Factor Analysis of Multi-Relational Data

Markéta Trnečková

Rigorous thesis

Department of Computer Science
Faculty of Science
Palacký University Olomouc
2019

Abstract – The Boolean factor analysis is an established method for analysis and preprocessing of Boolean data. In the basic setting, this method is designed for finding factors, new variables, which may explain or describe the original input data. Many real-world data sets are more complex than a simple data table. For example almost every web database is composed from many data tables and relations between them. In this thesis, a new approach to the Boolean factor analysis, which is tailored for multi-relational data, is presented. Sometimes, Boolean data can be limiting. Especially the relation between input matrices is not necessarily of a Boolean nature. Usually this relation represents linkages to some degree, e.g. how much a user likes or dislikes a movie. Using Boolean method for such data—data must be somehow binarized first—leads to a loss of information. We reformulate decomposition problem for multi-relational data with ordinal relations. Then we propose a new algorithm for such data along with an experimental evaluation.

Contents

Preface

This thesis focuses to explore the problem of finding hidden variables, i.e. factors in multi-relational data. The thesis is based on three papers listed below:

- Krmelova M., Trnecka M.: Boolean Factor Analysis of Multi-relational Data. In: Ojeda-Aciego M., Outrata J. (Eds.): CLA 2013: Proceedings of the 10th International Conference on Concept Lattices and Their Applications, 2013, pp. 187–198, La Rochelle, France, October 2013.
- Trnecka M., Trneckova M.: An Algorithm for the Multi-Relational Boolean Factor Analysis based on Essential Elements. In: K. Bertet, S. Rudolph (Eds.): CLA 2014: Proceedings of the 11th International Conference on Concept Lattices and Their Applications, 2014, pp. 107–118.
- Trnecka M., Trneckova M.: Decomposition of Boolean Multi-Relational Data with Graded Relations. In Proceedings of the 8th IEEE International Conference on Intelligent Systems (IEEE IS'16), 2016, pp. 221–226.

The full list of my publications can be found on my personal web page `www.marketa-trneckova.cz`.

This thesis consists of five chapters. The first chapter includes a brief introduction and an overview of related works. Second chapter contains a notation used in the thesis, a short introduction to BMF, and a background of the thesis are presented. Next chapter proceed with the main part of this thesis. In Chapter 3 we outline a problem setting, basic idea of our algorithm for a new kind of multi-relational data and algorithm itself. The algorithm is experimentally evaluated in Chapter 4. The thesis is closed by Chapter 5 containing a summary of the work.

Chapter 1

Introduction

The Boolean matrix factorization (or decomposition), also known as the Boolean factor analysis, has gained interest in the data mining community. Methods for decomposition of multi-relational data, i.e. complex data composed from many data tables interconnected via relations between objects or attributes of these data tables, were intensively studied, especially in the past few years. Multi-relational data is a more truthful and therefore often also more powerful representation of reality. An example of this kind of data can be an arbitrary relational database. In this work we start with the subset of multi-relational data, more precisely with the multi-relational Boolean data, where data tables and relations between them contain only 0s and 1s. Then we proceed towards more general case, where connection between data tables could have non boolean character.

It is important to say that many real-word data sets are more complex than one simple data table. Relations between this tables are crucial, because they carry additional information about the relationship between data and this information is important for understanding data as a whole. For this reason methods which can analyze multi-relational data usually take into account relations between data tables unlike classical Boolean matrix factorization methods which can handle only one data table.

The Multi-Relational Boolean matrix factorization (MBMF) is used for many data mining purposes. The basic task is to find new variables hidden in data, called multi-relational factors, which explain or describe the original input data. There exist several ways how to represent multi-relational factors. We represent multi-relational factor as an ordered set of classic factors from data tables, always one factor from each data table. The fact, that classic factors are connected into multi-relational factor is matter of semantic of relation between data tables.

The main problem is how to connect classic factors into one multi-relational.

The main aim of this work is to present the Boolean factor analysis of multi-relational data, which takes into account relations between data tables and extract more detailed information from this complex data and propose a new algorithm which utilize so-called essential elements from the theory of Boolean matrices. The essential elements provide information about factors which cover a particular part of data tables. This information can be used for a better connection of classic factors into one multi-relational factor. Moreover, in this paper we present a new decomposition method for multi-relational data composed from Boolean data tables interconnected via relation with values from ordered set L bounded by 0 and 1, such as the five-element scale $L = \{0, \frac{1}{4}, \frac{1}{2}, \frac{3}{4}, 1\}$. For forming multi-relational factors we use a calculus over Fuzzy logic.

1.1 Related Work

The Boolean matrix factorization (or decomposition), also known as the Boolean factor analysis, has gained interest in the data mining community during the past few years.

In the literature, we can find a wide range of theoretical and application papers about the Boolean factor analysis. The overview of the Boolean matrix theory can be found in [10]. A good overview from the BMF viewpoint is in e.g. [14]. For our work is the most important [3], where were first used formal concepts as factors.

Several heuristic algorithms for the BMF were proposed. Overview of BMF methods can be found in [2, 12].

From wide range of applications papers let us mentioned only [15] and [16], where the BMF is used for solving the Role mining problem.

In the literature, there can be found several methods for the latent factor analysis of ordinal data and also of multi-relational data [11], but using these methods for Boolean data has proved to be inconvenient many times.

The BMF of multi-relational data is not directly mentioned in any previous work. Indirectly, it is mentioned, in a very specific form, in [13] as Joint Subspace Matrix Factorization, where there are two Boolean matrices, which both share the same rows (or columns). The main aim is to find a set of shared factors (factors common for both matrices) and a set of specific factors (factors which are either in first or second matrix, not in both). This can be viewed as particular, very limited setting of our work.

From our point of view are also relevant works [6, 9]. These introduce the Relational formal concept analysis (RCA), i.e. the Formal concept analysis on multi-relational data. Our approach is different from the RCA. In our approach, we extract factors from each data table and connect these factors into more general factors. In RCA, they iteratively merge data tables into one in the following way: in each step they computed all formal concepts of one data table and these concepts are used as additional attributes for the merged data table. After obtaining a final merged data table, all formal concepts are extracted. Let us mention that our approach delivers more informative results than a simple use of BMF on merged data table from RCA, moreover getting merged data table is computationally hard.

Chapter 2

Preliminaries and basic notions

2.1 Boolean Matrix Factorization

We assume familiarity with the basic notions of FCA [4]. In this work, we use the binary matrix terminology, because it is more convenient from our point of view. Consider an $n \times m$ object-attribute matrix C with entries $C_{ij} \in \{0, 1\}$ expressing whether an object i has an attribute j or not, i.e. C can be understood as a binary relation between objects and attributes. Because there is no danger of confusion we can consider this matrix as a formal context $\langle X, Y, C\rangle$, where X represents a set of n objects and Y represents a set of m attributes.

A formal concept of $\langle X, Y, C\rangle$ is any pair $\langle E, F\rangle$ consisting of $E \subseteq X$ (so-called extent) and $F \subseteq Y$ (so-called intent) satisfying $E^{\uparrow} = F$ and $F^{\downarrow} = E$ where $E^{\uparrow} = \{y \in Y \,|\, \text{for each } x \in E : \langle x, y\rangle \in C\}$, and $F^{\downarrow} = \{x \in X \,|\, \text{ for each } y \in F : \langle x, y\rangle \in C\}$.

The goal of the BMF (the idea from [1, 8]) is to find decomposition

$$C = A \circ B \tag{2.1}$$

of I into a product of an $n \times k$ object-factor matrix A over $\{0, 1\}$, a $k \times m$ matrix B over $\{0, 1\}$, revealing thus k factors, i.e. new, possibly more fundamental attributes (or variables), which explain original m attributes. We want $k < m$ and, in fact, k as small as possible in order to achieve parsimony: The n objects described by m attributes via C may then be described by k factors via A, with B representing a relationship between the original attributes and the factors. This relation can be interpreted in the following way: an object i has an attribute j if and only if there exists a factor l such that i has l (or, l applies to i) and j is one of the particular manifestations of l.

The product $\circ$ in (2.1) is a Boolean matrix product, defined by

$$(A \circ B)_{ij} = \bigvee_{l=1}^{k} A_{il} \cdot B_{lj}, \tag{2.2}$$

where $\bigvee$ denotes maximum (truth function of logical disjunction) and $\cdot$ is the usual product (truth function of logical conjunction). For example the following matrix can be decomposed into two Boolean matrices with $k < m$.

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

The least k for which an exact decomposition $C = A \circ B$ exists is in the Boolean matrix theory called the Boolean rank (or Schein rank).

