Journal of Computer and System Sciences 76 (2010) 3–20

Contents lists available at ScienceDirect

Journal of Computer and System Sciences

www.elsevier.com/locate/jcss

Discovery of optimal factors in binary data via a novel method of matrix decomposition ☆

Radim Belohlavek [a,b,*], Vilem Vychodil [a,b]

[a] Department of Systems Science and Industrial Engineering, Binghamton University—SUNY, Binghamton, NY 13902, USA
[b] Department of Computer Science, Palacký University, Tomkova 40, CZ-779 00 Olomouc, Czech Republic

ARTICLE INFO

Article history:
Received 15 December 2007
Received in revised form 30 July 2008
Available online 18 May 2009

Dedicated to Professor Rudolf Wille

Keywords:
Binary matrix decomposition
Factor analysis
Binary data
Formal concept analysis
Concept lattice

ABSTRACT

We present a novel method of decomposition of an $n \times m$ binary matrix I into a Boolean product $A \circ B$ of an $n \times k$ binary matrix A and a $k \times m$ binary matrix B with k as small as possible. Attempts to solve this problem are known from Boolean factor analysis where I is interpreted as an object–attribute matrix, A and B are interpreted as object–factor and factor–attribute matrices, and the aim is to find a decomposition with a small number k of factors. The method presented here is based on a theorem proved in this paper. It says that optimal decompositions, i.e. those with the least number of factors possible, are those where factors are formal concepts in the sense of formal concept analysis. Finding an optimal decomposition is an NP-hard problem. However, we present an approximation algorithm for finding optimal decompositions which is based on the insight provided by the theorem. The algorithm avoids the need to compute all formal concepts and significantly outperforms a greedy approximation algorithm for a set covering problem to which the problem of matrix decomposition is easily shown to be reducible. We present results of several experiments with various data sets including those from CIA World Factbook and UCI Machine Learning Repository. In addition, we present further geometric insight including description of transformations between the space of attributes and the space of factors.

1. Problem description and preliminaries

1.1. Problem description

Matrix decomposition methods provide representations of an object–variable data matrix by a product of two different matrices, one describing a relationship between objects and hidden variables or factors, and the other describing a relationship between factors and the original variables. In this paper, we consider the following problem:

Problem. Given an $n \times m$ matrix I with $I_{ij} \in \{0, 1\}$, a decomposition of I is sought into a Boolean matrix product $A \circ B$ of an $n \times k$ matrix A with $A_{il} \in \{0, 1\}$ and a $k \times m$ matrix B with $B_{lj} \in \{0, 1\}$ with k as small as possible.

☆ Supported by grant No. 201/05/0079 of the Czech Science Foundation, by grant No. 1ET101370417 of GA AV ČR, and by institutional support, research plan MSM 6198959214. This paper is an extended version of "R. Belohlavek, V. Vychodil: On Boolean factor analysis with formal concepts as factors, in: SCIS & ISIS 2006, Tokyo, Japan, pp. 1054–1059."

* Corresponding author at: Thomas J. Watson School of Engineering and Applied Science, Binghamton University, Department of Systems Science and Industrial Engineering, Binghamton, NY, United States.
E-mail addresses: rbelohla@binghamton.edu (R. Belohlavek), vychodil@binghamton.edu (V. Vychodil).

0022-0000/$ – see front matter
doi:10.1016/j.jcss.2009.05.002

Note that the smallest number k mentioned above is known as the Schein rank of matrix I in the Boolean matrix theory, see [15, p. 37]. Recall that a Boolean matrix (or binary matrix) is a matrix whose entries are 0 or 1. A Boolean matrix product $A \circ B$ is defined by

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

where $\bigvee$ denotes maximum (truth function of logical disjunction) and $\cdot$ is the usual product (truth function of logical conjunction). As an example,

$$\begin{pmatrix} 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 & 0 \\ 1 & 0 & 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \circ \begin{pmatrix} 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 \end{pmatrix}.$$

Boolean matrix product represents a non-linear relationship between objects, factors, and original attributes. By this, we mean that if v_1 and v_2 are two $1 \times k$ binary vectors for which $v_1 + v_2$ (ordinary sum of vectors) is a binary vector, then $(v_1 + v_2) \circ B$ is in general different from $v_1 \circ B + v_2 \circ B$ (ordinary sum of matrices). Nevertheless, the relationship between objects, factors, and original attributes specified by $I = A \circ B$ has a clear verbal description. Namely, let I, A, and B represent an object–attribute relationship ($I_{ij} = 1$ means that object i has attribute j), an object–factors relationship ($A_{il} = 1$ means that factor l applies to object i), and a factor–attribute relationship ($B_{lj} = 1$ means that attribute j is one of the particular manifestations of factor l). Then $I = A \circ B$ says: "object i has attribute j if and only if there is a factor l such that l applies to i and j is one of the manifestations of l." From this description, we immediately see that a decomposition $I = A \circ B$ has the following interpretation:

Factor analysis interpretation. Decomposition of I into $A \circ B$ corresponds to discovery of k factors which explain the data represented by I.

1.2. Related work

Reducing dimensionality of data by mapping the data from a space of directly observable variables into a lower-dimensional space of new variables is of fundamental importance for understanding and management of data. Classical methods such as PCA (principal component analysis), IDA (independent component analysis), SVD (singular value decomposition), or FA (factor analysis) are available for data represented by real-valued matrices, see e.g. [12,21]. Non-negative matrix factorization [18] is an example of a method for decomposition of a constrained matrix into a product of constrained matrices. Namely, the constraint is that the matrix entries of both the input matrix and the output matrices are non-negative. For data represented by binary matrices, several methods have been proposed. Some of them are based on various extensions of the methods developed for real-valued matrices, see e.g. [19,26,27,29,33] and also [30] for further references. By and large, these methods yield a decomposition of a binary matrix into a product of possibly real-valued non-binary matrices which causes problems regarding interpretation, see e.g. [30].

Directly relevant to our paper are methods which attempt to decompose a binary matrix I into $A \circ B$ with $\circ$ being the Boolean product of binary matrices defined by (1). Early work on this topic includes [23,24,28]. These papers already contain complexity results showing hardness of problems related to such decompositions. Another source of information on composition and decomposition of binary matrices is a monograph on binary matrices [15]. These early works seem to be forgotten in the recent work on decompositions of binary matrices.

The recent work includes an approach using Hopfield-like associative neural networks which has been studied in a series of papers by Frolov et al., see e.g. [8]. This approach is a heuristic in which the factors correspond to attractors of the neural network. Little theoretical insight regarding the decomposition is used in this approach. In the introduction, the authors say [8, p. 698]: "However, there is no general algorithm for the analysis except a brute force search. This method requires an exponentially increasing number of trials as the dimension of the pattern space increases. It can, therefore, only be used when this dimension is relatively small." In this paper, we develop a theoretical insight which leads to quite an efficient algorithm for finding factors and matrix decompositions and which therefore eliminates the need for a brute force search mentioned in [8, p. 698]. [10] studies the problem of covering binary matrices with their submatrices containing 1s, which is relevant to our approach. Namely, as we note in Section 2.6, such a covering is equivalent to an approximation of a binary matrix by a Boolean product $A \circ B$ from below. Another paper on this topic is [14], where the authors proposed to use formal concepts as factors for the decomposition problem. [14] does not provide any theoretical insight and even does not say how formal concepts are to be used. Basically, it describes results of experiments with small data. [14] can be seen as the inspiration for our paper. We developed basic insight on the role of formal concepts for decompositions of binary matrices in our conference paper "R. Belohlavek, V. Vychodil: On Boolean factor analysis with formal concepts as factors, in: SCIS & ISIS 2006, Tokyo, Japan, pp. 1054–1059." The present paper is an extended version of this paper. [22] presents an algorithm for finding approximate decompositions of binary matrices into a Boolean product of binary matrices which is based on associations between columns of the matrix. [30] contains several remarks and experiments regarding the

general question of dimensionality of binary data, including comparisons of various methods for decomposition of binary matrices into products of non-binary matrices. [31] looks at relationship between several problems related to decomposition of binary matrices. [20] discusses problems related to this paper, namely a problem of covering a binary matrix by so-called pertinent formal concepts which are formal concepts minimizing an entropy function associated to a classification problem discussed in that paper. Unfortunately, the algorithm presented in [20] is not described clearly enough for us to implement it and perform comparisons with our algorithms.

1.3. Content in brief

We present a novel method for the problem of decomposition of binary matrices described in Section 1.1. We prove an optimality theorem saying that the decompositions with the least number k of factors are those where the factors are formal concepts in the sense of formal concept analysis. The constructive aspect of this theorem is that the set of formal concepts associated to the input matrix provides us with a non-redundant space of factors for optimal decompositions. This finding provides a new perspective for decompositions of binary matrices and Boolean factor analysis. It opens a way to methods which are better than the brute force search methods and their variants, see e.g. [8]. An immediate consequence of our theorem is that the decomposition problem is reducible to the well-known set covering problem. This reduction enables us to adopt a greedy approximation algorithm which is available for the set covering problem to find approximately optimal decompositions of binary matrices. We present another greedy approximation algorithm for finding optimal decompositions which significantly outperforms the algorithm for set covering problem. The algorithm avoids the need to compute the set of all formal concepts associated to the input matrix. We present experiments for comparing performance of the algorithms as well as experiments demonstrating factorization of real data sets. For this purpose, we use CIA World Factbook data and UCI Machine Learning Repository data. In addition, we present results on transformations between the space of original attributes and the space of factors. We outline future research topics including an extension of our approach to decomposition of matrices containing degrees from partially ordered scales such as the unit interval $[0, 1]$.

1.4. Preliminaries on formal concept analysis

Formal concept analysis (FCA) is a method for data analysis with applications in various domains [5,9]. Our paper can be seen as an application of FCA in data preprocessing. Namely, the factors in a binary matrix we are looking for are sought in the set of all formal concepts associated to the matrix.

