Factor analysis with ordinal attributes

Markéta Trnečková

Dissertation

Faculty of Science
Palacký University Olomouc
2016

Author
Markéta Trnečková (nee Krmelová)
Department of Computer Science
Faculty of Science
Palacký University Olomouc
17. listopadu 12
CZ–771 46 Olomouc
Czech Republic
www.marketa-trneckova.cz
marketa.trneckova@gmail.com

Keywords
Matrix decomposition, Factor analysis, Ordinal data, Fuzzy logic

Declaration
Hereby I declare that the thesis is my original work.

Some parts of this thesis are based on outcomes of the joint scientific work with Radim Bělohlávek (radim.belohlavek@acm.org) (Chapters 2, 3, 4, 5 and 6). All authors have even share in the results and findings contained in the respective parts.

Markéta Trnečková

Abstract The problem of matrix decomposition, also known as matrix factorization problem, is widely investigated in data mining community. Especially Boolean case, where entries of matrices are 0s and 1s. In this thesis we explore the extension of matrix decomposition problem for ordinal data, i.e. data where attributes are values from ordered scales. The replacement of the two-element set of Boolean values and Boolean operations by a multiple-valued set of grades and multiple-valued operations introduced various non-trivial problems. We examine existing algorithms for ordinal data and propose three new algorithms for matrix decomposition problem. We demonstrate that the proposed algorithms deliver decompositions with informative and easy-to-understand factors by analysing real datasets. Moreover, we also compare algorithms presented on synthetic datasets.

Acknowledgements

I would like to thank to my whole family and especially to my husband Martin for their support and love. My gratitude also belongs to my advisor, prof. RNDr. Radim Bělohlávek, DSc and Mgr. Petr Osička, Ph.D. for their valuable comments and advices to this thesis.

This thesis is supported by grant No. GA15-17899S of the Czech Science Foundation and by grant No. PrF_2016_027 of IGA of Palacký University Olomouc.

Contents

1 Introduction **1**
- 1.1 Problem setting . . . 1
- 1.2 Related work . . . 3

2 Preliminaries **7**
- 2.1 Fuzzy logic . . . 7
- 2.2 Decomposition problem and its two variants . . . 9
- 2.3 Formal concept analysis . . . 12
- 2.4 Errors in decomposition . . . 13

3 First observations **15**
- 3.1 Variants of decomposition problem in the general case . . . 15
- 3.2 Decomposition problem as a covering problem . . . 17
- 3.3 Role of entries in matrix . . . 18
- 3.4 Explanation of data by factors . . . 20
  - 3.4.1 General case . . . 21
  - 3.4.2 Selection of rows from dataset . . . 22

4 Previous algorithms **23**
- 4.1 Boolean factorization of ordinally scaled attributes . . . 23
- 4.2 Previous algorithms for ordinal data . . . 25
  - 4.2.1 GreCon$_L$ . . . 25
  - 4.2.2 GreConD$_L$ . . . 26
  - 4.2.3 Statistical methods . . . 26

5 New algorithms **31**
- 5.1 GreEss$_L$ . . . 31
  - 5.1.1 Essential parts of matrices over scales . . . 31

5.1.2 GreEss$_L$ algorithm 34
5.2 Asso$_L$ 37
5.2.1 Association matrix 37
5.2.2 Procedures Cover and Asso$_L$ 39
5.3 GreConD$_L$+ 42
5.3.1 Algorithm GreConD$_L$+ 43

6 Experimental evaluation **47**
6.1 Illustrative example 47
6.1.1 Results for GreConD$_L$ 49
6.1.2 Results for ordinal scaling 50
6.1.3 Results for NMF 52
6.1.4 Results for GreEss$_L$ 54
6.1.5 Results for Asso$_L$ 54
6.1.6 Results for GreConD$_L$+ 56
6.1.7 Choice of the scale of degrees 57
6.2 Real data 60
6.2.1 Evaluation 64
6.3 Synthetic data 68
6.3.1 Evaluation of explanation data 69
6.3.2 Selection of smaller I from J 73
6.3.3 Role of τ in Asso$_L$ algorithm 76

7 Conclusion **79**

Summary in Czech **81**

Bibliography **83**

Chapter 1

Introduction

1.1 Problem setting

Factor analysis and related techniques based on matrix decompositions are important methods of data analysis. In the past, considerable attention has been paid to the problem of Boolean matrix factorization (BMF) and its variants, because of its direct usefulness in data analysis and its role in understanding Boolean data.

The basic problem is to find for a given $n \times m$ Boolean matrix I, some $n \times k$ and $k \times m$ Boolean matrices A and B with a reasonably small k for which the Boolean product $A \circ B$ is (approximately) equal to I.

In this thesis, we are concerned with extending the problems and methods of BMF toward a more general case. Namely, instead of Boolean matrices whose entries are 0s and 1s, we consider matrices with entries taken from a partially ordered set L bounded by 0 and 1, such as for example the five-element scale $L = \{0, \frac{1}{4}, \frac{1}{2}, \frac{3}{4}, 1\}$. The entries of a Boolean matrix I represent presence ($I_{ij} = 1$) and absence ($I_{ij} = 0$) of attributes. In the more general case, the entries represent degrees to which attributes are present, i.e. degrees to which they apply to objects, with 0 and 1 representing full absence and full presence and the intermediate degrees, such as $\frac{3}{4}$, representing partial presence.

Several methods for real-valued matrices exist. The best known are for example singular value decomposition and principal component analysis. These methods are widely used but the produced results are often hard to interpret, because of a possible presence of negative coefficients. Another well-known method, non-negative matrix factorisation, deals with this issue, but inter-

pretation of results is not quite straightforward either.

Papers [5, 18] extended the Boolean matrix factorization problem and the methods developed in [19] to ordinal data. This thesis provides an overview of existing methods, presents three new algorithms inspired by the existing BMF algorithms and compares them. Particular parts (Chapters 2, 3, 4, 5 and 6) of this thesis are mainly based on the following articles:

[7] R. Belohlavek, M. Krmelova, "Factor Analysis of Sports Data via Decomposition of Matrices with Grades", In: Szathmary L., Priss U. (Eds.): CLA 2012: Proceedings of the 9th International Conference on Concept Lattices and Their Applications, 2012, pp. 293–304 Fuengirola (Málaga), Spain, October 2012,

[8] R. Belohlavek, M. Krmelova, "Beyond Boolean Matrix Decompositions: Toward Factor Analysis and Dimensionality Reduction of Ordinal Data", ICDM 2013, pp. 961–966, 2013,

[9] R. Belohlavek, M. Krmelova, "Factor Analysis of Ordinal Data via Decomposition of Matrices with Grades", *Annals of Mathematics and Artiffcial Intelligence* 72 (1–2) (2014), 23–44,

[14] R. Belohlavek, M. Trneckova, "The Asso algorithm for graded attributes", Unpublished manuscript,

[15] R. Belohlavek, M. Trneckova, "Toward a geometry of decompositions of matrices with grades", Unpublished manuscript,

[16] R. Belohlavek, M. Trneckova, "A decomposition algorithm for matrices with grades that admits overcovering", Unpublished manuscript.

[7] presents analyses of various sports datasets using the algorithm presented in [18]. The aim of [7] is to demonstrate that the method yields reasonable factors and explain in detail how the factor model and the factors are to be understood. [9] is an extended version of [7], we provided here extensive experimentation and in addition, we proposed ways to address questions regarding the ability to explain a given data by factors obtained from this or possibly different data described by the same attributes. In [8, 14, 15, 16] we present new theorems regarding decompositions of matrices with ordinal data and propose new algorithms based on these results along with an experimental evaluation.

The full list of my publications can be found at my personal webpages `http://www.marketa-trneckova.cz`.

This thesis consists of seven chapters. The first chapter contains a brief introduction to this work and also contains the list of my publications relevant to this thesis and a brief survey of related works. The second chapter defines the problem this thesis is dealing with and lists the used notation. In the third chapter we present first observations related to new theory behind the presented algorithms. The fourth chapter contains a brief description of existing algorithms that will be present in the experimental part of this thesis (Chapter 6) together with the new algorithms presented in Chapter 5. Chapter 5, the main part of this thesis, comprises description of three new algorithms, their definition and same theoretical insight behind them. Chapter 6, the experimental part of this thesis, consists of several experiments. Shows how all presented methods work and their results on a small illustrative example and provides us results of various experiments on both real and synthetics datasets. The thesis is closed by Chapter 7 containing a summary of the work.

1.2 Related work

This section summarizes the works directly related to the topics in this thesis. The main part of this thesis is devoted to matrix decompositions—factorization of a matrix into a product of two or more matrices. Roots of this decompositions lays in factor analysis, which aims is to find new hidden variables (factors) in data. Factor analysis was initiated in 1904 by Charles Spearman [55], when he wanted to determine whether there are common factors of human intelligence. He tested how well people performed on various tasks relating to intelligence.

Since the literature on matrix decompositions is too numerous we present here only a little part of it. Perhaps the best known methods designed for real-valued matrices are singular value decomposition (SVD) [56], principal component analysis (PCA) [26, 31], independent component analysis (ICA) [23] and network component analysis (NCA) [40]. These methods usually decompose $n \times m$ real input matrix I into a product of two (in case of PCA, ICA, NCA, NMF) or three matrices (in SVD). The constraints of each method are different. For example for PCA, we want one of the resulted matrices to be

orthogonal, in ICA we require all components of the resulting matrix independent. There exist many applications using these methods, for example image processing and compression [2] or data reduction [24]. When using these techniques, some issues appears—such as a difficulty to interpret negative coefficients. This problem is solved by the well-known non-negative matrix factorization (NMF) [38]. Even though NMF is conceptually very different from the methods that we propose, a comparison seems worth performing. Applications of NMF are numerous, let us mention several of them. Text mining—analysis of document-term matrix (constructed usually as weighted word frequency in a set of documents)—[50] analyse small subset of scientific abstracts from PubMed database, [51] clusters Wikipedia articles and scientific journals based on the citations. Another application is spectral analysis, for example classification of space objects and debris [21], or bioinformatics applications such as for example gene expression [58] and identify common patterns of mutations that occur in cancers [1].

The data mining community pays attention to Boolean matrix factorisation, which is the most related to this work. One of the first paper in this area is [49] in which NP-completeness of the basic decomposition problem is observed. The interest in BMF in data mining is due to Miettinen's works, especially [46] with the Asso algorithm whose extension for matrices with scales we propose. Another Miettinen's works related to BMF include Boolean CX and CUR decompositions (different kind of decomposition) [43], investigating sparcity in BMF [44], examining common factor of two and more matrices [45], selecting the number of factors using minimum description length (MDL) [47]. [29] is the first paper on "tiling" Boolean data, which is closely related to BMF since it corresponds to the from-below factorizations that we examine for matrices with scales.

The utilization of formal concepts (fixpoints of Galois connections) of Boolean matrices as factors, two BMF algorithms—a GreCon and a GreConD algorithms—and other issues are examined in [19]. One of these issues is a transformation between the space of attributes and the space of factors. This is used in machine learning for classification of Boolean data [10, 11]. Another paper about BMF related to our work is [12], which proposed the GreEss algorithm based on essential elements, which we generalize in this work. Not yet published is [13], which includes the algorithm GreConD+, a modification of the algorithm GreConD which allows for overcover error. [62] studies summarization of Boolean data and proposes an algorithm uti-

lizing MDL called PaNDa, which is the algorithm for mining top-k patterns in Boolean data in [41] (the problems are naturally reformulated as BMF problems). Modification of PaNDa algorithm with using several different cost functions called PaNDa+ algorithm was proposed in [42]. Another algorithm called Nassau utilizes the MDL principle for solving BMF problem (in different way that PaNDa) was presented in [35].

In this thesis we are interested in a more general case namely in factorising matrices with entries from an appropriate scale. Matrices over scales and other structures are examined in many papers, including those on matrices over semi-ring-like algebras [30] and binary fuzzy relations between finite universes, see e.g. [3, 32].

Directly related to this paper are also [5, 18], where the the role of formal concepts of matrices over scales is studied and a decomposition algorithms are proposed. [9] presents analyses of various sports datasets using this algorithm and studies further theoretical problems inspired by the analyses. For algorithms Asso_L and GreEss_L presented in this thesis we refer to [8, 14, 15] and for algorithm GreConD_L+ we refer to [16].

A theoretical basis of this work lays in formal concept analysis (of Boolean data) [28], ordered and combinatorial structures [54] and closure structures in the setting of fuzzy logic and structures over scales [3]. The scales with aggregation we utilize in our work have recently been investigated in the context of formal fuzzy logic [32, 33].

Methods of analysis of ordinal data also appear in the psychological literature but the tools employed are basically variations of classical factor analysis. That is, grades are represented by and treated like numbers which leads to loss of interpretability, similarly as in the case of Boolean data, see e.g. [59].

Possible extension of factor analysis is multi-relational factor analysis. In specific form was mentioned in [45] as joint subspace matrix factorization, where are two Boolean matrices and both share the same rows (or columns). Another paper related to this topics is [34], where is introduced the relational formal concept analysis, i.e. the formal concept analysis on multi-relational data. The multi-relational data are iteratively merged into one data table and than processed. The most relevant papers for this extension are [37, 60, 61], where was presented factorisation of multi-relational data. Also a heuristic algorithm was presented there.

Chapter 2

Preliminaries

This chapter describes the notation and calculus used in this thesis. We start with fuzzy logic, define problem of matrix decomposition and conclude with formal concept analysis which we mainly use for solving the decomposition problem.

2.1 Fuzzy logic

Fuzzy logic has been employed to handle the concept of partial truth, where the truth value may range between completely true and completely false. This approach has been proven to be useful in several areas and we utilize it in our work. The content of this section is based on [3].

Let us consider a set L of truth values. We assume that this set is partially ordered (partial ordering is denoted by $\leq$), contains a least element 0 and a greatest element 1.

Let a and b truth degrees from L, then in L exists a truth value which is greater than both a and b. The least element that is greater or equal to both a and b is called *supremum* of a and b. Analogously, we can define *infimum* of a and b—the greatest element from L which is smaller or equal to both a and b. We define the *lower cone* of A by $\mathcal{L}(A) = \{a \in L | a \leq b \text{ for all } b \in A\}$ and the *upper cone* of A by $\mathcal{U}(A) = \{a \in L | b \leq a \text{ for all } b \in A\}$. If $\mathcal{L}(A)$ has a greatest element a, then a is called the *supremum* of A (denoted $\bigvee A$) and dually if $\mathcal{U}(A)$ has a least element a, then a is called the *infimum* of A (denoted $\bigwedge A$). In particular, we assume that the partial order $\leq$ makes L a complete lattice [32] (i.e., arbitrary infima $\bigwedge$ and suprema $\bigvee$ exist in

L). This assumption is automatically satisfied if L is a finite chain (i.e. $a \leq b$ or $b \leq a$ for every $a, b \in L$), in which case $a \wedge b = \min(a, b)$ and $a \vee b = \max(a, b)$. We also need to define operation logical conjunction (denoted by $\otimes$). We assume that $\otimes$ is commutative, associative, has 1 as its neutral element ($a \otimes 1 = a = 1 \otimes a$), and distributes over arbitrary suprema, i.e. $a \otimes (\bigvee_{j \in J} b_j) = \bigvee_{j \in J}(a \otimes b_j)$. This leads to if a and b are truth degrees of propositions p_1 and p_2, then $a \otimes b$ is the truth degree of proposition "p_1 and p_2".

Importantly, $\otimes$ induces another operation, $\rightarrow$, called the *residuum of* $\otimes$, which plays the role of the truth function of implication and is defined by

$$a \rightarrow b = \max\{c \in L \,|\, a \otimes c \leq b\}. \tag{2.1}$$

Residuum, which may be looked at as a kind of division, satisfies an important technical condition called adjointness:

$$a \otimes b \leq c \text{ iff } a \leq b \rightarrow c,$$

which is also utilized below. This leads to algebraic structures called *residuated lattices*.

Definition 1. *A residuated lattice is an algebra* $\mathbf{L} = \langle L, \wedge, \vee, \otimes, \rightarrow, 0, 1\rangle$ *where*

(i) $\langle L, \wedge, \vee, 0, 1\rangle$ *is a lattice with a least element 0 and a greatest element 1,*

(ii) $\langle L, \otimes, 1\rangle$ *is a comutative monoid i.e.* $\otimes$ *is associative, commutative, and the identity* $x \otimes 1 = x$ *holds,*

(iii) $\otimes$ *and* $\rightarrow$ *satisfy the adjointness property, i.e.*

$$x \leq y \rightarrow z \text{ iff } x \otimes y \leq z$$

holds for each $x, y, z \in L$ *(*$\leq$ *denotes the lattice ordering).*

A residuated lattice is called *complete* if $\langle L, \wedge, \vee, 0, 1\rangle$ is a complete lattice.

Many examples of scales are known in many-valued logic [32, 33], among them those where L is the real unit interval $[0, 1]$ or its finite equidistant

subinterval, i.e. $L = \{0, \frac{1}{n}, \ldots, \frac{n-1}{n}, 1\}$, which are used in examples and experiments presented in the thesis.

Examples of $\otimes$ include
Łukasiewicz t-norm defined by

$$a \otimes b = \max(0, a + b - 1),$$

whose residuum is

$$a \to b = \min(1, 1 - a + b),$$

Gödel

$$a \otimes b = \min(a, b),$$

whose residuum is

$$a \to b = \begin{cases} 1 & \text{if } a \leq b \\ b & \text{if } a > b \end{cases}$$

and Goguen conjunction

$$a \otimes b = a \cdot b,$$

whose residuum is

$$a \to b = \begin{cases} 1 & \text{if } a \leq b \\ \frac{b}{a} & \text{if } a > b \end{cases}$$

As far as the choice of the operations on L is concerned, we mainly use Łukasiewicz in examples, because of some of its intuitive properties. For example, the implication $\to$ naturally corresponds to the natural distance in $[0, 1]$.

2.2 Decomposition problem and its two variants

Factor analysis is a method used to describe variability among observed, correlated variables in terms of a potentially smaller number of unobserved variables which are called factors. For example, it is possible that variations in several observed variables (such as performance of students) mainly reflect the variations in an unobserved variable (their intelligence).

Formally, the input data is represented by an $n \times m$ object–attribute matrix I and the "explanation" means a decomposition

$$I = A \circ B \tag{2.2}$$

(exact or approximate) of I into a product $A \circ B$ of an $n \times k$ object–factor matrix A and a $k \times m$ factor–attribute matrix B. What kind of matrices (real, Boolean, or other) and what kind of product $\circ$ are involved determines the semantics of the factor model.

Now we present two concrete variants of the decomposition problem. These two problems reflect two important views on BMF. The first one—the *discrete basis problem* (DBP) [46]—emphasizes the importance of the first k (presumably the most important) factors. The second one—the *approximate factorization problem* (AFP) [12]—emphasizes the need to account for (and thus to explain) a prescribed portion of data, which is specified by error ε.

Formally DBP is defined as follows:
Given $n \times m$ matrix I and positive integer k, find $n \times k$ matrix A and $k \times m$ matrix B that minimize $||I - A \circ B||$.

AFP is defined as follows:
Given $n \times m$ matrix I and prescribed error ε, find $n \times k$ matrix A and $k \times m$ matrix B with k as small as possible such that that minimize $||I - A \circ B|| \leq \varepsilon$.

Several other reasonable variants may be formulated but we restrict to these two because they reflect two basic views of the decomposition problem.

Our model (2.2) involves matrices containing degrees (or grades) of certain scales L and the product is the sup-$\otimes$ product, as described below. In particular, the matrix entry I_{ij} is a degree to which attribute j applies to object i, for example $I_{ij} = 0.5$. Similarly, A_{il} is the degree to which factor l applies to object i and B_{lj} is the degree to which attribute j is (one particular) manifestation of factor l. The case in which the scale L contains only two degrees, 0 and 1, called the Boolean case in what follows, corresponds to Boolean matrices and Boolean factor analysis [19] which is a special case of ours.

A verbal description of equation (2.2) reads:

> Object i has attribute j if and only if
> there exists factor l such that i has l (or, l applies to i) (2.3)
> and j is one of the particular manifestations of l.

Such description is certainly appealing and well understandable.

In the Boolean case, in which $L = \{0, 1\}$, the verbal description leads to

$(A \circ B)_{ij} = 1$ iff there exists $l \in \{1, \ldots, k\}$ such that $A_{il} = 1$ and $B_{lj} = 1$,

which may equivalently be described by the well-known formula

$$(A \circ B)_{ij} = \max_{l=1}^{k} \min(A_{il}, B_{lj}) \tag{2.4}$$

for Boolean matrix composition.

With a general scale L, we approach the situation according to the principles of (mathematical) fuzzy logic (see Section 2.1) as follows. Let us have the formulas $\varphi(i,l)$ saying "object i has factor l" and $\psi(l,j)$ saying "attribute j is a manifestation of factor l", and consider A_{il} the truth degree of $\varphi(i,l)$ and B_{lj} the truth degree of $\psi(l,j)$, i.e.

$$||\varphi(i,l)|| = A_{il} \text{ and } ||\psi(l,j)|| = B_{lj}. \tag{2.5}$$

Now, according to fuzzy logic, the truth degree of formula $\varphi(i,l) \,\&\, \psi(l,j)$ which says "object i has factor l and attribute j is a manifestation of factor l" is computed by

$$||\varphi(i,l) \,\&\, \psi(l,j)|| = ||\varphi(i,l)|| \otimes ||\psi(l,j)||$$

where $\otimes : L \times L \to L$ is a truth function of many-valued conjunction &, and hence the truth degree of $(\exists l)(\varphi(i,l)\&\psi(l,j)$ which says "there exists factor l such that object i has l and attribute j is a manifestation of l", i.e. the proposition involved in (2.3), is computed by

$$||(\exists)(\varphi(i,l) \,\&\, \psi(l,j))|| = \bigvee_{l=1}^{k} ||\varphi(i,l)|| \otimes ||\psi(l,j)||, \tag{2.6}$$

where $\bigvee$ denotes the supremum. Given into account (2.5), we see that a generalization of (2.4) to the case of possibly intermediate degrees is given by

$$(A \circ B)_{ij} = \bigvee_{l=1}^{k} A_{il} \otimes B_{lj}. \tag{2.7}$$

Therefore, with $\circ$ given by (2.7), the factor model (2.2) retains its meaning (2.3) even in the case when intermediate degrees are allowed.

Example 1. *With Łukasiewicz t-norm, let $I = A \circ B$:*

$$\begin{pmatrix} 0.5 & 1.0 & 0.0 \\ 0.0 & 0.5 & 0.5 \\ 0.0 & 0.5 & 0.0 \end{pmatrix} = \begin{pmatrix} 1.0 & 0.5 \\ 0.5 & 1.0 \\ 0.5 & 0.5 \end{pmatrix} \circ \begin{pmatrix} 0.5 & 1.0 & 0.0 \\ 0.0 & 0.5 & 0.5 \end{pmatrix}$$

2.3 Formal concept analysis

From the description in Section 2.2, it is clear that for any decomposition (2.2), the lth factor ($l \in \{1, \ldots, k\}$) is represented by two parts: the lth column $A_{\_l}$ of A and the lth row $B_{l\_}$ of B. As shown in [5], optimal factors for a decomposition of I (see below) are provided by formal concepts associated to I. In detail, let $X = \{1, \ldots, n\}$ (objects) and $Y = \{1, \ldots, m\}$ (attributes). Recall that a formal concept (*formal fuzzy concept*) of I is any pair $\langle C, D \rangle$ of L-sets (fuzzy sets) $C : \{1, \ldots, n\} \to L$ of objects and $D : \{1, \ldots, m\} \to L$ of attributes, see [4], that satisfies $C^{\uparrow} = D$ and $D^{\downarrow} = C$ where $^{\uparrow} : L^X \to L^Y$ and $^{\downarrow} : L^Y \to L^X$ are the concept-forming operators defined by

$$C^{\uparrow}(j) = \bigwedge_{i \in X} (C(i) \to I_{ij}) \text{ and } D^{\downarrow}(i) = \bigwedge_{j \in Y} (D(j) \to I_{ij}).$$

The set of all formal concepts of I is denoted by $\mathcal{B}(X, Y, I)$ or just $\mathcal{B}(I)$. The set $\mathcal{B}(I) = \{\langle C, D \rangle \mid C^{\uparrow} = D, D^{\downarrow} = C\}$ equipped with a partial order $\leq$, defined by $\langle C_1, D_1 \rangle \leq \langle C_2, D_2 \rangle$ iff $C_1 \leq C_2$ (iff $D_2 \leq D_1$), forms a complete lattice, called the *concept lattice* of I. The fuzzy set C is called *extent* and the fuzzy set D is called *intent*. $C(i) \in L$ is interpreted as the degree to which factor l applies to object i and $D(j) \in L$ is the degree to which attribute j is a manifestation of l.

Example 2. *Let us have $X = \{a, b, c\}$, $Y = \{1, 2, 3\}$ and matrix I from Example 1, i.e.*

$$I = \begin{pmatrix} 0.5 & 1.0 & 0.0 \\ 0.0 & 0.5 & 0.5 \\ 0.0 & 0.5 & 0.0 \end{pmatrix}$$

the set of all formal concepts and the corresponding concept lattice can be seen in Table 2.1 and Figure 2.1, respectively.

Optimality of using formal concepts as factors means the following. Let for a set

$$\mathcal{F} = \{\langle C_1, D_1 \rangle, \ldots, \langle C_k, D_k \rangle\} \subseteq \mathcal{B}(I) \tag{2.8}$$

of formal concepts denote by $A_{\mathcal{F}}$ and $B_{\mathcal{F}}$ the matrices defined by

$$(A_{\mathcal{F}})_{il} = (C_l)(i) \quad \text{and} \quad (B_{\mathcal{F}})_{lj} = (D_l)(j). \tag{2.9}$$

F_i	Extent	Intent
C_0	$\{^1/a, ^1/b, ^1/c\}$	$\{^0/1, ^{0.5}/2, ^0/3\}$
C_1	$\{^{0.5}/a, ^1/b, ^{0.5}/c\}$	$\{^0/1, ^{0.5}/2, ^{0.5}/3\}$
C_2	$\{^1/a, ^{0.5}/b, ^{0.5}/c\}$	$\{^{0.5}/1, ^1/2, ^0/3\}$
C_3	$\{^{0.5}/a, ^{0.5}/b, ^{0.5}/c\}$	$\{^{0.5}/1, ^1/2, ^{0.5}/3\}$
C_4	$\{^{0.5}/a, ^0/b, ^0/c\}$	$\{^1/1, ^1/2, ^{0.5}/3\}$
C_5	$\{^0/a, ^{0.5}/b, ^0/c\}$	$\{^{0.5}/1, ^1/2, ^1/3\}$
C_6	$\{^0/a, ^0/b, ^0/c\}$	$\{^1/1, ^1/2, ^1/3\}$

Table 2.1: Example 2: All formal concepts

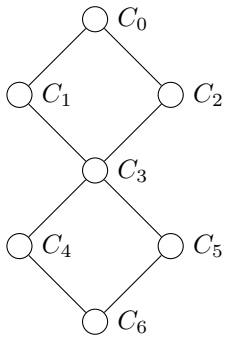


Figure 2.1: $\mathcal{B}(I)$

Then, whenever $I = A \circ B$ for $n \times k$ and $k \times m$ matrices A and B, there exists a set $\mathcal{F} \subseteq \mathcal{B}(I)$, $|\mathcal{F}| \leq k$ such that $I = A_{\mathcal{F}} \circ B_{\mathcal{F}}$, i.e. the optimal decompositions are attained by formal concepts as factors. Proof of this claim is below (Equation 2).

By $\text{rank}_{\mathbf{L}}(I)$ we denote the smallest k for which the above decomposition of I exists and call it the *(L-)rank* of I.

For two matrices $J_1, J_2 \in L^{n \times m}$ we put

$$J_1 \leq J_2 \text{ iff } (J_1)_{ij} \leq (J_2)_{ij} \text{ for every } i, j \tag{2.10}$$

in which case we say that J_1 is *contained* in J_2. $J \in L^{n \times m}$ is called a *rectangle* if $J = C \circ D$ for some column $C \in L^{n \times 1}$ and row $D \in L^{1 \times m}$. Note that in the Boolean case, rectangles are just tiles in terms of [29], i.e. rectangular areas filled with 1s. Unlike the Boolean case, the C and D for which $J = C \circ D$ are not unique. We say that a rectangle J *covers* $\langle i, j \rangle$ in I if $J_{ij} = I_{ij}$.

2.4 Errors in decomposition

When we desire exact decomposition, using formal concept as factors is beneficial, but it has a limitation—it never commit overcovering—when approximate factorization is needed. For factor model 2.2, we are talking about *uncovering* when $I_{ij} > (A \circ B)_{ij}$ and *overcovering* when $I_{ij} < (A \circ B)_{ij}$.

The error function E (distance) between I and approximate decomposition $(A \circ B)$ is sum of two components—E_u and E_o denoting uncover error and overcover error respectively, i.e $E = E_u + E_o$. Uncover and overcover

errors may be defined as follows

$$E_u = \sum_i \sum_j 1 - (I_{ij} \to (A \circ B)_{ij}),$$

$$E_o = \sum_i \sum_j 1 - ((A \circ B)_{ij} \to I_{ij}).$$

These two components are not symmetrical. While E_u can only decrease by adding more factors, E_o may only increase. This fact was presented in boolean case in [12].

Observation 1. *Let $A' \in L^{n\times(k+1)}$ and $B' \in L^{(k+1)\times m}$ result by adding a single column and row, respectively. Then $E_u(I, A' \circ B') \leq E_u(I, A \circ B)$ and $E_o(I, A' \circ B') \geq E_o(I, A \circ B)$.*

Chapter 3

First observations

This chapter provides first observations that lead to deeper theoretical insight to below presented algorithms. Results presented here are based on [7, 8, 9].

3.1 Variants of decomposition problem in the general case

In the previous chapter we describe two variants of decomposition problem, namely the discrete basis problem (DBP) and the approximate factorisation problem (AFP). In order to define generalization of the DBP a AFP problems for Boolean matrices to general problems over some scale L, we need to define closeness of matrices over L.

The first possible approach is to take as closeness of two matrices $I, J \in L^{n\times m}$ function

$$s_{=}(I,J) = \frac{\sum_{i,j=1}^{n,m} eq(I_{ij}, J_{ij})}{n \cdot m}.$$

Function $eq(a, b)$ here returns 1 if a is equal to b and 0 otherwise. In a sense, this is a pessimistic approach because it ignores the case where I_{ij} is close to but different from J_{ij}.

Let $s_L : L \times L \to [0, 1]$ be an appropriate function measuring closeness of degrees in L. For matrices $I, J \in L^{n\times m}$, put

$$s_{\approx}(I,J) = \frac{\sum_{i,j=1}^{n,m} s_L(I_{ij}, J_{ij})}{n \cdot m}, \tag{3.1}$$

i.e. $s_{\approx}(I, J) \in [0, 1]$ is the normalized sum over all matrix entries of the closeness of the corresponding entries in I and J. In general, we require

$s_L(a,b) = 1$ if and only if $a = b$, and $s_L(0,1) = s_L(1,0) = 0$, in which case $s_\approx(I,J) = 1$ if and only if $I = J$. We furthermore require that $a \leq b \leq c$ implies $s_L(a,c) \leq s_L(b,c)$. For the important case of L being a subchain of $[0,1]$, s_L may be defined by

$$s_L(a,b) = a \leftrightarrow b,$$

where $a \leftrightarrow b = \min(a \to b, b \to a)$ is the so-called *biresiduum* (many-valued equivalence from a logical point of view) of a and b (note that $\to$ is the residuum (2.1) of $\otimes$).

We use closeness because of its natural logical interpretation as a many-valued equivalence but, clearly, one could alternatively use distance instead of closeness.

In terms of above presented closeness, we now present generalisation of the two above presented problems of decomposition over scale L:

- *DBP(L)*: Given $I \in L^{n\times m}$ and a positive integer k, find $A \in L^{n\times k}$ and $B \in L^{k\times m}$ that maximize $s(I, A \circ B)$.

- *AFP(L)*: Given I and prescribed error $\varepsilon \in [0,1]$, find $A \in L^{n\times k}$ and $B \in L^{k\times m}$ with k as small as possible such that $s(I, A \circ B) \geq \varepsilon$.

As $s(I, A \circ B)$, we can take function $s_\approx$ or $s_=$.

In view of the provable difficulty of the AFP and DBP in the Boolean case [19, 46] and the remarks above, the following theorem is not surprising:

Theorem 1. *DBP(L) and AFP(L) are NP-hard optimization problems.*

Proof. The proof proceeds by adaptation of the proofs of NP-hardness of the AFP and DBP in the Boolean case, see [19] and [46]. We proceed for AFP(L) only, by showing that the restriction to instances with $\varepsilon = 1$ is NP-hard. Due to our assumptions, $s(I, A \circ B) \geq \varepsilon$ is equivalent to $A \circ B = I$ in this case. According to the definition of NP-hardness, it suffices to verify that the corresponding decision problem, Π, is NP-complete. Π consists in deciding whether for a given $I \in L^{n\times m}$ and k there exists $A \in L^{n\times k}$ and $B \in L^{k\times m}$ with $A \circ B = I$.

The Boolean version of Π is NP-complete because it is a reformulation (see e.g. [19]) of the set basis problem whose NP-completeness is due to [57]. To finish the proof it thus suffices to check that the restriction of Π to Boolean input matrices I is NP-complete. But the latter fact follows since

for a Boolean I, there exist $A \in L^{n\times k}$ and $B \in L^{k\times m}$ with $A \circ B = I$ iff there exist Boolean matrices $A \in \{0,1\}^{n\times k}$ and $B \in \{0,1\}^{k\times m}$ with $A \circ B = I$. Namely, if $A \circ B = I$ for $A \in L^{n\times k}$ and $B \in L^{k\times m}$ then $A' \circ B' = I$ for the Boolean A' and B' defined by $A'_{il} = 1$ if $A_{il} = 1$, $A'_{il} = 0$ if $A_{il} < 1$, and the same for B', which is easily seen from the isotony of $\otimes$. □

3.2 Decomposition problem as a covering problem

In Section 2.3, we present notation in formal concept analysis and present a definition of rectangles in I. The following lemma, which is easy to see, extends the observation in [5] and shows that an exact decomposition of I is equivalent to a coverage of entries in I by rectangles contained in I.

Lemma 1. *The following conditions are equivalent for any $I \in L^{n\times m}$:*

(a) *$I = A \circ B$ for some $A \in L^{n\times k}$ and $B \in L^{k\times m}$.*

(b) *There exist rectangles $J_1, \ldots, J_k \in L^{n\times m}$ such that $I = J_1 \vee \cdots \vee J_k$, i.e. $I_{ij} = \max_{l=1}^{k} (J_l)_{ij}$.*

(c) *There exist rectangles $J_1, \ldots, J_k \in L^{n\times m}$ contained in I such that every $\langle i, j\rangle$ in I is covered by some J_l.*

In particular, for the matrices A and B in (a), one may take the product of the lth column of A and the lth row of B to be the rectangle J_l in (b).

Importantly, Lemma 1 allows us to consider the problem of decomposition of I as a certain coverage problem, namely the problem of covering the entries in I by rectangles contained in I. Next we show that optimal in such coverage are rectangles that correspond to so-called formal concepts of I, which are fixpoints of certain operators and are studied in FCA, see [28] for the Boolean case and [4] for the general case with scales.

The following theorem shows that formal concepts of I are optimal factors for approximate decompositions of I that provide a *from-below approximation* of I, i.e. $A \circ B \leq I$ (note that these include exact decompositions $I = A \circ B$).

Theorem 2. *Let for $I \in L^{n\times m}$ there exist $A \in L^{n\times k}$ and $B \in L^{k\times m}$ such that $A \circ B \leq I$. Then there exists a set $\mathcal{F} \subseteq \mathcal{B}(I)$ of formal concepts of I with $|\mathcal{F}| \leq k$ such that for the $n \times |\mathcal{F}|$ and $|\mathcal{F}| \times m$ matrices $A_\mathcal{F}$ and $B_\mathcal{F}$ over L we have*

$$s(I, A_\mathcal{F} \circ B_\mathcal{F}) \geq s(I, A \circ B).$$

Proof. Since $A \circ B \leq I$, Lemma 1 implies that every rectangle $J_l = A_{\_l} \circ B_{l\_}$ is contained in I. Consider the pairs $\langle (A^{\mathrm{T}}_{\_l})^{\uparrow\downarrow}, (A^{\mathrm{T}}_{\_l})^{\uparrow}\rangle$. Every $\langle (A^{\mathrm{T}}_{\_l})^{\uparrow\downarrow}, (A^{\mathrm{T}}_{\_l})^{\uparrow}\rangle$ is a formal concept in $\mathcal{B}(I)$ (a well-known fact in FCA).

Moreover $A^{\mathrm{T}}_{\_l} \leq (A^{\mathrm{T}}_{\_l})^{\uparrow\downarrow}$, because $^{\uparrow\downarrow}$ is a closure operator. Since, $A_{\_l} \circ B_{l\_}$ is contained in I, a straightforward computation using adjointness of $\otimes$ and $\to$ implies $B_{l\_} \leq (A^{\mathrm{T}}_{\_l})^{\uparrow}$. Now consider the set

$$\mathcal{F} = \{\langle (A^{\mathrm{T}}_{\_1})^{\uparrow\downarrow}, (A^{\mathrm{T}}_{\_1})^{\uparrow}\rangle, \ldots, \langle (A^{\mathrm{T}}_{\_k})^{\uparrow\downarrow}, (A^{\mathrm{T}}_{\_k})^{\uparrow}\rangle\} \subseteq \mathcal{B}(I)$$

and the matrices $A_\mathcal{F}$ and $B_\mathcal{F}$. Clearly $\mathcal{F}$ contains at most k elements (it may happen $|\mathcal{F}| < k$). It is easy to check that the rectangle corresponding to $\langle (A^{\mathrm{T}}_{\_l})^{\uparrow\downarrow}, (A^{\mathrm{T}}_{\_l})^{\uparrow}\rangle$, i.e. the cross-product $(A_\mathcal{F})_{\_l} \circ (B_\mathcal{F})_{l\_}$, is contained in I and, due to the above observation, contains $J_l = A_{\_l} \circ B_{l\_}$. Hence,

$$A \circ B \leq \max_{l=1}^{k} J_l \leq \max_{l=1}^{k} (A_\mathcal{F})_{\_l} \circ (B_\mathcal{F})_{l\_} = A_\mathcal{F} \circ B_\mathcal{F} \leq I.$$

Since $a \leq b \leq c$ implies $s_L(a, c) \leq s_L(b, c)$, we readily obtain $s(I, A_\mathcal{F} \circ B_\mathcal{F}) \geq s(I, A \circ B)$, finishing the proof. □

3.3 Role of entries in matrix

We now examine in detail the coverage problem by rectangles, to which the decomposition problem may be transformed. An inspection of the concept lattice $\mathcal{B}(I)$ reveals an interesting fact—a possibility to differentiate the role of matrix entries for decompositions. In, particular, we identify a so-called *essential part* of I, a minimal set of entries whose coverage guarantees an exact decomposition of I. We show later that the number of such entries is significantly smaller than the number of all entries. Most importantly, the essential part may be seen as the part to focus on when computing decompositions. This view is studied in detail in Section 5.1.1 and is utilized in the design of a decomposition algorithm in Section 5.1.2.

Note that the idea of differentiating the role of entries is inspired by [12], but the situation is considerably more involved in the setting of scales compared to the Boolean case.

The results presented in this section are based on [8].

