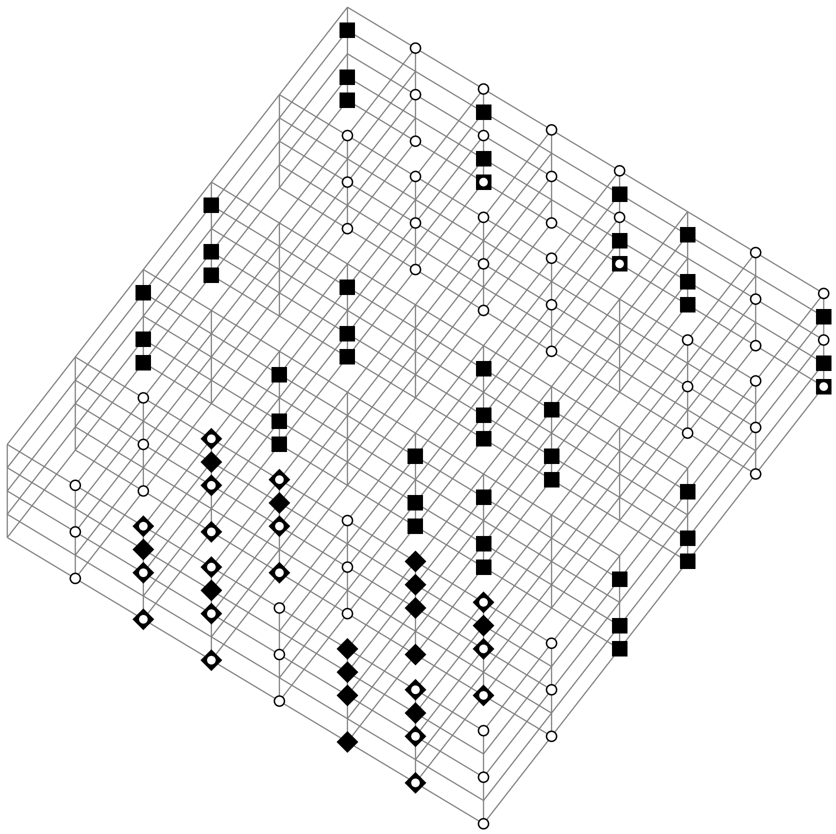

Fig. 2 Geometric meaning of factors.

shows factors F_1, F_2 and F_3 depicted as cuboids. Nodes in Figure 2 denoted by geometric shapes ■, ◆ and $\circ$ correspond to elements in I. Namely, a geometric shape is present on the intersection of $x_1 \in X_1$, $x_2 \in X_2$, and $x_3 \in X_3$ in the diagram iff $\langle x_1, x_2, x_3 \rangle \in I$. Furthermore, the shape is ■ iff $\langle x_1, x_2, x_3 \rangle \in I$ belongs to the factor F_1, i.e. iff x_1 belongs to the extent of F_1, x_2 belongs to the intent of F_1, and x_3 belongs to the modus of F_1. Analogously, ◆ and $\circ$

Algorithm 1: COMPUTEFACTORS(X_1, X_2, X_3, I)

1 **set** $\mathcal{S}$ **to** $\emptyset$;
2 **foreach** $\langle D_1, J\rangle \in \mathcal{B}(X_1, X_2 \times X_3, I^{X_1})$ **do**
3   **foreach** $\langle D_2, D_3\rangle \in \mathcal{B}(X_2, X_3, J)$ **do**
4   **if** $D_1 = (D_2 \times D_3)'$ **then**
5   **add** $\langle D_1, D_2, D_3\rangle$ **to** $\mathcal{S}$;
6   **end**
7   **end**
8 **end**
9 **set** $\mathcal{F}$ **to** $\emptyset$;
10 **set** U **to** I;
11 **while** $U \neq \emptyset$ **do**
12   **select** $\langle D_1, D_2, D_3\rangle \in \mathcal{S}$ **which maximizes** $|U \cap (D_1 \times D_2 \times D_2)|$;
13   **add** $\langle D_1, D_2, D_2\rangle$ **to** $\mathcal{F}$;
14   **set** U **to** $U \setminus (D_1 \times D_2 \times D_3)$;
15   **remove** $\langle D_1, D_2, D_3\rangle$ **from** $\mathcal{S}$;
16 **end**
17 **return** $\mathcal{F}$

indicate that $\langle x_1, x_2, x_3\rangle \in I$ belongs to the factors F_2 and F_3, respectively. Notice that some $\langle x_1, x_2, x_3\rangle \in I$ are covered by two factors (F_1 and F_3 or F_2 and F_3) which is graphically denoted by a composition of ■ and ○ or a composition of ◆ and ○).

