Boolean Factor Analysis of Multi-Relational Data

Marketa Krmelova, Martin Trnecka

Data Analysis and Modeling Lab (DAMOL)
Department of Computer Science, Palacky University, Olomouc
marketa.krmelova@gmail.com, martin.trnecka@gmail.com

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 paper we present a new approach to the Boolean factor analysis, which is tailored for multi-relational data. We show our approach on simple examples and also propose future research topics.

1 Introduction

Many data sets are Boolean by nature, that is, they contain only 0s and 1s. For example, any data recording the presence (or absence) of variables in observations are Boolean. Boolean data can be seen as a binary data table (or matrix or formal context) C, where the rows represent objects and the columns represent attributes of these objects. Between objects and attributes exists an incidence relation with meaning that an object i has an attribute j and this fact is represented by one in the Boolean table, i.e. $C_{ij} = 1$. If an object i has not an attribute j, than $C_{ij} = 0$.

Many real-word data sets are more complex that a simple data table. Usually, they are composed from many data tables, which are interconnected by relations. An example of such data can be found in almost every sector of human activity. We call this kind of data *multi-relational data*. In this kind of data, this relations are crucial, because they represent additional information about the relationship between data tables and this information is important for understanding data as a whole.

The Boolean factor analysis (BFA) is used for many data mining purposes. The basic task in the BFA is to find new variables, called factors, which may explain or describe original single input data. Finding factors is obviously an important step for understanding and managing data. Boolean nature of data is in this case beneficial especially from the standpoint of interpretability of the results. On the other hand BFA is suitable for single input Boolean data table with just one relation between objects and attributes. The main aim of this work

© paper author(s), 2013. Published in Manuel Ojeda-Aciego, Jan Outrata (Eds.): CLA 2013, pp. 187–198, ISBN 978–2–7466–6566–8, Laboratory L3i, University of La Rochelle, 2013. Copying permitted only for private and academic purposes.

is to present the BFA of multi-relational data, which takes into account relations between data tables and extract more detailed information from this complex data.

2 Preliminaries and basic notions

We assume familiarity with the basic notions of FCA [3]. 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 $E^{\uparrow} = \{y \in Y \,|\, \text{for each } x \in X : \langle x, y\rangle \in C\}$, and $F^{\downarrow} = \{x \in X \,|\, \text{ for each } y \in Y : \langle x, y\rangle \in C\}$.

The goal of the BMF (the idea from [1, 6]) is to find decomposition

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

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 (1) is a Boolean matrix product, defined by

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

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 [2]. 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 [2].

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.

3 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 [8]. A good overview from the BMF viewpoint is in e.g. [12]. For our work is the most important [2], where were first used formal concepts as factors.

Several heuristic algorithms for the BMF were proposed. In our work we adopt algorithm GreConD [2] (originally called Algorithm 2), but there exist several different approaches, which use so-called "tiles" in Boolean data [4], hyper-rectangles [15] or which introduce some noise [12, 10] in Boolean data.

From wide range of applications papers let us mentioned only [13] and [14], 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 [9], 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 [11] 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 [5, 7]. 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.

4 Boolean factor analysis of multi-relational data

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 $\mathcal{R}_{C_1C_2}$. 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 $\mathcal{R}_{C_1C_2}$ between data tables.

Definition 1. *Relation factor (pair factor) on data tables C_1 and C_2 is a pair $\left\langle F_1^i, F_2^j \right\rangle$, where $F_i^1 \in \mathcal{F}_1$ and $F_2^j \in \mathcal{F}_2$ ($\mathcal{F}_i$ denotes set of factors of data table C_i) and satisfying relation $\mathcal{R}_{C_1C_2}$.*

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_1^i and F_2^j form pair factor $\langle F_1^i, F_2^j \rangle$ if holds:

$$\bigcap_{k \in extent(F_1^i)} \mathcal{R}_k \neq \emptyset \text{ and } \bigcap_{k \in extent(F_1^i)} \mathcal{R}_k \subseteq intent(F_2^j),$$

  where $\mathcal{R}_k$ 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 [7]).

- F_1^i and F_2^j form pair factor $\langle F_1^i, F_2^j \rangle$ if holds:

$$\left(\left(\bigcap_{k \in extent(F_1^i)} \mathcal{R}_k \right) \cap intent(F_1^j) \right) \neq \emptyset.$$

  We called this approach *wide* (it is analogy of the wide operator in [7]).

– for any $\alpha \in [0,1]$, F_1^i and F_2^j form pair factor $\langle F_1^i, F_2^j \rangle$ if holds:

$$\frac{\left|\left(\bigcap_{k \in extent(F_1^i)} \mathcal{R}_k\right) \cap intent(F_2^j)\right|}{\left|\bigcap_{k \in extent(F_1^i)} \mathcal{R}_k\right|} \geq \alpha.$$

We called it an α-*approach*.

