SIAM J. Matrix Anal. Appl.
Vol. 34, No. 3, pp. 1189–1212

ROBUSTNESS ANALYSIS OF HOTTOPIXX, A LINEAR PROGRAMMING MODEL FOR FACTORING NONNEGATIVE MATRICES*

NICOLAS GILLIS†

Abstract. Although nonnegative matrix factorization (NMF) is NP-hard in general, it has been shown very recently that it is tractable under the assumption that the input nonnegative data matrix is close to being separable. (Separability requires that all columns of the input matrix belong to the cone spanned by a small subset of these columns.) Since then, several algorithms have been designed to handle this subclass of NMF problems. In particular, Bittorf et al. [*Adv. Neural Inform. Process. Syst.*, 25 (2012), pp. 1223–1231] proposed a linear programming model, referred to as Hottopixx. In this paper, we provide a new and more general robustness analysis of their method. In particular, we design a provably more robust variant using a postprocessing strategy which allows us to deal with duplicates and near duplicates in the data set.

Key words. nonnegative matrix factorization, separability, robustness to noise, linear programming, Hottopixx

AMS subject classifications. 15A23, 90C05, 65F30

DOI. 10.1137/120900629

1. Introduction.

Nonnegative matrix factorization (NMF) is a popular machine learning technique and allows one to express a set of nonnegative vectors as nonnegative linear combinations of nonnegative basis elements [9]. More formally, given a nonnegative matrix $M \in \mathbb{R}^{m\times n}_+$ corresponding to n vectors in an m-dimensional space and a factorization rank r, the aim is to find a basis matrix $U \in \mathbb{R}^{m\times r}_+$ and a weight matrix $V \in \mathbb{R}^{r\times n}_+$ such that the norm of the error $M - UV$ is minimized. Although NMF is NP-hard [10], Arora et al. [1] recently showed that it can be solved in polynomial time given that the matrix M is close to being separable. A nonnegative matrix $M \in \mathbb{R}^{m\times n}_+$ is r-separable if and only if it can be expressed as $M = WH$, where $W \in \mathbb{R}^{m\times r}_+$, $H \in \mathbb{R}^{r\times n}_+$, and each column of W is equal to a column of M. In other terms, $M \in \mathbb{R}^{m\times n}_+$ is r-separable if and only if

$$M \;=\; W\,[I_r, H']\Pi \;=\; [W, WH']\Pi$$

for some $H' \in \mathbb{R}^{r\times n}_+$ and some permutation matrix $\Pi \in \{0,1\}^{n\times n}$. Any nonnegative matrix is n-separable because of the trivial decomposition $M = MI_n$ with $r = n$, and the aim is to find a decomposition where r is as small as possible. It is rather straightforward to check that the smallest such r is the number of extreme rays of the cone generated by the columns of M, that is, $\text{cone}(M) = \{Mx \mid x \in \mathbb{R}^n_+\}$. Equivalently, if the entries of columns of matrix M are normalized to sum to one[1], the smallest such r is the number of vertices of the convex hull of the columns of

[*] Received by the editors November 28, 2012; accepted for publication (in revised form) by I. C. F. Ipsen May 29, 2013; published electronically August 6, 2013. This research presents results of the Belgian Network DYSCO (Dynamical Systems, Control, and Optimization), funded by the Interuniversity Attraction Poles Programme initiated by the Belgian Science Policy Office.
http://www.siam.org/journals/simax/34-3/90062.html

[†] ICTEAM Institute, Université catholique de Louvain, B-1348 Louvain-la-Neuve, Belgium (nicolas.gillis@uclouvain.be). The author is a postdoctoral fellow of the Fonds de la Recherche Scientifique (F.R.S.-FNRS).

[1] By a slight abuse of language, we say that a column sum to one if its entries sum to one.

M, that is, $\operatorname{conv}(M) = \{Mx \mid x \in \mathbb{R}^n_+, \sum_{i=1}^n x_i = 1\}$; see [8] and the references therein for more details about the geometric interpretation of the separable NMF problem.