4 Algorithms and Their Performance

4.1 Factorization algorithms

Since the problem of finding a minimal decomposition of $\langle X_1, X_2, X_3, I\rangle$ is reduced to a problem of finding a minimal subset $\mathcal{F} \subseteq \mathcal{T}(X_1, X_2, X_3, I)$ of formal concepts which cover I, we can reduce the problem of finding a matrix decomposition (1) to the set-covering problem. The universe U to be covered corresponds to $I \subseteq X_1 \times X_2 \times X_3$. The family $\mathcal{S}$ of subsets of the universe U that is used for finding a cover corresponds to the set of all triadic concepts $\mathcal{T}(X_1, X_2, X_3, I)$, i.e. $\mathcal{S} = \{D_1 \times D_2 \times D_3 \mid \langle D_1, D_2, D_3\rangle \in \mathcal{T}(X_1, X_2, X_3, I)\}$. In this setting, we are looking for $\mathcal{C} \subseteq \mathcal{S}$ as small as possible such that $\bigcup \mathcal{C} = U$. Thus, the decomposition problem is an instance of the set-covering problem. It is well known that the set covering optimization problem is NP-hard and the corresponding decision problem is NP-complete. However, there exists a greedy approximation algorithm for the set covering optimization problem which achieves an approximation ratio $\leq \ln(|U|) + 1$, see [3]. This gives us a "naive" greedy-approach algorithm as a starting point for computing all factor concepts.

Algorithm 1, implementing the above-mentioned greedy approach in our setting, computes a set of factor concepts by first computing the set of all triadic concepts which are stored in $\mathcal{S}$ (lines 1–8) and then iteratively selecting formal concepts from $\mathcal{S}$, maximizing their overlap with the remaining tuples

Algorithm 2: COMPUTEFACTORS(X_1, X_2, X_3, I)

```
set 𝓕 to ∅;
set U to I;
while U ≠ ∅ do
    set E1, E2, E3 to ∅;
    set removed to 0;
    repeat
        set ⟨D1, D2, D3⟩ to ⟨E1, E2, E3⟩;
        set removed to Size(U, D2, D3);
        select ⟨E2, E3⟩ ∈ Update(D2, D3) which maximizes Size(U, E2, E3);
        set ⟨E1, E2, E3⟩ to Factor(E2, E3);
    until removed ≥ Size(U, E2, E3) ;
    add ⟨D1, D2, D3⟩ to 𝓕;
    set U to U \ (D1 × D2 × D3);
end
return 𝓕
```

in U (lines 9–17). Notice that the triadic concepts are computed by a reduction to the dyadic case [6]. In line 2, we iterate over all dyadic concepts in $\mathcal{B}(X_1, X_2 \times X_3, I^{X_1})$ where $I^{X_1} = \{\langle x_1, \langle x_2, x_3\rangle\rangle \mid \langle x_1, x_2, x_3\rangle \in I\}$. In line 3, we iterate over all concepts in $\mathcal{B}(X_2, X_3, J)$ where J was obtained as an intent in the previous line. The condition in line 4 is needed to check whether D_1 is maximal (note that $'$ in line 4 denotes a concept forming operator induced by the dyadic formal context $\langle X_1, X_2 \times X_3, I^{X_1}\rangle$), i.e., whether $\langle D_1, D_2, D_3\rangle$ is a triadic concept. Notice that [6] computes triadic concepts by an analogous reduction which utilizes two nested NEXTCLOSURE algorithms [5]. Note that an arbitrary algorithm for computing dyadic formal concepts, such as the CbO or Lindig's algorithm, can do the job, see [10] for a survey and comparison of such algorithms.

