Handling Noise in Boolean Matrix Factorization*

Radim Belohlavek and Martin Trnecka
Dept. of Computer Science, Palacký University Olomouc, Czech Repiblic
radim.belohlavek@acm.com, martin.trnecka@gmail.com

Abstract

We critically examine and point out weaknesses of the existing considerations in Boolean matrix factorization (BMF) regarding noise and the algorithms' ability to deal with noise. We argue that the current understanding is underdeveloped and that the current approaches are missing an important aspect. We provide a new, quantitative way to assess the ability of an algorithm to handle noise. Our approach is based on a common-sense definition of robustness requiring that the computed factorizations should not be affected much by varying the noise in data. We present an experimental evaluation of several existing algorithms and compare the results to the observations available in the literature. In addition to providing justification of some properties claimed in the literature without proper justification, our experiments reveal properties which were not reported as well as properties which counter certain claims made in the literature. Importantly, our approach reveals a line separating robust-to-noise from sensitive-to-noise algorithms, which has not been revealed by the previous approaches.

1 Introduction

1.1 Basic Concepts and Notation

We denote an $n \times m$ Boolean matrix by M and interpret it primarily as an object-attribute matrix. That is, M_{ij} indicates that the object i does ($M_{ij} = 1$) or does not have ($M_{ij} = 0$) the attribute j, respectively. The set of all $n \times m$ Boolean matrices is denoted by $\{0,1\}^{n\times m}$. The ith row and jth column vector of M is denoted by $M_{i_}$ and $M_{_j}$, respectively.

The general problem in BMF is to find for a given $M \in \{0,1\}^{n\times m}$ matrices $A \in \{0,1\}^{n\times k}$ and $B \in \{0,1\}^{k\times m}$ for which

$$M \text{ (approximately) equals } A \circ B, \tag{1}$$

where

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

The least k for which an exact decomposition $M = A \circ B$ exists is called the *Boolean rank* (Schein rank) of M.

The approximate equality in (1) is commonly assessed in BMF by means of the L_1-norm $||\cdot||$ and the corresponding metric $E(\cdot,\cdot)$, defined for $C, D \in \{0,1\}^{n\times m}$ by

$$||C|| = \sum_{i,j=1}^{m,n} |C_{ij}|, \tag{2}$$

$$E(C,D) = ||C - D|| = \sum_{i,j=1}^{m,n} |C_{ij} - D_{ij}|. \tag{3}$$

Two important variants of the factorization problem, emphasizing the role of the first k factors and the need to account for a prescribed portion of data, respectively, are known as the discrete basis problem (DBP) and the approximate factorization problem (AFP); see [Miettinen, 2009; Miettinen *et al.*, 2008] and [Belohlavek and Vychodil, 2009; Belohlavek and Trnecka, 2015].

Note also an important geometric view of BMF: A decomposition $M = A \circ B$ using k factors essentially means a coverage of the 1s in M by k rectangular areas in M that are full of 1s, the lth rectangle being the crossproduct of the lth column in A and the lth row in B; see e.g. [Kim, 1982].

If for an approximate decomposition $M \approx A \circ B$, $M_{ij} = 1$ and $(A \circ B)_{ij} = 0$, one speaks of *uncovering* (or false negative) in entry $\langle i, j\rangle$; if $M_{ij} = 0$ and $(A \circ B)_{ij} = 1$, one speaks of *overcovering* (or false positive).

For the assessment of quality of decompositions one usually employs the error $E(M, A \circ B)$, see (3), or some variant of it. We utilize the *coverage quality* [Belohlavek and Vychodil, 2009; Belohlavek and Trnecka, 2015] of the first l factors delivered by a particular algorithm, which is a function of $A \in \{0,1\}^{n\times l}$ and $B \in \{0,1\}^{l\times m}$ defined by

$$c(l) = 1 - E(M, A \circ B)/||M||, \tag{4}$$

but as long as error is the main concern, the sole $E(M, A\circ B)$ or other variant of it may be used.

