

Bounded Simplex-Structured Matrix Factorization: Algorithms, Identifiability and Applications

Olivier Vu Thanh , Nicolas Gillis , *Member, IEEE*, and Fabian Lecron 

Abstract—In this article, we propose a new low-rank matrix factorization model dubbed bounded simplex-structured matrix factorization (BSSMF). Given an input matrix X and a factorization rank r , BSSMF looks for a matrix W with r columns and a matrix H with r rows such that $X \approx WH$ where the entries in each column of W are bounded, that is, they belong to given intervals, and the columns of H belong to the probability simplex, that is, H is column stochastic. BSSMF generalizes nonnegative matrix factorization (NMF), and simplex-structured matrix factorization (SSMF). BSSMF is particularly well suited when the entries of the input matrix X belong to a given interval; for example when the rows of X represent images, or X is a rating matrix such as in the Netflix and MovieLens datasets where the entries of X belong to the interval $[1,5]$. The simplex-structured matrix H not only leads to an easily understandable decomposition providing a soft clustering of the columns of X , but implies that the entries of each column of WH belong to the same intervals as the columns of W . In this article, we first propose a fast algorithm for BSSMF, even in the presence of missing data in X . Then we provide identifiability conditions for BSSMF, that is, we provide conditions under which BSSMF admits a unique decomposition, up to trivial ambiguities. Finally, we illustrate the effectiveness of BSSMF on two applications: extraction of features in a set of images, and the matrix completion problem for recommender systems.

Index Terms—Simplex-structured matrix factorization, nonnegative matrix factorization, identifiability, algorithms.

I. INTRODUCTION

LOW-RANK matrix factorizations have recently emerged as very efficient models for unsupervised learning; see, e.g., [44], [45] and the references therein. The most notable example is principal component analysis (PCA), which can be solved efficiently via the singular value decomposition. In the last 20 years, many new more sophisticated models have been proposed, such as sparse PCA that requires one of the factors to be sparse to improve interpretability [11], robust PCA to handle gross corruption and outliers [6], [7], and low-rank matrix

completion, also known as PCA with missing data, to handle missing entries in the input matrix [28].

Among such methods, nonnegative matrix factorization (NMF), popularized by Lee and Seung in 1999 [32], required the factors of the decomposition to be component-wise nonnegative. More precisely, given an input matrix $X \in \mathbb{R}^{m \times n}$ and a factorization rank r , NMF looks for a nonnegative matrix W with r columns and a nonnegative matrix H with r rows such that $X \approx WH$. NMF has been shown to be useful in many applications, including topic modeling, image analysis, hyperspectral unmixing, and audio source separation; see [9], [16], [20] for more examples. The main advantage of NMF compared to previously introduced low-rank models is that the nonnegativity constraints on the factors W and H lead to an easily interpretable part-based decomposition.

More recently, simplex-structured matrix factorization (SSMF) was introduced as a generalization of NMF [48]; see also [1] and the references therein. SSMF does not impose any constraint on W , while it requires H to be column stochastic, that is, $H(:,j) \in \Delta^r$ for all j , where $\Delta^r = \{x \in \mathbb{R}^r \mid x \geq 0, e^\top x = 1\}$ is the probability simplex and e is the vector of all ones of appropriate dimension. SSMF is closely related to various machine learning problems, such as latent Dirichlet allocation, clustering, and the mixed membership stochastic block model; see [3] and the references therein. Let us recall why SSMF is a generalization of NMF by considering the exact NMF model, $X = WH$. Let us normalize the input matrix such that the entries in each column sum to one (w.l.o.g. we assume X , and W , do not have a zero column), that is, such that $X^\top e = e$, and let us impose w.l.o.g. that the entries in each column of W also sum to one (by the scaling degree of freedom in the factorization WH), that is, $W^\top e = e$. Then, we have

$$X^\top e = e = (WH)^\top e = H^\top W^\top e = H^\top e, \quad (1)$$

so that H has to be column stochastic, since $H \geq 0$ and $H^\top e = e$ is equivalent to $H(:,j) \in \Delta^r$ for all j .

In this article, we introduce bounded simplex-structured matrix factorization (BSSMF). BSSMF imposes the columns of W to belong to a hyperrectangle, namely $W(i,j) \in [a_i, b_i]$ for all i, j for some parameters $a_i \leq b_i$ for all i . For simplicity, given $a \leq b \in \mathbb{R}^m$, we denote the hyperrectangle

$$[a, b] = \{y \in \mathbb{R}^m \mid a_i \leq y_i \leq b_i \text{ for all } i\},$$

Manuscript received 29 September 2022; revised 31 March 2023; accepted 31 May 2023. Date of publication 27 June 2023; date of current version 11 July 2023. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Thomas Oberlin. This work was supported in part by the Fonds de la Recherche Scientifique - FNRS (F.R.S.-FNRS) through Research Project under Grant T.0097.22, in part by FRIA F.R.S.-FNRS PhD grant and the Fonds Wetenschappelijk Onderzoek - Vlaanderen (FWO) through EOS Project under Grant O005318F-RG47, and in part by Francqui Foundation. (Corresponding author: Olivier Vu Thanh.)

The authors are with Mathematics and Operational Research, University of Mons, 7000 Mons, Belgium (e-mail: olivier.vuthanh@umons.ac.be; nicolas.gillis@umons.ac.be; fabian.lecron@umons.ac.be).

Digital Object Identifier 10.1109/TSP.2023.3289704

and refer to it as an interval. The hyperrectangle constraint on W is denoted as $W(:, j) \in [a, b]$ for all j . Let us formally define BSSMF.

Definition 1 (BSSMF): Let $X \in \mathbb{R}^{m \times n}$, let $r \leq \min(m, n)$ be an integer, and let $a, b \in \mathbb{R}^m$ with $a \leq b$. The pair $(W, H) \in \mathbb{R}^{m \times r} \times \mathbb{R}^{r \times n}$ is a BSSMF of X of size r for the interval $[a, b]$ if

$$W(:, k) \in [a, b] \text{ for all } k, \quad H \geq 0, \quad \text{and} \quad H^\top e = e.$$

Since the columns of H define convex combinations, the convex hull of the columns of $X = WH$ is contained in the convex hull of the columns of W , which is itself contained in the hyperrectangle $[a, b]$. This implies that the hyperrectangle $[a, b]$ must contain the columns of the data matrix, $X = WH$.

BSSMF reduces to SSMF when $a_i = -\infty$ and $b_i = +\infty$ for all i . When $X \geq 0$, BSSMF reduces to NMF when $a_i = 0$ and $b_i = +\infty$ for all i , after a proper normalization of X ; see the discussion around (1).

a) Outline and contribution of the article: The article is organized as follows. In Section II, we explain the motivation of introducing BSSMF. In Section III, we propose an efficient algorithm for BSSMF. In Section IV, we provide an identifiability result for BSSMF, which follows from an identifiability results for NMF. In Section V, we illustrate the effectiveness of BSSMF on two applications:

- *Image feature extraction:* The entries of X are pixel intensities. For example, for a gray level image, the entries of X belong to the interval $[0, 255]$.
- *Recommender systems:* The entries of X are ratings of users for some items (e.g., movies). These ratings belong to an interval, e.g., $[1, 5]$ for the Netflix and MovieLens datasets.

Remark 1 (Extended conference article): This article is an extended version of our conference article [47]. It provides significant new material:

- A more thorough discussion on the background and the motivations to introduce BSSMF; see Section II.
- A new algorithm handling missing data and accelerated via data centering; see Section III.
- Illustrations, examples, and proof for the identifiability of BSSMF; see Section IV-C.
- New numerical experiments on synthetic, and the MNIST and MovieLens datasets; see Sections III-C and V.

II. MOTIVATION OF BSSMF

The motivation to introduce BSSMF is mostly fourfold; this is described in the next four paragraphs.

a) Bounded low-rank approximation: When the data naturally belong to intervals, imposing the approximation to belong to the same interval allows to provide better approximations, taking into account this prior information. Imposing that the entries in W belong to some interval and that H is column stochastic resolves this issue. BSSMF implies that the columns of the approximation WH belong to the same interval as the columns of W . In fact, for all j ,

$$X(:, j) \approx WH(:, j) \in [a, b],$$

since $W(:, k) \in [a, b]^m$ for all k , and the entries of $H(:, j)$ are nonnegative and sum to one.

Another closely related model was proposed in [37] where the entries of the factors W and H are required to belong to bounded intervals. The authors showed that their model is suitable for clustering. Nonetheless, it is not clear how to choose the lower and upper bounds on the entries of W and H to obtain tight lower and upper bounds for their product WH . With BSSMF the choice for the lower and upper bounds is easier, e.g., choosing a_i and b_i to be the smallest and largest entry in $X(i, :)$, respectively, that is, bounding W in the same way the data matrix is; see Section IV-C for more details.

b) Interpretability: BSSMF allows us to easily interpret both factors: the columns of W can be interpreted in the same way as the columns of X (e.g., as movie ratings, or pixel intensities), while the columns of H provide a soft clustering of the columns of X as they are column stochastic. BSSMF can be interpreted geometrically similarly as SSMF and NMF: the convex hull of the columns of W , $\text{conv}(W)$, must contain $\text{conv}(X)$, since $X(:, j) = WH(:, j)$ for all j where H is column stochastic, while it is contained in the hyperrectangle $[a, b]$:

$$\text{conv}(X) \subseteq \text{conv}(W) \subseteq [a, b].$$

Imposing bounds on the approximation, via the element-wise constraints $a \leq WH \leq b$ for some $a, b \in \mathbb{R}$, was proposed in [27] and applied successfully to recommender systems. However, this model does not allow to interpret the basis factor, W , in the same way as the data. Some elements in W will probably be out the rating range because W is not directly constrained. Hence, the basis elements in W can only be interpreted as “eigen users”, while with BSSMF, the basis elements can be interpreted as virtual meaningful users. It is also difficult to interpret the factor H as it could contain negative contributions. In fact, only imposing $a \leq WH \leq b$ typically leads to dense factors W and H (that is, factors that do not contain many zeros, as opposed to sparse factors), while in most applications interpretability usually comes with a certain sparsity degree in at least one of the factors.

