Boolean Matrix Tri-Factorization

Christos Kolomvakis
Department of Mathematics and Operational Research
University of Mons
Mons, Belgium
christos.kolomvakis@umons.ac.be

Arnaud Vandaele
Department of Mathematics and Operational Research
University of Mons
Mons, Belgium
arnaud.vandaele@umons.ac.be

Nicolas Gillis
Department of Mathematics and Operational Research
University of Mons
Mons, Belgium
nicolas.gillis@umons.ac.be

ICASSP 2025 - 2025 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) | 979-8-3503-6874-1/25/$31.00 ©2025 IEEE | DOI: 10.1109/ICASSP49660.2025.10889808

***Abstract*—Matrix tri-factorizations (MTFs) aim to decompose an input matrix $\mathbf{X}$ into the product of three factor matrices, instead of only two as in standard matrix factorization (MF). In contrast to MF, MTF is able to cluster both rows and columns of $\mathbf{X}$ while quantifying the relationship among these two groups of clusters. When dealing with binary input matrices, Boolean matrix factorization (BMF) is a natural extension of MF. In this work we focus on Boolean matrix tri-factorization (BMTF) that extends BMF to the tri-factorization framework. We first show an identifiability result for BMTF, namely, we show that the factors are unique under certain sparsity conditions. Then we propose an algorithm to compute the factors of BMTF, and perform numerical experiments to show how it performs on synthetic and real data.**

***Index Terms*—Boolean matrix tri-factorization, block coordinate descent,integer programming.**

I. Introduction

Given an input matrix $\mathbf{X}$, matrix factorization models aim to approximate it as the product of two matrices, $\mathbf{W}$ and $\mathbf{H}$, which are called the factors. Depending on the application, we can impose constraints on the factors. Examples include the singular value decomposition (SVD) where the factors are orthogonal, sparse PCA where the factors are sparse [1], nonnegative matrix factorization (NMF) where the factors have nonnegative elements [2], and binary matrix factorization and Boolean matrix factorizations (BMF) where the factors have elements in $\{0,1\}$ [3]–[5]. Applications include document clustering, hyperspectral unmixing and recommender systems; see, e.g., [6], [7]. As an extension to matrix factorizations, matrix tri-factorization (MTF) models aim to decompose the input matrix into three factors. In this paper, we focus on the following definition, using the Frobenius norm to quantify the error of the approximation.

Definition 1 (MTF)**.** *Given a matrix $\mathbf{X} \in \mathbb{R}^{m\times n}$ and factorization ranks (r_1, r_2), MTF aims to find matrices $\mathbf{W} \in \mathbb{R}^{m\times r_1}$, $\mathbf{S} \in \mathbb{R}^{r_1\times r_2}$ and $\mathbf{H} \in \mathbb{R}^{r_2\times n}$ that solve*

$$\min_{\mathbf{W}\in\mathbb{R}^{m\times r_1},\mathbf{S}\in\mathbb{R}^{r_1\times r_2},\mathbf{H}\in\mathbb{R}^{r_2\times n}} \|\mathbf{X} - \mathbf{WSH}\|_F^2.$$

Nonnegative MTF (NMTF) requires that the factors, $\mathbf{W}$, $\mathbf{S}$ and $\mathbf{H}$, are component-wise nonnegative. This leads to an easy interpretation of NMTF: the columns of $\mathbf{W}$ (resp. rows of $\mathbf{H}$) provide a soft clustering of the rows (resp. columns) of $\mathbf{X}$ into r_1 (resp. r_2) clusters: $\mathbf{W}_{ik}$ indicates the membership value of the ith row of $\mathbf{X}$ in the kth row cluster, $\mathbf{H}_{\ell j}$ indicates the membership value of the jth column of $\mathbf{X}$ in the ℓth column cluster, while $\mathbf{S}_{k\ell}$ indicates the interaction between the kth row cluster and the ℓth column cluster.

A first variant of NMTF was explored in [8], adding orthogonality constraints, $\mathbf{W}^\top\mathbf{W} = \mathbf{I}_m$ and $\mathbf{H}^\top\mathbf{H} = \mathbf{I}_n$, where $\mathbf{I}_m$ denotes the m-by-m identity matrix. Orthogonality imposes that the clusters of rows and columns are disjoint since $\mathbf{W} \geq 0$ and $\mathbf{W}^\top\mathbf{W} = \mathbf{I}_m$ implies that $\mathbf{W}$ has at most one non-zero entry per row, and similarly for the columns of $\mathbf{H}$. They proposed multiplicative update algorithms and applied them for document clustering. There are numerous other works exploring NMTF in other contexts; see, e.g., [9]–[12] and the references therein.