Let $X = \{1, \ldots, n\}$ and $Y = \{1, \ldots, m\}$ be sets of objects and attributes, respectively, and I be a binary relation between X and Y. The triplet $\langle X, Y, I\rangle$ is called a formal context. Since a binary relation between X and Y can be represented by an $n \times m$ Boolean matrix, we denote both the binary relation and the corresponding Boolean matrix by I. That is, for the entry I_{ij} of (matrix) I we have $I_{ij} = 1$ iff the pair $\langle i, j\rangle$ belongs to (relation) I and $I_{ij} = 0$ if $\langle i, j\rangle$ does not belong to (relation) I. A formal concept of $\langle X, Y, I\rangle$ is any pair $\langle C, D\rangle$ of sets $C \subseteq X$ (so-called extent) and $D \subseteq Y$ (so-called intent) such that $C^{\uparrow} = D$ and $D^{\downarrow} = C$ where

$$C^{\uparrow} = \left\{ y \in Y \;\middle|\; \text{for each } x \in C\colon\ \langle x, y\rangle \in I \right\}$$

is the set of all attributes shared by all objects from C, and

$$D^{\downarrow} = \left\{ x \in X \;\middle|\; \text{for each } y \in D\colon\ \langle x, y\rangle \in I \right\}$$

is the set of all objects sharing all attributes from D. The set of all formal concepts of $\langle X, Y, I\rangle$ is denoted by $\mathcal{B}(X, Y, I)$. That is,

$$\mathcal{B}(X, Y, I) = \left\{ \langle C, D\rangle \;\middle|\; C^{\uparrow} = D,\ D^{\downarrow} = C \right\}.$$

Under partial order $\leqslant$ defined by

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

for $\langle C_1, D_1\rangle, \langle C_2, D_2\rangle \in \mathcal{B}(X, Y, I)$, $\mathcal{B}(X, Y, I)$ happens to be a complete lattice, so-called concept lattice associated to $\langle X, Y, I\rangle$ [9,32]. That is, for any subset K of $\mathcal{B}(X, Y, I)$, there exists the least formal concept in $\mathcal{B}(X, Y, I)$ greater than or equal to any formal concept from K (the supremum of K) as well as the greatest formal concept in $\mathcal{B}(X, Y, I)$ smaller than or equal to any formal concept from K (the infimum of K). Efficient algorithms for computing $\mathcal{B}(X, Y, I)$ exist and a good overview is provided by [17].

Example 1. Let a binary matrix describing a binary relation I between $X = \{1, \ldots, 4\}$ and $Y = \{1, \ldots, 5\}$ be given by

$$I = \begin{pmatrix} 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 & 0 \\ 1 & 0 & 0 & 0 & 1 \end{pmatrix}.$$

Then, the pair $\langle\{1, 2, 3\}, \{1, 2\}\rangle$ is a formal concept of $\mathcal{B}(X, Y, I)$ because $\{1, 2, 3\}^{\uparrow} = \{1, 2\}$ and $\{1, 2\}^{\downarrow} = \{1, 2, 3\}$. Further formal concepts include $\langle\{1, 2, 3, 4\}, \{1\}\rangle$, $\langle\{2\}, \{1, 2, 5\}\rangle$, $\langle\{2, 4\}, \{1, 5\}\rangle$, $\langle\{3\}, \{1, 2, 3, 4\}\rangle$, and $\langle\emptyset, \{1,\ 2, 3, 4, 5\}\rangle$.

2. Discovering formal concepts as optimal factors

2.1. Formal concepts are optimal factors

Consider an $n \times m$ binary matrix I. Our aim is to decompose I into a Boolean product $I = A \circ B$ of binary matrices A and B of dimensions $n \times k$ and $k \times m$, as described above.

We are going to consider decompositions of I into a product

$$A_{\mathcal{F}} \circ B_{\mathcal{F}}$$

of binary matrices $A_{\mathcal{F}}$ and $B_{\mathcal{F}}$ constructed from a set $\mathcal{F}$ of formal concepts associated to I. In particular, consider the concept lattice $\mathcal{B}(X, Y, I)$ associated to I, with $X = \{1, \ldots, n\}$ and $Y = \{1, \ldots, m\}$. Let

$$\mathcal{F} = \{\langle A_1, B_1\rangle, \ldots, \langle A_k, B_k\rangle\} \subseteq \mathcal{B}(X, Y, I),$$

i.e. $\mathcal{F}$ is a set of formal concepts from $\mathcal{B}(X, Y, I)$. Denote by $A_{\mathcal{F}}$ and $B_{\mathcal{F}}$ the $n \times k$ and $k \times m$ binary matrices defined by

$$(A_{\mathcal{F}})_{il} = \begin{cases} 1 & \text{if } i \in A_l, \\ 0 & \text{if } i \notin A_l, \end{cases} \quad \text{and} \quad (B_{\mathcal{F}})_{lj} = \begin{cases} 1 & \text{if } j \in B_l, \\ 0 & \text{if } j \notin B_l, \end{cases}$$

for $l = 1, \ldots, k$. That is, the lth column $(A_{\mathcal{F}})_{\_l}$ of $A_{\mathcal{F}}$ consists of the characteristic vector of A_l and the lth row $(B_{\mathcal{F}})_{l\_}$ of $B_{\mathcal{F}}$ consists of the characteristic vector of B_l.

Example 2. For the binary matrix I being

$$\begin{pmatrix} 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 & 0 \\ 1 & 0 & 0 & 0 & 1 \end{pmatrix},$$

$\langle\{1, 2, 3\}, \{1, 2\}\rangle$ and $\langle\{1, 2, 3, 4\}, \{1\}\rangle$ are formal concepts of the associated concept lattice. Denoting them by $\langle A_1, B_1\rangle$ and $\langle A_2, B_2\rangle$, respectively, and putting

$$\mathcal{F} = \{\langle A_1, B_1\rangle, \langle A_2, B_2\rangle\},$$

we have

$$A_{\mathcal{F}} = \begin{pmatrix} 1 & 1 \\ 1 & 1 \\ 1 & 1 \\ 0 & 1 \end{pmatrix} \quad \text{and} \quad B_{\mathcal{F}} = \begin{pmatrix} 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \end{pmatrix}.$$

In the previous example, $I \neq A_{\mathcal{F}} \circ B_{\mathcal{F}}$. The question of whether for every I there is some $\mathcal{F} \subseteq \mathcal{B}(X, Y, I)$ such that $I = A_{\mathcal{F}} \circ B_{\mathcal{F}}$ has a positive answer.

Theorem 1 *(Universality of formal concepts as factors). For every I there is $\mathcal{F} \subseteq \mathcal{B}(X, Y, I)$ such that $I = A_{\mathcal{F}} \circ B_{\mathcal{F}}$.*

Proof. The proof follows from the fact that $I_{ij} = 1$ iff there is a formal concept $\langle C, D\rangle \in \mathcal{B}(X, Y, I)$ such that $i \in C$ and $j \in D$. Therefore, taking $\mathcal{F} = \mathcal{B}(X, Y, I)$ we have: $(A_{\mathcal{F}} \circ B_{\mathcal{F}})_{ij} = 1$ iff there is l such that $(A_{\mathcal{F}})_{il} = 1$ and $(B_{\mathcal{F}})_{lj} = 1$ iff there is $\langle C_l, D_l\rangle \in \mathcal{B}(X, Y, I)$ with $i \in C_l$ and $j \in D_l$ iff $I_{ij} = 1$. □

Moreover, decompositions using formal concepts as factors are optimal in that they yield the least number of factors possible:

Theorem 2 *(Optimality of formal concepts as factors). Let $I = A \circ B$ for $n \times k$ and $k \times m$ binary matrices A and B. Then there exists a set $\mathcal{F} \subseteq \mathcal{B}(X, Y, I)$ of formal concepts of I with*

$$|\mathcal{F}| \leqslant k$$

such that for the $n \times |\mathcal{F}|$ and $|\mathcal{F}| \times m$ binary matrices $A_{\mathcal{F}}$ and $B_{\mathcal{F}}$ we have

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

Proof. First, it is an easy-to-observe fact that formal concepts of I are just maximal rectangles of matrix I, i.e. maximal submatrices of I, which are full of 1s. For instance, in the binary matrix

$$\begin{pmatrix} \mathbf{1} & \mathbf{1} & 0 & 0 & 0 \\ \mathbf{1} & \mathbf{1} & 0 & 0 & 1 \\ \mathbf{1} & \mathbf{1} & 1 & 1 & 0 \\ 1 & 0 & 0 & 0 & 1 \end{pmatrix},$$

the bold **1**s form a maximal rectangle, with rows 1, 2, 3, and columns 1 and 2. This rectangle corresponds to the formal concept $\langle\{1,2,3\},\{1,2\}\rangle$ of I. Contrary to that, the rectangle corresponding to rows 1 and 3, and columns 1 and 2, is not maximal because it is contained in the one consisting of bold **1**s.

Second, observe that $I = A \circ B$ for an $n \times k$ matrix A and a $k \times m$ matrix B means that I can be seen as a $\bigvee$-superposition, i.e. as a union, of k rectangles consisting of 1s (not necessarily maximal ones). Namely, since

$$I_{ij} = A_{i1} \cdot B_{1j} \vee \cdots \vee A_{ik} \cdot B_{kj},$$

I is a union of rectangles $J_l = A_{\_l} \circ B_{l\_}$, $l = 1, \ldots, k$. Note that $A_{\_l} \circ B_{l\_}$ is an $n \times m$ matrix which results by a Boolean multiplication of the lth column $A_{\_l}$ of A and the lth row $B_{l\_}$ of B. As an example, with $I = A \circ B$ being

$$\begin{pmatrix} 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 & 0 \\ 1 & 0 & 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \circ \begin{pmatrix} 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 \end{pmatrix},$$

the decomposition can be rewritten as a $\bigvee$-superposition

$$\begin{pmatrix} 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 & 0 \\ 1 & 0 & 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix} \vee \begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix} \vee \begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 \end{pmatrix} \vee \begin{pmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

of rectangles J_1, J_2, J_3, J_4, i.e. binary matrices whose 1s form rectangles.

Third, let now $I = A \circ B$ for an $n \times k$ binary matrix A and a $k \times m$ binary matrix B and consider the corresponding rectangles $J_1, \ldots, J_k$, i.e. binary matrices where 1s form rectangles. Each such a rectangle J_l is obviously contained in a maximal rectangle J'_l of I, i.e. $(J_l)_{ij} \leqslant (J'_l)_{ij}$ for all i, j. Now, every J'_l corresponds to a formal concept $\langle C_l, D_l\rangle$ in that $(J'_l)_{ij} = 1$ iff $i \in C_l$ and $j \in D_l$. Since

$$I_{ij} = \bigvee_{l=1}^{k} (J_l)_{ij} \leqslant \bigvee_{l=1}^{k} \left(J'_l\right)_{ij} \leqslant I_{ij},$$

a $\bigvee$-superposition of maximal rectangles J'_l yields I. Putting therefore

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

we get $I = A_{\mathcal{F}} \circ B_{\mathcal{F}}$ and $|\mathcal{F}| \leqslant k$. Note that since two distinct rectangles may be contained in a single maximal rectangle, we may have $|\mathcal{F}| < k$. □

We will see later that the geometric argument behind the proof enables us to see that the problem to find a decomposition of I into $A_{\mathcal{F}} \circ B_{\mathcal{F}}$ can be reduced to a particular instance of a set covering optimization problem.