It turns out that the separability assumption makes sense in several practical situations. For example, in document classification, each column of M corresponds to a document (that is, a vector of word counts) and is approximated with a nonnegative linear combination of the columns of matrix W which correspond to different topics (that is, bags of words). Separability of M requires that for each topic, there exists at least one document discussing only that topic. In practice, this condition is not often satisfied and it is more reasonable to assume separability of M^T (that is, each row of H is equal to a row of M) which requires that *for each topic, there exists at least one word used only by that topic*; see the discussions in [1, 2]. The separability assumption is also widely used in hyperspectral imaging and is referred to as the *pure-pixel assumption*; see [7] and the references therein.

In practice, the input separable matrix M is perturbed with some noise and it is therefore desirable to design robust algorithms; see [1, 2, 3, 4, 5, 7, 8]. In fact, in the noiseless case, the problem is rather easy and reduces to identifying the vertices of the convex hull of a set of points. In this paper, we will focus on the algorithm of Bittorf et al. [3], referred to as Hottopixx, which is described in the next section. As we will see, the robustness analysis provided by the authors is rather restrictive as it does not deal with duplicates nor near duplicates of the columns of W in the data set: the aim of this paper is to develop a more general analysis of their algorithm and design a provably more robust variant applicable to any noisy separable matrix (that is, allowing duplicates and near duplicates in the data).

1.1. Hottopixx: A linear programming model for separable NMF.

From now on, we will *always* assume that the columns of the input data matrix M have been normalized in order to sum to one, that is,

(i) the zero columns of M have been discarded and

(ii) each column of M is updated using $M(:,j) \leftarrow \frac{M(:,j)}{||M(:,j)||_1}$.

We will also always assume that we are given a noisy separable matrix $\tilde{M} = M + N$, where $N \in \mathbb{R}^{m\times n}$ is some noise added to the separable matrix M such that

$$||N||_1 = \max_{||x||_1 \leq 1} ||Nx||_1 = \max_j ||N(:,j)||_1 \leq \epsilon \quad \text{ for some } \epsilon \geq 0.$$

The matrix M is r-separable if and only if

$$\begin{aligned} (1.1) \qquad M &= WH = W[I_r, H']\Pi = [W, WH']\Pi \\ &= [W, WH']\Pi \underbrace{\Pi^{-1} \begin{pmatrix} I_r & H' \\ 0_{(n-r)\times r} & 0_{(n-r)\times(n-r)} \end{pmatrix}}_{X^0 \in \mathbb{R}^{n\times n}_+} \Pi = MX^0 \end{aligned}$$

for some $W \geq 0$, $H' \geq 0$ and some permutation matrix Π. Equation (1.1) shows that M is r-separable if and only if there exists a nonnegative matrix $X^0 \in \mathbb{R}^{n\times n}_+$ such that (1) X^0 contains $(n-r)$ all-zero rows and the r-by-r identity matrix as a submatrix (up to permutation) and (2) $M = MX^0$. Notice that because the columns of matrix M and W sum to one, the columns of the matrix H' have sum to one as well. Based on these observations, Bittorf et al. [3] proposed to solve the following optimization problem[2] in order to identify approximately the columns of the matrix W among the columns of the matrix $\tilde{M}$:

$$\min_{X \in \mathbb{R}^{n \times n}_{+}} \quad p^T \operatorname{diag}(X) \text{ such that}$$

$$||\tilde{M} - \tilde{M}X||_1 \leq 2\epsilon, \tag{1.2a}$$

$$\operatorname{tr}(X) = r, \tag{1.2b}$$

$$X(i,i) \leq 1 \text{ for all } i, \tag{1.2c}$$

$$X(i,j) \leq X(i,i) \text{ for all } i,j, \tag{1.2d}$$

where $p \in \mathbb{R}^n$ is any vector with distinct entries. Intuitively, the model reads as follows: we have to assign a weight in [0,1] (1.2c) to each column of M (that is, give a value to $X(i,i)$ for all i) for a total weight of r (1.2b). Moreover, we cannot use a column to reconstruct another column with a weight larger than the corresponding diagonal entry of X (1.2d), while we have to guarantee that the approximation error is small (1.2a). It is interesting to notice that the problem is always feasible: in fact, X^0 from (1.1) is a feasible solution of (1.2) since the columns of H' sum to one and

$$\begin{aligned} ||\tilde{M} - \tilde{M}X^0||_1 &= ||(M+N) - (M+N)X^0||_1 \\ &\leq ||M - MX^0||_1 + ||N||_1 + ||N||_1 ||X^0||_1 \leq 2\epsilon. \end{aligned}$$

