On Suboptimality of GreConD for Boolean Matrix Factorisation of Contranominal Scales

Dmitry I. Ignatov[1,2] and Alexandra Yakovleva[1]

[1] National Research University Higher School of Economics, Moscow
[2] St. Petersburg Department of Steklov Mathematical Institute of Russian Academy of Sciences, Russia
dignatov@hse.ru,yakovlevalexandra@yandex.ru

Abstract. In this paper we study certain properties of the GreConD algorithm for Boolean matrix factorisation, a popular technique in Data Mining with binary relational data. This greedy algorithm was inspired by the fact that the optimal number of factors for the Boolean matrix factorisation can be chosen among the formal concepts of the corresponding formal context. In particular, we consider one of the hardest cases (in terms of the numerous of possible factors), the so-called contranominal scales, and show that the output of GreConD is not optimal in this case. Moreover, we formally analyse its output by means of recurrences and generating functions and provide the reader with the closed form for the returned number of factors. An algorithm generating the optimal number of factors and the corresponding product matrices P and Q is also provided by us for the case of contranominal scales.

Keywords: Boolean Matrix Factorisation, Formal Concept Analysis, Schein rank, generating functions, greedy algorithms

1 Introduction

Boolean data analysis and Formal Concept Analysis are closely related [1]. For example, Boolean matrices describing binary relations can be considered as formal contexts and vice versa, and decomposition of Boolean matrices into the product of two Boolean matrices of possibly smaller sizes is one of such crossroads where two disciplines meet each other. Thus, it was shown that the optimal number of factors, that is the minimal size of common dimension of these two product matrices, can be found based on the family of corresponding formal concepts considered as factors for the original Boolean matrix [2]. Decomposition of object-attribute matrices into products of object-factor and factor-attribute matrices plays important role in Machine Learning and Data Mining [3]. One of the desired properties is the dimensional reduction that normally preserves with high accuracy similarly between objects or attributes in terms of dot product and makes it possible to recover the input matrix [4]. For example, in collaborative filtering domain Boolean Matrix Factorisation (BMF) was on par with the (truncated) Singular Value Decomposition approach in terms of obtained quality metrics [5,6]. It speeds up the computation on the decomposed matrices and allows finding homogeneous taste communities as those latent factors.

Another fruitful property of Boolean matrices is their cheap bit representation and related bit operations. The only obstacle for Boolean Matrix Factorisation to be widely adopted technique so far is that of determination of the optimal number factors k for Boolean matrices or Schein rank is NP-hard problem [7,2]. So, every good approximate algorithm geared towards minimisation of the number of factors can be taken into account [8].

One of the earlier proposed algorithm for BMF is GreConD. It follows a greedy strategy adding attributes one-by-one with subsequent computation of their closures and is not optimal in general. In this paper we address one very important for practice case of the input for this algorithm, the contranominal scale of arbitrary size n, i.e. square Boolean matrix with all ones except the main diagonal [9]. It is well-known that the number of patterns (formal concepts) for this case is 2^n. It is easy to show experimentally that GreConD is not optimal for this particular case by comparing its output with the theoretically deduced values of Schein rank for contranominal scales. However, the output solution follows an interesting pattern deserving a special treatment in terms of recurrences and generating functions. It allows us to formally analyse the discrepancy between this suboptimal solution and theoretically optimal one. Moreover, to know the theoretically optimal solution as the number of factors does not mean to provide a concrete factorisation. To fill the gap, we sketch a correct algorithm to this end.

The paper is organised as follows. In Section 2, we recall the reader the basic definitions of FCA and BMF and describe GreConD algorithm. In Section 3, we shortly describe GreConD with its pseudocode. In Section 4, we provide the reader with our experimental and theoretical analyses of the algorithm's suboptimality. The penultimate section, Section 5, presents the optimal algorithm to find BMF for formal contexts of contranominal scales. Finally, Section 6 briefly discusses future prospects and concludes the paper.

2 Boolean Matrix Factorisation and GreConD

2.1 BMF based on Formal Concept Analysis

Basic FCA definitions. Formal Concept Analysis (FCA) is a branch of applied algebra and it studies (formal) concepts and their hierarchies [10]. The adjective "formal" indicates a strict mathematical definition of a pair of sets, called, the extent and intent. This formalisation is possible because of the use of the algebraic lattice theory.