Contribution and outline of the paper. This paper introduces Boolean matrix tri-factorization (BMTF) which, to the best of our knowledge, has not been explored yet in the literature. A closely related model was proposed in [13] where $\mathbf{W}$ and $\mathbf{H}$ were restricted to be columns and rows of $\mathbf{X}$, resp., while [13] did not consider identifiability nor interpretability aspects which is a central focus of our work. In Section II, we recall the definition of BMF and formally define BMTF. In Section III, we discuss the identifiability of BMTF and prove identifiability with orthogonality constraints. In Section IV, we describe our proposed block coordinate descent algorithm for BMTF. We also provide a refinement procedure after updating the factors $\mathbf{W}$ and $\mathbf{H}$ to generate sparser and more expressive solutions. In Section V, we perform numerical experiments to assess the performance of our proposed algorithm on synthetic and real datasets. Finally, in Section VI we conclude the paper with some observations and future research directions.

II. Boolean matrix tri-factorization

Let us first define the matrix Boolean product and BMF.

Definition 2 (Boolean product)**.** *Given two Boolean matrices, $\mathbf{W} \in \{0,1\}^{m\times r}$ and $\mathbf{H} \in \{0,1\}^{r\times n}$, their Boolean product is denoted $\mathbf{W} \circ \mathbf{H} \in \{0,1\}^{m\times n}$ and is defined for all i, j as $(\mathbf{W} \circ \mathbf{H})_{ij} = \bigvee_{k=1}^{r} \mathbf{W}_{ik}\mathbf{H}_{kj}$, where $\vee$ is the logical OR operation (that is, $0 \vee 0 = 0$, $1 \vee 0 = 0 \vee 1 = 1 \vee 1 = 1$). Interestingly, $\mathbf{W} \circ \mathbf{H} = \min(1, \mathbf{WH})$ where $\mathbf{WH}$ is the usual matrix product between $\mathbf{W}$ and $\mathbf{H}$.*

The authors acknowledge the support by the European Union (ERC consolidator, eLinoR, no 101085607).

Definition 3 (BMF). *Given a Boolean matrix* $\mathbf{X} \in \{0,1\}^{m\times n}$ *and a factorization rank* r*, BMF aims to find matrices* $\mathbf{W} \in \{0,1\}^{m\times r}$ *and* $\mathbf{H} \in \{0,1\}^{r\times n}$ *that minimize* $\|\mathbf{X} - \mathbf{W} \circ \mathbf{H}\|_F^2$.

BMF has been used in medical domains [14]–[16] and recommender systems [17], [18]; see the survey paper [5] for more details and applications. In this paper, we consider a Boolean tri-factorization model that gives the opportunity for a more refined and flexible analysis of binary datasets. A two-factor model as BMF decomposes the data matrix $\mathbf{X}$ as the sum of r communities, $\mathbf{W}(:,k)\mathbf{H}(k,:)$ for $k = 1, 2, \ldots, r$, where a community is made of a cluster of rows and a cluster of columns. The kth cluster of rows, defined as $\{i \mid \mathbf{W}(i,k) = 1\}$, can only interact with the kth cluster of columns, $\{j \mid \mathbf{H}(k,j) = 1\}$, and the number of cluster of rows and columns must be the same. A three-factorization model allows for any interactions between these clusters, and for a different number of clusters in both dimensions. Let us define BMTF formally.

Definition 4 (BMTF). *Given a Boolean matrix* $\mathbf{X} \in \{0,1\}^{m\times n}$ *and factorization ranks* (r_1, r_2)*, BMTF aims to find* $\mathbf{W} \in \{0,1\}^{m\times r_1}$*,* $\mathbf{S} \in \{0,1\}^{r_1\times r_2}$ *and* $\mathbf{H} \in \{0,1\}^{r_2\times n}$ *that minimize* $\|\mathbf{X} - \mathbf{W} \circ \mathbf{S} \circ \mathbf{H}\|_F^2$.

BMTF is able to detect

- r_1 clusters[1] for the rows of $\mathbf{X}$, defined as $\{i \mid \mathbf{W}(i,k) = 1\}$ for $k = 1, 2, \ldots, r_1$,
- r_2 clusters for the columns of $\mathbf{X}$, defined as $\{j \mid \mathbf{H}(\ell,j) = 1\}$ for $\ell = 1, 2, \ldots, r_2$,
- the interactions between these clusters via the matrix $\mathbf{S}$ since $\mathbf{X} \approx \sum_{k=1}^{r_1}\sum_{\ell=1}^{r_2} \mathbf{W}(:,k)\mathbf{S}(k,\ell)\mathbf{H}(\ell,:)$.