Example 3. Consider the decomposition

$$\begin{pmatrix} 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 & 0 \\ 1 & 0 & 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \circ \begin{pmatrix} 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 \end{pmatrix},$$

and the corresponding rectangles $J_1, \ldots, J_4$, which are

$$\begin{pmatrix} 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}, \begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}, \begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 \end{pmatrix}, \begin{pmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}.$$

Furthermore, consider the formal concepts $\langle A_1, B_1\rangle = \langle\{1,2,3\},\{1,2\}\rangle$, $\langle A_2, B_2\rangle = \langle\{3\},\{1,2,3,4\}\rangle$, and $\langle A_3, B_3\rangle = \langle\{2,4\},\{1,5\}\rangle$ of I. Each of the rectangles corresponding to J_ls ($l = 1, \ldots, 4$) is contained in some of the maximal rectangles corresponding to $\langle A_1, B_1\rangle$, $\langle A_2, B_2\rangle$, or $\langle A_3, B_3\rangle$. Putting now $\mathcal{F} = \{\langle A_1, B_1\rangle, \langle A_2, B_2\rangle, \langle A_3, B_3\rangle\}$, we have $I = A_{\mathcal{F}} \circ B_{\mathcal{F}}$, i.e.

$$\begin{pmatrix} 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 & 0 \\ 1 & 0 & 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \circ \begin{pmatrix} 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 \\ 1 & 0 & 0 & 0 & 1 \end{pmatrix}.$$

Denoting by $(A_{\mathcal{F}})_{\_l}$ and $(B_{\mathcal{F}})_{l\_}$ the lth column of $A_{\mathcal{F}}$ and the lth row of $B_{\mathcal{F}}$, $I = A_{\mathcal{F}} \circ B_{\mathcal{F}}$ can further be rewritten as $I = (A_{\mathcal{F}})_{\_1} \circ (B_{\mathcal{F}})_{1\_} \vee (A_{\mathcal{F}})_{\_2} \circ (B_{\mathcal{F}})_{2\_} \vee (A_{\mathcal{F}})_{\_3} \circ (B_{\mathcal{F}})_{3\_}$, which yields a $\bigvee$-decomposition of I into maximal rectangles. With our example, we have

$$\begin{pmatrix} 1&1&0&0&0\\ 1&1&0&0&1\\ 1&1&1&1&0\\ 1&0&0&0&1 \end{pmatrix} = \begin{pmatrix} 1&1&0&0&0\\ 1&1&0&0&0\\ 1&1&0&0&0\\ 0&0&0&0&0 \end{pmatrix} \vee \begin{pmatrix} 0&0&0&0&0\\ 0&0&0&0&0\\ 1&1&1&1&0\\ 0&0&0&0&0 \end{pmatrix} \vee \begin{pmatrix} 0&0&0&0&0\\ 1&0&0&0&1\\ 0&0&0&0&0\\ 1&0&0&0&1 \end{pmatrix}.$$

2.1.1. Interpretation of factors

In factor analysis, factors are considered to represent general categories [13, p. 4]: "The principal aim of factor analysis is the resolution of a set of variables … in terms of (usually) a small number of categories called 'factors' … ." Note that the categories are sometimes called concepts. The original variables/attributes are considered as particular manifestations of these categories. The problem of interpretation of factors is a part of the whole process of factor analysis. In this regard, interpretation of formal concepts as factors in Boolean factor analysis is straightforward and easy to understand for a user. Namely, the conception of FCA is motivated by a robust understanding of a concept developed in traditional logic [16] in which a formal concept consists of its extent and intent. Accordingly, a formal concept $\langle C_l, D_l \rangle$ can be seen as a collection C_l of objects to which it applies (concept's extent) and a collection D_l of attributes to which it applies (concept's intent), see Section 1.4. Pragmatic aspects of this approach are discussed in [7]. Clear interpretability is an advantageous feature of using formal concepts as factors.

Definition. A set $\mathcal{F} \subseteq \mathcal{B}(X, Y, I)$ is called a set of *factor concepts* if $I = A_{\mathcal{F}} \circ B_{\mathcal{F}}$.

Note that the argument used in the proof of Theorem 2 implies that for any decomposition $I = A \circ B$, factors correspond to rectangles of 1s (submatrices of I consisting of 1s). To every such factor, therefore, corresponds a set C of objects and a set D of attributes. As a result, factors in Boolean factor analysis in general can be regarded as covering objects and applying to attributes. That is, every factor can be interpreted as having its extent C and intent D. The intuitive requirement to include all objects to which attributes from D apply and to include all attributes shared by all objects from C supports the idea of having maximal C and D and, therefore, having maximal rectangles. A crucial argument for maximal rectangles as factors is, nevertheless, Theorem 2 and further practical advantages of maximal rectangles including computational treatability, cf. [17].

2.1.2. Mandatory factors

The next theorem shows that certain formal concepts are mandatory in that they need to be included in every $\mathcal{F}$ for which $I = A_{\mathcal{F}} \circ B_{\mathcal{F}}$. Recall [9] that the sets $\mathcal{O}(X, Y, I)$ of all object concepts and $\mathcal{A}(X, Y, I)$ of all attribute concepts are defined by

$$\mathcal{O}(X, Y, I) = \left\{ \left\langle \{x\}^{\uparrow\downarrow}, \{x\}^{\uparrow} \right\rangle \mid x \in X \right\},$$
$$\mathcal{A}(X, Y, I) = \left\{ \left\langle \{y\}^{\downarrow}, \{y\}^{\downarrow\uparrow} \right\rangle \mid y \in Y \right\}.$$

Formal concepts which are both object and attribute concepts are mandatory:

Theorem 3 *(Mandatory factors). If $I = A_{\mathcal{F}} \circ B_{\mathcal{F}}$ for some $\mathcal{F} \subseteq \mathcal{B}(X, Y, I)$ then $\mathcal{O}(X, Y, I) \cap \mathcal{A}(X, Y, I) \subseteq \mathcal{F}$.*

Proof. If $\langle C, D \rangle \in \mathcal{O}(X, Y, I) \cap \mathcal{A}(X, Y, I)$, i.e. $\langle C, D \rangle = \langle \{x\}^{\uparrow\downarrow}, \{x\}^{\uparrow} \rangle$ for some $x \in X$ and $\langle C, D \rangle = \langle \{y\}^{\downarrow}, \{y\}^{\downarrow\uparrow} \rangle$ for some $y \in Y$, then $\langle C, D \rangle$ is the only formal concept from $\mathcal{B}(X, Y, I)$ for which $x \in C$ and $y \in D$. Namely, if $\langle C_1, D_1 \rangle$ is another such formal concept then from $\{x\} \subseteq C_1$ we get $C = \{x\}^{\uparrow\downarrow} \subseteq C_1^{\uparrow\downarrow} = C_1$. Applying the antitony of $^{\uparrow}$, we get $D = C^{\uparrow} \supseteq C_1^{\uparrow} = D_1$. Furthermore, from $\{y\} \subseteq D_1$ we get $D = \{y\}^{\downarrow\uparrow} \subseteq D_1^{\downarrow\uparrow} = D_1$. This gives $D = D_1$ and, as a consequence, $\langle C, D \rangle = \langle C_1, D_1 \rangle$. The maximal rectangle corresponding to $\langle C, D \rangle$ is thus the only one which covers the 1 at the intersection of the row and the column corresponding to x and y, respectively. □

2.2. Illustrative example

In this section, we present a small illustrative example. Suppose we have a record of a collection of patients. For every patient, our record contains a set of his symptoms. For our purpose, we consider a set Y of 8 symptoms which we denote by $1, \ldots, 8$, i.e. $Y = \{1, \ldots, 8\}$. The symptoms are described in Table 1.

Suppose our collection contains 12 patients. We denote the ith patient by i and put $X = \{1, \ldots, 12\}$. Our record describing patients and symptoms is given by the following 12×8 binary matrix I:

Table 1
Symptoms and their descriptions.

Symptom	Symptom description
1	headache
2	fever
3	painful limbs
4	swollen glands in neck
5	cold
6	stiff neck
7	rash
8	vomiting

Table 2
Formal concepts of data given by patients and their symptoms.

c_i	$\langle A_i, B_i \rangle$
c_0	$\langle \{\}, \{1,2,3,4,5,6,7,8\} \rangle$
c_1	$\langle \{1,5,9,11\}, \{1,2,3,5\} \rangle$
c_2	$\langle \{2,4,12\}, \{1,2,6,8\} \rangle$
c_3	$\langle \{3,6,7\}, \{2,5,7\} \rangle$
c_4	$\langle \{3,6,7,8,10\}, \{7\} \rangle$
c_5	$\langle \{1,3,5,6,7,9,11\}, \{2,5\} \rangle$
c_6	$\langle \{1,2,4,5,9,11,12\}, \{1,2\} \rangle$
c_7	$\langle \{1,2,3,4,5,6,7,9,11,12\}, \{2\} \rangle$
c_8	$\langle \{1,2,3,4,5,6,7,8,9,10,11,12\}, \{\} \rangle$

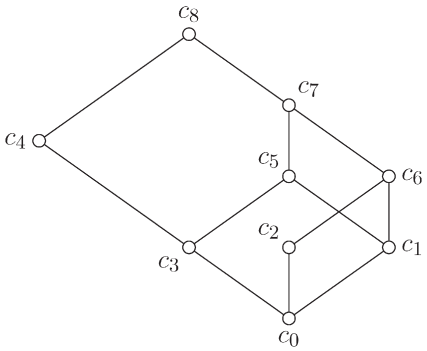

Fig. 1. Hasse diagram of concept lattice given by patients and their symptoms.

$$I = \begin{pmatrix} 1&1&1&0&1&0&0&0 \\ 1&1&0&0&0&1&0&1 \\ 0&1&0&0&1&0&1&0 \\ 1&1&0&0&0&1&0&1 \\ 1&1&1&0&1&0&0&0 \\ 0&1&0&0&1&0&1&0 \\ 0&1&0&0&1&0&1&0 \\ 0&0&0&0&0&0&1&0 \\ 1&1&1&0&1&0&0&0 \\ 0&0&0&0&0&0&1&0 \\ 1&1&1&0&1&0&0&0 \\ 1&1&0&0&0&1&0&1 \end{pmatrix}.$$

That is, rows correspond to patients, columns correspond to symptoms, $I_{ij} = 1$ if patient i has symptom j, and $I_{ij} = 0$ if patient i does not have symptom j.

