Formal concepts as optimal factors in Boolean factor analysis: implications and experiments*

Radim Belohlavek[1,2], Vilem Vychodil[2]

[1] Dept. Systems Science and Industrial Engineering
T. J. Watson School of Engineering and Applied Science
Binghamton University–SUNY, PO Box 6000, Binghamton, NY 13902–6000, USA
rbelohla@binghamton.edu
[2] Dept. Computer Science, Palacky University, Olomouc
Tomkova 40, CZ-779 00 Olomouc, Czech Republic
vilem.vychodil@upol.cz

Abstract. Boolean factor analysis aims at decomposing an objects × attributes Boolean matrix I into a Boolean product of an objects × factors Boolean matrix A and a factors × attributes Boolean matrix B, with the number of factors as small as possible. This paper is a continuation of our previous paper where we proved that formal concepts of I are optimal factors for Boolean factor analysis. In particular, we concentrate on the implications of the proof. Namely, on the fact that finding factors can be reduced to the set covering problem for which there exist efficient approximation algorithms. In this paper, we present the algorithm for finding factors which results this way and present several experiments on factorizing Boolean matrices.

1 Introduction and problem setting

The present paper concerns factor analysis of Boolean data and is a continuation of [4]. In [4], we proved that formal concepts are optimal factors in Boolean factor analysis and outlined some implications of the insight provided by the proof. The aim of this paper is to elaborate on one of those implications. Namely, on the fact that the problem of finding factors in Boolean factor analysis can be reduced to the well-known set covering problem. For the set covering problem, there exists an efficient approximation algorithm. This algorithm can thus be used for finding factors in Boolean factor analysis. Moreover, the algorithm can be sped up due to some specific features of Boolean factor analysis. In this paper, we present the thus resulting algorithm for finding factors in Boolean factor analysis. The main focus of the paper is on presenting examples on on Boolean factor analysis and experiments with the algorithm.

The idea of factor analysis is rooted in Spearman's monumental development of a psychological theory involving a single general factor and a number of specific factors [18]. Today, factor analysis is a well-established branch of statistical

* Supported by grant No. 1ET101370417 of GA AV ČR, by grant No. 201/05/0079 of the Czech Science Foundation, and by institutional support, research plan MSM 6198959214.

data analysis with applications in numerous fields and with support in several software packages, see e.g. [1, 7, 10]. According to Harman [10, p. 4], "The principal concern of factor analysis is the resolution of a set of variables linearly in terms of (usually) a small number of categories or 'factors'. ... A satisfactory solution will yield factors which convey all the essential information of the original set of variables. Thus, the chief aim is to attain scientific parsimony or economy of description."

The problem of factor analysis can be described as follows. Suppose we are given an $n \times m$ matrix I describing relationships between n objects and m variables. Each entry I_{ij} of the objects×variables matrix I contains the value of j-th variable on i-th object. The aim is to find k new variables, called factors, an $n \times k$ matrix A describing a relationship between objects and factors, and a $k \times m$ matrix B describing a relationship between factors and original variables, in such a way that I can be obtained from A and B. In case of linear factor analysis, I is required to be (approximately) equal to the usual matrix product $A \circ B$ of A and B. In addition, one requires that the number k of factors be less than the number m of original variables, attaining thus the dimension reduction of the space in which the objects are described. Every A's entry A_{il}, called factor loading, represents a value of l-th factor on i-th object; every B's entry B_{lj}, called factor score, represents the manifestation of j-th variable on l-th factor.

Several extensions of linear factor analysis have been proposed to deal with data for which linear resolution of the original variables in terms of factors is not appropriate, see e.g. [7, 10]. A particular example of such data is represented by Boolean variables (attributes), called also binary variables (attributes), yes-or-no variables (attributes). In this case, entries of I are 1s and 0s, i.e. $I_{ij} = 1$ or $I_{ij} = 0$ with 1/0 indicating that the i-th object has/does not have the j-th variable (attribute). That is, I is a so-called Boolean matrix. For instance, a patient (object) has or does not have headache (variable, attribute). The question of whether the methods of factor analysis are appropriate for Boolean variables has been discussed since 1940s, see e.g. Section 7 of [13]. It has been argued that common methods of factor analysis, both linear and non-linear are not appropriate to handle Boolean variables, see e.g. [13, 16, 19].