Remark 1. It is obvious, that for $\alpha = 0$ and replacing $\geq$ by $>$, we get the wide approach and for $\alpha = 1$, we get 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 1) and C_M (Table 2). C_W represents women and their characteristics and C_M represents men and their characteristics.

Table 1: C_W

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

Table 2: C_M

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

Table 3: $\mathcal{R}_{C_W C_M}$

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

Moreover, we consider relation $\mathcal{R}_{C_W C_M}$ (Table 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:

– $F_1^W = \langle \{\text{Abby, Daphne}\}, \{\textit{undergraduate, wants kids, is attractive}\} \rangle$
– $F_2^W = \langle \{\text{Becky, Daphne}\}, \{\textit{athlete, wants kids}\} \rangle$
– $F_3^W = \langle \{\text{Abby, Claire, Daphne}\}, \{\textit{undergraduate, is attractive}\} \rangle$

Factors of data table C_M are:

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

These factors were obtained via GreConD algorithm from [2]. We have two sets of factors (formal concepts), first set $\mathcal{F}_W = \{F_W^1, F_W^2, F_W^3\}$ factorising data table C_W and $\mathcal{F}_M = \{F_M^1, F_M^2, F_M^3\}$ factorising data table C_M.

Now we use so far unused relation $\mathcal{R}_{C_W C_M}$, between C_W and C_M to joint 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_W^i and F_M^j belongs to relational factor $\langle F_W^i, F_M^j\rangle$ iff F_W^i and F_M^j are in relation:

Narrow approach

	F_M^1	F_M^2	F_M^3	F_M^4
F_W^1	×			
F_W^2				
F_W^3	×			

Wide approach

	F_M^1	F_M^2	F_M^3	F_M^4
F_W^1	×			×
F_W^2	×	×	×	×
F_W^3	×			

0.6-approach

	F_M^1	F_M^2	F_M^3	F_M^4
F_W^1	×			
F_W^2		×		
F_W^3	×			

0.5-approach

	F_M^1	F_M^2	F_M^3	F_M^4
F_W^1	×			×
F_W^2		×		
F_W^3	×			

The relational factor in form $\langle F_W^i, F_M^j\rangle$ can be interpreted in the following ways:

- Women, who belong to extent of F_W^i like men who belong to extent of F_M^j. Specifically in this example, we can interpret factor $\langle F_W^1, F_M^1\rangle$, that Abby and Daphne should like Ben and Carl.
- Women, who belong to extent of F_W^i like men with characteristic in intent of F_M^j. Specifically in this example, we can interpret factor $\langle F_W^1, F_M^1\rangle$, that Abby and Daphne should like undergraduate men, who want kids.
- Women, with characteristic from intent F_W^i like men who belong to extent F_M^j. Specifically in this example, we can interpret factor $\langle F_W^1, F_M^1\rangle$, that undergraduate, attractive women, who want kids should like Ben and Carl.
- Women, with characteristic from intent F_W^i like men with characteristic in intent of F_M^j. Specifically in this example, we can interpret factor $\langle F_W^1, F_M^1\rangle$, that undergraduate, attractive women, who want kids should like undergraduate men, who want kids.

Interpretation of the relation between F_W^i and F_M^j is driven by used approach. If we obtain factor $\langle F_W^i, F_M^j\rangle$ by narrow approach, we can interpret relation between F_W^i and F_M^j: "women who belong to F_W^i, like men from F_j^M completely". For example factor $\langle F_W^1, F_M^1\rangle$ can be interpreted: "All undergraduate attractive women, who want kids, wants undergraduate men, who want kids."

If we obtain factor $\langle F_W^i, F_M^j \rangle$ by wide approach, we can interpret the relation between F_W^i and F_M^j: "women who belong to F_W^i, like something about the men from F_j^M". For example $\langle F_W^2, F_M^1 \rangle$ can be interpreted: "All athlete woman, who want kids, like undergraduate men or man, who want kids."