Definition 2. *$J \leq I$ is called an* essential part *of I if J is minimal w.r.t. $\leq$ having the property that for every $\mathcal{F} \subseteq \mathcal{B}(I)$, $J \leq A_{\mathcal{F}} \circ B_{\mathcal{F}}$ then $I = A_{\mathcal{F}} \circ B_{\mathcal{F}}$.*

In other words, the coverage of an essential part J by formal concepts of I guarantees the coverage of all entries in I. It turns out that certain intervals in $\mathcal{B}(I)$ play a crucial role for our considerations. For $C \in L^{1\times n}$, $D \in L^{1\times m}$, put

$$\gamma(C) = \langle C^{\uparrow\downarrow}, C^{\uparrow}\rangle \text{ and } \mu(D) = \langle D^{\downarrow}, D^{\downarrow\uparrow}\rangle,$$

and denote by $\mathcal{I}_{C,D}$ the interval

$$\mathcal{I}_{C,D} = [\gamma(C), \mu(D)]$$

in $\mathcal{B}(I)$, i.e. the set

$$[\gamma(C), \mu(D)] = \{\langle E, F\rangle \in \mathcal{B}(I) \mid \gamma(C) \leq \langle E, F\rangle \leq \mu(D)\}.$$

In particular, $\gamma(\{\,{}^{a}\!/\!_{x}\}) = \gamma(x, a)$ and $\mu(\{\,{}^{b}\!/\!_{y}\}) = \mu(y, b)$ are the mappings from the basic theorem of $\mathcal{L}$-concept lattices [3].

In Section 5.1.1 we will show that all the rectangles corresponding to the formal concepts in $\mathcal{I}_{C,D}$ cover the rectangle $C^{\mathrm{T}} \circ D$.

Now, for a given matrix $I \in L^{n\times m}$, let

$$\mathbf{I}_{ij} = \{\mathcal{I}_{\{{}^{a}\!/\!_{i}\},\{{}^{b}\!/\!_{j}\}} \mid a, b \in L, a \otimes b = I_{ij}\}$$

and put

$$\mathcal{I}_{ij} = \bigcup \mathbf{I}_{ij}.$$

Note that the situation is much easier in the Boolean case. Namely, if $I_{ij} > 0$, then $\mathcal{I}_{ij}$ consists of a single interval in the Boolean case because the only a and b for which $a \otimes b = 1$ are $a = b = 1$. In case of general scales, there may be several pairs of a and b for which $I_{ij} = a \otimes b$, hence several intervals of which $\mathcal{I}_{ij}$ consists, see Example 3.

Later in Section 5.1.1 we will prove important theorem which shows that $\mathcal{I}_{ij}$ is just the set of all formal concepts of I that cover $\langle i, j\rangle$.

Denote now by $\mathcal{E}(I) \in L^{n\times m}$ the matrix over L defined by

$$(\mathcal{E}(I))_{ij} = \begin{cases} I_{ij} & \text{if } \mathcal{I}_{ij} \text{ is } \neq \emptyset \text{ and minimal w.r.t. } \subseteq, \\ 0 & \text{otherwise.} \end{cases}$$

In Section 5.1.1, we will show that $\mathcal{E}(I)$ is an essential part of matrix I.

Example 3. *Let us have X, Y and I from Example 2. Than for entry $\langle b, 2\rangle$ we obtain two intervals. First one is bounded by factors $C_5 = \langle \{^{0.5}/b\}^{\uparrow\downarrow}, \{^{0.5}/b\}^{\uparrow}\rangle$ and $C_2 = \langle \{^{1}/2\}^{\downarrow}, \{^{1}/2\}^{\downarrow\uparrow}\rangle$ and second one is bounded by $C_1 = \langle \{^{1}/b\}^{\uparrow\downarrow}, \{^{1}/b\}^{\uparrow}\rangle$ and $C_0 = \langle \{^{0.5}/2\}^{\downarrow}, \{^{0.5}/2\}^{\downarrow\uparrow}\rangle$, see Figure 3.1.*

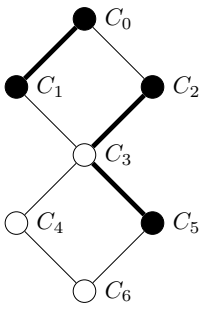


Figure 3.1: $\mathcal{B}(I)$ with intervals for entry $\langle b, 2\rangle$.

3.4 Explanation of data by factors

In this section we propose a way to address the following related problems. First, we want to determine what does it mean that a set of formal concepts explain well (or to a certain extent) a given dataset? Second, what does it mean that good factors of a given dataset explains well another dataset? The results in this section are based on paper [9].

If a set $\mathcal{F} \subseteq \mathcal{B}(X)$ of formal concepts of I satisfies $I = A_{\mathcal{F}} \circ B_{\mathcal{F}}$, we intuitively regard $\mathcal{F}$ as fully explaining the data represented by I and call $\mathcal{F}$ a set of *factor concepts*. In general, however, we are interested in $\mathcal{F}$ for which I is close to $A_{\mathcal{F}} \circ B_{\mathcal{F}}$, in particular if $\mathcal{F}$ is reasonably small. We can take into account above presented closeness $s_{\approx}$ and $s_{=}$ and say that $\mathcal{F}$ *explains* $100 \cdot s_{=}(I, A_{\mathcal{F}} \circ B_{\mathcal{F}})\%$ *of data* represented by I. Clearly, this means that

$100 \cdot s_{=}(I, A_{\mathcal{F}} \circ B_{\mathcal{F}})\%$ of all the $n \times m$ entries have the same values in I and $A_{\mathcal{F}} \circ B_{\mathcal{F}}$. Or analogously for $s_{\approx}$ we can say that entries from I and $A_{\mathcal{F}} \circ B_{\mathcal{F}}$ are in average $s_{\approx}(I, A_{\mathcal{F}} \circ B_{\mathcal{F}})$ close.

In the rest of the paper unless otherwise stated, we take $s_{\approx}$ as closeness function s.

3.4.1 General case

Let I and J be two matrices describing the sets X_1 and X_2 of objects by a common set Y of attributes. How can we answer the question of whether a set $\mathcal{F} \subseteq \mathcal{B}(I)$ of possibly good factors of I is a set of good factors of J? The concepts in $\mathcal{F}$ may not be directly used as concepts of J because for $\langle C, D\rangle \in \mathcal{F}$ we have $C \in L^{X_1}$ while we need $\in L^{X_2}$ for factors of J. A abundantly discussed the topicnatural option is to consider instead of $\mathcal{F}$ the set of concepts of J that are generated by the intents of the factors in $\mathcal{F}$, i.e. the set

$$\mathcal{F}^J = \{\langle D^{\downarrow_J}, D^{\downarrow_J\uparrow_J}\rangle \mid \langle C, D\rangle \in \mathcal{F}\}, \tag{3.2}$$

because the intents represent the meanings of concepts. One may then use $s_{=}(I, A_{\mathcal{F}_J} \circ B_{\mathcal{F}_J})$ or $s_{\approx}$ to asses how well the factors $\mathcal{F}$ of I explain the data represented by J.

Of a particular importance is the particular case when J results by adding rows to I (i.e. adding objects to those represented by I). Let us thus assume that $X_1 \subseteq X_2$ and that $I_{ij} = J_{ij}$ for $i \in X_1$ and $j \in Y$. We may proceed as above but the following observation presents a convenient simplification of the set $\mathcal{F}^J$.

Observation 1 *For the above notation,*

$$\mathcal{F}^J = \{\langle D^{\downarrow_J}, D\rangle \mid \langle C, D\rangle \in \mathcal{F}\}.$$

Proof. We need to show that every intent of I is an intent of J. We have $D = C^{\uparrow_I}$. Consider the L-set $E \in L^{X_2}$ defined by $E(i) = C(i)$ for $i \in X_1$ and $E(i) = 0$ for $i \in X_2 - X_1$. Then

$$\begin{aligned} E^{\uparrow_J}(j) &= \bigwedge_{i \in X_2} (E(i) \to J_{ij}) = \textstyle\bigwedge_{i \in X_1}(E(i) \to J_{ij}) \wedge \bigwedge_{i \in X_2 - X_1}(E(i) \to J_{ij}) \\ &= \textstyle\bigwedge_{i \in X_1}(C(i) \to I_{ij}) \wedge 1 = \bigwedge_{i \in X_1}(C(i) \to I_{ij}) = C^{\uparrow_I} = D, \end{aligned}$$

proving that D is an intent of J. □

Therefore an intent of a factor of I is also an intent of a possible factor of a larger dataset J.

3.4.2 Selection of rows from dataset

An interesting problem is how to select from a possibly large dataset J a smaller I such that the factors of I explain well J. This problem was presented in [9], but no solution was provided here.

More precisely, J is the $n \times m$ matrix and $k < n$ the non negative integer smaller than n. We want to choose the $k \times m$ matrix I created by k selected rows from J, such that the factors $\mathcal{F} \subseteq \mathcal{B}(I)$ explain well J.

To solve the above presented problem, we use the essential part of matrices presented in Section 3.3. We benefit from the fact that the essential elements in matrix have some useful properties. One of them is that the essential part of matrix J is a minimal set of entries whose coverage guarantees an exact decomposition of J. Moreover essential part $\mathcal{E}(J)$ can be computed easily.

The procedure is following. For matrix J, we compute $\mathcal{E}(J)$ and choose k rows that contain the most of the essential elements. The idea behind the procedure is quite simple. More covered essential elements leads to a bigger coverage of input data. In Section 6.3.2 we present results of experimentation with this heuristic.

Chapter 4

Previous algorithms

This chapter consists of previous algorithms for decomposition matrices with ordinal attributes. The presented methods will be included in comparison together with new algorithms presented in Chapter 5 in experimental evaluation in the Chapter 6.

4.1 Boolean factorization of ordinally scaled attributes

As was mentioned above, there is great effort devoted to the development of matrix methods for Boolean data, particularly Boolean matrix factorization (BMF), so there exist several algorithms for BMF. Small overview of BMF methods can be found e.g. in [12].

A natural question is if this methods could be used for our purpose, i.e. to perform decomposition of an input matrix I with grades as follows. A positive answer may be given as follows. First, one transforms I by ordinal scaling to a Boolean matrix $I^{\times}$. Second, one performs Boolean factor analysis to $I^{\times}$ and interprets the obtained set $\mathcal{F}^{\times}$ of factors of $I^{\times}$ in an appropriate way, taking the scaling procedure into account.

We presented this approach in paper [7], but experimental evaluation is missing. We compare results obtained by this approach with other algorithms in experimental part of this thesis (Chapter 6).

Given an input matrix $I \in L^{n \times m}$, consider the matrix $I^{\times} \in \{0,1\}^{n \times (m \cdot |L|)}$

defined by

$$I^{\times}_{ij_a} = \begin{cases} 1 & \text{if } a \leq I_{ij}, \\ 0 & \text{otherwise,} \end{cases}$$

for $i = 1, \ldots, n$, $j = 1, \ldots, m$, $a \in L$ (we assume a fixed sorting of the elements in L so that the order of columns in $I^{\times}$ is fixed). That is, $I^{\times}$ is the Boolean matrix resulting from I by simple ordinal scaling. In a sense, each graded attribute j is replaced by a collection of Boolean attributes j_a ($a \in L$); j_a applies to object i if i has j to a degree at least a. The concept lattices and other structures associated to $I^{\times}$ and their relationships to those associated to I are studied in [17, 20] and are utilized in what follows.

Recalling that $\mathrm{rank}_{\mathbf{2}}(I^{\times})$ and $\mathrm{rank}_{\mathbf{L}}(I)$ denote the Boolean rank of $I^{\times}$ and the $\mathbf{L}$-rank of I, respectively, i.e. the smallest numbers of factors using which $I^{\times}$ and I may be explained (factorized), we may formulate the following theorem.

Theorem 3. *For every I, $\mathrm{rank}_{\mathbf{L}}(I) \leq \mathrm{rank}_{\mathbf{2}}(I^{\times})$.*

Proof. Let $A_{\mathcal{F}^{\times}} \circ B_{\mathcal{F}^{\times}} = I^{\times}$ where $|\mathcal{F}^{\times}| = \mathrm{rank}_{\mathbf{2}}(I^{\times})$. Due to [19, Theorem 2], we may safely assume that $\mathcal{F}^{\times} \subseteq \mathcal{B}(I^{\times})$. Due to [17, 20], the ordinary concept lattice $\mathcal{B}(I^{\times})$ is embedded in the fuzzy concept lattice $\mathcal{B}(I)$ via the mapping taking every $\langle C^{\times}, D^{\times} \rangle \in \mathcal{B}(I^{\times})$ to $\langle C, D \rangle \in \mathcal{B}(I)$ with $D(j) = \bigvee_{j_a \in D} a$ and $C = c(C^{\times})^{\uparrow_I \downarrow_I}$ where $c(C^{\times})$ is the characteristic function of $C^{\times}$. Let $\mathcal{F} = \{\langle C, D \rangle \mid \langle C^{\times}, D^{\times} \rangle \in \mathcal{B}(I^{\times})\}$ denote the counterparts of the factors in $\mathcal{F}^{\times}$. To prove the claim, it is sufficient to show that $A_{\mathcal{F}} \circ B_{\mathcal{F}} = I$.

Since for every $\langle C^{\times}, D^{\times} \rangle \in \mathcal{B}(I^{\times})$ we have $c(C^{\times}) \subseteq c(C^{\times})^{\uparrow_I \downarrow_I}$, the set $\mathcal{G} = \{\langle c(C^{\times}), D \rangle \mid \langle C^{\times}, D^{\times} \rangle \in \mathcal{F}^{\times}\}$ satisfies $A_{\mathcal{G}} \circ B_{\mathcal{G}} \leq A_{\mathcal{F}} \circ B_{\mathcal{F}}$. Because $A_{\mathcal{F}} \circ B_{\mathcal{F}} \leq I$ for any $\mathcal{F} \subseteq \mathcal{B}(I)$ [5], it is sufficient to prove $I \leq A_{\mathcal{G}} \circ B_{\mathcal{G}}$. Let thus $I_{ij} = b$ for an arbitrary $\langle i, j \rangle$. Then $I^{\times}_{ij_b} = 1$, hence $A_{\mathcal{F}^{\times}} \circ B_{\mathcal{F}^{\times}} = I^{\times}$ implies that there exists $\langle C^{\times}, D^{\times} \rangle \in \mathcal{B}(I^{\times})$ such that $i \in C^{\times}$ and $j_b \in D^{\times}$, i.e. $(c(C^{\times}))(i) = 1$ and $D(j) \geq b$. It thus follows

$$I_{ij} = b = 1 \otimes b \leq (c(C^{\times}))(i) \otimes D(j) \leq \bigvee_{\langle E,F \rangle \in \mathcal{G}} E(i) \otimes F(j) = (A_{\mathcal{G}} \circ B_{\mathcal{G}})_{ij},$$

proving $I \leq A_{\mathcal{G}} \circ B_{\mathcal{G}}$.

□

Therefore, as far as the number of factors (usually considered as measuring goodness of explanation) is concerned, we are not worse off when directly factorizing I compared to the scale-and-Boolean-factorize method.

Remark 1. *The reason, partly apparent from the proof, is that the spaces of factors to attain optimal factorizations of I and $I^\times$ are the concept lattices $\mathcal{B}(I)$ and $\mathcal{B}(I^\times)$. Now, $\mathcal{B}(I^\times)$ is in general smaller than $\mathcal{B}(I)$, in fact embedded in $\mathcal{B}(I)$ in a natural way [17, 20]. Thus, the space of possible factors of I is richer than that of $I^\times$, leaving Theorem 3 a natural consequence of this fact. In addition to Theorem 3.*

The following example shows that the estimation is not tight and that in fact, $\mathrm{rank}_{\mathbf{L}}(I)$ can be significantly smaller than $\mathrm{rank}_{\mathbf{2}}(I^\times)$.

Example 4. *Let $L = \{0 = a_1, a_2, \ldots, a_n = 1\}$, consider an $n \times 1$ matrix I with $I_{i1} = a_i$. Then $I^\times$ is an $n \times n$ matrix, the square staircase matrix, given by $I^\times_{pq} = 1$ if $p \geq q$ and $= 0$ if $p < q$. It is clear that* $\mathrm{rank}_{\mathbf{L}}(I) = 1$ *and* $\mathrm{rank}_{\mathbf{2}}(I^\times) = n$.

We showed that such approach has severe limitations and later we examine this in experimental evaluation.

4.2 Previous algorithms for ordinal data

Pointing on limitations of Boolean factorization of ordinally scaled attributes, we claim that factor analyzing I directly using the methods suited for ordinal data has a significant advantage.

This section will be devoted to the brief introduction of known algorithms that can be more or less used for doing so. The first two presented algorithms are based on algorithms presented in [19] developed for Boolean data. Another algorithm is the algorithm widely used in statistics—perhaps the best known method designed for real-valued matrices—the non-negative matrix factorization (NMF) [38].

4.2.1 GreCon$_L$

As was mentioned above, this algorithm was presented in [19]. Moreover in [19] was proved that the optimal factors are obtained from the space of factors computed via FCA. The first algorithm (called Algorithm 1, later called GreCon[1]) is based on an algorithm for set covering problem. The algorithm can be simply used for our fuzzy setting. A disadvantage of this

[1] Greedy Concepts

approach is that such algorithm requires us to compute first the set $\mathcal{B}(I)$ of all formal fuzzy concepts and then select candidates for factor from $\mathcal{B}(I)$. Because $\mathcal{B}(I)$ can be exponentially large, this approach is time-consuming.

4.2.2 GreConD$_L$

The second algorithm for BMF presented in [19], (called Algorithm 2, later GreConD[2]) was modified for decomposition ordinal data in [18]. This algorithm is designed to avoid computing the set $\mathcal{B}(I)$ of all formal concepts. Instead, it computes concepts on demand.

This algorithm generates factors by looking for "promising columns". It works due to fact that each formal concept $\langle C, D\rangle$, each intent D is an union of intents $\{^{D(j)}/_j\}^{\uparrow\downarrow}$. As a consequence, we may construct any formal concept by adding sequentially $\{^a/_j\}^{\uparrow\downarrow}$ to the empty set of attributes. This algorithm follows a greedy approach that selects $j \in Y$ and degree $a \in L$ which maximize the size of

$$D \oplus_a j = \{\langle k, l\rangle \in U | \{D \cup \{^a/_j\}\}^{\downarrow}(k) \otimes \{D \cup \{^a/_j\}\}^{\downarrow\uparrow}(l) \geq I_{k,l}\}.$$

The symbol U there denotes the set of $\langle i, j\rangle$ of I for which the corresponding entry I_{ij} is not covered yet. A pseudocode of this algorithm is depicted in Algorithm 1.

4.2.3 Statistical methods

Statistical methods are widely used in many fields such as bioinformatics, medicine, chemistry and lot more. Representatives of these methods are for example principal component analysis (PCA) [53], independent component analysis (ICA) [23], network component analysis (NCA) [40], non-negative matrix factorisation (NMF) [38] or singular value decomposition (SVD) [56]. These methods usually decompose $n \times m$ input matrix I into a product of two (in case of PCA, ICA, NCA, NMF) or three matrices (in SVD). The constraints of each method are different. For example for PCA, we want one of the resulted matrices to be orthogonal, in ICA we require all components of resulted matrix independent and in NMF both matrices need to be non-negative. Due to this fact, non-negative matrix factorisation is method the most relevant to purpose of this thesis, so we omit the rest ones.

[2]Greedy Concepts on Demand

Algorithm 1: GreConD$_L$

Input: matrix I with entries from scale L
Output: set $\mathcal{F} \subseteq \mathcal{B}(I)$ of factor concepts

$\mathcal{F} \leftarrow \emptyset$
$U \leftarrow \{\langle i, j\rangle | I_{ij} > 0\}$
while U *is non-empty* **do**
  $D \leftarrow \emptyset$; $V \leftarrow 0$
  select $\langle j, a\rangle$ that maximizes $|D \oplus_a j|$
  while $|D \oplus_a j| > V$ **do**
    $V \leftarrow |D \oplus_a j|$
    $D \leftarrow \{D \cup \{^a/_j\}\}^{\downarrow\uparrow}$
    select $\langle j, a\rangle$ that maximizes $|D \oplus_a j|$
  end
  $C \leftarrow D^{\downarrow}$
  $\mathcal{F} \leftarrow \mathcal{F} \cup \{\langle C, D\rangle\}$
  remove entries $\langle i, j\rangle$ covered by $C^{\uparrow\downarrow} \otimes D^{\downarrow\uparrow}$ in I from U
end
return $\mathcal{F}$

Non-negative matrix factorization

Interest in this methods started with the paper [38]. There exists hundreds of papers about NMF, and most of them cite [38] although this method was developed by Pentti Paatero [52] five years earlier.

NMF can be stated as follows: Given a non-negative matrix $A \in \mathbb{R}^{m\times n}$ and a positive integer $k < min(\{m, n\})$, find non-negative matrices $W \in \mathbb{R}^{m\times k}$ and $H \in \mathbb{R}^{k\times n}$ to minimize the function

$$f(W, H) = \frac{1}{2}||A - WH||_F^2.$$

The product WH is called *non-negative factorisation of* A. However, A is not usually equal to the product WH, i.e. WH is an approximate factorisation of rank at most k.

Various alternative minimization strategies have been proposed. In the standard NMF algorithm W and H are initialized with random non-negative values and than iteratively computes better approximation. Algorithms for NMF can be divided into three general classes:

- Multiplicative update algorithms,

- Gradient descent algorithms,
- Alternating least squares algorithms.

Multiplicative update algorithms The algorithm presented here (Algorithm 2) is prototypical algorithm originated in [39].

Algorithm 2: Multiplicative update algorithm for NMF

Input: matrix $A \in \mathbb{R}^{m \times n}$, positive integer k
Output: matrices $W \in \mathbb{R}^{m \times k}$ and $H \in \mathbb{R}^{k \times n}$

1 $W \leftarrow rand(m, k)$
2 $H \leftarrow rand(k, n)$
3 **for** $i \in \{1, \ldots, maxiter\}$ **do**
4 $\quad H \leftarrow H \cdot (W^T A)/(W^T W H + 10^{-9})$
5 $\quad W \leftarrow W \cdot (AH^T)/(WHH^T + 10^{-9})$
6 **end**
7 **return** H, W

The value 10^{-9} in each update (lines 4 and 5) is added to avoid division by 0. Constant *maxiter* indicates the number of iterations. The assumption that this algorithm converges to a local minimum was later shown to be incorrect (see [22]), but this was the first well-known algorithm and newer algorithms are usually compared with it.

Gradient descent algorithms The second class of algorithms is based on the gradient descent method. Algorithms from this class repeatedly apply update rules (lines 4 and 5) see pseudocode depicted in Algorithm 3

ε_H and ε_W are step size parameters and they depend on concrete algorithm.

Alternating least squares algorithms All algorithms here are composed by least squares steps in an alternating fashion. They exploit the fact that while the optimization problem of non-negative matrix factorisation is not convex in both W and H, is convex in either W or H. Thus from given one matrix, the second one is computed with a simple leasts squares. An elementary algorithm from this group follows (see Algorithm 4).

Algorithm 3: Basic gradient descent algorithm for NMF

Input: matrix $A \in \mathbb{R}^{m \times n}$, positive integer k
Output: matrices $W \in \mathbb{R}^{m \times k}$ and $H \in R^{k \times n}$

$W \leftarrow rand(m, k)$
$H \leftarrow rand(k, n)$
for $i \in \{1, \ldots, maxiter\}$ **do**
    $H \leftarrow H - \varepsilon_H \frac{\partial f}{\partial H}$
    $W \leftarrow W - \varepsilon_W \frac{\partial f}{\partial W}$
end
return H, W

Algorithm 4: Basic alternating least squares algorithm for NMF

Input: matrix $A \in \mathbb{R}^{m \times n}$, positive integer k
Output: matrices $W \in \mathbb{R}^{m \times k}$ and $H \in R^{k \times n}$

$W \leftarrow rand(m, k)$
for $i \in \{1, \ldots maxiter\}$ **do**
    solve for H equation $W^T W H = W^T A$
    set all negative elements in H to 0
    solve for W equation $H H^T W^T = H A^T$
    set all negative elements in W to 0
end
return H, W

Chapter 5

New algorithms

In this chapter we present three algorithms for decomposition of matrices over scales. The first two algorithms are inspired by GreEss [12] and Asso [46], currently perhaps the best algorithms for the AFP and DBP, respectively. The third one is slightly modified GreConD$_L$, which is inspired by a modified binary version of GreConD called GreConD+ presented in [13]. The last mentioned algorithm—GreConD+—is based on algorithm which constructs a factorization from formal concepts, but computes a general factorization, since formal concepts never allow overcover error.

5.1 GreEss$_L$

In [12] a new algorithm based on the properties of essential parts $\mathcal{E}(I)$ of Boolean matrices I was presented. The algorithm uses the fact that $\mathcal{E}(I)$ represents the entries whose cover by arbitrary factors guarantees an exact decomposition of I. Another useful property is that the number of 1s in $\mathcal{E}(I)$ tends to be significantly smaller than the number of 1s in I. $\mathcal{E}(I)$ may be simpler to cover than I, also note that we can compute $\mathcal{E}(I)$ efficiently. These features hold also in fuzzy setting. An algorithm based on this idea appeared in [8, 15].

5.1.1 Essential parts of matrices over scales

This section refers to Section 2.3 and Section 3.3, where are described first observations on the role of entries in input matrix I. We defined so-called

essential part of I, dentoted $\mathcal{E}(I)$, whose cover by formal concepts of I guarantees the cover of all entries in I. We defined intervals $\mathcal{I}_{C,D}$ that play crucial role in this consideration.

The following lemma shows that all the rectangles corresponding to the formal concepts in $\mathcal{I}_{C,D}$ cover the rectangle $C^{\mathrm{T}} \circ D$.

Lemma 2. *If $\langle E, F\rangle \in \mathcal{I}_{C,D}$ then $C^{\mathrm{T}} \circ D \leq E^{\mathrm{T}} \circ F$.*

Proof. Since $\langle E, F\rangle \in \mathcal{I}_{C,D}$, we have $C^{\uparrow\downarrow} \leq E$ and $D^{\downarrow\uparrow} \leq F$. As $\langle E, F\rangle$ is a formal concept of I, we have $E = E^{\uparrow\downarrow}$ and $F = F^{\downarrow\uparrow}$. Since $^{\uparrow\downarrow}$ and $^{\downarrow\uparrow}$ are closure operators, we obtain $C \leq C^{\uparrow\downarrow} \leq C^{\uparrow\downarrow\uparrow\downarrow} = E^{\uparrow\downarrow} \leq E$ and similarly $D \leq F$. The claim now easily follows. □

In particular, consider $C = \{{}^{a}/_{i}\}$ by which we denote the "singleton" vector with zero components except $C_i = a$, and $D = \{{}^{b}/_{j}\}$ with analogous meaning. Then every concept $\langle E, F\rangle$ in $\mathcal{I}_{C,D} = \mathcal{I}_{\{{}^{a}/_{i}\},\{{}^{b}/_{j}\}}$ covers the entry $\langle i, j\rangle$ in $C^{\mathrm{T}} \circ D$. This means that if $a \otimes b = I_{ij}$, then every concept in $\mathcal{I}_{\{{}^{a}/_{i}\},\{{}^{b}/_{j}\}}$ covers the entry $\langle i, j\rangle$ in I. However, the entry $\langle i, j\rangle$ in I is covered also by other concepts than those in $\mathcal{I}_{\{{}^{a}/_{i}\},\{{}^{b}/_{j}\}}$. The following lemma is crucial in understanding this issue.

Lemma 3. *Let $\langle E, F\rangle \in \mathcal{B}(X, Y, I)$ and $a, b \in L$. Then $a \otimes b \leq E_i \otimes F_j$ if and only if for some c, d with $a \otimes b \leq c \otimes d$ we have $\langle E, F\rangle \in \mathcal{I}_{\{{}^{c}/_{i}\},\{{}^{d}/_{j}\}}$.*

Proof. If $a \otimes b \leq E_i \otimes F_j$, one may put $c = E_i$ and $d = F_j$. Namely, we then have to check $\langle E, F\rangle \in \mathcal{I}_{\{{}^{E_i}/_{i}\},\{{}^{F_j}/_{j}\}}$ which is equivalent to $\gamma(\{{}^{E_i}/_{i}\}) \leq E$ and $\mu(\{{}^{F_j}/_{j}\}) \leq F$. The first inequality is equivalent to $\{{}^{E_i}/_{i}\}^{\uparrow\downarrow} \leq E$ which is true. Namely, from the obvious fact $\{{}^{E_i}/_{i}\} \leq E$ we obtain by isotony and idempotency of $^{\uparrow\downarrow}$ that $\{{}^{E_i}/_{i}\}^{\uparrow\downarrow} \leq E^{\uparrow\downarrow} = E$. The second inequality is obtained symmetrically.

Conversely, assume that for some c, d with $a \otimes b \leq c \otimes d$ we have $\langle E, F\rangle \in \mathcal{I}_{\{{}^{c}/_{i}\},\{{}^{d}/_{j}\}}$. Lemma 2 then implies $\{{}^{c}/_{i}\}^{\mathrm{T}} \circ \{{}^{d}/_{j}\} \leq E^{\mathrm{T}} \circ F$, which entails $c \otimes d \leq E_i \otimes F_j$. Since $a \otimes b \leq c \otimes d$, the proof is finished. □

For a given matrix $I \in L^{n\times m}$ let

$$\mathbf{I}_{ij} = \{\mathcal{I}_{\{{}^{a}/_{i}\},\{{}^{b}/_{j}\}} \mid a, b \in L, a \otimes b = I_{ij}\}$$

and put

$$\mathcal{I}_{ij} = \bigcup \mathbf{I}_{ij}.$$

Next, we can show that $\mathcal{I}_{ij}$ is just the set of all formal concepts of I that cover $\langle i, j\rangle$.

Theorem 4. *The rectangle corresponding to* $\langle E, F\rangle \in \mathcal{B}(X, Y, I)$ *covers* $\langle i, j\rangle$ *in* I *iff* $\langle E, F\rangle \in \mathcal{I}_{ij}$.

Proof. If $E^{\mathrm{T}} \circ F$ covers $\langle i, j\rangle$, i.e. $I_{ij} = E_i \otimes F_j$, then since $I_{ij} = I_{ij} \otimes 1$, we obtain $\langle E, F\rangle \in \mathcal{I}_{ij}$ by Lemma 3.

Conversely, let $\langle E, F\rangle \in \mathcal{I}_{ij}$, i.e. $\langle E, F\rangle \in \mathcal{I}_{\{a/i\},\{b/j\}}$ for some a, b with $a \otimes b = I_{ij}$. Lemma 3 then implies $I_{ij} = a \otimes b \leq E_i \otimes F_j$. Since the definition of a formal concept of I along with adjointness yield that we always have $E_i \otimes F_j \leq I_{ij}$, we readily obtain $E_i \otimes F_j = I_{ij}$, finishing the proof. □

Denote now by $\mathcal{E}(I) \in L^{n\times m}$ the matrix over L defined by

$$(\mathcal{E}(I))_{ij} = \begin{cases} I_{ij} & \text{if } \mathcal{I}_{ij} \text{ is } \neq \emptyset \text{ and minimal w.r.t. } \subseteq, \\ 0 & \text{otherwise.} \end{cases}$$

The following two theorems provide the main result in this section and are utilized in the new algorithm.

Theorem 5. $\mathcal{E}(I)$ *is an essential part of* I.

Proof. First, $\mathcal{E}(I) \leq I$ follows from the definition of $\mathcal{E}(I)$. Second, consider any $\mathcal{F} \subseteq \mathcal{B}(I)$ for which $\mathcal{E}(I) \leq A_{\mathcal{F}} \circ B_{\mathcal{F}}$. We need to show $I = A_{\mathcal{F}} \circ B_{\mathcal{F}}$.

On one hand, $I \geq A_{\mathcal{F}} \circ B_{\mathcal{F}}$ is a consequence of the fact that every $\langle C, D\rangle \in \mathcal{F}$ is a formal concept of I. Namely, adjointness easily yields $C_i \otimes D_j \leq I_{ij}$ from which the required inequality directly follows.

It remains to prove $I \leq A_{\mathcal{F}} \circ B_{\mathcal{F}}$. Consider any $\langle i, j\rangle$ and the corresponding set $\mathcal{I}_{ij}$. Take any $\mathcal{I}_{i'j'} \subseteq \mathcal{I}_{ij}$ that is non-empty and minimal w.r.t $\subseteq$. The definition of $\mathcal{E}(I)$ implies $\mathcal{E}(I)_{i'j'} = I_{i'j'}$. Since $\mathcal{E}(I) \leq A_{\mathcal{F}} \circ B_{\mathcal{F}}$, the definition of $\circ$ and of $A_{\mathcal{F}}$ and $B_{\mathcal{F}}$ imply the existence of $\langle C, D\rangle \in \mathcal{F}$ for which $\mathcal{E}(I)_{i'j'} \leq C_{i'} \otimes D_{j'}$. Since $\langle C, D\rangle$ is a formal concept of I, we also have $C_{i'} \otimes D_{j'} \leq I_{i'j'} \leq \mathcal{E}(I)_{i'j'}$, hence the rectangle corresponding to $\langle C, D\rangle$ covers $\langle i', j'\rangle$. Thanks to Theorem 4 we get $\langle C, D\rangle \in \mathcal{I}_{i'j'}$ and since $\mathcal{I}_{i'j'} \subseteq \mathcal{I}_{ij}$, also $\langle C, D\rangle \in \mathcal{I}_{ij}$. Applying Theorem 4 again now yields that the rectangle corresponding to $\langle C, D\rangle$ covers $\langle i, j\rangle$, i.e. $I_{ij} = C_i \otimes D_j \leq (A_{\mathcal{F}} \circ B_{\mathcal{F}})_{ij}$ and since we always have $(A_{\mathcal{F}} \circ B_{\mathcal{F}})_{ij} \leq I_{ij}$, we obtain $(A_{\mathcal{F}} \circ B_{\mathcal{F}})_{ij} = I_{ij}$. Since $\langle i, j\rangle$ is arbitrary, $I = A_{\mathcal{F}} \circ B_{\mathcal{F}}$ follows. □

The next theorem shows how a factorization of $\mathcal{E}(I)$ may be used to obtain a factorization of I.

Theorem 6. *Let $\mathcal{G} \subseteq \mathcal{B}(\mathcal{E}(I))$ be a set of factor concepts of $\mathcal{E}(I)$, i.e. $\mathcal{E}(I) = A_\mathcal{G} \circ B_\mathcal{G}$. Then every set $\mathcal{F} \subseteq \mathcal{B}(I)$ containing for each $\langle C, D\rangle \in \mathcal{G}$ at least one concept from $\mathcal{I}_{C,D}$ is a set of factor concepts of I, i.e. $I = A_\mathcal{F} \circ B_\mathcal{F}$.*

Proof. Let for $\langle C, D\rangle \in \mathcal{G}$ denote by $\langle E, F\rangle_{\langle C,D\rangle}$ a concept in $\mathcal{F} \cap \mathcal{I}_{C,D}$ (it exists by assumption). Due to Lemma 2, $\langle E, F\rangle_{\langle C,D\rangle}$ covers the rectangle corresponding to $\langle C, D\rangle$. Since this this is true for every $\langle C, D\rangle \in \mathcal{G}$, it is easy to see that $A_\mathcal{G} \circ B_\mathcal{G} \leq A_\mathcal{F} \circ B_\mathcal{F}$. The assumption $\mathcal{E}(I) = A_\mathcal{G} \circ B_\mathcal{G}$ now yields $\mathcal{E}(I) \leq A_\mathcal{F} \circ B_\mathcal{F}$. As $\mathcal{E}(I)$ is an essential part of I, we get $I = A_\mathcal{F} \circ B_\mathcal{F}$, finishing the proof. □

Example 5. *Let us continue with inputs from Example 2. For matrix I, we obtain essential part*

$$\mathcal{E}(I) = \begin{pmatrix} 0.5 & 1.0 & 0.0 \\ 0.0 & 0.0 & 0.5 \\ 0.0 & 0.5 & 0.0 \end{pmatrix}.$$

5.1.2 GreEss$_L$ algorithm

The GreEss$_L$ algorithm, which we now present, is inspired by [12]. GreEss$_L$ is based on the results from Section 5.1.1 and some other facts mentioned below. It is primarily designed for AFP(L), but can also be used for DBP(L). The pseudocode depicted in Algorithm 5 describes computation of an exact decomposition of I but an easy modification makes it an algorithm for computation of ε-approximate decompositions (in line 3, stop when precision ε is reached).

In Algorithm 5 and Algorithm 6 the symbol $\emptyset$ denotes the empty set or the vector full of zeroes, depending on the context, $F \vee \{{}^a/_j\}$ denotes F with the component F_j updated to $F_j \vee a$, and $C \otimes D$ denotes the crossproduct of C and D, i.e. the rectangle for which $(C \otimes D)_{ij} = C_i \otimes D_j$. Moreover, U denotes the set of entries $\langle i, j\rangle$ not yet covered by the factors computed so far, and $cov(U, F, J)$ and $cov_I(U, D, \mathcal{E})$ denote the number of $\langle i, j\rangle \in U$ covered in I by the rectangle $F^{\downarrow_J} \otimes F^{\downarrow_J\uparrow_J}$ and $(D^{\downarrow_\mathcal{E}})^{\uparrow_I\downarrow_I} \otimes (D^{\downarrow_\mathcal{E}\uparrow_\mathcal{E}})^{\downarrow_I\uparrow_I}$, respectively. The fuzzy-set-like notation $\{{}^a/_j\} \in C^{\uparrow_I} \setminus F$ means $F_j < a \leq C_j^{\uparrow_I}$.

ComputeIntervals first computes $\mathcal{E}(I)$ (easy by definition) and then computes a set $\mathcal{G}$ of factors of $\mathcal{E}(I)$, each $\langle C, D\rangle \in \mathcal{G}$ representing the interval

$\mathcal{I}_{C,D}$ in $\mathcal{B}(I)$ from which it is possible to obtain a decomposition of I according to Theorem 6. In fact we use the following improvement of Theorem 6 whose proof is easy and thus omitted: for $\mathcal{G}$ it suffices (rather than being a set of factor concepts of $\mathcal{E}(I)$) that the crossproducts $C^{\uparrow_I\downarrow_I} \otimes D^{\downarrow_I\uparrow_I}$ corresponding to $\langle C, D\rangle \in \mathcal{G}$ cover all entries in I (line 11). The formal concepts in $\mathcal{G}$ are computed in a greedy manner from $\mathcal{E}(I)$ by sequentially increasing in D (initially set to $\emptyset$) the most promising value a of the most promising component j (line 5–9), until such increase is impossible. The formal concept $\langle C, D\rangle$, obtained by taking closures w.r.t. $\mathcal{E}$ in line 7, is added to $\mathcal{G}$ (line 10). The entries covered by $C^{\uparrow_I\downarrow_I} \otimes D^{\downarrow_I\uparrow_I}$ are removed from U. The selection is repeated until U is empty.

With $\mathcal{G}$ obtained this way, GreEss$_L$ performs a greedy search for factors, i.e. formal concepts, in the intervals $\mathcal{I}_{C,D}$, $\langle C, D\rangle \in \mathcal{G}$, in line 3–21. For every $\mathcal{I}_{C,D}$ we select the formal concept in $\mathcal{I}_{C,D}$ with best coverage in line 6–11 in a manner similar to the one used in ComputeIntervals, i.e. extending the initially empty F by most promising attributes j and degrees a. The condition $J \leftarrow D^{\downarrow_I} \otimes C^{\uparrow_I}$ which functions as a restriction speeding up the computation, guarantees that we do not leave $\mathcal{I}_{C,D}$ in this search. The best found concept $\langle E', F'\rangle$ over all the intervals is then added to $\mathcal{F}$ in line 18. The interval $\mathcal{I}_{C',D'}$ in which $\langle E', F'\rangle$ was found is removed from $\mathcal{G}$ in line 19 (hence is not searched in the remaining iterations) and U is updated accordingly.