1.2 Relevant Work in BMF

Due to limited scope, we focus on the directly relevant work only. Perhaps the first works on applications of BMF in data analysis are [Nau, 1976; Nau *et al.*, 1978], in which the authors have already been aware of NP-hardness of the basic decomposition problem due to NP-hardness of the set basis problem [Stockmeyer, 1975]. An early but currently virtually unknown is the 8M algorithm [Dixon, 1992] which we employ in our experiments below. Interest in BMF in current

*Supported by grant No. GA15-17899S of the Czech Science Foundation.

data mining is primarily due to the work of Miettinen et al. In particular, the DBP, the corresponding complexity results, and the ASSO algorithm discussed below appeared in [Miettinen, 2009; Miettinen *et al.*, 2008]. The ASSO algorithm is currently widely known and is often being used for comparison purposes when a new algorithm is devised. Little earlier, an important paper [Geerts *et al.*, 2004] appeared in which the authors presented the TILING algorithm for a problem (tiling problem) which upon easy reformulation is basically the problem of finding a decomposition of a Boolean matrix in which no overcovering is allowed. A much more efficient BMF algorithm, GRECOND, which does not commit overcovering, was presented in [Belohlavek and Vychodil, 2009] under the name Algorithm 2; the name GRECOND is used in [Belohlavek and Trnecka, 2015]. In [Belohlavek and Trnecka, 2015], the GREESS algorithm is proposed, which is based on a new theoretical insight regarding BMF. Unlike ASSO, which is designed for DBP, GRECOND and GREESS are primarily designed for the AFP problem, and so is TILING from the present perspective. HYPER [Xiang *et al.*, 2011] is another BMF algorithm which does not commit overcovering; its extension, HYPER+, based on a certain postprocessing of the HYPER results, computes general decompositions (i.e. those which may commit bot under- and overcovering). The PANDA algorithm [Lucchese *et al.*, 2010] is directly motivated by the problem of noise and we employ it in our experiments. [Lucchese *et al.*, 2014] presents PANDA+, an enhanced version of PANDA. [Karaev *et al.*, 2015] presents NASSAU, an algorithm principle whose aim is robustness w.r.t. a particular kind of noise.

The problem of noise in Boolean data has perhaps for the first time been discussed in the context of frequent itemset mining, see e.g. [Gupta *et al.*, 2008] and the references therein. Since [Miettinen, 2009; Miettinen *et al.*, 2008], considerations on noise became part of BMF research. The aim, implicitly or explicitly stated in many subsequent studies, is to devise algorithms able to handle noise. We return to this problem in detail below. Let us also mention at this point that the minimum description length (MDL) principle is employed in various ways as a mean to achieve this ability in some of these studies [Lucchese *et al.*, 2010; 2014; Karaev *et al.*, 2015], but also for other purposes such as identification of a reasonable number of factors explaining the data [Miettinen and Vreeken, 2011; 2014].

2 Noise in BMF—A Critical Examination

2.1 What is Noise in Boolean Data?

In the current BMF research, noise in Boolean data basically represents distortion of data, i.e. flipping some data entries of true data. That is, a given (observed) data represented by a matrix M, may be different from the true data M^{t}. Two kinds of noise are distinguished, the so-called *additive noise*, due to which some 0s in M^{t} are flipped to 1s, and *subtractive noise*, due to which 1s are flipped to 0s; if both kinds of flips occur, one speaks of noise or of *general noise.*

Considerations regarding noise in BMF go back to [Miettinen, 2009; Miettinen *et al.*, 2008] where noise has been used to generate synthetic datasets: To add $p\%$ of additive noise to a given matrix M means that the entries in M are changed to 1 randomly with probability $p\%$ (other options are possible, but we use this one); similarly for subtractive and general noise.

2.2 Is Noise Always a Reasonable Assumption?

Note first that the above usage of the term "noise" may seem strange. Namely, while noise is generally understood as representing random and mostly small fluctuations in data, for Boolean data, a small fluctuation in value is not possible: If an entry M_{ij} is to change, it has to change "completely," i.e. from truth to falsity of from falsity to truth. From this perspective, a possibly more appropriate term would be "error" (which may also bear unwanted connotations) or "lie" (which is used in logic in this context). Nevertheless, more important than terminology are considerations regarding the possible origin of noise, amount of noise, and its significance for real data, which we briefly mention now.

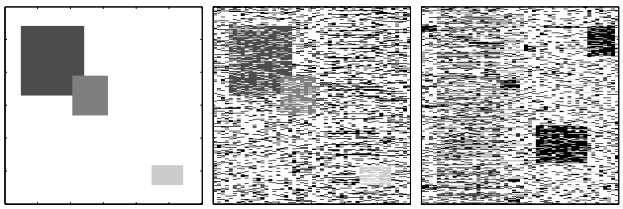
Even though we take it for granted that presence of noise in Boolean data is often a reasonable assumption, we need to point out that many real datasets do not contain noise because they simply contain verified truth. In addition, there exist applications of BMF, in which presence of noise would be counterintuitive or even damaging—a generic example is a scenario similar to that of the role mining problem [Lu *et al.*, 2008; Vaidya *et al.*, 2007] in which the data represents users and their security permissions. Therefore, contrary to the reasoning in some recent literature, one should not be concluding that algorithms that are not robust to noise are deficient: They simply may be appropriate for a different, but realistic and important scenario which is noise free. In fact, while real-case studies of the noise-free scenario are available in the literature, real-case studies in which noise appears are, by and large, missing. Consequently, several questions have not been addressed yet.