If we get $\langle F_W^i, F_M^j \rangle$ by α-approach with value α, we interpret the relation between F_W^i and F_M^j as: "women from F_W^i, like men from F_j^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_M^2, F_M^3, F_M^4 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_1^i from the data table C_1, two factors from the data table C_2 (for example $F_2^{j_1}$ and $F_2^{j_2}$) can satisfy the relation. In this case we can add factor $\langle F_1^i, F_2^{j_1} \& F_2^{j_2} \rangle$, where $F_2^{j_1} \& F_2^{j_2}$ means

$$extent(F_2^{j_1} \& F_2^{j_2}) = extent(F_2^{j_1}) \cup extent(F_2^{j_2})$$

and

$$intent(F_2^{j_1} \& F_2^{j_2}) = intent(F_2^{j_1}) \cap intent(F_2^{j_2}),$$

instead of $\langle F_1^i, F_2^{j_1} \rangle$ and $\langle F_1^i, F_2^{j_2} \rangle$ 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

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

and

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

This factors can be replaced with factor

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

Remark 5. Another, simpler approach to multi-relational data factorization is such, that we do factorization of the relation $\mathcal{R}_{C_1 C_2}$. 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 $\mathcal{R}_{C_W C_M}$ from Example 1 is:

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

Relational factor computed from this factor will be

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

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.

4.1 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 $\langle F_1^{i_1}, F_2^{i_2}, \dots F_n^{i_n}\rangle$, where $F_j^{i_j} \in \mathcal{F}_j$ where $j \in \{1,\dots,n\}$ ($\mathcal{F}_j$ denotes set of factors of data table C_j) and satisfying relations $\mathcal{R}_{C_l C_{l+1}}$ or $\mathcal{R}_{C_{l+1} C_l}$ 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 extended version of this paper. 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 4) represents people and their characteristic, C_R (Table 5) represents restaurants and their characteristics and C_C (Table 6) represents which ingredients are included in national cuisines.

Table 4: C_P

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

Table 5: C_R

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

Table 6: C_C

	vegetable	*fruit*	*fish*	*sea food*	*legumes*	*mutton*	*lamb*	*olive*	*vine*	*herbs*	*cheese*	*mushroom*	*hot spice*	*rice*	*beef*	*pork*	*poultry*	*bamboo shoot*	*nut*	*lard*	*rabbit*	*venison*	*insides*	*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 7: $R_{C_P C_C}$

	vegetable	*fruit*	*fish*	*sea food*	*legumes*	*mutton*	*lamb*	*olive*	*vine*	*herbs*	*cheese*	*mushroom*	*hot spice*	*rice*	*beef*	*pork*	*poultry*	*bamboo shoot*	*nut*	*lard*	*rabbit*	*venison*	*insides*	*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	×							×			×			×			×										

Relation $\mathcal{R}_{C_P C_C}$ (Table 7) represents relationship "person likes ingredients" and relation $\mathcal{R}_{C_R C_C}$ (Table 8) represents relationship "restaurant cooks national cuisine". In Tables 9, 10, 11, we can see factors of data tables C_P, C_R and C_C, respectively.

Table 8: $\mathcal{R}_{C_R C_C}$

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

Table 9: Factors of data table $\mathcal{C}_P$

F_P^i	*Extent*	*Intent*
F_P^1	{Adam, Ben, Dale, Frank}	{male}
F_P^2	{Adam, Emily, Frank}	{American}
F_P^3	{Carol, Emily, Gabby}	{female}
F_P^4	{Ben, Carol}	{European}
F_P^5	{Dale, Gabby}	{Asian}

Table 10: Factors of data table $\mathcal{C}_R$

F_R^i	*Extent*	*Intent*
F_R^1	{Restaurant 4, Restaurant 5}	{ordinal, cheap}
F_R^2	{Restaurant 1, Restaurant 2}	{luxury, expensive}
F_R^3	{Restaurant 3}	{luxury, cheap}

Table 11: Factors of data table $\mathcal{C}_C$

F_C^i	*Extent*	*Intent*
F_C^1	{Chinese, French, Spanish, Mexican, American, German}	{1, 3, 15, 16, 17}
F_C^2	{Greek, Spanish, Italian}	{1, 2, 3, 4, 8, 9, 10}
F_C^3	{French, Czech}	{1, 10, 11, 12, 15, 16, 17, 21, 22, 23}
F_C^4	{Chinese, Indian, Spanish, Mexican, Italian, Japanese}	{1, 3, 4, 14}
F_C^5	{Greek, French, Indian}	{1, 3, 4, 6, 7}
F_C^6	{Chinese}	{1, 3, 4, 5, 12, 13, 14, 15, 16, 17, 18, 19, 20, 25}
F_C^7	{Italian, American}	{1, 3, 4, 11, 27}
F_C^8	{Greek, Czech, Mexican}	{1, 2, 5}
F_C^9	{Indian, Mexican}	{1, 2, 3, 4, 13, 14, 17}
F_C^{10}	{Czech, Itelian, German}	{1, 2, 12}
F_C^{11}	{Czech, , American}	{1, 15, 16, 17, 26}
F_C^{12}	{Greek}	{1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 19}
F_C^{13}	{Greek, French, Spanish, Italian}	{1, 3, 4, 9, 10}
F_C^{14}	{Chinese, Czech}	{1, 5, 12, 15, 16, 17, 20}
F_C^{15}	{French, Czech, German}	{1, 12, 15, 16, 17, 22}
F_C^{16}	{Mexican}	{1, 2, 3, 4, 5, 13, 14, 15, 16, 17, 24}
F_C^{17}	{Chinese, Itelian}	{1, 3, 4, 12, 14, 25}