For example, let $\mathbf{X}$ be a data set where the rows correspond to animals and the columns represent characteristics (e.g., 'has fins', 'flies', 'has 4 legs'), while $\mathbf{X}(i,j) = 1$ if animal i has the characteristic j. BMTF can not only find clusters of animals (in $\mathbf{W}$) and characteristics (in $\mathbf{H}$), but also it can link the two sets of clusters through the factor $\mathbf{S}$; e.g., a cluster of fishes to a cluster of their characteristics such as 'aquatic', and 'has fins'; see Section V for a real-world example. In the next two sections, we discuss the identifiability of BMTF, and then we propose an algorithm to compute solutions to BMTF.

III. Identifiability via Orthogonality

A BMTF, $\mathbf{X}{=}\mathbf{W}{\circ}\mathbf{S}{\circ}\mathbf{H}$, is identifiable/unique if any other BMTF of $\mathbf{X}$ can only be obtained via permutations, that is, for any other BMTF $\mathbf{X}{=}\mathbf{W}'{\circ}\mathbf{S}'{\circ}\mathbf{H}'$ of the same size, we have $\mathbf{W}'{=}\mathbf{W}(:,\pi_1)$, $\mathbf{S}'{=}\mathbf{S}(\pi_1,\pi_2)$, and $\mathbf{H}'{=}\mathbf{W}(\pi_2,:)$, for some permutations π_1 of $\{1,2,\ldots,r_1\}$ and π_2 of $\{1,2,\ldots,r_2\}$.

It is crucial to note that when $r_1 \neq r_2$, plain BMTF is never identifiable (that is, BMTF without additional constraints). Assume w.l.o.g. that $r_1 > r_2$ and let $\mathbf{X} = \mathbf{W} \circ \mathbf{S} \circ \mathbf{H}$ be a BMTF. Let us show that we can always construct another BMTF of $\mathbf{X}$ which cannot be obtained as a permutation of $\mathbf{W} \circ \mathbf{S} \circ \mathbf{H}$. There are two cases:

[1]Note that the clusters do not need to be disjoint.

- If a column of $\mathbf{W}$ is equal to zero, say $\mathbf{W}(:,k) = 0$, then the corresponding row of $\mathbf{S}$, $\mathbf{S}(k,:)$, can take any value and the BMTF is not identifiable.
- Otherwise, another BMTF is given by

$$\mathbf{W}' = [\mathbf{W} \circ \mathbf{S}, \mathbf{0}_{m\times(r_1-r_2)}], \mathbf{S}' = [\mathbf{I}_{r_2}; \mathbf{0}_{(r_1-r_2)\times r_2}], \mathbf{H}' = \mathbf{H},$$

where $\mathbf{0}_{a\times b}$ is the a-by-b all-zero matrix, so that $\mathbf{W}'$ has $r_1 - r_2$ zero columns. Another way to make this observation is to realize that BMTF is an overparametrized BMF model, since $\mathbf{W}\circ\mathbf{S}\circ\mathbf{H} = (\mathbf{W}\circ\mathbf{S})\circ\mathbf{H} = \mathbf{W}\circ(\mathbf{S}\circ\mathbf{H})$. Hence, to be able to provide additional insight on the data set and to be identifiable, BMTF requires additional constraints. For example, a natural goal would be to look for the sparsest $\mathbf{W}$ and $\mathbf{H}$ to identify the smallest clusters that explain the data. Let us prove that BMTF is identifiable under the conditions that the clusters are disjoint, or equivalently that the columns of $\mathbf{W}$ (resp. rows of $\mathbf{H}$) are orthogonal. Before proving this result, let us provide a definition and a lemma.

Definition 5 (Orthogonal BMTF). *Orthogonal BMTF is the BMTF problem with the additional constraints that* $\mathbf{W}(:,i)^\top\mathbf{W}(:,j) = 0$ *for all* $i \neq j$ *and* $\mathbf{H}(k,:)\mathbf{H}(p,:)^\top = 0$ *for all* $k \neq p$.