Important among them is the question of *which levels of noise are realistic.* We contend that levels such as 40% or higher, which appear in the BMF literature, are too high to be realistic. Drawn from our own experience, errors in Boolean data are typically up to 5%. Unless real-case situations with higher levels of noise are existing, explorations of such levels seems to not to be well-grounded.

In any case, higher noise levels present a problem which has not been discussed to our knowledge so far, namely they *radically distort the data*, even to the extent that new significant factors may appear in it while the original factors may disappear. This is illustrated in Fig. 1. To ask a BMF algorithm to extract the original factors would therefore be wrong. The phenomenon at stake should therefore carefully be taken into account.

2.3 A Rationale for Robustness to Noise

We now provide an explicit formulation of a rationale for robustness of BMF algorithms to noise, which is the main problem of this paper. We claim that this rationale grasps well the expectations implicitly present in the existing studies. Consider matrices M in Fig. 2a and N in Fig. 2b. In the observed data M, one clearly recognizes three rectangles, the union of which forms the gray area in Fig. 2a, even though some of



(a) Original data (b) Noise added (c) Permuting (b)

Figure 1: Data in (b) results by adding 40% noise to (a); white represents 0s, black/gray represents 1s. The original three factors in (a) are no longer significant; new factors explain the noisy data. These are apparent from (c) which contains data (b) with appropriately permuted rows and columns.

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

(a) Observed data

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

(b) Hypothetical true data

Figure 2: Three factors of hypothetical true data explaining observed data

the entries inside the area contain 0 rather than 1. A natural view is that M results from the true data, represented by N, by error (noise). From this viewpoint, one is interested in discovering from the observed data M the three factors behind the true data N, i.e. discovering from M the 10×3 and 3×10 matrices A and B for which $A \circ B = N$.

Note now that since the gray areas contain 0s, the three factors commit overcovering. Consequently, none of the algorithms which do not commit overcovering (TILING, GRECOND, HYPER, GREESS) is able to discover these factors. This is the basic reason why such algorithms have limited capability in recovering factors when subtractive noise is added. On the other hand, the algorithms committing both under- and overcovering do not suffer from this restriction. Indeed, while for the data in Fig. 2b, all the algorithms considered here compute the three factors, the situation is very different for Fig. 2a: TILING, GRECOND, HYPER, and GREESS yield, respectively, 9, 9, 10, and 9 rather small and fragmented factors to explain the whole data. On the other hand 8M, ASSO, PANDA, HYPER+, NASSAU, and GRECOND+ yield, respectively, 3, 3, 5, 3, 3, and 3 factors, which is indicative of the algorithms' ability to handle noise which we exploit below.

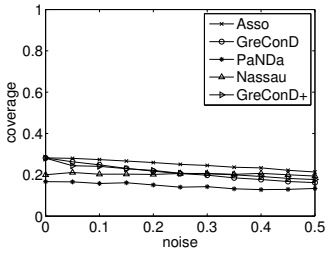


Figure 3: Current experiments to assess robustness to noise

2.4 A Critique of Current Approach to Assess Robustness

The present experiments aimed to evaluate robustness to noise, e.g. [Lucchese *et al.*, 2010; Miettinen, 2009; Miettinen *et al.*, 2008], result in graphs like the one in Fig. 3: It shows results of selected algorithms on a dataset of size 500×250 and density 15% which was obtained from randomly generated factors. Additive noise has been added with increasing noise levels up to 50%. The algorithms have been asked for the first $k = 5$ factors and the corresponding coverage qualities $c(k)$, see (4), of the k factors are plotted in the graphs.

What we see from the graphs is that with increasing noise, coverage quality of a particular algorithm generally decreases. It is understood in the literature that a small decrease in coverage quality indicates robustness to noise.

In fact, this is not the case. Namely, such view ignores the possibility that with more noise not only did the coverage drop, but the factors themselves may have changed substantially and this would clearly mean lack of robustness. As we shall see, this is what is indeed happening with various algorithms. Yet, Fig. 3 does not tell whether the factors changed, hence it does not characterize robustness to noise properly.