An optimal decomposition of the Boolean matrix can be found via Formal concept analysis. In this approach, the factors are represented by formal concepts, see [3]. The aim is to decompose the matrix C into a product $A_{\mathcal{F}} \circ B_{\mathcal{F}}$ of Boolean matrices constructed from a set $\mathcal{F}$ of formal concepts associated to C. Let

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

where $\mathcal{B}(X, Y, C)$ represents set of all formal concepts of context $\langle X, Y, C\rangle$. Denote by $A_{\mathcal{F}}$ and $B_{\mathcal{F}}$ the $n \times k$ and $k \times m$ binary matrices defined by

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

for $l = 1, \ldots, k$. In other words, $A_{\mathcal{F}}$ is composed from characteristic vectors A_l. Similarly for $B_{\mathcal{F}}$. The set of factors is a set $\mathcal{F}$ of formal concepts of $\langle X, Y, C\rangle$, for which holds $C = A_{\mathcal{F}} \circ B_{\mathcal{F}}$. For every C such a set always exists. For details see [3].

Interpretation factors as a formal concepts is very convenient for users and we follow this point of view in our work. Because a factor can be seen as a formal concept, we can consider the intent part (denoted by $intent(F)$) and the extent part (denoted by $extent(F)$) of the factor F.

2.2 Scales of Degrees and Truth Functions $\otimes$ and $\rightarrow$

In Section 3.3 we will go beyond the Boolean case, where relations in multi-relational data can be seen as matrices with entries from some ordinal scale.

Grades of ordinal scales are conveniently represented by numbers, such as $\{1, \ldots, 5\}$. These numbers could be normalized and taken from the unit interval $[0, 1]$.

Technically, we assume that the grades are taken from a certain class of partially ordered bounded scales L. In particular, we assume that L conforms to the structure of a complete residuated lattice used in Fuzzy logic.

Complete *residuated lattice* $\mathbf{L} = \langle L, \wedge, \vee, \otimes, \rightarrow, 0, 1\rangle$, where L is partially ordered set bounded by 0 and 1 in which arbitrary infima $\bigwedge$ and suprema $\bigvee$ exist. Operation $\otimes$ is comutative, associative and has 1. Operation $\rightarrow$ residuum $a \rightarrow b = \max\{c \in L \mid a \otimes c \leq b\}$. Let us note that $\otimes$ and $\rightarrow$ represent a true function of many-valued conjunction and implication. For more details see [5, 7].

In our experiments we mainly use finite scales with the the Gödel structure: $a \otimes b = \min(a, b)$ and $a \rightarrow b = 1$ if $a \leq b$ and $a \rightarrow b = b$ otherwise. Many other definitions of $\otimes$ and $\rightarrow$ exist [5].

Fuzzy logic can be utilized for modeling a relationship "being compatible" ("satisfying relation") between factors in multi-relational factor.

We consider the formulas $\varphi(i)$ saying "factor F_i is compatible with relation R" and $\psi(j)$ saying "factor F_j is compatible with relation R", and consider a the truth degree of $\varphi(i)$ and b the truth degree of $\psi(j)$, i.e.

$$||\varphi(i)|| = a \text{ and } ||\psi(j)|| = b. \tag{2.3}$$

Now, according to fuzzy logic, the truth degree of the formula $\varphi(i)\&\psi(j)$ saying "factor F_i is compatible with relation R and factor F_j is compatible with relation R" is computed by

$$||\varphi(i)\&\psi(j)|| = ||\varphi(i)|| \otimes ||\psi(j)|| \tag{2.4}$$

where $\otimes : L \times L \rightarrow L$ is a truth function of many-valued conjunction &.

We consider the formula $\vartheta(l)$ saying "object l is compatible with relation R" and consider c_l the truth degree of $\vartheta(l)$, i.e. $||\vartheta(l)|| = c_l$, where l is from some index set J. Then truth degree of formula $(\forall l)(\vartheta(l))$ which says "all objects from index set J are compatible with relation R ", is computed by

$$||(\forall l)(\vartheta(l))|| = \textstyle\bigwedge_{l \in J} ||\vartheta(l)||, \tag{2.5}$$

where $\bigwedge$ denotes the infimum.

We consider the formulas $\varphi(i, j)$ meaning "object i belongs to factor F_j" and $\psi(i, l)$ saying "object i has attribute l in relation R", and consider a the truth degree of $\varphi(i, j)$ and b the truth degree of $\psi(i, l)$. Now, according to fuzzy logic, the truth degree of the formula "if $\varphi(i, j)$ then $\psi(i, l)$" which says "if object i belongs to factor F_j then object i has attribute l in relation R" is computed by

$$||\varphi(i,j) \Rightarrow \psi(i,l)|| = ||\varphi(i,j)|| \rightarrow ||\psi(i,l)|| \tag{2.6}$$

where $\rightarrow$: $L \times L \rightarrow L$ is a truth function of many valued implication.

Chapter 3

Multi-relational factor analysis

3.1 Problem definition

In this section we describe our basic problem setting. We have two Boolean data tables C_1 and C_2, which are interconnected with relation $R_{C_1C_2}$. This relation is over the objects of first data table C_1 and the attributes of second data table C_2, i.e. it is an objects-attributes relation. In general, we can also define an objects-objects relation or an attributes-attributes relation. Our goal is to find factors, which explain the original data and which take into account the relation $R_{C_1C_2}$ between data tables.

Definition 1. *Relation factor (pair factor) on data tables C_1 and C_2 is a pair $\left\langle F_i^{C_1}, F_j^{C_2} \right\rangle$, where $F_i^{C_1} \in \mathcal{F}_{C_1}$ and $F_j^{C_2} \in \mathcal{F}_{C_2}$ ($\mathcal{F}_{C_i}$ denotes the set of factors of data table C_i) and satisfying relation $R_{C_1C_2}$.*

There are several ways how to define the meaning of "satisfying relation" from Definition 1. We will define the following three approaches (this definition holds for an object-attribute relation, other types of relations can be defined in similar way):

- $F_i^{C_1}$ and $F_j^{C_2}$ form pair factor $\langle F_i^{C_1}, F_j^{C_2} \rangle$ if holds:

$$\bigcap_{k \in extent(F_i^{C_1})} R_k \neq \emptyset \text{ and } \bigcap_{k \in extent(F_i^{C_1})} R_k \subseteq intent(F_j^{C_2}),$$

  where R_k is a set of attributes, which are in relation with an object k. This approach we called *narrow* (it is analogy of the narrow operator in [9]).

- $F_i^{C_1}$ and $F_j^{C_2}$ form pair factor $\langle F_i^{C_1}, F_j^{C_2}\rangle$ if holds:

$$\left(\left(\bigcap_{k\in extent(F_i^{C_1})} R_k\right) \cap intent(F_j^{C_2})\right) \neq \emptyset.$$

  We called this approach *wide* (it is analogy of the wide operator in [9]).

- for any $\alpha \in [0,1]$, F_1^i and $F_j^{C_2}$ form pair factor $\langle F_i^{C_1}, F_j^{C_2}\rangle$ if holds:

$$\frac{\left|\left(\bigcap_{k\in extent(F_i^{C_1})} R_k\right) \cap intent(F_j^{C_2})\right|}{\left|\bigcap_{k\in extent(F_i^{C_1})} R_k\right|} \geq \alpha.$$

  We called it an α-*approach*.

Remark 1. *It is obvious, that for $\alpha = 0$ and replacing $\geq$ by $>$, we obtain the wide approach and for $\alpha = 1$, we obtain the narrow one.*

Lemma 1. *For $\alpha_1 > \alpha_2$ holds, that a set of relation factors counted by α_1 is a subset of a set of relation factors obtained with α_2.*

We demonstrate our approach to factorisation of mutli-relational Boolean data by a small illustrative example.

Example 1. *Let us have two data tables C_W (Table 3.1) and C_M (Table 3.2). C_W represents women and their characteristics and C_M represents men and their characteristics.*

Table 3.1: C_W

	athlete	*undergraduate*	*wants kids*	*is attractive*
Abby		×	×	×
Becky	×		×	
Claire		×		×
Daphne	×	×	×	×

Table 3.2: C_M

	athlete	*undergraduate*	*wants kids*	*is attractive*
Adam	×			×
Ben		×	×	
Carl	×	×	×	
Dave			×	×

Table 3.3: $R_{C_W C_M}$

	athlete	*undergraduate*	*wants kids*	*is attractive*
Abby		×	×	
Becky	×		×	
Claire	×	×		×
Daphne	×	×	×	×

Moreover, we consider relation $R_{C_W C_M}$ (Table 3.3) between the objects of first the data table and the attributes of the second data table. In this case, it could be a relation with meaning "woman looking for a man with the characteristics".