A promising way to reveal factors in Boolean data is provided by Boolean factor analysis (BFA), see e.g. [8, 14, 17]. In BFA, a decomposition of an $n \times m$ matrix I with $I_{ij} \in \{0, 1\}$ 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. Note that a Boolean matrix product $A \circ B$ is defined by

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

where $\bigvee$ denotes maximum. Using Boolean matrices for the objects×factors and the factors×variables relationships, and using Boolean matrix product has the following advantage:

- One does not have to deal with the problem of rounding off real values to 0 and 1 which is the case when using common methods of factor analysis.

– Clear interpretability of factors. Namely, with the Boolean matrix product, $I = A \circ B$ says: "an object i has an 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".

Several methods for BFA can be found in the literature. Perhaps the most advanced one is based on using Hopfield-like associative neural networks, see [8, 17]. In this approach, factors correspond to stable points (attractors) of an associative neural network. [12] presents a different approach which is based on using genetic algorithms for the search of factors. Yet another approach, which served as an inspiration for our paper is presented in [11] where the authors try to exploit methods of formal concept analysis [5, 9] for BFA.

In [4], we presented several results on factorizing a Boolean matrix I using formal concepts associated to I. The main result says that formal concepts are optimal factors in BFA, meaning that for every decomposition of I into $A \circ B$ with k factors, i.e., A and B are $n \times k$ and $k \times m$ Boolean matrices, there exists a decomposition of I which uses formal concepts as factors such that the number of the formal concepts is at most k. In addition, we pointed out in [4] that the problem of finding a smallest set of factors in BFA can be reduced to set covering problem for which there exists an efficient approximation algorithm.

In this paper, we present the thus resulting algorithm for finding factors in BFA. The main aim of this paper is to present examples on BFA and experiments with the algorithm.

2 Formal concepts as optimal factors and the reduction to set covering problem

Recall first basic notions and notation from formal concept analysis (FCA) [5, 9]. Let $X = \{x_1, \ldots, x_n\}$ and $Y = \{y_1, \ldots, y_m\}$ be sets of objects and attributes, and I be a binary relation between X and Y. The triplet $\langle X, Y, I \rangle$ is called a formal context. Using the well-known fact that a binary relation between X and Y can be represented by an $n \times m$ Boolean matrix, we denote both the binary relation and its corresponding Boolean matrix by I. That is, for the entry I_{ij} of (matrix) I we have $I_{ij} = 1$ iff $\langle x_i, y_j \rangle$ belongs to (relation) I and $I_{ij} = 0$ if $\langle x_i, y_j \rangle$ do not belong to (relation) I. A formal concept of $\langle X, Y, I \rangle$ is any pair $\langle C, D \rangle$ of sets $C \subseteq X$ and $D \subseteq Y$ such that $C = D^{\downarrow}$ and $D = C^{\uparrow}$ where

$$D^{\downarrow} = \{x \in X \,|\, \text{for each } y \in D : \langle x, y \rangle \in I\}$$

is the set of all objects sharing all attributes from D, and

$$C^{\uparrow} = \{y \in Y \,|\, \text{for each } x \in C : \langle x, y \rangle \in I\}$$

is the set of all attributes shared by all objects from C. The set of all formal concepts of $\langle X, Y, I \rangle$ is denoted by $\mathcal{B}(X, Y, I)$. Under a natural partial order, $\mathcal{B}(X, Y, I)$ happens to be a complete lattice, so-called concept lattice of $\langle X, Y, I \rangle$ [9, 20]. It is a well-known fact that formal concepts of $\langle X, Y, I \rangle$ are just maximal

rectangles of matrix I which are full of 1s. For instance,

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

is a Boolean matrix representing a formal context with $X = \{x_1, \ldots, x_4\}$, $Y = \{y_1, \ldots, y_5\}$, and relation I for which $\langle x_1, y_1 \rangle \in I$, $\langle x_1, y_2 \rangle \in I$, $\langle x_1, y_3 \rangle \notin I$, etc. The bold **1**s form a maximal rectangle, with rows 1, 2, 3, and columns 1 and 2. Correspondingly therefore, $\langle \{x_1, x_2, x_3\}, \{y_1, y_2\} \rangle$ is a formal concept in $\langle X, Y, I \rangle$. This "geometrical" way of looking at formal concepts proves to quite useful in FCA.