Definition 1. Formal context $\mathbb{K}$ *is a triple* (G, M, I)*, where* G *is a set of* objects*,* M *is a set of* attributes*, and* $I \subseteq G \times M$ *is an incidence binary relation.*

The binary relation I is interpreted as follows: for $g \in G$, $m \in M$ we write gIm if the object g has the attribute m.

For a formal context $\mathbb{K} = (G, M, I)$ and any $A \subseteq G$ and $B \subseteq M$ a pair of mappings is defined:

$$A^{\uparrow} = \{m \in M \mid gIm \text{ for all } g \in A\}, \quad B^{\downarrow} = \{g \in G \mid gIm \text{ for all } m \in B\},$$

these mappings define Galois connection between partially ordered sets $(2^G, \subseteq)$ and $(2^M, \subseteq)$ on disjunctive union of G and M. The set A is called *closed set*, if $A^{\uparrow\downarrow} = A$ [11].

Definition 2. *A* formal concept *of the formal context $\mathbb{K} = (G, M, I)$ is a pair (A, B), where $A \subseteq G$, $B \subseteq M$, $A^{\uparrow} = B$ and $B^{\downarrow} = A$. The set A is called the* extent*, and B is the* intent *of the formal concept (A, B).*

It is evident that the extent and intent of any formal concept are closed sets.

The set of all formal concepts of a context $\mathbb{K}$ is denoted by $\mathfrak{B}(G, M, I)$.

The state-of-the-art surveys on advances in FCA theory and its applications can be found in [12,13].

Description of FCA-based BMF. Boolean Matrix Factorisation is a decomposition of the original matrix $I \in \{0,1\}^{n \times m}$, where $I_{ij} \in \{0,1\}$, into a Boolean matrix product $P \circ Q$ of binary matrices $P \in \{0,1\}^{n \times k}$ and $Q \in \{0,1\}^{k \times m}$ for the smallest possible number of k. We define Boolean matrix product as follows:

$$(P \circ Q)_{ij} = \bigvee_{l=1}^{k} P_{il} \cdot Q_{lj},$$

where $\bigvee$ denotes disjunction, and $\cdot$ conjunction.

For example, in collaborative filtering, matrix I can be considered as a matrix of binary relation between set X of objects (users), and a set Y of attributes (items that users have evaluated). In this case, we assume that xIy iff the user x evaluated object y. The triple (X, Y, I) naturally forms a formal context.

Consider a set $\mathcal{F} \subseteq \mathcal{B}(X, Y, I)$, a subset of all formal concepts of context (X, Y, I), and introduce matrices $P_{\mathcal{F}}$ and $Q_{\mathcal{F}}$:

$$(P_{\mathcal{F}})_{il} = \begin{cases} 1, i \in A_l, \\ 0, i \notin A_l, \end{cases} \quad (Q_{\mathcal{F}})_{lj} = \begin{cases} 1, j \in B_l, \\ 0, j \notin B_l. \end{cases},$$

where (A_l, B_l) is a formal concept from $\mathcal{F}$. We can consider decomposition of the matrix I into binary matrix product $P_{\mathcal{F}}$ and $Q_{\mathcal{F}}$ as described above. The following theorems are proved in [2]:

> **Theorem 1.** *(Universality of formal concepts as factors). For every I there is $\mathcal{F} \subseteq \mathcal{B}(X, Y, I)$, such that $I = P_{\mathcal{F}} \circ Q_{\mathcal{F}}$.*
>
> **Theorem 2.** *(Optimality of formal concepts as factors). Let $I = P \circ Q$ for $n \times k$ and $k \times m$ binary matrices P and Q. Then there exists a $\mathcal{F} \subseteq \mathcal{B}(X, Y, I)$ of formal concepts of I such that $|\mathcal{F}| \leq k$ and for the $n \times |\mathcal{F}|$ and $|\mathcal{F}| \times m$ binary matrices $P_{\mathcal{F}}$ and $Q_{\mathcal{F}}$ we have $I = P_{\mathcal{F}} \circ Q_{\mathcal{F}}$.*

There are several algorithms for finding $P_{\mathcal{F}}$ and $Q_{\mathcal{F}}$ by calculating formal concepts based on these theorems [2].