Finally, Bittorf et al. [3] identify approximately the columns of W by selecting the r columns of $\tilde{M}$ whose corresponding diagonal entries of an optimal solution of (1.2) are the largest; see Algorithm 1, referred to as *Hottopixx*. Note that the corresponding optimal weight matrix H can be obtained by solving another linear program; see Algorithm 2.

Algorithm 1. Hottopixx. Extracting columns of a separable matrix by linear programming [3].

```
Input: A noisy $r$-separable matrix $\tilde{M} = WH + N$, the noise level $||N||_1 \leq \epsilon$, and the
    number $r$ of columns of $W$.
Output: A matrix $\tilde{W}$ such that $||\tilde{W}(:,P) - W||_1$ is small for some permutation $P$.
Find the optimal solution $X^*$ of (1.2).
Let $\mathcal{K}$ be the index set corresponding to the $r$ largest diagonal entries of $X^*$.
Set $\tilde{W} = \tilde{M}(:,\mathcal{K})$.
```

Algorithm 2. Near-separable NMF using Hottopixx and linear programming [3].

```
Input: A noisy $r$-separable matrix $\tilde{M} = WH + N$, the noise level $||N||_1 \leq \epsilon$, and the
    number $r$ of columns of $W$.
Output: An nonnegative factorization $(\tilde{W}, \tilde{H})$ such that $||\tilde{M} - \tilde{W}\tilde{H}||_1$ is small.
Compute $\tilde{W}$ using Algorithm 1.
Solve $\tilde{H} = \operatorname{argmin}_{Y \geq 0} ||\tilde{M} - \tilde{W}Y||_1$.
```

[2] In [3], the model assumes separability of M^T so that (1.2) is equivalent to the model in [3] applied to M^T. We prefer here to work with the columns.

Before stating robustness results, it is important to define the conditioning of matrix W, which is a crucial characteristic of separable NMF problems. In fact, the better the columns of W are spread in the unit simplex $\Delta^m = \{x \in \mathbb{R}^m \mid x \geq 0, \sum_{i=1}^m = 1\}$, the more noise tolerant the data will be. In [1, 3], this conditioning is measured via the parameter

$$\alpha = \min_{1 \leq k \leq r, x \in \Delta^{r-1}} ||W(:,k) - W(:,\mathcal{R})x||_1, \qquad \text{where } \mathcal{R} = \{1,2,\dots,r\} \backslash \{k\},$$

and the matrix W is said to be *α-robustly simplicial*. (Notice that $\alpha \leq 2$ for any nonnegative matrix W whose columns sum to one.) In other words, α is the minimum among the ℓ_1-distances between a column of W and the convex hull of the other columns of W. It is necessary that $||N||_1 \leq \epsilon < \frac{\alpha}{2}$ for *any* separable NMF algorithm to be able to approximately recover the columns of W from the matrix $\tilde{M} = WH + N$. In fact, if $\epsilon \geq \frac{\alpha}{2}$, any r-separable matrix M with $r \geq 2$ can be perturbed so that one of the columns of the perturbed matrix $\tilde{M}$ corresponding to a column of W belongs to the convex hull of the other columns. In other words, we can perturb the matrix M so that it becomes $(r-1)$-separable and we could therefore not distinguish one of the columns of W from the columns of M. For example, with

$$W = \begin{pmatrix} \frac{\alpha}{2} I_r \\ (1 - \frac{\alpha}{2}) e^T \end{pmatrix}, \ H = I_r, \text{ and } N = \begin{pmatrix} \frac{-\alpha}{2} I_r \\ 0 \end{pmatrix}, \text{ we have } \tilde{M} = \begin{pmatrix} 0_{r \times r} \\ (1 - \frac{\alpha}{2}) e^T \end{pmatrix},$$

so that the matrix $M = WH$ is r-separable with W α-robustly simplicial and $||N||_1 = \frac{\alpha}{2}$, while $\tilde{M}$ is 1-separable.