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 associated to I. Denote by $A_\mathcal{F}$ and $B_\mathcal{F}$ the $n \times k$ and $k \times m$ Boolean matrices defined by

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

That is, the l-th column $(A_\mathcal{F})_{\_l}$ of $A_\mathcal{F}$ consists of the characteristic vector of A_l and the l-th row $(B_\mathcal{F})_{l\_}$ of $B_\mathcal{F}$ consists of the characteristic vector of B_l.

We are interested in a decomposition of I into $A_\mathcal{F} \circ B_\mathcal{F}$. If $I = A_\mathcal{F} \circ B_\mathcal{F}$, $\mathcal{F}$ can be seen as a set of factors and we call the formal concepts from $\mathcal{F}$ factor concepts. The l-th factor, i.e. formal concept $\langle A_l, B_l \rangle$, applies to the i-th object if $x_i \in A_l$; the j-th attribute y_j is a manifestation of the l-the factor if $y_j \in B_l$. Note that decomposing I by means of formal concepts has been proposed in [11]. However, the particular way of using formal concepts as factors is not described explicitly in [11].

Two questions then arise. First, does there always exist a $\mathcal{F} \subseteq \mathcal{B}(X, Y, I)$ such that $I = A_\mathcal{F} \circ B_\mathcal{F}$? Second, to what extent is it optimal to consider formal concepts from $\mathcal{B}(X, Y, I)$ as factors?

For the first question, it is a well-known fact of formal concept analysis that if we put $\mathcal{F} = \mathcal{B}(X, Y, I)$ (with any indexing of formal concepts), then $I = A_\mathcal{F} \circ B_\mathcal{F}$. In this sense, factorization using formal concepts is universal, see [4]. Note that putting $\mathcal{F} = \mathcal{B}(X, Y, I)$ is not practically useful because the number $|\mathcal{B}(X, Y, I)|$ of all formal concepts is usually larger than the number $|Y|$ of the original attributes. A better way consists in taking $\mathcal{F} = \mathcal{O}(X, Y, I)$ or $\mathcal{F} = \mathcal{A}(X, Y, I)$ where

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

are the sets of object-concepts and attribute-concepts, respectively. In both of these cases we have $I = A_\mathcal{F} \circ B_\mathcal{F}$ but still, this may not be the optimal decomposition.

The second question is answered by the following theorem which was proven in [4]. The theorem says that for any decomposition of I there is always at least as good a decomposition which uses formal concept as factors.

Theorem 1 (formal concepts are optimal factors). *Let $I = A \circ B$ for $n \times k$ and $k \times m$ Boolean matrices A and B. Then there exists $\mathcal{F} \subseteq \mathcal{B}(X, Y, I)$ with*

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

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

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

The proof is instructive and we therefore summarize its main points. First, observe that $I = A \circ B$ means that I can be written as a $\bigvee$-superposition of rectangles consisting of 1s. For instance, in case of $I = A \circ B$ 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 corresponding 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 I^1, I^2, I^3, I^4, i.e. Boolean matrices whose 1s form rectangles. The l-th Boolean matrix I^l results as a Boolean matrix multiplication of the l-the column of A ($n \times 1$ matrix) and the l-th row of B ($1 \times m$ matrix). Second, each such a rectangle is contained in a maximal rectangle of I, i.e. in a formal concept, and a $\bigvee$-superposition of these maximal rectangles gives I. Denoting therefore the collection of all these formal concepts by $\mathcal{F}$ yields the conclusion. Note that since two distinct rectangles may be contained in a single maximal rectangle, we may have $|\mathcal{F}| < k$.