Lemma 1. *Let* $\mathbf{S} \in \{0,1\}^{r_1\times r_2}$ *have distinct non-zero rows and columns. Then the unique exact orthogonal BMTF of* $\mathbf{S}$ *with ranks* (r_1, r_2) *is* $\mathbf{I}_{r_1} \circ \mathbf{S} \circ \mathbf{I}_{r_2}$*, up to permutations.*

Proof. Let $\mathbf{S}{=}\mathbf{W}{\circ}\mathbf{S}{\circ}\mathbf{H}$ be an orthogonal BMTF of $\mathbf{S}$. Because $\mathbf{S}$ has no zero columns, and $\mathbf{H}$ has at most a single non-zero entry per column, each column of $\mathbf{S}$ is equal to a column of $\mathbf{W}{\circ}\mathbf{S}'$. Moreover, since the columns of $\mathbf{S}$ are distinct and non-zero, and $\mathbf{W}{\circ}\mathbf{S}'$ has r_2 columns, $\mathbf{H}$ must be a permutation of the identity. Using the same argument on the rows, we conclude that $\mathbf{W}$ must be a permutation of the identity. □

Theorem 1. *Let* $\mathbf{X} = \mathbf{W} \circ \mathbf{S} \circ \mathbf{H}$ *be an orthogonal BMTF with ranks* (r_1, r_2) *where each column of* $\mathbf{W}$ *and* $\mathbf{H}^\top$ *has a least one non-zero element (no cluster is empty), and* $\mathbf{S} \in \{0,1\}^{r_1\times r_2}$ *has distinct non-zero rows and columns. Then* X *has a unique orthogonal BMTF.*

Proof. The uniqueness of $\mathbf{W}$ and $\mathbf{H}$ follows from the uniqueness of ONMF [7, Th 4.40, p.136]. Let us recall this result: if $\mathbf{X} = \mathbf{AB}$ where $\mathbf{B} \geq \mathbf{0}$ has orthogonal rows, and $\mathbf{A}$ has non-multiple columns, then $(\mathbf{A}, \mathbf{B})$ is an identifiable ONMF. Since $\mathbf{W}$ and $\mathbf{H}^\top$ has orthogonal columns, we have $\mathbf{W}\circ\mathbf{S}\circ\mathbf{H} = \mathbf{WSH}$. Applying the uniqueness result of ONMF to $(\mathbf{WS})\mathbf{H}$ and $\mathbf{W}(\mathbf{SH})$, we have that $\mathbf{H}$ is unique if $\mathbf{WS}$ has non-multiple columns and $\mathbf{W}$ is unique if $\mathbf{SH}$ has non-multiple rows. Since $\mathbf{W}$ has no zero column and is binary, $\mathbf{W}$ contains the identity matrix, hence the matrix $\mathbf{S}$ appears as a submatrix of $\mathbf{WS}$. Since $\mathbf{S}$ has distinct columns, $\mathbf{WS}$ also has. The same argument applies to $\mathbf{SH}$.

Finally, since $\mathbf{W}$ and $\mathbf{H}$ are unique and contain the identity as a submatrix, $\mathbf{S}$ has to be unique since $\mathbf{I}_{r_1}\mathbf{S}\mathbf{I}_{r_2}$ is the only orthogonal BMTF of $\mathbf{S}$; see Lemma 1. □

IV. Block-coordinate descent for BMTF

In this section, we propose a block-coordinate descent (BCD) method to solve BMTF. It relies on our previous work that proposed a BCD scheme for BMF [19]. The scheme is a standard approach for matrix and tensor factorizations: optimize over each factor individually, in our case $\mathbf{W}$, $\mathbf{S}$ and then $\mathbf{H}$, while the others are fixed. To optimize $\mathbf{W}$, we need to solve m independent Boolean least squares (BoolLS) problem:

$$\min_{\mathbf{W}(i,:)\in\{0,1\}^{r_1}} \|\mathbf{X}(i,:)-\mathbf{W}(i,:)\mathbf{H}\|_F^2. \quad (1)$$

Each subproblem has only r_1 binary variables and can be solved relatively fast using an integer programming software, and we use Gurobi [20]. For $\mathbf{H}$, we need to solve n BoolLS in r_2 variables, one for each column of $\mathbf{H}$. Note however that the worst case complexity is $O(m2^{r_1}+n2^{r_2})$. The update of $\mathbf{S}$ needs to be handled slightly differently. Using the property that $\operatorname{vec}(\mathbf{ABC}) = (\mathbf{C}^\top \otimes \mathbf{A}) \operatorname{vec}(\mathbf{B})$, where $\otimes$ is the Kronecker product and vec vectorizes a matrix as a vector column wise, the problem in $\mathbf{S}$ is a BoolLS in $r_1 r_2$ variables:

$$\min_{\mathbf{S}\in\{0,1\}^{r_1\times r_2}} \|\operatorname{vec}(\mathbf{X})-\min\big(1,(\mathbf{H}^\top\otimes\mathbf{W})\operatorname{vec}(\mathbf{S})\big)\|_F^2, \quad (2)$$

with worst-case complexity $O(2^{r_1 r_2})$.