In addition, the graphs do not tell us whether a particular algorithm found the factors used to generate the data, i.e. whether it found the ground truth. Interestingly, our new experiments below show that an algorithm may be robust to noise in that the factors computed do not change much when noise is added, yet it may not have a good capability of discovering ground truth, and vice versa.

One therefore faces three possibly *distinct properties of a particular algorithm*: good *coverage quality* (according to DBP or AFP view), *robustness to noise*, and the *ability to discover ground truth.* From the above perspective, the current approaches have serious limitations in assessing these properties.

3 New Way to Assess Robustness to Noise

We now propose a new experimental scenario to assess robustness to noise of a given BMF algorithm. As we show in Section 4, it has the required capability to discriminate the properties of algorithms mentioned above. The basic idea is inspired by the above intuitive understanding of robustness (Section 2.3) and is the following. Suppose a BMF algorithm computes from a given matrix M a set $\mathcal{F}$ of factors and the corresponding matrices A and B. Suppose we change M slightly, by introducing noise, to a new matrix, M', and let

the algorithm compute, this time from M', a new set of factors, $\mathcal{F}'$, and the corresponding matrices A' and B'. We ask whether and to what extent are $\mathcal{F}$ and $\mathcal{F}'$ similar.

3.1 Similarity of Factorizations

To assess similarity of the sets $\mathcal{F}$ and $\mathcal{F}'$ of factors, we propose the measure $Sim(\mathcal{F}, \mathcal{F}')$ described below. By experiments on small datasets, we observed that the measure reasonably captures similarity between two sets of factors. Analogous measures, employing alternative formulas for $Sim(\mathcal{F}, \mathcal{F}')$ are possible but led to similar conclusions regarding robustness of algorithms.

The similarity measure $Sim(\mathcal{F}, \mathcal{F}')$ of two sets of factors, $\mathcal{F}$ and $\mathcal{F}'$, is computed as follows. Recall from Section 1.1 that each factor computed from $M \in \{0,1\}^{n\times m}$ corresponds to a rectangular area in M and may hence be viewed as the Cartesian product $C \times D$ for some $C \subseteq \{1,\ldots,n\}$ and $D \subseteq \{1,\ldots,m\}$. We may then define a similarity $s(\langle C_1, D_1\rangle, \langle C_2, D_2\rangle)$ of two factors as the well-known Jaccard index of their corresponding rectangles $C_1 \times D_1$ and $C_2 \times D_2$, i.e. as

$$s(\langle C_1, D_1\rangle, \langle C_2, D_2\rangle) = \frac{|C_1 \times D_1 \cap C_2 \times D_2|}{|C_1 \times D_1 \cup C_2 \times D_2|}.$$

Finally, we put $Sim(\mathcal{F}, \mathcal{F}')$ equal to

$$\min\left(\frac{\sum_{c\in\mathcal{F}} \max_{c'\in\mathcal{F}'} s(c,c')}{|\mathcal{F}|}, \frac{\sum_{c'\in\mathcal{F}'} \max_{c\in\mathcal{F}} s(c,c')}{|\mathcal{F}'|}\right).$$

That is, $Sim(\mathcal{F}, \mathcal{F}')$ may be regarded as the minimum of two numbers, one expressing the average similarity of a factor in $\mathcal{F}$ and its most similar factor in $\mathcal{F}'$, the other expressing the symmetric value with $\mathcal{F}$ and $\mathcal{F}'$ exchanged. Clearly, $Sim(\mathcal{F}, \mathcal{F}')$ ranges from 0 to 1, with higher values indicating higher similarity, and is easy to compute.

3.2 Assessing the Ability to Discover Ground Truth

Interestingly, the proposed measure $Sim(\mathcal{F}, \mathcal{F}')$ may be used to assess an algorithm's capability to discover ground truth in the sense described above: One just takes for $\mathcal{F}$ the original factors used to generate the input data M (i.e. the ground truth) and for $\mathcal{F}'$ the set of factors delivered by a given algorithm. The value $Sim(\mathcal{F}, \mathcal{F}')$—the similarity between the original and the discovered factors—then indicates the capability to discover ground truth.

4 Experimental Evaluation

Due to limited scope, we restrict to selected experimental results regarding robustness to noise and the ability to recover ground truth. These are representative of the larger set of experiments which we conducted.