Using our example, consider formal concepts $\langle A_1, B_1 \rangle = \langle \{x_1, x_2, x_3\}, \{y_1, y_2\} \rangle$, $\langle A_2, B_2 \rangle = \langle \{x_3\}, \{y_1, y_2, y_3, y_4\} \rangle$, $\langle A_3, B_3 \rangle = \langle \{x_2, x_4\}, \{y_1, y_5\} \rangle$, of I. Then, each of the rectangles corresponding to I^ls 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 $I = A_{\mathcal{F}} \circ B_{\mathcal{F}}$. Denoting by $(A_{\mathcal{F}})_{\_l}$ and $(B_{\mathcal{F}})_{l\_}$ the l-th column of $A_{\mathcal{F}}$ and the l-th 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 shows a $\bigvee$-decomposition of I into maximal rectangles. With our example, we have

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

Remark 1. In factor analysis, factors considered to represent general categories, sometimes called concepts, of which the original variables are particular manifestations. The problem of interpretability of factors is a part of the whole process of factor analysis. From this point of view, interpretation of formal concepts as factors in BFA is clear for a user. Namely, the notion of a formal concept results as a mathematical formalization of the notion of a concept as worked out in the

traditional logic. A formal concept $\langle A_l, B_l \rangle$ can be seen as a "unit of thought" consisting of a collection A_l of objects to which it applies (concept's extent) and a collection B_l of attributes to which it applies (concept's intent). Clear interpretability is one of the advantageous features of having formal concepts as factors.

Consider now the following problem we call the BFA Problem [4]:

INPUT: Boolean matrix I,
OUTPUT: smallest $\mathcal{F} \subseteq \mathcal{B}(X, Y, I)$ for which $I = A_{\mathcal{F}} \circ B_{\mathcal{F}}$.

As mentioned in [4], BFA Problem is reducible to the set covering optimization problem, for which we refer to [6]. Recall that 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 fewest sets (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 a difficult problem. It 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 $\leq \ln(|\mathcal{U}|) + 1$, see [6].

The idea of the proof of Theorem 1 allows us to see that $I = A_{\mathcal{F}} \circ B_{\mathcal{F}}$ means that I is covered by the rectangles corresponding to $\langle A_l, B_l \rangle$'s from $\mathcal{F}$. Consequently, the BFA Problem is reducible to the set covering problem by putting $\mathcal{U} = \{\langle x_i, y_j \rangle \,|\, I_{ij} = 1\}$ and $\mathcal{S} = \{C \times D \,|\, \langle C, D \rangle \in \mathcal{B}(X, Y, I)\}$. 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 $\langle C, D \rangle \in \mathcal{B}(X, Y, I)$. The above-mentioned greedy approximation algorithm can therefore be used to find approximately optimal solutions for the BFA Problem. However, the particular nature of the BFA Problem enables us to speed up the algorithm. It is easy to see that if $I = A_{\mathcal{F}} \circ B_{\mathcal{F}}$, then $\mathcal{F}$ needs to contain all formal concepts from $\mathcal{B}(X, Y, I)$ which are both object concepts and attribute concepts, i.e. $\mathcal{O}(X, Y, I) \cap \mathcal{A}(X, Y, I) \subseteq \mathcal{F}$. Therefore, one can include $\mathcal{O}(X, Y, I) \cap \mathcal{A}(X, Y, I)$ in $\mathcal{F}$ right in the beginning. Recall that the set $\mathcal{O}(X, Y, I)$ of all object concepts and the set $\mathcal{A}(X, Y, I)$ of all attribute concepts is defined by

$$\mathcal{O}(X, Y, I) = \{\langle \{x\}^{\uparrow\downarrow}, \{x\}^{\uparrow} \rangle\} \text{ and } \mathcal{A}(X, Y, I) = \{\langle \{y\}^{\downarrow}, \{y\}^{\downarrow\uparrow} \rangle\}.$$

The resulting algorithm for computing the factors follows:

Algorithm 1 (Compute factor concepts)