Algorithm 1 can be significantly improved to get a better performance in terms of computation time. Namely, the main drawback of Algorithm 1 is that it first computes the set $\mathcal{S}$ of all triadic concepts of I and then selects (usually) a small subset of it being the set of factor concepts by iteratively going through $\mathcal{S}$. It is well-known that the number of elements in $\mathcal{S}$ is usually large and may be larger than exponential w.r.t. $\min(|X_1|, |X_2|, |X_3|)$ in the worst case. As a result, since Algorithm 1 iterates through $\mathcal{S}$ every time it computes a new factor, it has an exponential time delay complexity. Algorithm 2 overcomes this problem and computes a set of factor concepts directly without the need to compute all triadic concepts. The way Algorithm 2 works leads to a polynomial time delay complexity, as easily seen from its description below. The algorithm is based on the idea of incremental modification of a promising triadic concept by extending its intent and modus so that the concept covers as much of the remaining values in U as possible.

Algorithm 2 uses the following auxiliary sets. First, a set

$$\text{Update}(D_2, D_3) \subseteq 2^{X_2} \times 2^{X_3}$$

which represents a set of pairs of sets of attributes and conditions that result from $D_2 \subseteq X_2$ and $D_3 \subseteq X_3$ by adding a new attribute, a new condition, or both (line 9). Formally, Update(D_2, D_3) can be defined as a union of the following three subsets of $2^{X_2} \times 2^{X_3}$:

$$U_1 = \{D_2 \cup \{x_2\} \mid x_2 \in X_2 \setminus D_2\} \times \{D_3\},$$
$$U_2 = \{D_2\} \times \{D_3 \cup \{x_3\} \mid x_3 \in X_3 \setminus D_3\},$$
$$U_3 = \{D_2 \cup \{x_2\} \mid x_2 \in X_2 \setminus D_2\} \times \{D_3 \cup \{x_3\} \mid x_3 \in X_3 \setminus D_3\}.$$

Each element $\langle E_2, E_3 \rangle \in$ Update(D_2, D_3) is used as an enlarged candidate for replacing D_2 and D_3 if a factor concept that corresponds to $\langle E_2, E_3 \rangle$ covers a larger part of U than the current factor corresponding to $\langle D_2, D_3 \rangle$. In order to give a precise meaning of the covered part of U, we introduce a number

$$\mathrm{Size}(U, D_2, D_3) = \max\bigl(|U \cap \mathfrak{b}^{\times}_{23}(D_2, D_3)|, |U \cap \mathfrak{b}^{\times}_{32}(D_3, D_2)|\bigr),$$

where $\mathfrak{b}^{\times}_{ik}(C_i, C_k) = E_i \times E_j \times E_k$ whenever $\mathfrak{b}_{ik}(C_i, C_k) = \langle E_i, E_j, E_k \rangle$. In Algorithm 2 (lines 6–11), triadic concepts $\mathfrak{b}_{23}(D_2, D_3)$ and $\mathfrak{b}_{32}(D_3, D_2)$ are considered as candidates for a factor concept. Thus, Size(U, D_2, D_3) measures the maximum overlap of such concepts with U (we wish to maximize the overlap, see line 9). Finally, we introduce Factor(D_2, D_3) to denote one of the latter concepts having the greater overlap with U:

$$\mathrm{Factor}(D_2, D_3) = \begin{cases} \mathfrak{b}_{23}(D_2, D_3), & \text{if } \mathrm{Size}(U, D_2, D_3) = |U \cap \mathfrak{b}^{\times}_{23}(D_2, D_3)|, \\ \mathfrak{b}_{32}(D_3, D_2), & \text{otherwise.} \end{cases}$$

Using the notation introduced above, Algorithm 2 collects new concepts in set $\mathcal{F}$ until the whole U is covered (line 3). During each iteration of the while-loop (lines 3–14), it finds the largest concept in I covering the remaining U. In the repeat-until loop (lines 6–11), each concept is iteratively modified by adding new attributes and conditions (line 10) until its overlap with U starts decreasing (line 11). The concept with maximum overlap is then added to $\mathcal{F}$ (line 12). Clearly, Algorithm 2 is sound and produces a set of factor concepts.