Our intention is to find a set $\mathcal{F}$ of factor concepts. That is, we want to find some $\mathcal{F} \subseteq \mathcal{B}(X, Y, I)$ such that $I = A_{\mathcal{F}} \circ B_{\mathcal{F}}$. Let us first look at the concept lattice $\mathcal{B}(X, Y, I)$. $\mathcal{B}(X, Y, I)$ contains 9 formal concepts. That is $\mathcal{B}(X, Y, I) = \{c_0, \ldots, c_8\}$ and each c_i is of the form $c_i = \langle A_i, B_i \rangle$ where $A_i \subseteq X$ is a set of patients, $B_i \subseteq Y$ is a set of symptoms such that $A_i^{\uparrow} = B_i$ and $B_i^{\downarrow} = A_i$. That is, B_i is the set of all symptoms common to all patients from A_i, and A_i is the set of all patients sharing all symptoms from B_i. All formal concepts from $\mathcal{B}(X, Y, I)$ are depicted in Table 2.

For instance, the extent A_3 of the formal concept c_3 consists of patients $3, 6, 7$, and the intent B_3 of c_3 consists of attributes $2, 5, 7$. The concept lattice $\mathcal{B}(X, Y, I)$ equipped with a partial order $\leqslant$ (the subconcept-superconcept hierarchy) can be visualized by its Hasse diagram. The Hasse diagram of $\mathcal{B}(X, Y, I)$ is shown in Fig. 1. We can see from the diagram that, e.g., $c_3 \leqslant c_4$, i.e., formal concept c_4 is more general than formal concept c_3. This is because the extent A_3 of c_3 is contained in the extent A_4 of c_4, i.e. $A_3 \subseteq A_4$, meaning that each patient covered by c_3 is also covered by c_4. Equivalently,

Table 3
Diseases and their symptoms (an excerpt).

Disease	Symptoms
chickenpox	rash
flu	headache, fever, painful limbs, cold
measles	fever, cold, rash
meningitis	headache, fever, stiff neck, vomiting
⋮	⋮

Table 4
Description of formal concepts of data given by patients and their symptoms.

Formal concept	Concept description
c_0	empty concept
c_1	"flu"
c_2	"meningitis"
c_3	"measles"
c_4	"chickenpox"
c_5	"suspicion of flu or measles"
c_6	"suspicion of flu or meningitis"
c_7	"suspicion of flu or measles or meningitis"
c_8	universal concept

the intent B_4 of c_4 is contained in the intent B_3 of c_3, i.e. $B_4 \subseteq B_3$, meaning that each attribute characteristic for c_4 is also a characteristic attribute of c_3. In a similar way, one can see that $c_3 \leqslant c_8$, $c_2 \leqslant c_7$, etc.

Let us now look at the meaning of formal concepts $c_0, \ldots, c_8$ from $\mathcal{B}(X, Y, I)$. One naturally expects that the concepts, which are given by groups of patients (concept extents) and groups of symptoms (concept intents) are related to diseases. This is, indeed the case. Table 3 shows an excerpt from a family medical book. We can see that, e.g., formal concept c_1 represents flu because c_1 covers just headache, fever, painful limbs, cold, which are exactly the characteristic attributes of flu. In the same way, c_2, c_3, and c_4 represent meningitis, measles, and chickenpox. However, $\mathcal{B}(X, Y, I)$ contains also formal concepts which can be interpreted as "suspicion of (a disease)." For instance, c_5 can be interpreted as "suspicion of flu or measles" because the intent B_5 of c_5 contains attributes 2 (fever) and 5 (cold) which belong to the characteristic attributes of both flu and measles. Note that $\mathcal{B}(X, Y, I)$ also contains an "empty concept" c_0 (the concept is the least concept in the conceptual hierarchy and applies to no patient) and the universal concept c_8 (applies to all patients). The empty and the universal concepts are usually not interesting. Verbal descriptions of formal concepts from $c_0, \ldots, c_8$ are presented in Table 4. Note also that the verbal description and the conceptual hierarchy are compatible. For instance, we have $c_1 \leqslant c_6$, i.e. c_6 represents a weaker concept (more general concept) than c_1, which corresponds to their descriptions: c_1 is "flu," c_6 is "suspicion of flu or meningitis."

Consider now the problem of factorization of I. Our aim is to illustrate the above results. According to Theorem 1, we can factorize I by $\mathcal{F} = \mathcal{B}(X, Y, I)$. However, since $|\mathcal{F}| = 9 > 8 = |Y|$, this would mean that we would end up with a number of factors bigger than the number of the original attributes. Therefore, we are looking for some small set $\mathcal{F} \subset \mathcal{B}(X, Y, I)$ of factor concepts. First, as is easily seen, the empty concept and the universal concept can always be disregarded. Furthermore, $\mathcal{B}(X, Y, I)$ contains the following object- and attribute-concepts:

$$\mathcal{O}(X, Y, I) = \{c_1, c_2, c_3, c_4\}, \qquad \mathcal{A}(X, Y, I) = \{c_0, c_1, c_2, c_4, c_5, c_6, c_7\}.$$

Thus,

$$\mathcal{O}(X, Y, I) \cap \mathcal{A}(X, Y, I) = \{c_1, c_2, c_4\}.$$

According to Theorem 3, $\mathcal{F}$ needs to contain $\mathcal{O}(X, Y, I) \cap \mathcal{A}(X, Y, I) = \{c_1, c_2, c_4\}$. Then, $\mathcal{F}'' = \{c_1, c_2, c_4\}$ is almost a set of factor concepts. Namely, for $I'' = A_{\mathcal{F}''} \circ B_{\mathcal{F}''}$, I'' differs from I only in that $I''_{3,5} = I''_{6,5} = I''_{7,5} = 0$ while $I_{3,5} = I_{6,5} = I_{7,5} = 1$. From Fig. 1 we can directly see that the only formal concepts which cover the entries $\langle 3, 5\rangle$, $\langle 6, 5\rangle$, and $\langle 7, 5\rangle$, where I contains 1s, are c_3 and c_5. Therefore, there are just two minimal sets of factor concepts, namely

$$\mathcal{F} = \{c_1, c_2, c_3, c_4\} \quad \text{and} \quad \mathcal{F}' = \{c_1, c_2, c_4, c_5\}.$$

Thus, I can be decomposed by $I = A_{\mathcal{F}} \circ B_{\mathcal{F}}$ or $I = A_{\mathcal{F}'} \circ B_{\mathcal{F}'}$. For $I = A_{\mathcal{F}} \circ B_{\mathcal{F}}$ we have

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

and similarly for $A_{\mathcal{F}'}$ and $B_{\mathcal{F}'}$. From the point of view of dimension reduction, instead in an 8-dimensional space of symptoms (as described by I), the patients are now described in a 4-dimensional space of disease-like concepts (as described by $A_{\mathcal{F}}$).

2.3. Transformations between attributes space and factors space

Let now $I = A \circ B$ be a decomposition of I such that $A = A_{\mathcal{F}}$ and $B = B_{\mathcal{F}}$ for a set $\mathcal{F} \subseteq \mathcal{B}(X, Y, I)$ of formal concepts. For every object i ($1 \leqslant i \leqslant n$) we can consider its representation in the m-dimensional Boolean space $\{0,1\}^m$ of attributes and its representation in the k-dimensional Boolean space $\{0,1\}^k$ of factors. In the space of attributes, the vector representing object i is the ith row $I_{i\_}$ of I. In the space of factors, the vector representing i is the ith row $A_{i\_}$ of A. We are going to consider transformations between the space $\{0,1\}^m$ of attributes and the space $\{0,1\}^k$ of factors.

Let $\leqslant$ denote the componentwise partial order on the set $\{0,1\}^p$ of binary vectors, i.e. $\langle V_1, \ldots, V_p \rangle \leqslant \langle W_1, \ldots, W_p \rangle$ iff $V_1 \leqslant W_1, \ldots, V_p \leqslant W_p$. Clearly, $\{0,1\}^p$ equipped with $\leqslant$ forms a Boolean lattice. We are going to consider the mappings $g : \{0,1\}^m \to \{0,1\}^k$ and $h : \{0,1\}^k \to \{0,1\}^m$ defined for $P \in \{0,1\}^m$ and $Q \in \{0,1\}^k$ by

$$\big(g(P)\big)_l = \bigwedge_{j=1}^{m} (B_{lj} \to P_j), \tag{2}$$

$$\big(h(Q)\big)_j = \bigvee_{l=1}^{k} (Q_l \cdot B_{lj}), \tag{3}$$

for $1 \leqslant l \leqslant k$ and $1 \leqslant j \leqslant m$. Here, $\to$ denotes the truth function of classical implication ($0 \to 0 = 0 \to 1 = 1 \to 1 = 1$, $1 \to 0 = 0$), $\cdot$ denotes the usual product, and $\bigwedge$ and $\bigvee$ denote minimum and maximum, respectively. (2) says that the lth component of $g(P) \in \{0,1\}^k$ is 1 if and only if for every attribute j, $P_j = 1$ for all positions j for which $B_{lj} = 1$, i.e. the lth row of B is included in P. (3) says that the jth component of $h(Q) \in \{0,1\}^m$ is 1 if and only if there is factor l such that $Q_l = 1$ and $B_{lj} = 1$, i.e. attribute j is a manifestation of at least one factor from Q.

The following theorem justifies (2) and (3).

Theorem 4. *For $i \in \{1, \ldots, n\}$,*

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

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

Proof. $h(A_{i\_}) = I_{i\_}$ follows directly from $I = A \circ B$. To see $g(I_{i\_}) = A_{i\_}$, note first that since $A = A_{\mathcal{F}}$ and $B = B_{\mathcal{F}}$, the lth row $B_{l\_}$ of B coincides with the characteristic vector $c(D_l)$ of the intent D_l of a formal concept $\langle C_l, D_l \rangle \in \mathcal{F}$, and that the lth column $A_{\_l}$ of A coincides with the characteristic vector $c(C_l)$. Therefore, using $C_l = D_l^{\downarrow}$, we get

$$\big(g(I_{i\_})\big)_l = \bigwedge_{j=1}^{m} \big(B_{lj} \to (I_{i\_})_j\big) = \bigwedge_{j=1}^{m} \big(\big(c(D_l)\big)_j \to I_{ij}\big) = \big(c\big(D_l^{\downarrow}\big)\big)_i = \big(c(C_l)\big)_i = A_{il},$$

which proves the theorem. □

Note that it is essential for the previous theorem that the decomposition $I = A \circ B$ uses formal concepts as factors (in fact, essential is that columns of A are extents of formal concepts). The following theorem shows the basic properties of g and h.