Imposing sparsity. As explained in Section III, BMTF is in general not identifiable unless additional constraints are imposed. We consider sparsity constraints, and generate sparse solutions by adding explicit constraints on the rows of $\mathbf{W}$ (resp. columns of $\mathbf{H}$) when solving (1): for all i, $\sum_{k=1}^{r_1} \mathbf{W}(i,k) \leq K_W$, where $1 \leq K_W \leq r_1$ is a sparsity parameter, and similarly for $\mathbf{H}$. For example, setting $K_W{=}1$ corresponds to the orthogonality constraint discussed in Section III: each row of $\mathbf{X}$ belongs to at most one cluster. Sparsity is a natural constraint as it corresponds to identify the smallest clusters that explain the data.

Generating sparser and more expressive solutions. When $r_1 > r_2$, we have observed that BCD often generates $\mathbf{W}$'s with zero columns, because of the identifiability issues discussed in Section III. In order to generate sparser solutions and avoid rank-deficient ones, we resort to a refinement procedure that will generate sparser and more expressive solutions. This procedure is inspired by [21], and has been used recently in [22] for tensor factorizations. Let $\mathbf{W}{\in}\{0,1\}^{m\times r_1}$, and assume that the support of $\mathbf{W}(:,i)$ (that is, the set of indices corresponding to non-zero entries) contains that of $\mathbf{W}(:,j)$. Then we construct $(\mathbf{W}',\mathbf{S}')$ such that $\mathbf{W}'{\circ}\mathbf{S}'{=}\mathbf{W}{\circ}\mathbf{S}$ and $\mathbf{W}'$ is sparser than $\mathbf{W}$: $\mathbf{W}'(:,i){=}\mathbf{W}(:,i){-}\mathbf{W}(:,j)$, $\mathbf{S}'(i,:){=}\mathbf{S}(i,:) \vee \mathbf{S}(j,:)$, and the other columns of $\mathbf{W}'$ (resp. rows of $\mathbf{S}'$) are equal to that of $\mathbf{W}$ (resp. $\mathbf{S}$). Using this observation, the lemma below follows (the formal proof is straightforward and will be described in an extended version of the paper).

Lemma 2. *Let* $\mathbf{W} \in \{0,1\}^{m\times r_1}$ *have distinct columns. Let* $\mathbf{P} \in \{-1,0,1\}^{r_1\times r_1}$ *and* $\mathbf{Q} \in \{0,1\}^{r_1\times r_1}$ *be as follows:*
(1) $\mathbf{P}(i,i) = \mathbf{Q}(i,i) = 1$ *for all i, and*
(2) $\mathbf{P}(j,i) = -1$ *and* $\mathbf{Q}(j,i) = 1$ *for all* $i \neq j$ *such that the support of* $\mathbf{W}(:,i)$ *contains that of* $\mathbf{W}(:,j)$*.*
Then $\mathbf{W}{=}\mathbf{W}' \circ \mathbf{Q}$*, where* $\mathbf{W}'{=}\max(0,\mathbf{WP}) \in \{0,1\}^{m\times r_1}$ *is sparser than* $\mathbf{W}$ *(since* $\mathbf{P}$ *has negative entries on the off-diagonal) and more expressive[2] in BMTF since* $\mathbf{W}\circ\mathbf{S} = \mathbf{W}'\circ(\mathbf{Q}\circ\mathbf{S}) = \mathbf{W}'\circ\mathbf{S}'$ *for any* $\mathbf{S}$ *and for* $\mathbf{S}' = \mathbf{Q}\circ\mathbf{S}$*.*

Note that if $\mathbf{W}$ has two identical columns, one of them can be set to zero, and $\mathbf{S}$ can be updated accordingly without changing $\mathbf{W}\circ\mathbf{S}$. To reinitialize zero columns of $\mathbf{W}$ (and similarly for rows of $\mathbf{H}$), we set a single entry to 1, at the position where the row of the residual $\mathbf{R} = \mathbf{X} - \mathbf{W}\circ\mathbf{S}\circ\mathbf{H}$ has the most entries equal to one.

To summarize, we propose the following **refinement procedure**: (i) set to zero duplicated columns of $\mathbf{W}$, (ii) make the $\mathbf{W}$ sparser and more expressive via Lemma 2, (iii) reinitialize zero columns with a single non-zero entry as described above.