A closely related model that tackles blind source separation is bounded component analysis (BCA) proposed in [10], [14], where the sources are assumed to belong to compact sets (hyperrectangle being a special case), while no constraints is imposed on the mixing matrix. Again, without any constraints on the mixing matrix, BCA will generate dense factors with negative linear combinations which are difficult to interpret. Let us note that their motivation is different than ours, as their objective is to extract mixed sources, while ours is to extract interpretable features and decompose data through them. In [39], the authors also proposed a blind source separation algorithm for bounded sources based on geometrical concepts. The mixtures are assumed to belong to a parallelogram. The proposed separation technique is relies on mapping this parallelogram to a rectangle. Again, their objective is to extract mixed sources. Nonetheless, working with a domain different than a hyperrectangle could be of interest for future work.

c) Identifiability: Identifiability is key in practice as it allows to recover the ground truth that generated the data; see the discussion in [16], [29] and the references therein. A drawback of

SSMF is that it is never identifiable, see Section IV-A for further details. On the counterpart NMF can be identifiable, which is discussed in Section IV-B. Nonetheless, the conditions are not mild. For NMF to be identifiable, it is necessary that the supports of the columns of W (that is, the set of non-zero entries) are not contained in one another (this is called a Sperner family), and similarly for the supports of the rows of H ; see, e.g., [30], [40]. This requires the presence of zeros in each column of W and row of H , which can be a strong condition in some applications. For example, in hyperspectral unmixing, W is typically not sparse because it recovers spectral signatures of constitutive materials which are typically positive. Although the conditions for NMF (and SSMF) to be identifiable can be weakened using additional constraints and regularization terms, it then requires hyperparameter tuning procedures. In [43], they propose a model where the columns of H belong to a polytope. Using a maximum volume criterion on the convex hull of H , their model is identifiable under the condition that the convex hull of H contains the ellipsoid of maximum volume inscribed in the constraining polytope. The use of the maximum volume criterion also requires hyperparameter tuning. In [10], [14], the sources are identifiable by optimizing some geometric criterion, respectively minimizing a perimeter, and maximizing the ratio between the volume of an ellipsoid and the volume of a hyperrectangle. These identifiability conditions are not relevant to our model. As we will see in Section IV, BSSMF is identifiable under relatively mild conditions, while it does not require parameter tuning, as opposed to most regularized structured matrix factorization models that are identifiable. Let us note that it is also possible to formulate identifiable nonlinear matrix approximation models like the bilinear model of [12], but this is out of the scope of this article.

d) Robustness to overfitting: Another drawback of NMF and SSMF is that they are rather sensitive to the choice of r . When r is chosen too large, these two models are over-parameterized and will typically lead to overfitting. This is a well-known behaviour that can be addressed with additional regularization terms that need to be fine tuned [41]. As we will see experimentally in Section V-C for matrix completion, without any parameter tuning, BSSMF is much more robust to overfitting than NMF and unconstrained matrix factorization. The reason is that the additional bound constraints on W and sum-to-one constraint on H prevents columns of W and of WH from going outside the feasible range, $[a, b]$. In turn, BSSMF will be less sensitive to noise and an overestimation of r . For example, an outlier that falls outside the feasible set $[a, b]$ will not pose problems to BSSMF, while it may significantly impact the NMF and SSMF solutions.

III. INERTIAL BLOCK-COORDINATE DESCENT ALGORITHM FOR BSSMF

In this article, we consider the following BSSMF problem

$$\min_{W, H} g(W, H) := \frac{1}{2} \|X - WH\|_F^2$$

such that $W(:, k) \in [a, b]$ for all k ,

$$H \geq 0, \text{ and } H^\top e = e, \quad (2)$$

that uses the squared Frobenius norm to measure the error of the approximation.

A. Proposed Algorithm

Most NMF algorithms rely on block coordinate descent methods, that is, they update a subset of the variables at a time, such as the popular multiplicative updates of Lee and Seung [33], the hierarchical alternating least squares algorithm [8], [19], and a fast gradient based algorithm [23]; see, e.g., [20, Chapter 8] and the references therein for more detail. More recently, an inertial block majorization minimization framework for non-smooth non-convex optimization (TITAN) was introduced in [25] and has been shown to be particularly powerful to solve matrix and tensor factorization problems [24], [38], [46].

To solve (2), we therefore apply TITAN which updates one block W or H at a time while fixing the value of the other block. In order to update W (resp. H), TITAN chooses a block surrogate function for W (resp. H), embeds an inertial term to this surrogate function and then minimizes the obtained inertial surrogate function. We have $\nabla_W g(W, H) = -(X - WH)H^\top$ which is Lipschitz continuous in W with the Lipschitz constant $\|HH^\top\|$. Similarly, $\nabla_H g(W, H) = -W^\top(X - WH)$ is Lipschitz continuous in H with constant $\|W^\top W\|$. Hence, we choose the Lipschitz gradient surrogate for both W and H and choose the Nesterov-type acceleration as analysed in [25, Section 4.2.1] and [25, Remark 4.1], see also [25, Section 6.1] and [46] for similar applications.

In the case of missing entries in X , let us consider the more general model

$$\min_{W, H} g(W, H) := \frac{1}{2} \|M \circ (X - WH)\|_F^2$$

$$\text{such that } W(:, k) \in [a, b] \text{ for all } k,$$

$$H \geq 0, \text{ and } H^\top e = e, \quad (3)$$

where \circ corresponds to the Hadamard product, and M is a weight matrix which can model missing entries using $M(i, j) = 0$ when $X(i, j)$ is missing, and $M(i, j) = 1$ otherwise. It can also be used in other contexts; see, e.g., [18], [21], [42]. TITAN can also be used to solve (3), where the gradients are equal to $\nabla_W g(W, H) = -(M \circ (X - WH))H^\top$ and $\nabla_H g(W, H) = -W^\top(M \circ (X - WH))$. We acknowledge that the identifiability result that will be presented in Section IV does not hold for the case where some data are missing, this is an interesting direction of future research. Algorithm 1 describes TITAN for solving the general problem (3), where $[\cdot]_b^a$ is the column-wise projection on $[a, b]$ and $[\cdot]_{\Delta^r}$ is the column wise projection on the simplex Δ^r . When some data are missing, the Lipschitz constant of the gradients relatively to W and H could be smaller than $\|HH^\top\|$ and $\|W^\top W\|$, respectively. Relatively to H for instance, a smaller Lipschitz constant would be $\max_i \|W^\top(M(:, i)e^\top) \circ W\|$. We arbitrarily choose to keep $\|HH^\top\|$ and $\|W^\top W\|$ even when some data are missing since those values are faster to compute. Due to our derived algorithm being a particular instance of TITAN with Lipschitz

Algorithm 1: BSSMF.

input : Input data matrix $X \in \mathbb{R}^{m \times n}$, bounds $a \leq b \in \mathbb{R}^m$, initial factors $W \in \mathbb{R}^{m \times r}$ s.t. $W(:, k) \in [a, b]$ for all k and simplex structured $H \in \mathbb{R}_+^{r \times n}$, weights $M \in [0, 1]^{m \times n}$

output: W and H

- 1 $\alpha_1 = 1, \alpha_2 = 1, W_{old} = W, H_{old} = H,$
 $L_W^{prev} = L_W = \|HH^\top\|_2, L_H^{prev} = L_H = \|W^\top W\|_2$
- 2 **repeat**
- 3 **while** *stopping criteria not satisfied* **do**
- 4 $\alpha_0 = \alpha_1, \alpha_1 = (1 + \sqrt{1 + 4\alpha_0^2})/2$
- 5 $\beta_W =$
 $\min \left[(\alpha_0 - 1)/\alpha_1, 0.9999\sqrt{L_W^{prev}/L_W} \right]$
- 6 $\bar{W} \leftarrow W + \beta_W(W - W_{old})$
- 7 $W_{old} \leftarrow W$
- 8 $W \leftarrow \left[\bar{W} + \frac{(M \circ (X - \bar{W}H))H^\top}{L_W} \right]_a^b$
- 9 $L_W^{prev} = L_W$
- 10 $L_H \leftarrow \|W^\top W\|_2$
- 11 **while** *stopping criteria not satisfied* **do**
- 12 $\alpha_0 = \alpha_2, \alpha_2 = (1 + \sqrt{1 + 4\alpha_0^2})/2$
- 13 $\beta_H =$
 $\min \left[(\alpha_0 - 1)/\alpha_2, 0.9999\sqrt{L_H^{prev}/L_H} \right]$
- 14 $\bar{H} \leftarrow H + \beta_H(H - H_{old})$
- 15 $H_{old} \leftarrow H$
- 16 $H \leftarrow \left[\bar{H} + \frac{W^\top (M \circ (X - W\bar{H}))}{L_H} \right]_{\Delta^r}$
- 17 $L_H^{prev} \leftarrow L_H$
- 18 $L_W = \|HH^\top\|_2$
- 19 **until** *some stopping criteria is satisfied*

gradient surrogates [25, Section 4.2], Algorithm 1 guarantees a subsequential convergence, that is, every limit point of the generated sequence is a stationary point of Problem (2). The Julia code for Algorithm 1 is available on gitlab¹ (a MATLAB code is also available on gitlab² but it does not handle missing data for now). We omit the implementation details here, but let us mention that when data are missing, our Julia implementation does not compute the Hadamard product with M explicitly but rather takes advantage of the sparsity of the data by using multithreading to improve the computational time. The projections $[\cdot]_a^b$ and $[\cdot]_{\Delta^r}$ are also computed using multithreading.

A simple choice to initialize the factors, W and H , in Algorithm 1 is to randomly initialize them: for all i , each entry of $W(i, :)$ is generated using the uniform distribution in the interval $[a_i, b_i]$, while H is generated using a uniform distribution in $[0, 1]^{r \times n}$ whose columns are then projected on the simplex Δ^r .

B. Accelerating BSSMF Algorithms via Data Centering

Not only the BSSMF model is invariant to translations of the input data (this is explained in details in Section IV-C), but also the optimization, because of the simplex constraints. In particular, for any $c \in \mathbb{R}$ and denoting J the matrix of all ones of appropriate dimension, minimizing

$$f(W, H) := \frac{1}{2} \|X - WH\|_F^2 \quad (4)$$

or

$$f_c(W, H) := \frac{1}{2} \|X - cJ - (W - cJ)H\|_F^2, \quad (5)$$

is equivalent in BSSMF, since $cJ = cJH$ as H is column stochastic. However, *outside the feasible set*, f and f_c do not have the same topology. Computing the gradients, we have $\nabla_H f(W, H) = W^\top(WH - X)$ which is Lipschitz continuous in H with the Lipschitz constant $\|W^\top W\|$, and $\nabla_H f_c(W, H) = W_c^\top(W_c H - X_c)$ which is Lipschitz continuous in H with the Lipschitz constant $\|W_c^\top W_c\|$, where $W_c = W - cJ$ and $X_c = X - cJ$. Particularly, for BSSMF, since W can be interpreted in the same way as X , we can expect $\text{mean}(X) \approx \text{mean}(W)$, where $\text{mean}(\cdot)$ is the empirical mean of the entries of the input. Consequently, if we choose $c = \text{mean}(X)$, we expect³ the Lipschitz constant $\|W_c^\top W_c\|$ to be smaller than $\|W^\top W\|$. A smaller Lipschitz constant means that, when updating H , the gradient steps are allowed to be larger without losing any convergence guarantee. Hence, with the right translation on our data X , the optimization problem on H is unchanged on the feasible set but Algorithm 1 can be accelerated.