3 GreConD

There are several algorithms for finding $P_{\mathcal{F}}$ and $Q_{\mathcal{F}}$ by calculating formal concepts based on aforementioned theorems [2]. This paper studies the work of GreConD (Algoritm 2 from [2]), one of the existing algorithms for BMF. GreConD avoids computation of all possible formal concepts and therefore works much faster [2]. Time estimation of the calculations in the worst case yields $O(k|G||M|^3)$ [5], where k is the number of found factors (and can be omitted as a constant term), $|G|$ is the number of objects, $|M|$ is the number of attributes.

Define $\mathcal{U} = \{\langle i, j\rangle | I_{i,j} = 1\}$ for a Boolean matrix I. The main idea of the algorithm is to maximize the set

$$D \oplus y := ((D \cup \{y\})^{\downarrow} \times (D \cup \{y\})^{\downarrow\uparrow}) \cap \mathcal{U}$$

successively adding columns to intent D of formal concept (C, D).

Below we provide pseudocode for GreConD.

Algorithm 3.1 GreConD

```
 1: INPUT: I (Boolean matrix)
 2: OUTPUT: $\mathcal{F}$ (set of factor concepts)
 3:
 4: $\mathcal{U} \leftarrow \{\langle i, j\rangle | I_{i,j} = 1\}$
 5: $\mathcal{F} \leftarrow \emptyset$
 6: while $\mathcal{U} \neq \emptyset$ do
 7:     $D \leftarrow \emptyset$
 8:     $V \leftarrow 0$
 9:     while there is $j \notin D$ such that $|D \oplus j| > V$ do
10:         select $j \notin D$ that maximizes $|D \oplus j|$
11:             $D \leftarrow (D \cup \{j\})^{\downarrow\uparrow}$
12:             $V \leftarrow |(D^{\downarrow} \times D) \cap \mathcal{U}|$
13:     end while
14:     $C \leftarrow D^{\downarrow}$
15:     add $(C, D)$ to $\mathcal{F}$
16:     for each $\langle i, j\rangle \in C \times D$ do
17:         remove $\langle i, j\rangle$ from $\mathcal{U}$
18:     end for
19: end while
20: return $\mathcal{F}$
```

The set $\mathcal{U}$ contains not yet covered object-attribute pairs by any of the previously found factors. When the newly found factor (C, D) is added to $\mathcal{F}$, all the pairs from $C \times D$ should be deleted from $\mathcal{U}$ (lines 15-18). When $\mathcal{U}$ is empty, the GreConD terminates (line 6, the main loop). The inner loop (lines 9–13) maximizes the cardinality $D \oplus j$ while it is still possible by examining attributes not in D.

4 GreConD on contranominal scale

In this section we show that GreConD is optimal for ordinal and nominal scales, but not optimal on contranominal scale. We also construct an optimal algorithm for contranominal scale.

4.1 Optimality on ordinal and nominal scales

In FCA, scales are used to represent the so-called multi-valued contexts (cf. relational tables in databases) as one-valued contexts; the latter we also consider here as Boolean matrices.

First, let us consider two elementary scales. The *nominal scale* is defined as a formal context $\mathbb{N}_n = (\{1, \ldots, n\}, \{1, \ldots, n\}, =)$ and is used to scale mutually exclusive attributes like traffic light signals (red, green, yellow). The *ordinal scale* is defined as $\mathbb{O}_n = (\{1, \ldots, n\}, \{1, \ldots, n\}, \leq)$ and is applied in cases where the values are ordered like university grades (poor, normal, good, excellent).

It follows from our experiment that the number of factors obtained by GreConD on ordinal and nominal scales are equal to the size of scales. We can prove that these numbers are optimal.

Proposition 1. *The number of factors n obtained by GreConD for a nominal scale $\mathbb{N}_n$ is optimal.*

Proof. Note that for a nominal scale of size n any concept with nonempty extent and intent has the form $(\{i\}, \{i\})$ $(i \in \{1, \ldots, n\})$. Furthermore, the number of formal concepts is equal to the number of factors by definition. □

Proposition 2. *The number of factors n obtained by GreConD for an ordinal $\mathbb{O}_n$ is optimal.*