To proof correctness of this algorithm we provide its detailed description. GreEss$_L$ uses function ComputeIntervalswhich computes a set of concepts $\mathcal{G}$ by first computing the matrix $\mathcal{E}(I)$ and then computing the concepts of $\mathcal{B}(X, Y, \mathcal{E}(I))$ in a greedy manner inspired by [18] and adding them to $\mathcal{G}$. The difference between original algorithm from [18] is that we maximize size of $C^{\uparrow_I\downarrow_I} \otimes D^{\downarrow_I\uparrow_I}$ for concept $\langle C, D\rangle \in \mathcal{B}(X, Y, \mathcal{B}(I))$. This is possible since the factors for I are selected form the interval $\mathcal{I}_{C,D}$ (due to Lemma 2). GreEss$_L$ than picks at most one concept from every interval $\mathcal{I}_{C,D}$ for $\langle C, D\rangle \in \mathcal{G}$ until U is covered. It selects the intervals in a greedy manner similar to the one we described above.

Example 6. *For Example 2* ComputeIntervals *computes essential part of input matrix (see Example 5) and returns a set*

$$\mathcal{G} = \{\langle\{{}^{1}/a, {}^{0.5}/b, {}^{0.5}/c\}, \{{}^{0.5}/1, {}^{0.5}/2\}\rangle, \langle\{{}^{0.5}/a, {}^{1}/b, {}^{0.5}/c\}, \{{}^{0.5}/3\}\rangle\}.$$

From this set GreEss$_L$ *computes concepts* C_2 *and* C_1 *as factors.*

Algorithm 5: GreEss$_L$

Input: matrix I with entries in scale L
Output: set $\mathcal{F}$ of factors for which $I = A_{\mathcal{F}} \circ B_{\mathcal{F}}$

$\mathcal{G} \leftarrow$ ComputeIntervals(I)
$U \leftarrow \{\langle i,j\rangle | I_{ij} > 0\}$; $\mathcal{F} \leftarrow \emptyset$
while U *is non-empty* **do**
  $s \leftarrow 0$
  foreach $\langle C, D\rangle \in \mathcal{G}$ **do**
    $J \leftarrow D^{\downarrow_I} \otimes C^{\uparrow_I}$; $F \leftarrow \emptyset$; $s_{\langle C,D\rangle} \leftarrow 0$
    while *exists* $\{^a/_j\} \in C^{\uparrow_I} \setminus F$ *s.t.* $cov(U, F \vee \{^a/_j\}, J) > s_{\langle C,D\rangle}$ **do**
      select $\{^a/_j\}$ maximizing $cov(U, F \vee \{^a/_j\}, J)$
      $F \leftarrow (F \vee \{^a/_j\})^{\downarrow_J\uparrow_J}$
      $E \leftarrow (F \vee \{^a/_j\})^{\downarrow_J}$
      $s_{\langle C,D\rangle} \leftarrow cov(U, F, J)$
    end
    if $s_{\langle C,D\rangle} > s$ **then**
      $\langle E', F'\rangle \leftarrow \langle E, F\rangle$
      $\langle C', D'\rangle \leftarrow \langle C, D\rangle$
      $s \leftarrow s_{\langle C,D\rangle}$
    end
  end
  add $\langle E', F'\rangle$ to $\mathcal{F}$
  remove $\langle C', D'\rangle$ from $\mathcal{G}$
  remove from U all $\langle i,j\rangle$ covered by $E' \otimes F'$ in I
end
return $\mathcal{F}$

Algorithm 6: COMPUTEINTERVALS

Input: matrix I with entries in scale L
Output: set $\mathcal{G} \subseteq \mathcal{B}(\mathcal{E}(I))$

$\mathcal{E} \leftarrow \mathcal{E}(I)$
$U \leftarrow \{\langle i, j\rangle | \mathcal{E}_{ij} > 0\}$
while U *is non-empty* **do**
  $D \leftarrow \emptyset$; $s \leftarrow 0$
  while *exists* $\{^a/_j\} \in D$ *s.t.* $cov_I(U, D \vee \{^a/_j\}, \mathcal{E}) > s$ **do**
    select $\{^a/_j\}$ maximizing $cov_I(U, D \vee \{^a/_j\}, \mathcal{E})$
    $D \leftarrow (D \vee \{^a/_j\})^{\downarrow\varepsilon\uparrow\varepsilon}$; $C \leftarrow (D \vee \{^a/_j\})^{\downarrow\varepsilon}$
    $s \leftarrow cov_I(U, D, \mathcal{E})$
  end
  add $\langle C, D\rangle$ to $\mathcal{G}$
  remove from U entries $\langle i, j\rangle$ covered by $C^{\uparrow_I\downarrow_I} \otimes D^{\downarrow_I\uparrow_I}$ in I
end
return $\mathcal{G}$

5.2 Asso$_L$

ASSO$_L$ is inspired by ASSO algorithm [46], currently the best known algorithm for DBP. A preliminary version of the algorithm was presented in [8]. The final version of the algorithm will be presented in detail in an extended version of the paper [8] in [14].

5.2.1 Association matrix

Recall that the ordinary ASSO is based on the idea of using the rows of the association matrix $\mathcal{A}$ of I as candidate basis vectors, i.e. rows of the $k \times m$ factor-attribute matrix B. $\mathcal{A}$ is an $m \times m$ Boolean matrix such that $\mathcal{A}_{pq} = 1$ if the confidence $c(p, q)$ of the association rule $\{p\} \Rightarrow \{q\}$ given by I exceeds a given threshold τ.

The confidence $c(p, q)$ may be understood as a conditional probability, namely that "an object has attribute q provided it has attribute p", given that objects as elementary events are equally probable. In presence of grades, we consider conditional probabilities $c_a(p, q)$ that "an object has attribute q provided it has attribute p to degree at least a". Loosely speaking, $c_a(p, q)$ is the confidence that the presence of p to degree at least a implies the presence

of q. Unlike in the Boolean case, the collections of objects sharing some attributes to prescribed degrees are naturally conceived as fuzzy sets rather than ordinary sets. Thus, the collection $\{^a/p\}^\downarrow$ of all objects having attribute p at least to degree a is a fuzzy set of objects to which object $i = 1, \ldots, n$ belongs to degree

$$\{^a/p\}^\downarrow(i) = a \to I(i,p),$$

see [3]. Likewise, the collection of objects having p to degree at least a and having q is defined by

$$\{^a/p, {}^1/q\}^\downarrow(i) = (a \to I(i,p)) \wedge I(i,q).$$

These formulas may be obtained from considerations on Galois connections induced by graded relations [3] (as these are the mathematical counterparts of assignments of objects sharing a given collection of attributes) but may also be obtained on intuitive grounds.

In evaluating conditional probability that defines $c_a(p,q)$ we deal with fuzzy events (many-valued events) and probabilities of fuzzy events in the sense of Zadeh [63]. That is, the probability measure of fuzzy events involved in our situation is a function P assigning to every fuzzy set A of objects a number $P(A) \in [0,1]$—the probability of the fuzzy event A. Assuming as in the classical case that the objects as elementary events are equally probable, Zadeh's formulas for conditional probabilities $P(\cdot|\cdot)$ of fuzzy events yield that the confidence in question is defined by

$$\begin{aligned} c_a(p,q) &= P(\{^1/q\}^\downarrow | \{^a/p\}^\downarrow) = \frac{P(\{^a/p\}^\downarrow \cap \{^1/q\}^\downarrow)}{P(\{^a/p\}^\downarrow)} = \frac{P(\{^a/p, {}^1/q\}^\downarrow)}{P(\{^a/p\}^\downarrow)} \\ &= \frac{|\{^a/p, {}^1/q\}^\downarrow|}{|\{^a/p\}^\downarrow|}, \end{aligned}$$

where $|A|$ denotes the cardinality of a fuzzy set A. With $|A| = \sum_{i=1}^n A(i)$ we thus obtain

$$|\{^a/p, {}^1/q\}^\downarrow| = \sum_{i=1}^n \{^a/p, {}^1/q\}^\downarrow(i), \quad \text{and } |\{^a/p\}^\downarrow| = \sum_{i=1}^n \{^a/p\}^\downarrow(i).$$

Note also that in deriving the formula for $c_a(p,q)$ we used $\{^a/p\}^\downarrow \cap \{^1/q\}^\downarrow = (\{^a/p\} \cup \{^1/q\})^\downarrow = \{^a/p, {}^1/q\}^\downarrow$ which is a basic property of Galois connections [3]. The confidence is a number in $[0,1]$ which may be transformed to

a truth value in L using a user-defined threshold $\tau \in [0,1]$. The reason is in principle the same as in the Boolean case, namely to obtain from the vectors of confidence values, $\langle \ldots, c_a(p,q), \ldots \rangle$, appropriate vectors of grades in L, i.e. the candidate basis vectors. However, the thresholding process is more involved compared to the Boolean case, and we propose to accomplish it by the rounding function round_τ defined for $r \in [0,1]$ by

$$\text{round}_\tau(r) = \begin{cases} r_+ = \min\{a \in L \mid a \geq r\} \text{ if } r_+ \leftrightarrow r \geq \tau, \\ r_- = \max\{a \in L \mid a < r\} \text{ otherwise.} \end{cases}$$

Here, $r_+ \leftrightarrow r = \min(r_+ \to r, r \to r_+)$ is the many-valued logical equivalence mentioned above. One may observe that if $L = \{0,1\}$ we obtain the thresholding involved in the ordinary Asso.

This way, we may define for every attribute p and every suitable grade $a \in L - \{0\}$ a candidate basis vector, i.e. a row $\mathcal{A}_{(p,a),\text{-}}$ of a prospective association matrix $\mathcal{A}$, by

$$\mathcal{A}_{(i,a),j} = \text{round}_\tau(c_a(p,q)).$$

Picking now a set $K \subseteq L - \{0\}$ of suitable grades, we obtain an association matrix $\mathcal{A} \in L^{(m \cdot |K|) \times m}$. One may verify that if $L = \{0,1\}$ and $K = \{1\}$ then $\mathcal{A}$ is just the ordinary $m \times m$ association matrix defined in [46]. The presence of intermediate grades allows us to broaden the set of candidate basis vectors. Namely, in addition to the possible choice $K = \{1\}$, we may pick K containing more grades, e.g. $K = L - \{0\}$, and thus enlarge the search space for factorization.

5.2.2 Procedures Cover and Asso$_L$

The basic idea of the Asso algorithm may be described as follows. The algorithm iteratively computes k factors one by one, with the provision that it stops with less than k factors if the addition of any new factor would only worsen the error function, i.e. would decrease the value of s in our case. Let A and B denote the object-factor and factor-attribute matrices computed so far. The next factor, which is described by a new column and a new row to be added to A and B, is computed as follows. For every candidate row of B, i.e. the row of the association matrix $\mathcal{A}$, one determines the best corresponding candidate column of A. "Best" means such that the value of a function Cover (see Equation 5.1) is maximized. The candidate row of B

and column of A with the highest value of COVER are then added as a new factor to B and A.

The purpose of the function COVER is to yield a high value for factors whose addition is likely to lead to a good resulting matrices A and B, i.e. with high value of s. In the Boolean case, this means that we want a high number C of entries $\langle i, j\rangle$ for which $I_{ij} = 1$ and $(A \circ B)_{ij} = 1$, i.e. 1s in I that are "covered" by the factors, and a small number O of entries for which $I_{ij} = 0$ and $(A \circ B)_{ij} = 1$, i.e. are "overcovered" by the factors. This reasoning leads to the formula

$$w^+ \cdot C - w^- \cdot O$$

as the definition of COVER in the Boolean case. The weights reflect relative importance of C and O. In practice, one works with w^- larger than w^+ because "overcovering" cannot be undone by adding further factors. Hence, the presence of a single $\langle i, j\rangle$ with $(A \circ B)_{ij} = 1$ and $I_{ij} = 0$ represents a more serious harm than a presence of a single $\langle i, j\rangle$ with $(A \circ B)_{ij} = 0$ and $I_{ij} = 1$, because the latter discrepancy may be corrected by adding an appropriate factor in the next steps of the algorithm.

An appropriate form of the COVER function in the setting with general scales is more delicate. One reason is that the coverage of entry $\langle i, j\rangle$ of I is a matter of degree. We therefore need to account for a partial coverage and a partial overcoverage. For instance, if $I_{ij} = 0.5$ and $(A \circ B)_{ij} = 0.4$, then one may consider $\langle i, j\rangle$ almost covered and thus consider $I_{ij} \leftrightarrow (A \circ B)_{ij} = 0.5 \leftrightarrow 0.4 = 0.9$ as the degree to which $\langle i, j\rangle$ is covered. Likewise, if $I_{ij} = 0.5$ and $(A \circ B)_{ij} = 0.6$, then $\langle i, j\rangle$ is slightly overcovered and $\neg(I_{ij} \leftrightarrow (A \circ B)_{ij}) = \neg(0.5 \leftrightarrow 0.6) = 0.1$ may be thought of as a degree to which $\langle i, j\rangle$ is overcovered. Using a similar reasoning as in the Boolean case, one could obtain the value of COVER by adding the degrees corresponding to the first type of entries, multiply them with w^+ and subtract from this number the w^--multiple of the sum of the degrees corresponding to the second type of entries. This, however, would not yet be an appropriate approach. For consider a situation in which $I_{ij} = 0.5$, w^- is even five times larger than w^+, and the so far computed matrices A and B yield $(A \circ B)_{ij} = 0.3$. Suppose we now have two options. First, adding a factor resulting in A_1 and B_1 with $(A_1 \circ B_1)_{ij} = 0.4$; second, adding a factor resulting in A_2 and B_2 with $(A_2 \circ B_2)_{ij} = 0.52$. Intuitively, the second choice is preferable because the factor commits only a slight overcovering of $I_{ij} = 0.5$. However, the function COVER

described above would lead to the selection of the first factor. Namely (for simplicity, we disregard entries other than $\langle i, j\rangle$), the first factor contributes by $w^+ \cdot (I_{ij} \leftrightarrow (A_1 \circ B_1)_{ij}) = w^+ \cdot 0.9$, while the second one contributes by $-w^- \cdot \neg(I_{ij} \leftrightarrow (A_2 \circ B_2)_{ij}) = -w^- \cdot 0.02$, i.e. even represents a decrease in value of Cover. The point is that the entries which are overcovered, i.e. $I_{ij} < (A \circ B)_{ij}$, need to be looked at as follows: They need to be penalized for overcovering by $w^- \cdot \neg(I_{ij} \leftrightarrow (A \circ B)_{ij})$ but at the same time rewarded for full covering by $w^+ \cdot 1$. This type of problem is degenerate in the Boolean case in which the reward can be ignored because it would pertain to all entries with $I_{ij} = 0$, would be equal for all such entries, and would hence have no influence on the choice of factors. This explains why the function Cover for the ordinary Asso algorithm does not contain any rewarding term for the overcovered entries.

The above reasoning leads to the following definition of Cover. Let $\mathcal{F}$ denote a set of factors (with a fixed ordering of its elements), i.e. pairs $\langle C, D\rangle$ where $C \in L^{1\times n}$ and $D \in L^{1\times m}$, and let $A_\mathcal{F}$ and $B_\mathcal{F}$ be the matrices defined as in (2.9).

Then we put

$$\begin{aligned}
&\text{Cover}(A_\mathcal{F}, B_\mathcal{F}, I, w^+, w^-) = \\
&\quad +w^+ \cdot \sum\{I_{ij} \leftrightarrow (A_\mathcal{F} \circ B_\mathcal{F})_{ij} \mid I_{ij} \geq (A_\mathcal{F} \circ B_\mathcal{F})_{ij}\} \\
&\quad +w^+ \cdot |\{\langle i, j\rangle \mid I_{ij} < (A_\mathcal{F} \circ B_\mathcal{F})_{ij}\}| \\
&\quad -w^- \cdot \sum\{1 - (I_{ij} \leftrightarrow (A_\mathcal{F} \circ B_\mathcal{F})_{ij}) \mid I_{ij} < (A_\mathcal{F} \circ B_\mathcal{F})_{ij}\}.
\end{aligned} \tag{5.1}$$

The above procedure for computing a set $\mathcal{F}$ of factors is described by Algorithm 7.

Note that the selection in line 4 proceeds by finding for every row $\mathcal{A}_{(i,a)_-}$ of $\mathcal{A}$ the best C w.r.t Cover and then selecting the best found pair $\langle C, \mathcal{A}_{(i,a)_-}\rangle$. Due to the properties of Cover, the best C for a given $\mathcal{A}_{(i,a)_-}$ is found efficiently in a componentwise manner, i.e. by finding the best C_p for every $p = 1, \ldots, n$ (independently of the other C_qs).

Example 7. *Let us have matrix I from Example 2. For $\tau = 0.75$ we obtain association matrix*

Algorithm 7: ASSO$_L$

Input: matrix $I \in L^{n\times m}$, $k \geq 1$, w^+, w^-, τ, $K \subseteq L - \{0\}$
Output: set $\mathcal{F}$ of factors

```
1 compute association matrix 𝒜
2 ℱ ← ∅
3 for l = 1, ..., k do
4 |   select ⟨C, 𝒜_(i,a)_⟩ maximizing COVER(ℱ ∪ {⟨C, 𝒜_(i,a)_⟩}, I, w⁺, w⁻)
5 |   add ⟨C, 𝒜_(i,a)_⟩ to ℱ
6 end
7 return ℱ
```

$$\mathcal{A} = \begin{pmatrix} 1.0 & 1.0 & 0.0 \\ 0.5 & 1.0 & 0.5 \\ 0.0 & 1.0 & 1.0 \end{pmatrix}.$$

For $w^+ = 1$, $w^- = 1$ we obtain two factors $\langle\{^1/a, ^{0.5}/b, ^{0.5}/c\}$,$\{^{0.5}/1, ^1/2, ^{0.5}/3\}\rangle$ and $\langle\{^0/a, ^{0.5}/b, ^0/c\}$,$\{^0/1, ^1/2, ^1/3\}\rangle$. Resulted matrix is

$$A_{\mathcal{F}} \circ B_{\mathcal{F}} = \begin{pmatrix} 0.5 & 1.0 & 0.5 \\ 0.0 & 0.5 & 0.5 \\ 0.0 & 0.5 & 0.0 \end{pmatrix}.$$

5.3 GreConD$_L$+

This algorithm is based on GRECOND$_L$ algorithm which has one big restriction. It never allows the overcover error. This fact is very limiting especially in case where we do not need an exact decomposition.

The algorithm is based on BMF algorithm GRECOND+ presented in [13], which has not been published yet. Our preliminary version of GRECOND$_L$+ is generalization of GRECOND+ and has not been published as well. This preliminary version is included in comparison in experiments in Chapter 6.

GRECOND+ reflects two ideas. First, formal concepts of the factorized matrix form a crucial part of resulted factors. The second idea—inspired by 8M method [25] which is one of the oldest BMF algorithm—already constructed factor could be improved or eliminated depending on the another added factors.

GreConD+ extends algorithm GreConDand in each step returns a formal concept (factor) which covers the most still uncovered entries, i.e. which minimizes uncovered error. Such factor is then extended in a greedy manner by further columns and rows for which the gain of decreased uncovered error is larger than the loss due to overcover error, both formed by added columns/rows. After this phase, where a new factor is created, the set of obtained factors (created in previous iterations) is examined and some of them could be modified or even removed. This step is inspired by 8M method and leads to a decrease of overcover error.

GreConD$_L$+ is generalization of GreConD+ algorithm for data over some scale L. Results from this section are based on [16].

5.3.1 Algorithm GreConD$_L$+

Algorithm presented in this section is modification of previously presented algorithm GreConD$_L$ from [19], described in Section 4.2.2. Pseudocode of this algorithm is depicted below (see Algorithm 8).

The main loop of the algorithm (lines 2–27) is executed until all the nonzero entries of I are covered by at least one factor in $\mathcal{F}$. Clearly, a different stopping criterion is possible—stopping after a prescribed number of factors is computed, which corresponds to the DBP problem, or after the overall error does not exceed ε, which corresponds to AFP. The code between lines 4 and 9 works like original GreConD$_L$ algorithm. In this part the formal concept which covers the maximal part of still uncovered entries, i.e. minimise uncover error, is selected. Resulted concept $\langle C, D\rangle$ is taken as a nucleus and then its expansion $\langle E, F\rangle$ is computed by Expansion algorithm (Algorithm 9). For simplicity Expansion described in Algorithm 9 is restricted to adding columns with positive *gain* (described later) until no $\{^a/_j\}$ with positive *gain* exists. Extension for rows is straightforward. Thus extended factor $\langle C \cup E, D \cup F\rangle$ is added to $\mathcal{F}$ (line 11). The loop between lines 12 and 15 ensures that all matrix entries covered by this factor are removed from $\mathcal{U}$. Last loop (lines 16–26) goes through all the factors and factor is removed from $\mathcal{F}$ iff every non-zero entry is covered by other factors at least in the same degree. If this is not possible, one replace degree of column j in B by degree of j in $nucleus(B)$ for which all non-zero entries are covered by remaining factors (there exist factors covering all entries at least in the same degree as B).

Algorithm 8: GreConD+

Input: $n \times m$ matrix I, number w
Output: set $\mathcal{F}$ of factors

$\mathcal{U} \leftarrow \{\langle i, j\rangle \,|I_{ij} \neq 0\}$; $\mathcal{F} \leftarrow \emptyset$
while $\mathcal{U} \neq \emptyset$ **do**
  $D \leftarrow \emptyset$; $V \leftarrow 0$
  while *exists* $\{^a/_j\} \notin D$ *such that* $|D \oplus_a j| > V$ **do**
    select $\{^a/_j\} \notin D$ that maximizes $|D \oplus_a j|$
    $D \leftarrow (D \cup \{^a/_j\})^{\downarrow\uparrow}$
    $V \leftarrow |D \oplus_a j|$
  end
  $C \leftarrow D^{\downarrow}$
  $\langle E, F\rangle \leftarrow$ Expansion$(\langle C, D\rangle, w)$
  add $\langle C \cup E, D \cup F\rangle$ **to** $\mathcal{F}$
  for $\langle i, j\rangle \in \mathcal{U}$ **do**
    if $I_{ij} \leq (C \cup E)_i \otimes (D \cup F)_j$ **then**
      $\mathcal{U} \leftarrow \mathcal{U} - \langle i, j\rangle$
  end
  foreach *factor* $\langle A, B\rangle \in \mathcal{F}$ **do**
    if *for each* $\langle i, j\rangle$ *with* $A_i \otimes B_j > 0$ *there is* $\langle G, H\rangle \in \mathcal{F} - \langle A, B\rangle$ *with* $A_i \otimes B_j \leq G_i \otimes H_j$ **then**
      remove $\langle A, B\rangle$ **from** $\mathcal{F}$
    else
      foreach j *such that* $B_j > nucleus(B)_j$ **do**
        if *for each* $A_i \otimes B_j$ *there is* $\langle G, H\rangle \in \mathcal{F} - \langle A, B\rangle$ *with* $A_i \otimes B_j \leq G_i \otimes H_j$ **then**
          $B_j \leftarrow nucleus(B)_j$
        end
      end
    end
  end
end
return $\mathcal{F}$

Algorithm 9: EXPANSION

Input: pair $\langle C, D\rangle$, number w
Output: expansion $\langle E, F\rangle$ of $\langle C, D\rangle$

```
1 E ← ∅; F ← ∅
2 repeat
3 |   select column j and a ∈ L such (D ∪ F)_j < a maximizing gain({a/j})
4 |   if gain({a/j}) > 0 then
5 |   |   add {a/j} to F
6 |   end
7 until F did not change
8 return ⟨E, F⟩
```

Function *gain* is in the general case more complicated than the same function in BMF case. There are three cases. Denote

$$\begin{aligned} new_{ij} &= (A_{\mathcal{F}} \circ B_{\mathcal{F}})_{ij} \vee C_i \otimes (D \cup F \cup \{^a/_j\})_j, \\ old_{ij} &= (A_{\mathcal{F}} \circ B_{\mathcal{F}})_{ij} \vee C_i \otimes (D \cup F)_j. \end{aligned}$$

Then

(1) $new_{ij} \leq I_{ij}$, i.e. I_{ij} is still not covered, but coverage is increased by $[I_{ij} \leftrightarrow new_{ij}] - [I_{ij} \leftrightarrow old_{ij}]$, which is equal to $new_{ij} - old_{ij} = \neg(new_{ij} \rightarrow old_{ij})$.

(2) $old_{ij} < I_{ij} < new_{ij}$, we overcover I_{ij}. Value of *gain* needs to be increased by $I_{ij} - old_{ij} = \neg(I_{ij} \rightarrow old_{ij})$, but also decreased by weighted overcover error $w \cdot (new_{ij} - I_{ij})$.

(3) $I_{ij} \leq old_{ij} < new_{ij}$, overcover is increased, so *gain* needs to be also increased by $w \cdot (new_{ij} - old_{ij})$

Function *gain* for $\{^a/_j\}$ returns:

$$\begin{aligned} gain(\{^a/_j\}) &= \sum_{i=1,j=1}^{n,m} \{new_{ij} - old_{ij} \mid new_{ij} \leq I_{ij}\} \\ &+ \sum_{i=1,j=1}^{n,m} \{(I_{ij} - old_{ij}) - w \cdot (new_{ij} - I_{ij}) \mid old_{ij} < I_{ij} < new_{ij}\} \\ &- w \cdot \sum_{i=1,j=1}^{n,m} \{new_{ij} - old_{ij} \mid I_{ij} \leq old_{ij}\} \end{aligned}$$

Example 8. *Let us have matrix I from Example 2. The first obtained factor is* $\langle\{^{1}/a, ^{0.5}/b, ^{0.5}/c\}\ ,\{^{0.5}/1, ^{1}/2, ^{0}/3\}\rangle$*. The corresponding rectangle is equal*

$$\begin{pmatrix} 0.5 & 1.0 & 0.0 \\ 0.0 & 0.5 & 0.0 \\ 0.0 & 0.5 & 0.0 \end{pmatrix}.$$

For small w we can add new column and cover all entries in input matrix by single factor.

Chapter 6

Experimental evaluation

This chapter is devoted to extensive experimental evaluation of algorithms presented in Chapter 4 and Chapter 5. We compare the resulted factorizations quality in two different ways. Firstly in terms how many factors we obtain, secondly we analyse the meaning of these factors. We used both synthetic and real datasets. The advantage of using synthetic datasets is that we can test our algorithms on datasets with known characteristics. On the other hand using real datasets enable us to study meaning of resulted factors.

6.1 Illustrative example

The purpose of this illustrative example is twofold. First, we want to demonstrate a usefulness of the presented methods. Second, show how resulted factorisations look like. This example was originally presented in [8] to demonstrate how the algorithm GreEss$_L$ works. We extend the example to compare results obtained by other algorithms from Chapter 4 and Chapter 5.

The data in Table 6.1 describes 5 most popular dog breeds and their 11 attributes[1] (we analyze the full set of 151 breeds in Section 6.2.1).

We take as the complete residuated lattice six-element Łukasiewicz chain

$$L = \{0, 0.2, 0.4, 0.6, 0.8, 1\}$$

with Łukasiewicz operation $\otimes$ defined in Section 2.1.

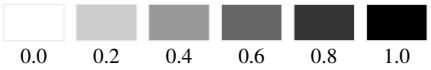
[1] `http://www.petfinder.com/`

The transformation from the table with ordinal scale (Table 6.1) to the matrix with degrees from $L = \{0, 0.2, 0.4, 0.6, 0.8, 1\}$ (Table 6.2) is accomplished using function

$$s : [1, \ldots, 6] \to L \text{ defined by } s_{ij} = \frac{o_{ij} - 1}{5},$$

where i is an object (dog breed), j is an attribute (characteristic) and o_{ij} is original value for the object i and the attribute j.

We represent the grades in L by shades of gray as follows:



In particular, we can attach to the degrees linguistic labels such as "not at all" to 0, "somewhat" to 0.2, "rather not" to 0.4, "rather yes" to 0.6, "almost fully" to 0.8 and "fully" to 1.

The 5×11 object-attribute matrix I and its decomposition $I = A_{\mathcal{F}} \circ B_{\mathcal{F}}$ into the object-factor and factor-attribute matrices $A_{\mathcal{F}}$ and $B_{\mathcal{F}}$ are shown for various algorithms in subsections below. The decomposition is obtained using the algorithms described in Chapters 4 and 5 and utilizes a set $\mathcal{F}$ of formal concepts as factors.

	Energy	Playfulness	Friend. towards dogs	Friend. tow. strangers	Friend. tow. other pets	Protection ability	Exercise	Affection	Ease of training	Watchdog ability	Grooming
Labrador Retrievers	5	6	5	6	6	3	4	6	6	5	3
Golden Retrievers	4	6	6	6	6	3	4	6	6	4	4
Yorkshire terriers	5	5	3	4	3	2	2	4	3	6	5
German shepherds	4	3	2	3	4	6	5	4	6	6	3
Beagles	4	4	6	6	6	2	4	6	2	5	2

Table 6.1: Five most popular dog breeds

	Energy	Playfulness	Friend. towards dogs	Friend. tow. strangers	Friend. tow. other pets	Protection ability	Exercise	Affection	Ease of training	Watchdog ability	Grooming
Labrador Retrievers	0.8	1.0	0.8	1.0	1.0	0.4	0.6	1.0	1.0	0.8	0.4
Golden Retrievers	0.6	1.0	1.0	1.0	1.0	0.4	0.6	1.0	1.0	0.6	0.6
Yorkshire terriers	0.8	0.8	0.4	0.6	0.4	0.2	0.2	0.6	0.4	1.0	0.8
German shepherds	0.6	0.4	0.2	0.4	0.6	1.0	0.8	0.6	1.0	1.0	0.4
Beagles	0.6	0.6	1.0	1.0	1.0	0.2	0.6	1.0	0.2	0.8	0.2

Table 6.2: Five most popular dog breeds

In Figures below is every factor F_l represented by the lth column in $A_{\mathcal{F}}$ and the lth row in $B_{\mathcal{F}}$. The entry $(A_{\mathcal{F}})_{il}$ indicates the degree to which factor l applies to breed i, while $(B_{\mathcal{F}})_{lj}$ represents the degree to which attribute j is a particular manifestation of factor l.

6.1.1 Results for GreConD$_L$

In Figure 6.1 are shown seven factors obtained via GreConD$_L$ algorithm presented in Section 4.2.2.

Factor F_1 is manifested by the three kinds of "Friendliness" and "Affection" (attributes with high degrees in the first row of $B_{\mathcal{F}}$) and applies in particular to Labradors, Golden Retrievers and Beagels (breeds with high degrees in the first column of $A_{\mathcal{F}}$), and to some extent to Yorkshires. The factor may hence be termed *friendliness*. On the other hand, the three attributes with the highest degree in the row of F_2 plus a high degree of "Exercise" tell us that this factor is naturally interpreted as *guardian dog*. The corresponding column shows that F_2 applies to German shepherds and separates them clearly from the other breeds. Factor F_3 may be interpreted as *dogs suitable*

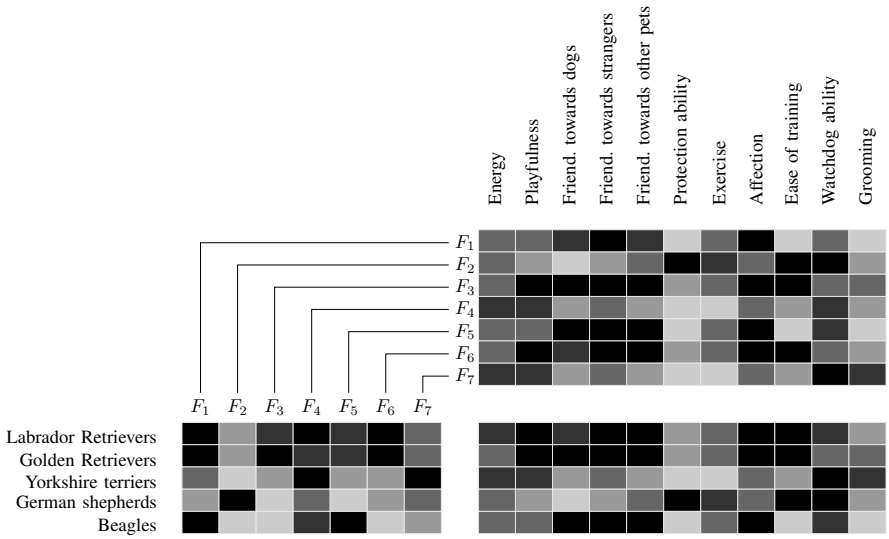


Figure 6.1: GreConD$_L$: Decomposition $I = A_{\mathcal{F}} \circ B_{\mathcal{F}}$. I, $A_{\mathcal{F}}$, and $B_{\mathcal{F}}$ are the bottom-right, bottom-left, and top matrix, respectively.

for kids, because it is manifested by high "Friendliness", "Playfulness", "Affection", and "Ease of training", and applies to Golden Retrievers (in degree 1) and Labrador Retrievers (in degree 0.8).

Interestingly, F_1, F_2, and F_3 explain, by and large, the whole data and hence, the other factors may be neglected. Namely, denoting by $A_{\mathcal{F}_3}$ and $B_{\mathcal{F}_3}$ the 5×3 and 3×11 matrices (parts of $A_{\mathcal{F}}$ and $B_{\mathcal{F}}$), the degree $s(I, A_{\mathcal{F}_3} \circ B_{\mathcal{F}_3})$ of similarity of I to $A_{\mathcal{F}_3} \circ B_{\mathcal{F}_3}$, i.e. reconstructability of the original data I from the first three factors, equals 0.92. In particular, the percentage of matrix I is explained using the first l factors for $l = 1, \ldots 7$, i.e. by the set $\mathcal{F}_l = \{\langle C_1, D_1 \rangle, \ldots, \langle C_l, D_l \rangle\}$, is shown in Figure 6.2, where a concept $\langle C_l, D_l \rangle$ is visualized by the 5×11 matrix $J_{\mathcal{F}_l}$ defined by $(J_{\mathcal{F}_l}) = A_{\mathcal{F}_l} \otimes B_{\mathcal{F}_l}$. One may then also observe the matrix $I_p = \bigvee_{l=1}^{p} J_{\mathcal{F}_l}$ for $p = 1, \ldots, 7$, which results by adding together the first p factors (note that $I_7 = I$).

6.1.2 Results for ordinal scaling

We performed experiments with the approach examined in Section 4.2. We transformed the input matrix $I \in L^{5 \times 11}$ with grades in $L = \{0, 0.2, 0.4, 0.6, 0.8, 1\}$ to a Boolean matrix $I^{\times} \in \{0,1\}^{5 \times (11 \cdot 6)} = \{0,1\}^{5 \times 66}$ and computed a set $\mathcal{G}^{\times} \subseteq \mathcal{B}(I^{\times})$ of factors of $I^{\times}$ using the GreConD algorithm from

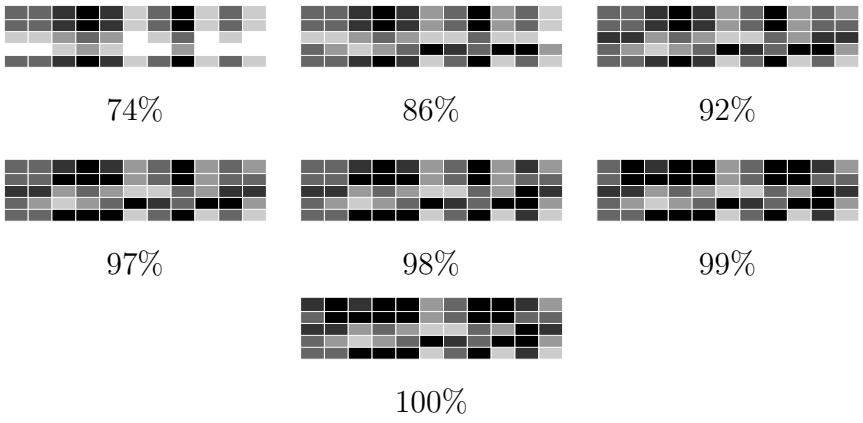


Figure 6.2: Matrices $A_{\mathcal{F}_l} \circ B_{\mathcal{F}_l}$, representing the $\bigvee$-superposition of the set $\mathcal{F}_l = \{\langle C_1, D_1\rangle, \ldots, \langle C_l, D_l\rangle\}$ of the first l factors, $l = 1, \ldots, 7$, and the corresponding percentage of I explained by the first l factors.

[19]. Even though the decomposition algorithms are only approximation algorithms, the experimental results confirm the theoretical ones from Section 4.2—the number of factors of I is in general smaller than the number of factors of $I^\times$. In this case, we obtained 8 factors of $I^\times$, compared to the 7 factors of I obtained for by GreConD$_L$. The factors, $F_1, \ldots, F_8$, are depicted in a concise way in Figure. 6.3. As before, $A_{\mathcal{G}^\times}$ is the bottom-left matrix and its columns represent the factor extents, which are now ordinary sets of objects (breeds). To save space, the 8×66 Boolean matrix $B_{\mathcal{G}^\times}$ is represented by the top 8×11 matrix with grades as follows. For every attribute y, instead of the 6 columns $y_0, y_{0.2}, \ldots, y_1$ of $B_{\mathcal{G}^\times}$, the 8×11 matrix contains just one column which contains in row F_l the largest degree a for which y_a belongs to the intent of F_l. This way, the intent of F_l, an ordinary set of the scaled Boolean attributes y_a, is uniquely described because if y_b is in the intent and $c \leq b$, then y_c in in the intent as well. The corresponding percentage $100 \cdot s_\approx\%$ (which is the same as $100 \cdot s_=\%$ in the Boolean case) of $I^\times$ explained by the first $l = 1, \ldots, 8$ factors is 63%, 81%, 87%, 93%, 98%, 99%, 99.6%, and 100%, respectively.

The factors may naturally be compared to those from Section 6.1.1 and the concise representation of the intents used in Figure 6.3 facilitates this comparison. We may notice that factors F_2, F_4, F_5 and F_6 here are very similar to factors F_6, F_2, F_7 and F_4 respectively from Section 6.1.1, they even pairwise equivalent intents. These factors are clearly interpretable. Factors

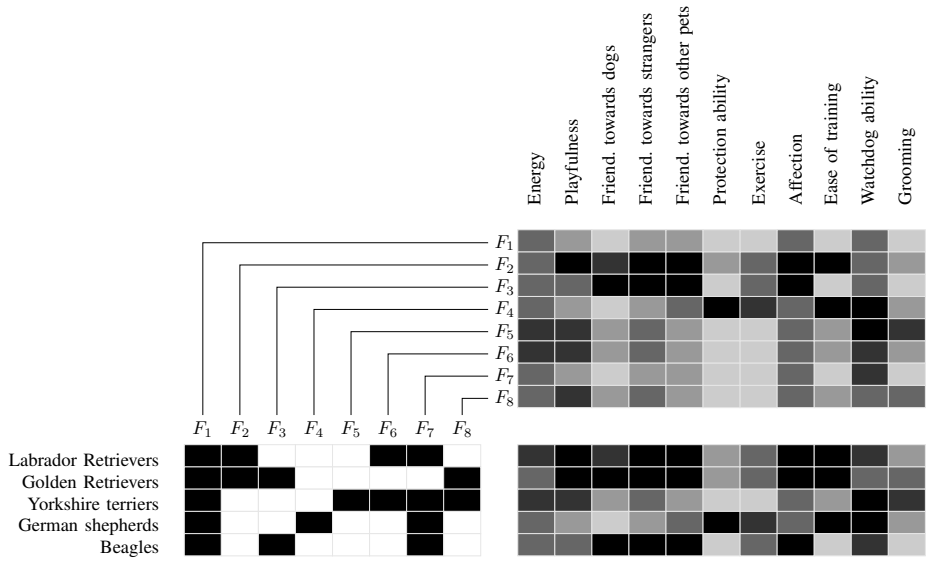


Figure 6.3: Decomposition of $I^{\times} = A_{\mathcal{F}^{\times}} \circ B_{\mathcal{G}^{\times}}$. I, $A_{\mathcal{F}}$, and $\circ B_{\mathcal{G}}$ are the bottom-right, bottom-left, and top matrix, respectively.