Theorem 5. *For $P, P' \in \{0,1\}^m$ and $Q, Q' \in \{0,1\}^k$:*

$$P \leqslant P' \quad \text{implies} \quad g(P) \leqslant g(P'), \tag{4}$$

$$Q \leqslant Q' \quad \text{implies} \quad h(Q) \leqslant h(Q'), \tag{5}$$

$$h\big(g(P)\big) \leqslant P, \tag{6}$$

$$Q \leqslant g\big(h(Q)\big). \tag{7}$$

Proof. By routine verification using (2) and (3). (Note that the properties can also be proved by observing that g and h are so-called isotone Galois connections associated to matrix B, see [11].) □

The following properties can easily be derived from (4)–(7):

Corollary 1.

$$g(P) = g\big(h\big(g(P)\big)\big), \tag{8}$$

$$h(Q) = h\big(g\big(h(Q)\big)\big), \tag{9}$$

$$g\Big(\bigwedge_{s\in S} P_s\Big) = \bigwedge_{s\in S} g(P_s), \tag{10}$$

$$h\Big(\bigvee_{t\in T} Q_t\Big) = \bigvee_{t\in T} h(Q_t). \tag{11}$$

Observe now that properties (4)–(7) can be regarded as natural requirements for mappings transforming vectors between the attributes and factors spaces: (4) says that the more attributes an object has, the more factors apply, while (5) says that the more factors apply, the more attributes an object has. This is in accordance with the factor model implied by the decomposition $I = A \circ B$ which uses the Boolean matrix product. Namely, according to this model, an object has an attribute iff there is a factor which applies to the object and is associated to the attribute. Therefore, "more attributes" is positively correlated with "more factors" for an object. (6) corresponds to the idea that common attributes associated to all the factors which apply to a given object need to be contained in the collection of all attributes possessed by that object. (7) also has a simple meaning though the verbal description is somewhat cumbersome.

To get a further understanding of the transformations between the space of attributes and the space of factors, let for $P \in \{0,1\}^m$ and $Q \in \{0,1\}^k$ denote by $g^{-1}(Q)$ the set of all vectors mapped to Q by g and by $h^{-1}(P)$ the set of all vectors mapped to P by h, i.e.

$$g^{-1}(Q) = \big\{P \in \{0,1\}^m \bigm| g(P) = Q\big\},$$

$$h^{-1}(P) = \big\{Q \in \{0,1\}^k \bigm| h(Q) = P\big\}.$$

Then we have:

Theorem 6.

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

Note that $S \subseteq \{0,1\}^p$ is a convex subset if $V \in S$ whenever $U \leqslant V \leqslant W$ for some $U, W \in S$.

Proof. (1) Let P be from $g^{-1}(Q)$, i.e. $g(P) = Q$. Then, in particular, $Q \leqslant g(P)$. Using (5) and (6), $h(Q) \leqslant h(g(P)) \leqslant P$. Moreover, using (8) we get $Q = g(P) = ghg(P) = gh(Q)$, hence $h(Q) \in g^{-1}(Q)$. Therefore, $h(Q)$ is the least vector of $g^{-1}(Q)$. Let now $U, W \in g^{-1}(Q)$ and $U \leqslant V \leqslant W$. (4) yields $Q = g(U) \leqslant g(V) \leqslant g(W) = Q$, hence $g(V) = Q$, proving that $g^{-1}(Q)$ is convex.

The proof of (2) is analogous. □

Theorem 6 describes the geometry behind g and h: The space $\{0,1\}^m$ of attributes and the space $\{0,1\}^k$ of factors are partitioned into an equal number of convex subsets. The subsets of the space of attributes have least elements, the subsets of the space of factors have greatest elements. Every element of one of the convex subset of the space of attributes is mapped by g to the greatest element of the corresponding convex subset of the space of factors. Every element of one of the convex subset of the space of factors is mapped by h to the least element of the corresponding convex subset of the space of attributes. Theorem 6 is illustrated in Fig. 2

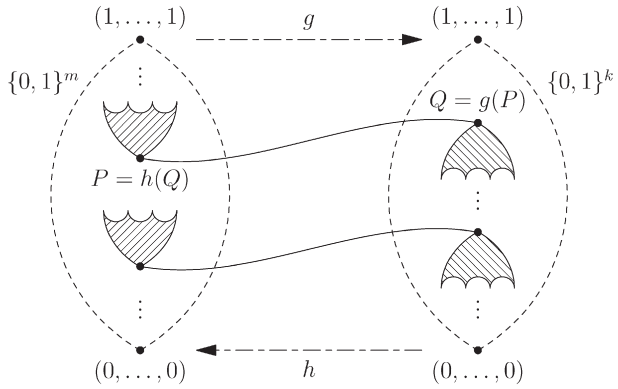

Fig. 2. Illustration of Theorem 6.

2.4. Reduction to set covering optimization problem and complexity

2.4.1. Reduction to set covering optimization problem

The above considerations allow us to see that the problem of finding a decomposition $I = A_{\mathcal{F}} \circ B_{\mathcal{F}}$ is reducible to the set covering optimization problem for which we refer to [6]. In the set covering optimization problem, we are given a set $\mathcal{U}$ and a collection $\mathcal{S} \subseteq 2^{\mathcal{U}}$ of subsets of $\mathcal{U}$ with $\bigcup \mathcal{S} = \mathcal{U}$. The goal is to find a set $\mathcal{C} \subseteq \mathcal{S}$ with the smallest number of sets possible, i.e. with $|\mathcal{C}|$ as small as possible, such that $\mathcal{C}$ covers $\mathcal{U}$, i.e. such that $\mathcal{U} = \bigcup \mathcal{C}$. The set covering optimization problem is NP-hard and the corresponding decision problem is NP-complete. However, there exists an efficient greedy approximation algorithm for the set covering optimization problem which achieves an approximation ratio $\leqslant \ln(|\mathcal{U}|) + 1$, see [6]. We will exploit this algorithm in Section 2.5.

From the geometric insight provided above, we can see that $I = A_{\mathcal{F}} \circ B_{\mathcal{F}}$ means that I is covered by the rectangles corresponding to $\langle C_l, D_l \rangle$'s from $\mathcal{F}$. Consequently, given I and the associated $\mathcal{B}(X, Y, I)$, which as we mentioned can be efficiently computed form I, the problem of finding a decomposition $I = A_{\mathcal{F}} \circ B_{\mathcal{F}}$ is reducible to the set covering problem by putting $\mathcal{U} = \{\langle i, j \rangle \mid I_{ij} = 1\}$ and $\mathcal{S} = \{C \times D \mid \langle C, D \rangle \in \mathcal{B}(X, Y, I)\}$. That is, the set $\mathcal{U}$ to be covered is the set of all pairs for which the corresponding entry I_{ij} is 1, and the set $\mathcal{S}$ of sets which can be used for covering $\mathcal{U}$ is the set of "rectangular sets" of positions corresponding to formal concepts $\langle C, D \rangle \in \mathcal{B}(X, Y, I)$.

2.4.2. Complexity of decomposition

The reduction described in the previous paragraph does not answer the question of the complexity of the factorization problem. In this paragraph we provide a simple argument showing that the problem is NP-hard and its decision version, i.e. decide for given I and positive integer k whether there exists a decomposition $I = A \circ B$ with the inner dimension k, is NP-complete. Note that this result is presented in a slightly different terms in [23], see also [24]. Namely, [23] deals with covering by submatrices of I which contain 1s, rather than with decompositions of I. But such coverings correspond uniquely to decompositions as shown in this paper. To show the reduction, let us first recall the set basis problem, the decision version of which is the following. 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 $C = \{C_1, \ldots, C_k\}$ of subsets $C_l \subseteq \{1, \ldots, m\}$ such that for every S_i there is a subset $D_i \subseteq \{C_1, \ldots, C_k\}$ for which $\bigcup D_i = S_i$ (i.e., the union of all sets from D_i is equal to S_i)? In the corresponding optimization problem, given S, the problem is to find the least C satisfying the above conditions. The set basis decision problem is NP-complete and the corresponding optimization problem is NP-hard [28]. The set basis problem is easily reducible to the problem of factorization: Given S, define an $n \times m$ binary matrix I by $I_{ij} = 1$ if and only if $j \in S_i$. One can check that $I = A \circ B$ for $n \times k$ and $k \times m$ binary matrices A and B if and only if C_l ($l = 1, \ldots, k$) and D_i, defined by $j \in C_l$ iff $B_{lj} = 1$ and $C_l \in D_i$ iff $A_{il} = 1$, represent a solution to the set basis problem given by S. Therefore, the problem to find a factorization $I = A \circ B$ with k as small as possible is NP-hard, and the corresponding decision problem is NP-complete:

Theorem 7. *The problem to find a decomposition $I = A \circ B$ of an $n \times m$ binary matrix I into an $n \times k$ binary matrix A and a $k \times m$ binary matrix B with k as small as possible is NP-hard and the corresponding decision problem is NP-complete.*

Note that this result is also reported in [22,31] where the authors were apparently not familiar with [23,24]. Therefore, unless $P = NP$, there is no polynomial-time algorithm which solves the factorization problem.

2.5. Greedy approximation algorithms

Let us now turn to algorithms. We propose two approximation algorithms for the solution of the problem of decomposition of a binary matrix I. Due to Theorem 2, we look for a smallest set $\mathcal{F}$ of factor concepts of I, i.e. for a smallest set $\mathcal{F} \subseteq \mathcal{B}(X, Y, I)$ of formal concepts of I for which $I = A_{\mathcal{F}} \circ B_{\mathcal{F}}$.

Algorithm 1

INPUT: I (Boolean matrix)
OUTPUT: $\mathcal{F}$ (set of factor concepts)

set $\mathcal{S}$ ***to*** $\mathcal{B}(X, Y, I)$
set $\mathcal{U}$ ***to*** $\{\langle i, j\rangle \mid I_{ij} = 1\}$
set $\mathcal{F}$ ***to*** $\emptyset$
for each $\langle C, D\rangle \in \mathcal{S}$:
  if $\langle C, D\rangle \in \mathcal{O}(X, Y, I) \cap \mathcal{A}(X, Y, I)$:
    add $\langle C, D\rangle$ ***to*** $\mathcal{F}$
    remove $\langle C, D\rangle$ ***from*** $\mathcal{S}$
    for each $\langle i, j\rangle \in C \times D$:
      remove $\langle i, j\rangle$ ***from*** $\mathcal{U}$
while $\mathcal{U} \neq \emptyset$:
  do select $\langle C, D\rangle \in \mathcal{S}$ ***that maximizes*** $(C \times D) \cap \mathcal{U}$:
    add $\langle C, D\rangle$ ***to*** $\mathcal{F}$
    remove $\langle C, D\rangle$ ***from*** $\mathcal{S}$
    for each $\langle i, j\rangle \in C \times D$:
      remove $\langle i, j\rangle$ ***from*** $\mathcal{U}$
return $\mathcal{F}$

2.5.1. First algorithm: Algorithm 1

The above-mentioned reduction of the problem of finding a small set $\mathcal{F}$ of factor concepts to the set covering optimization problem enables us to utilize a greedy approximation algorithm for the set covering optimization problem described in [6] which we mentioned in Section 2.4. Theorem 3 enables us to speed such algorithm up by inserting the mandatory factor concepts in the beginning. As a result, we get Algorithm 1, our first algorithm.