Remark 2. *Generally, nothing precludes the object-object relation (whose meaning might be "woman likes a man") and the attribute-attribute relation (whose meaning might be "the characteristics of women are compatible with the characteristics of men in the second data table").*

Factors of data table C_W *are:*

- $F_1^{C_W} = \langle\{Abby,\ Daphne\}, \{undergraduate,\ wants\ kids,\ is\ attractive\}\rangle$
- $F_2^{C_W} = \langle\{Becky,\ Daphne\}, \{athlete,\ wants\ kids\}\rangle$
- $F_3^{C_W} = \langle\{Abby,\ Claire,\ Daphne\}, \{undergraduate,\ is\ attractive\}\rangle$

Factors of data table C_M *are:*

- $F_1^{C_M} = \langle\{Ben,\ Carl\}, \{undergraduate,\ wants\ kids\}\rangle$
- $F_2^{C_M} = \langle\{Adam\}, \{athlete,\ is\ attractive\}\rangle$
- $F_3^{C_M} = \langle\{Adam,\ Carl\}, \{athlete\}\rangle$
- $F_4^{C_M} = \langle\{Dave\}, \{wants\ kids,\ is\ attractive\}\rangle$

These factors were obtained via GRECOND *algorithm from [3]. We have two sets of factors (formal concepts), first set* $\mathcal{F}_{C_W} = \{F_1^{C_W}, F_2^{C_W}, F_3^{C_W}\}$ *factorising data table* C_W *and* $\mathcal{F}_{C_M} = \{F_1^{C_M}, F_2^{C_M}, F_3^{C_M}\}$ *factorising data table* C_M*.*

Now we use so far unused relation $R_{C_W C_M}$*, between* C_W *and* C_M *to join factors of* C_W *with factors of* C_M *into relational factors. For the above defined approaches we get results which are shown below. We write it as binary relations, i.e* $F_i^{C_W}$ *and* $F_j^{C_M}$ *belongs to relational factor* $\langle F_i^{C_W}, F_j^{C_M}\rangle$ *iff* $F_i^{C_W}$ *and* $F_j^{C_M}$ *are in relation:*

Narrow approach

	$F_1^{C_M}$	$F_2^{C_M}$	$F_3^{C_M}$	$F_4^{C_M}$
$F_1^{C_W}$	×			
$F_2^{C_W}$				
$F_3^{C_W}$	×			

Wide approach

	$F_1^{C_M}$	$F_2^{C_M}$	$F_3^{C_M}$	$F_4^{C_M}$
$F_1^{C_W}$	×			×
$F_2^{C_W}$	×	×	×	×
$F_3^{C_W}$	×			

0.6-*approach*

	$F_1^{C_M}$	$F_2^{C_M}$	$F_3^{C_M}$	$F_4^{C_M}$
$F_1^{C_W}$	×			
$F_2^{C_W}$		×		
$F_3^{C_W}$	×			

0.5-*approach*

	$F_1^{C_M}$	$F_2^{C_M}$	$F_3^{C_M}$	$F_4^{C_M}$
$F_1^{C_W}$	×			×
$F_2^{C_W}$		×		
$F_3^{C_W}$	×			

The relational factor in form $\langle F_i^{C_W}, F_j^{C_M}\rangle$ *can be interpreted in the following ways:*

- *Women, who belong to extent of* $F_i^{C_W}$ *like men who belong to extent of* $F_j^{C_M}$*. Specifically in this example, we can interpret factor* $\langle F_1^{C_W}, F_1^{C_M}\rangle$*, that Abby and Daphne should like Ben and Carl.*

- *Women, who belong to extent of* $F_i^{C_W}$ *like men with characteristic in intent of* $F_j^{C_M}$*. Specifically in this example, we can interpret factor* $\langle F_1^{C_W}, F_1^{C_M}\rangle$*, that Abby and Daphne should like undergraduate men, who want kids.*

- *Women, with characteristic from intent* $F_i^{C_W}$ *like men who belong to extent* $F_j^{C_M}$*. Specifically in this example, we can interpret factor* $\langle F_1^{C_W}, F_1^{C_M}\rangle$*, that undergraduate, attractive women, who want kids should like Ben and Carl.*

- *Women, with characteristic from intent* $F_i^{C_W}$ *like men with characteristic in intent of* $F_j^{C_M}$*. Specifically in this example, we can interpret factor* $\langle F_1^{C_W}, F_1^{C_M}\rangle$*, that undergraduate, attractive women, who want kids should like undergraduate men, who want kids.*

Interpretation of the relation between $F_i^{C_W}$ *and* $F_j^{C_M}$ *is driven by the approach used. If we obtain factor* $\langle F_i^{C_W}, F_j^{C_M}\rangle$ *by narrow approach, we can interpret the relation between* $F_i^{C_W}$ *and* $F_j^{C_M}$*: "women who belong to* $F_i^{C_W}$*, like men from* $F_j^{C_M}$ *completely". For example factor* $\langle F_1^{C_W}, F_1^{C_M}\rangle$ *can be interpreted: "All undergraduate attractive women, who want kids, wants undergraduate men, who want kids."*

If we obtain factor $\langle F_i^{C_W}, F_j^{C_M}\rangle$ *by wide approach, we can interpret the relation between* $F_i^{C_W}$ *and* $F_j^{C_M}$*: "women who belong to* $F_i^{C_W}$*, like something about the men from* $F_j^{C_M}$*". For example* $\langle F_2^{C_W}, F_1^{C_M}\rangle$ *can be interpreted: "All athlete woman, who want kids, like undergraduate men or man, who want kids."*

If we get $\langle F_i^{C_W}, F_j^{C_M}\rangle$ *by* α*-approach with value* α*, we interpret the relation between* $F_i^{C_W}$ *and* $F_j^{C_M}$ *as: "women from* $F_i^{C_W}$*, like men from* $F_j^{C_M}$ *enough", where* α *determines measurement of tolerance.*

Remark 3. *Not all factors from data tables* C_W *or* C_M *must be present in any relational factor. It depends on the used relation. For example in Example 1 in narrow approach, the factors* $F_2^{C_M}, F_3^{C_M}, F_4^{C_M}$ *are not involved. In this case, we can add these simple factors to the set of relational factors and consider two types of factors. This factors are not pair factors, but classical factors from* C_W *or* C_M*. Of course this depends on a particular application.*

Remark 4. *For one factor $F_i^{C_1}$ from the data table C_1, two factors from the data table C_2 (for example $F_{j_1}^{C_2}$ and $F_{j_2}^{C_2}$) can satisfy the relation. In this case we can add factor $\langle F_i^{C_1}, F_{j_1}^{C_2} \& F_{j_2}^{C_2} \rangle$, where $F_{j_1}^{C_2} \& F_{j_2}^{C_2}$ means*

$$extent(F_{j_1}^{C_2} \& F_{j_2}^{C_2}) = extent(F_{j_1}^{C_2}) \cup extent(F_{j_2}^{C_2})$$

and

$$intent(F_{j_1}^{C_2} \& F_{j_2}^{C_2}) = intent(F_{j_1}^{C_2}) \cap intent(F_{j_2}^{C_2}),$$

instead of $\langle F_i^{C_1}, F_{j_1}^{C_2} \rangle$ and $\langle F_i^{C_1}, F_{j_2}^{C_2} \rangle$ to the relation factor set (in the case, that we consider an object-attribute relation). For example, by using 0.5-*approach in Example 1, we get relational factors*

$$\big\langle \langle \{Abby, Daphne\}, \{undergraduate, wants\ kids, is\ attractive\} \rangle, \\ \langle \{Ben, Carl\}, \{undergraduate, wants\ kids\} \rangle \big\rangle$$

and

$$\big\langle \langle \{Abby, Daphne\}, \{undergraduate, wants\ kids, is\ attractive\} \rangle, \\ \langle \{Dave\}, \{wants\ kids, is\ attractive\} \rangle \big\rangle.$$

This factors can be replaced with factor

$$\big\langle \langle \{Abby, Daphne\}, \{undergraduate, wants\ kids, is\ attractive\} \rangle, \\ \langle \{Ben, Carl, Dave\}, \{wants\ kids\} \rangle \big\rangle.$$

Remark 5. *Another, simpler approach to multi-relational data factorization is such, that we do factorization of the relation $R_{C_1C_2}$. This is correct because we can imagine the relation between data tables C_1 and C_2 as another data table. For each factor, we take the extent of this factor and compute concept in C_1, which contains this extent. Similarly for intents of factors and concepts in C_2. For example one of the factors of $R_{C_WC_M}$ from Example 1 is:*

$$\langle \{Becky,\ Daphne\}, \{athlete,\ wants\ kids\} \rangle.$$

Relational factor computed from this factor will be

$$\big\langle \langle \{Becky,\ Daphne\}, \{athlete,\ wants\ kids\} \rangle, \\ \langle \{Carl\}, \{athlete,\ undergraduate,\ wants\ kids\} \rangle \big\rangle.$$

This approach seems to be better in terms of that we get pair of concepts for every factors, but we do not get an exact decomposition of data tables C_1

and C_2. Moreover this approach can not be extended to n-ary relations.

3.1.1 n-tuple relational factors, n-ary relations

Above approaches can be generalized for more than two data tables. In this generalization, we do not get factor pairs, but generally factor n-tuples. Now we extend Definition 1 to general definition of relational factor.