F_1, F_3 and F_5 from Section 6.1.1 have some similarities with factors to F_2 and F_3 but interpretation of F_2 and F_3 is not so clear as interpretability the above four. The remaining factors here, F_1, F_7 and F_8 have no counterparts among those in Section 6.1.1 and seem to be not very interesting, particularly F_1, where all attributes are present in small degrees and which applies to all breeds.

To conclude, our experiments confirm that when using the alternative approach examined in Section 4.2, the number of factors needed for explaining data is larger. Moreover, the first, i.e. the most important, factor is not so clearly interpretable and perhaps also not so interesting compared to those obtained by the methods examined in this thesis, which directly works with degrees. We also lost some information such as that Golden Retrievers and Beagles have in high degree characteristics (intent of factor F_6) as Labrador Retrievers and Yorkshire terriers.

6.1.3 Results for NMF

We examine two algorithms for Non-negative matrix factorization presented in Section 4.2.3 such as an Alternating least-squares algorithm and a Multi-

plicative update algorithm. Since resulted matrices W and H include values that are not from scale L, we can not show them as boxes with shades of gray. Interpretation is also slightly different.

Resulted matrices obtained by an alternating least-squares algorithm follow:

$$W = \begin{pmatrix} 1.8423 & 0.1833 & 0.6805 & 0.6591 \\ 2.2351 & 0.0000 & 0.5498 & 0.5300 \\ 1.0538 & 0.0000 & 0.1061 & 1.4114 \\ 0.5271 & 0.0000 & 1.6407 & 0.6844 \\ 0.0000 & 2.0849 & 0.2787 & 0.4019 \end{pmatrix}$$

$$H = \begin{pmatrix} 0.1587 & 0.3955 & 0.4389 & 0.4156 & 0.4021 & 0.0009 & 0.1553 & 0.3739 & 0.3351 & 0.0430 & 0.1154 \\ 0.1856 & 0.2336 & 0.4819 & 0.4459 & 0.4489 & 0.0000 & 0.2274 & 0.4247 & 0.0300 & 0.2130 & 0.0000 \\ 0.1373 & 0.0057 & 0.0000 & 0.0659 & 0.2528 & 0.5660 & 0.4399 & 0.1913 & 0.5061 & 0.3260 & 0.0000 \\ 0.4524 & 0.2791 & 0.0000 & 0.1140 & 0.0000 & 0.0973 & 0.0000 & 0.1350 & 0.0000 & 0.6667 & 0.4819 \end{pmatrix}.$$

The factorization is not exact, matrix product of matrices W and H is a lower rank approximation of I. They are chosen to minimize the root-mean-squared residual between I and product WH. This residual is equal to 0.0358.

Interpretation of obtained factors is a little bit different than interpretation of factors computed via methods based on FCA (such as GreConD_L, GreEss$_L$ etc.). Row i of I is approximately a linear combination of the rows of H with the coefficients being row i of W.

One may observe that attributes with high coefficients in third row are "Protective ability", "Exercise", "Ease of training" similarly like factor F_2 from Section 6.1.1 and with high coefficient belongs to German shepherds in matrix W.

Matrices W and H obtained by Multiplicative update algorithm provide approximation with residual equal to 0.0331. Matrices W and H follow

$$W = \begin{pmatrix} 1.2028 & 0.6604 & 1.0638 & 0.3687 \\ 1.3107 & 0.4606 & 1.2466 & 0.2361 \\ 0.1601 & 0.2178 & 0.5103 & 1.4527 \\ 0.0544 & 1.8614 & 0.4139 & 0.1928 \\ 1.8644 & 0.3788 & 0.0449 & 0.4229 \end{pmatrix}$$

$$H = \begin{pmatrix} 0.1739 & 0.2186 & 0.4913 & 0.4496 & 0.4560 & 0.0008 & 0.2332 & 0.4260 & 0.0201 & 0.1850 & 0.0011 \\ 0.2399 & 0.0601 & 0.0328 & 0.1237 & 0.2508 & 0.5110 & 0.4059 & 0.2353 & 0.3936 & 0.4625 & 0.0968 \\ 0.1746 & 0.5108 & 0.1964 & 0.2440 & 0.2173 & 0.1026 & 0.0610 & 0.2208 & 0.6455 & 0.0361 & 0.2869 \\ 0.4402 & 0.3376 & 0.1424 & 0.2604 & 0.1108 & 0.0052 & 0.0265 & 0.2537 & 0.0029 & 0.5914 & 0.4208 \end{pmatrix}.$$

Similar observation like in case of previous decomposition, also here we can find factor (F_2), which we can be labeled *guardian dog*. Although the matrices W and H are initialized randomly in the algorithm and each run of the experiment returns slightly different results, one can find some similarities in the results of each run.

6.1.4 Results for GreEss$_L$

GreEss$_L$ in this example returns smaller number of factor than earlier mentioned GreCond$_L$. We obtain six factors instead of seven, but all of them are more or less the same as factors obtained in Section 6.1.1. For more details see Figure 6.4.

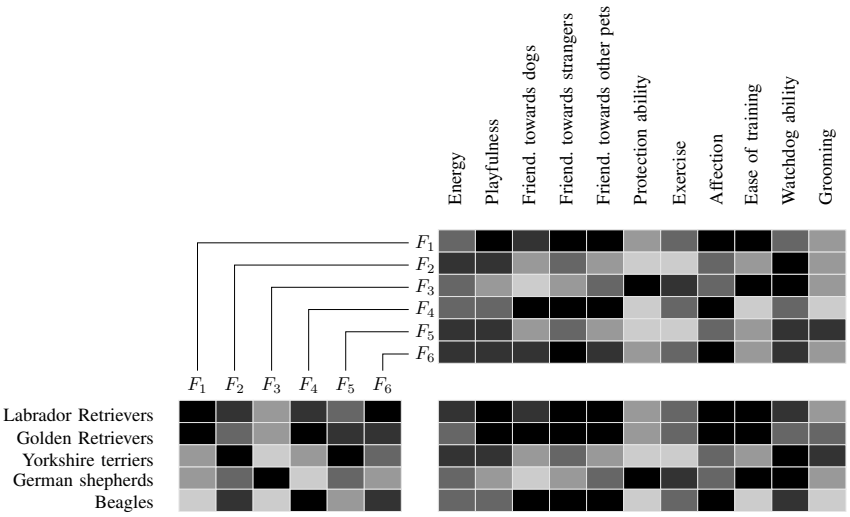


Figure 6.4: GreEss$_L$: Decomposition $I = A_{\mathcal{F}} \circ B_{\mathcal{F}}$. I, $A_{\mathcal{F}}$, and $B_{\mathcal{F}}$ are the bottom-right, bottom-left, and top matrix, respectively.

F_1, F_2, and F_3 explain, by and large, the whole data. Namely, denoting by $A_{\mathcal{F}_3}$ and $B_{\mathcal{F}_3}$ the 5×3 and 3×11 matrices (parts of $A_{\mathcal{F}}$ and $B_{\mathcal{F}}$), the degree $s(I, A_{\mathcal{F}_3} \circ B_{\mathcal{F}_3})$ of similarity of I to $A_{\mathcal{F}_3} \circ B_{\mathcal{F}_3}$, i.e. reconstructability of the original data I from the first three factors, equals 0.92. In particular, the percentage of matrix I explained using the first l factors for $l = 1, \ldots 6$ is 71%, 84%, 92%, 98%, 99% and 100%. In comparison with coverage function in Section 6.1.1) the coverage function grows slower in the first three factors but then converges faster to 100%.

6.1.5 Results for Asso$_L$

As was mentioned earlier, Asso$_L$ usually does not return exact decomposition. We present here three obtained factors for setting: $w_0 = 1$, $w_1 = 1$, i.e.

overcover and uncover errors have same weight and $\tau = 0.9$ (later we will show that choice of τ in ASSO_L does not rapidly change the result).

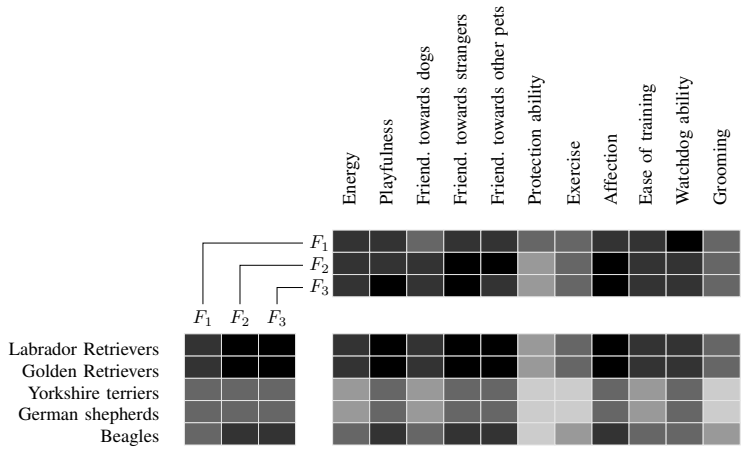


Figure 6.5: ASSO_L: Decomposition $I \approx J = A_{\mathcal{F}} \circ B_{\mathcal{F}}$. J, $A_{\mathcal{F}}$, and $B_{\mathcal{F}}$ are the bottom-right, bottom-left, and top matrix, respectively.

Factors gradually cover 0.72%, 0.84% and 0.844% of input data. Factors are depicted in Figure 6.5 and $\bigvee$-superposition of this factors is depicted in Figure 6.6.

Figure 6.6: $\bigvee$-superposition of factor concepts

The obtained factors are hard to explain, the most important factor (the first one) does not hold any important information. It is caused by the fact that when $|L| > 2$ (non-Boolean case), rectangles with values "around the middle" in L, such as 0.4 and 0.6 in this example, which may be produced as factors by ASSO_L have a good coverage and are thus sometimes selected by ASSO_L in spite of a possible difficulty in interpreting such factors.

In more detail, for Boolean data the values I_{ij} in the input matrix I are approximated by 0 or 1 of $(A_{\mathcal{F}} \circ B_{\mathcal{F}})_{ij}$ only. Hence, in case of mismatch

the entry $\langle i, j \rangle$ contributes by $I_{ij} \leftrightarrow (A_{\mathcal{F}} \circ B_{\mathcal{F}})_{ij} = 0$ to the numerator in (Equation 3.1). With more degrees in L the situation is different. For example, if $\frac{1}{2}$ is available and if $0 \leftrightarrow \frac{1}{2} = 1 \leftrightarrow \frac{1}{2} = \frac{1}{2}$ then already the trivial matrix $A_{\mathcal{F}} \circ B_{\mathcal{F}}$ with all entries equal to $\frac{1}{2}$, which is obtained from the "constant average factors", always satisfies $s(I, A_{\mathcal{F}} \circ B_{\mathcal{F}}) \geq \frac{1}{2}$.

6.1.6 Results for GreConD$_L$+

Like in Asso$_L$, we permit overcover errors in GreConD$_L$+. How big is this error is driven by a choice of the parameter w. The larger w, the smaller overcover error.

For example, if we take $w = 0.5$ we obtain coverage by three factors and overall overcover error is 34%. The first factor covers 85% with overcover error 29%, the second one covers 97% (error 34%), the last one ensures full coverage and does not increase the overcover error. With $w = 1$ we need four factors to cover all inputs and error is 28%. Computed factorizations can be seen in Figures 6.7 and 6.8.

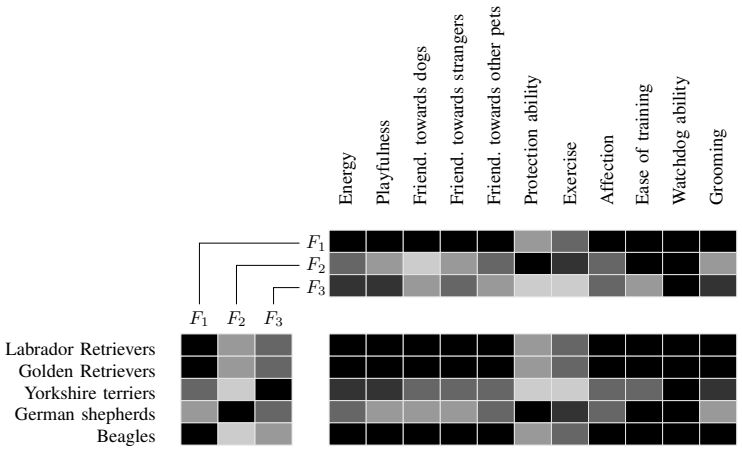


Figure 6.7: GreConD$_L$+, $w = 0.5$: Decomposition $I \approx J = A_{\mathcal{F}} \circ B_{\mathcal{F}}$. J, $A_{\mathcal{F}}$, and $B_{\mathcal{F}}$ are the bottom-right, bottom-left, and top matrix, respectively.

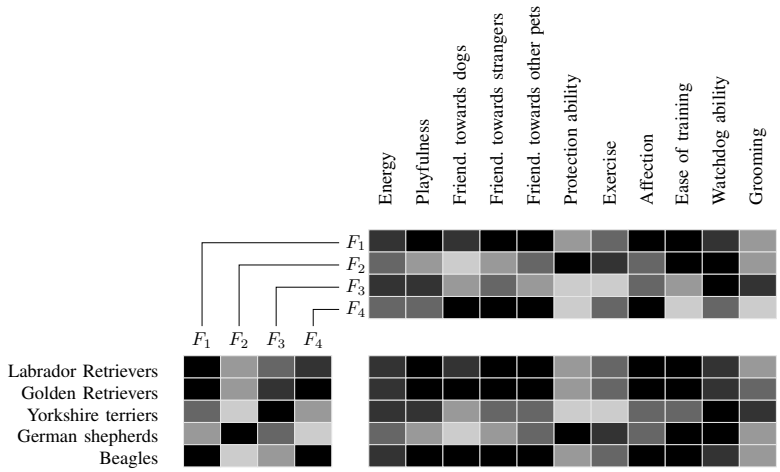


Figure 6.8: GreConD$_L$+, $w = 1$: Decomposition $I \approx J = A_{\mathcal{F}} \circ B_{\mathcal{F}}$. J, $A_{\mathcal{F}}$, and $B_{\mathcal{F}}$ are the bottom-right, bottom-left, and top matrix, respectively.

The choice of w slightly changes the obtained factors, but the most important factors (first ones), are very similar.

Unlike factors obtained by Asso$_L$, meaning of factors is more relevant. We can see that there are factors *friendliness*, *guardian dog* like in Section 6.1.1, i.e. factors F_1 and F_2 are nearly the same as factors F_1 and F_2 from Section 6.1.1.

6.1.7 Choice of the scale of degrees

Due to nature of input data in the above examples we used a six-element scale L equipped it with Łukasiewicz operations.

In general it is advisable to choose the number of degrees from five to nine. This is motivated by the well-known Miller's 7 ± 2 phenomenon [48]. Small scales L with up to 7 ± 2 degrees are preferable because humans can understand and use such scales easily.

In [7] we indeed experienced with a larger number of degrees, say 10, the undesired effect which is in accordance with the 7 ± 2 phenomenon. Among the extracted factors there were ones which, although mutually distinct, appeared similar to a human expert. The reason behind it was that they involved closed degrees, which are intuitively not distinct enough, such as 0.7 and 0.8. This impairs the interpretation of the factors in that even

though each of the extracted factors alone makes a relatively good sense, the extracted factors do not satisfy the intuitive requirement of clear distinctiveness from each other. Moreover, for the purpose of identifying and linguistically labeling the most important factors, a relatively smaller number of degrees seems sufficient. In addition to that, with more degrees of L in the input matrix I, resulting in the process of scaling I becomes more complex and thus the number of factors to decompose I gets larger. Therefore, "too much precision" introduced by having a larger L impairs the practical aspects of the analysis in several ways. We illustrate it on the dataset from [18]—results of top 5 athletes in the 2004 Olympic Decathlon. Originally five-element Łukasiewicz chain was used. Figure 6.9 shows the factors obtained using GreConD$_L$. Figure 6.10 shows the result of decomposition of matrix I representing the same decathlon data but using a eleven-element scale $L = \{0, 0.1, \ldots, 0.9, 1\}$. We can see that, for instance, F_1, F_3, and F_6 are rather similar to F_4, F_1, and F_3 from set of factors obtained using smaller L. However, several factors here are mutually similar and difficult to distinguish.

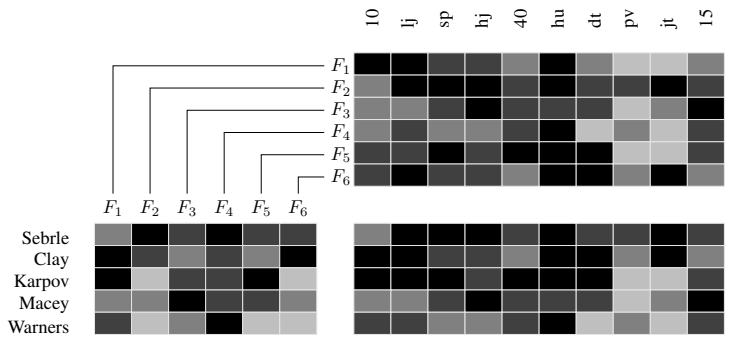


Figure 6.9: Decomposition $I = A_{\mathcal{F}} \circ B_{\mathcal{F}}$, using Łukasiewicz operations.

In the previous examples we used Łukasiewicz operations, because of some of its intuitive properties discussed in Section 2.1. Our experience is that with the other operations, we obtain different factors but the corresponding sets of factors have important factors in common. Figure 6.11 shows the result of decomposition of the matrix I from [18] using Gödel operations. Notice that F_1, F_2, and F_3 obtained using Łukasiewicz operations are similar to F_7, F_2, and F_8 here. Let us note that F_1, F_2, and F_3 obtained using Łukasiewicz operations also have their counterparts among the factors obtained using the

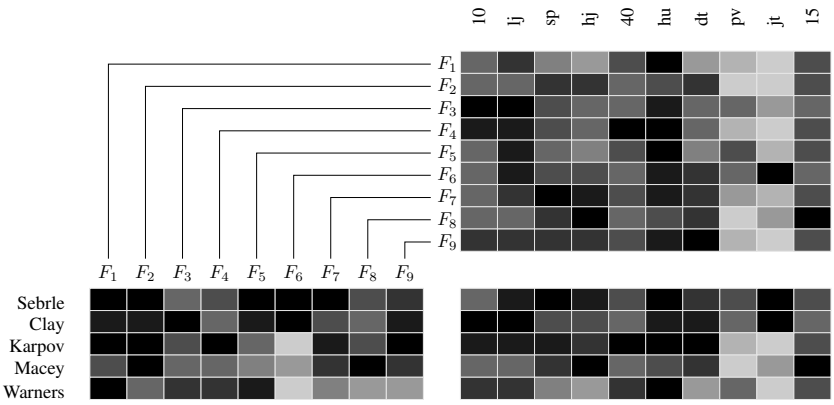


Figure 6.10: Decomposition $I = A_{\mathcal{F}} \circ B_{\mathcal{F}}$ using eleven-element scale L.

Goguen operations. Apparently, a more profound treatment of the problem of the choice of the operations on the scales is needed. This concerns not only factor analysis and FCA in fuzzy setting but fuzzy set theory and its applications in general. In this regard, the theory of measurement from mathematical psychology seems an appropriate framework for such considerations (see [6] for some first steps in FCA in a fuzzy setting).

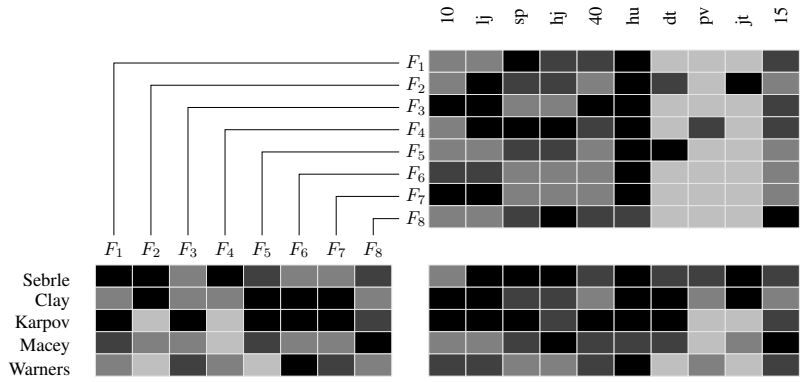


Figure 6.11: Decomposition $I = A_{\mathcal{F}} \circ B_{\mathcal{F}}$, using Gödel operations.

6.2 Real data

In this section we present results of selected analyses of real data which we used in our evaluation. Our primary aim is to illustrate that the presented decomposition methods can extract natural and easy-to-understand factors from ordinal data. The datasets and their characteristics are described in Table 6.3, in which $|L|$ denotes the number of truth degrees in the scale L and $||I||$ denotes the number of non-zero entries in the input matrix I. Since we are interested also in analysis via algorithm GreEss$_L$, another interesting characteristic is number of non-zero entries in the essential part $\mathcal{E}(I)$. As one can see, the reduction in the essential part $\mathcal{E}(I)$ compared to that of I is significant which is an important fact in view of the results in Section 5.1.1.

Most of in this section presented datasets and results for Asso$_L$ and GreEss$_L$ was already presented in [8], however new dataset Rio was added and also results for other presented algorithms—namely GreConD on ordinally scaled attributes, GreConD$_L$ GreConD$_L$+ and Asso$_L$—are included.

The factors obtained by Asso$_L$ are generally not so easy to interpret compared to those obtained by methods based on FCA—a feature which we observed on the most examples—but reveal, in most cases, a similar insight. The reason is a good interpretability of formal concepts and the usage of formal concepts as factors. Another reason is described in Section 6.2.1. Factors obtained by GreConD$_L$, GreEss$_L$ and GreConD$_L$+ are basically very similar. This is why we mainly focus on describing the factors obtained by GreEss$_L$, unless those of other methods reveal a different insight.

dataset	size	$\|L\|$	$\|\|I\|\|$	$\|\|\mathcal{E}(I)\|\|$	$\|\|\mathcal{E}(I)\|\|/\|\|I\|\|$
Breeds	151×11	6	1963	362	0.184
Decathlon	28×10	5	266	59	0.221
IPAQ	4510×16	3	41624	1281	0.031
Music	900×26	7	20377	5952	0.292
Rio	87×31	4	402	332	0.820

Table 6.3: Real data

Dog breeds[2] extends the dataset from Section 6.1 to 151 breeds. GreEss$_L$ found 20 factors providing an exact decomposition of the 151×11 matrix I, but already the set $\mathcal{F}_3$ consisting of the first three most important factors explains a large portion of the data. In particular, the degree $s(I, A_{\mathcal{F}_3} \circ B_{\mathcal{F}_3})$ of closeness of I to the matrix $A_{\mathcal{F}_3} \circ B_{\mathcal{F}_3}$ reconstructed from the first three factors, which is defined by (3.1), equals 0.795. Among these factors is a formal concept containing the attributes "playfulness", "ease of train", and "affection" to degree 1 and "energy" to a high degree. This factor may be interpreted as the ability to *excel in sports* (such as agility, flyball, frisbee) and to serve as *guide* and *therapy dogs*. This factor applies to high degree to breeds such as Golden Retriever, Labrador Retriever, or Papillon. Another factor is a formal concept containing "Protection ability" and "Watchdog ability" with high degrees. Such factor may be interpreted as the ability to serve as a *guardian dog* and applies e.g. to American Staffordshire Terrier, Anatolian Shepherd, Belgian Malinois, Belgian Sheepdog, Kuvasz, German Shepherd Dog, and Doberman Pinscher. Interestingly, these two factors are similar to factors F_3 and F_2 described in Section 6.1. In fact, the factors F_1, F_2, and F_3 from Section 6.1, when extended to the 151×11 (in terms of Section 3.4.1) matrix, cover 0.85 of the matrix according to s, illustrating an interesting natural property that we observed in several examples.

Decathlon[3] extends the dataset from [18] to 28×10 matrix I (28 athletes, 10 disciplines of decathlon) using a five-element scale L. Table 6.4 contains the matrix I restricted to the first 10 athletes.

Using GreEss$_L$, we obtained 10 factors that we consulted with an experienced decathlon coach. Among the most important factors are the ones that can be interpreted as *speed*, containing to high degrees the attributes "100m", "Long jump", "400m", and "Hurdles"; *explosiveness*, containing to high degrees the attributes "Long jump", "Shot put", "High jump", and "Javelin"; and a factor containing "High jump" and "1500m", typical of light-weight athletes. All these factors were found natural by the decathlon coach.

Asso$_L$ computed a set $\mathcal{F}$ of 5 factors which reconstruct 80% of the the input data, i.e. $s(I, A_{\mathcal{F}} \circ B_{\mathcal{F}}) = 0.8$. Covering by factors is in order 76% for first factor, 86% for first two factors, 87%, 88% and 88, 2% for all five factors.

The most interesting is the second most important in terms of coverage,

[2] `http://www.petfinder.com/`

[3] `http://www.sports-reference.com/`

	10	*lj*	*sp*	*hj*	40	*hu*	*di*	*pv*	*ja*	15
Sebrle	0.50	1.00	1.00	1.00	0.75	1.00	0.75	0.75	1.00	0.75
Clay	1.00	1.00	0.75	0.75	0.50	1.00	1.00	0.50	1.00	0.50
Karpov	1.00	1.00	1.00	0.75	1.00	1.00	1.00	0.25	0.25	0.75
Macey	0.50	0.50	0.75	1.00	0.75	0.75	0.75	0.25	0.50	1.00
Warners	0.75	0.75	0.50	0.50	0.75	1.00	0.25	0.50	0.25	0.75
Zsivoczky	0.50	0.50	0.75	1.00	0.50	0.50	0.50	0.50	0.75	0.75
Hernu	0.50	0.50	0.50	0.50	0.75	0.75	0.50	0.50	0.25	1.00
Nool	0.75	0.75	0.50	0.25	0.75	0.50	0.25	1.00	0.50	0.75
Bernard	0.75	0.75	0.50	1.00	0.50	1.00	0.50	0.25	0.25	0.75
Schwarzl	0.50	0.75	0.25	0.25	0.50	0.75	0.25	0.75	0.25	0.75

Legend: 10—100 meters sprint race; *lj*—long jump; *sp*—shot put; *hj*—high jump; 40—400 meters sprint race; *hu*—110 meters hurdles; *di*—discus throw; *pv*—pole vault; *ja*—javelin throw; 15—1500 meters run.

Table 6.4: 2004 Olympic Games Decathlon

which contains "1500m" with degree 1, and "100m", "400m", and "Hurdles" with degree 0.75, i.e. a factor that may be naturally termed as *running capability*.

In this example, we observed factors which are not easy to interpret and whose appearance is discussed in Section 6.2.1, namely factors which apply to relatively higher degrees to all athletes and are manifested to high degrees by every discipline.

IPAQ data[4] consists of international questionnaire data regarding physical activity of population and involves 4510 respondents answering 16 questions using a three-element scale. This questionnaire is considered important from health management point of view, particularly as a source for making government decisions regarding health policy. The questions include respondents age, sex, body-mass-index (BMI), health, to what extent the person bicycles, walks, etc. GreEss$_L$ produced 17 factor concepts providing an exact decomposition of the 4510×16 matrix. As with the other examples, the first 3–4 factors may be considered sufficient to explain the data. First

[4] `http://www.ipaq.ki.se/`, Belohlavek et al., Inf. Sciences 181(2011), 1774–1786.

factor explains 64%, first two factors 74% and three factors explain 80% of data. One may see great reduction of input entries using essential part in this dataset. GreEss$_L$ returns the smallest number of factors compared to other methods providing exact decomposition. On the other hand factors obtained via GreConD$_L$ or GreConD$_L$+ are more interesting, because factors obtained via GreEss$_L$ correspond more or less to a single attribute. Based on the attributes present in the factors, the first factor returned from GreConD or GreConD+ corresponds to and thus may be interpreted as *healthy people with good education who cycle on a regular basis*; the second one as *people with normal BMI who walk on a daily basis*; the third one as *people who are employed, own a car, and cycle on a regular basis.* All these groups are considered important according to a kinanthropologist expert.

Music data The data comes from [27] and consists of results of a study inquiring people's perception of some speed of song depending of various characteristics of the songs. The data was collected by questionnaires involving 30 participants who were presented 30 samples (29 complex music samples and one simple tone of 528Hz). The participants recorded their emotional experience using 26 attributes (such as "Pleasant", "Happy", "Exciting", "Restful", "Intelligible", "Ugly", "Valuable", "Interesting", "Slow", "Meaningful", "Active", "Tense", "Predictable", "Closed", "Violent", "Strong", "Known", "Variable", or "Like it"), each using a 6-element scale L, along with a retrospective time duration and time passage judgement. The data is then represented by a 900×26 matrix with entries in L. Using GreEss$_L$, we obtained 29 factors. The authors of this study examined the factors and concluded that the groups of music samples corresponding to the factors are meaningful and that the factors can be interpreted in terms of emotional experience. For example, an interesting factor with a good coverage contained songs No. 5, 7, 16, and 26, all of which are melancholic. Another factor was the one clearly separating the simple tone to which it applied to degree 1, while applying to degree 0 or to other degrees close to 0 to the other samples. Among other interesting factors are the one manifested to a high degree by attributes "Ugly" and "Violent"; the one manifested by "Restful", "Safe", "Stable", and "Inert"; and the factor manifested by "Successful", "Valuable", "Meaningful", and "Significant". All these factors represent significant categories of songs.

Rio data [5] represents 87×31 matrix I and consist of 87 countries that obtained any medal in one of 31 sport (such as Archery, Athletics, Badminton, Basketball, Boxing, ...) on Olympics games in Rio de Janeiro 2016. L contains four grades—1 means that country won at least one gold medal, $\frac{2}{3}$ at least silver medal, $\frac{1}{3}$ at least one bronze medal and 0 no medal in this sport. This dataset is very sparse in comparison with other presented datasets. Great portion of input entries are Essential, i.e. we observe that the ratio $||\mathcal{E}(I)||/||I||$ of the number of entries in $\mathcal{E}(I)$ to the corresponding number for I is high.

Using GreEss$_L$ we computed 32 factors, but it is sufficient take 19 factors to explain more than 90% of data.

Among the most important factors can be found factor containing *martial arts*, which has degree 1 in the attributes "Judo", "Wrestling" and high degree in "Weightlifting". Another one can be interpreted as *water sports*, containing in high degrees the attributes "Canoeing", "Rowing", "Sailing", and "Swimming".

Asso$_L$ returned different factors. Let us mentioned one factor, which grouped attributes "Archery" and "Shooting", i.e. sports with *the ability to aim*.

6.2.1 Evaluation

Table 6.5 and Table 6.6 display the numbers of factors produced by the algorithms from Section 4 and Section 5 that are needed to achieve a prescribed coverage. That is, we observe the smallest l such that for the set $\mathcal{F}$ of the first l factors produced by the respective algorithm, $s_{\approx}(I, A_{\mathcal{F}} \circ B_{\mathcal{F}})$ ($s_{=}(I, A_{\mathcal{F}} \circ B_{\mathcal{F}})$ respectively) exceeds the prescribed value. For example, the first row in Table 6.5 corresponding to Breeds indicates that we need six factors in case of GreConD of ordinally scaled attributes, three in case of GreConD$_L$, two in case of GreEss$_L$ etc. to have $s_{\approx}(I, A_{\mathcal{F}} \circ B_{\mathcal{F}}) \geq 0.75$. "NA" indicates that the prescribed coverage is not achievable by the factors produced by Asso$_L$. Observe that in accordance with the theoretical results, "NA" never appears for other algorithm than Asso$_L$, because the other algorithms eventually compute an exact decomposition.

[5] `https://www.rio2016.com/en/medal-count`

dataset	s	ordinally scaled attributes	number of factors needed				
			GreConD$_L$	Asso$_L$	GreEss$_L$	GreConD$_L$+ ($w = 0.5$)	GreConD$_L$+ ($w = 1$)
Breeds	0.75	6	3	2	3	1	1
	0.85	12	5	3	7	2	3
	0.95	25	9	NA	11	6	8
	1	57	16	NA	15	14	13
Decathlon	0.75	5	1	1	3	1	1
	0.85	8	4	2	5	1	2
	0.95	16	8	NA	8	4	6
	1	31	15	NA	10	11	13
IPAQ	0.75	6	8	1	10	2	2
	0.85	10	12	1	12	3	4
	0.95	19	18	NA	15	8	9
	1	46	32	NA	17	20	23
Music	0.75	28	7	1	7	3	1
	0.85	51	13	NA	14	5	6
	0.95	105	24	NA	25	13	18
	1	280	36	NA	29	29	30
Rio	0.75	1	12	1	2	1	1
	0.85	9	16	1	6	1	1
	0.95	30	24	18	17	5	8
	1	79	35	NA	32	32	33

Table 6.5: Quality of decompositions (real data) for $s_{\approx}$.

dataset	s	ordinally scaled attributes	number of factors needed: GreConD$_L$	Asso$_L$	GreEss$_L$	GreConD$_L$+ ($w = 0.5$)	GreConD$_L$+ ($w = 1$)
Breeds	0.50	1	7	NA	3	NA	5
	0.75	6	7	NA	3	NA	NA
	0.95	25	12	NA	11	NA	NA
	1	57	16	NA	15	NA	NA
Decathlon	0.50	3	3	NA	3	1	2
	0.75	5	5	NA	6	NA	NA
	0.95	16	11	NA	9	NA	NA
	1	31	15	NA	10	NA	NA
IPAQ	0.50	1	1	NA	2	2	1
	0.75	6	6	NA	7	NA	10
	0.95	19	28	NA	14	NA	NA
	1	46	32	NA	17	NA	NA
Music	0.50	7	10	NA	9	NA	25
	0.75	28	20	NA	19	NA	NA
	0.95	105	34	NA	26	NA	NA
	1	280	36	NA	29	NA	NA
Rio	0.50	1	1	1	1	1	1
	0.75	1	1	1	1	1	1
	0.95	30	12	NA	13	NA	17
	1	79	35	NA	32	NA	NA

Table 6.6: Quality of decompositions (real data) for $s_=$.

Results for $s_{\approx}$ The results illustrate that, by and large, the number factors produced by GreConD on dataset with ordinally scaled attributes is significantly larger in comparison with other algorithms. Moreover the first couple of factors produced by Asso_L and GreConD_L+ has a better coverage compared to the same number of factors produced by GreEss_L or GreConD_L. On the other hand, beyond certain coverage, Asso_L stops producing factors and is not able to compute an (exact) decomposition of I, while other algorithms always compute an exact decomposition, with a reasonably small number of factor needed for coverage very close to 1. This is congruent with the fact that Asso_L and GreConD_L+ are primarily designed for $\text{DBP}(L)$ and the rest of the algorithms is primarily designed for $\text{AFP}(L)$, as well as with the available evidence from the Boolean case.

We found that factors produced by Asso_L are not easy to interpret compared to other algorithms. There are two reasons. The first, mentioned in Section 6.2, is the usage of formal concepts as factors by GreConD, GreConD_L, GreEss_L, GreConD_L+ and their good interpretability. The second one consists in that when $|L| > 2$ (non-Boolean case) rectangles with values "around the middle" in L, such as $\frac{1}{2}$, which may be produced as factors by Asso_L have a good coverage and are thus sometimes selected by Asso_L in spite of a possible difficulty in interpreting such factors. In more detail, note that for Boolean data, the values I_{ij} in the input matrix I are approximated by 0 or 1 of $(A_{\mathcal{F}} \circ B_{\mathcal{F}})_{ij}$ only. Hence, in case of mismatch the entry $\langle i, j \rangle$ contributes by $I_{ij} \leftrightarrow (A_{\mathcal{F}} \circ B_{\mathcal{F}})_{ij} = 0$ to the numerator in (3.1). With more degrees in L, the situation is different. For example, if $\frac{1}{2}$ is available and if $0 \leftrightarrow \frac{1}{2} = 1 \leftrightarrow \frac{1}{2} = \frac{1}{2}$, then already the trivial matrix $A_{\mathcal{F}} \circ B_{\mathcal{F}}$ with all entries equal to $\frac{1}{2}$, which is obtained from the "constant average factors", always satisfies $s_{\approx}(I, A_{\mathcal{F}} \circ B_{\mathcal{F}}) \geq \frac{1}{2}$. One therefore has to be aware of this effect of presence in L of the "middle" degrees on the values of $s_{\approx}$.

New algorithm GreEss_L requires less factors to achieve a prescribed coverage than the previous algorithm GreConD_L from [18]. The reason is a better utilization of the geometry of decompositions by GreEss_L particularly of the essential part of I.

Results for $s_{=}$ As was mentioned in Section 6.1.2 in Boolean case (algorithm GreConD on ordinally scaled attributes) it holds that $s_{\approx}$ is equal to $s_{=}$, so results in corresponding column are the same. Big change is in case of GreConD_L+. This algorithm like Asso_L algorithm allows overcover error,

so we are not able usually achieve $s_{=} = 1$. This error grows with smaller parameter w. More precisely in Breed dataset finally only 47% of entries matrix I are the same as appropriate element in $A_{\mathcal{F}} \circ B_{\mathcal{F}}$ for parameter $w = 0.5$ and 65% for parameter $w = 1$. In Table 6.7 we show the percentage of the same entries for all datasets.

dataset	Asso_L	GreConD_L+ $(w = 0.5)$	GreConD_L+ $(w = 1)$
Breed	29%	47%	65%
Decathlon	24%	51%	73%
IPAQ	24%	69%	80%
Music	12%	35%	52%
Rio	88%	87%	97%

Table 6.7: Percentage of $s_{=}$

6.3 Synthetic data

We used synthetic data organized in collections Set 1–5, each consisting of 500 $n \times m$ matrices I. The characteristics of these datasets are described in Table 6.8. Each matrix I is obtained as a product of $n \times k$ and $k \times m$ randomly generated matrices A and B in which entries from scale L are selected according to a prescribed probability distribution. For instance, in Set 2 we used a five-element scale $L = \{0, \frac{1}{4}, \frac{1}{2}, \frac{3}{4}, 1\}$ with the probabilities $p(a)$ of the degrees $a \in L$ in A and B being $p(0) = p(\frac{1}{4}) = \frac{1}{8}$ and $p(\frac{1}{2}) = p(\frac{3}{4}) = p(1) = \frac{1}{4}$. The probability distributions generalize the commonly considered densities of Boolean matrices, e.g. for $L = \{0, 1\}$ the distribution $[\frac{1}{4}\ \frac{3}{4}]$ corresponds to density 0.75. Table 6.9 contains the average characteristics of synthetic data with the averages over all matrices in Set i. The characteristics are the same as for the real data. One observes that the reduction in number of nonzero entries is significant as in the case of real data. We present similar experiment in [8]. Set 5 was added and the rest of the sets have the same characteristic, but they are different, for the purpose of this thesis we generate new ones.

dataset	size	$\|L\|$	k	distribution on L in A and B
Set 1	50×50	3	10	$[\frac{1}{3}\ \frac{1}{3}\ \frac{1}{3}]$
Set 2	50×50	5	10	$[\frac{1}{8}\ \frac{1}{8}\ \frac{1}{4}\ \frac{1}{4}\ \frac{1}{4}]$
Set 3	100×50	5	25	$[\frac{1}{8}\ \frac{1}{8}\ \frac{1}{4}\ \frac{1}{4}\ \frac{1}{4}]$
Set 4	100×100	5	20	$[\frac{1}{8}\ \frac{1}{8}\ \frac{1}{4}\ \frac{1}{4}\ \frac{1}{4}]$
Set 5	500×100	6	25	$[\frac{1}{6}\ \frac{1}{6}\ \frac{1}{6}\ \frac{1}{6}\ \frac{1}{6}\ \frac{1}{6}]$

Table 6.8: Synthetic data.

dataset	avg $\|\|I\|\|$	avg $\|\|\mathcal{E}(I)\|\|$	avg $\|\|\mathcal{E}(I)\|\|/\|\|I\|\|$
Set 1	2449	195	0.080
Set 2	2503	355	0.141
Set 3	4983	602	0.121
Set 4	10000	2087	0.209
Set 5	49997	14216	0.284

Table 6.9: Characteristics of synthetic data.