4.1 Algorithms and Data

We used the following algorithms (see Section 1.2): 8M, TILING, ASSO, GRECOND, HYPER, PANDA, GREESS, and NASSAU. We decided not to include the results for HYPER+ because it basically performs a postprocessing of HYPER's output and because a similar postprocessing could be applied to other algorithms; note, however, that the results for HYPER+ correspond to those for HYPER as regards the trends observed. We also include GRECOND+, a simple extension of GRECOND which essentially consists in adding in a greedy manner to each factor computed as in GRECOND the columns and rows which possibly contain 0s and in certain straightforward modification of the previously computed factors.

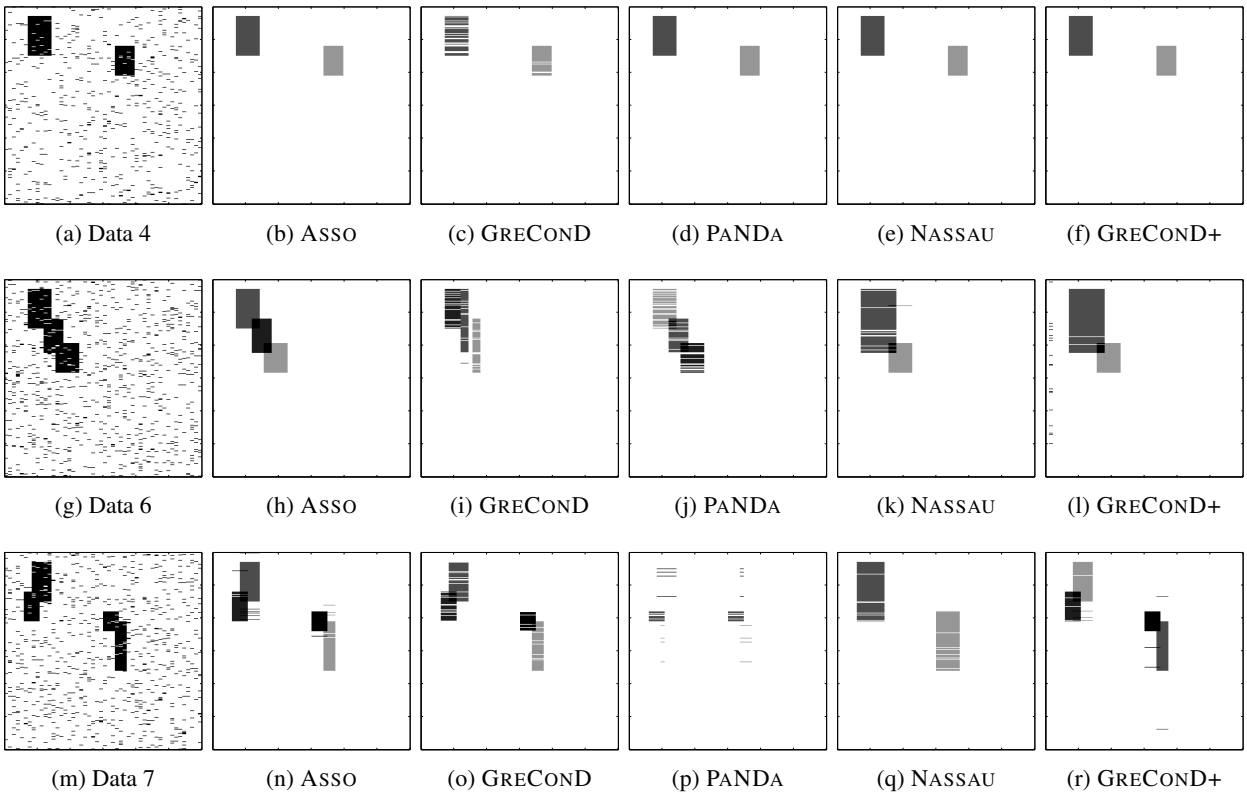
We used data commonly used in BMF experiments, both real and synthetic. Due to limited space, we only report results for synthetic data matrices M of size 500×250 obtained as Boolean products $A \circ B$ of $500 \times k$ and $k \times 250$ randomly generated matrices A and B for varying k with density of M (percentage of 1s) around 15%. As a representative of real data, we present results for the 231×79 Domino dataset (see e.g. [Ene *et al.*, 2008]) which has a 10% density.