Definition 2. *Relation factor on data tables C_1, $C_2, \dots C_n$ is a n-tuple $\langle F_{i_1}^{C_1}, F_{i_2}^{C_2}, \dots F_{i_n}^{C_n} \rangle$, where $F_{i_j}^{C_j} \in \mathcal{F}_{C_j}$ where $j \in \{1, \dots, n\}$ ($\mathcal{F}_{C_j}$ denotes set of factors of data table C_j) and satisfying relations $R_{C_l C_{l+1}}$ or $R_{C_{l+1} C_l}$ for $l \in \{1, \dots, n-1\}$.*

We considered only binary relations between data tables, for which holds, that there exists only one relation interconnecting data tables C_i and C_{i+1} for $i \in \{1, \dots, n-1\}$. We left more general relations into the Section 3.3. Let us mentioned, that this generalization of our approach is possible in the opposite of Remark 5. We show n-tuple relational factors on example.

Example 2. *Let data table C_P (Table 3.4) represents people and their characteristic, C_R (Table 3.5) represents restaurants and their characteristics and C_C (Table 3.6) represents which ingredients are included in national cuisines.*

Table 3.4: C_P

	European	*Asian*	*American*	*male*	*female*
Adam			×	×	
Ben	×			×	
Carol	×				×
Dale		×		×	
Emily					×
Frank				×	
Gabby		×			×

Table 3.5: C_R

	luxury	*ordinary*	*expensive*	*cheap*
Restaurant 1	×		×	
Restaurant 2	×		×	
Restaurant 3	×			×
Restaurant 4		×		×
Restaurant 5		×		×

Relation $R_{C_P C_C}$ (Table 3.7) represents relationship "person likes ingredients" and relation $R_{C_R C_C}$ (Table 3.8) represents relationship "restaurant cooks national cuisine". In Tables 3.9, 3.10, 3.11, we can see factors of data tables C_P, C_R and C_C, respectively.

One of the relational factors, which we get by 0.5-approach, is $\langle F_1^{C_P}, F_{11}^{C_C}, F_3^{C_R} \rangle$ and could be interpreted as "men would enjoy eating in luxury restaurants where the meals are cheap". Another factor is $\langle F_3^{C_P}, F_2^{C_C}, F_1^{C_R} \rangle$

Table 3.6: C_C

	vegetable	*fruit*	*fish*	*seafood*	*legumes*	*mutton*	*lamb*	*olive*	*vine*	*herbs*	*cheese*	*mushroom*	*hot spice*	*rice*	*beef*	*pork*	*poultry*	*bamboo shoot*	*nut*	*lard*	*rabbit*	*venison*	*innards*	*corn*	*pasta/noodle*	*potato*	*pastry*
	1	*2*	*3*	*4*	*5*	*6*	*7*	*8*	*9*	*10*	*11*	*12*	*13*	*14*	*15*	*16*	*17*	*18*	*19*	*20*	*21*	*22*	*23*	*24*	*25*	*26*	*27*
Greek	×	×	×	×	×	×	×	×	×	×	×								×								
Chinese	×		×	×	×							×	×	×	×	×	×	×	×	×					×		
French	×		×	×		×	×		×	×	×	×			×	×	×				×	×	×				
Indian	×	×	×	×		×	×						×	×			×										
Czech	×	×			×					×	×	×			×	×	×			×	×	×	×			×	
Spanish	×	×	×	×				×	×	×				×	×	×	×										
Mexican	×	×	×	×	×								×	×	×	×	×							×			
Italian	×	×	×	×				×	×	×	×	×		×											×		×
American	×		×	×							×				×	×	×									×	×
Japanese	×		×	×										×													
German	×	×	×									×			×	×	×					×					

Table 3.7: $R_{C_P C_C}$

	vegetable	*fruit*	*fish*	*seafood*	*legumes*	*mutton*	*lamb*	*olive*	*vine*	*herbs*	*cheese*	*mushroom*	*hot spice*	*rice*	*beef*	*pork*	*poultry*	*bamboo shoot*	*nut*	*lard*	*rabbit*	*venison*	*innards*	*corn*	*pasta/noodle*	*potato*	*pastry*
	1	*2*	*3*	*4*	*5*	*6*	*7*	*8*	*9*	*10*	*11*	*12*	*13*	*14*	*15*	*16*	*17*	*18*	*19*	*20*	*21*	*22*	*23*	*24*	*25*	*26*	*27*
Adam				×							×		×		×					×						×	
Ben												×			×	×	×			×	×	×					
Carol	×	×						×	×	×	×			×			×	×						×			
Dale			×	×										×			×		×		×	×			×		×
Emily			×				×		×			×					×						×	×			
Frank					×	×							×		×	×											
Gabby	×							×			×			×			×										

and could be interpreted as "women enjoy eating in ordinary cheap restaurants".

Table 3.8: $R_{C_R C_C}$

	Greek	*Chinese*	*French*	*Indian*	*Czech*	*Spanish*	*Mexican*	*Italian*	*American*	*Japanese*	*German*
Restaurant 1	×		×		×	×		×			
Restaurant 2	×	×		×			×	×		×	
Restaurant 3					×			×	×		×
Restaurant 4						×	×	×	×		
Restaurant 5		×		×						×	×

Table 3.9: Factors of data table C_P

$F_i^{C_P}$	*Extent*	*Intent*
$F_1^{C_P}$	{Adam, Ben, Dale, Frank}	{male}
$F_2^{C_P}$	{Adam, Emily, Frank}	{American}
$F_3^{C_P}$	{Carol, Emily, Gabby}	{female}
$F_4^{C_P}$	{Ben, Carol}	{European}
$F_5^{C_P}$	{Dale, Gabby}	{Asian}

Table 3.10: Factors of data table C_R

$F_i^{C_R}$	*Extent*	*Intent*
$F_1^{C_R}$	{Restaurant 4, Restaurant 5}	{ordinary, cheap}
$F_2^{C_R}$	{Restaurant 1, Restaurant 2}	{luxury, expensive}
$F_3^{C_R}$	{Restaurant 3}	{luxury, cheap}

Table 3.11: Factors of data table C_C

$F_i^{C_C}$	*Extent*	*Intent*
$F_1^{C_C}$	{Chinese, French, Spanish, Mexican, American, German}	$\{1, 3, 15, 16, 17\}$
$F_2^{C_C}$	{Greek, Spanish, Italian}	$\{1, 2, 3, 4, 8, 9, 10\}$
$F_3^{C_C}$	{French, Czech}	$\{1, 10, 11, 12, 15, 16, 17, 21, 22, 23\}$
$F_4^{C_C}$	{Chinese, Indian, Spanish, Mexican, Italian, Japanese}	$\{1, 3, 4, 14\}$
$F_5^{C_C}$	{Greek, French, Indian}	$\{1, 3, 4, 6, 7\}$
$F_6^{C_C}$	{Chinese}	$\{1, 3, 4, 5, 12, 13, 14, 15, 16, 17, 18, 19, 20, 25\}$
$F_7^{C_C}$	{Italian, American}	$\{1, 3, 4, 11, 27\}$
$F_8^{C_C}$	{Greek, Czech, Mexican}	$\{1, 2, 5\}$
$F_9^{C_C}$	{Indian, Mexican}	$\{1, 2, 3, 4, 13, 14, 17\}$
$F_{10}^{C_C}$	{Czech, Itelian, German}	$\{1, 2, 12\}$
$F_{11}^{C_C}$	{Czech, , American}	$\{1, 15, 16, 17, 26\}$
$F_{12}^{C_C}$	{Greek}	$\{1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 19\}$
$F_{13}^{C_C}$	{Greek, French, Spanish, Italian}	$\{1, 3, 4, 9, 10\}$
$F_{14}^{C_C}$	{Chinese, Czech}	$\{1, 5, 12, 15, 16, 17, 20\}$
$F_{15}^{C_C}$	{French, Czech, German}	$\{1, 12, 15, 16, 17, 22\}$
$F_{16}^{C_C}$	{Mexican}	$\{1, 2, 3, 4, 5, 13, 14, 15, 16, 17, 24\}$
$F_{17}^{C_C}$	{Chinese, Itelian}	$\{1, 3, 4, 12, 14, 25\}$

3.1.2 Representation of connection between factors

We can represent the relational factors via graph (n-partite). See Figure 3.1, which presents the results from the previous example. Each group of nodes ($F_i^{C_P}, F_i^{C_C}, F_i^{C_R}$) represents factors of a specific data table. Between two nodes, there is an edge iff factors representing nodes satisfy the input relation. Relational factor is path between nodes, which include at most one node from each group. For example, $\left\langle F_2^{C_P}, F_3^{C_C}, F_1^{C_R} \right\rangle$ is a relational factor because there is an edge between nodes $F_2^{C_P}$ and $F_3^{C_C}$ and between $F_3^{C_C}$ and $F_1^{C_R}$.