Initialization. We initialize the matrix $\mathbf{W}$ either by randomly sampling r_1 columns of $\mathbf{X}$ or by binarizing an NMF solution of $\mathbf{X}$, as done in [19]. W.l.o.g. we assume $r_1 \geq r_2$ (otherwise, transpose $\mathbf{X}$), and the matrix $\mathbf{S}$ is initialized with the identity matrix $\mathbf{I}_{r_2}$ while the remaining rows are filled with a single 1 in random positions. Finally, here is our proposed algorithm:

BCD algorithm for BMTF

Input: $\mathbf{X} \in \{0,1\}^{m\times n}$, ranks $(r_1,\, r_2)$, sparsity K_W, K_H
0. Initialize $\mathbf{W}{\in}\{0,1\}^{m\times r_1}$ and $\mathbf{S}{\in}\{0,1\}^{r_1\times r_2}$ as above.
ṅoindent 1. while $\mathbf{W}$, $\mathbf{S}$ or $\mathbf{H}$ change
  1.1. Update $\mathbf{H}$ by column with BoolLS with sparsity K_H, and perform the refinement procedure.
  1.2. Update $\mathbf{S}$ by solving the BoolLS (2).
  1.3. Update $\mathbf{W}$ by row (1) with sparsity K_W, and perform the refinement procedure.
  1.4. Update $\mathbf{S}$ again by solving the BoolLS (2).

The above steps ensure that the BCD scheme generates a sequence of solutions with non-increasing objective function.

V. Numerical Experiments

All experiments in this section are performed on Julia v.1.9.2 with Gurobi version 11 on a laptop with an intel i7 1255U processor @ 1.7 GHz and 16 GB RAM.

Synthetic experiment. We first construct an orthogonal BMTF as follows: $\mathbf{X}{=}\mathbf{W}{\circ}\mathbf{S}{\circ}\mathbf{H}{\in}\{0,1\}^{120\times 80}$ where $\mathbf{W} \in \{0,1\}^{120\times 7}$ has one non-zero element per row. Each column of $\mathbf{W}$ has $17{=}\lfloor 120/7 \rfloor$ non-zero elements, and the last one has 18. The same procedure is used to generate the rows of $\mathbf{H}$. For $\mathbf{S}$, each row has 2 non-zero elements: We list all the possible binary vectors of dimension $5(= r_2)$ with 2 entries equal to one, and pick $7(= r_1)$ of them randomly as the rows of $\mathbf{S}$. Then, to make the problem more challenging, we add to $\mathbf{W}$ one nonzero per row at a random position, and then one nonzero per columns of $\mathbf{H}$. We denote (z_W, z_S, z_H) the number of non-zeros per row of $\mathbf{W}$, per row of $\mathbf{S}$ and per column of $\mathbf{H}$. For each experimental setting, we run our algorithm with 25 random initializations (we use the NMF-based initialization for $\mathbf{W}$) and keep the best solution (unless

[2] That is, any BMTF generated using $\mathbf{W}$ can also be generated using $\mathbf{W}'$.

it finds a solution with zero error in which case we stop it), and we repeat this procedure 50 times (50 Monte Carlo trials). Table I reports the percentage of times the ground truth $\mathbf{W}$, $\mathbf{S}$ and $\mathbf{H}$ were found correctly among the 50 runs, and in parenthesis the percentage of elements found wrong on average. The fifth column (err.) reports the percentage of runs when the average relative error $\frac{\|\mathbf{X}-\mathbf{W}\circ\mathbf{S}\circ\mathbf{H}\|_F}{\|\mathbf{X}\|_F}$ is equal to zero, and in brackets it reports its value. The last column reports the average time in seconds needed to generate one BMTF solution with one niitialization. For the case where $\mathbf{W}$ and $\mathbf{H}$ are orthogonal (that is, $z_W=z_H=1$, first row), the ground-truth factors are always recovered (on an average using less than 7 initializations). This illustrates two facts: our identifiability results (Theorem 1) and the fact that our algorithm performs well as it is able to recover the unique solution in all 50 trials. For the case $\mathbf{W}$ is not orthogonal but $\mathbf{H}$ is (second row), the algorithm can still find the ground-truth $\mathbf{H}$ in most cases, although it cannot recover $\mathbf{W}$ and $\mathbf{S}$, which are non-unique: in fact, in 45% of the trials, BCD finds a solution with a zero relative error but $\mathbf{W}$ and $\mathbf{S}$ are not recovered. The problem becomes even harder when we add the extra nonzeros on $\mathbf{H}$ as well (third row). However, although we cannot recover the groundtruth, the recovered factors share many entries with it (about 75%).

(z_W, z_S, z_H)	$\mathbf{W}$	$\mathbf{S}$	$\mathbf{H}$	err.	time
(1,2,1)	100%	100%	100%	100%	1.13 s.
(2,2,1)	0% (21%)	0% (16%)	99.8% (0.16‰)	45% (9%)	1.43 s.
(2,2,2)	0% (24%)	0% (24%)	0% (26%)	0.7% (28%)	1.32 s.