One of the relational factors, which we get by 0.5-approach, is $\langle F_P^1, F_C^{11}, F_R^3 \rangle$ and could be interpreted as "men would enjoy eating in luxury restaurants where the meals are cheap". Another factor is $\langle F_P^3, F_C^2, F_R^1 \rangle$ and could be interpreted as "women enjoy eating in ordinal cheap restaurants".

4.2 Representation of connection between factors

We can represent the relational factors via graph (n-partite). See Figure 1, which presents the results from the previous example. Each group of nodes (F_P^i, F_C^i, F_R^i) 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, $\langle F_P^2, F_C^3, F_R^1 \rangle$ is a relational factor because there is an edge between nodes F_P^2 and F_C^3 and between F_C^3 and F_R^1.

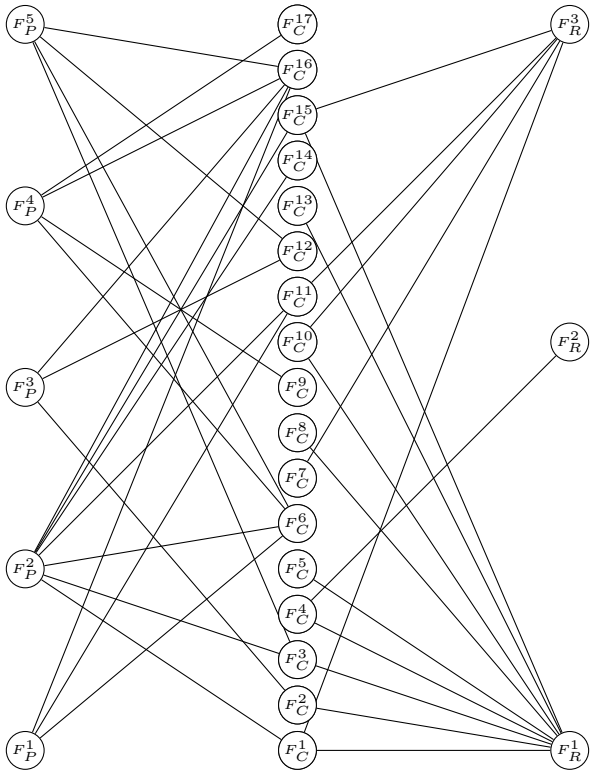

Fig. 1: Representation factors connections via graph.

5 Conclusion and Future Research

In this paper we present the new approach to BMF of multi-relational data, i.e. data which are composed from many data tables and relations between them. 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.

A future research shall include the following topics: generalization multi-relational Boolean factorization for ordinal data, especially data over residuated lattices. Design an effective algorithm for computing relational factors. Develop new approaches for connecting factors which utilize statistical methods and last but not least drive factor selection in the second data table, using information about factors in the first one and relation between them, for obtaining more relevant data.

Acknowledgment We acknowledge support by the Operational Program Education for Competitiveness Project No. CZ.1.07/2.3.00/20.0060 co-financed by the European Social Fund and Czech Ministry of Education.

References

1. Bartholomew D. J., Knott M.: Latent Variable Models and Factor Analysis, 2nd Ed., London, Arnold, 1999.
2. 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.
3. Ganter B., Wille R.: Formal Concept Analysis: Mathematical Foundations. Springer, Berlin, 1999.
4. Geerts F., Goethals B., Mielikäinen T.: Tiling databases, Proc. Discovery Science 2004, pp. 278–289.
5. 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,.
6. Harman H. H.: Modern Factor Analysis, 2nd Ed. The Univ. Chicago Press, Chicago, 1970.
7. 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.
8. Kim K.H.: Boolean Matrix Theory and Applications. Marcel Dekker, New York, 1982.
9. 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.
10. Lucchese C., Orlando S., Perego R.: Mining top-K patterns from binary datasets in presence of noise, SIAM DM 2010, pp. 165–176.
11. Miettinen P.: On Finding Joint Subspace Boolean Matrix Factorizations. Proc. SIAM International Conference on Data Mining (SDM2012), pp. 954-965, 2012.
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. 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.
14. 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.
15. Xiang Y., Jin R., Fuhry D., Dragan F. F.: Summarizing transactional databases with overlapped hyperrectangles, Data Mining and Knowledge Discovery 23(2011), 215–251.