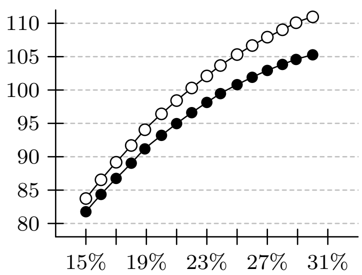

Fig. 3 Empirical comparison of the numbers of factors computed by Algorithm 1 and Algorithm 2 (x-axis: percentage of 1s in randomly generated three-dimensional matrices; y-axis: numbers of factors; black nodes: average numbers of factors computed by Algorithm 1; white nodes: average numbers of factors computed by Algorithm 2).

Remark 3 In practice, Algorithm 1 produces better results than Algorithm 2 in terms of the smaller number of computed factors. This is expected since Algorithm 1 uses the whole set of triadic concepts to search for the factors.

On the other hand, Algorithm 2 is faster than Algorithm 1 by an order of magnitude. This is because the expensive operation of computing all triadic concepts is omitted in Algorithm 2. Even if Algorithm 2 delivers worse results (on average), our empirical experiments have shown that the average difference of results obtained by both the algorithms is negligible if we factorize matrices with relatively low percentages of 1s.

The results of experiments related to this issue are depicted in Fig. 3. The graph in Fig. 3 shows two curves corresponding to average numbers of factors computed by Algorithm 1 (curve with black nodes) and Algorithm 2 (curve with white nodes) using a sample of $300,000$ randomly generated matrices of various sizes, each matrix containing about ten thousand entries. One can see that with growing density of 1s (i.e., growing percentage of 1s in matrices), the difference between the average numbers of factors grows and Algorithm 2 computes more factors than Algorithm 1. With smaller percentages of 1s, the difference is negligible. Since most large real-world data sets represented by three-dimensional matrices are typically sparse (very low percentages of 1s), Algorithm 2 is a preferred choice since it delivers almost as good results as Algorithm 1 in considerably less time.

4.2 Approximate factorization

Another important issue in factor analysis is *approximate factorizability,* i.e. the ability to find a set of factor concepts $\mathcal{F}$ such that $\circ(A_{\mathcal{F}}, B_{\mathcal{F}}, C_{\mathcal{F}})$ equals I at least to a specified degree of approximation. By a degree of approximation we mean a ratio (usually specified in percents) given by

$$\frac{|A_{\mathcal{F}} \times B_{\mathcal{F}} \times C_{\mathcal{F}}|}{|I|}. \tag{11}$$

As one can see, the ratio equals 1 (or 100%) iff $\mathcal{F}$ is a set of factor concepts, i.e. $I = \circ(A_{\mathcal{F}}, B_{\mathcal{F}}, C_{\mathcal{F}})$ (exact decomposition). From the point of view of approximation, one is interested in finding, given a ratio r, a set $\mathcal{F}$ of triadic concepts such that the degree of approximation given by (11) is at least r. In other words, we are interested in factors which explain at least $r\%$ of the input data. Notice that both Algorithm 1 and Algorithm 2 can be easily modified to compute approximate factorizations by adding an additional parameter r and a new halting condition (see line 11 in Algorithm 1 and line 3 in Algorithm 2) which stops looking for further factor concepts whenever the threshold value r has been reached.

Remark 4 We performed a series of experiments to explore the behavior of approximate decompositions, in particular to observe the (average) numbers of factors that are needed to achieve a high approximation degree (for instance, 80 % and higher). The experiments show that with the first few factors computed either by Algorithm 1 or Algorithm 2 one (usually) achieves relatively high degrees of approximation. This observation is based on experiments with