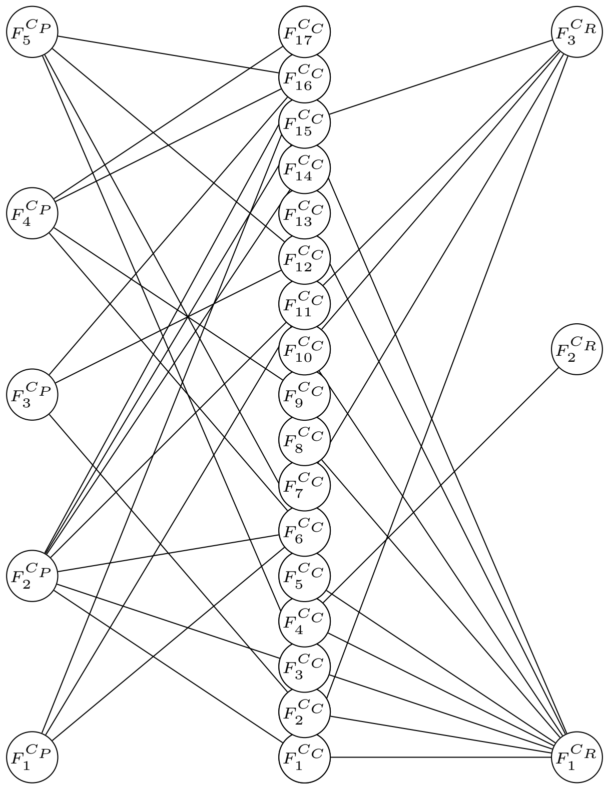


Figure 3.1: Representation factors connections via graph.

3.2 Algorithm for MBMF

Before we present the algorithm for the MBMF we show on a simple example basic ideas that are behind the algorithm. For this purpose we take the same data as in Example 1 (with different labelling).

As we mentioned above if we take tables C_1, C_2 and relation $R_{C_1C_2}$, we obtain with the narrow approach two connections between factors, i.e. two

Table 3.12: C_1

	a	b	c	d
1		×	×	×
2	×		×	
3		×		×
4	×	×	×	×

Table 3.13: C_2

	e	f	g	h
5	×			×
6		×	×	
7	×	×	×	
8			×	×

Table 3.14: $R_{C_1C_2}$

	e	f	g	h
1		×	×	
2	×		×	
3	×	×		×
4	×	×	×	×

multi-relational factors. These factors explain only 60 percent of data. There usually exist more factorizations of Boolean data table. Factors in our example were obtained with using GreConD algorithm from [3]. GreConD algorithm select in each iteration a factor which covers the biggest part of still uncovered data. Now we are in the situation, where we want to obtain a different set of factors, with more connections between them. For this purpose we can use essential elements. Firstly we compute essential parts of C_1 (denoted $Ess(C_1)$) and C_2 (denoted $Ess(C_1)$). With the essential part of data table we mean all essential elements (tables 3.18 and 3.19).

Table 3.15: $Ess(C_1)$

	a	b	c	d
1			×	
2	×			
3		×		×
4				

Table 3.16: $Ess(C_2)$

	e	f	g	h
5	×			×
6		×		
7	×			
8			×	×

Each essential element in $Ess(C_1)$ is defined via interval in concept lattice of C_1 (Fig. 3.2a) and similarly for essential elements in $Ess(C_2)$ (Fig 3.2b). In Fig. 3.2a is highlighted interval $\mathcal{I}_{1c}$ corresponding to essential element $(C_1)_{1c}$. In Fig. 3.2b is highlighted interval corresponding to essential element $(C_2)_{8g}$. Let us note that concept lattices here are only for illustration purpose. For computing $Ess(C_1)$ and $Ess(C_2)$ is not necessary to construct concept lattices at all. Now, if we use the fact that we can take an arbitrary concept (factor) from each interval to obtain a complete factorization of data table, we have several options which concepts can be connect into one. More precisely we can take two intervals and try to connect each concept from the first interval with concepts from the second one. Again, we obtain full factorization of input data tables, but now we can select factors with regard to a relation between them.

For example, if we take highlighted intervals, we obtain possibly four connections. First highlighted interval contains two concepts $c_1 = \langle\{1,2,4\},\{c\}\rangle$

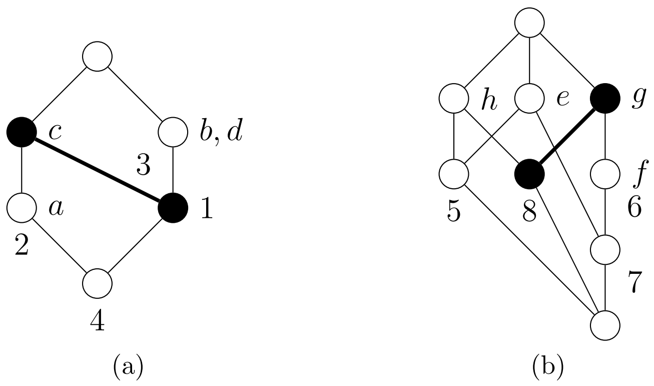


Figure 3.2: Concept lattices of C_1 (a) and C_2 (b)

and $c_2 = \langle\{1,4\},\{b,c,d\}\rangle$. Second consist of concepts $d_1 = \langle\{6,7,8\},\{g\}\rangle$ and $d_2 = \langle\{8\},\{g,h\}\rangle$. Only two connections ($c_1$ with d_1 and c_1 with d_2) satisfy relation $R_{C_1C_2}$, i.e. can be connected.

For two intervals it is not necessary to try all combination of factors. If we are not able to connect concept $\langle A,B\rangle$ from the first interval with concept $\langle C,D\rangle$ from the second interval, we are not able connect $\langle A,B\rangle$ with any concept $\langle E,F\rangle$ from the second interval, where $\langle C,D\rangle \subseteq \langle E,F\rangle$. Also if we are not able to connect concept $\langle A,B\rangle$ from the first interval with concept $\langle E,F\rangle$ from the second interval, we are not able connect any concept $\langle C,D\rangle$ from the first interval, where $\langle C,D\rangle \subseteq \langle A,B\rangle$, with concept $\langle E,F\rangle$. Let us note that $\subseteq$ is classical subconcept-superconcept ordering.

Even if we take this search space reduction into account, search in this intervals is still time consuming. We propose an heuristic approach which takes attribute concepts in intervals of the second data table, i.e. the bottom elements in each interval. In intervals of the first data table we take greatest concepts which can be connected via relation, i.e. set of common attributes in relation is non-empty. The idea behind this heuristic is that a bigger set of objects possibly have a smaller set of common attributes in a relation and this leads to bigger probability to connect this factor with some factor from the second data table, moreover, if we take factor which contains the biggest set of attributes in intervals of the second data table.

Because we do not want to construct the whole concept lattice and search in it, we compute candidates for greatest element directly from relation $R_{C_1C_2}$. We take all objects belonging to the top element of interval $\mathcal{I}_{ij}$ from the first data table and compute how many of them belong to each attribute in the relation. We take into account only attributes belonging to object i.

Table 3.17: Connections between factors

	$F_1^{C_2}$	$F_2^{C_2}$	$F_3^{C_2}$	$F_4^{C_2}$
$F_1^{C_1}$			×	
$F_2^{C_1}$		×	×	
$F_3^{C_1}$		×	×	×

We take as candidate the greatest set of objects belonging to some attribute in a relation, which satisfies that if we compute a closure of this set in the first data table, resulting set of objects do not have empty set of common attributes in a relation.

Applying this heuristic on data from the example, we obtain three factors in the first data table, $F_1^{C_1} = \langle\{2,4\},\{a,c\}\rangle$, $F_2^{C_1} = \langle\{1,3,4\},\{c,d\}\rangle$, $F_3^{C_1} = \langle\{1,2,4\},\{c\}\rangle$ and four factors $F_1^{C_2} = \langle\{5\},\{e,h\}\rangle$, $F_2^{C_2} = \langle\{6,7\},\{f,g\}\rangle$, $F_3^{C_2} = \langle\{7\},\{e,f,g\}\rangle$, $F_4^{C_2} = \langle\{8\},\{g,h\}\rangle$ from the second one. Between this factors, there are six connections satisfying the relation. These connections are shown in table 3.17.

We form multi-relational factors in a greedy manner. In each step we connect factors, which cover the biggest part of still uncovered part of data tables C_1 and C_2. Firstly, we obtain multi-relational factor $\langle F_2^{C_1}, F_2^{C_2}\rangle$ which covers 50 percent of the data. Then we obtain factor $\langle F_3^{C_1}, F_4^{C_2}\rangle$ which covers together with first factor 75 percent of the data and last we obtain factor $\langle F_1^{C_1}, F_3^{C_2}\rangle$. All these factors cover 90 percent of the data. By adding other factors we do not obtain better coverage of input data. These three factors cover the same part of input data as six connections from table 3.17.

Remark 6. *As we mentioned above and what we can see in the example, multi-relational factors are not always able to explain the whole data. This is due to nature of data. Simply there is no information how to connect some classic factors, e.g. in the example no set of objects from C_1 has in $R_{C_1C_2}$ a set of common attributes equal to $\{e,h\}$ (or only $\{e\}$ or only $\{h\}$). From this reason we are not able to connect any factor from C_1 with factor $F_1^{C_2}$.*

Remark 7. *In previous part we explain the idea of the algorithm on a object-attribute relation between data tables. It is also possible consider different kind of relation, e.g. object-object, attribute-object or attribute-attribute relation. Without loss of generality we present the algorithm only for the object-attribute relation. Modification to a different kind of relation is very simple.*

Now we are going to describe the pseudo-code (Algorithm 1) of our algorithm for MBMF. Input to this algorithm are two Boolean data tables C_1 and C_2, binary relation $R_{C_1C_2}$ between them and a number $p \in [0,1]$ which represent how large part of C_1 and C_2 we want to cover by multi-relational factors, e.g. value 0.9 mean that we want to cover 90 percent of entries in input data tables. Output of this algorithm is a set $\mathcal{M}$ of multi-relational factors that covers the prescribed portion of input data (if it is possible to obtain prescribed coverage). The first computed factor covers the biggest part of data.