Let us illustrate this behavior on a small example with $m = 2$, $n = 1$, $r = 2$. We choose

$$X = \begin{pmatrix} 0.4 & 0.3 \\ 0.7 & 0.2 \end{pmatrix} \begin{pmatrix} 0.4 \\ 0.6 \end{pmatrix}.$$

We fix

$$W = \begin{pmatrix} 0.4 & 0.3 \\ 0.7 & 0.2 \end{pmatrix},$$

and try to solve, with respect to H , (4) and (5) with $c = \text{mean}(X)$. We perform 5 projected gradient steps and display the results on Fig. 1. On the left, 5 projected gradient steps are performed to update H based on the original data X . On the right, 5 projected gradient steps are performed to update H based on the centered data X . The feasible sets (in dash) are exactly the same, and therefore the optimal solutions are also the same. However, we observe that the landscape of the cost function outside the feasible region is smoother when the data are centered. This allows the solver to converge faster towards the optimal solution, as the gradients point better towards the optimal solution and the stepsizes are larger. The improvement regarding the convergence speed by applying centering with real data will probably not be as drastic as in this small example. Still, minimizing a smoother function is always advantageous, and this will be shown empirically on real data in Section III-C.

¹[Online]. Available: <https://gitlab.com/vuthanho/bssmf.jl>

²[Online]. Available: <https://gitlab.com/vuthanho/bounded-simplex-structured-matrix-factorization>

³We empirically noticed that, very often, with $c = \text{mean}(W)$, $\|W_c^\top W_c\|$ is of the same order of $\min_c \|W_c^\top W_c\|$.

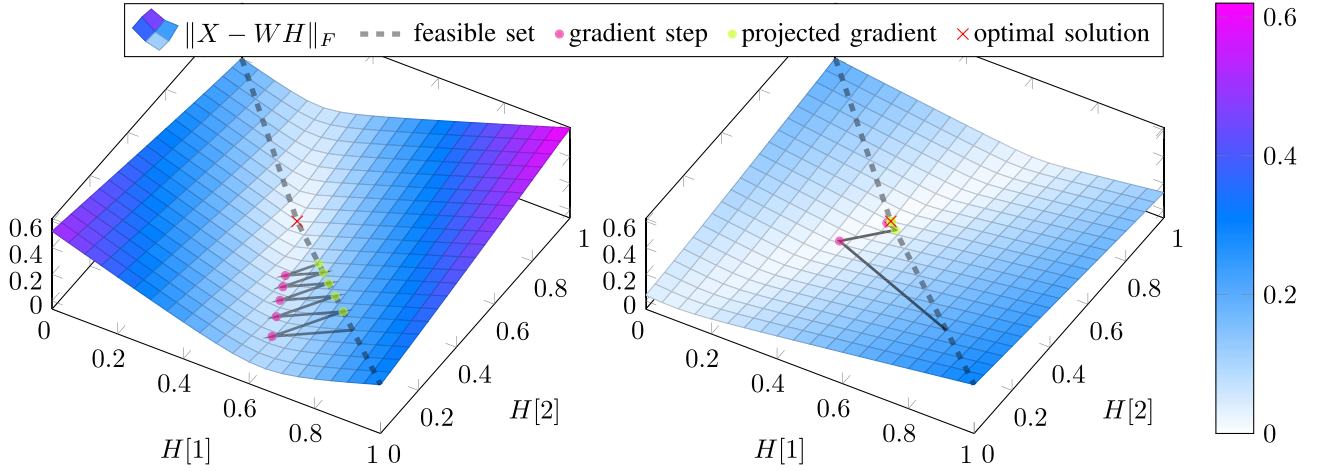


Fig. 1. Influence of centering the data on the cost function topology regarding H via a small example ($m = 2, r = 2, n = 1$). Left: without centering. Right: with centering. Five projected gradient steps are shown, decomposed through one gradient descent step followed by its projection onto the feasible set.

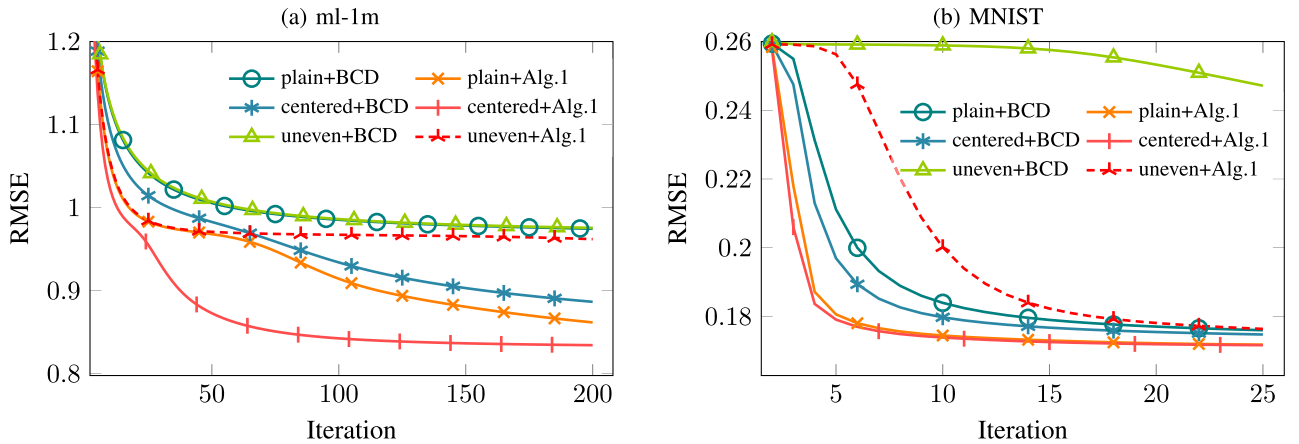


Fig. 2. Evolution of the training error for ml-1m and MNIST, averaged on 10 runs. For ml-1m, $r = 5$, 1 inner iteration. For MNIST, $r = 50$, 10 inner iterations.

C. Convergence Speed and Effect of Acceleration Strategies on Real Data

In this subsection, the goal is twofold: (1) show the effect of the extrapolation in TITAN by comparing Algorithm 1 to a non-extrapolated block coordinate descent, and (2) show the acceleration effect of centering the data.

We will apply the BSSMF model on MNIST and ml-1m (these two datasets are properly introduced respectively in Section V-A and Section V-C) in six different scenarios: 3 data related scenarios \times 2 algorithmic related scenarios. The data scenarios are raw data, centered data, and data to which a positive offset is added (respectively called ‘plain’, ‘centered’ and ‘uneven’ in Fig. 2). Note that ‘uneven’, that is, adding a positive constant, will worsen the landscape, as $\|W_c^T W_c\|$ will increase (since $a \geq 0$). This scenario will allow to further validate our observation about the acceleration effect of preprocessing. For each data case, 2 algorithms are tested: (1) Algorithm 1, and (2) a standard block coordinate descent (BCD) which is Algorithm 1 where the β ’s are fixed to 0; this corresponds to the popular

proximal alternating linearized minimization (PALM) algorithm [5]. When the algorithms are compared on the same data scenario, Algorithm 1 always converges faster and to a better solution than BCD. We also observe that when the data are centered, applying the same algorithm always lead to faster convergence than on the plain case. On ml-1m (Fig. 2(a)), applying BCD on the centered data is almost as fast as applying Algorithm 1 in the plain case, meaning that a good preprocessing is almost as important as a good acceleration strategy. If we look at the root-mean-square error (RMSE) on the test set, centered data are even more important than a good acceleration strategy. Actually, at the end of the experiment from Fig. 2(a), on the test set, centered+BCD has an RMSE of 0.91 while plain+Algorithm 1 has an RMSE of 0.94. Still on ml-1m, Algorithm 1 benefits remarkably from centering the data in comparison to the plain case. We also note an improvement on MNIST (Fig. 2(b)) which is not as noticeable as on ml-1m (probably because this problem is less difficult to solve, which is corroborated by the number of iterations required to converge). Here, the uneven case is just shown to remind that the data points cloud position is of much

TABLE I

COMPUTATION TIME OF ALGORITHM 1 IN THE EXPERIMENT SETTINGS OF FIG. 2(a) DEPENDING ON THE NUMBER OF USED THREADS

# threads	1	2	4	6	8	10	12
time (s)	30.53	5.14	2.98	2.85	3.00	2.78	3.31

TABLE II

COMPUTATION TIME OF ALGORITHM 1 IN THE EXPERIMENT SETTINGS OF FIG. 2(b) DEPENDING ON THE NUMBER OF USED THREADS

# threads	1	2	4	6	8	10	12
time (s)	27.79	21.92	16.67	15.22	15.73	16.01	16.65

concern and should not be neglected. Globally, regardless of the dataset, applying Algorithm 1 on centered data is the best strategy as compared with using plain data. As a consequence, it will be our default choice for the experiments in Section V.

As mentioned above, when entries are missing, Algorithm 1 can take advantage of the sparsity of the data and uses multithreading. We report in Table I the computation time of Algorithm 1 in the experiment settings of Fig. 2(a), given by the macro @btime from the package BenchmarkTools.jl. When the dataset is full, like with MNIST, multithreading is only used for the projections $[\cdot]_b^a$ and $[\cdot]_{\Delta r}$. The computation time in the settings of the experiment in Fig. 2(b) is reported in Table II. Note that there is no distinction between Algorithm 1 and BCD in terms of computation time because the computation of the acceleration is negligible compared to the other computations.

IV. IDENTIFIABILITY

Let us first define a factorization model.

Definition 2 (Factorization model): Given a matrix $X \in \mathbb{R}^{m \times n}$, and an integer $r \leq \min(m, n)$, a factorization model is an optimization model of the form

$$\begin{aligned} & \min_{W \in \mathbb{R}^{m \times r}, H \in \mathbb{R}^{r \times n}} g(W, H) \\ & \text{such that } X = WH, \\ & W \in \Omega_W \text{ and } H \in \Omega_H, \end{aligned} \quad (6)$$

where $g(W, H)$ is some criterion, and Ω_W and Ω_H are the feasible sets for W and H , respectively.

Let us define the identifiability of a factorization model, and essential uniqueness of a pair (W, H) .

Definition 3 (Identifiability / Essential uniqueness): Let $X \in \mathbb{R}^{m \times n}$, and $r \leq \min(m, n)$ be an integer. Let (W, H) be a solution to a given factorization model (6). The pair (W, H) is essentially unique for the factorization model (6) of matrix X if and only if any other pair $(W', H') \in \mathbb{R}^{m \times r} \times \mathbb{R}^{r \times n}$ that solves the factorization model (6) satisfies, for all k ,

$$W'(:, k) = \alpha_k W(:, \pi(k))$$

and

$$H'(k, :) = \alpha_k^{-1} H(\pi(k), :),$$

where π is a permutation of $\{1, 2, \dots, r\}$, and $\alpha_k \neq 0$ for all k . In other terms, (W', H') can only be obtained as a permutation

and scaling of (W, H) . In that case, the factorization model is said to be identifiable for the matrix X .

A key question in theory and practice is to determine conditions on X , g , Ω_W and Ω_H that lead to identifiable factorization models; see, e.g., [16], [29] for discussions.