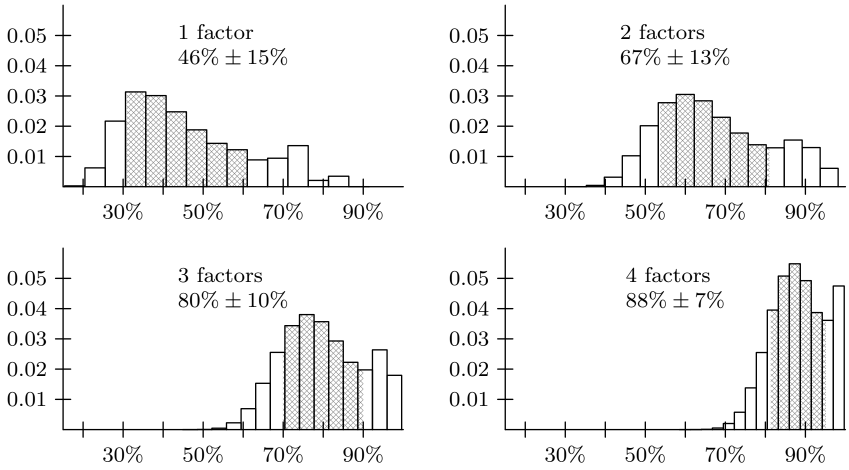

Fig. 4 Relative frequency histograms of degrees of approximation obtained using one up to four factors of randomly-generated 7-factorizable three-dimensional matrices.

approximate decompositions of randomly generated matrices with various percentages of 1s.

Fig. 4 depicts relative frequency histograms of degrees of approximation obtained by using the first four factors in a sample of 850, 000 randomly generated 7-factorizable three-dimensional matrices of various sizes, each matrix containing about one million entries. The top-left histogram shows the degree of approximation obtained by using the first factor. The top-right histogram shows the degree of approximation obtained by using the first two factors. Analogously, the bottom-left and bottom-right histograms refer to first three and first four factors. The hatched areas of the histograms are delimited by the intervals of the mean degrees of approximation ± the standard deviations. The mean values are also present in the diagrams. As one can see, the first two factors cover nearly 70 % of the input data (on average), and the first four factors cover nearly 90 % of the input data. Thus, even if the generated matrices are 7-factorizable, i.e., 7 factors are needed to achieve 100 % degree of factorization (the exact factorization), only the first four factors are sufficient to achieve 90 % degree of approximation which can be quite surprising. Hence, the approximate decomposition can help to reveal important factors covering most part of the data. In this particular case of 7-factorizable matrices, we can say that (typically) the first four factors are the most important ones.

5 Conclusions and Future Research

We proposed a new approach to factor analysis of three-way binary data. The approach utilizes triadic formal concepts as factors. Such a choice is justified by a theorem showing that triadic formal concepts provide us with optimal factorizations of three-way binary data. In addition, as illustrated by an example, triadic formal concepts are easily interpretable, resulting in an easy-to-understand output of the factor analysis. We proposed a greedy algorithm for computing suboptimal factorizations and provided its experimental evaluation.

The proposed method is to be further explored in several directions. First, one may impose additional constraints on the computed factor concepts, such as a suitable requirement of independence of factors. Second, one may look for other heuristics to compute suboptimal decompositions. Third, one may explore approximate factorizations that approximate the input matrix not only from below, as the proposed method does, but also from above using cuboids that contain a certain number of zero entries. Fourth, one may investigate the applications of the proposed method in data analysis of three-way binary data as well as the applications in data preprocessing (see e.g. [16] for applications of factor analysis of two-way binary data in classifying binary data).

A particular topic for future research is the relationship of the proposed method to the method from [8] where a link between a decomposition of three-way data and formal concept analysis was established for the first time. The authors in [8] were concerned with a covering of I by triadic concepts, however in a way different from the one proposed in our paper. Namely, they search for a triadic concept that covers a large part of I. The subrelation of I corresponding to such a concept (i.e. the Cartesian product of the extent, intent, and modus) is then removed from I. Such a removal continues until I contains no entries. Such a coverage may be described as matrix decomposition of I as described in our paper. It is easy to find an example showing that the method from [8] may produce decompositions with a larger than optimal number of factors. One obvious way to make the number of factors smaller is suggested by Theorem 1 and its proof: One may merge the triadic concepts produced by the method into a single triadic concept of I in which they are contained (such triadic concepts may exist because they are computed from different subrelations of I by the method in [8]). Extending furthermore every other concept produced by the method to a triadic concept of I will transform the decomposition of I from [8] into a decomposition described in this paper. An evaluation of this procedure and a detailed comparison of the performance of the two methods, both from the efficiency and factor quality point of view is a topic for future research.