TABLE I
PERCENTAGE OF TIME THE GROUND-TRUTH FACTORS, OR ZERO ERRORS, ARE FOUND AMONG THE 50 MONTE CARLO RUNS ON SYNTHETIC DATA SETS. IN BRACKETS, WE REPORT THE PERCENTAGE OF ENTRIES FOUND WRONG ON AVERAGE FOR THE FACTORS $\mathbf{W}$, $\mathbf{S}$ AND $\mathbf{H}$, AND REPORT THE AVERAGE RELATIVE ERROR.

Experiment on a real data set. Let us consider an experiment on the "zoo" real dataset [23]. Each row represents an animal, and each column represents a characteristic; see Tables II-III for examples. We have removed some characteristics and some animals from the dataset as done in [24] (e.g., the characteristic 'breathes' which is equal to one for almost all animals) to obtain a matrix $\mathbf{X} \in \{0,1\}^{99\times 14}$. Here, we are considering $(r_1, r_2) = (5, 3)$. Over 1500 Monte Carlo trials, each of which took on average 0.58 seconds, we report the factors with the lowest error $\|\mathbf{X}-\mathbf{W}\circ\mathbf{S}\circ\mathbf{H}\|_F/\|\mathbf{X}\|_F$=33%, which is relatively high since the ranks chosen are small ($r_2 = 3$) and real data do not perfectly fit low-rank models. The parameters considered for $\mathbf{W}$ and $\mathbf{H}$ were $(K_W, K_H) = (2, 2)$. Tables II-III show the clusters. Because of limited space, we show only parts of the animals clusters. The interested reader can rerun the experiment as the code will be made available upon request. We will now present the matchings that we receive

Aquatic animals: bass, carp, catfish, chub, crayfish, dogfish, ..., octopus, penguin, pike, ..., stingray, tuna
Birds: chicken, crow, dove, duck, flamingo, gnat, ..., parakeet, penguin, pheasant, ..., vulture, wasp, wren
Mixed cluster: crab, crayfish, flea, frog, fruitbat, gnat, gorilla, honeybee, ..., wasp
Aquatic birds: gull, skimmer, skua
Mammals: aardvark, antelope, bear, boar, buffalo, calf, ..., vole, wallaby, wolf

TABLE II
CLUSTERS OF ANIMALS.

Birds: feathers, eggs, airborne, less than 4 legs
Mixed: eggs, aquatic, predator, toothed, fins, less than 4 legs, tail
Mammals: hair, milk, toothed, 4 legs, tail

TABLE III
CLUSTERS OF CHARACTERISTICS.

from the matrix $\mathbf{S}$:

$\mathbf{H}$ clusters / $\mathbf{W}$ clusters	Birds	Mixed	Mammals
Aquatic animals	0	1	0
Birds	1	0	0
Mixed cluster	0	0	0
Aquatic birds	1	1	0
Mammals	0	0	1

We observe that the clusters make sense. Furthermore, many animals are assigned to multiple clusters per the property of BMTF, e.g., the "penguin" is assigned to aquatic animals and birds, both being correct assignments.

VI. CONCLUSION

In this paper, we introduced Boolean matrix tri-factorization (BMTF). We gave motivations as to why this model is useful, discussed identifiability, and proposed a BCD algorithm that includes a clever refinement procedure. Our numerical experiments show that the BCD algorithm is able to find good solutions and can be used meaningfully on a real-world data set. Further work includes a deeper understanding of the conditions under which BMTF is identifiabile, as done for example for BMF in [25], [26], the use of BMTF for other applications, and the development of algorithms scalable to large-scale data.

REFERENCES

[1] H. Zou, T. Hastie, and R. Tibshirani, "Sparse principal component analysis," *Journal of Computational and Graphical Statistics*, vol. 15, no. 2, pp. 265–286, 2006.
[2] D. D. Lee and H. S. Seung, "Learning the parts of objects by non-negative matrix factorization," *Nature*, vol. 401, no. 6755, pp. 788–791, 1999.
[3] P. Miettinen, T. Mielikäinen, A. Gionis, G. Das, and H. Mannila, "The discrete basis problem," *IEEE Trans. Knowl. Data Eng.*, vol. 20, no. 10, pp. 1348–1362, 2008.

[4] Z. Zhang, T. Li, C. Ding, and X. Zhang, "Binary matrix factorization with applications," in *IEEE Int. Conf. on Data Mining*, 2007, pp. 391–400.

[5] P. Miettinen and S. Neumann, "Recent developments in Boolean matrix factorization," in *International Joint Conference on Artificial Intelligence*, 2021.