Proof. Note that for the ordinal scale of size n and for any nonempty $A \subseteq \{max(A), \ldots, n\}$ it holds that $A^{\uparrow} = \{1, \ldots, n\}$. Besides, $\{max(A), \ldots, n\}^{\downarrow} = \{1, \ldots, max(A)\}$. Therefore, concepts for the ordinal scale are $(\{1, \ldots, k\}, \{k, \ldots, n\})$ for $k \in \{1, \ldots, n\}$. Since GreConD needs to cover every object-attribute pair, each pair $(\{i\}, \{i\})$ for $i \in \{1, \ldots, n\}$ should be covered as well, which requires exactly n concepts $(\{1, \ldots, i\}, \{i, \ldots, n\})$. □

4.2 Suboptimality on contranominal scale

For every set S the *contranominal scale* is defined as $\mathbb{N}_S^c = (S, S, \neq)$. In what follows, we consider $\mathbb{N}_n^c$ with $S = \{1, \ldots, n\}$ without loss of generality.

Factorizing contranominal scales of sizes from 1 to 128 by GreConD[3] we obtain a sequence of the number of factors a_n (n is the size of a scale):

$$a_1 = 0, \quad a_2 = 2, \quad a_3 = 3, \quad a_4 = 4,$$

[3] Our Python implementation of GreConD for these experiments: `https://bit.ly/GreConDsub`

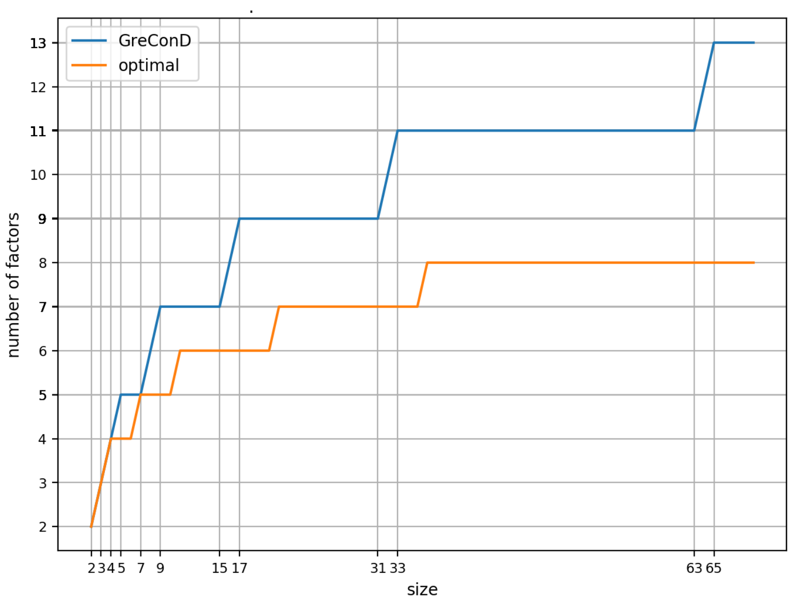

Fig. 1. The number of factors for GreConD and its theoretical optimum for contranominal scales of increasing size.

$$
\begin{gathered}
a_5 = a_6 = a_7 = 5, \quad a_8 = 6, \\
a_9 = \cdots = a_{15} = 7, \quad a_{16} = 8, \\
a_{17} = \cdots = a_{31} = 9, \quad a_{32} = 10, \\
a_{33} = \cdots = a_{63} = 11, \quad a_{64} = 12, \\
a_{65} = \cdots = a_{127} = 13, \quad a_{128} = 14 \ldots
\end{gathered}
$$

Note that the number of obtained factors increases by one when the size of a scale is a power of two or a power of two plus one.

The sequence can be defined as follows:

$$a_1 = 0, a_2 = 2,$$

$$a_n = 2 \log_2 n \text{ if } \exists k : n = 2^k,$$

$$a_n = a_{2^{\lfloor \log_2 n \rfloor}} + 1 \text{ for other } n.$$

Thus analytic form for the sequence is the following.

Conjecture. *The number of factors obtained by GreConD on contranominal scale is described by the sequence*

$$a_n = 2 \cdot \lfloor \log_2 n \rfloor + 1 - [n = 2^{\lfloor \log_2 n \rfloor}],$$

n being the size of a scale.

One way to obtain a simpler closed form of the considered sequence is to analyse its generating function [14].

Let $G(z) = \sum\limits_{n} a_n z^n$ is the associated generating function for the sequence a_n. The sequence a_n can be rewritten in the following way: $a_1 = 0$, $a_2 = 2$, while $a_n = a_{n-1} + \lfloor \log_2 n \rfloor - \lfloor \log_2 (n-1) \rfloor + \lceil \log_2 n \rceil - \lceil \log_2 (n-1) \rceil$. One can check that one of the respective differences of rounded logarithms takes on 1 when $n = 2^k$ or $n - 1 = 2^k$ for some $k > 0$.