In order to prove robustness results for Algorithm 2, Bittorf et al. [3] used the following observation.

LEMMA 1.1. *Suppose M is normalized and admits a rank-r separable factorization WH, and suppose $\tilde{M} = M + N$ with $||N||_1 \leq \epsilon$. If $\tilde{W}$ is such that $||\tilde{W}(:,P) - W||_1 \leq \delta$ for some $\delta \geq 0$ and some permutation P; then Algorithm 2 constructs a factorization $(\tilde{W}, \tilde{H})$ satisfying $||\tilde{M} - \tilde{W}\tilde{H}||_1 \leq \epsilon + \delta$.*

Proof. Denoting $N_W = \tilde{W}(:,P) - W$, we have

$$\begin{aligned} ||\tilde{M} - \tilde{W}\tilde{H}||_1 &= \text{argmin}_{Y \geq 0} ||M + N - (W + N_W)Y||_1 \\ &\leq ||M + N - (W + N_W)H||_1 \\ &\leq ||N||_1 + ||N_W H||_1 + ||M - WH||_1 \\ &\leq \epsilon + \delta \end{aligned}$$

since the columns of H sum to one. ◻

Lemma 1.1 allows us to focus on proving robustness results for Algorithm 1. In fact, any result that applies to Algorithm 1 directly applies to Algorithm 2. In this paper, we will therefore focus our attention on Algorithm 1, as was implicitly done in [3].

We can now state the robustness result for Hottopixx proposed in [3].

THEOREM 1.2 ([3, Thm. 3.2]). *Suppose $\tilde{M} = M + N$, where M is normalized and admits a rank-r separable factorization WH with W α-robustly simplicial and $||N||_1 \leq \epsilon$. Suppose that the there is no duplicate of the columns of W and that for all columns with index j such that $M(:,j) \neq W(:,k)$ for all k, we have a margin constraint $||M(:,j) - W(:,k)||_1 \geq d_0$ for all k. Suppose also that $\epsilon < \frac{\min(\alpha d_0, \alpha^2)}{9(r+1)}$.*

Then Algorithm 1 *identifies correctly the columns of* W*, that is, it extracts a matrix* $\tilde{W}$ *satisfying* $||\tilde{W}(:,P) - W||_1 \leq \epsilon$ *for some permutation* P*.*

Note that the above robustness result only deals with input data matrices *without duplicates or near duplicates* of the columns of W (because of the margin constraint). In other terms, the columns of W must be isolated for the robustness result to apply. This is a very unnatural condition. For example, in document data sets, separability requires that for each topic there exists at least one word used only by that topic; see the introduction. In this context, the additional margin condition requires that for each topic there exists *one and only one* word associated with that topic, which is rather impractical. In this paper, we propose a postprocessing strategy for Hottopixx so that duplicates and near duplicates are allowed in the data set.

1.2. Conditioning and κ-robustly conical matrices. Because the columns of the variable X in (1.2) are not required to sum to one, it turns out that it will be easier to work with the following parameter measuring the conditioning of matrix W:

$$\kappa = \min_{1\leq k\leq r} \min_{x\in\mathbb{R}^{r-1}_+} ||W(:,k) - W(:,\mathcal{R})x||_1, \qquad \text{where } \mathcal{R} = \{1,2,\dots,r\}\backslash\{k\},$$

and the matrix W is said to be *κ-robustly conical.* We have that κ is the minimum among the ℓ_1-distances between a column of W and the convex *cone* generated by the other columns of W. If the columns of W sum to one (which will always be assumed), $\kappa \leq 1$ and we can relate α and κ as follows.