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

set $\mathcal{S}$ ***to*** $\mathcal{B}(X, Y, I)$
set $\mathcal{U}$ ***to*** $\{\langle x_i, y_j \rangle \,|\, I_{ij} = 1\}$
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 ⟨C, D⟩ to F
    remove ⟨C, D⟩ from S
    for each ⟨x, y⟩ ∈ C × D:
      remove ⟨x, y⟩ from U
while (U ≠ ∅):
  do select ⟨C, D⟩ ∈ S that maximizes (C × D) ∩ U:
    add ⟨C, D⟩ to F
    remove ⟨C, D⟩ from S
    for each ⟨x, y⟩ ∈ C × D:
      remove ⟨x, y⟩ from U
return F
```

3 Experiments with Boolean factor analysis

In this section, we present experiments on factorization of Boolean matrices. In the experiments, we employed the algorithm described in the end of the previous section.

Experiment 1 The first experiment concerns analysis of factors which determine attributes of European Union countries. We have taken information from the Rank Order pages of the CIA World Factbook 2006[3] and created a Boolean matrix consisting of 27 rows (EU countries) and 141 columns (yes/no attributes). The attributes are scaled versions of the numerical values taken from the Factbook.

The total number of formal concepts present in the matrix is 3963. From this amount of concepts, Algorithm 1 computes only a small number of factor concepts. In particular, we obtained a set $\mathcal{F}$ of 49 factor concepts, i.e. formal concepts for which $I = A_{\mathcal{F}} \circ B_{\mathcal{F}}$. That is, the 27×141 matrix I has been decomposed into a Boolean product of a 27×49 Boolean matrix $A_{\mathcal{F}}$ representing a relationship between EU countries and the factors and a 49×141 Boolean matrix $B_{\mathcal{F}}$ representing a relationship between the factors and the original attributes. The factor concepts can be considered factors explaining completely the original 141 attributes. Note that the original attributes are socio-economic attributes. However, due to the limited scope, we restrict ourselves to listing the numbers of attributes and factors and leave the socio-economic interpretation of the factors to a future work, to be done possibly with an expert economist.

Experiment 2 We have used Algorithm 1 to compute factor concepts from large data sets. In the case of large data sets, it seems to be of interest whether there is a set of factors which approximately explain the data.

Here we present the results for the well-known MUSHROOM data set which can be found at the UCI Machine Learning Repository[4]. The MUSHROOM

[3] https://www.cia.gov/library/publications/download/
[4] http://www.ics.uci.edu/~mlearn/

database is presented as a collection of so-called "item sets", i.e. it is a collection of sets of items. The collection can be transformed into a Boolean matrix with rows corresponding to items sets, attributes corresponding to items and table entries indicating whether an item set given by the row contains an item given by the column. The MUSHROOM database contains 8124 objects and 119 attributes. The corresponding Boolean matrix contains 238710 formal concepts.

Let us thus turn our attention to factor concepts which approximately explain the data. That is, our aim is to find a set $\mathcal{F}$ of factor concepts such that I is 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. Recall that Algorithm 1 finishes its computation if each 1 in from the input table is covered by at least one factor. We might modify the halting condition of the algorithm so that

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

In either case, we obtain a set $\mathcal{F}$ of factor concepts so that $A_{\mathcal{F}} \circ B_{\mathcal{F}} \leq I$. It is desirable to have $A_{\mathcal{F}} \circ B_{\mathcal{F}}$ as close to I as possible while having a reasonable number of factors at the same time. The closeness of $A_{\mathcal{F}} \circ B_{\mathcal{F}}$ will 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

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