Example 4. Algorithm 1 is based on selecting formal concepts which have the maximal overlap with $\mathcal{U}$. To demonstrate Algorithm 1, consider the following binary matrix:

$$\begin{pmatrix} 1 & 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 1 & 0 & 1 \end{pmatrix}.$$

The matrix has 16 non-zero entries. We need to cover them with the smallest number of formal concepts (rectangles). In the first step, Algorithm 1 selects the concept $\langle\{1, 4, 5\}, \{3, 6\}\rangle$ whose "value" given by the size of $\{1, 4, 5\} \times \{3, 6\}$ is 6. The rectangle corresponding to $\langle\{1, 4, 5\}, \{3, 6\}\rangle$ is indicated by bold 1s in the following matrix:

$$\begin{pmatrix} 1 & 0 & \mathbf{1} & 0 & 1 & \mathbf{1} \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 1 & 1 & 1 \\ 0 & 0 & \mathbf{1} & 0 & 0 & \mathbf{1} \\ 0 & 1 & \mathbf{1} & 1 & 0 & \mathbf{1} \end{pmatrix}.$$

In the next step, the algorithm removes the bold 1s, i.e. we get a new matrix:

$$\begin{pmatrix} 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 \end{pmatrix}.$$

We now repeat the procedure with this new matrix. The next concept $\langle C, D\rangle$ which maximizes $(C \times D) \cap \mathcal{U}$ is $\langle\{3, 5\}, \{2, 4, 6\}\rangle$. The "value" of this concept is 5 because the overlap of the rectangle $\{3, 5\} \times \{2, 4, 6\}$ with the remaining 1s in the matrix contains five 1s. After removing the rectangle from the matrix, we have:

$$\begin{pmatrix} 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}.$$

Then, the algorithm selects $\langle\{1, 3\}, \{1, 5, 6\}\rangle$ and removes four 1s from the matrix. Finally, it chooses $\langle\{1, 2, 4, 5\}, \{3\}\rangle$ and removes the remaining 1. Then, the algorithm stops and returns

$$\mathcal{F} = \big\{\big\langle\{1, 4, 5\}, \{3, 6\}\big\rangle, \big\langle\{3, 5\}, \{2, 4, 6\}\big\rangle, \big\langle\{1, 3\}, \{1, 5, 6\}\big\rangle, \big\langle\{1, 2, 4, 5\}, \{3\}\big\rangle\big\}.$$

This gives the following factorization $A_{\mathcal{F}} \circ B_{\mathcal{F}} = I$:

$$\begin{pmatrix} 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 \\ 1 & 1 & 0 & 1 \end{pmatrix} \circ \begin{pmatrix} 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 0 & 0 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 1 & 0 & 1 \end{pmatrix}.$$

Algorithm 2

INPUT: I (Boolean matrix)
OUTPUT: $\mathcal{F}$ (set of factor concepts)

set $\mathcal{U}$ ***to*** $\{\langle i, j\rangle \mid I_{ij} = 1\}$
set $\mathcal{F}$ ***to*** $\emptyset$
while $\mathcal{U} \neq \emptyset$:
  set D ***to*** $\emptyset$
  set V ***to*** 0
  while there is $j \notin D$ ***such that*** $|D \oplus j| > V$:
    do select $j \notin D$ ***that maximizes*** $D \oplus j$:
      set D ***to*** $(D \cup \{j\})^{\downarrow\uparrow}$
      set V ***to*** $|(D^{\downarrow} \times D) \cap \mathcal{U}|$
  set C ***to*** $D^{\downarrow}$
  add $\langle C, D\rangle$ ***to*** $\mathcal{F}$
  for each $\langle i, j\rangle \in C \times D$:
    remove $\langle i, j\rangle$ ***from*** $\mathcal{U}$
return $\mathcal{F}$

2.5.2. Second algorithm: Algorithm 2

A crucial part of Algorithm 1 is the selection of a concept $\langle C, D\rangle \in \mathcal{S}$ which maximizes $(C \times D) \cap \mathcal{U}$. A straightforward way to make the selection is to compute all the formal concepts first and then, during each iteration, select the one with the largest $(C \times D) \cap \mathcal{U}$. This way, which is used in Algorithm 1, produces good results (the number of factors is usually near the optimum). For large data sets, it is however not efficient. Namely, even though formal concepts provide us with the space of optimal factors, their number can still be large. This impairs speed and memory efficiency.

We now modify Algorithm 1 as follows. Instead of going through all formal concepts, which are the candidates for factor concepts, we construct the factor concepts by adding sequentially "promising columns." This idea is based on the fact that, for each formal concept $\langle C, D\rangle$, D can be expressed as $D = \bigcup_{y \in D} \{y\}^{\downarrow\uparrow}$. We also use the observation that if $y \notin D$ then $\langle (D \cup \{y\})^{\downarrow}, (D \cup \{y\})^{\downarrow\uparrow}\rangle$ is a formal concept with $D \subset (D \cup \{y\})^{\downarrow\uparrow}$. Therefore, we may construct any concept by adding sequentially $\{y\}^{\downarrow\uparrow}$ to the empty set of attributes. In our case, the greedy approach makes us to select $y \in Y$ which maximizes

$$D \oplus y = \big(\big(D \cup \{y\}\big)^{\downarrow} \times \big(D \cup \{y\}\big)^{\downarrow\uparrow}\big) \cap \mathcal{U}. \tag{12}$$

Hence, we modify Algorithm 1 in the following way. Instead of going through all possible concepts, we just go through columns which maximize the value of the factor being constructed. This way we get Algorithm 2, our second algorithm.

Example 5. In general, Algorithms 1 and 2 produce different results. For instance, if we take the binary matrix from Example 4, Algorithm 2 will proceed as follows. First, it selects $1 \in Y$ because $|\emptyset \oplus 1| = 6$, which is the maximum value. Since, $\{1\}^{\downarrow} = \{1, 3\}$ and $\{1\}^{\downarrow\uparrow} = \{1, 5, 6\}$, the first factor concept selected is $\langle \{1, 3\}, \{1, 5, 6\}\rangle$. No further attribute is added to $\{1, 5, 6\}$ because it would not increase the number of 1s covered by this factor. After removing the corresponding rectangle, we obtain matrix

$$\begin{pmatrix} 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 1 & 0 & 1 \end{pmatrix}.$$

In the next step, attribute 2 is selected and the induced concept $\langle \{2\}^{\downarrow}, \{2\}^{\downarrow\uparrow}\rangle = \langle \{3, 5\}, \{2, 4, 6\}\rangle$ is the second factor concept. The process continues with attributes 3 and 6 so that we finally obtain

$$\mathcal{F} = \big\{\big\langle \{1, 3\}, \{1, 5, 6\}\big\rangle, \big\langle \{3, 5\}, \{2, 4, 6\}\big\rangle, \big\langle \{1, 2, 4, 5\}, \{3\}\big\rangle, \big\langle \{1, 3, 4, 5\}, \{6\}\big\rangle\big\},$$

inducing the following factorization $A_{\mathcal{F}} \circ B_{\mathcal{F}} = I$:

$$\begin{pmatrix} 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 \\ 0 & 1 & 1 & 1 \end{pmatrix} \circ \begin{pmatrix} 1 & 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 1 & 0 & 1 \end{pmatrix}.$$

2.5.3. Algorithm 1 and Algorithm 2: experiments and comparisons

In the previous two examples, we have seen that the algorithms produce different sets of factor concepts in general. Moreover, the algorithms may produce different numbers of factor concepts for the same binary matrix. We have performed experiments with randomly generated datasets and large datasets from the UCI Machine Learning Repository [1] to compare behavior and performance of the algorithms.

First, we observed how close is the number of factors found by Algorithm 1 and Algorithm 2 to the actual least number of factors. We were generating 20×20 binary matrices with 50% of 0s and 1s. Then, we were determining the optimal

Table 5
Approximation of optimal factorizations.

Optimal number of factors	Algorithm 1 number of factors	Algorithm 2 number of factors
5	5.313 ± 0.520	5.255 ± 0.473
6	6.545 ± 0.683	6.443 ± 0.666
7	7.785 ± 0.832	7.672 ± 0.781
8	9.182 ± 1.032	9.139 ± 1.097
9	10.542 ± 1.150	10.416 ± 1.107
10	11.718 ± 1.210	11.641 ± 1.213
11	12.775 ± 1.259	12.740 ± 1.290
12	13.800 ± 1.226	13.751 ± 1.229
13	14.652 ± 1.235	14.623 ± 1.306
14	15.284 ± 1.124	15.401 ± 1.162
15	16.068 ± 1.054	16.056 ± 1.011

Table 6
Factorization of randomly generated matrices.

Density % of 1s	Algorithm 1 number of factors	Algorithm 2 number of factors
94%	6.740 ± 0.821	6.710 ± 0.836
92%	7.026 ± 0.894	7.008 ± 0.861
90%	7.538 ± 0.889	7.532 ± 0.903
88%	8.278 ± 0.935	8.338 ± 1.005
85%	9.478 ± 1.012	9.592 ± 1.106
75%	11.620 ± 1.091	11.844 ± 1.141
50%	14.651 ± 0.529	14.791 ± 0.412
25%	14.590 ± 0.645	14.556 ± 0.649
15%	13.452 ± 1.095	13.374 ± 1.127
12%	11.893 ± 1.471	11.753 ± 1.462
10%	10.914 ± 1.685	10.782 ± 1.632
8%	9.876 ± 1.672	9.744 ± 1.633
7%	8.912 ± 1.683	8.810 ± 1.651

number of factors using a brute-force algorithm and compared the number with the number of factors which were found by Algorithm 1 and Algorithm 2. The results are shown in Table 5. Rows of Table 5 correspond to the optimal numbers of factors. The second column of Table 5 shows the average number of factors determined by Algorithm 1. The numbers are written in the form of "average value $\pm$ standard deviation." The last column presents results for Algorithm 2. We can see that the performance of both of the algorithms is practically the same. The average numbers of factors are close to the optimum value and, in most cases, the optimum lies within (or close to) one standard deviation of the average value. Note that the total number of matrices from which we obtained the results in Table 5 was 80,000.