After receiving the review reports, we learned about a recent paper [13] on factor analysis of three-way binary data. There is a small overlap in the theoretical parts of the papers ((1) of Theorem 1 for which we present a simpler argument). The paper contains a different method for computing the factor decomposition. A detailed comparison of the two approaches is a topic for future research.

Acknowledgements R. Belohlavek and V. Vychodil acknowledge support by Grant No. P202/10/0262 of the Czech Science Foundation. The paper is an extended version of the conference contributions "Belohlavek R., Vychodil V.: Optimal factorization of three-way

binary data. FDPM 2010, pp. 61–66" and "Glodeanu C.V.: Triadic factor analysis. CLA 2010, 127–138".

References

1. Belohlavek R., Vychodil V.: Discovery of optimal factors in binary data via a novel method of matrix decomposition. *J. Computer and System Sci.* **76**(1)(2010), 3–20.
2. Cichocki A. *et al.*: *Nonnegative Matrix and Tensor Factorizations: Applications to Exploratory Multi-way Data Analysis and Blind Source Separation.* J. Wiley, 2009.
3. Cormen T. H. *et al.*: *Introduction to Algorithms, 2nd Ed.* MIT Press, 2001.
4. Frolov A. A. *et al.*: Boolean factor analysis by Hopfield-like autoassociative memory. *IEEE Trans. Neural Networks* **18**(3)(2007), 698–707.
5. Ganter B., Wille R.: *Formal Concept Analysis. Mathematical Foundations.* Springer, Berlin, 1999.
6. Jäschke R., Hotho A., Schmitz C., Ganter B., Stumme G.: TRIAS – An Algorithm for Mining Iceberg Tri-Lattices. *Proc. ICDM 2006*, pp. 907–911.
7. Kolda T. G., Bader B. W.: Tensor decompositions and applications. *SIAM Review* **51**(3)(2009), 455–500.
8. Krolak-Schwerdt,S. , Orlik P., Ganter B.: TRIPAT: A model for analyzing three-mode binary data *Studies in Classification, Data Anlysis and Knowledge Organization*, **4**(1994), 298–307.
9. Kroonenberg P. M.: *Applied Multiway Data Analysis.* J. Wiley, 2008.
10. Kuznetsov S., Obiedkov S.: Comparing performance of algorithms for generating concept lattices. *J. Experimental and Theoretical Articial Intelligence* **14**(2–3)(2002), 189–216.
11. Lehmann F., Wille R.: A triadic approach to formal concept analysis. *Lecture Notes in Computer Science* **954**(1995), 32–43.
12. Mickey M. R., Mundle P., Engelman L.: Boolean factor analysis. In: W.J. Dixon (Ed.), BMDP statistical software manual, vol. 2, 849–860, Berkeley, CA: University of California Press, 1990.
13. Miettinen P.: Boolean tensor factorization. ICDM 2011, 11th IEEE International Conference on Data Mining, December 2011, pp. 447–456.
14. Miettinen P., Mielikäinen T., Gionis A., Das G., Mannila H.: The Discrete Basis Problem. PKDD 2006, *Lecture Notes in Artificial Intelligence* **4213**, pp. 335–346.
15. Nau D. S., Markowsky G., Woodbury M. A., Amos D. B.: A Mathematical Analysis of Human Leukocyte Antigen Serology. *Math. Biosciences* **40**(1978), 243–270.
16. Outrata J.: Boolean factor analysis for data preprocessing in machine learning. In: Draghici S. et al (Eds.): Proc. ICMLA 2010, Intern. Conf. on Machine Learning and Applications, Washington, DC, pp. 899-902, IEEE, DOI 10.1109/ICMLA.2010.141.
17. Smilde A., Bro R., Geladi P.: *Multi-way Analysis: Applications in the Chemical Sciences.* J. Wiley, 2004.
18. Stockmeyer L. J.: The set basis problem is NP-complete. IBM Research Report RC5431, Yorktown Heights, NY, 1975.
19. Tang F., Tao H.: Binary principal component analysis. Proc. British Machine Vision Conference 2006, pp. 377–386, 2006.
20. Tatti N., Mielikäinen T., Gionis A., Mannila H.: What is the dimension of your binary data? ICDM 2006, pp. 603–612.
21. Wille R.: The basic theorem of triadic concept analysis. *Order* **12**(1995), 149–158.