First, in lines 1–2 we compute essential part of C_1 and C_2. In lines 2–4 we initialize variables U_{C_1} and U_{C_2}. These variables are used for storing information about still uncovered part of input data. We repeat the main loop (lines 5–18) until we obtain a required coverage or until it is possible to add new multi-relational factors which cover still uncovered part (lines 12–14).

In the main loop for each essential element we select the best candidate from interval $\mathcal{I}_{ij}$ from the first data table in the greedy manner described in the algorithm idea, i.e. we take the greatest concept which can be connected via relation. Than we try to connect this candidate with factors from the second data table. We compute cover function and we add to $\mathcal{M}$ the multi-relational factor maximizing this coverage.

In lines 16–17 we remove from U_{C_1} and U_{C_2} entries which are covered by actually added multi-relational factor.

Our implementation of the algorithm follows the pseudo-code conceptually, but not in details. For example we speed up the algorithm by precomputing candidates or instead computing candidates for each essential elements, we compute candidates for essential areas, i.e. essential elements which are covered by one formal concept.

Remark 8. *The input of our algorithm are two Boolean data tables and one relation between them. In general we can have more data tables and relations. Generalization of our algorithm for such input is possible. Due to lack of space we mentioned only an idea of this generalization. For the input data tables $C_1, C_2, \dots, C_n$ and relations $R_{C_iC_{i+1}}, i \in \{1, 2, \dots, n-1\}$ we firstly compute multi-relational factors for C_{n-1} and C_n. Then iteratively compute multi-relational factors for C_{n-2} and C_{n-1}. From this pairs we construct n-tuple multi-relational factor.*

We do not make a detail analysis of the time complexity of the algorithm. Even our slow implementation in MATLAB is fast enough for factorization usually large datasets in a few minutes.

Algorithm 1: Algorithm for the multi-relational BFA

Input: Boolean matrices C_1, C_2 and relation $R_{C_1C_2}$ between them and $p \in [0, 1]$

Output: set $\mathcal{M}$ of multi-relational factors

$E_{C_1} \leftarrow Ess(C_1)$
$E_{C_2} \leftarrow Ess(C_2)$
$U_{C_1} \leftarrow C_1$
$U_{C_2} \leftarrow C_2$
while $(|U_{C_1}| + |U_{C_2}|)/(|C_1| + |C_2|) \geq p$ **do**
  foreach *essential element* $(E_{C_1})_{ij}$ **do**
    compute the best candidate $\langle a, b\rangle$ from interval $\mathcal{I}_{ij}$
  end
  $\langle A, B\rangle \leftarrow$ select candidate which maximizes the cover of C_1
  select non-empty row i in E_{C_2} for which is $A^{\uparrow_{R_{C_1C_2}}} \subseteq (C_2)_{i\_}^{\downarrow\uparrow_{C_2}}$ and which maximize cover of C_1 and C_2
  $\langle C, D\rangle \leftarrow \langle (C_2)_{i\_}^{\uparrow\downarrow_{C_2}}, (C_2)_{i\_}^{\uparrow_{C_2}}\rangle$
  if *value of cover function for* C_1 *and* C_2 *is equal to zero* **then**
    break
  end
  add $\langle\langle A, B\rangle, \langle C, D\rangle\rangle$ **to** $\mathcal{M}$
  set $(U_{C_1})_{ij} = 0$ where $i \in A$ and $j \in B$
  set $(U_{C_1})_{ij} = 0$ where $i \in C$ and $j \in D$
end
return $\mathcal{F}$

3.3 Multi-relational factor analysis of data over graded relation

3.3.1 Problem Settings

Our goal—similarly as in MBFA—is to compute a set of the most important *multi-relational factors* for two input Boolean matrices C_1 and C_2 and relation $R_{C_1C_2}$ (with grades from some scale L) between them. The multi-relation factor on C_1 and C_2 is an ordered triple $\left\langle F_i^{C_1}, F_j^{C_2}, d\right\rangle$, where $F_i^{C_1} \in \mathcal{F}_{C_1}$, $F_j^{C_2} \in \mathcal{F}_{C_2}$ ($\mathcal{F}_{C_1}$ and $\mathcal{F}_{C_2}$ represent sets of factors from C_1 and C_2 respectively) and both are compatible with the relation $R_{C_1C_2}$ (satisfy relation $R_{C_1C_2}$) in degree $d \in L$.

3.3.2 Idea of the Algorithm

The main issue is how to understand that "factors $F_i^{C_1} \in \mathcal{F}_{C_1}$ and $F_j^{C_2} \in \mathcal{F}_{C_2}$ are compatible in a relation $R_{C_1C_2}$ in degree d". Intuitively—in case of object-attribute relation—we want all objects from $F_i^{C_1}$ to be compatible with relation $R_{C_1C_2}$ and also all attributes from $F_j^{C_2}$ to be compatible with this relation. Proposition that "object x is compatible with relation" means: if object x is in $F_i^{C_1}$ then x has all attributes from $F_j^{C_2}$ in relation $R_{C_1C_2}$. Similarly proposition that "attribute y is compatible with relation" means: if attribute y is in $F_j^{C_2}$ then y applies to all objects from $F_i^{C_1}$ in relation $R_{C_1C_2}$. This leads—using formulas from 2.2—to a single formula. Degree d of satisfaction of this formula is computed in a following way:

$$\begin{aligned} d \;=\; & \left(\bigwedge_{x\in A}\left(x \to \bigwedge_{y\in D} R_{C_1C_2}(x,y)\right)\right)\otimes \\ & \left(\bigwedge_{y\in D}\left(y \to \bigwedge_{x\in A} R_{C_1C_2}(x,y)\right)\right). \end{aligned} \tag{3.1}$$

Let us note that the previous formula is valid in case of object-attribute relation, i.e. relation $R_{C_1C_2}$ is between object of C_1 and attributes of C_2. It could be generalized to any type of relation (object-object, attribute-attribute, attribute-object relation). Moreover, it is not needed to be restricted to only two data tables and one relation between them. We can easily generalize our approach to more data tables and relations between them.

3.3.3 Algorithm

Now we are going to describe the pseudo-code of our algorithm (Algorithm 2) for above described data.

The algorithm takes Boolean matrices C_1 and C_2 and object-attribute relation (with grades over L) $R_{C_1C_2}$ between them as an input. Output of this algorithm is a set of multi-relational factors $\mathcal{F}$. On lines 1–2, we compute Boolean factors of C_1 and C_2 respectively. For this purpose we utilized simple Boolean matrix factorization algorithm which uses a basic idea behind BMF algorithm GreEss—so called *essential elements*—introduced in [2]. For computing exact decomposition of Boolean matrix it is sufficient to take only one arbitrary concept from each essential interval bounded by object and attribute concepts in concept lattice whose elements are $\mathcal{B}(X, Y, I)$. Due to some useful features we take object concepts as factors of C_1 and attribute concepts as factors of C_2. On lines 3–4 we store yet uncovered part of C_1 and C_2 in U_{C_1} and U_{C_2} respectively. Then for each factor (lines 5–7) from $\mathcal{F}_{C_1}$ we compute a set of candidates—factors from $\mathcal{F}_{C_2}$—that could be connected (are compatible in relation $R_{C_1C_2}$ in degree $d > 0$ computed via formula (3.1)). In main loop (lines 8–14) we select factor from $\mathcal{F}_{C_1}$ and factor from related set of candidates that cover the biggest part of U_{C_1} and U_{C_2} and we add it to output set $\mathcal{F}$ (line 10). Then we remove all covered entries from sets U_{C_1} and U_{C_2} (lines 11–12). We repeat the main loop until factors improving coverage of U_{C_1} and U_{C_2} exist.

Algorithm 2: Computing multi-relational factors

Input: Boolean matrices C_1, C_2 and relation $R_{C_1C_2}$.
Output: Set $\mathcal{F}$ of multi-relational factors.