6.3.1 Evaluation of explanation data

We now provide an experimental evaluation of the presented algorithms on synthetic data. We observe the ability of the extracted factors to explain, i.e. reconstruct, the input data and measure it by the degree of similarity $s_{\approx}(I, A_{\mathcal{F}} \circ B_{\mathcal{F}})$ defined by (3.1), where $\mathcal{F}$ is the examined set of factors (usually the first k factors obtained by the algorithm). In view of Section 2.3, we speak of coverage of data by factors.

Table 6.10, Table 6.11 and Figure 6.12 display selected results of coverage $s_{\approx}$, defined by (3.1), by the first k factors for the datasets and the two algorithms. We also include the percentage $s_{=}$ of entries $\langle i, j \rangle$ for which $I_{ij} = (A_{\mathcal{F}_k} \circ B_{\mathcal{F}_k})_{ij}$, where $\mathcal{F}_k$ is the set of the first k factors. $s_{=}$ is a stronger measure than $s_{\approx}$ since it does not take into account the closeness of the corresponding entries in I and $A_{\mathcal{F}_k} \circ B_{\mathcal{F}_k}$ and only considers equal entries. "–" means that no new factors were produced increasing k.

Note that the values of $s_{\approx}$ tend to be high even for a small number of computed factors and that they are higher than what one usually observes for Boolean data. The reason is the same like in the case of real datasets and is explained in Section 6.2.1.

In results on synthetics data, we observe the same behaviour as in case of real datasets presented in Section 6.2.1.

dataset	k	ordinally scaled attributes	coverage $s/s_=$ by the first k factors GreConD$_L$	Asso$_L$	GreEss$_L$	GreConD$_L$+ ($w = 0.5$)	GreConD$_L$+ ($w = 1$)
Set 1	1	0.648	0.576/0.350	0.878/0.761	0.525/0.309	0.745/0.470	0.745/0.470
	4	0.837	0.866/0.744	0.899/0.805	0.866/0.744	0.943/0.780	0.936/0.781
	11	0.975	0.992/0.985	–	1/1	1/0.858	0.999/0.892
	12	0.982	0.995/0.990	–	–	–	1/0.892
	17	0.999	1/1	–	–	–	–
	19	1	–	–	–	–	–
Set 2	1	0.674	0.620/0.253	0.795/0.389	0.632/0.206	0.836/0.410	0.836/0.410
	2	0.763	0.782/0.434	0.839/0.410	0.820/0.483	0.921/0.524	0.921/0.524
	10	0.958	0.995/0.980	–	1/1	0.999/0.683	0.998/0.735
	11	0.967	0.997/0.989	–	–	1/0.684	0.999/0.738
	12	0.975	0.998/0.524	–	–	–	1/0.738
	13	0.980	1/1	–	–	–	–
	23	1	–	–	–	–	–
Set 3	1	0.780	0.684/0.188	0.899/0.789	0.728/0.349	0.852/0.412	0.852/0.412
	3	0.845	0.828/0.386	0.950/0.807	0.790/0.508	0.923/0.632	0.923/0.632
	19	0.966	0.966/0.867	–	0.979/0.950	1/1	0.998/0.867
	27	0.987	0.986/0.947	–	0.998/0.977	–	1/1
	39	0.998	0.998/0.994	–	1/1	–	–
	47	0.999	1/1	–	–	–	–
	53	1	–	–	–	–	–

Table 6.10: Coverage $s_\approx$ and $s_=$ by the first k factors.

dataset	k	ordinally scaled attributes	coverage $s/s_=$ by the first k factors GreConD$_L$	Asso$_L$	GreEss$_L$	GreConD$_L$+ ($w = 0.5$)	GreConD$_L$+ ($w = 1$)
Set 4	1	0.738	0.651/0.211	0.921/0.704	0.648/0.205	0.846/0.427	0.846/0.427
	4	0.840	0.827/0.603	0.939/0.722	0.854/0.512	0.967/0.694	0.963/0.701
	21	0.985	0.975/0.824	–	0.994/0.979	1/0.771	0.999/0.805
	27	0.997	0.998/0.905	–	1/1	–	1/0.808
	29	0.998	1/1	–	–	–	–
	37	1	–	–	–	–	–
Set 5	1	0.569	0.511/0.111	0,886/0,535	0.477/0.068	0.765/0.234	0.765/0.234
	5	0.750	0.821/0.419	0,887/0,541	0.798/0.328	0.953/0.546	0.931/0.531
	27	0.948	0.995/0.979	–	0.995/0.949	1/0.621	0.999/0.701
	31	0.964	0.998/0.990	–	0.999/0.974	–	1/0.832
	36	0.974	0.999/0.999	–	1/1	–	–
	42	0.986	1/1	–	–	–	–
	109	1	–	–	–	–	–

Table 6.11: Coverage $s_\approx$ and $s_=$ by the first k factors ctd.

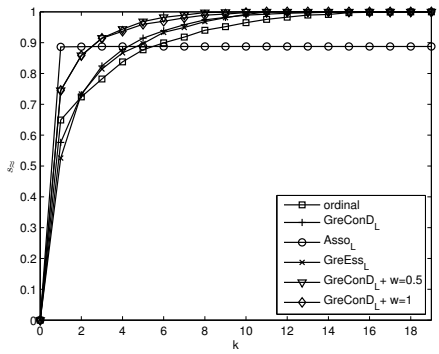
6.3.2 Selection of smaller I from J

In this section we provide experimental evaluation of heuristic algorithm presented in Section 3.4.2. We run two kind of experiments.

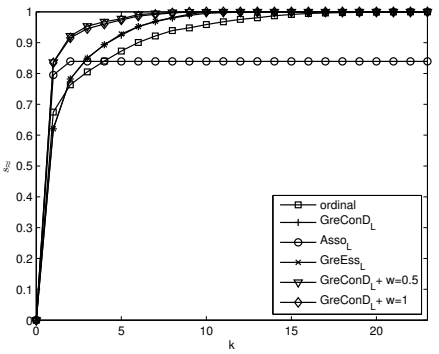
The first one is a comparison of coverage matrix J by factors from I obtained by our algorithm with coverage by optimal I. Optimal I means that we compute factors and then coverage for all combination of k rows. Because of time complexity of this approach, we compute it on smaller datasets (as well as real datasets from previous section). Choice by our algorithm is not unambiguous. Usually there is more rows with same number of essential elements—in this case, we try all possible combinations and take them as a set of solutions. In the most instances of our algorithm, the optimal solution is in the set of our solutions or at least coverage obtained by solutions from the set are very close to optimal coverage (difference is in average 4%).

Goal of the second type of experiments is to determine how many rows k must be selected to obtain a good coverage. We take collections of sets from Section 6.3. Their characteristics are described in Table 6.8. For each set we compute coverage when we take in order one to 50 percent of rows (selected via our heuristic). In Figure 6.13 we show averaged final coverage depending on the parameter k for each collection of sets.

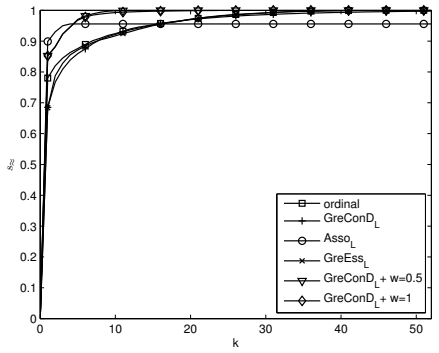
Is sufficient in average to take 20% of rows to obtain coverage greater than 90% (30% of rows for coverage greater than 95%). This reduction is significant and leads to computation factorization of smaller matrix. Since the time complexity of algorithms for matrix decomposition presented in this thesis depends on size of input matrix, this leads to a faster computation of approximate decomposition.



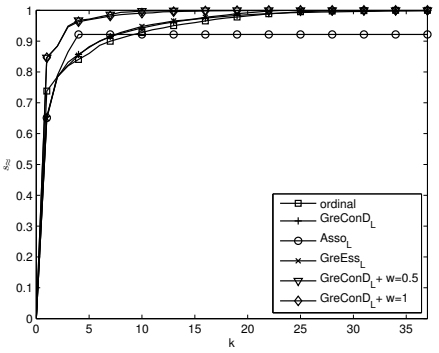
Set 1



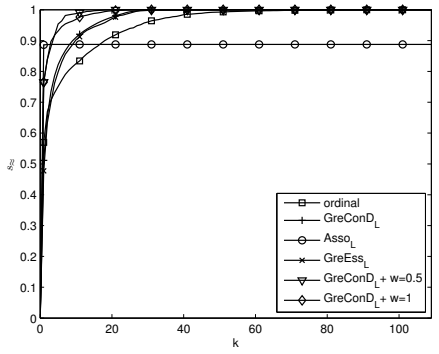
Set 2



Set 3

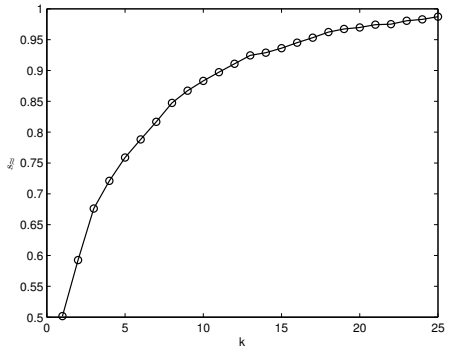


Set 4

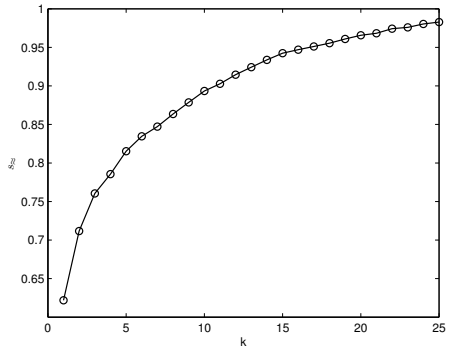


Set 5

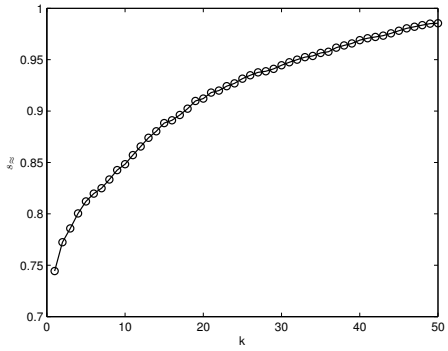
Figure 6.12: Coverage $s_{\approx}$ by the k factors



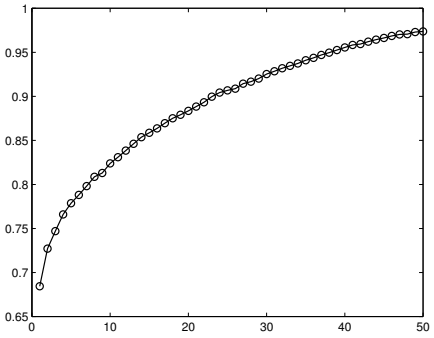
Set 1



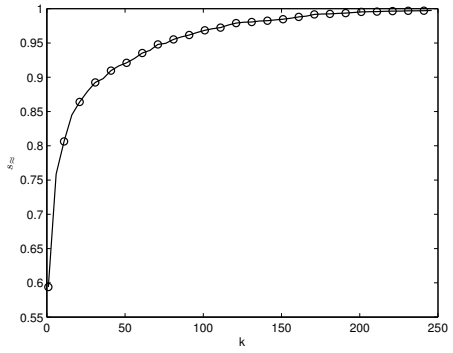
Set 2



Set 3



Set 4



Set 5

Figure 6.13: Coverage if J by factors of k rows.

6.3.3 Role of τ in Asso$_L$ algorithm

In presence of several degrees in L, one may observe a new phenomenon. It is known that for Boolean data the selection of the threshold τ significantly influences the performance of Asso [46]. An intuitive explanation is that with 0 and 1 as the only degrees, the decision based on τ whether to round off the confidence value to 0 or 1 is significant. We observed that in the setting with several degrees, the choice of τ becomes less significant as the number of degrees increases. This is a good feature for a user because the value of τ needs to be selected by the user but there are no known principles, except for ad hod recommendations, how to make such a choice.

dataset	size	$\|L\|$	k	distribution on L in A and B
Set 1	150×150	3	10	$[\frac{1}{3}\ \frac{1}{3}\ \frac{1}{3}]$
Set 2	150×100	5	10	$[\frac{1}{8}\ \frac{1}{8}\ \frac{1}{4}\ \frac{1}{4}\ \frac{1}{4}]$
Set 3	100×150	11	10	$[\frac{1}{16}\ \frac{1}{16}\ \frac{1}{16}\ \frac{1}{16}\ \frac{1}{16}\ \frac{1}{16}\ \frac{1}{8}\ \frac{1}{8}\ \frac{1}{8}\ \frac{1}{8}\ \frac{1}{8}]$
Set 4	100×100	21	10	$\frac{1}{21}$ for all
Set 5	100×100	101	10	$\frac{1}{101}$ for all

Table 6.12: Synthetic data.

Table 6.12 describes characteristics of the synthetic datasets which we used in the experiments. Table 6.13 presents the values of coverage $s_{\approx}$ corresponding to the first factor and to all the factors obtained. The values are observed for different values of τ. As one can see, as the size of L increases, the coverage values for different values of τ tend to be the same. The same tendency is seen in Table 6.14 with the stronger measure, $s_{=}$, replacing $s_{\approx}$. Note that the low values in this table, particularly for scales L with a larger number of degrees, indicating a low number of entries for which the input matrix and the matrix reconstructed from the factors have equal values, are due to the aim of Asso$_L$ to generate approximate rather than exact decompositions.

For all of the datasets we obtain best coverage for τ between 0.85 and 0.95. For datasets with smaller size of L we obtain different coverage for $\tau = 0.85$ and $\tau = 0.95$. In datasets Set 4 and Set 5 this difference is small. See Table 6.13. The entries depict mean coverage for first factor/coverage for all factors.

dataset	$\tau = 0.85$	$\tau = 0.9$	$\tau = 0.95$
Set 1	0.85/0.87	0.87/0.88	0.83/0.86
Set 2	0.87/0.87	0.86/0.90	0.85/0.89
Set 3	0.87/0.87	0.88/0.88	0.87/0.87
Set 4	0.88/0.88	0.88/0.88	0.88/0.88
Set 5	0.87/0.88	0.87/0.88	0.87/0.88

Table 6.13: Coverage $s_{\approx}$ by the first factor/by all factors obtained for different values of τ.

dataset	$\tau = 0.85$	$\tau = 0.9$	$\tau = 0.95$
Set 1	0.68/0.73	0.63/0.65	0.46/0.50
Set 2	0.29/0.33	0.35/0.37	0.26/0.28
Set 3	0.12/0.17	0.12/0.19	0.14/0.19
Set 4	0.04/0.06	0.04/0.07	0.04/0.06
Set 5	0.006/0.01	0.006/0.01	0.006/0.01

Table 6.14: Coverage $s_{=}$ by the first factor/by all factors obtained by $\textsc{Asso}_L$ for different values of τ.

Chapter 7

Conclusion

In this thesis we generalized the Boolean matrix decomposition problem (BMF), took into account matrices over scales which represent ordinal data. We proposed answers to natural question: "How well a set of factors explains 'the data?" Moreover, we present a problem of explaining data by factors obtained from reduced data—data having same attributes but the smaller number of objects. We propose heuristic to deal with problem of selection from a possibly large dataset a smaller one such that the factors of the reduced dataset explain the large dataset well. The experimental results reveal that the heuristic returns very good, nearly optimal solutions.

The main part of this thesis presents existing and new algorithms for decomposition of matrices with ordinal attributes. We introduce three new algorithms, namely GreEss$_L$, Asso$_L$ and GreConD$_L$+, all based on more or less known BMF algorithms. We supported the correctness of these algorithms by theoretical results regarding the geometry of decompositions and by experimental evaluation presented in this thesis. It turns out that the methods yield reasonable and, in a sense, robust factors and that the results of the methods are easy to understand. We also shown that methods suited for ordinal matrices returns better results, than BMF methods on scaled data.

Decomposition of matrices over some scale is still not well understood problem. There is a lot of unresolved issues including for example: the choice of the scale of degrees, the operation $\otimes$ or a problem we addressed in Chapter 6—Asso$_L$ returns rectangles with values "around the middle" in L.

Abundantly discussed topic in data mining community is, in case of BMF, noise in Boolean data. This issue should be investigated in general case as well.

Shrnutí v českém jazyce

Rozklady Booleovských matic (BMF), rozklady matic, které obsahují pouze nuly a jedničky, také známé jako faktorizace Booleovských dat, se čím dál tím více těší pozornosti dataminingové komunity. Cílem BMF je hledání, v datech skrytých důležitých informací – faktorů – pomocí nichž lze vysvětlit či popsat originální data. Postupem času vznikla celá řada metod pro BMF. Cílem této práce je prozkoumat rozšíření těchto maticových rozkladů pro data, která nemají jen binární charakter, ale jejichž vstupy jsou z uspořádané škály. Takováto generalizace sebou přináší několik netriviálních problémů, které jsou rovněž diskutovány v této práci.

První část práce je věnována popisu problému rozkladů matic s odrinálními daty, na který můžeme nahlížet jako na problém pokrývání. Jsou zde stručně popsány matematické základy, které při řešení využíváme. Jedním z nich je Fuzzy logika, obzvláště pak kalkulus na reziduovaných svazech. Dále pak Formální konceptuální analýza – zde s výhodou využíváme faktu, že faktory vybrané z množiny formálních konceptů tvoří optimální řešení.

Navíc v první části práce prezentujeme nové teoretické výsledky, které pak využijeme při návrhu nových algoritmů. Především využíváme toho, že v binárním případě se ukázalo, že ne všechna políčka jsou rovnocenná. K tomu abychom pokryli celá vstupní data stačí pokrýt jen některá políčka. Tyto prvky nazýváme esenciální a jsou definovány přes minimální intervaly v konceptuálním svazu. Ukazujeme, že i v obecném případě lze nalézt ekvivalentní pojem. Generalizace není úplně přímočará a přináší několik výzev. Například každému esenciálnímu prvku může odpovídat více intervalů v kontrastu s binárním případem, kde interval je pouze jeden.

V druhé části práce představujeme již existující metody pro dekompozici matic s ordinálnímu daty. Konkrétně GreCon$_L$, GreConD$_L$ a Nonnegative matrix factorisation (NMF). Navíc demonstrujeme možnosti využití existujících BMF metod na data, která získáme ordinálním škálováním. Dále

představíme tři nové algoritmy, jejichž myšlenka pochází z BMF algoritmů.

Poslední část je věnována experimentální analýze a srovnání představených algoritmů. Zaměřujeme se především na interpretovatelnost faktorů, získaných z jednotlivých metod, počty faktorů a kvalitu pokrytí – jak velká část dat je vysvětlena získanými faktory. Experimenty provádíme na syntetických a reálných datech.

Bibliography

[1] L. B. Alexandrov, S. Nik-Zainal, D. C. Wedge, P. J. Campbell, M. R. Stratton, "Deciphering Signatures of Mutational Processes Operative in Human Cancer", *Cell Reports* 3 (1) (2013), 246–259.

[2] H. Andrews, C. Patterson, "Singular Value Decomposition (SVD) Image Coding", *IEEE Transactions on Communications* 24 (4) (2003), 425–432.

[3] R. Belohlavek, *Fuzzy Relational Systems: Foundations and Principles*, Kluwer, Academic/Plenum Publishers, New York, 2002.

[4] R. Belohlavek, "Concept Lattices and Order in Fuzzy Logic", *Annals of Pure and Applied Logic* 128 (1–3) (2004), 277–298.

[5] R. Belohlavek, "Optimal Decompositions of Matrices with Entries from Residuated Lattices", *J. Logic and Computation* 22 (6) (2012), 1405–1425.

[6] R. Belohlavek, "Ordinally Equivalent Data: A Measurement-Theoretic Look at Formal Concept Analysis of Fuzzy Attributes", *Int. Journal of Approximate Reasoning* 54 (9) (2013), 1496–1506.

[7] R. Belohlavek, M. Krmelova, "Factor Analysis of Sports Data via Decomposition of Matrices with Grades", In: Szathmary L., Priss U. (Eds.): CLA 2012: Proceedings of the 9th International Conference on Concept Lattices and Their Applications, 2012, pp. 293–304 Fuengirola (Málaga), Spain, October 2012.

[8] R. Belohlavek, M. Krmelova, "Beyond Boolean Matrix Decompositions: Toward Factor Analysis and Dimensionality Reduction of Ordinal Data", ICDM 2013, pp. 961–966, 2013.

[9] R. Belohlavek, M. Krmelova, "Factor Analysis of Ordinal Data via Decomposition of Matrices with Grades", *Annals of Mathematics and Artiffcial Intelligence* 72 (1–2) (2014), 23–44.

[10] R. Belohlavek, J. Outrata, M. Trnecka, "Impact of Boolean Factorization as Preprocessing Methods for Classification of Boolean Data", In: Szathmary L., Priss U. (Eds.): CLA 2012: Proceedings of the 9th International Conference on Concept Lattices and Their Applications, 2012, pp. 305–316, Fuengirola (Málaga), Spain, October 2012.

[11] R. Belohlavek, J. Outrata, M. Trnecka, "Impact of Boolean Factorization as Preprocessing Methods for Classification of Boolean data", *Annals of Mathematics and Artificial Intelligence* 72(1-2)(2014), 3–22.

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

[13] R. Belohlavek, M. Trnecka, "A New Algorithm for Boolean Matrix Factorization which Admits Overcovering", To appear in Discrete Applied Mathematics.

[14] R. Belohlavek, M. Trneckova, "The Asso algorithm for graded attributes", Unpublished manuscript.

[15] R. Belohlavek, M. Trneckova, "Toward a geometry of decompositions of matrices with grades", Unpublished manuscript.

[16] R. Belohlavek, M. Trneckova, "A decomposition algorithm for matrices with grades that admits overcovering", Unpublished manuscript.

[17] R. Belohlavek, V. Sklenar, J. Zacpal, "Crisply Generated Fuzzy Concepts", In: B. Ganter and R. Godin (Eds.): ICFCA 2005, Lecture Notes in Artificial Intelligence 3403, pp. 268–283, Springer-Verlag, Berlin/Heidelberg, 2005.

[18] R. Belohlavek, V. Vychodil, "Factor Analysis of Incidence Data via Novel Decomposition of Matrices", *Lecture Notes in Artificial Intelligence* 5548(2009), 83–97.

[19] R. Belohlavek, V. Vychodil, "Discovery of Optimal Factors in Binary Data via A Novel Method of Matrix Decomposition", *J. Computer and System Sciences* 76(1)(2010), 3–20.

[20] R. Belohlavek, V. Vychodil, "Formal Concept Analysis and Linguistic Hedges", *Int. J. General Systems* 41(5)(2012), 503–532.

[21] M. W. Berry, M. Browne, A. N. Langville, V. P. Pauca, R. J. Plemmons, "Algorithms and Applications for Approximate Nonnegative Matrix Factorization", *Computational Statistics & Data Analysis* 52 1(2007), 155–173.

[22] M. Chu, F. Diele, R. Plemmons, R. Ragni, "Optimality, Computation, and Interpretations of Nonnegative Matrix Factorizations", Unpublished Report, (2004) available at `http://www.wfu.edu/~plemmons`.

[23] P. Comon, "Independent Component Analysis, A New Concept?", *Signal Precessing* 36 (1994), 287–314.

[24] P. Cunningham, "Dimension Reduction", University College Dublin, Technical Report UCD-CSI-2007-7, 2007.

[25] W. J. Dixon(ed.), "BMDP Statistical Software Manual", Berkeley, CA: University of California Press, 1992.

[26] L. Eldén, "Matrix Methods in Data Mining and Pattern Recognition", SIAM, 2007.

[27] K. Flaska, P. Cakirpaloglu, "Identification of the Multidimensional Model of Subjective Time Experience", *Int. Studies in Time Perspective*, Imprensa da Universidade de Coimbra (2013), 259–273.

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

[29] F. Geerts, B. Goethals, T. Mielikäinen, "Tiling Databases", Proc. Discovery Science 2004, pp. 278–289.

[30] J. S. Golan, *Semirings and their Applications*, Springer, 1999.

[31] G. Golub, C. Van Loan, *Matrix Computations*, Johns Hopkins University Press, 1996.

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

[33] P. Hájek, *Metamathematics of Fuzzy Logic*, Kluwer, 1998.

[34] M. Huchard, A. Napoli, H. M. Rouane, P. Valtchev, "A Proposal for Combining Formal Concept Analysis and Description Logics for Mining Relational Data", ICFCA 2007, pp. 51–65, 2007.

[35] S. Karaev, P. Miettinen, J. Vreeken, "Getting to Know the Unknown Unknowns: Destructive-Noise Resistant Boolean Matrix Factorization", Proc. 2015 SIAM International Conference on Data Mining (SDM '15), pp. 325–333, 2015.

[36] M. Krmelova, M. Trnecka, V. Kreinovich, B. Wu, "How to Distinguish True Dependence from Varying Independence?", *Journal of Intelligent Technologies and Applied Statistics* 6(4)(2013), 339–351.

[37] M. Krmelova, M. Trnecka, "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.

[38] D. Lee, H. Seung, "Learning the Parts of Objects by Non-Negative Matrix Factorization", *Nature* 401 (1999), 788–791.

[39] D. Lee, H. Seung, "Algorithms for Non-Negative Matrix Factorisation", *Advances in Neural Information Processing Systems* 13 (2001), 556–562.

[40] R. Liao, Y-L. Boscolo, L. M. Yang, C. S. Tran, V. P. Roychowdhury, "Network Component Analysis", *PNAS* 100 (2003), 15522–15527.

[41] C. Lucchese, S. Orlando, R. Perego, "Mining Top-K Patterns From Binary Datasets in Presence of Noise", In: SIAM DM 2010, pp. 165–176, 2010.

[42] C. Lucchese, S. Orlando, R. Perego, "A Unifying Framework for Mining Approximate Top-k Binary Patterns", *IEEE Transactions On Knowledge and Data Engineering* 26(12):2900–2913, 2014.

[43] P. Miettinen, "The Boolean Column and Column-Row Matrix Decompositions", *Data Mining and Knowledge Discovery* 17(2008), 39–56.

[44] P. Miettinen, "Sparse Boolean Matrix Factorizations", Proc. IEEE ICDM 2010, pp. 935–940, 2010.

[45] P. Miettinen, "On Finding Joint Subspace Boolean Matrix Factorizations", In: SDM, pp. 954–965, 2012.

[46] P. Miettinen, T. Mielikäinen, A. Gionis, G. Das, H. Mannila, "The Discrete Basis Problem", *IEEE TKDE* 20 (2008), 1348–62.

[47] P. Miettinen, J. Vreeken, "Model Order Selection for Boolean Matrix Factorization", ACM SIGKDD 2011, pp. 51–59, 2011.

[48] G. T. Miller, "The mAgical Number Seven, Plus or Minus Two", *Psychol. Rev.* 63 (1956), 81–97.

[49] D. S. Nau, G. Markowsky, M. A. Woodbury, D. B. Amos, "A Mathematical Analysis of Human Leukocyte Antigen Serology", *Math. Biosci* 40 (1978), 243–270.

[50] F. A. Nielsen, D. Balslev, L. K. Hansen, "Mining the Posterior Cingulate: Segregation Between Memory and Pain Components", *NeuroImage* 27 3 (2005), 520–522.

[51] F. A. Nielsen, *Clustering of Scientific Citations in Wikipedia* Wikimania (2008).

[52] P. Paatero, U. Tapper, "Positive Matrix Factorization: A Non-negative Factor Model with Optimal Utilization of Error", *Environmetrics*, 5 (1994), 111–126.

[53] K. Pearson, "On Lines and Planes of Closest Fit to Systems of Points in Space", *Philosophical Magazine*, 2 (1901), 559–572.

[54] D. A. Simovici, C. Djeraba, *Mathematical Tools for Data Mining*, Springer, 2008.

[55] C. Spearman, "General Intelligence", Objectively Determined and Measured, *American Journal of Psychology*, 15 (1901), 201–293.

[56] G. W. Stewart, "On the Early History of The Singular Value Decomposition", *SIAM Review*, 35 (1993) 551–566

[57] L. Stockmeyer, *The Set Basis Problem is NP-complete.* Tech. Rep. RC5431, IBM, Yorktown Heights, NY, USA, 1975.

[58] L. Taslaman, B. Nilsson, "A Framework for Regularized Non-Negative Matrix Factorization, With Application to The Analysis of Gene Expression Data", *PLoS One 7* 11 (2012).

[59] N. Tatti, T. Mielikäinen, A. Gionis, H. Mannila, "What is The Dimension of Your Binary Data?", Proc. IEEE ICDM 2006, pp. 603–612, 2006.

[60] M. Trnecka, M. Trneckova, "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, pp. 107–118, 2014.

[61] M. Trnecka, M. Trneckova, "Decomposition of Boolean Multi-Relational Data with Graded Relations", In: Proceedings of the 8th IEEE International Conference on Intelligent Systems (IEEE IS'16), pp. 221–226, 2016.

[62] Y. Xiang, R. Jin, D. Fuhry, F. F. Dragan, "Summarizing Transactional Databases with Overlapped Hyperrectangles", *Data Mining and Knowledge Discovery* 23 (2011), 215–251.

[63] L. A. Zadeh, "Probability Measures of Fuzzy Events", *J. Math. Anal. Appl.* 23 (1968), 421–427.

FACTOR ANALYSIS WITH ORDINAL ATTRIBUTES

Markéta Trnečková

Author Paper of Dissertation Thesis

Department of Computer Science
Faculty of Science
Palacký University Olomouc
2016

Uchazeč
Mgr. Markéta Trnečková
marketa.trneckova@gmail.com
www.marketa-trneckova.cz

Školitel
prof. RNDr. Radim Bělohlávek, DSc.

Místo a termín obhajoby

Oponenti

S disertační prací a posudky se bude možné seznámit na katedře informatiky PřF UP, 17. listopadu 12, 771 46 Olomouc.

Abstract – The problem of matrix decomposition, also known as matrix factorization problem, is widely investigated in data mining community. Especially Boolean case, where entries of matrices are 0s and 1s. In this work we explore the extension of matrix decomposition problem for ordinal data, i.e. data where attributes are values from ordered scales. The replacement of the two-element set of Boolean values and Boolean operations by a multiple-valued set of grades and multiple-valued operations introduced various non-trivial problems. We examine existing algorithms for ordinal data and propose three new algorithms for matrix decomposition problem. We demonstrate that the proposed algorithms deliver decompositions with informative and easy-to-understand factors by analysing real datasets. Moreover, we also compare algorithms presented on synthetic datasets.

Chapter 1

Introduction

1.1 Problem setting

Factor analysis and related techniques based on matrix decompositions are important methods of data analysis. In the past, considerable attention has been paid to the problem of Boolean matrix factorization (BMF) and its variants, because of its direct usefulness in data analysis and its role in understanding Boolean data.

The basic problem is to find for a given $n \times m$ Boolean matrix I, some $n \times k$ and $k \times m$ Boolean matrices A and B with a reasonably small k for which the Boolean product $A \circ B$ is (approximately) equal to I.

In this work, we are concerned with extending the problems and methods of BMF toward a more general case. Namely, instead of Boolean matrices whose entries are 0s and 1s, we consider matrices with entries taken from a partially ordered set L bounded by 0 and 1, such as for example the five-element scale $L = \{0, \frac{1}{4}, \frac{1}{2}, \frac{3}{4}, 1\}$. The entries of a Boolean matrix I represent presence ($I_{ij} = 1$) and absence ($I_{ij} = 0$) of attributes. In the more general case, the entries represent degrees to which attributes are present, i.e. degrees to which they apply to objects, with 0 and 1 representing full absence and full presence and the intermediate degrees, such as $\frac{3}{4}$, representing partial presence.

Several methods for real-valued matrices exist. The best known are for example singular value decomposition and principal component analysis. These methods are widely used but the produced results are often hard to interpret, because of a possible presence of negative coefficients. Another well-known method, non-negative matrix factorisation, deals with this issue, but interpretation of results is not quite straightforward either.

Papers [5, 18] extended the Boolean matrix factorization problem and the methods developed in [19] to ordinal data. This paper provides an overview of existing methods, presents three new algorithms inspired by the existing BMF algorithms and compares them. Particular parts (Chapters 2, 3, 4, 5 and 6) of this work are mainly based on the articles [7, 8, 9, 14, 15, 16]. The full list of my publications can be found at my personal webpages `http://www.marketa-trneckova.cz`.

This work consists of seven chapters. The first chapter contains a brief introduction to this work and also contains the list of my publications relevant to this work and a brief survey of related works. The second chapter defines the problem this work is dealing with and lists the used notation. In the third chapter we present first observations related to new theory

behind the presented algorithms. The fourth chapter contains a brief description of existing algorithms that will be present in the experimental part of this work (Chapter 6) together with the new algorithms presented in Chapter 5. Chapter 5, the main part of this paper, comprises description of three new algorithms, their definition and same theoretical insight behind them. Chapter 6, the experimental part of this work, consists of several experiments. Shows how all presented methods work and their results on a small illustrative example and provides us results of various experiments on both real and synthetics datasets. The work is closed by Chapter 7 containing a summary of the work.

1.2 Related work

This section summarizes the works directly related to the topics in this paper. The main part of this work is devoted to matrix decompositions—factorization of a matrix into a product of two or more matrices. Roots of this decompositions lays in factor analysis, which aims is to find new hidden variables (factors) in data. Factor analysis was initiated in 1904 by Charles Spearman [55], when he wanted to determine whether there are common factors of human intelligence. He tested how well people performed on various tasks relating to intelligence.

Since the literature on matrix decompositions is too numerous we present here only a little part of it. Perhaps the best known methods designed for real-valued matrices are singular value decomposition (SVD) [56], principal component analysis (PCA) [26, 31], independent component analysis (ICA) [23] and network component analysis (NCA) [40]. These methods usually decompose $n \times m$ real input matrix I into a product of two (in case of PCA, ICA, NCA, NMF) or three matrices (in SVD). The constraints of each method are different. For example for PCA, we want one of the resulted matrices to be orthogonal, in ICA we require all components of the resulting matrix independent. There exist many applications using these methods, for example image processing and compression [2] or data reduction [24]. When using these techniques, some issues appears—such as a difficulty to interpret negative coefficients. This problem is solved by the well-known non-negative matrix factorization (NMF) [38]. Even though NMF is conceptually very different from the methods that we propose, a comparison seems worth performing. Applications of NMF are numerous, let us mention several of them. Text mining—analysis of document-term matrix (constructed usually as weighted word frequency in a set of documents)—[50] analyse small subset of scientific abstracts from PubMed database, [51] clusters Wikipedia articles and scientific journals based on the citations. Another application is spectral analysis, for example classification of space objects and debris [21], or bioinformatics applications such as for example gene expression [58] and identify common patterns of mutations that occur in cancers [1].

The data mining community pays attention to Boolean matrix factorisation, which is the most related to this work. One of the first paper in this area is [49] in which NP-completeness of the basic decomposition problem is observed. The interest in BMF in data mining is due to Miettinen's works, especially [46] with the Asso algorithm whose extension for matrices with scales we propose. Another Miettinen's works related to BMF include Boolean CX and CUR decompositions (different kind of decomposition) [43], investigating sparcity in BMF [44], examining common factor of two and more matrices [45], selecting the number of factors using minimum description length (MDL) [47]. [29] is the first paper on "tiling" Boolean

data, which is closely related to BMF since it corresponds to the from-below factorizations that we examine for matrices with scales.

The utilization of formal concepts (fixpoints of Galois connections) of Boolean matrices as factors, two BMF algorithms—a GreCon and a GreConD algorithms—and other issues are examined in [19]. One of these issues is a transformation between the space of attributes and the space of factors. This is used in machine learning for classification of Boolean data [10, 11]. Another paper about BMF related to our work is [12], which proposed the GreEss algorithm based on essential elements, which we generalize in this work. Not yet published is [13], which includes the algorithm GreConD+, a modification of the algorithm GreConD which allows for overcover error. [62] studies summarization of Boolean data and proposes an algorithm utilizing MDL called PaNDa, which is the algorithm for mining top-k patterns in Boolean data in [41] (the problems are naturally reformulated as BMF problems). Modification of PaNDa algorithm with using several different cost functions called PaNDa+ algorithm was proposed in [42]. Another algorithm called Nassau utilizes the MDL principle for solving BMF problem (in different way that PaNDa) was presented in [35].

In this work we are interested in a more general case namely in factorising matrices with entries from an appropriate scale. Matrices over scales and other structures are examined in many papers, including those on matrices over semi-ring-like algebras [30] and binary fuzzy relations between finite universes, see e.g. [3, 32].

Directly related to this paper are also [5, 18], where the the role of formal concepts of matrices over scales is studied and a decomposition algorithms are proposed. [9] presents analyses of various sports datasets using this algorithm and studies further theoretical problems inspired by the analyses. For algorithms Asso_L and GreEss_L presented in this work we refer to [8, 14, 15] and for algorithm GreConD_L+ we refer to [16].

A theoretical basis of this work lays in formal concept analysis (of Boolean data) [28], ordered and combinatorial structures [54] and closure structures in the setting of fuzzy logic and structures over scales [3]. The scales with aggregation we utilize in our work have recently been investigated in the context of formal fuzzy logic [32, 33].

Methods of analysis of ordinal data also appear in the psychological literature but the tools employed are basically variations of classical factor analysis. That is, grades are represented by and treated like numbers which leads to loss of interpretability, similarly as in the case of Boolean data, see e.g. [59].

Possible extension of factor analysis is multi-relational factor analysis. In specific form was mentioned in [45] as joint subspace matrix factorization, where are two Boolean matrices and both share the same rows (or columns). Another paper related to this topics is [34], where is introduced the relational formal concept analysis, i.e. the formal concept analysis on multi-relational data. The multi-relational data are iteratively merged into one data table and than processed. The most relevant papers for this extension are [37, 60, 61], where was presented factorisation of multi-relational data. Also a heuristic algorithm was presented there.

Chapter 2

Preliminaries

2.1 Fuzzy logic

Let us consider a set L of truth values. We assume that this set is partially ordered (partial ordering is denoted by $\leq$), contains a least element 0 and a greatest element 1.