THEOREM 1.3. *For any α-robustly simplicial and κ-robustly conical nonnegative matrix W whose columns sum to one, we have*

$$\kappa \leq \alpha \leq 2\kappa.$$

Proof. The first inequality follows directly from the definition. The second is proved in Appendix A. □

Therefore, it is essentially equivalent to working with α or κ as they differ by a multiplicative factor of at most 2.

1.3. Contribution and outline of the paper. In this paper, we provide a new analysis of Hottopixx (Algorithm 1). This in turn allows us to design a postprocessing strategy leading to a provably more robust variant (Algorithm 3) which is applicable to any separable matrix (that is, duplicates and near duplicates are allowed in the data set as opposed to the original robustness result from [3]).

In the first part of the paper (section 2), we analyze the case where no duplicates or near duplicates are allowed in the data set and focus on the following proposition.

PROPOSITION 1. *Suppose $\tilde{M} = M + N$, where M is normalized, admits a rank-r separable factorization WH, where W is κ-robustly simplicial with $\kappa > 0$, and has the form* (1.1) *with* $\max_{i,j} H'_{ij} \leq \beta \leq 1$*. Suppose also that* $||N||_1 \leq \epsilon$ *and* ϵ *is sufficiently small. Then Algorithm* 1 *extracts a matrix* $\tilde{W}$ *satisfying* $||W - \tilde{W}(:,P)||_1 \leq \epsilon$ *for some permutation* P*.*

Note that the condition on the entries of H' is implied by the margin constraint of Theorem 1.2 since

$$||M(:,j) - W(:,k)||_1 \geq d_0 \text{ for all } 1 \leq k \leq r \quad \Rightarrow \quad \max_i H(i,j) \leq \beta = 1 - \frac{d_0}{2};$$

see Lemma 3.2. Hence, by Theorem 1.2, Proposition 1 holds for $\epsilon < \frac{\min(2\alpha(1-\beta),\alpha^2)}{9(r+1)}$. In section 2, we prove that

- $\epsilon \leq \frac{\kappa(1-\beta)}{9(r+1)}$ is sufficient for Proposition 1 to hold (Theorem 2.3), while
- $\epsilon < \frac{\kappa(1-\beta)}{(1-\beta)(r-1)+1}$ is necessary for Proposition 1 to hold for any $r \geq 3$ and $\beta < 1$ (Theorem 2.4).

Hence, our analysis gets rid of the term α^2 from Theorem 1.2 and is close to being tight.

In the second part of the paper (section 3), we do not make any assumption on the input separable matrix and focus on the following proposition.

PROPOSITION 2. *Suppose $\tilde{M} = M + N$ where M is normalized and admits a rank-r separable factorization WH, where W is κ-robustly simplicial with $\kappa > 0$. Suppose also that $||N||_1 \leq \epsilon$ and ϵ is sufficiently small. Then Algorithm 1 extracts a matrix $\tilde{W}$ satisfying $||W - \tilde{W}(:,P)||_1 \leq \delta$ for some permutation P and some $\delta \geq 0$.*

We first show that it is necessary for Proposition 2 to hold that $\epsilon < \frac{\kappa}{r-1}$ (Corollary 2.5) and that $\delta \geq 3\frac{\epsilon}{\alpha} + \frac{3}{2}\epsilon$ for any $\epsilon < \frac{\alpha}{2}$ (Theorem 3.1). (We also show that this lower bound on δ applies to a broader class of separable NMF algorithms.) Then, we propose a postprocessing of the solution of the linear program (1.2) (see Algorithm 3) for which the following result holds.

Theorem 3.5 (see section 3.2). Let $M = WH$ be a normalized r-separable matrix where W is κ-robustly conical with $\kappa > 0$. Let also $\tilde{M} = M + N$ with $||N||_1 \leq \epsilon$. If

$$\epsilon < \frac{\omega\kappa}{99(r+1)},$$

where $\omega = \min_{i \neq j} ||W(:,i) - W(:,j)||_1$ (note that $\omega \geq \kappa$), then Algorithm 3 extracts a matrix $\tilde{W}$ such that

$$||W - \tilde{W}(:,P)||_1 \leq 49(r+1)\frac{\epsilon}{\kappa} + 2\epsilon \quad \text{for some permutation } P.$$