[6] M. Udell, C. Horn, R. Zadeh, S. Boyd *et al.*, "Generalized low rank models," *Foundations and Trends® in Machine Learning*, vol. 9, no. 1, pp. 1–118, 2016.

[7] N. Gillis, *Nonnegative Matrix Factorization*. Philadelphia, PA: SIAM, 2020.

[8] C. Ding, T. Li, W. Peng, and H. Park, "Orthogonal nonnegative matrix t-factorizations for clustering," in *12th ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining*, 2006, pp. 126–135.

[9] G. Chen, F. Wang, and C. Zhang, "Collaborative filtering using orthogonal nonnegative matrix tri-factorization," *Information Processing & Management*, vol. 45, no. 3, pp. 368–379, 2009. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0306457308001167

[10] J. Yoo and S. Choi, "Orthogonal nonnegative matrix tri-factorization for co-clustering: Multiplicative updates on stiefel manifolds," *Information Processing & Management*, vol. 46, no. 5, pp. 559–570, 2010. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0306457310000038

[11] M. Ortiz-Bouza and S. Aviyente, "Community detection in multiplex networks based on orthogonal nonnegative matrix tri-factorization," *IEEE Access*, 2024.

[12] A. Dache, A. Vandaele, and N. Gillis, "Orthogonal symmetric nonnegative matrix tri-factorization," in *34th IEEE International Workshop on Machine Learning for Signal Processing (MLSP)*. IEEE, 2024.

[13] P. Miettinen, "The Boolean column and column-row matrix decompositions," in *European Conference on Machine Learning and Knowledge Discovery in Databases*, ser. ECMLPKDD'08, 2008, p. 17.

[14] L. Liang, K. Zhu, and S. Lu, "BEM: mining coregulation patterns in transcriptomics via Boolean matrix factorization," *Bioinformatics 36 (13)*, pp. 4030–4037, 2020.

[15] A. Haddad, F. Shamsi, L. Zhu, and L. Najafizadeh, "Identifying dynamics of brain function via Boolean matrix factorization," in *Asilomar Conference on Signals, Systems, and Computers*, 2018.

[16] C. Wan, W. Chang, T. Zhao, M. Li, S. Cao, and C. Zhang, "Fast and efficient Boolean matrix factorization by geometric segmentation," in *AAAI Conference on Artificial Intelligence*, 2019. [Online]. Available: https://api.semanticscholar.org/CorpusID:211076357

[17] R. Cabral Farias and S. Miron, "A generalized approach for Boolean matrix factorization," *Signal Processing*, vol. 206, p. 108887, 2023. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0165168422004261

[18] B. Thirunavukarasu, N. Richi, and C. Yuen, "People to people recommendation using coupled nonnegative Boolean matrix factorization," in *Int. Conf. on Soft-computing and Network Security (ICSNS)*, 2018.

[19] C. Kolomvakis, A. Vandaele, and N. Gillis, "Algorithms for Boolean matrix factorization using integer programming," in *33rd International Workshop on Machine Learning for Signal Processing (MLSP)*. IEEE, 2023.

[20] Gurobi Optimization, LLC, "Gurobi Optimizer Reference Manual," 2023, https://www.gurobi.com.

[21] N. Gillis, "Sparse and unique nonnegative matrix factorization through data preprocessing," *The Journal of Machine Learning Research*, vol. 13, no. 1, pp. 3349–3386, 2012.

[22] L. Grasedyck, M. Klever, and S. Krämer, "Quasi-orthogonalization for alternating non-negative tensor factorization," *Electronic Transactions on Numerical Analysis*, vol. 62, pp. 22–57, 2024.

[23] D. Dua and C. Graff, "UCI machine learning repository," 2017. [Online]. Available: http://archive.ics.uci.edu/ml

[24] C. Wan, P. Dang, T. Zhao, Y. Zang, C. Zhang, and S. Cao, "Bias aware probabilistic Boolean matrix factorization," in *Thirty-Eighth Conference on Uncertainty in Artificial Intelligence*, 2022, pp. 2035–2044.

[25] S. Miron, M. Diop, A. Larue, E. Robin, and D. Brie, "Boolean decomposition of binary matrices using a post-nonlinear mixture approach," *Signal Processing*, vol. 178, p. 107809, 2021.

[26] D. Desantis, E. Skau, D. P. Truong, and B. Alexandrov, "Factorization of binary matrices: Rank relations, uniqueness and model selection of Boolean decomposition," *ACM Trans. Knowl. Discov. Data*, vol. 16, no. 6, 2022. [Online]. Available: https://doi.org/10.1145/3522594