Let a and b truth degrees from L, then in L exists a truth value which is greater than both a and b. The least element that is greater or equal to both a and b is called *supremum* of a and b. Analogously, we can define *infimum* of a and b—the greatest element from L which is smaller or equal to both a and b. We define the *lower cone* of A by $\mathcal{L}(A) = \{a \in L | a \leq b \text{ for all } b \in A\}$ and the *upper cone* of A by $\mathcal{U}(A) = \{a \in L | b \leq a \text{ for all } b \in A\}$. If $\mathcal{L}(A)$ has a greatest element a, then a is called the *supremum* of A (denoted $\bigvee A$) and dually if $\mathcal{U}(A)$ has a least element a, then a is called the *infimum* of A (denoted $\bigwedge A$). In particular, we assume that the partial order $\leq$ makes L a complete lattice [32] (i.e., arbitrary infima $\bigwedge$ and suprema $\bigvee$ exist in L). This assumption is automatically satisfied if L is a finite chain (i.e. $a \leq b$ or $b \leq a$ for every $a, b \in L$), in which case $a \wedge b = \min(a, b)$ and $a \vee b = \max(a, b)$. We also need to define operation logical conjunction (denoted by $\otimes$). We assume that $\otimes$ is commutative, associative, has 1 as its neutral element ($a \otimes 1 = a = 1 \otimes a$), and distributes over arbitrary suprema, i.e. $a \otimes (\bigvee_{j \in J} b_j) = \bigvee_{j \in J} (a \otimes b_j)$. This leads to if a and b are truth degrees of propositions p_1 and p_2, then $a \otimes b$ is the truth degree of proposition "p_1 and p_2".

Importantly, $\otimes$ induces another operation, $\rightarrow$, called the *residuum of* $\otimes$, which plays the role of the truth function of implication and is defined by

$$a \rightarrow b = \max\{c \in L \,|\, a \otimes c \leq b\}. \tag{2.1}$$

Residuum, which may be looked at as a kind of division, satisfies an important technical condition called adjointness:

$$a \otimes b \leq c \text{ iff } a \leq b \rightarrow c,$$

which is also utilized below. This leads to algebraic structures called *residuated lattices*.

Definition 1. *A residuated lattice is an algebra* $\mathbf{L} = \langle L, \wedge, \vee, \otimes, \rightarrow, 0, 1 \rangle$ *where*

(i) $\langle L, \wedge, \vee, 0, 1 \rangle$ *is a lattice with a least element 0 and a greatest element 1,*

(ii) $\langle L, \otimes, 1 \rangle$ *is a comutative monoid i.e.* $\otimes$ *is associative, commutative, and the identity* $x \otimes 1 = x$ *holds,*

(iii) $\otimes$ *and* $\rightarrow$ *satisfy the adjointness property, i.e.*

$$x \leq y \rightarrow z \text{ iff } x \otimes y \leq z$$

holds for each $x, y, z \in L$ *(*$\leq$ *denotes the lattice ordering).*

A residuated lattice is called *complete* if $\langle L, \wedge, \vee, 0, 1\rangle$ is a complete lattice.

Many examples of scales are known in many-valued logic [32, 33], among them those where L is the real unit interval $[0, 1]$ or its finite equidistant subinterval, i.e. $L = \{0, \frac{1}{n}, \ldots, \frac{n-1}{n}, 1\}$, which are used in examples and experiments presented in the work.

As far as the choice of the operations on L is concerned, we mainly use Łukasiewicz in examples, because of some of its intuitive properties. For example, the implication $\rightarrow$ naturally corresponds to the natural distance in $[0, 1]$.

2.2 Decomposition problem and its two variants

Factor analysis is a method used to describe variability among observed, correlated variables in terms of a potentially smaller number of unobserved variables which are called factors. For example, it is possible that variations in several observed variables (such as performance of students) mainly reflect the variations in an unobserved variable (their intelligence).

Formally, the input data is represented by an $n \times m$ object–attribute matrix I and the "explanation" means a decomposition

$$I = A \circ B \tag{2.2}$$

(exact or approximate) of I into a product $A \circ B$ of an $n \times k$ object–factor matrix A and a $k \times m$ factor–attribute matrix B. What kind of matrices (real, Boolean, or other) and what kind of product $\circ$ are involved determines the semantics of the factor model.

Now we present two concrete variants of the decomposition problem. These two problems reflect two important views on BMF. The first one—the *discrete basis problem* (DBP) [46]—emphasizes the importance of the first k (presumably the most important) factors. The second one—the *approximate factorization problem* (AFP) [12]—emphasizes the need to account for (and thus to explain) a prescribed portion of data, which is specified by error ε.

Formally DBP is defined as follows: Given $n \times m$ matrix I and positive integer k, find $n \times k$ matrix A and $k \times m$ matrix B that minimize $||I - A \circ B||$.

AFP is defined as follows: Given $n \times m$ matrix I and prescribed error ε, find $n \times k$ matrix A and $k \times m$ matrix B with k as small as possible such that that minimize $||I - A \circ B|| \leq \varepsilon$.

Several other reasonable variants may be formulated but we restrict to these two because they reflect two basic views of the decomposition problem.

Our model (2.2) involves matrices containing degrees (or grades) of certain scales L and the product is the sup-$\otimes$ product, as described below. In particular, the matrix entry I_{ij} is a degree to which attribute j applies to object i, for example $I_{ij} = 0.5$. Similarly, A_{il} is the degree to which factor l applies to object i and B_{lj} is the degree to which attribute j is (one particular) manifestation of factor l. The case in which the scale L contains only two degrees, 0 and 1, called the Boolean case in what follows, corresponds to Boolean matrices and Boolean factor analysis [19] which is a special case of ours.

A verbal description of equation (2.2) reads: "Object i has attribute j if and only if there exists factor l such that i has l (or, l applies to i) and j is one of the particular manifestations of l." Such description is certainly appealing and well understandable.

In the Boolean case, in which $L = \{0, 1\}$, the verbal description leads to

$$(A \circ B)_{ij} = 1 \text{ iff there exists } l \in \{1, \ldots, k\} \text{ such that } A_{il} = 1 \text{ and } B_{lj} = 1,$$

which may equivalently be described by the well-known formula

$$(A \circ B)_{ij} = \max_{l=1}^{k} \min(A_{il}, B_{lj}) \tag{2.3}$$

for Boolean matrix composition.

With a general scale L, we approach the situation according to the principles of (mathematical) fuzzy logic (see Section 2.1) as follows. Let us have the formulas $\varphi(i, l)$ saying "object i has factor l" and $\psi(l, j)$ saying "attribute j is a manifestation of factor l", and consider A_{il} the truth degree of $\varphi(i, l)$ and B_{lj} the truth degree of $\psi(l, j)$, i.e.

$$||\varphi(i, l)|| = A_{il} \text{ and } ||\psi(l, j)|| = B_{lj}. \tag{2.4}$$

Now, according to fuzzy logic, the truth degree of formula $\varphi(i, l) \,\&\, \psi(l, j)$ which says "object i has factor l and attribute j is a manifestation of factor l" is computed by

$$||\varphi(i, l) \,\&\, \psi(l, j)|| = ||\varphi(i, l)|| \otimes ||\psi(l, j)||$$

where $\otimes : L \times L \to L$ is a truth function of many-valued conjunction &, and hence the truth degree of $(\exists l)(\varphi(i, l) \& \psi(l, j)$ which says "there exists factor l such that object i has l and attribute j is a manifestation of l", i.e. the proposition is computed by

$$||(\exists)(\varphi(i, l) \,\&\, \psi(l, j))|| = \bigvee_{l=1}^{k} ||\varphi(i, l)|| \otimes ||\psi(l, j)||, \tag{2.5}$$

where $\bigvee$ denotes the supremum. Given into account (2.4), we see that a generalization of (2.3) to the case of possibly intermediate degrees is given by

$$(A \circ B)_{ij} = \bigvee_{l=1}^{k} A_{il} \otimes B_{lj}. \tag{2.6}$$

Therefore, with $\circ$ given by (2.6), the factor model (2.2) retains its meaning even in the case when intermediate degrees are allowed.

2.3 Formal concept analysis

From the description in Section 2.2, it is clear that for any decomposition (2.2), the lth factor ($l \in \{1, \ldots, k\}$) is represented by two parts: the lth column $A_{\_l}$ of A and the lth row $B_{l\_}$ of B. As shown in [5], optimal factors for a decomposition of I (see below) are provided by formal concepts associated to I. In detail, let $X = \{1, \ldots, n\}$ (objects) and $Y = \{1, \ldots, m\}$ (attributes). Recall that a formal concept (*formal fuzzy concept*) of I is any pair $\langle C, D \rangle$ of L-sets (fuzzy sets) $C : \{1, \ldots, n\} \to L$ of objects and $D : \{1, \ldots, m\} \to L$ of attributes,

see [4], that satisfies $C^{\uparrow} = D$ and $D^{\downarrow} = C$ where $^{\uparrow}\colon L^X \to L^Y$ and $^{\downarrow}\colon L^Y \to L^X$ are the concept-forming operators defined by

$$C^{\uparrow}(j) = \bigwedge_{i \in X}(C(i) \to I_{ij}) \text{ and } D^{\downarrow}(i) = \bigwedge_{j \in Y}(D(j) \to I_{ij}).$$

The set of all formal concepts of I is denoted by $\mathcal{B}(X, Y, I)$ or just $\mathcal{B}(I)$. The set $\mathcal{B}(I) = \{\langle C, D\rangle \mid C^{\uparrow} = D, D^{\downarrow} = C\}$ equipped with a partial order $\leq$, defined by $\langle C_1, D_1\rangle \leq \langle C_2, D_2\rangle$ iff $C_1 \leq C_2$ (iff $D_2 \leq D_1$), forms a complete lattice, called the *concept lattice* of I. The fuzzy set C is called *extent* and the fuzzy set D is called *intent*. $C(i) \in L$ is interpreted as the degree to which factor l applies to object i and $D(j) \in L$ is the degree to which attribute j is a manifestation of l.

Optimality of using formal concepts as factors means the following. Let for a set $\mathcal{F} = \{\langle C_1, D_1\rangle, \ldots, \langle C_k, D_k\rangle\} \subseteq \mathcal{B}(I)$ of formal concepts denote by $A_{\mathcal{F}}$ and $B_{\mathcal{F}}$ the matrices defined by

$$(A_{\mathcal{F}})_{il} = (C_l)(i) \quad \text{and} \quad (B_{\mathcal{F}})_{lj} = (D_l)(j). \tag{2.7}$$

Then, whenever $I = A \circ B$ for $n \times k$ and $k \times m$ matrices A and B, there exists a set $\mathcal{F} \subseteq \mathcal{B}(I)$, $|\mathcal{F}| \leq k$ such that $I = A_{\mathcal{F}} \circ B_{\mathcal{F}}$, i.e. the optimal decompositions are attained by formal concepts as factors.

By $\text{rank}_{\mathbf{L}}(I)$ we denote the smallest k for which the above decomposition of I exists and call it the *(L-)rank* of I.

For two matrices $J_1, J_2 \in L^{n \times m}$ we put $J_1 \leq J_2$ iff $(J_1)_{ij} \leq (J_2)_{ij}$ for every i, j in which case we say that J_1 is *contained* in J_2. $J \in L^{n \times m}$ is called a *rectangle* if $J = C \circ D$ for some column $C \in L^{n \times 1}$ and row $D \in L^{1 \times m}$. Note that in the Boolean case, rectangles are just tiles in terms of [29], i.e. rectangular areas filled with 1s. In general case, the C and D for which $J = C \circ D$ are not unique. We say that a rectangle J *covers* $\langle i, j\rangle$ in I if $J_{ij} = I_{ij}$.

2.4 Errors in decomposition

When we desire exact decomposition, using formal concept as factors is beneficial, but it has a limitation—it never commit overcovering—when approximate factorization is needed. For factor model 2.2, we are talking about *uncovering* when $I_{ij} > (A \circ B)_{ij}$ and *overcovering* when $I_{ij} < (A \circ B)_{ij}$.

The error function E (distance) between I and approximate decomposition $(A \circ B)$ is sum of two components—E_u and E_o denoting uncover error and overcover error respectively, i.e $E = E_u + E_o$. Uncover and overcover errors may be defined as follows

$$E_u = \sum_i \sum_j 1 - (I_{ij} \to (A \circ B)_{ij}),\ E_o = \sum_i \sum_j 1 - ((A \circ B)_{ij} \to I_{ij}).$$

These two components are not symmetrical. While E_u can only decrease by adding more factors, E_o may only increase. This fact was presented in boolean case in [12].

Observation 1. *Let $A' \in L^{n \times (k+1)}$ and $B' \in L^{(k+1) \times m}$ result by adding a single column and row, respectively. Then $E_u(I, A' \circ B') \leq E_u(I, A \circ B)$ and $E_o(I, A' \circ B') \geq E_o(I, A \circ B)$.*

Chapter 3

First observations

This chapter provides first observations that lead to deeper theoretical insight to below presented algorithms. Results presented here are based on [7, 8, 9].

3.1 Variants of decomposition problem

In the previous chapter we describe two variants of decomposition problem, namely the discrete basis problem (DBP) and the approximate factorisation problem (AFP). In order to define generalization of the DBP a AFP problems for Boolean matrices to general problems over some scale L, we need to define closeness of matrices over L.

The first possible approach is to take as closeness of two matrices $I, J \in L^{n\times m}$ function

$$s_{=}(I,J) = \frac{\sum_{i,j=1}^{n,m} eq(I_{ij}, J_{ij})}{n \cdot m}.$$

Function $eq(a, b)$ here returns 1 if a is equal to b and 0 otherwise. In a sense, this is a pessimistic approach because it ignores the case where I_{ij} is close to but different from J_{ij}.

Let $s_L : L \times L \to [0, 1]$ be an appropriate function measuring closeness of degrees in L. For matrices $I, J \in L^{n\times m}$, put

$$s_{\approx}(I,J) = \frac{\sum_{i,j=1}^{n,m} s_L(I_{ij}, J_{ij})}{n \cdot m}, \tag{3.1}$$

i.e. $s_{\approx}(I, J) \in [0, 1]$ is the normalized sum over all matrix entries of the closeness of the corresponding entries in I and J. In general, we require $s_L(a, b) = 1$ if and only if $a = b$, and $s_L(0, 1) = s_L(1, 0) = 0$, in which case $s_{\approx}(I, J) = 1$ if and only if $I = J$. We furthermore require that $a \leq b \leq c$ implies $s_L(a, c) \leq s_L(b, c)$. For the important case of L being a subchain of $[0, 1]$, s_L may be defined by $s_L(a, b) = a \leftrightarrow b$, where $a \leftrightarrow b = \min(a \to b, b \to a)$ is the so-called *biresiduum* (many-valued equivalence from a logical point of view) of a and b (note that $\to$ is the residuum (2.1) of $\otimes$).

In terms of above presented closeness, we now present generalisation of the two above presented problems of decomposition over scale L:

– *DBP(L)*: Given $I \in L^{n\times m}$ and a positive integer k, find $A \in L^{n\times k}$ and $B \in L^{k\times m}$ that maximize $s(I, A \circ B)$.

– *AFP(L)*: Given I and prescribed error $\varepsilon \in [0,1]$, find $A \in L^{n\times k}$ and $B \in L^{k\times m}$ with k as small as possible such that $s(I, A \circ B) \geq \varepsilon$.

As $s(I, A \circ B)$, we can take function $s_{\approx}$ or $s_{=}$.

In view of the provable difficulty of the AFP and DBP in the Boolean case [19, 46] and the remarks above, the following theorem is not surprising:

Theorem 1. *DBP(L) and AFP(L) are NP-hard optimization problems.*

3.2 Decomposition problem as a covering problem

In Section 2.3, we present notation in formal concept analysis and present a definition of rectangles in I. The following lemma, which is easy to see, extends the observation in [5] and shows that an exact decomposition of I is equivalent to a coverage of entries in I by rectangles contained in I.

Lemma 1. *The following conditions are equivalent for any $I \in L^{n\times m}$:*

(a) *$I = A \circ B$ for some $A \in L^{n\times k}$ and $B \in L^{k\times m}$.*

(b) *There exist rectangles $J_1, \ldots, J_k \in L^{n\times m}$ such that $I = J_1 \vee \cdots \vee J_k$, i.e. $I_{ij} = \max_{l=1}^{k} (J_l)_{ij}$.*

(c) *There exist rectangles $J_1, \ldots, J_k \in L^{n\times m}$ contained in I such that every $\langle i, j\rangle$ in I is covered by some J_l.*

In particular, for the matrices A and B in (a), one may take the product of the lth column of A and the lth row of B to be the rectangle J_l in (b).

Importantly, Lemma 1 allows us to consider the problem of decomposition of I as a certain coverage problem, namely the problem of covering the entries in I by rectangles contained in I.

The following theorem shows that formal concepts of I are optimal factors for approximate decompositions of I that provide a *from-below approximation* of I, i.e. $A \circ B \leq I$ (note that these include exact decompositions $I = A \circ B$).

Theorem 2. *Let for $I \in L^{n\times m}$ there exist $A \in L^{n\times k}$ and $B \in L^{k\times m}$ such that $A \circ B \leq I$. Then there exists a set $\mathcal{F} \subseteq \mathcal{B}(I)$ of formal concepts of I with $|\mathcal{F}| \leq k$ such that for the $n \times |\mathcal{F}|$ and $|\mathcal{F}| \times m$ matrices $A_{\mathcal{F}}$ and $B_{\mathcal{F}}$ over L we have*

$$s(I, A_{\mathcal{F}} \circ B_{\mathcal{F}}) \geq s(I, A \circ B).$$

3.3 Role of entries in matrix

We now examine in detail the coverage problem by rectangles, to which the decomposition problem may be transformed. An inspection of the concept lattice $\mathcal{B}(I)$ reveals an interesting fact—a possibility to differentiate the role of matrix entries for decompositions. In,

particular, we identify a so-called *essential part* of I, a minimal set of entries whose coverage guarantees an exact decomposition of I. We show later that the number of such entries is significantly smaller than the number of all entries. Most importantly, the essential part may be seen as the part to focus on when computing decompositions. This view is studied in detail in Section 5.1.1 and is utilized in the design of a decomposition algorithm in Section 5.1.2.

Note that the idea of differentiating the role of entries is inspired by [12], but the situation is considerably more involved in the setting of scales compared to the Boolean case.

The results presented in this section are based on [8].

Definition 2. *$J \leq I$ is called an* essential part *of I if J is minimal w.r.t. $\leq$ having the property that for every $\mathcal{F} \subseteq \mathcal{B}(I)$, $J \leq A_\mathcal{F} \circ B_\mathcal{F}$ then $I = A_\mathcal{F} \circ B_\mathcal{F}$.*

In other words, the coverage of an essential part J by formal concepts of I guarantees the coverage of all entries in I. It turns out that certain intervals in $\mathcal{B}(I)$ play a crucial role for our considerations. For $C \in L^{1\times n}$, $D \in L^{1\times m}$, put $\gamma(C) = \langle C^{\uparrow\downarrow}, C^{\uparrow}\rangle$ and $\mu(D) = \langle D^{\downarrow}, D^{\downarrow\uparrow}\rangle$, and denote by $\mathcal{I}_{C,D}$ the interval $\mathcal{I}_{C,D} = [\gamma(C), \mu(D)]$ in $\mathcal{B}(I)$, i.e. the set $[\gamma(C), \mu(D)] = \{\langle E, F\rangle \in \mathcal{B}(I) \mid \gamma(C) \leq \langle E, F\rangle \leq \mu(D)\}$.

In particular, $\gamma(\{\,{}^{a}\!/\!_{x}\}) = \gamma(x, a)$ and $\mu(\{\,{}^{b}\!/\!_{y}\}) = \mu(y, b)$ are the mappings from the basic theorem of $\mathcal{L}$-concept lattices [3].

In Section 5.1.1 we will show that all the rectangles corresponding to the formal concepts in $\mathcal{I}_{C,D}$ cover the rectangle $C^{\mathrm{T}} \circ D$.

Now, for a given matrix $I \in L^{n\times m}$, let $\mathbf{I}_{ij} = \{\mathcal{I}_{\{a/i\},\{b/j\}} \mid a, b \in L, a \otimes b = I_{ij}\}$ and put $\mathcal{I}_{ij} = \bigcup \mathbf{I}_{ij}$.

Note that the situation is much easier in the Boolean case. Namely, if $I_{ij} > 0$, then $\mathcal{I}_{ij}$ consists of a single interval in the Boolean case because the only a and b for which $a \otimes b = 1$ are $a = b = 1$. In case of general scales, there may be several pairs of a and b for which $I_{ij} = a \otimes b$, hence several intervals of which $\mathcal{I}_{ij}$ consists.

Later in Section 5.1.1 we will prove important theorem which shows that $\mathcal{I}_{ij}$ is just the set of all formal concepts of I that cover $\langle i, j\rangle$.

Denote now by $\mathcal{E}(I) \in L^{n\times m}$ the matrix over L defined by

$$(\mathcal{E}(I))_{ij} = \begin{cases} I_{ij} & \text{if } \mathcal{I}_{ij} \text{ is } \neq \emptyset \text{ and minimal w.r.t. } \subseteq, \\ 0 & \text{otherwise.} \end{cases}$$

In Section 5.1.1, we will show that $\mathcal{E}(I)$ is an essential part of matrix I.

3.4 Explanation of data by factors

If a set $\mathcal{F} \subseteq \mathcal{B}(X)$ of formal concepts of I satisfies $I = A_\mathcal{F} \circ B_\mathcal{F}$, we intuitively regard $\mathcal{F}$ as fully explaining the data represented by I and call $\mathcal{F}$ a set of *factor concepts*. In general, however, we are interested in $\mathcal{F}$ for which I is close to $A_\mathcal{F} \circ B_\mathcal{F}$, in particular if $\mathcal{F}$ is reasonably small. We can take into account above presented closeness $s_\approx$ and $s_=$ and say that $\mathcal{F}$ *explains* $100 \cdot s_=(I, A_\mathcal{F} \circ B_\mathcal{F})\%$ *of data* represented by I. Clearly, this means that $100 \cdot s_=(I, A_\mathcal{F} \circ B_\mathcal{F})\%$ of all the $n \times m$ entries have the same values in I and $A_\mathcal{F} \circ B_\mathcal{F}$. Or

analogously for $s_{\approx}$ we can say that entries from I and $A_{\mathcal{F}} \circ B_{\mathcal{F}}$ are in average $s_{\approx}(I, A_{\mathcal{F}} \circ B_{\mathcal{F}})$ close.

In the rest of the paper unless otherwise stated, we take $s_{\approx}$ as closeness function s.

3.4.1 General case

Let I and J be two matrices describing the sets X_1 and X_2 of objects by a common set Y of attributes. How can we answer the question of whether a set $\mathcal{F} \subseteq \mathcal{B}(I)$ of possibly good factors of I is a set of good factors of J? The concepts in $\mathcal{F}$ may not be directly used as concepts of J because for $\langle C, D\rangle \in \mathcal{F}$ we have $C \in L^{X_1}$ while we need $\in L^{X_2}$ for factors of J. A abundantly discussed the topicnatural option is to consider instead of $\mathcal{F}$ the set of concepts of J that are generated by the intents of the factors in $\mathcal{F}$, i.e. the set

$$\mathcal{F}^J = \{\langle D^{\downarrow_J}, D^{\downarrow_J\uparrow_J}\rangle \mid \langle C, D\rangle \in \mathcal{F}\}, \tag{3.2}$$

because the intents represent the meanings of concepts. One may then use $s_{=}(I, A_{\mathcal{F}_J} \circ B_{\mathcal{F}_J})$ or $s_{\approx}$ to asses how well the factors $\mathcal{F}$ of I explain the data represented by J.

Of a particular importance is the particular case when J results by adding rows to I (i.e. adding objects to those represented by I). Let us thus assume that $X_1 \subseteq X_2$ and that $I_{ij} = J_{ij}$ for $i \in X_1$ and $j \in Y$. We may proceed as above but the following observation presents a convenient simplification of the set $\mathcal{F}^J$.

For the above notation,

$$\mathcal{F}^J = \{\langle D^{\downarrow_J}, D\rangle \mid \langle C, D\rangle \in \mathcal{F}\}.$$

Therefore an intent of a factor of I is also an intent of a possible factor of a larger dataset J.

3.4.2 Selection of rows from dataset

An interesting problem is how to select from a possibly large dataset J a smaller I such that the factors of I explain well J. This problem was presented in [9], but no solution was provided here.

More precisely, J is the $n \times m$ matrix and $k < n$ the non negative integer smaller than n. We want to choose the $k \times m$ matrix I created by k selected rows from J, such that the factors $\mathcal{F} \subseteq \mathcal{B}(I)$ explain well J.

To solve the above presented problem, we use the essential part of matrices presented in Section 3.3. We benefit from the fact that the essential elements in matrix have some useful properties. One of them is that the essential part of matrix J is a minimal set of entries whose coverage guarantees an exact decomposition of J. Moreover essential part $\mathcal{E}(J)$ can be computed easily.

The procedure is following. For matrix J, we compute $\mathcal{E}(J)$ and choose k rows that contain the most of the essential elements. The idea behind the procedure is quite simple. More covered essential elements leads to a bigger coverage of input data.

Chapter 4

Previous algorithms

4.1 Boolean factorization of ordinally scaled attributes

Small overview of BMF methods can be found e.g. in [12]. A natural question is if this methods could be used for our purpose, i.e. to perform decomposition of an input matrix I with grades as follows. A positive answer may be given as follows. First, one transforms I by ordinal scaling to a Boolean matrix $I^\times$. Second, one performs Boolean factor analysis to $I^\times$ and interprets the obtained set $\mathcal{F}^\times$ of factors of $I^\times$ in an appropriate way, taking the scaling procedure into account. We presented this approach in paper [7], but experimental evaluation is missing.

Given an input matrix $I \in L^{n\times m}$, consider the matrix $I^\times \in \{0,1\}^{n\times(m\cdot|L|)}$ defined by

$$I^\times_{ij_a} = \begin{cases} 1 & \text{if } a \leq I_{ij}, \\ 0 & \text{otherwise,} \end{cases}$$

for $i = 1,\ldots,n$, $j = 1,\ldots,m$, $a \in L$ (we assume a fixed sorting of the elements in L so that the order of columns in $I^\times$ is fixed). That is, $I^\times$ is the Boolean matrix resulting from I by simple ordinal scaling. In a sense, each graded attribute j is replaced by a collection of Boolean attributes j_a ($a \in L$); j_a applies to object i if i has j to a degree at least a. The concept lattices and other structures associated to $I^\times$ and their relationships to those associated to I are studied in [17, 20] and are utilized in what follows.

Recalling that $\mathrm{rank}_{\mathbf{2}}(I^\times)$ and $\mathrm{rank}_{\mathbf{L}}(I)$ denote the Boolean rank of $I^\times$ and the $\mathbf{L}$-rank of I, respectively, i.e. the smallest numbers of factors using which $I^\times$ and I may be explained (factorized), we may formulate the following theorem.

Theorem 3. *For every I,* $\mathrm{rank}_{\mathbf{L}}(I) \leq \mathrm{rank}_{\mathbf{2}}(I^\times)$.

4.2 Previous algorithms for ordinal data

Pointing on limitations of Boolean factorization of ordinally scaled attributes, we claim that factor analyzing I directly using the methods suited for ordinal data has a significant advantage.

4.2.1 GreCon$_L$

As was mentioned above, this algorithm was presented in [19]. Moreover in [19] was proved that the optimal factors are obtained from the space of factors computed via FCA. The first algorithm (called Algorithm 1, later called GreCon) is based on an algorithm for set covering problem. The algorithm can be simply used for our fuzzy setting. A disadvantage of this approach is that such algorithm requires us to compute first the set $\mathcal{B}(I)$ of all formal fuzzy concepts and then select candidates for factor from $\mathcal{B}(I)$. Because $\mathcal{B}(I)$ can be exponentially large, this approach is time-consuming.

4.2.2 GreConD$_L$

The second algorithm for BMF presented in [19], (called Algorithm 2, later GreConD) was modified for decomposition ordinal data in [18]. This algorithm is designed to avoid computing the set $\mathcal{B}(I)$ of all formal concepts. Instead, it computes concepts on demand.

This algorithm generates factors by looking for "promising columns". It works due to fact that each formal concept $\langle C, D\rangle$, each intent D is an union of intents $\{^{D(j)}/_j\}^{\uparrow\downarrow}$. As a consequence, we may construct any formal concept by adding sequentially $\{^a/_j\}^{\uparrow\downarrow}$ to the empty set of attributes. This algorithm follows a greedy approach that selects $j \in Y$ and degree $a \in L$ which maximize the size of

$$D \oplus_a j = \{\langle k, l\rangle \in U | \{D \cup \{^a/_j\}\}^{\downarrow}(k) \otimes \{D \cup \{^a/_j\}\}^{\downarrow\uparrow}(l) \geq I_{k,l}\}.$$

4.2.3 Statistical methods

Statistical methods are widely used in many fields such as bioinformatics, medicine, chemistry and lot more. The non-negative matrix factorisation is method the most relevant to purpose of this work, so we omit the rest ones.

Non-negative matrix factorization

Interest in this methods started with the paper [38]. There exists hundreds of papers about NMF, and most of them cite [38] although this method was developed by Pentti Paatero [52] five years earlier.

NMF can be stated as follows: Given a non-negative matrix $A \in \mathbb{R}^{m\times n}$ and a positive integer $k < min(\{m, n\})$, find non-negative matrices $W \in \mathbb{R}^{m\times k}$ and $H \in \mathbb{R}^{k\times n}$ to minimize the function

$$f(W, H) = \frac{1}{2}||A - WH||_F^2.$$

The product WH is called *non-negative factorisation of* A. However, A is not usually equal to the product WH, i.e. WH is an approximate factorisation of rank at most k.

Various alternative minimization strategies have been proposed. In the standard NMF algorithm W and H are initialized with random non-negative values and than iteratively computes better approximation. Algorithms for NMF can be divided into three general classes: "Multiplicative update algorithms", "Gradient descent algorithms" and "Alternating least squares algorithms".

Chapter 5

New algorithms

In this chapter we present three algorithms for decomposition of matrices over scales. The first two algorithms are inspired by GreEss [12] and Asso [46], currently perhaps the best algorithms for the AFP and DBP, respectively. The third one is slightly modified GreConD$_L$, which is inspired by a modified binary version of GreConD called GreConD+ presented in [13].

5.1 GreEss$_L$

In [12] a new algorithm based on the properties of essential parts $\mathcal{E}(I)$ of Boolean matrices I was presented. The algorithm uses the fact that $\mathcal{E}(I)$ represents the entries whose cover by arbitrary factors guarantees an exact decomposition of I. Another useful property is that the number of 1s in $\mathcal{E}(I)$ tends to be significantly smaller than the number of 1s in I. $\mathcal{E}(I)$ may be simpler to cover than I, also note that we can compute $\mathcal{E}(I)$ efficiently. These features hold also in fuzzy setting. An algorithm based on this idea appeared in [8, 15].

5.1.1 Essential parts of matrices over scales

This section refers to Section 2.3 and Section 3.3, where are described first observations on the role of entries in input matrix I. We defined so-called essential part of I, dentoted $\mathcal{E}(I)$, whose cover by formal concepts of I guarantees the cover of all entries in I. We defined intervals $\mathcal{I}_{C,D}$ that play crucial role in this consideration.

Lemma 2. *If $\langle E, F\rangle \in \mathcal{I}_{C,D}$ then $C^{\mathrm{T}} \circ D \leq E^{\mathrm{T}} \circ F$.*

In particular, consider $C = \{{}^{a}/_{i}\}$ by which we denote the "singleton" vector with zero components except $C_i = a$, and $D = \{{}^{b}/_{j}\}$ with analogous meaning. Then every concept $\langle E, F\rangle$ in $\mathcal{I}_{C,D} = \mathcal{I}_{\{{}^{a}/_{i}\},\{{}^{b}/_{j}\}}$ covers the entry $\langle i, j\rangle$ in $C^{\mathrm{T}} \circ D$. This means that if $a \otimes b = I_{ij}$, then every concept in $\mathcal{I}_{\{{}^{a}/_{i}\},\{{}^{b}/_{j}\}}$ covers the entry $\langle i, j\rangle$ in I. However, the entry $\langle i, j\rangle$ in I is covered also by other concepts than those in $\mathcal{I}_{\{{}^{a}/_{i}\},\{{}^{b}/_{j}\}}$.

Lemma 3. *Let $\langle E, F\rangle \in \mathcal{B}(X, Y, I)$ and $a, b \in L$. Then $a \otimes b \leq E_i \otimes F_j$ if and only if for some c, d with $a \otimes b \leq c \otimes d$ we have $\langle E, F\rangle \in \mathcal{I}_{\{{}^{c}/_{i}\},\{{}^{d}/_{j}\}}$.*

For a given matrix $I \in L^{n\times m}$ let $\mathbf{I}_{ij} = \{\mathcal{I}_{\{a/i\},\{b/j\}} \mid a, b \in L, a \otimes b = I_{ij}\}$ and put

$$\mathcal{I}_{ij} = \bigcup \mathbf{I}_{ij}.$$

Theorem 4. *The rectangle corresponding to $\langle E, F\rangle \in \mathcal{B}(X, Y, I)$ covers $\langle i, j\rangle$ in I iff $\langle E, F\rangle \in \mathcal{I}_{ij}$.*

Theorem 5. *$\mathcal{E}(I)$ is an essential part of I.*

Theorem 6. *Let $\mathcal{G} \subseteq \mathcal{B}(\mathcal{E}(I))$ be a set of factor concepts of $\mathcal{E}(I)$, i.e. $\mathcal{E}(I) = A_{\mathcal{G}} \circ B_{\mathcal{G}}$. Then every set $\mathcal{F} \subseteq \mathcal{B}(I)$ containing for each $\langle C, D\rangle \in \mathcal{G}$ at least one concept from $\mathcal{I}_{C,D}$ is a set of factor concepts of I, i.e. $I = A_{\mathcal{F}} \circ B_{\mathcal{F}}$.*

5.1.2 GreEss$_L$ algorithm

The GreEss$_L$ algorithm, which we now present, is inspired by [12]. GreEss$_L$ is based on the results from Section 5.1.1 and some other facts mentioned below. It is primarily designed for AFP(L), but can also be used for DBP(L). The pseudocode depicted in Algorithm 1 describes computation of an exact decomposition of I but an easy modification makes it an algorithm for computation of ε-approximate decompositions (in line 3, stop when precision ε is reached).

In Algorithm 1 and Algorithm 2 the symbol $\emptyset$ denotes the empty set or the vector full of zeroes, depending on the context, $F \vee \{^a/_j\}$ denotes F with the component F_j updated to $F_j \vee a$, and $C \otimes D$ denotes the crossproduct of C and D, i.e. the rectangle for which $(C \otimes D)_{ij} = C_i \otimes D_j$. Moreover, U denotes the set of entries $\langle i, j\rangle$ not yet covered by the factors computed so far, and $cov(U, F, J)$ and $cov_I(U, D, \mathcal{E})$ denote the number of $\langle i, j\rangle \in U$ covered in I by the rectangle $F^{\downarrow_J} \otimes F^{\downarrow_J\uparrow_J}$ and $(D^{\downarrow_\mathcal{E}})^{\uparrow_I\downarrow_I} \otimes (D^{\downarrow_\mathcal{E}\uparrow_\mathcal{E}})^{\downarrow_I\uparrow_I}$, respectively. The fuzzy-set-like notation $\{^a/_j\} \in C^{\uparrow_I} \setminus F$ means $F_j < a \leq C_j^{\uparrow_I}$.

ComputeIntervals first computes $\mathcal{E}(I)$ (easy by definition) and then computes a set $\mathcal{G}$ of factors of $\mathcal{E}(I)$, each $\langle C, D\rangle \in \mathcal{G}$ representing the interval $\mathcal{I}_{C,D}$ in $\mathcal{B}(I)$ from which it is possible to obtain a decomposition of I according to Theorem 6. In fact we use the following improvement of Theorem 6 whose proof is easy and thus omitted: for $\mathcal{G}$ it suffices (rather than being a set of factor concepts of $\mathcal{E}(I)$) that the crossproducts $C^{\uparrow_I\downarrow_I} \otimes D^{\downarrow_I\uparrow_I}$ corresponding to $\langle C, D\rangle \in \mathcal{G}$ cover all entries in I (line 11). The formal concepts in $\mathcal{G}$ are computed in a greedy manner from $\mathcal{E}(I)$ by sequentially increasing in D (initially set to $\emptyset$) the most promising value a of the most promising component j (line 5–9), until such increase is impossible. The formal concept $\langle C, D\rangle$, obtained by taking closures w.r.t. $\mathcal{E}$ in line 7, is added to $\mathcal{G}$ (line 10). The entries covered by $C^{\uparrow_I\downarrow_I} \otimes D^{\downarrow_I\uparrow_I}$ are removed from U. The selection is repeated until U is empty.

With $\mathcal{G}$ obtained this way, GreEss$_L$ performs a greedy search for factors, i.e. formal concepts, in the intervals $\mathcal{I}_{C,D}$, $\langle C, D\rangle \in \mathcal{G}$, in line 3–21. For every $\mathcal{I}_{C,D}$ we select the formal concept in $\mathcal{I}_{C,D}$ with best coverage in line 6–11 in a manner similar to the one used in ComputeIntervals, i.e. extending the initially empty F by most promising attributes j and degrees a. The condition $J \leftarrow D^{\downarrow_I} \otimes C^{\uparrow_I}$ which functions as a restriction speeding up the computation, guarantees that we do not leave $\mathcal{I}_{C,D}$ in this search. The best found concept $\langle E', F'\rangle$ over all the intervals is then added to $\mathcal{F}$ in line 18. The interval $\mathcal{I}_{C',D'}$ in

which $\langle E', F' \rangle$ was found is removed from $\mathcal{G}$ in line 19 (hence is not searched in the remaining iterations) and U is updated accordingly.

Algorithm 1: GreEss$_L$

Input: matrix I with entries in scale L
Output: set $\mathcal{F}$ of factors for which $I = A_{\mathcal{F}} \circ B_{\mathcal{F}}$

```
1  G ← ComputeIntervals(I)
2  U ← {⟨i, j⟩ | I_ij > 0}; F ← ∅
3  while U is non-empty do
4      s ← 0
5      foreach ⟨C, D⟩ ∈ G do
6          J ← D^{↓I} ⊗ C^{↑I}; F ← ∅; s_⟨C,D⟩ ← 0
7          while exists {a/j} ∈ C^{↑I} \ F s.t. cov(U, F ∨ {a/j}, J) > s_⟨C,D⟩ do
8              select {a/j} maximizing cov(U, F ∨ {a/j}, J)
9              F ← (F ∨ {a/j})^{↓J↑J}
10             E ← (F ∨ {a/j})^{↓J}
11             s_⟨C,D⟩ ← cov(U, F, J)
12         end
13         if s_⟨C,D⟩ > s then
14             ⟨E', F'⟩ ← ⟨E, F⟩
15             ⟨C', D'⟩ ← ⟨C, D⟩
16             s ← s_⟨C,D⟩
17         end
18     end
19     add ⟨E', F'⟩ to F
20     remove ⟨C', D'⟩ from G
21     remove from U all ⟨i, j⟩ covered by E' ⊗ F' in I
22 end
23 return F
```

To proof correctness of this algorithm we provide its detailed description. GreEss$_L$ uses function ComputeIntervalswhich computes a set of concepts $\mathcal{G}$ by first computing the matrix $\mathcal{E}(I)$ and then computing the concepts of $\mathcal{B}(X, Y, \mathcal{E}(I))$ in a greedy manner inspired by [18] and adding them to $\mathcal{G}$. The difference between original algorithm from [18] is that we maximize size of $C^{\uparrow_I \downarrow_I} \otimes D^{\downarrow_I \uparrow_I}$ for concept $\langle C, D \rangle \in \mathcal{B}(X, Y, \mathcal{B}(I))$. This is possible since the factors for I are selected form the interval $\mathcal{I}_{C,D}$ (due to Lemma 2). GreEss$_L$ than picks at most one concept from every interval $\mathcal{I}_{C,D}$ for $\langle C, D \rangle \in \mathcal{G}$ until U is covered. It selects the intervals in a greedy manner similar to the one we described above.