In the next three sections, we discuss the identifiability of SSMF, NMF, and BSSMF.

A. Simplex-Structured Matrix Factorization (SSMF)

Without further requirements, SSMF is never identifiable; which follows from a result for semi-NMF which is a factorization model that requires only one factor, H , to be nonnegative [22]. Let $X = WH$ be an SSMF of X . We can obtain other SSMF of X using the following transformation: for any $\alpha \geq 0$, let

$$W(\alpha) := W \left((1 + \alpha)I - \frac{\alpha}{r}J \right),$$

and

$$\begin{aligned} H(\alpha) &:= \left(\frac{1}{1 + \alpha}H + \frac{\alpha}{(1 + \alpha)r}J \right) \\ &= \left(\frac{1}{1 + \alpha}I + \frac{\alpha}{(1 + \alpha)r}J \right) H, \end{aligned}$$

where I is the identity matrix of appropriate dimension, and the last equality follows from $e^\top H = e^\top$. The matrix $H(\alpha)$ is column stochastic since H and $\frac{J}{r}$ are. One can check that $(W(\alpha), H(\alpha))$ is not a permutation and scaling of (W, H) for $\alpha > 0$, while $WH = W(\alpha)H(\alpha)$ since⁴

$$\begin{aligned} A(\alpha) &:= \left((1 + \alpha)I - \frac{\alpha}{r}J \right)^{-1} \\ &= \frac{1}{1 + \alpha}I + \frac{\alpha}{(1 + \alpha)r}J. \end{aligned}$$

Geometrically, to obtain $W(\alpha)$, the columns of W are moved towards the exterior of $\text{conv}(W)$ and hence the convex hull of the column of $W(\alpha)$ contains the convex hull of the columns of W and hence contains $\text{conv}(X)$. This follows from the fact that $W = W(\alpha)A(\alpha)$, where A is column stochastic.

To obtain identifiability of SSMF, one needs either to impose additional constraint on W and/or H such as sparsity [1], or look for a solution minimizing a certain function g . In particular, the solution (W, H) that minimizes the volume of the convex hull of the columns of W (see Theorem 1 below for a formula) is essentially unique given that H satisfies the so-called sufficiently scattered condition (SSC). The SSC is defined as follows.

Definition 4 (Sufficiently scattered condition): The matrix $H \in \mathbb{R}_+^{r \times n}$ is sufficiently scattered if the following two conditions are satisfied:

[SSC1]: $\mathcal{C} = \{x \in \mathbb{R}_+^r \mid e^\top x \geq \sqrt{r-1}\|x\|_2\} \subseteq \text{cone}(H)$, where $\text{cone}(H) = \{x \mid x = Hy, y \geq 0\}$ denotes the conical hull of the columns of H .

⁴This is an invertible M-matrix, with positive diagonal elements and negative off-diagonal elements, whose inverse is nonnegative [4].

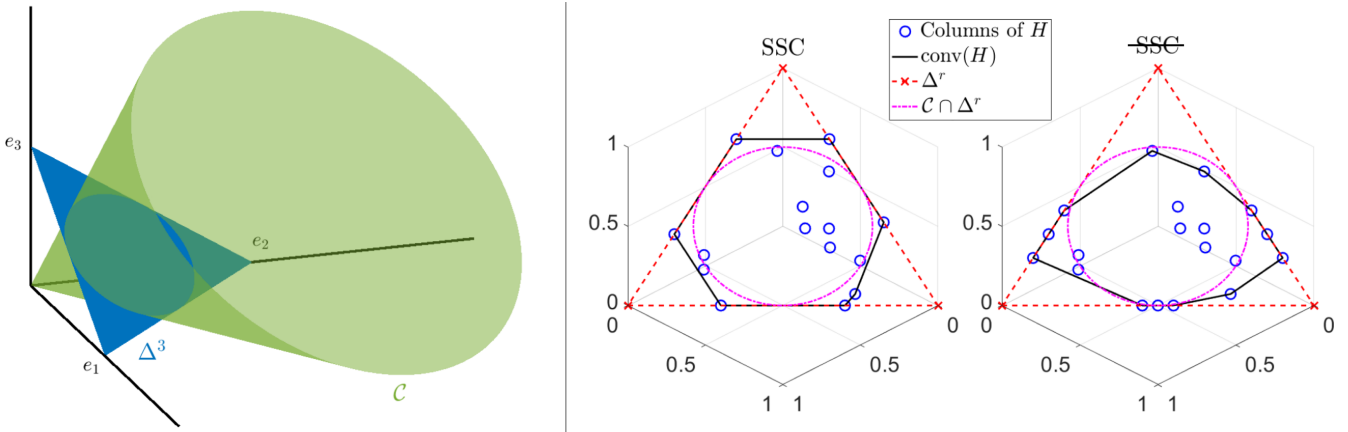


Fig. 3. Illustration of the SSC in three dimensions. On the left: the sets Δ^3 and \mathcal{C} , they intersect at $(0,0.5,0.5)$, $(0.5,0,0.5)$, and $(0.5,0.5,0)$. (This left figure is similar to [26, Fig. 2] and we are grateful to the authors for providing us with the code to generate it.) On the right: examples of a matrix $H \in \mathbb{R}^{3 \times n}$ satisfying the SSC (left), and not satisfying the SSC (right). The left (resp. right) figure is adapted from [26] (resp. [1]).

[SSC2]: There does not exist any orthogonal matrix Q such that $\text{cone}(H) \subseteq \text{cone}(Q)$, except for permutation matrices. (An orthogonal matrix Q is a square matrix such that $Q^\top Q = I$),

SSC1 requires the columns of H to contain the cone \mathcal{C} , which is tangent to every facet of the nonnegative orthant; see Fig. 3. Hence, satisfying SSC1 requires some degree of sparsity as H needs to contain at least $r - 1$ zeros per row [20, Th. 4.28]. SSC2 is a mild regularity condition which is typically satisfied when SSC1 is satisfied. For more discussions on the SSC, we refer the interested reader to [16] and [20, Chapter 4.2.3], and the references therein. For SSMF, we have the following identifiability result.

Theorem 1: [17], [36] **The minimum-volume SSMF factorization model,**

$$\min_{W \in \mathbb{R}^{m \times r}, H \in \mathbb{R}^{r \times n}} \det(W^\top W)$$

such that $X = WH$ and $H(:, j) \in \Delta^r$ for all j ,

is identifiable if the pair $(W, H) \in \mathbb{R}^{m \times r} \times \mathbb{R}^{r \times n}$ satisfies $\text{rank}(W) = r$ and H is sufficiently scattered.

The quantity $\det(W^\top W)$ measures the volume of $\text{conv}(W)$ within the column space of W . Note that this result has been generalized to the case where the columns of H belong to a given polytope instead of the probability simplex; see [43].

In practice, because of noise and model misfit, SSMF optimization models need to balance the data fitting term which measures the discrepancy between X and WH , and the volume regularization for $\text{conv}(W)$. Typically a problem with objective function of the form

$$\|X - WH\|_F^2 + \lambda \det(W^\top W),$$

is solved. This requires the tuning of the parameter λ , which is a nontrivial process [2], [49].

B. Nonnegative Matrix Factorization (NMF)

As opposed to SSMF, NMF decompositions can be identifiable without the use of additional requirements. The first identifiability result was proposed in [13]. Their conditions, based

on separability, are quite strong. In the context of nonnegative source separation, [40] proposed some necessary conditions for the uniqueness of the solution. One of the most relaxed sufficient condition for identifiability is based on the SSC.

Theorem 2: [26, Theorem 4] If $W^\top \in \mathbb{R}^{r \times m}$ and $H \in \mathbb{R}^{r \times n}$ are sufficiently scattered, then the Exact NMF (W, H) of $X = WH$ of size $r = \text{rank}(X)$ is essentially unique.

In practice, it is not likely for both W^\top and H to satisfy the SSC. Typically H will satisfy the SSC, as it is typically sparse. However, in many applications, W^\top will not satisfy the SSC; in particular in applications where W is not sparse, e.g., in hyperspectral unmixing, recommender systems, or imaging. This is why regularized NMF models have been introduced, including sparse and minimum-volume NMF. We refer the interested reader to [20, Chapter 4] for more details.

C. Bounded Simplex-Structured Matrix Factorization (BSSMF)

A main motivation to introduce BSSMF is that it is identifiable under weaker conditions than NMF. We now state our main identifiability result for BSSMF, it is a consequence of the identifiability result of NMF and the following simple observation: $X = WH$ is a BSSMF for the interval $[a, b]$ implies that $be^\top - X = (be^\top - W)H$ and $X - ae^\top = (W - ae^\top)H$ are Exact NMF decompositions.

Theorem 3: Let $W \in \mathbb{R}^{m \times r}$ and $H \in \mathbb{R}^{r \times n}$ satisfy $W(:, k) \in [a, b]$ for all k for some $a \leq b$, $H \geq 0$, and $H^\top e = e$. If $\begin{pmatrix} W - ae^\top \\ be^\top - W \end{pmatrix}^\top \in \mathbb{R}^{r \times 2m}$ and $H \in \mathbb{R}^{r \times n}$ are sufficiently scattered, then the BSSMF (W, H) of $X = WH$ of size $r = \text{rank}(X)$ for the interval $[a, b]$ is essentially unique.

Proof: Let (W, H) be a BSSMF of X for the interval $[a, b]$. As in the proof of Lemma 2, we have

$$X - ae^\top = WH - ae^\top = (W - ae^\top)H,$$

since $e^\top = e^\top H$. This implies that $(W - ae^\top, H)$ is an Exact NMF of $X - ae^\top$, since $W - ae^\top$ and H are nonnegative.

Similarly, we have

$$be^\top - X = be^\top - WH = (be^\top - W)H,$$

which implies that $(be^\top - W, H)$ is an Exact NMF of $be^\top - X$, since $be^\top - W \geq 0$. Therefore, we have the Exact NMF

$$\begin{pmatrix} X - ae^\top \\ be^\top - X \end{pmatrix} = \begin{pmatrix} W - ae^\top \\ be^\top - W \end{pmatrix} H.$$

By Theorem 2, this Exact NMF is unique if $\begin{pmatrix} W - ae^\top \\ be^\top - W \end{pmatrix}^\top$ and H satisfy the SSC. This proves the result: in fact, the derivations above hold for any BSSMF of X . Hence, if (W, H) was not an essentially unique BSSMF of X , there would exist another Exact NMF of $\begin{pmatrix} W - ae^\top \\ be^\top - W \end{pmatrix}^\top$, not obtained by permutation and scaling of $\begin{pmatrix} W - ae^\top \\ be^\top - W \end{pmatrix}$, a contradiction. \square

Let us note that $W - ae^\top$ and H being SSC, or $be^\top - W$ and H being SSC, are also sufficient conditions for identifiability. These conditions are stronger, as $W - ae^\top$ being SSC or $be^\top - W$ being SSC implies that $\begin{pmatrix} W - ae^\top \\ be^\top - W \end{pmatrix}^\top$ is SSC. However, $\begin{pmatrix} W - ae^\top \\ be^\top - W \end{pmatrix}^\top$ does not imply that $W - ae^\top$ or $be^\top - W$ is SSC.