4.2 Revisiting Datasets from Gupta at al.'s Paper

Before presenting our main experiments, we briefly report results on the 1000×50 datasets from [Gupta *et al.*, 2008] with noise added, because our observations are different from those reported by the authors of PANDA in [Lucchese *et al.*, 2010], who reported that of the available BMF algorithms, only PANDA is able to discover in their example from [Gupta *et al.*, 2008] the original factors when noise is added. The results we obtained on similar datasets are illustrated on data 4, 6, and 7 from [Gupta *et al.*, 2008] in Fig. 4. For one, we may observe that ASSO, which was explicitly included in [Lucchese *et al.*, 2010], performs very well and in fact better than PANDA, which suffers on data 7. Taking into account this and other experiments, we conclude that PANDA's overall performance is not as good as reported in [Lucchese *et al.*, 2010]; for coverage quality, see also [Belohlavek and Trnecka, 2015]. Secondly, we see that all the selected algorithms behave reasonably well, including to some extent GRECOND, which we selected as a representative of algorithms that do not commit overcovering. Third, the figure illustrates NASSAU's tendency to output larger factors (rectangles) which result by merging the original, smaller factors.

4.3 Robustness to Noise

Table 1 presents the results of experiments conducted according to Section 3. 1000 factorizations for each noise level added to the Domino data were computed for each algorithm. The table contains the values of the similarity measure $Sim(\mathcal{F}, \mathcal{F}')$ where $\mathcal{F}$ is the set of the first k factors obtained by the algorithm for Domino with no noise added while $\mathcal{F}'$ is the set of the first k factors obtained when the respective noise level was added to Domino; the number k varies. The experiment clearly separates TILING, GRECOND, HYPER, and GREESS—the algorithms which do not commit overcovering—from the other algorithms that may produce general factorizations. These four algorithms obtain very low scores, which corresponds well to the intuition demonstrated above: Adding noise leads to a fragmentation of tiles existing in the data and thus to a fragmentation of the factors computed by these algorithms. All the algorithms which may commit overcovering behave reasonably well from this point



(a) Data 4 (b) ASSO (c) GRECOND (d) PANDA (e) NASSAU (f) GRECOND+

(g) Data 6 (h) ASSO (i) GRECOND (j) PANDA (k) NASSAU (l) GRECOND+

(m) Data 7 (n) ASSO (o) GRECOND (p) PANDA (q) NASSAU (r) GRECOND+

Figure 4: Experiments on data 4, 6, and 7 from [Gupta *et al.*, 2008]

of view, i.e. introducing noise does not affect the computed factors too much. This explicit separation of algorithms based on their robustness to noise is not present in the existing literature. Note that NASSAU exhibits greatest fluctuations, which is partly due to the randomization involved in this algorithm. Note also that greater scores for larger k are partly due to the general tendency of the measure Sim to yield larger values for large sets of factors, partly due to the fact that the algorithms indeed tend to produce factors similar to those they produced for the data without the noise when one takes into account more and more factors (larger k).

4.4 Recovery of Ground Truth

In this section, we briefly present experiments demonstrating the ability to recover ground truth, as described in Section 3.2. Table 2 has a meaning analogous to that of Table 1 except that a synthetic 500×250 dataset generated by $k = 5$ known factors has been used and that in the values of $Sim(\mathcal{F}, \mathcal{F}')$, $\mathcal{F}$ is the set of the 5 factors used to generate the dataset and $\mathcal{F}'$ is the set of the first 5 factors obtained by the respective algorithm. Again, 1000 iterations for each level of noise have been performed. We depict the results for additive and subtractive noise. The following may be observed. Even the algorithms that do not commit overcovering except for HYPER achieve a reasonable ability to recover the original factors, but this ability deteriorates rapidly with increasing noise. HYPER's behavior is due to its preference of narrow, column-like factors. The algorithms designed to be robust to noise perform better, with ASSO and GRECOND+ obtaining the best scores. NASSAU exhibits an interesting behavior particularly for additive noise, which is worth further analysis.

5 Conclusions

We observed that the intuitive requirements regarding BMF algorithms' robustness to noise are not properly assessed by the current experiments. Consequently, we proposed a new way to assess robustness. We demonstrated that robustness to noise, the ability to recover ground truth, and the coverage quality are three different aspects which need to be taken into account when assessing quality of BMF algorithms. We presented several kinds of experimental observations. Some of them confirm, now on a clear methodological ground, intuitive expectations, regarding the existing algorithms. Other experiments reveal new properties of existing algorithms, some of which are different from what has been reported in the literature. Importantly, our new approach to robustness reveals a clear line separating the algorithms robust to noise from those which are not robust. Future research shall include the following issues: 1) development of further ways to asses BMF algorithms' robustness to noise, including the contributions in [Tatti and Vreeken, 2012]; 2) thorough analysis of robustness in the context of real as well as synthetic data, including recent algorithms such as PANDA+ and NASSAU; 3) real-case studies in the research on noise in BMF; 4) development of a reasonable, purpose-directed methodology to asses quality of BMF algorithms.