Let us sum $a_n z^n$ as follows:

$$\sum_{n\geq 2} a_n z^n = \sum_{n\geq 2} a_{n-1} z^n + \sum_{n\geq 2} (\lfloor \log_2 n \rfloor - \lfloor \log_2 (n-1) \rfloor + \lceil \log_2 n \rceil - \lceil \log_2 (n-1) \rceil) z^n$$

Let $U_n = \lceil log_2 n \rceil$ and $L_n = \lfloor log_2 n \rfloor$, then

$$G(z) = zG(z) + \sum_{n\geq 2} L_n z^n - \sum_{n\geq 2} L_{n-1} z^n + \sum_{n\geq 2} U_n z^n - \sum_{n\geq 2} U_{n-1} z^n \ .$$

Now, let $L(z) = \sum\limits_{n\geq 2} L_n z^n$ and $U(z) \sum\limits_{n\geq 2} U_n z^n$, then

$$G(z) = zG(z) + L(z) - zL(z) + U(z) - zU(z) \text{ or}$$

$$G(z)(1 - z) = (L(z) + U(z))(1 - z) \ .$$

For $z \neq 1$ we have

$$a_n = [z^n]G(z) = \lfloor \log_2 n \rfloor + \lceil \log_2 n \rceil \ .$$

Next, we show that the number of factors obtained by GreConD on contranominal scale is not optimal.

First, we provide the definition of Schein rank.

Definition 3. *[15] For vectors v, w the matrix $(v_i w_j)$ is called cross-vector*[4].

Definition 4. *[15] Schein rank of a Boolean matrix A is the least number of Boolean cross-vectors summing up to A.*

Theorem 3. *[16] Schein rank of contranominal scale of size n equals $N(n)$, where $l = N(k)$ $(k \in \mathbb{N})$ is defined as the least number, such that $k \leq \binom{l}{\lfloor l/2 \rfloor}$.*

We also provide several first values of $N(n)$[5]:
$N(1) = 1, N(2) = 2, N(3) = 3, N(4) = N(5) = N(6) = 4, N(7) = \cdots = N(10) =$

[4] Note that we deal with column vectors according to data analysis conventions; so, $(v_i w_j)$ is the outer product of v and w.

[5] See also OEIS sequence A305233: `https://oeis.org/A305233`

5,
$N(11) = \cdots = N(20) = 6, N(21) = \cdots = N(35) = 7, N(36) = \cdots = N(70) = 8,$
$N(71) = \cdots = N(126) = 9, N(127) = \cdots = N(252) = 10 \ldots$
Note that for contranominal scales of sizes $2, 3, 4, 7$ GreConD does find Schein rank, i.e. optimal number of factors. However, for the remaining sizes ($n > 1$) GreConD finds suboptimal number of factors.

5 Optimal algorithm for contranominal scale

Let us construct an algorithm that would factorize contranominal scale with optimal number of factors. We use Sperner's theorem.

Definition 5. *A family of incomparable (with respect to set inclusion) sets is called a Sperner family, or an antichain of sets.*

Theorem 4. *[17] (Sperner) For an n-element set the size of a largest antichain does not exceed $\binom{n}{\lfloor n/2 \rfloor}$.*
Equality holds iff an antichain consists of all subsets of size $\lceil n/2 \rceil$ or all subsets of size $\lfloor n/2 \rfloor$.

Theorem 3 [16] states that the optimal number of factors for contranominal scale of size n is equal to $N(n)$. Therefore, BMF with the optimal number of factors (we call it *optimal* BMF) has object-factor matrix of size $n \times N(n)$. From the proof [16] it follows that the minimal set of factors for contranominal scale is an antichain. Next, we show how to find an antichain of a given length n.

Let us find all combinations of elements from the set $\{1, \ldots, N(n)\}$ by $\lfloor N(n)/2 \rfloor$ elements (for example, by Algorithm T from [18][p. 359]). Note that by Sperner's theorem a set of those combinations is the largest antichain for the $N(n)$-element set of factors. Also, $\binom{N(n)}{\lfloor N(n)/2 \rfloor} \geq n$ by definition of $N(n)$. Next, for every combination we make a binary vector r of length $N(n)$ with $r_i = 1 \iff$ the corresponding combination contains the element i. Finally, we obtain object-factor matrix by choosing an n-element subset of binary vectors and placing it in object-factor matrix.