The condition that $\begin{pmatrix} W - ae^\top \\ be^\top - W \end{pmatrix}^\top$ is SSC is much weaker than requiring W^\top to be SSC in NMF. In fact, in NMF, W^\top is SSC requires that it contains zero entries (at least $r - 1$ per row [20, Th. 4.28]; this can also be seen on the right of Fig. 3 in the case $r = 3$). Since the SSC is only defined for nonnegative matrices and W^\top contains zeros, a has to be equal to the zero vector. In this case, W^\top being SSC implies that $W^\top - ea^\top$ is SSC, and hence the corresponding BSSMF is identifiable. However, the reverse is not true. In fact, $\begin{pmatrix} W - ae^\top \\ be^\top - W \end{pmatrix}^\top$ being SSC means that sufficiently many values in W are equal to its minimum and maximum bounds in a and b . For example, in recommender systems, with $W(i, j) \in [1, 5]$ for all (i, j) , many entries of W are expected to be equal to 1 or to 5 (the minimum and maximum ratings), so that $\begin{pmatrix} W - ae^\top \\ be^\top - W \end{pmatrix}^\top$ will contain many zero entries, and hence likely to satisfy the SSC [15]. On the other hand, W is positive, and hence it cannot be part of an essentially unique Exact NMF.

Let us illustrate the difference between NMF and BSSMF on a simple example.

Example 1 (Non-unique NMF vs. unique BSSMF): Let $\omega \in (0, 1)$ and let

$$A_\omega = \begin{pmatrix} \omega & 1 & 1 & \omega & 0 & 0 \\ 1 & \omega & 0 & 0 & \omega & 1 \\ 0 & 0 & \omega & 1 & 1 & \omega \end{pmatrix}.$$

For $\omega < 0.5$, A_ω satisfies the SSC, while it does not for $\omega \leq 0.5$; see [30, Example 3], [26, Example 2], [20, Example 4.16]. Let us take

$$H = 3A_{1/3} = \begin{pmatrix} 1 & 3 & 3 & 1 & 0 & 0 \\ 3 & 1 & 0 & 0 & 1 & 3 \\ 0 & 0 & 1 & 3 & 3 & 1 \end{pmatrix},$$

which satisfies the SSC, and

$$W^\top = 3A_{2/3} = \begin{pmatrix} 2 & 3 & 3 & 2 & 0 & 0 \\ 3 & 2 & 0 & 0 & 2 & 3 \\ 0 & 0 & 2 & 3 & 3 & 2 \end{pmatrix},$$

which does not satisfy the SSC, but has some degree of sparsity. The NMF of

$$X = WH = \begin{pmatrix} 11 & 9 & 6 & 2 & 3 & 9 \\ 9 & 11 & 9 & 3 & 2 & 6 \\ 3 & 9 & 11 & 9 & 6 & 2 \\ 2 & 6 & 9 & 11 & 9 & 3 \\ 6 & 2 & 3 & 9 & 11 & 9 \\ 9 & 3 & 2 & 6 & 9 & 11 \end{pmatrix}$$

is not essentially unique. For example,

$$X = \begin{pmatrix} 0 & 3 & 1 \\ 1 & 3 & 0 \\ 3 & 1 & 0 \\ 3 & 0 & 1 \\ 1 & 0 & 3 \\ 0 & 1 & 3 \end{pmatrix} \begin{pmatrix} 0 & 2 & 3 & 3 & 2 & 0 \\ 3 & 3 & 2 & 0 & 0 & 2 \\ 2 & 0 & 0 & 2 & 3 & 3 \end{pmatrix}$$

is another decomposition which cannot be obtained as a scaling and permutation of (W, H) .

However, the BSSMF of X is unique, taking $a_i = 0$ and $b_i = 3$ for all i . In fact, $(3 - W)^\top$ satisfies the SSC, as it is equal to $3A_{1/3}$, up to permutation of its columns:

$$\begin{aligned} 3 - W^\top &= \begin{pmatrix} 1 & 0 & 0 & 1 & 3 & 3 \\ 0 & 1 & 3 & 3 & 1 & 0 \\ 3 & 3 & 1 & 0 & 0 & 1 \end{pmatrix} \\ &= 3A_{1/3}(:, [4, 5, 6, 1, 2, 3]). \end{aligned}$$

Therefore, by Theorem 3, the BSSMF of X is unique.

a) Scaling ambiguity: BSSMF is in fact more than essentially unique in the sense of Definition 3. In fact, the scaling ambiguity can be removed because of H being simplex structured, as shown in the following lemma.

Lemma 1: Let $H \in \mathbb{R}^{r \times n}$ such that $e^\top H = e^\top$ and $\text{rank}(H) = r$. Let $D \in \mathbb{R}^{r \times r}$ be a diagonal matrix, and let $H' = DH$ be a scaling of the rows of H , and such that $e^\top H' = e^\top$. Then D must be the identity matrix, that is, $D = I$.

Proof: Let us denote $H^\dagger \in \mathbb{R}^{n \times r}$ the right inverse of H , which exists and is unique since $\text{rank}(H) = r$, so that $HH^\dagger = I$. We have

$$\begin{aligned} e^\top H' &= e^\top DH = e^\top \\ \Rightarrow e^\top DHH^\dagger &= e^\top H^\dagger = e^\top \\ \text{since } e^\top H^\dagger &= e^\top HH^\dagger = e^\top \\ \Rightarrow e^\top D &= e^\top \Rightarrow D = I. \end{aligned}$$

Note that this lemma does not require H , H' and D to be nonnegative. \square

b) Geometric interpretation of BSSMF: Solving BSSMF is equivalent to finding a polytope with r vertices within the hyperrectangle defined by $[a, b]$ that reconstructs as well as possible the data points. The fact that BSSMF is constrained

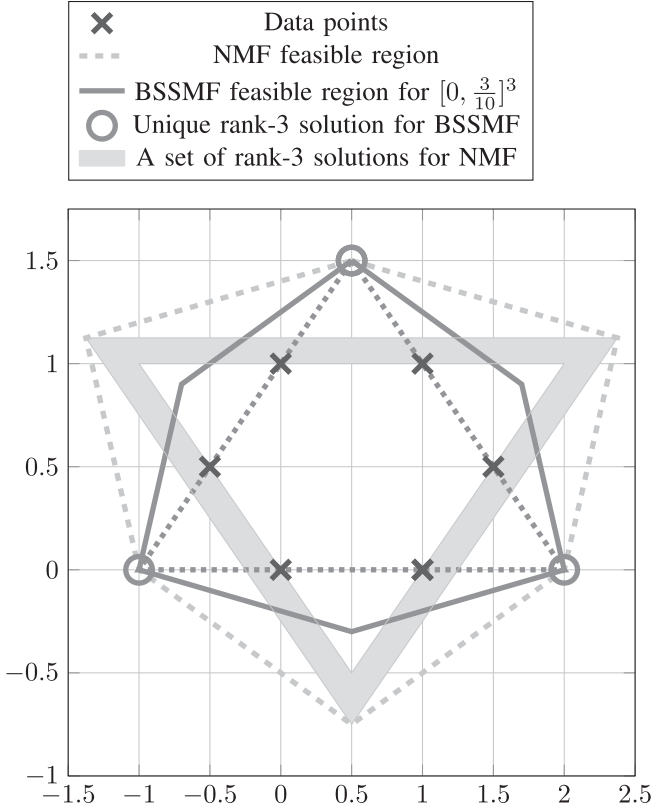


Fig. 4. Geometric interpretation of BSSMF for Example 1. Any triangle in the gray filled area containing the data points is a rank-3 solution for NMF. On the contrary, there is a unique rank-3 solution for BSSMF since there is a unique triangle containing the data points in the BSSMF feasible set.

within a hyperrectangle makes BSSMF more constrained than NMF, and hence more likely to be essentially unique. This will be illustrated empirically in Section V-B. Let us provide a toy example to better understand the distinction between NMF and BSSMF, namely let us use Example 1 with $W = \frac{3}{10}A_{2/3}$ and $H = \frac{2}{3}A_{1/2}$ so that $X = WH$ is column stochastic. Fig. 4 represents the feasible regions of NMF and BSSMF for the hypercube $[a, b] = [0, \frac{3}{10}]^3$ in a two-dimensional space within the affine hull of W ; see [20] for the details on how to construct such a representation. For this rank-3 factorization problem, solving NMF and BSSMF is equivalent to finding a triangle nested between the convex hull of the data points and the corresponding feasible region. BSSMF has a unique solution, that is, there is a unique triangle between the data points and the BSSMF feasible region. On the other hand, NMF is not identifiable: for example, any triangle within the gray area containing the data points is a solution.

In summary, for the BSSMF of $X = WH$ to be essentially unique, W must contain sufficiently many entries equal to the lower and upper bounds, while H must be sufficiently sparse.

c) *Choice of a and b :* In practice, if a and b are unknown, it may be beneficial to choose them such that as many entries of X are equal to the lower and upper bounds, and hence BSSMF is more likely to be identifiable. Let us denote $\tilde{a}_i = \min_j X(i, j)$ and $\tilde{b}_i = \max_j X(i, j)$ for all i , and let $X = WH$ be a BSSMF

for the hyperrectangle $[a, b]$. We have $\tilde{a} \geq a$ and $\tilde{b} \leq b$ since $H(:, j) \in \Delta^r$ for all j . Hence, without any prior information, it makes sense to use a BSSMF with interval $[\tilde{a}, \tilde{b}]$ which is contained in $[a, b]$.

Remark 2: Interestingly, as shown in Lemma 2 below, in the exact case, that is, when $X = WH$, we can assume w.l.o.g. that $[a_i, b_i] = [0, 1]$ for all i in BSSMF.

Lemma 2: Let $a \in \mathbb{R}^m$ and $b \in \mathbb{R}^m$ be such that $a_i < b_i$ for all i . The matrix $X = WH$ admits a BSSMF for the interval $[a, b]$ if and only if the matrix $\frac{[X - ae^\top]}{[(b-a)e^\top]}$ admits a BSSMF for the interval $[0, 1]^m$, where $\frac{[\cdot]}{[\cdot]}$ is the component-wise division of two matrices of the same size.