k	Noise (%)	8M	TILING	ASSO	GRECOND	PANDA	HYPER	GREESS	NASSAU	GRECOND+
5	0.1	0.985	0.228	1.000	0.228	1.000	0.063	0.239	0.722	1.000
	0.5	0.984	0.210	0.998	0.210	0.998	0.063	0.220	0.358	0.998
	1	0.789	0.186	0.998	0.187	0.992	0.063	0.195	0.261	0.998
	2	0.834	0.161	0.998	0.161	0.995	0.063	0.167	0.306	0.998
	5	0.760	0.111	0.997	0.112	0.880	0.063	0.115	0.236	0.982
	10	0.666	0.084	0.977	0.084	0.806	0.061	0.089	0.332	0.947
	30	0.427	0.050	0.771	0.052	0.535	0.056	0.061	0.363	0.738
10	0.1	1.000	0.459	1.000	0.459	0.833	0.127	0.481	0.777	1.000
	0.5	0.976	0.415	1.000	0.415	0.813	0.123	0.434	0.548	1.000
	1	0.893	0.373	0.996	0.375	0.764	0.115	0.389	0.584	0.999
	2	0.963	0.315	0.995	0.313	0.802	0.108	0.321	0.442	0.995
	5	0.789	0.223	0.946	0.224	0.616	0.104	0.229	0.442	0.980
	10	0.693	0.164	0.935	0.163	0.594	0.103	0.171	0.509	0.980
	30	0.400	0.107	0.674	0.108	0.420	0.090	0.114	0.499	0.815
15	0.1	0.995	0.825	0.994	0.904	0.833	0.248	0.924	0.802	0.999
	0.5	0.954	0.759	0.987	0.826	0.778	0.241	0.830	0.685	0.994
	1	0.908	0.625	0.933	0.687	0.831	0.225	0.686	0.519	0.997
	2	0.888	0.542	0.893	0.573	0.751	0.211	0.573	0.432	0.972
	5	0.742	0.366	0.863	0.360	0.725	0.192	0.355	0.421	0.921
	10	0.640	0.259	0.789	0.268	0.528	0.160	0.257	0.561	0.873
	30	0.368	0.161	0.559	0.166	0.436	0.126	0.157	0.480	0.801

Table 1: Robustness to noise (Domino dataset, general noise).

Noise	Change (%)	8M	TILING	ASSO	GRECOND	PANDA	HYPER	GREESS	NASSAU	GRECOND+
Aditive	0.1	0.887	0.726	0.974	0.726	0.728	0.015	0.726	0.520	0.932
	0.5	0.854	0.482	0.974	0.482	0.697	0.014	0.482	0.612	0.928
	1	0.791	0.353	0.973	0.356	0.592	0.014	0.356	0.688	0.927
	2	0.788	0.257	0.973	0.260	0.631	0.013	0.260	0.747	0.929
	5	0.760	0.146	0.964	0.151	0.579	0.012	0.151	0.867	0.922
	10	0.754	0.106	0.924	0.111	0.534	0.011	0.108	0.905	0.920
	30	0.765	0.075	0.388	0.082	0.526	0.010	0.082	0.843	0.902
Subtractive	0.1	0.632	0.683	0.991	0.683	0.650	0.085	0.715	0.732	0.914
	0.5	0.593	0.443	0.991	0.441	0.673	0.084	0.461	0.637	0.895
	1	0.555	0.288	0.983	0.290	0.660	0.084	0.326	0.704	0.858
	2	0.533	0.209	0.956	0.213	0.682	0.084	0.238	0.677	0.821
	5	0.591	0.105	0.855	0.105	0.555	0.083	0.104	0.537	0.665
	10	0.562	0.058	0.602	0.058	0.488	0.081	0.054	0.723	0.530
	30	0.804	0.013	0.109	0.013	0.187	0.068	0.013	0.619	0.159

Table 2: Recovery of ground truth on synthetic dataset 500×250 with $k = 5$.

References

[Belohlavek and Trnecka, 2015] R. Belohlavek and M. Trnecka. From-below approximations in Boolean matrix factorization: Geometry and new algorithm. *Journal of Computer and System Sciences*, 81(8):1678–1697, June 2015.

[Belohlavek and Vychodil, 2009] R. Belohlavek and V. Vychodil. Discovery of optimal factors in binary data via a novel method of matrix decomposition. *Journal of Computer and System Sciences*, 76(1):3–20, May 2009.