The next example demonstrates factorization of binary matrices with varying "sparseness"/"density." The results of this experiment correspond to the common intuition that "sparse" and "dense" matrices can be factorized with small number of factors whereas matrices with approximately equal numbers of 1s and 0s are not that good for factorization, cf. [8]. The values in Table 6, show results of factorization of 15×15 matrices of various densities. By density we mean the percentage of 1s which appear in a matrix. Again, both of the algorithms are comparable in terms of the number of factor concepts they produce. This time, the rows contain average numbers of factor concepts found in matrices of a given density. We can see from Table 6 that for randomly generated matrices with average density, the number of computed factors is nearly the same as the number of the original attributes. On the other hand, sparse matrices can usually be factorized by a considerably smaller number of factors.

We now turn our attention to performance of the algorithms. Algorithm 1 can deliver better factorizations than Algorithm 2 because it selects factor concepts form the system of all formal concepts associated to the binary matrix I. When the binary matrix is large, Algorithm 2 is much faster than Algorithm 1. For instance, we have tested the computing time with several data sets from the UCI Machine Learning Repository [1]. For the well-known MUSHROOM data set, which consists of 8124 objects and 119 attributes, there are $238,710$ associated formal concepts. During its computation, Algorithm 1 first computes all the concepts, stores them and then iterates over the $238,710$ concepts multiple times. On the contrary, Algorithm 2 just goes over the 119 attributes and it does not precompute all the concepts. Table 7 compares efficiency of both the algorithms on the MUSHROOM data set. The algorithms were implemented in ANSI C and executed on an Intel Xeon 4 CPU 3.20 GHz machine with 1 GB RAM. Algorithm 2 outperforms Algorithm 1 both in terms of memory consumption and the time needed to find the factorization. Notice the significant speed-up which is due to not computing and iterating multiple times the set of all formal concepts associated to the input data.

Table 7
MUSHROOM factorization performance.

	Algorithm 1	Algorithm 2
Time	18 min, 5.66 s	12.47 s
Memory	97 MB RAM	2 MB RAM

2.6. Approximate factorization

In case of large matrices I, it is particularly appealing to look for approximate decompositions of I. That is, we look for A and B such that I is approximately equal to $A \circ B$. We know from the discussion above that for every set $\mathcal{F} \subseteq \mathcal{B}(X, Y, I)$ of formal concepts, $A_{\mathcal{F}} \circ B_{\mathcal{F}}$ approximates I from below. That is, $A_{\mathcal{F}} \circ B_{\mathcal{F}} \leqslant I$, i.e. $(A_{\mathcal{F}} \circ B_{\mathcal{F}})_{ij} \leqslant I_{ij}$ for every row i and column j. Furthermore, by adding further concepts to $\mathcal{F}$, we tighten the approximation. That is, for $\mathcal{F} \subseteq \mathcal{F}'$ we have $A_{\mathcal{F}} \circ B_{\mathcal{F}} \leqslant A_{\mathcal{F}'} \circ B_{\mathcal{F}'} \leqslant I$. Intuitively, while exact factorization may require a large number of factors, a considerably smaller number of factors may account for a large portion of the data set. That is, while the requirement of I being equal to $A_{\mathcal{F}} \circ B_{\mathcal{F}}$ may imply the need for a large $\mathcal{F}$, a considerably smaller $\mathcal{F}$ may be enough to meet a weaker requirement of I being approximately equal to $A_{\mathcal{F}} \circ B_{\mathcal{F}}$. From the perspective of the results presented in this paper, solutions to the approximate factorization problem can be looked for by a slight modification of Algorithm 1 and Algorithm 2. Recall that the algorithms finish the computation if each 1 from the input table is covered by at least one factor. We can modify the halting condition of the algorithm so that the algorithm

- stops if the number of factors exceeds threshold n; or
- stops if the factors cover "almost all 1s" of the input matrix I.

In either case, we obtain a set $\mathcal{F}$ of factor concepts for which $A_{\mathcal{F}} \circ B_{\mathcal{F}} \leqslant I$. Due to this fact, closeness of $A_{\mathcal{F}} \circ B_{\mathcal{F}}$ to I can be assessed as follows. For I and $\mathcal{F} \subseteq \mathcal{B}(X, Y, I)$, define $A(I, \mathcal{F})$ by

$$A(I, \mathcal{F}) = \frac{\text{Area}(\mathcal{F})}{\text{Area}(\mathcal{B}(X, Y, I))},$$

where for $\mathcal{G} \subseteq \mathcal{B}(X, Y, I)$ we put

$$\text{Area}(\mathcal{G}) = \big|\big\{\langle i, j\rangle \;\big|\; (A_{\mathcal{G}} \circ B_{\mathcal{G}})_{ij} = 1\big\}\big|.$$

Hence, $\text{Area}(\mathcal{G})$ is the number of 1s in the matrix I covered by the set $\mathcal{G}$ of formal concepts. As a consequence, $\text{Area}(\mathcal{B}(X, Y, I))$ is the number of 1s in the input matrix. We call $A(I, \mathcal{F})$ a degree of approximation of I by $\mathcal{F}$. Clearly, $100 \cdot A(I, \mathcal{F})$ is the percentage of 1s in the input matrix I which are covered by factors from $\mathcal{F}$. Observe that $A(I, \mathcal{F}) \in [0, 1]$ and in addition, $A(I, \mathcal{F}) = 1$ iff $I = A_{\mathcal{F}} \circ B_{\mathcal{F}}$, i.e., iff the factors completely explain the data. Note that looking for a set $\mathcal{F}$ of approximate factors, i.e. such with a sufficiently high $A(I, \mathcal{F})$, is particularly appealing for larger matrices I. Experiments on approximate factorization are presented in Section 3.

3. Experiments

In this section, we present several examples of factorization using Algorithm 1 and Algorithm 2.

Example 6. The first experiment concerns analysis of factors which determine attributes of European Union countries. We used data from the Rank Order pages of the CIA World Factbook 2006.[1] The data describe socio-economic indicators such as "GDP per capita," "energy production/consumption," "population growth," "military expenditures," "industrial production," etc. The indicators have numerical values. For instance, "GDP per capita" is measured in thousands USD. Using binarization, we transformed the data to a binary matrix consisting of 27 rows (EU countries) and 235 columns (yes/no attributes). For the binarization, we proceed as follows. For every socio-economic indicator y, we first compute its basic statistical characteristics (quartiles, sample mean, and standard deviation) of the corresponding sample given by the values on 27 EU countries. Then, we introduce five Boolean (yes/no) attributes based on these characteristics. In detail, for v being a numerical value of indicator y, we consider five Boolean attributes with values 1 (or 0) if the following conditions are (or are not) satisfied:

(a) $v < q_1$ (v is smaller than the first quartile),
(b) $q_1 \leqslant v$ and $v \leqslant q_3$ (v lies in the inter-quartile range),
(c) $v > q_3$ (v is greater than the third quartile),

[1] https://www.cia.gov/library/publications/download/.

(d) $v \in [\overline{y} - s, \overline{y} + s]$ (v falls within one standard deviation of the mean),
(e) $v \in [\overline{y} - 2s, \overline{y} + 2s]$ (v falls within two standard deviations of the mean).

The resulting binary matrix serves as testing data for the above-described method of factor analysis. Let us also mention that replacing the original numerical values by the above-mentioned Boolean attributes may help users understand the data. Often, people are not interested in exact numerical values of socio-economic indicators. Rather, they are (intuitively) working with notions like "value is close to the average" which is captured the Boolean attribute described by (d). This is particularly true if one needs to draw conclusions regarding performance of EU countries, policy, etc.

The total number of formal concepts present in the 27×235 Boolean matrix is 299,982, i.e. with X denoting the 27 EU countries, Y denoting the 235 Boolean attributes, and I denoting the corresponding Boolean matrix, $|\mathcal{B}(X, Y, I)| = 299{,}982$. According to Theorem 2, $\mathcal{B}(X, Y, I)$ can be considered a space of optimal potential factors. From the formal concepts of $\mathcal{B}(X, Y, I)$, Algorithm 1 computes a set $\mathcal{F} \subseteq \mathcal{B}(X, Y, I)$ of factor concepts which explain the data in I, i.e. a set $\mathcal{F}$ for which $I = A_{\mathcal{F}} \circ B_{\mathcal{F}}$. Running Algorithm 1, we obtained a set $\mathcal{F}$ of 53 factor concepts, i.e. there exists $\mathcal{F}$ with

$$|\mathcal{F}| = 53$$

for which $I = A_{\mathcal{F}} \circ B_{\mathcal{F}}$. This means that the 27×235 matrix I can been decomposed into a Boolean product of a 27×53 binary matrix $A_{\mathcal{F}}$ representing a relationship between EU countries and the factors and a 53×235 binary matrix $B_{\mathcal{F}}$ representing a relationship between the factors and the original attributes. Note that the relationship between EU countries and the original attributes (described by I) can be retrieved from the relationship between EU countries and factors (described by $A_{\mathcal{F}}$) and the relationship between factors and original attributes (described by $B_{\mathcal{F}}$) according to: Country x has attribute y iff there is a factor f such that f applies to x and y is one of the particular manifestations of f. The transformations between the 235-dimensional space of attributes and the 53-dimensional space of factors are accomplished by mappings g and h defined by (2) and (3).

Due to space limitations, we cannot list all the attributes and factors. Note however, that the factors have a natural interpretation providing us with an insight into the data. For instance, the largest factor applies to all the EU countries except for France, Germany, Italy, Netherlands, Spain, and UK. The manifestations of this factor in terms of the original attributes are attribute (d) for "GDP per capita" and attribute (d) for "oil import in bbl/day." Therefore, the verbal description of this factor ("GDP within two standard deviations of the mean and oil import within two standard deviations of the mean") has a clear meaning. Note that this factor does not apply to France, Germany, Italy, and UK due to their high GDP per capita, and does not apply to Netherlands and Spain due to their high oil imports.

Example 7. Intuitively, if we factor-analyze a related subgroup of EU countries in Example 6, we expect that their 235 attributes are reducible to still a smaller number of factors. For instance, if we focus on the ten countries which joined EU in 2004, there is a set $\mathcal{F}$ of 19 factors, i.e. for the 10×235 Boolean matrix I we have $I = A_{\mathcal{F}} \circ B_{\mathcal{F}}$ where $A_{\mathcal{F}}$ is a 10×19 Boolean matrix and $B_{\mathcal{F}}$ is a 19×235 Boolean matrix. Therefore, the 235 Boolean attributes are explained using just 19 factors ($\frac{1}{12}$ reduction). The larger reduction in this case can be seen consistent with the common knowledge that the new EU countries have similar socio-economic characteristics.