Proof: Let us show the direction \Rightarrow , the other is obtained exactly in the same way. Let the matrix $X = WH$ admit a BSSMF for the interval $[a, b]$. We have

$$X - ae^\top = WH - ae^\top = (W - ae^\top)H,$$

since $e^\top H = e^\top$, as H is column stochastic. This shows that $X' = X - ae^\top$ admits a BSSMF for the interval $[0, b - a]$ since $W' = (W - ae^\top) \in [0, b - a]$. For simplicity, let us denote $c = b - a > 0$. We have $X' = W'H$, while

$$\frac{[X - ae^\top]}{[(b-a)e^\top]} = \frac{[X']}{[ce^\top]} = \frac{[W'H]}{[ce^\top]} = \frac{[W']}{[ce^\top]}H,$$

because H is column stochastic. In fact, for all i, j ,

$$\begin{aligned} \frac{[W'H]_{i,j}}{[ce^\top]_{i,j}} &= \frac{\sum_k W'(k, i)H(k, j)}{c_i} \\ &= \sum_k \frac{W'(k, i)}{c_i} H(k, j) \\ &= \left(\frac{[W']}{[ce^\top]} H \right)_{i,j}. \end{aligned}$$

Hence $\frac{[X - ae^\top]}{[(b-a)e^\top]}$ admits a BSSMF for the interval $[0, 1]^m$ since H is column stochastic, and all columns of $\frac{[W']}{[ce^\top]} = \frac{[W - ae^\top]}{[(b-a)e^\top]}$ belong to $[0, 1]^m$. \square

Remark 3 (What if $a_i = b_i$ for some i ?): Lemma 2 does not cover the case $a_i = b_i$ for some i . In that case, we have $W(i, :) = a_i = b_i$ and therefore $X(i, :) = W(i, :)H = a_i e^\top = b_i e^\top$. This is not an interesting situation, and rows of X with identical entries can be removed. In fact, after the transformation $X - ae^\top$, these rows are identically zero.

Lemma 2 highlights another interesting property of BSSMF: as opposed to NMF, it is invariant to translations of the entries of the input matrix, given that a and b are translated accordingly. For example, in recommender systems datasets such as Netflix and MovieLens, $X(i, j) \in \{1, 2, 3, 4, 5\}$ for all i, j . Changing the scale, say to $\{0, 1, 2, 3, 4\}$, does not change the interpretation of the data, but will typically impact the NMF solution significantly⁵, while the BSSMF solution will be unchanged, if the interval is translated from $[1, 5]$ to $[0, 4]$ since H is invariant by translation on X . This property is in fact coming from SSMF.

⁵In fact, for NMF, it would make more sense to work on the datasets translated to $[0, 4]$, as it would potentially allow it to be identifiable: zeros in X imply zeros in W and H , which are therefore more likely to satisfy the SSC.

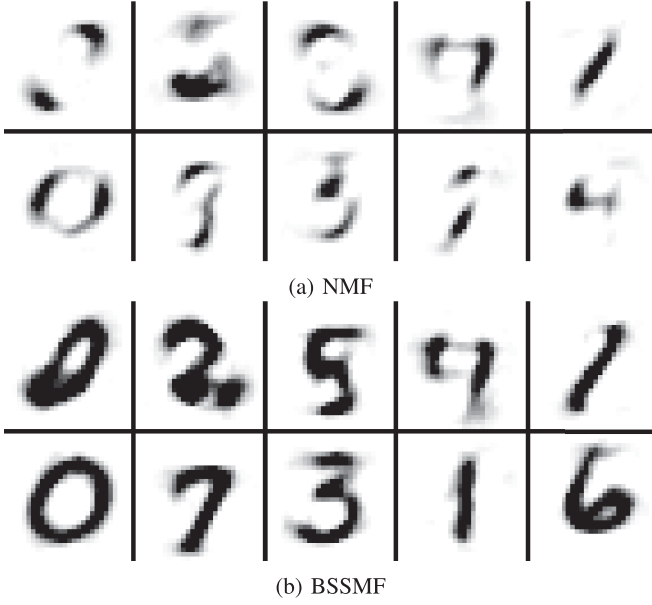


Fig. 5. Reshaped columns of the basis matrix W for $r = 10$ for MNIST with 500 digits.

V. NUMERICAL EXPERIMENTS

The goal of this section is to highlight the motivation points mentioned in Section II on real data sets. All experiments are run on a PC with an Intel(R) Core(TM) i7-9750H CPU @ 2.60 GHz and 16 GiB RAM. Let us recall that in order to retrieve NMF from Algorithm 1, the bounds need to be set to $(a, b) = (0, +\infty)$ and the projection step on the probability simplex in line 16 should be replaced by a projection on the nonnegative orthant. Hence, in our experiments, both NMF and BSSMF are solved with the same code implementation.

A. Interpretability

When applied on a pixel-by-image matrix, NMF allows to automatically extract common features among a set of images. For example, if each row of X is a vectorized facial image, the rows of W will correspond to facial features [32].

Let us compare NMF with BSSMF on the widely used MNIST handwritten digits dataset (60,000 images, 28×28 pixels) [31]. Each column of X is a vectorized handwritten digit. For BSSMF to make more sense, we preprocess X so that the intensities of the pixels in each digit belong to the interval $[0, 1]$ (first remove from $X(:, j)$ its minimum entry, then divide by the maximum entry minus the minimum entry).

Let us take a toy example with $n = 500$ randomly selected digits and $r = 10$, in order to visualize the natural interpretability of BSSMF. The choice of n is made solely for computational time considerations. For larger n , Fig. 5(b) might change but we will not lose interpretability. Fig. 5(a) shows the features learned by NMF which look like parts of digits. On the other hand, the features learned by BSSMF in Fig. 5(b) look mostly like real digits, because of the bound constraint and the simplex structure. In fact, as it is well known [32] that NMF learns

$$\begin{array}{|c|} \hline 8 \\ \hline \end{array} = 0.33 \times \begin{array}{|c|} \hline 5 \\ \hline \end{array} + 0.13 \times \begin{array}{|c|} \hline 3 \\ \hline \end{array} + 0.36 \times \begin{array}{|c|} \hline 1 \\ \hline \end{array} + \dots$$

Fig. 6. Decomposition of an eight by BSSMF with $r = 10$.

part-based representations, in this case, parts of digits. In other words, the columns of W in NMF identify subset of pixels that are activated simultaneously in as many images as possible. Now, by the scaling degree of freedom, assume w.l.o.g. that $W(:, j) \in [0, 1]^m$ for all j in NMF. Since the columns of W are parts of digits, each digits will have to use several of these parts, with an intensity close to one, so that H will be far from being column stochastic. BSSMF, with the simplex constraint on H and the bound constraints on W , therefore cannot learn such a part-based representation. This is the reason why BSSMF learns more global features that, added on top of each other, reconstruct the digits. As it is shown in the MNIST experiment, these features look like digits themselves. Interestingly, if we progressively increase the upper bound, we would see that BSSMF progressively learns parts of digits, like NMF (using a lower bound of zero, that is, BSSMF with $[0, u]^m$ with $u \geq 1$). This is an indirect way of balancing the sparsity between W and H . The larger the upper bound, the more relaxed is BSSMF and hence the sparser W will be (given that the lower bound is 0). In Fig. 5(b), we distinguish numbers (like 7, 3 and 6). From a clustering point of view, this is of much interest because a column of H which is near a ray of the probability simplex can directly be associated with the corresponding digit from W . In this toy example, due to r being small, an 8 cannot be seen. Nonetheless, an eight can be reconstructed as the weighted sum of the representations of a 5, a 3 and an italic 1; see Fig. 6 for an example. Note that since BSSMF is more constrained than NMF, its reconstruction error might be larger than that of NMF. For our example ($r = 10$), BSSMF has relative error $\|X - WH\|_F / \|X\|_F$ of 61.56%, and NMF of 59.04%. This is not always a drawback. In some applications, due to the presence of noise, although the reconstruction error of BSSMF is larger than that of NMF, the accuracy of the estimated factors W and H could be better, because it uses the prior information and is less prone to overfitting and less sensitive to outliers. See also the discussion in Section V-C where NMF has a lower RMSE than BSSMF on the training set, but a larger RMSE than BSSMF on the test set. Note that we also compute NMFs using Algorithm 1 where the projections are performed on the nonnegative orthant, instead of on the bounded set for W and on the probability simplex for H . The stopping criteria in line 19, 3 and 11 of Algorithm 1 are a maximum number of iterations equal to 500, 20 and 20, respectively, for both algorithms.

B. Identifiability

As it is NP-hard to check the SSC [26], we perform experiments on MNIST and synthetic data where only a necessary condition for SSC1 is verified, namely [20, Alg. 4.2].

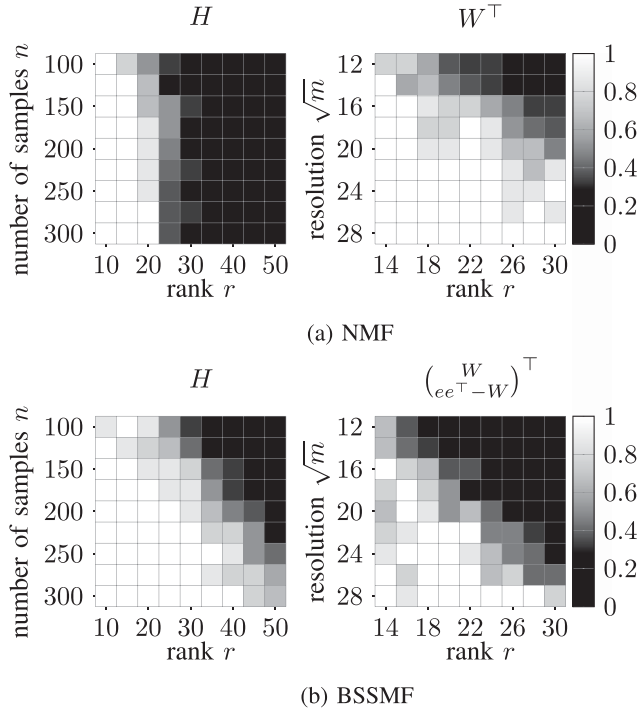


Fig. 7. Ratio, over 10 runs, of the factors generated by NMF in Fig. 7(a) and by BSSMF in Fig. 7(b) that satisfy the necessary condition for SSC1 (white squares indicate that all matrices meet the necessary condition, black squares that none do).

a) MNIST dataset: On MNIST, to see when H satisfies this condition, we first vary n from 100 to 300 for m fixed ($=28 \times 28$). For W^T , we fix n to 300, and downscale the resolution m from 28×28 to 12×12 with a linear interpolation (`imresize3` in MATLAB), and the rank r is varied from 12 to 30. Recall that both factors need to satisfy the SSC to correspond to an essentially unique factorization. In Fig. 7(a), we see that W^T of NMF often satisfies the necessary condition. This is due to NMF learning “parts” of objects [32], which are sparse by nature, and sparse matrices are more likely to satisfy the SSC (Definition 4). On the contrary, even for a relatively large n , H is too dense to satisfy the necessary condition. For $r \geq 30$, the factor H generated by NMF never satisfies the condition. Meanwhile, in Fig. 7(b) we see that H of BSSMF always satisfies the condition when $n \geq 225$ for $r = 30$ and more generally, if n and m are large enough, both H and $(\frac{W}{J-W})^T$ satisfy the necessary condition. This substantiates that BSSMF provides essentially unique factorizations more often than NMF does.