for each $\mathcal{G} \subseteq \mathcal{B}(X, Y, I)$. Hence, $\text{Area}(\mathcal{G})$ is the number of 1s in the matrix given by a set of rectangles $\mathcal{G}$. As a consequence, $\text{Area}(\mathcal{B}(X, Y, I))$ is the number of 1s in the input matrix. $A(I, \mathcal{F})$ will be called a degree of approximation of I by $\mathcal{F}$. Furthermore, $A(I, \mathcal{F}) \cdot 100$ is the percentage of 1s in the input matrix I which are covered by factors from $\mathcal{F}$. Clearly, if $\mathcal{F}$ is a set of the exact factor concepts, i.e. $A_{\mathcal{F}} \circ B_{\mathcal{F}} = I$, then $\text{Area}(\mathcal{B}(X, Y, I)) = \text{Area}(\mathcal{F})$ which yields $A(I, \mathcal{F}) = 1$. Observe that $A(I, \mathcal{F}) \in [0, 1]$ and in addition, $A(I, \mathcal{F}) = 1$ iff I equals $A_{\mathcal{F}} \circ B_{\mathcal{F}}$, i.e., iff the factors completely explain the data.

Our experiments with the MUSHROOM data set have shown that most of the information contained in the data set can be expressed through a relatively small number of factor concepts. The results of our experiment can be depicted by a graph shown in Fig. 1. The graph shows a relationship between the number of factor concepts and the degree of approximation of the original data set. We can see from the picture that even if we take a relatively small number of factor concepts, we achieve high degree of approximation. For instance, if we take first 6 factor concepts returned by Algorithm 1, we get $F(I, \mathcal{F}) \cdot 100\% = 51.89\%$. This means that *more than half the information 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. The growth of the degree of approximation is shown in Table 1. The tables say that, for instance, if we wish to achieve 90.36%

Fig. 1. Relationship between the number of factors and the approximation of the original Boolean matrix.

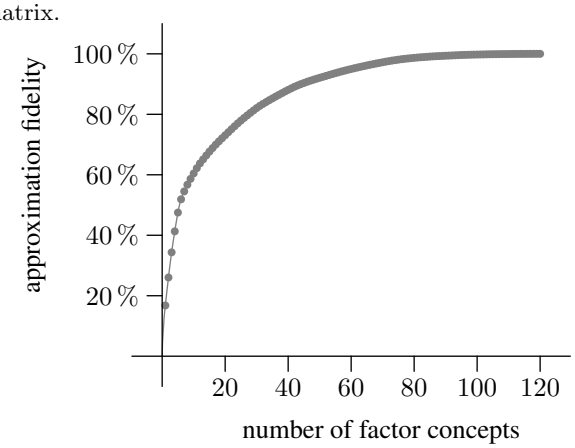

Table 1. Number of factor concepts vs. degree of approximation

factors (n)	1	2	3	4	5	6
fidelity (%)	16.78	26.03	34.35	41.29	47.51	51.89

factors (n)	10	20	45	60	100	119
fidelity (%)	60.44	73.07	90.36	94.98	99.76	100.00

approximation then it suffices to take 45 factor concepts which is significantly less than the number of the original attributes; 95% is guaranteed if we use 60 factor concepts, etc.

4 Conclusions and future research

We presented an algorithm for finding factors in Boolean factor analysis. The algorithm is based on a theorem, proven in our earlier paper, that the problem of BFA can be reduced to a problem of a covering of entries containing 1s in a given Boolean matrix I with maximal subrectangles of I which consist of 1s. This way, the problem of BFA is reducible to a particular instance of a set covering problem for which there exists an approximation algorithm. The algorithm can be sped up by further insight provided by formal concept analysis. We presented examples of Boolean factor analysis and experiments with the algorithm.

Future research will include the following problems:

– As we have seen, further insight provided by the particular nature of the set covering problem can speed up the greedy approximation algorithm. One such speed up results from the inclusion of mandatory concepts which we presented. Other ways of improving the algorithm as well as looking for other algorithms need to be investigated.

– Approximate decomposition, i.e. looking for A and B such that I is approximately equal to $A \circ B$, cf. the above experiments with MUSHROOM data. Both theoretical insight and experiments are needed in this direction.
– We did not impose any restrictions on $\mathcal{F}$ except for $I = A_{\mathcal{F}} \circ B_{\mathcal{F}}$. It might be desirable to look for $\mathcal{F}$ such that the number of attributes in the formal concepts' extents are restricted in one way or another. For instance, all formal concepts from $\mathcal{F}$ have approximately the same number of attributes, this means the level of generality of all factors is approximately the same. In general, a question of what is a good set $\mathcal{F}$ of factor concepts needs to be investigated. A small number of factor concepts, considered in this paper as the only criterion, might not always be the best one by itself.