Example 8. This example further illustrates the comparison of Algorithm 1 and Algorithm 2 on the MUSHROOM data set. Both algorithms compute the same number of factor concepts but the sets of factor concepts are different. Nevertheless, both the corresponding factors from the different collections cover approximately the same number of 1s in the matrix. Namely, if we form a sequence $\langle x_1, y_1 \rangle, \ldots, \langle x_k, y_k \rangle$ where x_i and y_i are the numbers of 1's covered by ith factor produced by Algorithm 1 and Algorithm 2, respectively, and if we plot the points $\langle x_i, y_i \rangle$ in a two-dimensional plot (a qq-plot), we can see that the points are gathered close to the diagonal. Fig. 3 shows this situation (the plot uses axes with a logarithmic scale).

Example 9. The last example illustrates approximate factorization, cf. Section 2.6, of the MUSHROOM data set. Recall that the MUSHROOM data set contains 8124 objects and 119 attributes and that the corresponding binary matrix I contains 238,710 formal concepts, i.e. $|\mathcal{B}(X, Y, I)| = 238{,}710$. Experiments have shown that most of the information contained in the data set can be represented through a relatively small number of factor concepts, i.e. most 1s in I are covered by a relatively small number of concepts from $\mathcal{B}(X, Y, I)$. The results can be seen from the graphs shown in Fig. 4. The left and the right part describe the results for Algorithm 1 and Algorithm 2, respectively. Notice the similarity of the graphs (cf. the previous example). The graph shows a relationship between the number of factor concepts and the degree of approximation of the original data set. We can see that even if we take a relatively small number of factor concepts, we achieve a high degree of approximation. For instance, if we take first 6 factor concepts returned by Algorithm 1, we get $A(I, \mathcal{F}) \cdot 100\% = 51.89\%$. This can be interpreted as *more than half the data contained in the MUSHROOM data set can be explained by six factors.* The growth of the degree of approximation is rapid for first 10 factor concepts. For instance, first 2, 3, and 4 factors cover 26.03%, 34.35%, and 41.29% of the incidence data, respectively. If we use 60 factors, 95% of the data is covered.

Note that the need for approximate factorization of binary data is quite natural. In case of larger data sets, users are usually interested in major dependencies and structures hidden in the data. The approach to approximate Boolean factor

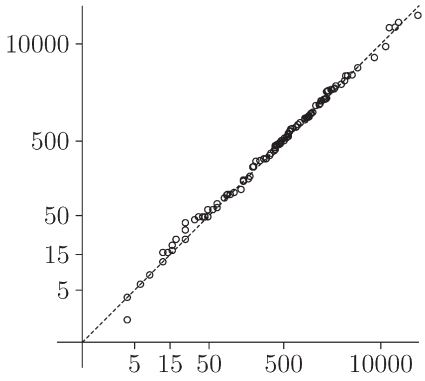

Fig. 3. Corresponding factors cover approximately the same number of 1s.

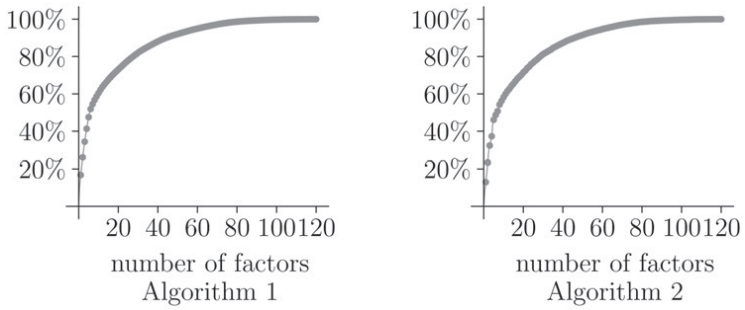

Fig. 4. Relationship between the number of factors and the approximation degree.

analysis can help achieve this goal: a user can specify the approximation degree $a\%$ as an additional constraint, meaning that he is interested in factors which explain at least $a\%$ of the data. Investigation of further variations of the problem of approximate factorization and developments of methods is necessary.

4. Conclusions and future research

We presented a novel approach to factorization of binary data. The approach is based on a theorem presented in this paper, describing a space of optimal factors for decomposition of binary matrices. We presented greedy approximation algorithms for finding a smallest set of factors in the space of optimal factors. The first one is based on an approximation algorithm for the set covering optimization problem, the second one is its modification which avoids the need to compute the set of all formal concepts. We presented examples demonstrating applications of Boolean factor analysis and performance of the algorithms. The following is a list of problems left for future research:

- Criteria for good factors: Investigation of further criteria for discovery of factors including psychologically motivated criteria. It might be desirable to look for $\mathcal{F}$ such that the numbers of attributes in the formal concepts' intents are restricted in a suitable way. For instance, one might require all formal concepts from $\mathcal{F}$ have approximately the same number of attributes, i.e. the level of generality of all factors be approximately the same. Another criterion might be a suitably defined independence of factors.
- Further heuristics for improvement of Algorithm 1 and Algorithm 2. The heuristics are supposed to be drawn from further theoretical insights.
- Approximate factorization: Can we combine the approximation "from below" provided by factor concepts with "approximation from above"? The goal is to stop looking for additional factor concepts once the new factor concepts do not contribute much to covering the yet uncovered part of I. After stopping, try to approximate I with a small number of rectangles from above. Such rectangles are not formal concepts and their inclusion leads to inclusion of 1s for positions where there are 0s in I.
- Extension to graded incidence data: Work is in progress on extending the methods from binary matrices to matrices containing more general entries, such as numbers from the unit interval $[0, 1]$, instead of just 0 and 1, expressing degrees to which attributes apply to objects. This is possible using an extension of formal concept analysis to the setting of fuzzy logic, see e.g. [3,4].
- Relationships between BFA using formal concepts and approaches via associative networks, see [2,25] for hints.
- In general, applications of the dimensionality reduction aspect of decompositions of binary matrices in the fields of knowledge discovery and machine learning.

References

[1] A. Asuncion, D.J. Newman, UCI machine learning repository, University of California, Irvine, School of Information and Computer Sciences, http://www.ics.uci.edu/~mlearn/MLRepository.html, 2007.
[2] R. Belohlavek, Representation of concept lattices by bidirectional associative memories, Neural Comput. 12 (10) (2000) 2279–2290.
[3] R. Belohlavek, Concept lattices and order in fuzzy logic, Ann. Pure Appl. Logic 128 (1–3) (2004) 277–298.
[4] R. Belohlavek, J. Dvorak, J. Outrata, Fast factorization by similarity in formal concept analysis of data with fuzzy attributes, J. Comput. System Sci. 73 (6) (2007) 1012–1022.
[5] C. Carpineto, G. Romano, Concept Data Analysis. Theory and Applications, J. Wiley, 2004.
[6] T.H. Cormen, C.E. Leiserson, R.L. Rivest, C. Stein, Introduction to Algorithms, 2nd ed., MIT Press, 2001.
[7] J.H. Correia, G. Stumme, R. Wille, U. Wille, Conceptual knowledge discovery—A human-centered approach, Appl. Artificial Intelligence 17 (3) (2003) 281–302.
[8] 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, May.
[9] B. Ganter, R. Wille, Formal Concept Analysis. Mathematical Foundations, Springer, Berlin, 1999.
[10] F. Geerts, B. Goethals, T. Mielikäinen, Tiling databases, in: Proc. DS 2004, in: Lecture Notes in Comput. Sci., vol. 3245, 2004, pp. 278–289.
[11] G. Georgescu, A. Popescu, Non-dual fuzzy connections, Arch. Math. Logic 43 (2004) 1009–1039.
[12] G.A. Golub, C.F. Van Loan, Matrix Computations, 3rd ed., The Johns Hopkins University Press, 1995.
[13] H.H. Harman, Modern Factor Analysis, 2nd ed., The Univ. of Chicago Press, Chicago, 1970.
[14] A. Keprt, V. Snášel, Binary factor analysis with help of formal concepts, in: Proc. CLA 2004, Ostrava, Czech Republic, 2004, ISBN 80-248-0597-9, pp. 90–101.
[15] K.H. Kim, Boolean Matrix Theory and Applications, M. Dekker, 1982.
[16] W. Kneale, M. Kneale, The Development of Logic, Clarendon Press, Oxford, 1984.
[17] S. Kuznetsov, S. Obiedkov, Comparing performance of algorithms for generating concept lattices, J. Exp. Theor. Artificial Intelligence 14 (2–3) (2002) 189–216.
[18] D. Lee, H. Seung, Learning parts of objects by non-negative matrix factorization, Nature 401 (1999) 788–791.
[19] J.D. Leeuw, Principal component analysis of binary data. Application to roll-call analysis [online], available at: http://gifi.stat.ucla.edu, 2003.
[20] M. Maddouri, Towards a machine learning approach based on incremental concept formation, Intelligent Data Analysis 6 (2002) 1–15.
[21] R.P. McDonald, Factor Analysis and Related Methods, Lawrence Erlbaum Associates, Inc., 1985.
[22] P. Miettinen, T. Mielikäinen, A. Gionis, G. Das, H. Mannila, The discrete basis problem, in: Proc. PKDD 2006, in: Lecture Notes in Artificial Intelligence, vol. 4213, 2006, pp. 335–346.
[23] D.S. Nau, Specificity covering: Immunological and other applications, computational complexity and other mathematical properties, and a computer program, A.M. Thesis, Technical Report CS-1976-7, Computer Sci. Dept., Duke Univ., Durham, NC, 1976.
[24] 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.
[25] R.K. Rajapakse, M. Denham, Fast access to concepts in concept lattices via bidirectional associative memories, Neural Comput. 17 (10) (2005) 2291–2300.
[26] Sajama, A. Orlitsky, Semi-parametric exponential family PC, in: L.K. Saul, et al. (Eds.), Advances in Neural Information Processing Systems 17, MIT Press, Cambridge, MA, 2005, http://books.nips.cc/papers/files/nips17/NIPS2004_0152.pdf.
[27] 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.
[28] L.J. Stockmeyer, The set basis problem is NP-complete, IBM Research Report RC5431, Yorktown Heights, NY, 1975.
[29] F. Tang, H. Tao, Binary principal component analysis, in: Proc. British Machine Vision Conference 2006, 2006, pp. 377–386.
[30] 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.
[31] J. Vaidya, V. Atluri, Q. Guo, The role mining problem: Finding a minimal descriptive set of roles, in: ACM Symposium on Access Control Models and Technologies, June, 2007, pp. 175–184.
[32] R. Wille, Restructuring lattice theory: An approach based on hierarchies of concepts, in: I. Rival (Ed.), Ordered Sets, Reidel, Dordrecht/Boston, 1982, pp. 445–470.
[33] Z. Zivkovic, J. Verbeek, Transformation invariant component analysis for binary images, in: 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, vol. 1, CVPR'06, pp. 254–259.