Because the necessary condition $\epsilon < \frac{\kappa}{r-1}$ also applies to Algorithm 3, the bound for ϵ of Theorem 3.5 is tight up to a factor ω (and some constant multiplicative factor). Moreover, because of the necessary condition on δ (see above), Theorem 3.5 is tight up to a factor r (and some constant multiplicative factor). Finally, we show that it is necessary for Proposition 2 to hold that $\epsilon \leq \frac{\kappa}{(r-1)^2}$ for any $\delta < \kappa + \epsilon$ (Theorem 3.6), which demonstrates that Hottopixx cannot achieve a better bound than Algorithm 3. We also compare Algorithm 3 with the algorithm of Arora et al. [1] in section 3.4.

In the last part of the paper (section 4), we illustrate our results on some synthetic data sets: we show that the postprocessing makes Hottopixx more robust to noise and able to deal with duplicates and near duplicates of the columns of W.

1.4. Notation.

The set of m-by-n real matrices is denoted $\mathbb{R}^{m \times n}$; for $A \in \mathbb{R}^{m \times n}$, we denote the jth column of A by $A(:,j)$, the ith row of A by $A(i,:)$, and the entry at position (i,j) by $A(i,j)$; for $b \in \mathbb{R}^{m \times 1} = \mathbb{R}^m$, we denote the ith entry of b by $b(i)$. Notation $A(\mathcal{I}, \mathcal{J})$ refers to the submatrix of A with row and column indices respectively in $\mathcal{I}$ and $\mathcal{J}$. The matrix A^T is the transpose of A. The ℓ_1-norm $||.||_1$ of a vector is defined as $||b||_1 = \sum_i |b(i)|$ and of a matrix as $||A||_1 = \max_j ||A(:,j)||_1$. The ℓ_∞-norm $||.||_\infty$ of a vector is defined as $||b||_\infty = \max_i |b(i)|$. We will denote by E_n the n-by-n matrix of all-ones, $0_{m \times n}$ the m-by-n the matrix of all-zeros, and I_n the n-by-n identity matrix. We will also denote e_i the ith column of the identity matrix, e the all-one vector, and 0 the all-zero vector; their dimensions will be clear from the context. The vector of the diagonal entries of a matrix A is denoted $\text{diag}(A)$, while its trace is denoted $\text{tr}(A) = e^T \text{diag}(A)$. For a set $\mathcal{K}$, $|\mathcal{K}|$ denotes its cardinality.

2. Analysis without duplicates or near duplicates.

In this section, we focus on Proposition 1: we show that $\epsilon \leq \frac{\kappa(1-\beta)}{9(r+1)}$ is sufficient for Proposition 1 to hold (Theorem 2.3), while $\epsilon < \frac{\kappa(1-\beta)}{(r+1)(1-\beta)+1}$ is necessary for any $r \geq 3$ and $\beta < 1$ (Theorem 2.4).

LEMMA 2.1. *Suppose $\tilde{M} = M + N$, where M is normalized and $||N||_1 \leq \epsilon < 1$, and suppose X is a feasible solution of* (1.2). *Then, for all $1 \leq j \leq n$,*

$$||X(:,j)||_1 \leq 1 + \frac{4\epsilon}{1-\epsilon} \quad \text{and} \quad ||M(:,j) - MX(:,j)||_1 \leq \frac{4\epsilon}{1-\epsilon}.$$

Proof. For all $1 \leq j \leq n$,

$$1 - \epsilon \leq ||M(:,j)||_1 - ||N(:,j)||_1 \leq ||M(:,j) + N(:,j)||_1 = ||\tilde{M}(:,j)||_1,$$

from which we obtain

$$\begin{aligned} 2\epsilon \geq ||\tilde{M}(:,j) - \tilde{M}X(:,j)||_1 &\geq ||\tilde{M}X(:,j)||_1 - ||\tilde{M}(:,j)||_1 \\ &\geq ||MX(:,j)||_1 - ||NX(:,j)||_1 - (1+\epsilon) \\ &\geq ||X(:,j)||_1 - \epsilon||X(:,j)||_1 - 1 - \epsilon, \end{aligned}$$