– New factors can be seen as new attributes using which the objects are described. Since the number of new attributes (factors) is less than or equal to the number of the original attributes, a general question is this: Can we use the new attributes for more efficient reasoning and manipulation of objects? For instance: Is it useful to extract association rules over the objects which contain the new attributes? Is it useful to construct decision trees to classify objects using the new attributes?
– Another topic relates to the question of whether there are connections between formal concept analysis, concept lattices and related structures on the one hand, and associative neural networks on the other hand. In the light of [8] and the present paper, this question should be pursued. Useful hints in this respect could be provided by [2] and its followers such as [15].
– The last remark we want to make concerns the possibility to extend factorization of Boolean 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]. A paper on this topic is in preparation.

References

1. Bartholomew, D. J., Knott M.: *Latent Variable Models and Factor Analysis, 2nd Ed.*, London, Arnold, 1999.
2. Belohlavek R.: Representation of concept lattices by bidirectional associative memories. *Neural Computation* **12**, 10(2000), 2279–2290.
3. Belohlavek R.: *Fuzzy Relational Systems: Foundations and Principles.* Kluwer, Academic/Plenum Publishers, New York, 2002.
4. Belohlavek R., Vychodil V.: Formal concepts are optimal factors in Boolean factor analysis (submitted). Preliminary version appeared as On Boolean factor analysis with formal concepts as factors. SCIS & ISIS 2006, Int. Conf. Soft Computing and Intelligent Systems & Int. Symposium on Intelligent Systems, Sep 20-24, 2006, Tokyo, Japan, pp. 1054-1059.
5. Carpineto C., Romano G.: *Concept Data Analysis. Theory and Applications.* J. Wiley, 2004.
6. Cormen T. H., Leiserson C. E., Rivest R. L., Stein C.: *Introduction to Algorithms, 2nd Ed.* MIT Press, 2001.

7. Cudeck R., MacCallum R. C. (Eds.): *Factor Analysis at 100: Historical Developments and Future Directions.* Lawrence Erlbaum Associates, Inc., 2007.
8. Frolov A. A., Húsek D., Muraviev I. P., Polyakov P. A.: Boolean factor analysis by Hopfield-like autoassociative memory. *IEEE Transactions on Neural Networks* Vol. **18**, No. 3, May 2007, pp. 698–707.
9. Ganter B., Wille R.: *Formal Concept Analysis. Mathematical Foundations.* Springer, Berlin, 1999.
10. Harman H. H.: *Modern Factor Analysis, 2nd Ed.* The Univ. Chicago Press, Chicago, 1970.
11. Keprt A., Snášel V. Binary factor analysis with help of formal concepts. In *Proc. CLA 2004*, Ostrava, Czech Republic, 2004, pp. 90-101, ISBN 80–248–0597–9.
12. Keprt A., Snášel V.: Binary Factor Analysis with Genetic Algorithms. Proceedings of 4th IEEE WSTST 2005, Muroran, Japan. Springer Verlag, BerlinHeidelberg, Germany, 2005, pp. 1259-1268.
13. McDonald R. P.: *Factor Analysis and Related Methods.* Lawrence Erlbaum Associates, Inc., 1985.
14. Mickey, M.R., Mundle, P. and 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.
15. Rajapakse R.K, Denham M.: Fast access to concepts in concept lattices via bidirectional associative memories. *Neural Computation* **17**, 10(2005), 2291–2300.
16. Řezanková H., Húsek, D., Frolov, A.A.: Using standard statistical procedures for Boolean factorization. In *Proc. SIS 2003*, Naples, Italy, 2003, ISBN 88–8399–053–6.
17. Sirota, A. M., Frolov, A. A., Húsek, D.: Nonlinear factorization in sparsely encoded Hopfield-like neural networks. *ESANN European Symposium on Artifical Neural Networks*, Bruges, Belgium, 1999,pp. 387–392.
18. Spearman C.: General intelligence, objectively determined and measured. *Amer. J. Psychology* **15**(1904), 201–293.
19. Veiel H.O.: Psychopathology and Boolean factor analysis: a mismatch. *Psychological Medicine*. Vol. **15** (1985), issue 3 (Aug), pp. 623-630, ISSN 0033–2917.
20. Wille R.: Restructuring lattice theory: an approach based on hierarchies of concepts. In: I. Rival (Ed.): Ordered Sets, 445–470, Reidel, Dordrecht-Boston, 1982.