Based on the constructed object-factor matrix, we find the factor-attribute matrix. We apply the derivation operator $^{\downarrow}$ to every factor f in the object-factor matrix, then we apply the derivation operator operator $^{\uparrow}$ to the set of the obtained objects in object-attribute matrix. Finally, we make binary row for the obtained set of attributes and place it in f-row in factor-attribute matrix.

Thus, we get optimal BMF for contranominal scale.

Note that we can simplify the procedure of construction of the factor-attribute matrix using the following property.

Property 1. For contranominal scale of size n and any subsets A and B of sets of objects and attributes respectively it holds that

$$A^{\uparrow} = \{1, \ldots, n\} \backslash A; B^{\downarrow} = \{1, \ldots, n\} \backslash B.$$

Proof. Using the definition of contranominal scale and the derivation operator(s) we get:
$A^{\uparrow} = \cap_{a \in A}(\{1, \ldots, n\} \backslash a) = \{1, \ldots, n\} \backslash A.$
The proof for B is similar. □

Now we can remake the recovering of factor-attribute B matrix from object-factor matrix A. If A_k is the k-th column in the matrix A, then $\sim A_k$ (here $\sim$ is a logical negation) is a k-th row in the matrix B.

Example. Let us demonstrate our algorithm on contranominal scale of size 5.

$N(5) = 4$, hence BMF has 4 factors. Generate 5 different combinations from the set $\{1, 2, 3, 4\}$: $\{1, 2\}, \{1, 3\}, \{1, 4\}, \{2, 3\}, \{2, 4\}$. Therefore, we obtain the object-factor matrix A:
$A = \begin{pmatrix} 1\,1\,0\,0 \\ 1\,0\,1\,0 \\ 1\,0\,0\,1 \\ 0\,1\,1\,0 \\ 0\,1\,0\,1 \end{pmatrix}$. The first column of matrix A consists of vector $\big(1, 1, 1, 0, 0\big)^T$, so vector $\big(0, 0, 0, 1, 1\big)$ is the first row of factor-attribute matrix. Similarly, we fill the rest of the rows in matrix B and obtain the optimal BMF:

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

Proposition 3. *The number of optimal BMFs (found by the proposed algorithm) of the contranominal scale of size $n > 0$ is $n!\binom{q}{n}$, where $q = \binom{N(n)}{\lfloor N(n)/2 \rfloor}$.*

Proof. Recall that we choose an n-element set from all the combinations of numbers from the set $\{1, \ldots, N(n)\}$ by $\lfloor N(n)/2 \rfloor$ elements in order to get rows of an object-factor matrix. Further, there are $n!$ ways to arrange every obtained n-element set of combinations as rows of object-factor matrix.

We conclude the proof noting that there is a unique way to build the factor-attribute matrix having the object-factor matrix. □

Note that the case $n = 1$, i.e. when $I = (0)$, has two more solutions in addition to $P \circ Q = (1) \circ (0)$; namely, $(0) \circ (1) = (0) \circ (0)$.

6 Conclusion

In the paper we considered important case for Boolean matrix factorisation based on our experimental and theoretical analyses of the behaviour of the GreConD algorithm. We hypothesise that the number of output factors for the contranominal scales in case of GreConD is $\lfloor \log_2 n \rfloor + \lceil \log_2 n \rceil$ based on the substantial observed fragment of its output for different values of the scale size n.

We have also proposed an optimal algorithm w.r.t. Schein rank to find one out of $n!\binom{q}{n}$ optimal Boolean matrix factorisation for this case, where $q = \binom{N(n)}{\lfloor N(n)/2 \rfloor}$.

As a future research direction we would like to continue our previous investigations of Boolean matrix factorisation for collaborative filtering problems [5,6] with an updated knowledge on suboptimality in case of contranominal scales presence as well to extend this approach to Boolean tensors.

Acknowledgments. The paper was prepared within the framework of the HSE University Basic Research Program and was also supported in part through computational resources of HPC facilities at HSE University. The first author was also supported by Russian Science Foundation under grant 17-11-01276 at St. Petersburg Department of Steklov Mathematical Institute of Russian Academy of Sciences, Russia and by RFBR (Russian Foundation for Basic Research) according to the research project No 19-29-01151. The foundations had no role in study design, data collection and analysis, writing the manuscript, and decision to publish.