since the columns of M sum to one and M and X are nonnegative. This implies that $||X(:,j)||_1 \leq \frac{1+3\epsilon}{1-\epsilon} = 1 + \frac{4\epsilon}{1-\epsilon}$, and $||NX(:,j)||_1 \leq ||N||_1||X(:,j)||_1 \leq \epsilon\left(\frac{1+3\epsilon}{1-\epsilon}\right)$. We then have

$$\begin{aligned} 2\epsilon \geq ||\tilde{M}(:,j) - \tilde{M}X(:,j)||_1 &= ||M(:,j) + N(:,j) - (M+N)X(:,j)||_1 \\ &\geq ||M(:,j) - MX(:,j)||_1 - \epsilon - \epsilon\left(\frac{1+3\epsilon}{1-\epsilon}\right), \end{aligned}$$

hence $||M(:,j) - MX(:,j)||_1 \leq 3\epsilon + \epsilon\left(\frac{1+3\epsilon}{1-\epsilon}\right) = \frac{4\epsilon}{1-\epsilon}$. □

Remark 1. Note that a sum-to-one constraint on the columns of X could be added to the model (1.2) while keeping linearity and would make the analysis simpler. In fact, we would directly have $||X(:,j)||_1 = 1$ and $||M(:,j) - MX(:,j)||_1 \leq 4\epsilon$ for all j, and the error bounds from Theorems 2.3 and 3.5 could be slightly improved. However, in this paper we stick with the original formulation proposed in [3].

LEMMA 2.2. *Let $\tilde{M} = M + N$, where M is normalized, admits a rank-r separable factorization WH, where W is κ-robustly conical with $\kappa > 0$, and $||N||_1 \leq \epsilon < 1$, and has the form* (1.1) *with $\max_{i,j} H'_{ij} \leq \beta < 1$. Also let X be any feasible solution of* (1.2). *Then*

$$X(j,j) \geq 1 - \frac{8\epsilon}{\kappa(1-\beta)(1-\epsilon)}$$

for all j such that $M(:,j) = W(:,k)$ for some $1 \leq k \leq r$.

Proof. Let $\mathcal{K}$ be the set of r indices such that $M(:,\mathcal{K}) = W$. Let also $1 \leq k \leq r$ and denote $j = \mathcal{K}(k)$ so that $M(:,j) = W(:,k)$. By Lemma 2.1,

$$||W(:,k) - WHX(:,j)||_1 \leq \frac{4\epsilon}{1-\epsilon}. \tag{2.1}$$

Since $H(k,j) = 1$,

$$\begin{aligned} WHX(:,j) &= W(:,k)H(k,:)X(:,j) + W(:,\mathcal{R})H(\mathcal{R},:)X(:,j) \\ &= W(:,k)\Big(X(j,j) + H(k,\mathcal{J})X(\mathcal{J},j)\Big) + W(:,\mathcal{R})y, \end{aligned}$$

where $\mathcal{R} = \{1, 2, \ldots, r\} \backslash \{k\}$, $\mathcal{J} = \{1, 2, \ldots, n\} \backslash \{j\}$, and $y = H(\mathcal{R}, :)X(:, j) \geq 0$. We have

$$\eta = X(j,j) + H(k,\mathcal{J})X(\mathcal{J},j) \leq X(j,j) + \beta \left(1 + \frac{4\epsilon}{1-\epsilon} - X(j,j)\right), \tag{2.2}$$

since $||H(k,\mathcal{J})||_\infty \leq \beta$ and $||X(:,j)||_1 \leq 1 + \frac{4\epsilon}{1-\epsilon}$ (Lemma 2.1). Hence

$$||W(:,k) - WHX(:,j)||_1 \geq (1-\eta) \left\| W(:,k) - W(:,\mathcal{R})\frac{y}{1-\eta} \right\|_1 \geq (1-\eta)\kappa. \tag{2.3}$$