b) Synthetic datasets: Let us now perform an experiment to show how BSSMF is more likely than NMF to recover factors closer to the true ones, even when the sufficient conditions for identifiability are not satisfied. As there is no groundtruth for NMF and BSSMF on MNIST, we generate synthetic data as follows. Our synthetic datasets are of size 100×100 , and their factorization rank is 10. The matrix H is generated randomly with values uniformly distributed between zero and one, and we randomly set 30% of the values to zero. This allows us to ensure

that H satisfies the SSC. The reason behind ensuring that H is SSC is that both NMF (Theorem 2) and BSSMF (Theorem 3) require that H satisfies the SSC.⁶ As we want to emphasize on how likely it is to retrieve the true factors for NMF and BSSMF, we make sure that their common conditions for identifiability are satisfied. The matrix W is also generated randomly with values uniformly distributed between zero and one, and we then set a percentage of $p_{0,1}$ of the entries to zero and one, with the same probability to be equal to zero or one. Hence, $p_{0,1}$ percent of the values in W touches the lower and upper bounds in BSSMF. Finally, we let $X = WH$ to get our synthetic data. We solve NMF and BSSMF on X using Algorithm 1. To assess the quality of the solutions, we report the average of the mean removed spectral angle (MRSA) between the columns of the true W and the estimated W (after an optimal permutation of the columns), as this is standard in the NMF literature. Given any two vectors a and b , their MRSA is defined as

$$\text{MRSA}(a, b) = \frac{100}{\pi} \arccos \left(\frac{(a - \bar{a}e)^T(b - \bar{b}e)}{\|a - \bar{a}e\|_2 \|b - \bar{b}e\|_2} \right) \in [0, 100],$$

where $\bar{\cdot}$ is the average of the entries of a vector.

We vary the percentage $p_{0,1}$ of values touching the lower and upper bounds in W (namely, 0 and 1) from 0% to 30% with a 5% increment. For each value of $p_{0,1}$, the test is performed 20 times. Let us note that among the generated true W 's, between $p_{0,1} = 0\%$ and $p_{0,1} = 15\%$, $(\frac{W}{J-W})^T$ never satisfies the necessary conditions for SSC1. For $p_{0,1} = 20\%$, 3 out of the 20 generated $(\frac{W}{J-W})^T$ satisfies the necessary conditions for SSC1, 10 out of 20 for $p_{0,1} = 25\%$, and 17 out of 20 for $p_{0,1} = 30\%$. Let us also note that for all values of $p_{0,1}$ within the considered range, W never satisfies the necessary conditions for SSC1. The distribution of the average MRSA is reported in Fig. 8. Clearly, the MRSA is always smaller for BSSMF compared to NMF, even when the necessary conditions for SSC1 are not satisfied for $(\frac{W}{J-W})^T$; this is because the feasible set of BSMF is contained in that of NMF, and hence the generated factors are more likely to be closer to the groundtruth. This also illustrates that the conditions of Theorem 3 for the identifiability of BSSMF are only sufficient, since BSSMF finds solutions with MRSA close to machine epsilon when these conditions are not fulfilled.

C. Robustness to Overfitting

In this section we compare unconstrained matrix factorization (MF), NMF and BSSMF on the matrix completion problem; more precisely, on rating datasets for recommendation systems. Let X be an item-by-user matrix and suppose that user j has rated item i , that rating would be stored in $X_{i,j}$. The matrix X is then highly incomplete since a user has typically only rated a few of the items. In this context, NMF looks for nonnegative factors W and H such that $M \circ X \approx M \circ (WH)$, where $M_{i,j}$ is equal to 1 when user j rated item i and is equal to 0

⁶In this experiment, because n and r are smaller, we could checked that the SSC is satisfied (not a necessary condition), using Gurobi (<https://www.gurobi.com/>), a global optimization software.

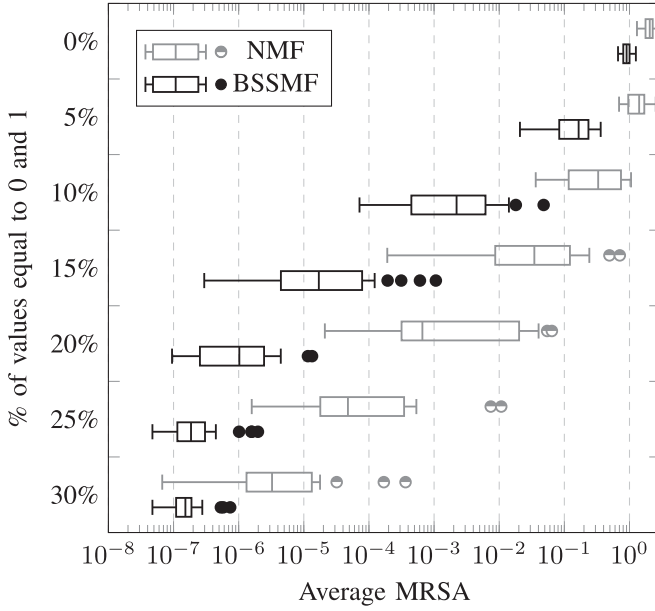


Fig. 8. Boxplots of the average MRSA between the true W and the estimated W by NMF and BSSMF for the hypercube $[0, 1]^{100}$ over 20 trials, depending on the percentage, $p_{0,1}$, of values equal to 0 and 1 in the true W .

otherwise. A missing rating $X_{i,j}$ is then estimated by computing $W(i, :)H(:, j)$. Features learned by NMF on rating datasets tend to be parts of typical users. Yet, the nonnegative constraint on the factors hardly makes the features interpretable by a practitioner. Suppose that the rating a user can give is an integer between 1 and 5 like in many rating systems, NMF can learn features whose values may fall under the minimum rating 1 or may exceed the maximum rating 5. Consequently, the features cannot directly be interpreted as typical users. On the contrary, with BSSMF, the extracted features will directly be interpretable if the lower and upper bounds are set to the minimum and maximum ratings. On top of that, BSSMF is expected to be less sensitive to overfitting than NMF since its feasible set is more constrained.

This last point will be highlighted in the following experiment on the ml-1m dataset,⁷ which contains 1 million ratings from 6040 users on 3952 movies. As in [35], we split the data in two sets : a training set and a test set. The test set contains 500 users. We also remove any movie that has been rated less than 5 times from both the training and test sets. For the test set, 80% of a user's ratings are considered as known. The remaining 20% are kept for evaluation. During the training, we learn W only on the training set. During the testing, the learned W is used to predict those 20% kept ratings of the test set by solving the H part only on the 80% known ratings. This simulates new users that were not taken into account during the training, but for whom we would still want to predict the ratings. The reported RMSEs are computed on the 20% kept ratings of the test set. In order to challenge the overfitting issue, we vary r in $\{1, 5, 10, 20, 50, 100\}$ for BSSMF, NMF and an unconstrained MF which are all computed

TABLE III
RMSE ON THE TEST SET ACCORDING TO r , AVERAGED \pm STANDARD DEVIATION ON 10 RUNS ON ML-1M

r	BSSMF	NMF	MF
1	$0.97 \pm 2 \cdot 10^{-5}$	0.88 ± 0.002	$0.91 \pm 5 \cdot 10^{-6}$
5	0.87 ± 0.001	0.87 ± 0.003	0.87 ± 0.003
10	0.86 ± 0.002	0.87 ± 0.001	0.87 ± 0.002
20	0.87 ± 0.002	0.87 ± 0.002	0.88 ± 0.002
50	0.88 ± 0.002	0.90 ± 0.004	0.93 ± 0.004
100	0.89 ± 0.003	0.92 ± 0.003	0.99 ± 0.004

TABLE IV
RMSE ON THE TEST SET ACCORDING TO r , AVERAGED \pm STANDARD DEVIATION ON 10 RUNS ON ML-100K

r	BSSMF	NMF	MF
1	$0.98 \pm 1 \cdot 10^{-4}$	$0.91 \pm 3 \cdot 10^{-5}$	$0.91 \pm 5 \cdot 10^{-5}$
5	0.89 ± 0.005	0.89 ± 0.01	0.89 ± 0.008
10	0.90 ± 0.008	0.90 ± 0.009	0.92 ± 0.01
20	0.91 ± 0.01	0.93 ± 0.01	0.97 ± 0.02
50	0.93 ± 0.01	0.97 ± 0.01	1.06 ± 0.03
100	0.94 ± 0.01	1.01 ± 0.007	1.13 ± 0.02

using Algorithm 1, where the projections onto the feasible sets are adapted accordingly (projection onto the nonnegative orthant for NMF, no projection for unconstrained MF). The stopping criteria in line 19, 3 and 11 of Algorithm 1 are a maximum number of iterations equal to 200, 1 and 1, respectively, for all algorithms. The experiment is conducted on 10 random initializations and the average RMSEs are reported in Table III. As expected, BSSMF and NMF are more robust to overfitting than unconstrained MF. Additionally, BSSMF is also clearly more robust to overfitting than NMF. Its worse RMSE is 0.89 with $r = 100$ (and it is still equal to 0.89 with $r = 200$), while, for NMF, the RMSE is 0.92 when $r = 100$ (which is worse than a rank-one factorization giving a RMSE of 0.91).

The same experiment is conducted on the ml-100k dataset⁸ which contains 100,000 ratings from 1,700 movies rated by 1,000 users. The test set contains 50 users. The results are reported in Table IV, and the observations are similar: BSSMF is significantly more robust to overfitting than NMF and unconstrained MF.

VI. CONCLUSION

In this article, we proposed a new factorization model, namely bounded simplex structured matrix factorization (BSSMF). Fitting this model retrieves interpretable factors: the learned basis features can be interpreted in the same way as the original data while the activations are nonnegative and sum to one, leading to a straightforward soft clustering interpretation. Instead of learning parts of objects as NMF, BSSMF learns objects that can be used to explain the data through convex combinations. We have proposed a dedicated fast algorithm for BSSMF, and showed that, under mild conditions, BSSMF is essentially unique. We also showed that the constraints in BSSMF make it robust to overfitting on rating datasets without adding any regularization term.

⁷[Online]. Available: <https://grouplens.org/datasets/movielens/1m/>

⁸[Online]. Available: <https://grouplens.org/datasets/movielens/100k/>

Further work include:

- the use of BSSMF for other applications,
- the generalization of BSSMF, e.g., as done in [43] for SSMF where the feasible set for the columns of H , namely the probability simplex, is replaced by any polytope,
- the design of more efficient algorithms for BSSMF, and
- the design of algorithms for other BSSMF models, e.g., with other data fitting terms such as the Kullback-Leibler divergence, as done recently in [34] for SSMF with non-negativity constraint on W .

ACKNOWLEDGMENT

We sincerely thank the Associate Editor and the reviewers for taking the time to carefully read the article, and for the very detailed and insightful feedback that helped us improve our article.