5.2 Asso$_L$

Asso$_L$ is inspired by Asso algorithm [46], currently the best known algorithm for DBP. A preliminary version of the algorithm was presented in [8]. The final version of the algorithm will be presented in detail in an extended version of the paper [8] in [14].

Algorithm 2: COMPUTEINTERVALS

Input: matrix I with entries in scale L
Output: set $\mathcal{G} \subseteq \mathcal{B}(\mathcal{E}(I))$

$\mathcal{E} \leftarrow \mathcal{E}(I)$
$U \leftarrow \{\langle i, j\rangle | \mathcal{E}_{ij} > 0\}$
while U *is non-empty* **do**
  $D \leftarrow \emptyset$; $s \leftarrow 0$
  while *exists* $\{^a/_j\} \in D$ *s.t.* $cov_I(U, D \vee \{^a/_j\}, \mathcal{E}) > s$ **do**
    select $\{^a/_j\}$ maximizing $cov_I(U, D \vee \{^a/_j\}, \mathcal{E})$
    $D \leftarrow (D \vee \{^a/_j\})^{\downarrow\varepsilon\uparrow\varepsilon}$; $C \leftarrow (D \vee \{^a/_j\})^{\downarrow\varepsilon}$
    $s \leftarrow cov_I(U, D, \mathcal{E})$
  end
  add $\langle C, D\rangle$ to $\mathcal{G}$
  remove from U entries $\langle i, j\rangle$ covered by $C^{\uparrow_I\downarrow_I} \otimes D^{\downarrow_I\uparrow_I}$ in I
end
return $\mathcal{G}$

5.2.1 Association matrix

Recall that the ordinary ASSO is based on the idea of using the rows of the association matrix $\mathcal{A}$ of I as candidate basis vectors, i.e. rows of the $k \times m$ factor-attribute matrix B. $\mathcal{A}$ is an $m \times m$ Boolean matrix such that $\mathcal{A}_{pq} = 1$ if the confidence $c(p, q)$ of the association rule $\{p\} \Rightarrow \{q\}$ given by I exceeds a given threshold τ.

The confidence $c(p, q)$ may be understood as a conditional probability, namely that "an object has attribute q provided it has attribute p", given that objects as elementary events are equally probable. In presence of grades, we consider conditional probabilities $c_a(p, q)$ that "an object has attribute q provided it has attribute p to degree at least a". Loosely speaking, $c_a(p, q)$ is the confidence that the presence of p to degree at least a implies the presence of q. Unlike in the Boolean case, the collections of objects sharing some attributes to prescribed degrees are naturally conceived as fuzzy sets rather than ordinary sets. Thus, the collection $\{^a/p\}^{\downarrow}$ of all objects having attribute p at least to degree a is a fuzzy set of objects to which object $i = 1, \ldots, n$ belongs to degree

$$\{^a/p\}^{\downarrow}(i) = a \to I(i, p),$$

see [3]. Likewise, the collection of objects having p to degree at least a and having q is defined by

$$\{^a/p, {}^1/q\}^{\downarrow}(i) = (a \to I(i, p)) \wedge I(i, q).$$

These formulas may be obtained from considerations on Galois connections induced by graded relations [3] (as these are the mathematical counterparts of assignments of objects sharing a given collection of attributes) but may also be obtained on intuitive grounds.

In evaluating conditional probability that defines $c_a(p, q)$ we deal with fuzzy events (many-valued events) and probabilities of fuzzy events in the sense of Zadeh [63]. That is, the probability measure of fuzzy events involved in our situation is a function P assigning to every fuzzy set A of objects a number $P(A) \in [0, 1]$—the probability of the fuzzy event A. Assuming as in the classical case that the objects as elementary events are equally probable,

Zadeh's formulas for conditional probabilities $P(\cdot|\cdot)$ of fuzzy events yield that the confidence in question is defined by

$$\begin{aligned} c_a(p,q) &= P(\{{}^1/q\}^\downarrow|\{{}^a/p\}^\downarrow) = \frac{P(\{{}^a/p\}^\downarrow \cap \{{}^1/q\}^\downarrow)}{P(\{{}^a/p\}^\downarrow)} = \frac{P(\{{}^a/p, {}^1/q\}^\downarrow)}{P(\{{}^a/p\}^\downarrow)} \\ &= \frac{|\{{}^a/p, {}^1/q\}^\downarrow|}{|\{{}^a/p\}^\downarrow|}, \end{aligned}$$

where $|A|$ denotes the cardinality of a fuzzy set A. With $|A| = \sum_{i=1}^n A(i)$ we thus obtain

$$|\{{}^a/p, {}^1/q\}^\downarrow| = \sum_{i=1}^n \{{}^a/p, {}^1/q\}^\downarrow(i), \quad \text{and } |\{{}^a/p\}^\downarrow| = \sum_{i=1}^n \{{}^a/p\}^\downarrow(i).$$

Note also that in deriving the formula for $c_a(p,q)$ we used $\{{}^a/p\}^\downarrow \cap \{{}^1/q\}^\downarrow = (\{{}^a/p\} \cup \{{}^1/q\})^\downarrow = \{{}^a/p, {}^1/q\}^\downarrow$ which is a basic property of Galois connections [3]. The confidence is a number in $[0,1]$ which may be transformed to a truth value in L using a user-defined threshold $\tau \in [0,1]$. The reason is in principle the same as in the Boolean case, namely to obtain from the vectors of confidence values, $\langle \ldots, c_a(p,q), \ldots \rangle$, appropriate vectors of grades in L, i.e. the candidate basis vectors. However, the thresholding process is more involved compared to the Boolean case, and we propose to accomplish it by the rounding function round_τ defined for $r \in [0,1]$ by

$$\mathrm{round}_\tau(r) = \begin{cases} r_+ = \min\{a \in L \mid a \geq r\} \text{ if } r_+ \leftrightarrow r \geq \tau, \\ r_- = \max\{a \in L \mid a < r\} \text{ otherwise.} \end{cases}$$

Here, $r_+ \leftrightarrow r = \min(r_+ \to r, r \to r_+)$ is the many-valued logical equivalence mentioned above. One may observe that if $L = \{0,1\}$ we obtain the thresholding involved in the ordinary Asso.

This way, we may define for every attribute p and every suitable grade $a \in L - \{0\}$ a candidate basis vector, i.e. a row $\mathcal{A}_{(p,a),\_}$ of a prospective association matrix $\mathcal{A}$, by

$$\mathcal{A}_{(i,a),j} = \mathrm{round}_\tau(c_a(p,q)).$$

Picking now a set $K \subseteq L - \{0\}$ of suitable grades, we obtain an association matrix $\mathcal{A} \in L^{(m \cdot |K|) \times m}$. One may verify that if $L = \{0,1\}$ and $K = \{1\}$ then $\mathcal{A}$ is just the ordinary $m \times m$ association matrix defined in [46]. The presence of intermediate grades allows us to broaden the set of candidate basis vectors. Namely, in addition to the possible choice $K = \{1\}$, we may pick K containing more grades, e.g. $K = L - \{0\}$, and thus enlarge the search space for factorization.

5.2.2 Procedures Cover and Asso$_L$

The basic idea of the Asso algorithm may be described as follows. The algorithm iteratively computes k factors one by one, with the provision that it stops with less than k factors if the addition of any new factor would only worsen the error function, i.e. would decrease the value of s in our case. Let A and B denote the object-factor and factor-attribute matrices computed so far. The next factor, which is described by a new column and a new row to

be added to A and B, is computed as follows. For every candidate row of B, i.e. the row of the association matrix $\mathcal{A}$, one determines the best corresponding candidate column of A. "Best" means such that the value of a function COVER (see Equation 5.1) is maximized. The candidate row of B and column of A with the highest value of COVER are then added as a new factor to B and A.

The purpose of the function COVER is to yield a high value for factors whose addition is likely to lead to a good resulting matrices A and B, i.e. with high value of s. In the Boolean case, this means that we want a high number C of entries $\langle i, j\rangle$ for which $I_{ij} = 1$ and $(A \circ B)_{ij} = 1$, i.e. 1s in I that are "covered" by the factors, and a small number O of entries for which $I_{ij} = 0$ and $(A \circ B)_{ij} = 1$, i.e. are "overcovered" by the factors. This reasoning leads to the formula

$$w^+ \cdot C - w^- \cdot O$$

as the definition of COVER in the Boolean case. The weights reflect relative importance of C and O. In practice, one works with w^- larger than w^+ because "overcovering" cannot be undone by adding further factors. Hence, the presence of a single $\langle i, j\rangle$ with $(A \circ B)_{ij} = 1$ and $I_{ij} = 0$ represents a more serious harm than a presence of a single $\langle i, j\rangle$ with $(A \circ B)_{ij} = 0$ and $I_{ij} = 1$, because the latter discrepancy may be corrected by adding an appropriate factor in the next steps of the algorithm.

An appropriate form of the COVER function in the setting with general scales is more delicate. One reason is that the coverage of entry $\langle i, j\rangle$ of I is a matter of degree. We therefore need to account for a partial coverage and a partial overcoverage. For instance, if $I_{ij} = 0.5$ and $(A \circ B)_{ij} = 0.4$, then one may consider $\langle i, j\rangle$ almost covered and thus consider $I_{ij} \leftrightarrow (A \circ B)_{ij} = 0.5 \leftrightarrow 0.4 = 0.9$ as the degree to which $\langle i, j\rangle$ is covered. Likewise, if $I_{ij} = 0.5$ and $(A \circ B)_{ij} = 0.6$, then $\langle i, j\rangle$ is slightly overcovered and $\neg(I_{ij} \leftrightarrow (A \circ B)_{ij}) = \neg(0.5 \leftrightarrow 0.6) = 0.1$ may be thought of as a degree to which $\langle i, j\rangle$ is overcovered. Using a similar reasoning as in the Boolean case, one could obtain the value of COVER by adding the degrees corresponding to the first type of entries, multiply them with w^+ and subtract from this number the w^--multiple of the sum of the degrees corresponding to the second type of entries. This, however, would not yet be an appropriate approach. For consider a situation in which $I_{ij} = 0.5$, w^- is even five times larger than w^+, and the so far computed matrices A and B yield $(A \circ B)_{ij} = 0.3$. Suppose we now have two options. First, adding a factor resulting in A_1 and B_1 with $(A_1 \circ B_1)_{ij} = 0.4$; second, adding a factor resulting in A_2 and B_2 with $(A_2 \circ B_2)_{ij} = 0.52$. Intuitively, the second choice is preferable because the factor commits only a slight overcovering of $I_{ij} = 0.5$. However, the function COVER described above would lead to the selection of the first factor. Namely (for simplicity, we disregard entries other than $\langle i, j\rangle$), the first factor contributes by $w^+ \cdot (I_{ij} \leftrightarrow (A_1 \circ B_1)_{ij}) = w^+ \cdot 0.9$, while the second one contributes by $-w^- \cdot \neg(I_{ij} \leftrightarrow (A_2 \circ B_2)_{ij}) = -w^- \cdot 0.02$, i.e. even represents a decrease in value of COVER. The point is that the entries which are overcovered, i.e. $I_{ij} < (A \circ B)_{ij}$, need to be looked at as follows: They need to be penalized for overcovering by $w^- \cdot \neg(I_{ij} \leftrightarrow (A \circ B)_{ij})$ but at the same time rewarded for full covering by $w^+ \cdot 1$. This type of problem is degenerate in the Boolean case in which the reward can be ignored because it would pertain to all entries with $I_{ij} = 0$, would be equal for all such entries, and would hence have no influence on the choice of factors. This explains why the function COVER for the ordinary ASSO algorithm does not contain any rewarding term for the overcovered entries.

The above reasoning leads to the following definition of COVER. Let $\mathcal{F}$ denote a set of factors (with a fixed ordering of its elements), i.e. pairs $\langle C, D\rangle$ where $C \in L^{1\times n}$ and $D \in L^{1\times m}$, and let $A_\mathcal{F}$ and $B_\mathcal{F}$ be the matrices defined as in (2.7).

Then we put

$$\begin{aligned}\textsc{Cover}(A_\mathcal{F}, B_\mathcal{F}, I, w^+, w^-) = \\ +w^+ \cdot \sum\{I_{ij} \leftrightarrow (A_\mathcal{F} \circ B_\mathcal{F})_{ij} \mid I_{ij} \geq (A_\mathcal{F} \circ B_\mathcal{F})_{ij}\} \\ +w^+ \cdot |\{\langle i, j\rangle \mid I_{ij} < (A_\mathcal{F} \circ B_\mathcal{F})_{ij}\}| \\ -w^- \cdot \sum\{1 - (I_{ij} \leftrightarrow (A_\mathcal{F} \circ B_\mathcal{F})_{ij}) \mid I_{ij} < (A_\mathcal{F} \circ B_\mathcal{F})_{ij}\}.\end{aligned} \tag{5.1}$$

The above procedure for computing a set $\mathcal{F}$ of factors is described by Algorithm 3.

Algorithm 3: $\textsc{Asso}_L$

Input: matrix $I \in L^{n\times m}$, $k \geq 1$, w^+, w^-, τ, $K \subseteq L - \{0\}$
Output: set $\mathcal{F}$ of factors

```
1 compute association matrix 𝒜
2 ℱ ← ∅
3 for l = 1, ..., k do
4 |   select ⟨C, 𝒜_(i,a)_⟩ maximizing Cover(ℱ ∪ {⟨C, 𝒜_(i,a)_⟩}, I, w+, w−)
5 |   add ⟨C, 𝒜_(i,a)_⟩ to ℱ
6 end
7 return ℱ
```

Note that the selection in line 4 proceeds by finding for every row $\mathcal{A}_{(i,a)\_}$ of $\mathcal{A}$ the best C w.r.t COVER and then selecting the best found pair $\langle C, \mathcal{A}_{(i,a)\_}\rangle$. Due to the properties of COVER, the best C for a given $\mathcal{A}_{(i,a)\_}$ is found efficiently in a componentwise manner, i.e. by finding the best C_p for every $p = 1, \ldots, n$ (independently of the other C_qs).

5.3 GreConD$_L$+

GRECOND+ reflects two ideas. First, formal concepts of the factorized matrix form a crucial part of resulted factors. The second idea—inspired by 8M method [25] which is one of the oldest BMF algorithm—already constructed factor could be improved or eliminated depending on the another added factors.

GRECOND+ extends algorithm GRECONDand in each step returns a formal concept (factor) which covers the most still uncovered entries, i.e. which minimizes uncovered error. Such factor is then extended in a greedy manner by further columns and rows for which the gain of decreased uncovered error is larger than the loss due to overcover error, both formed by added columns/rows. After this phase, where a new factor is created, the set of obtained factors (created in previous iterations) is examined and some of them could be modified or even removed. This step is inspired by 8M method and leads to a decrease of overcover error.

GRECOND$_L$+ is generalization of GRECOND+ algorithm for data over some scale L. Results from this section are based on [16].

5.3.1 Algorithm GreConD$_L$+

Algorithm presented in this section is modification of previously presented algorithm GreConD$_L$ from [19], described in Section 4.2.2. Pseudocode of this algorithm is depicted below (see Algorithm 4).

Algorithm 4: GreConD+

Input: $n \times m$ matrix I, number w
Output: set $\mathcal{F}$ of factors

```
1  U ← {⟨i, j⟩ | I_ij ≠ 0}; F ← ∅
2  while U ≠ ∅ do
3  |  D ← ∅; V ← 0
4  |  while exists {a/j} ∉ D such that |D ⊕_a j| > V do
5  |  |  select {a/j} ∉ D that maximizes |D ⊕_a j|
6  |  |  D ← (D ∪ {a/j})^↓↑
7  |  |  V ← |D ⊕_a j|
8  |  end
9  |  C ← D^↓
10 |  ⟨E, F⟩ ← Expansion(⟨C, D⟩, w)
11 |  add ⟨C ∪ E, D ∪ F⟩ to F
12 |  for ⟨i, j⟩ ∈ U do
13 |  |  if I_ij ≤ (C ∪ E)_i ⊗ (D ∪ F)_j then
14 |  |  |  U ← U − ⟨i, j⟩
15 |  end
16 |  foreach factor ⟨A, B⟩ ∈ F do
17 |  |  if for each ⟨i, j⟩ with A_i ⊗ B_j > 0 there is ⟨G, H⟩ ∈ F − ⟨A, B⟩ with
   |  |  A_i ⊗ B_j ≤ G_i ⊗ H_j then
18 |  |  |  remove ⟨A, B⟩ from F
19 |  |  else
20 |  |  |  foreach j such that B_j > nucleus(B)_j do
21 |  |  |  |  if for each A_i ⊗ B_j there is ⟨G, H⟩ ∈ F − ⟨A, B⟩ with A_i ⊗ B_j ≤ G_i ⊗ H_j
   |  |  |  |  then
22 |  |  |  |  |  B_j ← nucleus(B)_j
23 |  |  |  |  end
24 |  |  |  end
25 |  |  end
26 |  end
27 end
28 return F
```

The main loop of the algorithm (lines 2–27) is executed until all the nonzero entries of I are covered by at least one factor in $\mathcal{F}$. Clearly, a different stopping criterion is possible—stopping after a prescribed number of factors is computed, which corresponds to the DBP problem, or after the overall error does not exceed ε, which corresponds to AFP. The code between lines 4 and 9 works like original GreConD$_L$ algorithm. In this part the formal concept which covers the maximal part of still uncovered entries, i.e. minimise uncover error, is selected. Resulted concept $\langle C, D\rangle$ is taken as a nucleus and then its expansion $\langle E, F\rangle$ is computed by Expansion algorithm (Algorithm 5). For simplicity Expansion described in Algorithm 5 is restricted to adding columns with positive *gain* (described later) until no

Algorithm 5: EXPANSION

Input: pair $\langle C, D\rangle$, number w
Output: expansion $\langle E, F\rangle$ of $\langle C, D\rangle$

```
1  E ← ∅; F ← ∅
2  repeat
3  |  select column j and a ∈ L such (D ∪ F)_j < a maximizing gain({a/j})
4  |  if gain({a/j}) > 0 then
5  |  |  add {a/j} to F
6  |  end
7  until F did not change
8  return ⟨E, F⟩
```

$\{^a/_j\}$ with positive *gain* exists. Extension for rows is straightforward. Thus extended factor $\langle C \cup E, D \cup F\rangle$ is added to $\mathcal{F}$ (line 11). The loop between lines 12 and 15 ensures that all matrix entries covered by this factor are removed from $\mathcal{U}$. Last loop (lines 16–26) goes through all the factors and factor is removed from $\mathcal{F}$ iff every non-zero entry is covered by other factors at least in the same degree. If this is not possible, one replace degree of column j in B by degree of j in $nucleus(B)$ for which all non-zero entries are covered by remaining factors (there exist factors covering all entries at least in the same degree as B).

Function *gain* is in the general case more complicated than the same function in BMF case. There are three cases. Denote

$$\begin{aligned} new_{ij} &= (A_{\mathcal{F}} \circ B_{\mathcal{F}})_{ij} \vee C_i \otimes (D \cup F \cup \{^a/_j\})_j, \\ old_{ij} &= (A_{\mathcal{F}} \circ B_{\mathcal{F}})_{ij} \vee C_i \otimes (D \cup F)_j. \end{aligned}$$

Then

(1) $new_{ij} \leq I_{ij}$, i.e. I_{ij} is still not covered, but coverage is increased by $[I_{ij} \leftrightarrow new_{ij}] - [I_{ij} \leftrightarrow old_{ij}]$, which is equal to $new_{ij} - old_{ij} = \neg(new_{ij} \rightarrow old_{ij})$.

(2) $old_{ij} < I_{ij} < new_{ij}$, we overcover I_{ij}. Value of *gain* needs to be increased by $I_{ij} - old_{ij} = \neg(I_{ij} \rightarrow old_{ij})$, but also decreased by weighted overcover error $w \cdot (new_{ij} - I_{ij})$.

(3) $I_{ij} \leq old_{ij} < new_{ij}$, overcover is increased, so *gain* needs to be also increased by $w \cdot (new_{ij} - old_{ij})$

Function *gain* for $\{^a/_j\}$ returns:

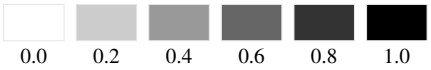
$$\begin{aligned} gain(\{^a/_j\}) &= \sum_{i=1,j=1}^{n,m} \{new_{ij} - old_{ij} \mid new_{ij} \leq I_{ij}\} \\ &+ \sum_{i=1,j=1}^{n,m} \{(I_{ij} - old_{ij}) - w \cdot (new_{ij} - I_{ij}) \mid old_{ij} < I_{ij} < new_{ij}\} \\ &- w \cdot \sum_{i=1,j=1}^{n,m} \{new_{ij} - old_{ij} \mid I_{ij} \leq old_{ij}\} \end{aligned}$$

Chapter 6

Experimental evaluation

6.1 Illustrative example

The data describes 5 most popular dog breeds and their 11 attributes[1]. We take as the complete residuated lattice six-element Łukasiewicz chain ($L = \{0, 0.2, 0.4, 0.6, 0.8, 1\}$) and represent the grades in L by shades of gray as follows:



The 5×11 object-attribute matrix I and its decomposition $I = A_{\mathcal{F}} \circ B_{\mathcal{F}}$ into the object-factor and factor-attribute matrices $A_{\mathcal{F}}$ and $B_{\mathcal{F}}$ are shown for various algorithms in subsections below.

6.1.1 Results for GreConD$_L$

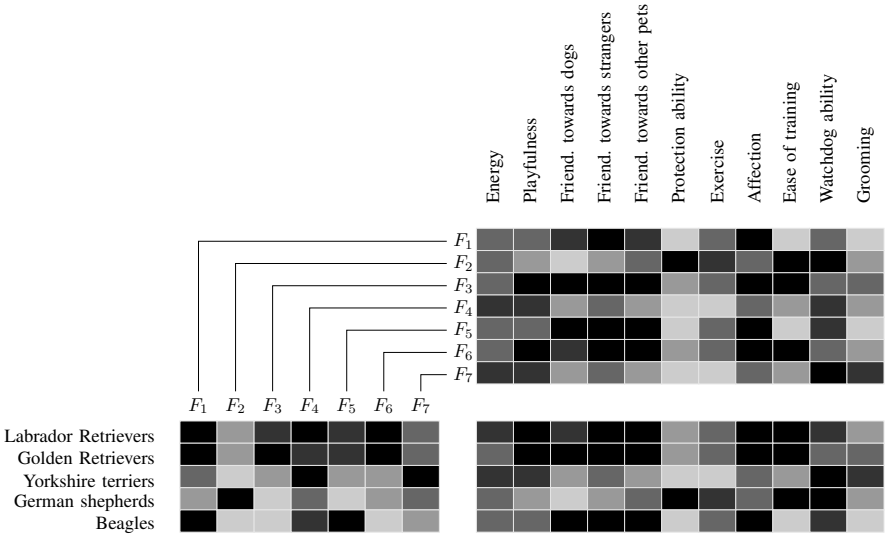


Figure 6.1: GreConD$_L$: Decomposition $I = A_{\mathcal{F}} \circ B_{\mathcal{F}}$. I, $A_{\mathcal{F}}$, and $B_{\mathcal{F}}$ are the bottom-right, bottom-left, and top matrix, respectively.

[1] `http://www.petfinder.com/`

In Figure 6.1 are shown seven factors obtained via GreConD$_L$. Factor F_1 is manifested by the three kinds of "Friendliness" and "Affection" and applies in particular to Labradors, Golden Retrievers and Beagels in the first column of $A_{\mathcal{F}}$) , and to some extent to Yorkshires. The factor may hence be termed *friendliness*. On the other hand, the three attributes with the highest degree in the row of F_2 plus a high degree of "Exercise" tell us that this factor is naturally interpreted as *guardian dog*. The corresponding column shows that F_2 applies to German shepherds and separates them clearly from the other breeds. Factor F_3 may be interpreted as *dogs suitable for kids*, because it is manifested by high "Friendliness", "Playfulness", "Affection", and "Ease of training", and applies to Golden Retrievers (in degree 1) and Labrador Retrievers (in degree 0.8).

Interestingly, F_1, F_2, and F_3 explain, by and large, the whole data and hence, the other factors may be neglected. Namely, denoting by $A_{\mathcal{F}_3}$ and $B_{\mathcal{F}_3}$ the 5×3 and 3×11 matrices (parts of $A_{\mathcal{F}}$ and $B_{\mathcal{F}}$), the degree $s(I, A_{\mathcal{F}_3} \circ B_{\mathcal{F}_3})$ of similarity of I to $A_{\mathcal{F}_3} \circ B_{\mathcal{F}_3}$, i.e. reconstructability of the original data I from the first three factors, equals 0.92.

6.1.2 Results for ordinal scaling

We transformed the input matrix $I \in L^{5\times 11}$ to a Boolean matrix $I^{\times} \in \{0,1\}^{5\times(11\cdot 6)} = \{0,1\}^{5\times 66}$ and computed a set $\mathcal{G}^{\times} \subseteq \mathcal{B}(I^{\times})$ of factors of $I^{\times}$ using the GreConD algorithm from [19]. We obtained 8 factors of $I^{\times}$, compared to the 7 factors of I obtained for by GreConD$_L$. The factors, $F_1, \ldots, F_8$, are depicted in a concise way in Figure. 6.2. As before, $A_{\mathcal{G}^{\times}}$ is the bottom-left matrix and its columns represent the factor extents, which are now ordinary sets of objects. To save space, the 8×66 Boolean matrix $B_{\mathcal{G}^{\times}}$ is represented by the top 8×11 matrix with grades as follows. For every attribute y, instead of the 6 columns $y_0, y_{0.2}, \ldots, y_1$ of $B_{\mathcal{G}^{\times}}$, the 8×11 matrix contains just one column which contains in row F_l the largest degree a for which y_a belongs to the intent of F_l. This way, the intent of F_l, an ordinary set of the scaled Boolean attributes y_a, is uniquely described because if y_b is in the intent and $c \leq b$, then y_c in in the intent as well. The corresponding percentage $100 \cdot s_{\approx}\%$ (which is the same as $100 \cdot s_{=}\%$ in the Boolean case) of $I^{\times}$ explained by the first $l = 1, \ldots, 8$ factors is 63%, 81%, 87%, 93%, 98%, 99%, 99.6%, and 100%, respectively.

The factors may naturally be compared to those from Section 6.1.1 and the concise representation of the intents used in Figure 6.2 facilitates this comparison. We may notice that factors F_2, F_4, F_5 and F_6 here are very similar to factors F_6, F_2, F_7 and F_4 respectively from Section 6.1.1, they even pairwise equivalent intents. These factors are clearly interpretable. Factors F_1, F_3 and F_5 from Section 6.1.1 have some similarities with factors to F_2 and F_3 but interpretation of F_2 and F_3 is not so clear as interpretability the above four. The remaining factors here, F_1, F_7 and F_8 have no counterparts among those in Section 6.1.1 and seem to be not very interesting, particularly F_1, where all attributes are present in small degrees and which applies to all breeds.

To conclude, our experiments confirm that when using the alternative approach examined in Section 4.2, the number of factors needed for explaining data is larger. Moreover, the first, i.e. the most important, factor is not so clearly interpretable and perhaps also not so interesting compared to those obtained by the methods examined in this work, which directly works with degrees. We also lost some information such as that Golden Retrievers and Beagles have in high degree characteristics (intent of factor F_6) as Labrador Retrievers

and Yorkshire terriers.

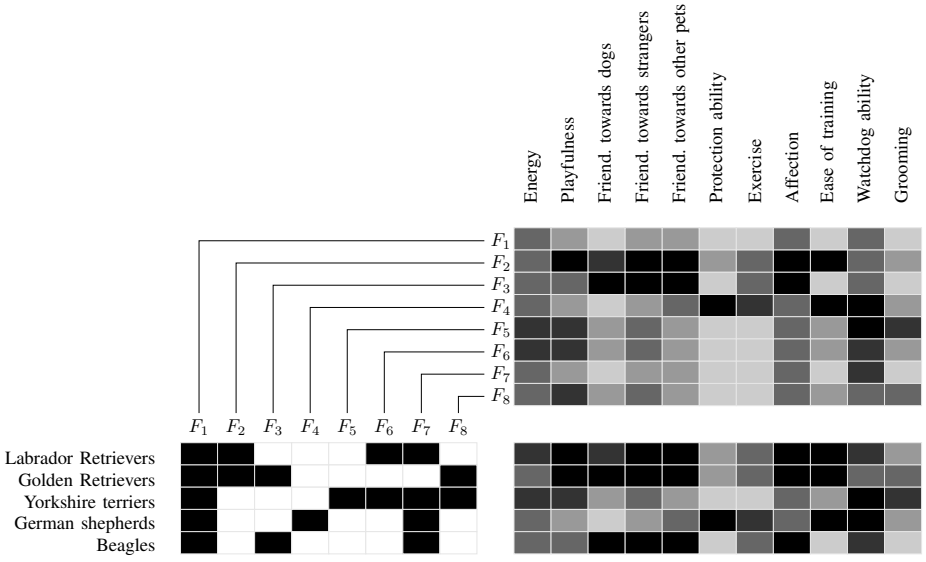


Figure 6.2: Decomposition of $I^{\times} = A_{\mathcal{F}^{\times}} \circ B_{\mathcal{G}^{\times}}$. I, $A_{\mathcal{F}}$, and $\circ B_{\mathcal{G}}$ are the bottom-right, bottom-left, and top matrix, respectively.

6.1.3 Results for NMF

We examine two algorithms for Non-negative matrix factorization such as an Alternating least-squares algorithm and a Multiplicative update algorithm. Since resulted matrices W and H include values that are not from scale L, we can not show them as boxes with shades of gray. Interpretation is also slightly different.

The resulted factorization is not exact, matrix product of matrices W and H is a lower rank approximation of I. They are chosen to minimize the root-mean-squared residual between I and product WH. This residual is equal to 0.0358.

Interpretation of obtained factors is a little bit different than interpretation of factors computed via methods based on FCA (such as GreConD$_L$, GreEss$_L$ etc.). Row i of I is approximately a linear combination of the rows of H with the coefficients being row i of W.

Attributes with high coefficients in one of the rows are "Protective ability", "Exercise", "Ease of training" similarly like factor F_2 from Section 6.1.1 and with high coefficient belongs to German shepherds in matrix W.

Matrices W and H obtained by Multiplicative update algorithm provide approximation with residual equal to 0.0331.

Similar observation like in case of previous decomposition, also here we can find factor (F_2), which we can be labeled *guardian dog*.

6.1.4 Results for GreEss$_L$

GreEss$_L$ in this example returns smaller number of factor than earlier mentioned GreConD$_L$. We obtain six factors instead of seven, but all of them are more or less the same as factors obtained in Section 6.1.1. For more details see Figure 6.3.

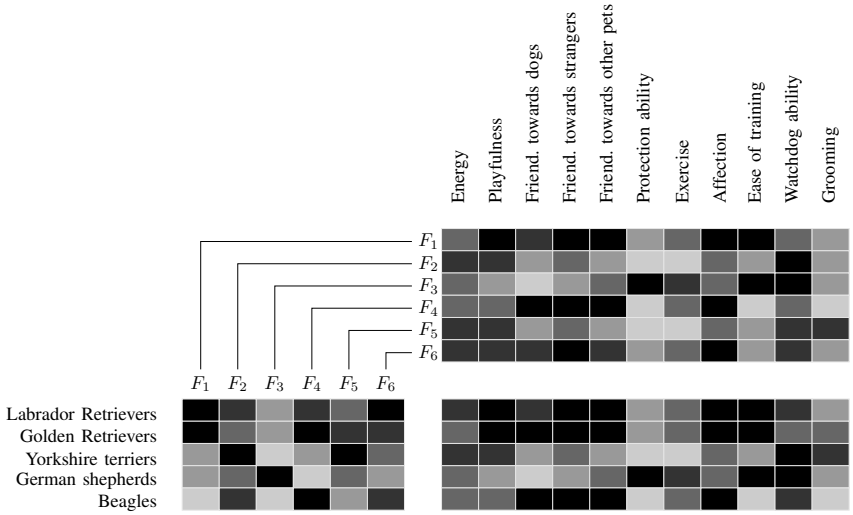


Figure 6.3: GreEss$_L$: Decomposition $I = A_{\mathcal{F}} \circ B_{\mathcal{F}}$. I, $A_{\mathcal{F}}$, and $B_{\mathcal{F}}$ are the bottom-right, bottom-left, and top matrix, respectively.

F_1, F_2, and F_3 explain, by and large, the whole data. In particular, the percentage of matrix I explained using the first l factors for $l = 1, \ldots 6$ is 71%, 84%, 92%, 98%, 99% and 100%. In comparison with coverage function in Section 6.1.1) the coverage function grows slower in the first three factors but then converges faster to 100%.

6.1.5 Results for Asso$_L$

As was mentioned earlier, Asso$_L$ usually does not return exact decomposition. We present here three obtained factors for setting: $w_0 = 1$, $w_1 = 1$, i.e. overcover and uncover errors have same weight and $\tau = 0.9$ (later we will show that choice of τ in Asso$_L$ does not rapidly change the result). Factors gradually cover 0.72%, 0.84% and 0.844% of input data.

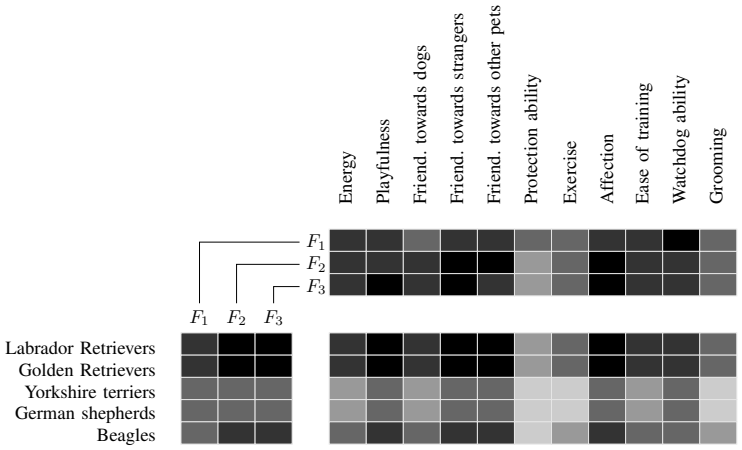


Figure 6.4: Asso$_L$: Decomposition $I \approx J = A_{\mathcal{F}} \circ B_{\mathcal{F}}$. J, $A_{\mathcal{F}}$, and $B_{\mathcal{F}}$ are the bottom-right, bottom-left, and top matrix, respectively.

The obtained factors are hard to explain, the most important factor (the first one) does not hold any important information. It is caused by the fact that when $|L| > 2$ (non-Boolean

case), rectangles with values "around the middle" in L, such as 0.4 and 0.6 in this example, which may be produced as factors by ASSO_L have a good coverage and are thus sometimes selected by ASSO_L in spite of a possible difficulty in interpreting such factors.

6.1.6 Results for GreConD$_L$+

Like in ASSO_L, we permit overcover errors in GreConD$_L$+. How big is this error is driven by a choice of the parameter w. The larger w, the smaller overcover error.

For example, if we take $w = 0.5$ we obtain coverage by three factors and overall overcover error is 34%. The first factor covers 85% with overcover error 29%, the second one covers 97% (error 34%), the last one ensures full coverage and does not increase the overcover error. With $w = 1$ we need four factors to cover all inputs and error is 28%. Computed factorizations can be seen in Figures 6.5 and 6.6.

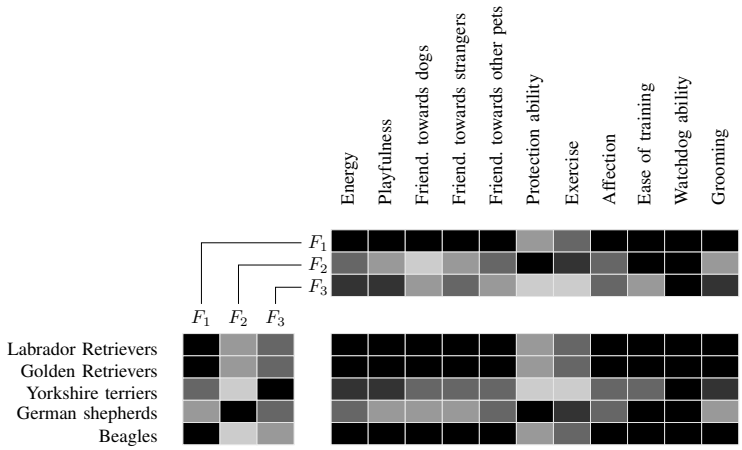


Figure 6.5: GreConD$_L$+, $w = 0.5$: Decomposition $I \approx J = A_{\mathcal{F}} \circ B_{\mathcal{F}}$. J, $A_{\mathcal{F}}$, and $B_{\mathcal{F}}$ are the bottom-right, bottom-left, and top matrix, respectively.

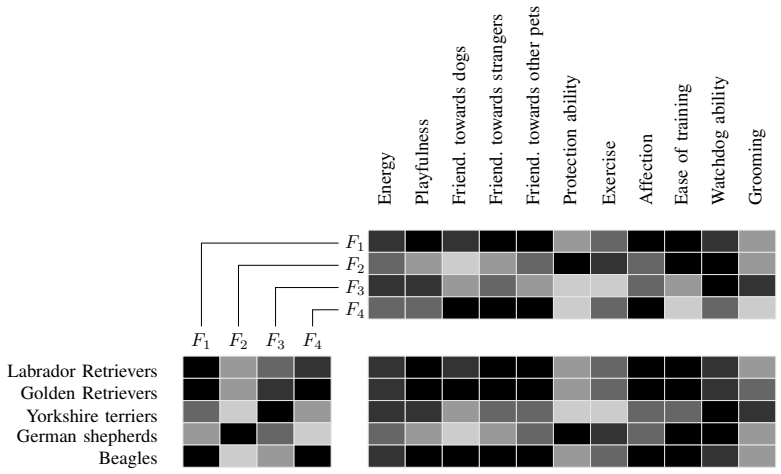


Figure 6.6: GreConD$_L$+, $w = 1$: Decomposition $I \approx J = A_{\mathcal{F}} \circ B_{\mathcal{F}}$. J, $A_{\mathcal{F}}$, and $B_{\mathcal{F}}$ are the bottom-right, bottom-left, and top matrix, respectively.

The choice of w slightly changes the obtained factors, but the most important factors (first ones), are very similar.

Unlike factors obtained by ASSO_L, meaning of factors is more relevant. We can see that there are factors *friendliness*, *guardian dog* like in Section 6.1.1, i.e. factors F_1 and F_2 are nearly the same as factors F_1 and F_2 from Section 6.1.1.

6.2 Real data

The datasets and their characteristics are described in Table 6.1, in which $|L|$ denotes the number of truth degrees in the scale L and $||I||$ denotes the number of non-zero entries in the input matrix I. Since we are interested also in analysis via algorithm GREESS_L, another interesting characteristic is number of non-zero entries in the essential part $\mathcal{E}(I)$.