[Dixon, 1992] W. J. Dixon. *BMDP statistical software manual: To accompany BMDP release 7*. University of California Press, 1992.

[Ene *et al.*, 2008] A. Ene, W. Horne, N. Milosavljevic, P. Rao, R. Schreiber, and R. E. Tarjan. Fast exact and heuristic methods for role minimization problems. In *Proceedings of the 13th ACM Symposium on Access Control Models and Technologies*, pages 1–10, Estes Park, Colorado, USA, June 2008. ACM.

[Geerts *et al.*, 2004] F. Geerts, B. Goethals, and T. Mielikainen. Tiling databases. In *Proceedings of the 7th International Conference on Discovery Science*, pages 278–289, Berlin, Heidelberg, Oct 2004. Springer Berlin Heidelberg.

[Gupta *et al.*, 2008] R. Gupta, G. Fang, B. Field, M. Steinbach, and V. Kumar. Quantitative evaluation of approximate frequent pattern mining algorithms. In *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '08, pages 301–309, New York, NY, USA, 2008. ACM.

[Karaev *et al.*, 2015] S. Karaev, P. Miettinen, and J. Vreeken. Getting to know the unknown unknowns: Destructive-noise resistant Boolean matrix factorization. In *Proceedings of the 2015 SIAM International Conference on Data Mining, SDM 2015*, pages 325–333, 2015.

[Kim, 1982] K. H. Kim. *Boolean matrix theory and applications*. Monographs and textbooks in pure and applied mathematics. Dekker, 1982.

[Lu *et al.*, 2008] H. Lu, J. Vaidya, and V. Atluri. Optimal Boolean matrix decomposition: Application to role engineering. In *Proceedings of the 2008 IEEE 24th International Conference on Data Engineering*, ICDE '08, pages 297–306, Washington, DC, USA, 2008. IEEE Computer Society.

[Lucchese *et al.*, 2010] C. Lucchese, S. Orlando, and R. Perego. Mining top-k patterns from binary datasets in presence of noise. In *SIAM DM 2010*, pages 165–176. SIAM, 2010.

[Lucchese *et al.*, 2014] C. Lucchese, S. Orlando, and R. Perego. A unifying framework for mining approximate top-k binary patterns. *IEEE Transactions on Knowledge and Data Engineering*, 26(12):2900–2913, Dec 2014.

[Miettinen and Vreeken, 2011] P. Miettinen and J. Vreeken. Model order selection for Boolean matrix factorization. In *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge discovery and data mining*, pages 51–59. ACM, 2011.

[Miettinen and Vreeken, 2014] P. Miettinen and J. Vreeken. Mdl4bmf: Minimum description length for Boolean matrix factorization. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 8(4):18, 2014.

[Miettinen *et al.*, 2008] P. Miettinen, T. Mielikainen, A. Gionis, G. Das, and H. Mannila. The discrete basis problem. *IEEE Transactions on Knowledge and Data Engineering*, 20(10):1348–1362, Oct 2008.

[Miettinen, 2009] P. Miettinen. *Matrix decomposition methods for data mining: Computational complexity and algorithms*. PhD thesis, Helsingin yliopisto, 2009.

[Nau *et al.*, 1978] D. S. Nau, G. Markowsky, M. A. Woodbury, and A. D. Bernard. A mathematical analysis of human leukocyte antigen serology. *Mathematical Biosciences*, 40(3-4):243–270, 1978.

[Nau, 1976] D. S. Nau. Specificity covering. Technical report, Tech. Rep. CS-1976-7, Duke University, 1976.

[Stockmeyer, 1975] L. J. Stockmeyer. *The set basis problem is NP-complete*. Research reports. IBM Thomas J. Watson Research Division, 1975.

[Tatti and Vreeken, 2012] N. Tatti and J. Vreeken. Comparing apples and oranges: Measuring differences between exploratory data mining results. *Data Mining and Knowledge Discovery*, 25(2):173–207, 2012.

[Vaidya *et al.*, 2007] J. Vaidya, V. Atluri, and Q. Guo. The role mining problem: Finding a minimal descriptive set of roles. In *Proceedings of the 12th ACM Symposium on Access Control Models and Technologies*, pages 175–184. ACM, 2007.

[Xiang *et al.*, 2011] Y. Xiang, R. Jin, D. Fuhry, and F. F. Dragan. Summarizing transactional databases with overlapped hyperrectangles. *Data Mining and Knowledge Discovery*, 23(2):215–251, Sep 2011.