1 $\mathcal{F}_{C_1} \leftarrow$ Boolean factors of C_1 $\mathcal{F}_{C_2} \leftarrow$ Boolean factors of C_2 $U_{C_1} \leftarrow C_1$ $U_{C_2} \leftarrow C_2$
2 **foreach** $\langle A, B\rangle \in \mathcal{F}_{C_1}$ **do**
3   compute set of all candidates $\mathcal{F}_{\langle A,B\rangle} \subseteq \mathcal{F}_{C_2}$ which are compatible in $R_{C_1C_2}$ with $\langle A, B\rangle$ in degree $d > 0$
4 **end**
5 **while** *exist* $\langle A, B\rangle$ *and* $\langle C, D\rangle \in \mathcal{F}_{\langle A,B\rangle}$ *which can be connected and improve coverage* **do**
6   **select** $\langle A, B\rangle$ and corresponding $\langle C, D\rangle \in \mathcal{F}_{\langle A,B\rangle}$ that cover the biggest parts of U_{C_1} and U_{C_2}
7   **add** $\langle\langle A, B\rangle, \langle C, D\rangle, d\rangle$ **to** $\mathcal{F}$ **remove** all entries in $\langle A, B\rangle$ **from** U_{C_1} **remove** all entries in $\langle C, D\rangle$ **from** U_{C_2} **remove** $\langle C, D\rangle$ **from** $\mathcal{F}_{\langle A,B\rangle}$
8 **end**

Remarks

The select operation from line 9 guarantees that the first computed multi-relational factors are the most important ones, i.e. describe the biggest portion of data. Unfortunately we are not able to always explain (cover) the whole input. This is due to the nature of data.

Algorithm 1 can be modified for computing multi-relational factors that explain prescribed portion of data. This corresponds with AFP problem. For more details see [2].

3.3.4 Illustrative example

Let us have two data tables C_1 (Table 3.18), where rows represent some people and attributes are their characteristics and table C_2 (Table 3.19), which holds information about restaurants (rows) and cuisine they serve (attributes). Object-attribute relation $R_{C_1C_2}$ (Table 3.20), between C_1 and C_2 can then have a meaning "person likes the cuisine". Let us assume that values in relation are from scale $\{0, 0.5, 1\}$. Where 0 represents - "person does not like the cuisine", 0.5 - "person likes the cuisine a little bit" and 1 - "person likes the cuisine".

Table 3.18: Data table C_1

	a	b	c	d
1		×	×	×
2	×		×	
3		×		×
4	×	×	×	×

Table 3.19: Data table C_2

	e	f	g	h
5	×			×
6		×	×	
7	×	×	×	
8			×	×

Table 3.20: Relation $R_{C_1C_2}$

	e	f	g	h
1	0	1	0.5	1
2	0.5	0	0.5	1
3	1	0	0	1
4	0.5	0.5	1	1

Firstly we compute factors of C_1 and C_2 via the above-mentioned BMF method. The factors of the first data table C_1 are:
$F_1^{C_1} = \langle\{1, 4\}, \{b, c, d\}\rangle$,
$F_2^{C_1} = \langle\{2, 4\}, \{a, c\}\rangle$,
$F_3^{C_1} = \langle\{1, 3, 4\}, \{b, d\}\rangle$

and the factors of the second table C_2 are:
$F_1^{C_2} = \langle\{5, 7\}, \{e\}\rangle$,
$F_2^{C_2} = \langle\{6, 7\}, \{f, g\}\rangle$,
$F_3^{C_2} = \langle\{6, 7, 8\}, \{g\}\rangle$,
$F_4^{C_2} = \langle\{5, 8\}, \{h\}\rangle$.

Using formula (3.1), we obtain degrees in which the factors of C_1 and C_2 are connected. Resulting degrees d are presented in the Table 3.21. We can for example see, that factor $F_3^{C_1}$ can form a multi-relational factor only with factor $F_4^{C_2}$ in degree 1.

Table 3.21: degrees d

	$F_1^{C_2}$	$F_2^{C_2}$	$F_3^{C_2}$	$F_4^{C_2}$
$F_1^{C_1}$	0	0.5	0.5	1
$F_2^{C_1}$	0.5	0	0.5	0.5
$F_3^{C_1}$	0	0	0	1

Nonzero entries in each row in Table 3.21 correspond to the set of candidates in Algorithm 1.

From this we iteratively choose multi-relational factors, that cover maximal portion of yet uncovered part of data tables C_1 and C_2. So first we obtain multi-relational factor $\left\langle F_1^{C_1}, F_2^{C_2}, 0.5\right\rangle$, which covers 55% of data table C_1 and 44% of data table C_2. Than we obtain a multi-relational factors $\left\langle F_2^{C_1}, F_1^{C_2}, 0.5\right\rangle$, $\left\langle F_3^{C_1}, F_4^{C_2}, 1\right\rangle$ and $\left\langle F_1^{C_1}, F_3^{C_2}, 0.5\right\rangle$. In this case, we now have covered the whole input data, so we do not need to add another multi-relational factor.

Interpretation of $\left\langle F_3^{C_1}, F_4^{C_2}, 1\right\rangle$ could be: "All people with characteristics b and d enjoy meal in restaurants 5 and 8—they like the cuisine which these restaurants serve".

Chapter 4

Experimental Evaluation

We used our algorithm from Section 3.3 in the evaluation on synthetic and real data. We studied both the ability of the extracted factors to cover the input data and the interpretation of factors.

4.1 Synthetic Data

The main factor for quality of overall decomposition is a density of relational matrix. To demonstrate this fact, we used randomly generated data.

To eliminate influence of input matrices C_1 and C_2, we fixed them. C_1 has a size 1000×500 and approximate density of ones 25% and C_2 has a size 500×1000 and the same density.

Relational matrix has a size 500×500. Grades of this matrix are from the following scale

$$L = \{0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1\}.$$

We wanted to demonstrate that the number of zeros in this relation plays a crucial role. We used 10 different sets of relational matrices with different distribution of grades. For example relations from Set 1 have a distribution of zeros equal to $\frac{90}{100}$ and distribution of the rest of grades is equal to $\frac{1}{100}$, i.e. approximately 90% of entries is equal to 0. In other sets we decreased the number of zeros and kept approximately the same distribution for the rest of the grades.

Each set contains 1000 of such relations. Results and characteristic of these sets are shown in Table 4.1. First column represents average percentage of zeros in each set, second, third and fourth column holds information about resulting coverage of C_1 and C_2 and total coverage respectively. All presented results are averaged through all 1000 relations in each set.

Table 4.1: Results for synthetic data

	average percent of zeros	average coverage of C_1	average coverage of C_2	average total coverage
Set 1	89%	65%	58%	62%
Set 2	81%	75%	69%	72%
Set 3	72%	85%	79%	82%
Set 4	61%	93%	90%	91%
Set 5	52%	95%	93%	94%
Set 6	39%	99%	98%	98%
Set 7	28%	99.8%	99.6%	99.7%
Set 8	20%	99.9%	99.9%	99.9%
Set 9	15%	99.9%	100%	99.9%
Set 10	10%	100%	100%	100%

In Table 4.1 we can see that Set 3 has approximately 72% of zero entries. For this set, our algorithm returns multi-relational factors, that explain (cover) 85% of entries of C_1, 79% of C_2. That represents 82% of whole data.

Number of different grades does not play role from the standpoint of coverage. On the other hand, they play role in quality (degree d in which individual factors satisfy relation) of factors. Therefore, we obtain analogous results by using different L with the same distributions of zeros.

4.2 Real Data

MovieLens

For quality evaluation of factors obtained by algorithm introduced in 3.3.3 we used well known real dataset MovieLens[1]. MovieLens contains two data tables and one relation between them. First one represents a set of users and their attributes, e.g. gender, age, professions. Second one represents a set of movies with their attributes, e.g. the year of production or film genre. Last part of this dataset is a relation between data tables. This relation represents movie ratings made by users. Ratings are made on a 5-star scale (values 1-5, 1 means that the user does not like the movie and 5 means that he likes the movie).

We used 10M version of MovieLens dataset. We chose users that rate the

[1] `http://grouplens.org/datasets/movielens/`

most and films that are rated the most. Ratings were normalized to $[0, 1]$ interval. By our algorithm we obtained 46 multi-relational factors. These factors cover 98 percent of input data tables. Figure 4.1 shows cumulative coverage of User and Movie data tables.

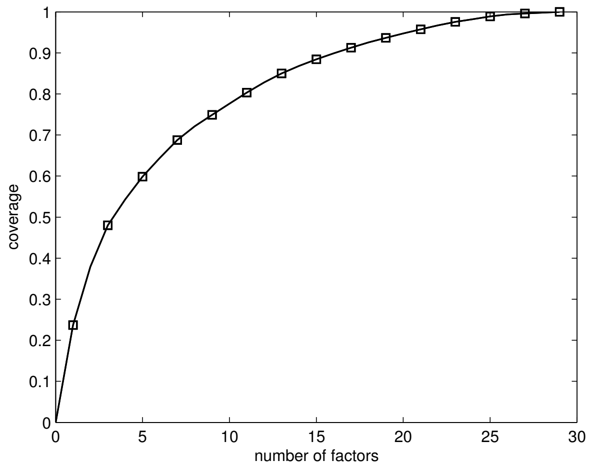


Figure 4.1: Cumulative coverage

On the x-axis there are numbers of factors and on the y-axis there is corresponding coverage of input data tables. One mean that all input entries are covered. We can see that 25 factors are sufficient for covering more than 80% of input data.