Another reason is described in Section 6.2.1. Factors obtained by GRECOND_L, GREESS_L and GRECOND_L+ are basically very similar. This is why we mainly focus on describing the factors obtained by GREESS_L, unless those of other methods reveal a different insight.

dataset	size	$\|L\|$	$\|\|I\|\|$	$\|\|\mathcal{E}(I)\|\|$	$\|\|\mathcal{E}(I)\|\|/\|\|I\|\|$
Breeds	151×11	6	1963	362	0.184
Decathlon	28×10	5	266	59	0.221
IPAQ	4510×16	3	41624	1281	0.031
Music	900×26	7	20377	5952	0.292
Rio	87×31	4	402	332	0.820

Table 6.1: Real data

Dog breeds[2] extends the dataset from Section 6.1 to 151 breeds. GREESS_L found 20 factors providing an exact decomposition of the 151×11 matrix I, but already the set $\mathcal{F}_3$ consisting of the first three most important factors explains a large portion of the data. In particular, the degree $s(I, A_{\mathcal{F}_3} \circ B_{\mathcal{F}_3})$ of closeness of I to the matrix $A_{\mathcal{F}_3} \circ B_{\mathcal{F}_3}$ reconstructed from the first three factors, which is defined by (3.1), equals 0.795. Among these factors is a formal concept containing the attributes "playfulness", "ease of train", and "affection" to degree 1 and "energy" to a high degree. This factor may be interpreted as the ability to *excel in sports* (such as agility, flyball, frisbee) and to serve as *guide* and *therapy dogs*. This factor applies to high degree to breeds such as Golden Retriever, Labrador Retriever, or Papillon. Another factor is a formal concept containing "Protection ability" and "Watchdog ability" with high degrees. Such factor may be interpreted as the ability to serve as a *guardian dog* and applies e.g. to American Staffordshire Terrier, Anatolian Shepherd, Belgian Malinois, Belgian Sheepdog, Kuvasz, German Shepherd Dog, and Doberman Pinscher. Interestingly, these two factors are similar to factors F_3 and F_2 described in Section 6.1. In fact, the factors F_1, F_2, and F_3 from Section 6.1, when extended to the 151×11 (in terms of Section 3.4.1) matrix, cover 0.85 of the matrix according to s, illustrating an interesting natural property that we observed in several examples.

[2] `http://www.petfinder.com/`

Decathlon[3] extends the dataset from [18] to 28×10 matrix I (28 athletes, 10 disciplines of decathlon) using a five-element scale L.

Using GreEss$_L$, we obtained 10 factors that we consulted with an experienced decathlon coach. Among the most important factors are the ones that can be interpreted as *speed*, containing to high degrees the attributes "100m", "Long jump", "400m", and "Hurdles"; *explosiveness*, containing to high degrees the attributes "Long jump", "Shot put", "High jump", and "Javelin"; and a factor containing "High jump" and "1500m", typical of lightweight athletes. All these factors were found natural by the decathlon coach.

Asso$_L$ computed a set $\mathcal{F}$ of 5 factors which reconstruct 80% of the the input data, i.e. $s(I, A_{\mathcal{F}} \circ B_{\mathcal{F}}) = 0.8$. Covering by factors is in order 76% for first factor, 86% for first two factors, 87%, 88% and 88, 2% for all five factors.

The most interesting is the second most important in terms of coverage, which contains "1500m" with degree 1, and "100m", "400m", and "Hurdles" with degree 0.75, i.e. a factor that may be naturally termed as *running capability*.

In this example, we observed factors which are not easy to interpret and whose appearance is discussed in Section 6.2.1, namely factors which apply to relatively higher degrees to all athletes and are manifested to high degrees by every discipline.

IPAQ data[4] consists of international questionnaire data regarding physical activity of population and involves 4510 respondents answering 16 questions using a three-element scale. This questionnaire is considered important from health management point of view, particularly as a source for making government decisions regarding health policy. The questions include respondents age, sex, body-mass-index (BMI), health, to what extent the person bicycles, walks, etc. GreEss$_L$ produced 17 factor concepts providing an exact decomposition of the 4510×16 matrix. As with the other examples, the first 3–4 factors may be considered sufficient to explain the data. First factor explains 64%, first two factors 74% and three factors explain 80% of data. One may see great reduction of input entries using essential part in this dataset. GreEss$_L$ returns the smallest number of factors compared to other methods providing exact decomposition. On the other hand factors obtained via GreConD$_L$ or GreConD$_L$+ are more interesting, because factors obtained via GreEss$_L$ correspond more or less to a single attribute. Based on the attributes present in the factors, the first factor returned from GreConD or GreConD+ corresponds to and thus may be interpreted as *healthy people with good education who cycle on a regular basis*; the second one as *people with normal BMI who walk on a daily basis*; the third one as *people who are employed, own a car, and cycle on a regular basis*.

Music data The data comes from [27] and consists of results of a study inquiring people's perception of some speed of song depending of various characteristics of the songs. The data was collected by questionnaires involving 30 participants who were presented 30 samples (29 complex music samples and one simple tone of 528Hz). The participants recorded their emotional experience using 26 attributes each using a 6-element scale L, along with a retrospective time duration and time passage judgement. The data is then represented by a 900×26 matrix with entries in L. Using GreEss$_L$, we obtained 29 factors. The

[3] `http://www.sports-reference.com/`

[4] `http://www.ipaq.ki.se/`, Belohlavek et al., Inf. Sciences 181(2011), 1774–1786.

authors of this study examined the factors and concluded that the groups of music samples corresponding to the factors are meaningful and that the factors can be interpreted in terms of emotional experience. For example, an interesting factor with a good coverage contained songs No. 5, 7, 16, and 26, all of which are melancholic. Another factor was the one clearly separating the simple tone to which it applied to degree 1, while applying to degree 0 or to other degrees close to 0 to the other samples. Among other interesting factors are the one manifested to a high degree by attributes "Ugly" and "Violent"; the one manifested by "Restful", "Safe", "Stable", and "Inert"; and the factor manifested by "Successful", "Valuable", "Meaningful", and "Significant". All these factors represent significant categories of songs.

Rio data [5] represents 87×31 matrix I and consist of 87 countries that obtained any medal in one of 31 sport on Olympics games in Rio de Janeiro 2016. L contains four grades—1 means that country won at least one gold medal, $\frac{2}{3}$ at least silver medal, $\frac{1}{3}$ at least one bronze medal and 0 no medal in this sport. This dataset is very sparse in comparison with other presented datasets. Great portion of input entries are Essential, i.e. we observe that the ratio $||\mathcal{E}(I)||/||I||$ of the number of entries in $\mathcal{E}(I)$ to the corresponding number for I is high.

Using GreEss$_L$ we computed 32 factors, but it is sufficient take 19 factors to explain more than 90% of data.

Among the most important factors can be found factor containing *martial arts*, which has degree 1 in the attributes "Judo", "Wrestling" and high degree in "Weightlifting". Another one can be interpreted as *water sports*, containing in high degrees the attributes "Canoeing", "Rowing", "Sailing", and "Swimming".

Asso$_L$ returned different factors. Let us mentioned one factor, which grouped attributes "Archery" and "Shooting", i.e. sports with *the ability to aim*.

6.2.1 Evaluation

Table 6.2 and Table 6.3 display the numbers of factors produced by the algorithms from Section 4 and Section 5 that are needed to achieve a prescribed coverage. That is, we observe the smallest l such that for the set $\mathcal{F}$ of the first l factors produced by the respective algorithm, $s_{\approx}(I, A_{\mathcal{F}} \circ B_{\mathcal{F}})$ ($s_{=}(I, A_{\mathcal{F}} \circ B_{\mathcal{F}})$ respectively) exceeds the prescribed value. For example, the first row in Table 6.2 corresponding to Breeds indicates that we need six factors in case of GreConD of ordinally scaled attributes, three in case of GreConD$_L$, two in case of GreEss$_L$ etc. to have $s_{\approx}(I, A_{\mathcal{F}} \circ B_{\mathcal{F}}) \geq 0.75$. "NA" indicates that the prescribed coverage is not achievable by the factors produced by Asso$_L$. Observe that in accordance with the theoretical results, "NA" never appears for other algorithm than Asso$_L$, because the other algorithms eventually compute an exact decomposition.

Results for $s_{\approx}$ The results illustrate that, by and large, the number factors produced by GreConD on dataset with ordinally scaled attributes is significantly larger in comparison with other algorithms. Moreover the first couple of factors produced by Asso$_L$ and GreConD$_L$+ has a better coverage compared to the same number of factors produced

[5] `https://www.rio2016.com/en/medal-count`

dataset	s	ordinally scaled attributes	number of factors needed: GreConD$_L$	Asso$_L$	GreEss$_L$	GreConD$_L$+ ($w = 0.5$)	GreConD$_L$+ ($w = 1$)
Breeds	0.75	6	3	2	3	1	1
	0.85	12	5	3	7	2	3
	0.95	25	9	NA	11	6	8
	1	57	16	NA	15	14	13
Decathlon	0.75	5	1	1	3	1	1
	0.85	8	4	2	5	1	2
	0.95	16	8	NA	8	4	6
	1	31	15	NA	10	11	13
IPAQ	0.75	6	8	1	10	2	2
	0.85	10	12	1	12	3	4
	0.95	19	18	NA	15	8	9
	1	46	32	NA	17	20	23
Music	0.75	28	7	1	7	3	1
	0.85	51	13	NA	14	5	6
	0.95	105	24	NA	25	13	18
	1	280	36	NA	29	29	30
Rio	0.75	1	12	1	2	1	1
	0.85	9	16	1	6	1	1
	0.95	30	24	18	17	5	8
	1	79	35	NA	32	32	33

Table 6.2: Quality of decompositions (real data) for $s_{\approx}$.

dataset	s	ordinally scaled attributes	number of factors needed: GreConD$_L$	Asso$_L$	GreEss$_L$	GreConD$_L$+ ($w = 0.5$)	GreConD$_L$+ ($w = 1$)
Breeds	0.50	1	7	NA	3	NA	5
	0.75	6	7	NA	3	NA	NA
	0.95	25	12	NA	11	NA	NA
	1	57	16	NA	15	NA	NA
Decathlon	0.50	3	3	NA	3	1	2
	0.75	5	5	NA	6	NA	NA
	0.95	16	11	NA	9	NA	NA
	1	31	15	NA	10	NA	NA
IPAQ	0.50	1	1	NA	2	2	1
	0.75	6	6	NA	7	NA	10
	0.95	19	28	NA	14	NA	NA
	1	46	32	NA	17	NA	NA
Music	0.50	7	10	NA	9	NA	25
	0.75	28	20	NA	19	NA	NA
	0.95	105	34	NA	26	NA	NA
	1	280	36	NA	29	NA	NA
Rio	0.50	1	1	1	1	1	1
	0.75	1	1	1	1	1	1
	0.95	30	12	NA	13	NA	17
	1	79	35	NA	32	NA	NA

Table 6.3: Quality of decompositions (real data) for $s_{=}$.

by GreEss$_L$ or GreConD$_L$. On the other hand, beyond certain coverage, Asso$_L$ stops producing factors and is not able to compute an (exact) decomposition of I, while other algorithms always compute an exact decomposition, with a reasonably small number of factor needed for coverage very close to 1. This is congruent with the fact that Asso$_L$ and GreConD$_L$+ are primarily designed for DBP(L) and the rest of the algorithms is primarily designed for AFP(L), as well as with the available evidence from the Boolean case.

We found that factors produced by Asso$_L$ are not easy to interpret compared to other algorithms. There are two reasons. The first, mentioned in Section 6.2, is the usage of formal concepts as factors by GreConD, GreConD$_L$, GreEss$_L$, GreConD$_L$+ and their good interpretability. The second one consists in that when $|L| > 2$ (non-Boolean case) rectangles with values "around the middle" in L, such as $\frac{1}{2}$, which may be produced as factors by Asso$_L$ have a good coverage and are thus sometimes selected by Asso$_L$ in spite of a possible difficulty in interpreting such factors. In more detail, note that for Boolean data, the values I_{ij} in the input matrix I are approximated by 0 or 1 of $(A_{\mathcal{F}} \circ B_{\mathcal{F}})_{ij}$ only. Hence, in case of mismatch the entry $\langle i, j \rangle$ contributes by $I_{ij} \leftrightarrow (A_{\mathcal{F}} \circ B_{\mathcal{F}})_{ij} = 0$ to the numerator in (3.1). With more degrees in L, the situation is different. For example, if $\frac{1}{2}$ is available and if $0 \leftrightarrow \frac{1}{2} = 1 \leftrightarrow \frac{1}{2} = \frac{1}{2}$, then already the trivial matrix $A_{\mathcal{F}} \circ B_{\mathcal{F}}$ with all entries equal to $\frac{1}{2}$, which is obtained from the "constant average factors", always satisfies $s_{\approx}(I, A_{\mathcal{F}} \circ B_{\mathcal{F}}) \geq \frac{1}{2}$. One therefore has to be aware of this effect of presence in L of the "middle" degrees on the values of $s_{\approx}$.

New algorithm GreEss$_L$ requires less factors to achieve a prescribed coverage than the previous algorithm GreConD$_L$ from [18]. The reason is a better utilization of the geometry of decompositions by GreEss$_L$ particularly of the essential part of I.

Results for $s_=$ As was mentioned in Section 6.1.2 in Boolean case (algorithm GreConD on ordinally scaled attributes) it holds that $s_{\approx}$ is equal to $s_=$, so results in corresponding column are the same. Big change is in case of GreConD$_L$+. This algorithm like Asso$_L$ algorithm allows overcover error, so we are not able usually achieve $s_= = 1$. This error grows with smaller parameter w. More precisely in Breed dataset finally only 47% of entries matrix I are the same as appropriate element in $A_{\mathcal{F}} \circ B_{\mathcal{F}}$ for parameter $w = 0.5$ and 65% for parameter $w = 1$. In Table 6.4 we show the percentage of the same entries for all datasets.

dataset	Asso$_L$	GreConD$_L$+ ($w = 0.5$)	GreConD$_L$+ ($w = 1$)
Breed	29%	47%	65%
Decathlon	24%	51%	73%
IPAQ	24%	69%	80%
Music	12%	35%	52%
Rio	88%	87%	97%

Table 6.4: Percentage of $s_=$

6.3 Synthetic data

We used synthetic data organized in collections Set 1–5, each consisting of 500 $n \times m$ matrices I. The characteristics of these datasets are described in Table 6.5. Each matrix I is obtained as a product of $n \times k$ and $k \times m$ randomly generated matrices A and B in which entries from scale L are selected according to a prescribed probability distribution. For instance, in Set 2 we used a five-element scale $L = \{0, \frac{1}{4}, \frac{1}{2}, \frac{3}{4}, 1\}$ with the probabilities $p(a)$ of the degrees $a \in L$ in A and B being $p(0) = p(\frac{1}{4}) = \frac{1}{8}$ and $p(\frac{1}{2}) = p(\frac{3}{4}) = p(1) = \frac{1}{4}$. The probability distributions generalize the commonly considered densities of Boolean matrices, e.g. for $L = \{0, 1\}$ the distribution $[\frac{1}{4}\ \frac{3}{4}]$ corresponds to density 0.75. Table 6.5 also contains the average characteristics of synthetic data with the averages over all matrices in Set i. The characteristics are the same as for the real data. One observes that the reduction in number of nonzero entries is significant as in the case of real data. We present similar experiment in [8]. Set 5 was added and the rest of the sets have the same characteristic, but they are different, for the purpose of this work we generate new ones.

dataset	size	$\|L\|$	k	distribution on L in A and B	avg $\|\|I\|\|$	avg $\|\|\mathcal{E}(I)\|\|$	avg $\|\|\mathcal{E}(I)\|\|/\|\|I\|\|$
Set 1	50×50	3	10	$[\frac{1}{3}\ \frac{1}{3}\ \frac{1}{3}]$	2449	195	0.080
Set 2	50×50	5	10	$[\frac{1}{8}\ \frac{1}{8}\ \frac{1}{4}\ \frac{1}{4}\ \frac{1}{4}]$	2503	355	0.141
Set 3	100×50	5	25	$[\frac{1}{8}\ \frac{1}{8}\ \frac{1}{4}\ \frac{1}{4}\ \frac{1}{4}]$	4983	602	0.121
Set 4	100×100	5	20	$[\frac{1}{8}\ \frac{1}{8}\ \frac{1}{4}\ \frac{1}{4}\ \frac{1}{4}]$	10000	2087	0.209
Set 5	500×100	6	25	$[\frac{1}{6}\ \frac{1}{6}\ \frac{1}{6}\ \frac{1}{6}\ \frac{1}{6}\ \frac{1}{6}]$	49997	14216	0.284

Table 6.5: Synthetic data and their characteristics.

6.3.1 Evaluation of explanation data

We observe the ability of the extracted factors to explain, i.e. reconstruct, the input data and measure it by the degree of similarity $s_{\approx}(I, A_{\mathcal{F}} \circ B_{\mathcal{F}})$ defined by (3.1), where $\mathcal{F}$ is the examined set of factors (usually the first k factors obtained by the algorithm). In view of Section 2.3, we speak of coverage of data by factors.

Table 6.6, Table 6.7 and Figure 6.7 display selected results of coverage $s_{\approx}$, defined by (3.1), by the first k factors for the datasets and the two algorithms. We also include the percentage $s_{=}$. "–" means that no new factors were produced increasing k.

Note that the values of $s_{\approx}$ tend to be high even for a small number of computed factors and that they are higher than what one usually observes for Boolean data. The reason is the same like in the case of real datasets and is explained in Section 6.2.1.

In results on synthetics data, we observe the same behaviour as in case of real datasets presented in Section 6.2.1.

6.3.2 Role of τ in Asso$_L$ algorithm

In presence of several degrees in L, one may observe a new phenomenon. It is known that for Boolean data the selection of the threshold τ significantly influences the performance of

dataset	k	ordinally scaled attributes	coverage $s/s_=$ by the first k factors GreConD$_L$	Asso$_L$	GreEss$_L$	GreConD$_L$+ ($w = 0.5$)	GreConD$_L$+ ($w = 1$)
Set 1	1	0.648	0.576/0.350	0.878/0.761	0.525/0.309	0.745/0.470	0.745/0.470
	4	0.837	0.866/0.744	0.899/0.805	0.866/0.744	0.943/0.780	0.936/0.781
	11	0.975	0.992/0.985	–	1/1	1/0.858	0.999/0.892
	12	0.982	0.995/0.990	–	–	–	1/0.892
	17	0.999	1/1	–	–	–	–
	19	1	–	–	–	–	–
Set 2	1	0.674	0.620/0.253	0.795/0.389	0.632/0.206	0.836/0.410	0.836/0.410
	2	0.763	0.782/0.434	0.839/0.410	0.820/0.483	0.921/0.524	0.921/0.524
	10	0.958	0.995/0.980	–	1/1	0.999/0.683	0.998/0.735
	11	0.967	0.997/0.989	–	–	1/0.684	0.999/0.738
	12	0.975	0.998/0.524	–	–	–	1/0.738
	13	0.980	1/1	–	–	–	–
	23	1	–	–	–	–	–
Set 3	1	0.780	0.684/0.188	0.899/0.789	0.728/0.349	0.852/0.412	0.852/0.412
	3	0.845	0.828/0.386	0.950/0.807	0.790/0.508	0.923/0.632	0.923/0.632
	19	0.966	0.966/0.867	–	0.979/0.950	1/1	0.998/0.867
	27	0.987	0.986/0.947	–	0.998/0.977	–	1/1
	39	0.998	0.998/0.994	–	1/1	–	–
	47	0.999	1/1	–	–	–	–
	53	1	–	–	–	–	–

Table 6.6: Coverage $s_\approx$ and $s_=$ by the first k factors.

Asso [46]. An intuitive explanation is that with 0 and 1 as the only degrees, the decision based on τ whether to round off the confidence value to 0 or 1 is significant. We observed that in the setting with several degrees, the choice of τ becomes less significant as the number of degrees increases. This is a good feature for a user because the value of τ needs to be selected by the user but there are no known principles, except for ad hod recommendations, how to make such a choice.

We used five sets of synthetic datasets with size of L in order 3, 5, 11, 21 and 101. Table 6.8 presents the values of coverage $s_\approx$ and $s_=$ corresponding to the first factor and to all the factors obtained. The values are observed for different values of τ. As one can see, as the size of L increases, the coverage values for different values of τ tend to be the same. Note that the low values in $s_=$, particularly for scales L with a larger number of degrees, indicating a low number of entries for which the input matrix and the matrix reconstructed from the factors have equal values, are due to the aim of Asso$_L$ to generate approximate rather than exact decompositions.

For all of the datasets we obtain best coverage for τ between 0.85 and 0.95. For datasets with smaller size of L we obtain different coverage for $\tau = 0.85$ and $\tau = 0.95$. In datasets Set 4 and Set 5 this difference is small. See Table 6.8. The entries depict mean coverage for first factor/coverage for all factors.

dataset	k	ordinally scaled attributes	coverage $s/s_=$ by the first k factors GreConD$_L$	Asso$_L$	GreEss$_L$	GreConD$_L$+ ($w = 0.5$)	GreConD$_L$+ ($w = 1$)
Set 4	1	0.738	0.651/0.211	0.921/0.704	0.648/0.205	0.846/0.427	0.846/0.427
	4	0.840	0.827/0.603	0.939/0.722	0.854/0.512	0.967/0.694	0.963/0.701
	21	0.985	0.975/0.824	–	0.994/0.979	1/0.771	0.999/0.805
	27	0.997	0.998/0.905	–	1/1	–	1/0.808
	29	0.998	1/1	–	–	–	–
	37	1	–	–	–	–	–
Set 5	1	0.569	0.511/0.111	0,886/0,535	0.477/0.068	0.765/0.234	0.765/0.234
	5	0.750	0.821/0.419	0,887/0,541	0.798/0.328	0.953/0.546	0.931/0.531
	27	0.948	0.995/0.979	–	0.995/0.949	1/0.621	0.999/0.701
	31	0.964	0.998/0.990	–	0.999/0.974	–	1/0.832
	36	0.974	0.999/0.999	–	1/1	–	–
	42	0.986	1/1	–	–	–	–
	109	1	–	–	–	–	–

Table 6.7: Coverage $s_\approx$ and $s_=$ by the first k factors ctd.

	$s_\approx$			$s_=$		
dataset	$\tau = 0.85$	$\tau = 0.9$	$\tau = 0.95$	$\tau = 0.85$	$\tau = 0.9$	$\tau = 0.95$
Set 1	0.85/0.87	0.87/0.88	0.83/0.86	0.68/0.73	0.63/0.65	0.46/0.50
Set 2	0.87/0.87	0.86/0.90	0.85/0.89	0.29/0.33	0.35/0.37	0.26/0.28
Set 3	0.87/0.87	0.88/0.88	0.87/0.87	0.12/0.17	0.12/0.19	0.14/0.19
Set 4	0.88/0.88	0.88/0.88	0.88/0.88	0.04/0.06	0.04/0.07	0.04/0.06
Set 5	0.87/0.88	0.87/0.88	0.87/0.88	0.006/0.01	0.006/0.01	0.006/0.01

Table 6.8: Coverage by the first factor/by all factors obtained for different values of τ.

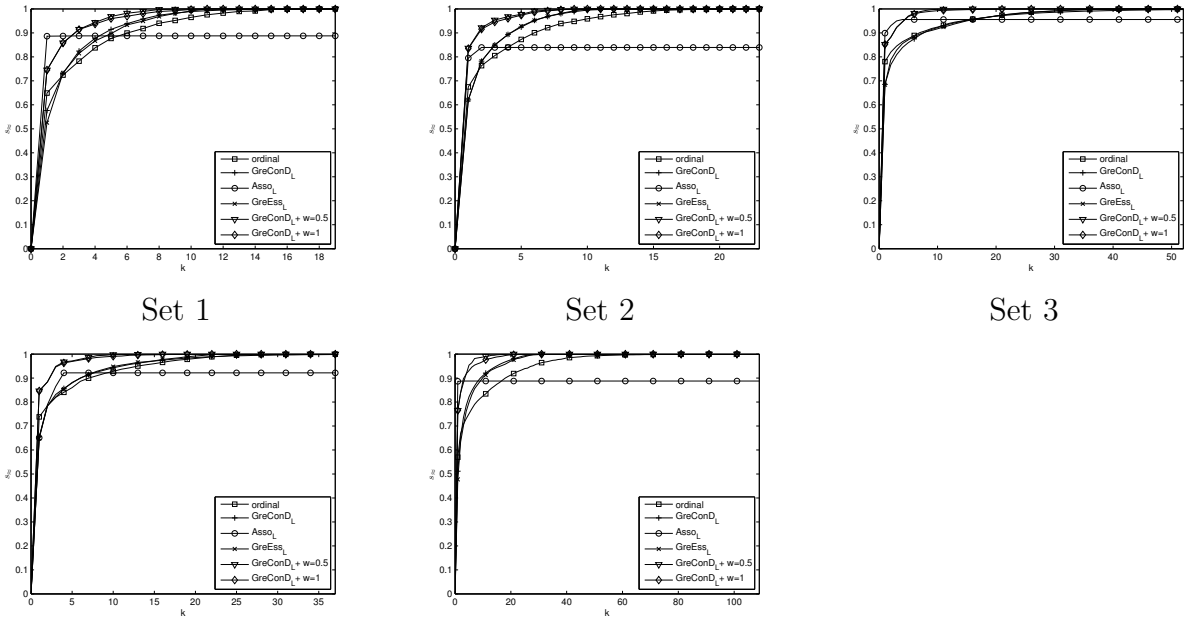


Figure 6.7: Coverage $s_\approx$ by the k factors

Chapter 7

Conclusion

In this paper we generalized the Boolean matrix decomposition problem (BMF), took into account matrices over scales which represent ordinal data. We proposed answers to natural question: "How well a set of factors explains 'the data?" Moreover, we present a problem of explaining data by factors obtained from reduced data—data having same attributes but the smaller number of objects. We propose heuristic to deal with problem of selection from a possibly large dataset a smaller one such that the factors of the reduced dataset explain the large dataset well.

The main part of this work presents existing and new algorithms for decomposition of matrices with ordinal attributes. We introduce three new algorithms, namely GreEss$_L$, Asso$_L$ and GreConD$_L$+, all based on more or less known BMF algorithms. We supported the correctness of these algorithms by theoretical results regarding the geometry of decompositions and by experimental evaluation presented in this paper. It turns out that the methods yield reasonable and, in a sense, robust factors and that the results of the methods are easy to understand. We also shown that methods suited for ordinal matrices returns better results, than BMF methods on scaled data.

Decomposition of matrices over some scale is still not well understood problem. There is a lot of unresolved issues including for example: the choice of the scale of degrees, the operation $\otimes$ or a problem we addressed in Chapter 6—Asso$_L$ returns rectangles with values "around the middle" in L.

Abundantly discussed topic in data mining community is, in case of BMF, noise in Boolean data. This issue should be investigated in general case as well.

Shrnutí v českém jazyce

Rozklady Booleovských matic (BMF), rozklady matic, které obsahují pouze nuly a jedničky, také známé jako faktorizace Booleovských dat, se čím dál tím více těší pozornosti dataminingové komunity. Cílem BMF je hledání, v datech skrytých důležitých informací – faktorů – pomocí nichž lze vysvětlit či popsat originální data. Postupem času vznikla celá řada metod pro BMF. Cílem této práce je prozkoumat rozšíření těchto maticových rozkladů pro data, která nemají jen binární charakter, ale jejichž vstupy jsou z uspořádané škály. Takováto generalizace sebou přináší několik netriviálních problémů, které jsou rovněž diskutovány v této práci.

První část práce je věnována popisu problému rozkladů matic s odrinálními daty, na který můžeme nahlížet jako na problém pokrývání. Jsou zde stručně popsány matematické základy, které při řešení využíváme. Jedním z nich je Fuzzy logika, obzvláště pak kalkulus na reziduovaných svazech. Dále pak Formální konceptuální analýza – zde s výhodou využíváme faktu, že faktory vybrané z množiny formálních konceptů tvoří optimální řešení.

Navíc v první části práce prezentujeme nové teoretické výsledky, které pak využijeme při návrhu nových algoritmů. Především využíváme toho, že v binárním případě se ukázalo, že ne všechna políčka jsou rovnocenná. K tomu abychom pokryli celá vstupní data stačí pokrýt jen některá políčka. Tyto prvky nazýváme esenciální a jsou definovány přes minimální intervaly v konceptuálním svazu. Ukazujeme, že i v obecném případě lze nalézt ekvivalentní pojem. Generalizace není úplně přímočará a přináší několik výzev. Například každému esenciálnímu prvku může odpovídat více intervalů v kontrastu s binárním případem, kde interval je pouze jeden.

V druhé části práce představujeme již existující metody pro dekompozici matic s ordinálnímu daty. Konkrétně GreCon$_L$, GreConD$_L$ a Non-negative matrix factorisation (NMF). Navíc demonstrujeme možnosti využití existujících BMF metod na data, která získáme ordinálním škálováním. Dále představíme tři nové algoritmy, jejichž myšlenka pochází z BMF algoritmů.

Poslední část je věnována experimentální analýze a srovnání představených algoritmů. Zaměřujeme se především na interpretovatelnost faktorů, získaných z jednotlivých metod, počty faktorů a kvalitu pokrytí – jak velká část dat je vysvětlena získanými faktory. Experimenty provádíme na syntetických a reálných datech.

Bibliography

[1] L. B. Alexandrov, S. Nik-Zainal, D. C. Wedge, P. J. Campbell, M. R. Stratton, "Deciphering Signatures of Mutational Processes Operative in Human Cancer", *Cell Reports* 3 (1) (2013), 246–259.

[2] H. Andrews, C. Patterson, "Singular Value Decomposition (SVD) Image Coding", *IEEE Transactions on Communications* 24 (4) (2003), 425–432.

[3] R. Belohlavek, *Fuzzy Relational Systems: Foundations and Principles*, Kluwer, Academic/Plenum Publishers, New York, 2002.

[4] R. Belohlavek, "Concept Lattices and Order in Fuzzy Logic", *Annals of Pure and Applied Logic* 128 (1–3) (2004), 277–298.

[5] R. Belohlavek, "Optimal Decompositions of Matrices with Entries from Residuated Lattices", *J. Logic and Computation* 22 (6) (2012), 1405–1425.

[6] R. Belohlavek, "Ordinally Equivalent Data: A Measurement-Theoretic Look at Formal Concept Analysis of Fuzzy Attributes", *Int. Journal of Approximate Reasoning* 54 (9) (2013), 1496–1506.

[7] R. Belohlavek, M. Krmelova, "Factor Analysis of Sports Data via Decomposition of Matrices with Grades", CLA 2012, pp. 293–304, 2012.

[8] R. Belohlavek, M. Krmelova, "Beyond Boolean Matrix Decompositions: Toward Factor Analysis and Dimensionality Reduction of Ordinal Data", ICDM 2013, pp. 961–966, 2013.

[9] R. Belohlavek, M. Krmelova, "Factor Analysis of Ordinal Data via Decomposition of Matrices with Grades", *Annals of Mathematics and Artiffcial Intelligence* 72 (1–2) (2014), 23–44.

[10] R. Belohlavek, J. Outrata, M. Trnecka, "Impact of Boolean Factorization as Preprocessing Methods for Classification of Boolean Data", CLA 2012, pp. 305–316, 2012.

[11] R. Belohlavek, J. Outrata, M. Trnecka, "Impact of Boolean Factorization as Preprocessing Methods for Classification of Boolean data", *Annals of Mathematics and Artificial Intelligence* 72(1-2)(2014), 3–22.

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

[13] R. Belohlavek, M. Trnecka, "A New Algorithm for Boolean Matrix Factorization which Admits Overcovering", To appear in Discrete Applied Mathematics.

[14] R. Belohlavek, M. Trneckova, "The Asso algorithm for graded attributes", Unpublished manuscript.

[15] R. Belohlavek, M. Trneckova, "Toward a geometry of decompositions of matrices with grades", Unpublished manuscript.

[16] R. Belohlavek, M. Trneckova, "A decomposition algorithm for matrices with grades that admits overcovering", Unpublished manuscript.

[17] R. Belohlavek, V. Sklenar, J. Zacpal, "Crisply Generated Fuzzy Concepts", ICFCA 2005, Lecture Notes in Artificial Intelligence 3403, pp. 268–283, 2005.

[18] R. Belohlavek, V. Vychodil, "Factor Analysis of Incidence Data via Novel Decomposition of Matrices", *Lecture Notes in Artificial Intelligence* 5548(2009), 83–97.

[19] R. Belohlavek, V. Vychodil, "Discovery of Optimal Factors in Binary Data via A Novel Method of Matrix Decomposition", *J. Comp. and System Sciences* 76(1)(2010), 3–20.

[20] R. Belohlavek, V. Vychodil, "Formal Concept Analysis and Linguistic Hedges", *Int. J. General Systems* 41(5)(2012), 503–532.

[21] M. W. Berry, M. Browne, A. N. Langville, V. P. Pauca, R. J. Plemmons, "Algorithms and Applications for Approximate Nonnegative Matrix Factorization", *Computational Statistics & Data Analysis* 52 1(2007), 155–173.

[22] M. Chu, F. Diele, R. Plemmons, R. Ragni, "Optimality, Computation, and Interpretations of Nonnegative Matrix Factorizations", Unpublished Report, (2004) available at `http://www.wfu.edu/~plemmons`.

[23] P. Comon, "Independent Component Analysis, A New Concept?", *Signal Precessing* 36 (1994), 287–314.

[24] P. Cunningham, "Dimension Reduction", University College Dublin, Technical Report UCD-CSI-2007-7, 2007.

[25] W. J. Dixon(ed.), "BMDP Statistical Software Manual", Berkeley, CA: University of California Press, 1992.

[26] L. Eldén, "Matrix Methods in Data Mining and Pattern Recognition", SIAM, 2007.

[27] K. Flaska, P. Cakirpaloglu, "Identification of the Multidimensional Model of Subjective Time Experience", *Int. Studies in Time Perspective*, Imprensa da Universidade de Coimbra (2013), 259–273.

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

[29] F. Geerts, B. Goethals, T. Mielikäinen, "Tiling Databases", Proc. Discovery Science 2004, pp. 278–289.

[30] J. S. Golan, *Semirings and their Applications*, Springer, 1999.

[31] G. Golub, C. Van Loan, *Matrix Computations*, Johns Hopkins University Press, 1996.

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

[33] P. Hájek, *Metamathematics of Fuzzy Logic*, Kluwer, 1998.

[34] M. Huchard, A. Napoli, H. M. Rouane, P. Valtchev, "A Proposal for Combining Formal Concept Analysis and Description Logics for Mining Relational Data", ICFCA 2007, pp. 51–65, 2007.

[35] S. Karaev, P. Miettinen, J. Vreeken, "Getting to Know the Unknown Unknowns: Destructive-Noise Resistant Boolean Matrix Factorization", Proc. 2015 SIAM International Conference on Data Mining (SDM '15), pp. 325–333, 2015.

[36] M. Krmelova, M. Trnecka, V. Kreinovich, B. Wu, "How to Distinguish True Dependence from Varying Independence?", *Journal of Intelligent Technologies and Applied Statistics* 6(4)(2013), 339–351.

[37] M. Krmelova, M. Trnecka, "Boolean Factor Analysis of Multi-relational Data", CLA 2013, pp. 187–198, 2013.

[38] D. Lee, H. Seung, "Learning the Parts of Objects by Non-Negative Matrix Factorization", *Nature* 401 (1999), 788–791.

[39] D. Lee, H. Seung, "Algorithms for Non-Negative Matrix Factorisation", *Advances in Neural Information Processing Systems* 13 (2001), 556–562.

[40] R. Liao, Y-L. Boscolo, L. M. Yang, C. S. Tran, V. P. Roychowdhury, "Network Component Analysis", *PNAS* 100 (2003), 15522–15527.

[41] C. Lucchese, S. Orlando, R. Perego, "Mining Top-K Patterns From Binary Datasets in Presence of Noise", In: SIAM DM 2010, pp. 165–176, 2010.

[42] C. Lucchese, S. Orlando, R. Perego, "A Unifying Framework for Mining Approximate Top-k Binary Patterns", *IEEE Transactions On Knowledge and Data Engineering* 26(12):2900–2913, 2014.

[43] P. Miettinen, "The Boolean Column and Column-Row Matrix Decompositions", *Data Mining and Knowledge Discovery* 17(2008), 39–56.

[44] P. Miettinen, "Sparse Boolean Matrix Factorizations", Proc. IEEE ICDM 2010, pp. 935–940, 2010.

[45] P. Miettinen, "On Finding Joint Subspace Boolean Matrix Factorizations", In: SDM, pp. 954–965, 2012.

[46] P. Miettinen, T. Mielikäinen, A. Gionis, G. Das, H. Mannila, "The Discrete Basis Problem", *IEEE TKDE* 20 (2008), 1348–62.

[47] P. Miettinen, J. Vreeken, "Model Order Selection for Boolean Matrix Factorization", ACM SIGKDD 2011, pp. 51–59, 2011.

[48] G. T. Miller, "The mAgical Number Seven, Plus or Minus Two", *Psychol. Rev.* 63 (1956), 81–97.

[49] D. S. Nau, G. Markowsky, M. A. Woodbury, D. B. Amos, "A Mathematical Analysis of Human Leukocyte Antigen Serology", *Math. Biosci* 40 (1978), 243–270.

[50] F. A. Nielsen, D. Balslev, L. K. Hansen, "Mining the Posterior Cingulate: Segregation Between Memory and Pain Components", *NeuroImage* 27 3 (2005), 520–522.

[51] F. A. Nielsen, *Clustering of Scientific Citations in Wikipedia* Wikimania (2008).

[52] P. Paatero, U. Tapper, "Positive Matrix Factorization: A Non-negative Factor Model with Optimal Utilization of Error", *Environmetrics*, 5 (1994), 111–126.

[53] K. Pearson, "On Lines and Planes of Closest Fit to Systems of Points in Space", *Philosophical Magazine*, 2 (1901), 559–572.

[54] D. A. Simovici, C. Djeraba, *Mathematical Tools for Data Mining*, Springer, 2008.

[55] C. Spearman, "General Intelligence", Objectively Determined and Measured, *American Journal of Psychology*, 15 (1901), 201–293.

[56] G. W. Stewart, "On the Early History of The Singular Value Decomposition", *SIAM Review*, 35 (1993) 551–566

[57] L. Stockmeyer, *The Set Basis Problem is NP-complete.* Tech. Rep. RC5431, IBM, Yorktown Heights, NY, USA, 1975.

[58] L. Taslaman, B. Nilsson, "A Framework for Regularized Non-Negative Matrix Factorization, With Application to The Analysis of Gene Expression Data", *PLoS One 7* 11 (2012).

[59] N. Tatti, T. Mielikäinen, A. Gionis, H. Mannila, "What is The Dimension of Your Binary Data?", Proc. IEEE ICDM 2006, pp. 603–612, 2006.

[60] M. Trnecka, M. Trneckova, "An Algorithm for the Multi-Relational Boolean Factor Analysis based on Essential Elements", CLA 2014, pp. 107–118, 2014.

[61] M. Trnecka, M. Trneckova, "Decomposition of Boolean Multi-Relational Data with Graded Relations", IEEE IS'16, pp. 221–226, 2016.

[62] Y. Xiang, R. Jin, D. Fuhry, F. F. Dragan, "Summarizing Transactional Databases with Overlapped Hyperrectangles", *Data Mining and Knowledge Discovery* 23 (2011), 215–251.

[63] L. A. Zadeh, "Probability Measures of Fuzzy Events", *J. Math. Anal. Appl.* 23 (1968), 421–427.