We would like to thank Profs. Radim Belohlavek, Jan Outrata, Martin Trnecka, and Vilem Vychodil for lasting collaboration and Prof. Donald Knuth for personal written explanation of a nontrivial piece from Concrete Mathematics [14] regarding summation properties.

References

1. Janostik, R., Konecny, J., Krajca, P.: Interface between Logical Analysis of Data and Formal Concept Analysis. Eur. J. Oper. Res. **284**(2) (2020) 792–800
2. Belohlavek, R., Vychodil, V.: Discovery of optimal factors in binary data via a novel method of matrix decomposition. Journal of Computer and System Sciences **76**(1) (2010) 3 – 20 Special Issue on Intelligent Data Analysis.
3. Miettinen, P., Neumann, S.: Recent Developments in Boolean Matrix Factorization. In Bessiere, C., ed.: Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI 2020, ijcai.org (2020) 4922–4928
4. Belohlávek, R., Outrata, J., Trnecka, M.: Impact of Boolean factorization as preprocessing methods for classification of Boolean data. Ann. Math. Artif. Intell. **72**(1-2) (2014) 3–22
5. Ignatov, D.I., Nenova, E., Konstantinova, N., Konstantinov, A.V.: Boolean Matrix Factorisation for Collaborative Filtering: An FCA-Based Approach. In Agre, G., Hitzler, P., Krisnadhi, A.A., Kuznetsov, S.O., eds.: Artificial Intelligence: Methodology, Systems, and Applications - 16th International Conference, AIMSA 2014, Varna, Bulgaria, September 11-13, 2014. Proceedings. Volume 8722 of Lecture Notes in Computer Science., Springer (2014) 47–58
6. Akhmatnurov, M., Ignatov, D.I.: Context-Aware Recommender System Based on Boolean Matrix Factorisation. In Yahia, S.B., Konecny, J., eds.: Proceedings of the Twelfth International Conference on Concept Lattices and Their Applications, Clermont-Ferrand, France, October 13-16, 2015. Volume 1466 of CEUR Workshop Proceedings., CEUR-WS.org (2015) 99–110
7. Miettinen, P., Mielikäinen, T., Gionis, A., Das, G., Mannila, H.: The discrete basis problem. IEEE Trans. Knowl. Data Eng. **20**(10) (2008) 1348–1362

8. Miettinen, P., Vreeken, J.: MDL4BMF: Minimum Description Length for Boolean Matrix Factorization. ACM Trans. Knowl. Discov. Data **8**(4) (2014) 18:1–18:31
9. Albano, A., Chornomaz, B.: Why concept lattices are large: extremal theory for generators, concepts, and VC-dimension. Int. J. Gen. Syst. **46**(5) (2017) 440–457
10. Ganter, B., Wille, R.: Formal Concept Analysis: Mathematical Foundations. Springer, Berlin/Heidelberg (1999)
11. Birkhoff, G.: Lattice Theory. 11th printing edn. Harvard University, Cambridge, MA (2011)
12. Poelmans, J., Ignatov, D.I., Kuznetsov, S.O., Dedene, G.: Formal concept analysis in knowledge processing: A survey on applications. Expert Syst. Appl. **40**(16) (2013) 6538–6560
13. Poelmans, J., Kuznetsov, S.O., Ignatov, D.I., Dedene, G.: Formal concept analysis in knowledge processing: A survey on models and techniques. Expert Syst. Appl. **40**(16) (2013) 6601–6623
14. Graham, R.L., Knuth, D.E., Patashnik, O.: Concrete Mathematics: A Foundation for Computer Science, 2nd Ed. Addison-Wesley (1994)
15. Kim, K.H.: Boolean matrix theory and applications. Marcel Dekker, New York and Basel (1982)
16. Marenich, E.: Determining the Schein Rank of Boolean Matrices. Matrix Methods: Theory, Algorithms and Applications (2010) 85–103
17. Sperner, E.: Ein Satz über Untermengen einer endlichen Menge. Math Z **27** (1928) 544–548
18. Knuth, D.E.: Combinatorial Algorithms. Volume 4A of The Art of Computer Programming. Addison-Wesley Professional (January 2011)