The most important factors obtained via our algorithm are:

- College female students rated action, sci-fi and thriller movies from 1980s with at least three stars.
- Females students of elementary school rated new comedy films with at least three stars.
- College males students rated action, adventure and fantasy movies with at least four stars.
- Middle aged males rated new drama films at with at least three stars.
- Late forties females working as academics or educators rated films from 1970s with five stars.
- Females in the age of 25–34 rated children, animated and comedy movies with four stars.

Arguably, all obtained factors seem to be reasonable.

MovieLens with binary relation

4.2.1 Experimental evaluation

Due to the fact that the binary case is a special case of ordinal scale, our approach can be also used on data with binary relation.

We convert the ordinal relation in to binary one. We use three different scaling. The first is that user rates a movie. The second is that a user does not like a movie (he rates movie with 1–2 stars). The last one is that user likes a movie (rates 4–5). This does not mean, that users do like (respective do not like) some genre, it means, that movies from this genre are or are not worth to see. We took the middle size version of the MovieLens dataset and we made a restriction to 3000 users and movies that were rated by that users. We take users, who rate movies the most, and we obtain dimension of the first data table 3000×30 and dimension of the second data table is 3671×26. Let us just note that for obtaining object-attribute relation we need to transpose Movies data table.

Relation "user rates a movie" make sense, because user rates a movie if he has seen it. We can understand this relation as user has seen movie. We get 29 multi-relational factors, that cover almost 100% of data (99.97%). Values of coverage, i.e. how large part of input data is covered can be seen in Figure 4.2. Graphs in Figure 4.3 show coverage of Users data table and Movies data table separately.

We can also see that for explaining more than 90 percent of data are sufficient 17 factors. This is significant reduction of input data.

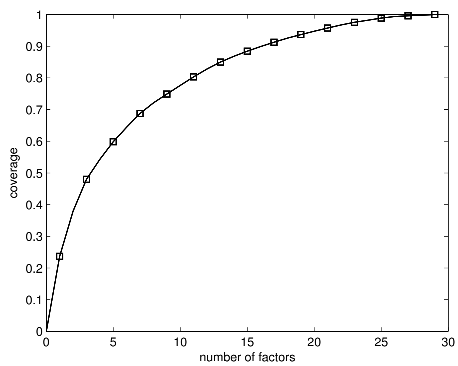
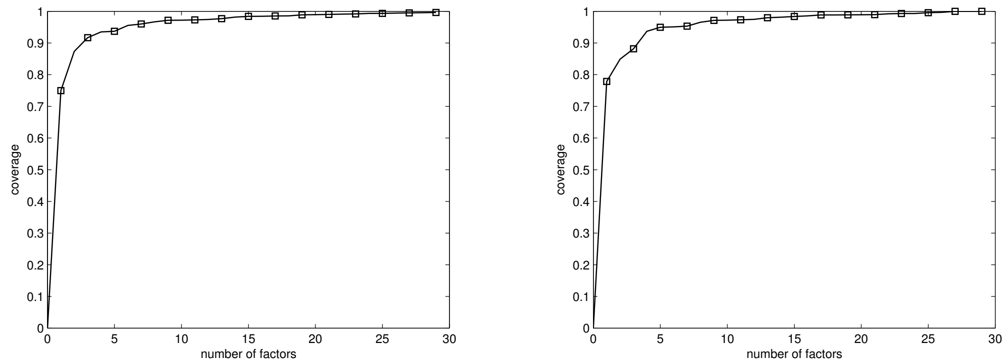


Figure 4.2: Cumulative coverage of input data



(a) Coverage of Users data table

(b) Coverage of Movies data table

Figure 4.3: Coverage of input data tables

The most important factors are:

- Males rate new movies (movies from 1991 to 2000).
- Young adult users (ages 25–34) rate drama movies.
- Females rate comedy movies.
- Youth users (18–24) rate action movies.

Another interesting factors are:

- Old users (from category 56+) rate movies from their childhood (movies from 1941 to 1950).
- Users in age range 50–55 rate children's movies. Users in this age usually have grand children.
- K-12 students rate animation movies.

Due to lack of space, we skip details about factors in relation "user does not like a movie" and relation "user does like a movie". In the first relation we get 30 factors, that covers 99.99% of data. In the second one, we get 29 factors, covering 99.96% of data. Compute all multi-relational factors on this datasets take approximately 5 minutes.

Remark 9. *In case of MovieLens we are able to reconstruct input data tables almost wholly for each three relations. Interesting question is what about the relation, i.e. can we reconstruct the relation between data tables? Answer is yes, we can. Multi-relational factors carry also information about the relation between data tables. So we can reconstruct it, but with some error. This error is a result of choosing the narrow approach.*

Reconstruction error of relation is interesting information and can be minimize if we take this error into account in phase of computing coverage. In other words we want maximal coverage with minimal relation reconstruction error. This leads to more complicated algorithm because we need weights to compute a value of utility function. We implement also this variant of algorithm. Requirement of minimal reconstruction error and maximal coverage seems to be contradictory, but this claim need more detailed study. Also it is necessary to determine correct weight settings.

Chapter 5

Conclusions

In this thesis the new approach to BMF of multi-relational data, i.e. data which are composed from many data tables and relations between them, has been presented. This approach, as opposed from to BMF, takes into account the relations and uses these relations to connect factors from individual data tables into one complex factor, which delivers more information than the simple factors.

The new algorithm for multi-relational Boolean matrix factorization, that uses essential elements from binary matrices for constructing better multi-relational factors, with regard to relations between each data table, has been presented. We test the algorithm on, in data mining well known, dataset MovieLens. From these experiments, we obtain interesting and easy interpretable results, moreover, the number of obtained multi-relational factors needed for explaining almost whole data is reasonable small. We extend a problem of multi-relational Boolean matrix decomposition toward a more general case. Our new approach is tailored for multi-relational data that contains a relation with degrees from some scale. We used calculus over Fuzzy logic to solve a problem how to connect factors into multi-relational factors.

We also present a new algorithm for this general case. Various experiments on real and synthetic data show that our algorithm produces relevant and interpretable results. Moreover—depending on the density of relation—multi-relational factors produced by our algorithm tend to cover (explain) a big portion of input data.

Let us mention, that the algorithm presented in this work can be also used for even more general mutli-relation data such as data where all input is over some scale—including data tables. We do not present this feature mainly due to fact that such kind of data are not widely used yet.

Bibliography

[1] Bartholomew D. J., Knott M.: Latent Variable Models and Factor Analysis, 2nd Ed., London, Arnold, 1999.

[2] Belohlavek R., Trnecka M.: From-Below Approximations in Boolean Matrix Factorization: Geometry and New Algorithm. ournal of Computer and System Sciences 81(8)(2015), 1678-1697.

[3] Belohlavek R., Vychodil V.: Discovery of optimal factors in binary data via a novel method of matrix decomposition. J. Comput. Syst. Sci. 76(1):3–20, 2010.

[4] Ganter B., Wille R.: Formal Concept Analysis: Mathematical Foundations. Springer, Berlin, 1999.

[5] Gottwald S.: A Treatise on Many-Valued Logics. Research Studies Press, Baldock, Hertfordshire, England, 2001.

[6] Hacene M. R., Huchard M., Napoli A., Valtechev P.: Relational concept analysis: mining concept lattices from multi-relational data. Ann. Math. Artif. Intell. 67(1)(2013), 81–108,.

[7] Hájek P.: Metamathematics of Fuzzy Logic. Kluwer, Dordrecht (1998).

[8] Harman H. H.: Modern Factor Analysis, 2nd Ed. The Univ. Chicago Press, Chicago, 1970.

[9] Huchard M., Napoli A., Rouane H. M., Valtchev P.: A proposal for combining formal concept analysis and description logics for mining relational data. ICFCA 2007.

[10] Kim K.H.: Boolean Matrix Theory and Applications. Marcel Dekker, New York, 1982.

[11] Lippert, C., Weber, S. H., Huang, Y., Tresp, V., Schubert, M., and Kriegel, H.-P.: Relation-prediction in multi- relational domains using matrix-factorization. In NIPS 2008 Workshop on Structured Input - Structured Output, NIPS, 2008.

[12] Miettinen P., Mielikäinen T., Gionis A., Das G., Mannila H., The discrete basis problem, IEEE Trans. Knowledge and Data Eng. 20(10)(2008), 1348-1362.

[13] Miettinen P.: On Finding Joint Subspace Boolean Matrix Factorizations. Proc. SIAM International Conference on Data Mining (SDM2012), pp. 954-965, 2012.

[14] Miettinen P., Mielikäinen T., Gionis A., Das G., Mannila H.: The discrete basis problem, IEEE Trans. Knowledge and Data Eng. 20(10)(2008), 1348–1362.

[15] Nau D.S., Markowsky G., Woodbury M.A., Amos D.B.: A mathematical analysis of human leukocyte antigen serology. Math Bioscience 40(1978), 243–270.

[16] Vaidya J., Atluri V., Guo Q.: The role mining problem: finding a minimal descriptive set of roles. In: Proc. SACMAT 2007, pp. 175–184, 2007.