REFERENCES

- [1] M. Abdolali and N. Gillis, "Simplex-structured matrix factorization: Sparsity-based identifiability and provably correct algorithms," *SIAM J. Math. Data Sci.*, vol. 3, no. 2, pp. 593–623, 2021.
- [2] A. M. S. Ang and N. Gillis, "Algorithms and comparisons of nonnegative matrix factorizations with volume regularization for hyperspectral unmixing," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 12, no. 12, pp. 4843–4853, Dec. 2019.
- [3] A. Bakshi, C. Bhattacharyya, R. Kannan, D. P. Woodruff, and S. Zhou, "Learning a latent simplex in input sparsity time," in *Proc. Int. Conf. Learn. Representations*, 2021.
- [4] A. Berman and R. J. Plemmons, *Nonnegative Matrices in the Mathematical Sciences*, vol. 9. Philadelphia, PA, USA: SIAM, 1994.
- [5] J. Bolte, S. Sabach, and M. Teboulle, "Proximal alternating linearized minimization for nonconvex and nonsmooth problems," *Math. Program.*, vol. 146, no. 1, pp. 459–494, 2014.
- [6] E. J. Candès, X. Li, Y. Ma, and J. Wright, "Robust principal component analysis?," *J. ACM*, vol. 58, no. 3, 2011, Art. no. 11.
- [7] V. Chandrasekaran, S. Sanghavi, P. A. Parrilo, and A. S. Willsky, "Rank-sparsity incoherence for matrix decomposition," *SIAM J. Optim.*, vol. 21, no. 2, pp. 572–596, 2011.
- [8] A. Cichocki, R. Zdunek, and S.-I. Amari, "Hierarchical ALS algorithms for nonnegative matrix and 3D tensor factorization," in *Lecture Notes in Computer Science*, vol. 4666. Berlin, Germany: Springer, 2007, pp. 169–176.
- [9] A. Cichocki and R. Zdunek, *Anh Huy Phan, and Shun-Ichi Amari. Non-negative Matrix and Tensor Factorizations: Applications to Exploratory Multi-Way Data Analysis and Blind Source Separation*. Hoboken, NJ, USA: Wiley, 2009.
- [10] S. Cruces, "Bounded component analysis of linear mixtures: A criterion of minimum convex perimeter," *IEEE Trans. Signal Process.*, vol. 58, no. 4, pp. 2141–2154, Apr. 2010.
- [11] A. D'Aspremont, L. E. L. Ghaoui, M. I. Jordan, and G. R. G. Lanckriet, "A direct formulation for sparse PCA using semidefinite programming," *SIAM Rev.*, vol. 49, no. 3, pp. 434–448, 2007.
- [12] Y. Deville, "From separability/identifiability properties of bilinear and linear-quadratic mixture matrix factorization to factorization algorithms," *Digit. Signal Process.*, vol. 87, pp. 21–33, 2019.
- [13] D. Donoho and V. Stodden, "When does non-negative matrix factorization give a correct decomposition into parts?," In in *Proc. Adv. Neural Inf. Process. Syst.*, 2004, pp. 1141–1148.
- [14] T. Alper Erdogan, "A class of bounded component analysis algorithms for the separation of both independent and dependent sources," *IEEE Trans. Signal Process.*, vol. 61, no. 22, pp. 5730–5743, Nov. 2013.
- [15] X. Fu, K. Huang, and N. D. Sidiropoulos, "On identifiability of nonnegative matrix factorization," *IEEE Signal Process. Lett.*, vol. 25, no. 3, pp. 328–332, Mar. 2018.
- [16] X. Fu, K. Huang, N. D. Sidiropoulos, and W.-K. Ma, "Nonnegative matrix factorization for signal and data analytics: Identifiability, algorithms, and applications," *IEEE Signal Process. Mag.*, vol. 36, no. 2, pp. 59–80, Mar. 2019.
- [17] X. Fu, W.-K. Ma, K. Huang, and N. D. Sidiropoulos, "Blind separation of quasi-stationary sources: Exploiting convex geometry in covariance domain," *IEEE Trans. Signal Process.*, vol. 63, no. 9, pp. 2306–2320, May 2015.
- [18] K. R. Gabriel and S. Zamir, "Lower rank approximation of matrices by least squares with any choice of weights," *Technometrics*, vol. 21, no. 4, pp. 489–498, 1979.
- [19] N. Gillis and F. Glineur, "Accelerated multiplicative updates and hierarchical ALS algorithms for nonnegative matrix factorization," *Neural Comput.*, vol. 24, no. 4, pp. 1085–1105, 2012.
- [20] N. Gillis, *Nonnegative Matrix Factorization*. Philadelphia, PA, USA: SIAM, 2020.
- [21] N. Gillis and F. Glineur, "Low-rank matrix approximation with weights or missing data is NP-hard," *SIAM J. Matrix Anal. Appl.*, vol. 32, no. 4, pp. 1149–1165, 2011.
- [22] N. Gillis and A. Kumar, "Exact and heuristic algorithms for semi-nonnegative matrix factorization," *SIAM J. Matrix Anal. Appl.*, vol. 36, no. 4, pp. 1404–1424, 2015.
- [23] N. Guan, D. Tao, Z. Luo, and B. Yuan, "NeNMF: An optimal gradient method for nonnegative matrix factorization," *IEEE Trans. Signal Process.*, vol. 60, no. 6, pp. 2882–2898, Jun. 2012.
- [24] L. T. K. Hien, N. Gillis, and P. Patrinos, "Inertial block proximal methods for non-convex non-smooth optimization," in *Proc. 37th Int. Conf. Mach. Learn.*, 2020, pp. 5671–5681.
- [25] L. T. K. Hien, D. N. Phan, and N. Gillis, "An inertial block majorization minimization framework for nonsmooth nonconvex optimization," *J. Mach. Learn. Res.*, vol. 24, pp. 1–41, 2023.
- [26] K. Huang, N. D. Sidiropoulos, and A. Swami, "Non-negative matrix factorization revisited: Uniqueness and algorithm for symmetric decomposition," *IEEE Trans. Signal Process.*, vol. 62, no. 1, pp. 211–224, Jan. 2014.
- [27] R. Kannan, M. Ishteva, and H. Park, "Bounded matrix factorization for recommender system," *Knowl. Inf. Syst.*, vol. 39, no. 3, pp. 491–511, 2014.
- [28] Y. Koren, R. Bell, and C. Volinsky, "Matrix factorization techniques for recommender systems," *Computer*, vol. 42, no. 8, pp. 30–37, 2009.
- [29] R. Kueng and J. A. Tropp, "Binary component decomposition part I: The positive-semidefinite case," *SIAM J. Math. Data Sci.*, vol. 3, no. 2, pp. 544–572, 2021.
- [30] H. Laurberg, M. G. Christensen, M. D. L. K. P. Hansen, and S. Jensen, "Theorems on positive data: On the uniqueness of NMF," *Comput. Intell. Neurosci.*, 2008, pp. 94–102, 2008.
- [31] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proc. IEEE*, vol. 86, no. 11, pp. 2278–2324, Nov. 1998.
- [32] D. D. Lee and H. S. Seung, "Learning the parts of objects by non-negative matrix factorization," *Nature*, vol. 401, pp. 788–791, 1999.
- [33] D. D. Lee and H. S. Seung, "Algorithms for non-negative matrix factorization," in *Proc. Adv. Neural Inf. Process. Syst.*, 2001, pp. 556–562.
- [34] V. Leplat, N. Gillis, and J. Idier, "Multiplicative updates for NMF with β -divergences under disjoint equality constraints," *SIAM J. Matrix Anal. Appl.*, vol. 42, no. 2, pp. 730–752, 2021.
- [35] D. Liang, G. Rahul, M. D. K. Hoffman, and T. Jebara, "Variational autoencoders for collaborative filtering," in *Proc. World Wide Web Conf.*, 2018, pp. 689–698.
- [36] C.-H. Lin, W.-K. Ma, W.-C. Li, C.-Y. Chi, and A. Ambikapathi, "Identifiability of the simplex volume minimization criterion for blind hyperspectral unmixing: The no-pure-pixel case," *IEEE Trans. Geosci. Remote Sens.*, vol. 53, no. 10, pp. 5530–5546, Oct. 2015.
- [37] K. Liu, X. Li, Z. Zhu, L. Brand, and H. Wang, "Factor-bounded nonnegative matrix factorization," *ACM Trans. Knowl. Discov. Data*, vol. 15 no. 6, pp. 1–18, May 2021.
- [38] A. M. Shun, J. E. Ang, N. C. Gillis, and L. T. K. Hien, "Accelerating block coordinate descent for nonnegative tensor factorization," *Numer. Linear Algebra Appl.*, vol. 28, no. 5, 2021, Art. no. e2373.
- [39] A. Mansour, N. Ohnishi, and C. G. Püntonet, "Blind multiuser separation of instantaneous mixture algorithm based on geometrical concepts," *Signal Process.*, vol. 82, no. 8, pp. 1155–1175, 2002.
- [40] S. Moussaoui, D. Brie, and J. Idier, "Non-negative source separation: Range of admissible solutions and conditions for the uniqueness of the solution," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process.*, 2005, pp. v/289–v/292.

- [41] S. Rendle, W.L. K. Zhang, and Y. Koren, "Revisiting the performance of iALS on item recommendation benchmarks," in *Proc. 16th ACM Conf. Recommender Syst.*, 2022, pp. 427–435.
- [42] N. Srebro and T. Jaakkola, "Weighted low-rank approximations," in *Proc. 20th Int. Conf. Mach. Learn.*, 2003, pp. 720–727.
- [43] G. Tatli and A. T. Erdogan, "Generalized polytopic matrix factorization," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process.* 2021, pp. 3235–3239.
- [44] M. Udell and A. Townsend, "Why are Big Data matrices approximately low rank?," *SIAM J. Math. Data Sci.*, vol. 1, no. 1, pp. 144–160, 2019.
- [45] N. Vaswani, Y. Chi, and T. Bouwmans, "Rethinking PCA for modern data sets: Theory, algorithms, and applications [scanning the issue]," *Proc. IEEE*, vol. 106, no. 8, pp. 1274–1276, Aug. 2018.
- [46] O. Vu Thanh, A. M. S. Ang, N. Gillis, and L. T. K. Hien, "Inertial majorization-minimization algorithm for minimum-volume NMF," in *Proc. 29th Eur. Signal Process. Conf.*, 2021, pp. 1065–1069.
- [47] O. V. Thanh, N. Gillis, and F. Lecron, "Bounded simplex-structured matrix factorization," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process.*, 2022, pp. 9062–9066.
- [48] R. Wu, W.-K. Ma, and X. Fu, "A stochastic maximum-likelihood framework for simplex structured matrix factorization," in *IEEE Int. Conf. Acoust. Speech Signal Process.*, 2017, pp. 2557–2561.
- [49] L. Zhuang, C.-H. Lin, A. T. Mário Figueiredo, and J. M. Bioucas-Dias, "Regularization parameter selection in minimum volume hyperspectral unmixing," *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 12, pp. 9858–9877, Dec. 2019.