Combining (2.1), (2.2), and (2.3), we obtain

$$1 - \left(X(j,j) + \beta \left(1 + \frac{4\epsilon}{1-\epsilon} - X(j,j)\right)\right) \leq \frac{4\epsilon}{\kappa(1-\epsilon)},$$

which gives, using the fact that $\kappa, \beta \leq 1$,

$$X(j,j) \geq 1 - \frac{8\epsilon}{\kappa(1-\beta)(1-\epsilon)}. \qquad \square$$

THEOREM 2.3. *It is sufficient for Proposition* 1 *to hold that*

$$\epsilon \leq \frac{\kappa(1-\beta)}{9(r+1)}.$$

Proof. If $\epsilon = 0$, the proof is given in [3, Thm. 3.1]: for each $1 \leq k \leq r$, there exists a unique $j \in \{1, 2, \ldots, n\}$ such that $M(:,j) = W(:,k)$ and $X(j,j) = 1$. (This follows easily from the fact that the entries of p are distinct.) (Note that in the noiseless case when $\epsilon = 0$, duplicates and near duplicates are allowed in the data set since β can be equal to one.) Otherwise $\epsilon > 0$ and $\beta < 1$. Let X be a feasible solution of (1.2) (which always exists since the feasible set of (1.2) is nonempty). If we prove that the r diagonal entries of X corresponding to the columns of W are larger than all the other ones (because $\beta < 1$, there are no duplicates of the columns of W in the data set), then we are done. In fact, these columns will then be identified by Algorithm 2 and we will have $||W - \tilde{W}(:,P)||_1 \leq \epsilon$ for some permutation P. (Notice that we do not need an optimal solution: any feasible solution identifies the columns of W.)

Let $\mathcal{K}$ be the set of r indices such that $M(:,\mathcal{K}) = W$. Assume that

$$X(k,k) > \frac{r}{r+1} \qquad \text{for all } k \in \mathcal{K}. \tag{2.4}$$

Since $\text{tr}(X) = r$ and $X \geq 0$, we have

$$\sum_{j \notin \mathcal{K}} X(j,j) = r - \sum_{k \in \mathcal{K}} X(k,k) < r - r\frac{r}{r+1} = \frac{r}{r+1} < X(k,k) \quad \text{ for all } k \in \mathcal{K},$$

implying that $X(j,j) < X(k,k)$ for all $k \in \mathcal{K}, j \notin \mathcal{K}$, which gives the result. It remains to show that (2.4) holds. By Lemma 2.2,

$$X(k,k) \geq 1 - \frac{8\epsilon}{\kappa(1-\beta)(1-\epsilon)} > 1 - \frac{9\epsilon}{\kappa(1-\beta)},$$

since $\frac{8}{1-\epsilon} < 9$ for any $\epsilon \leq \frac{\kappa(1-\beta)}{9(r+1)} \leq \frac{1}{18}$ as $\kappa > 0$, $\beta < 1$ and $r \geq 1$. Finally, for $\epsilon \leq \frac{(1-\beta)\kappa}{9(r+1)}$, $X(i,i) > \frac{r}{r+1}$ and the proof is complete. □

Remark 2. The proof of Theorem 2.3 actually does not make use of the constraints $X(i,j) \leq X(i,i)$ for all i,j. The reason is that the assumption $\max_{i,j} H'_{ij} \leq \beta < 1$ implies that there is no duplicate of the columns of W in the data set (if $\beta = 1$, $\epsilon = 0$ and Algorithm 2 is guaranteed to work [3, Thm. 3.1]). This implies that for being able to reconstruct sufficiently well each column of W, the corresponding diagonal entry of X must be large independently of the other entries of the corresponding column of X.

Therefore, in case there is no duplicate in the data set (or some some preprocessing has been used to remove them), these constraints can be discarded. (A similar observation was made in [3].) Moreover, since Theorem 2.3 only requires feasibility in that case, any feasible solution of the corresponding relaxed linear program will correctly identify the columns of W.

THEOREM 2.4. *For Proposition* 1 *to hold when* $r \geq 3$ *and* $\beta < 1$, *it is necessary that*

$$\epsilon < \frac{\kappa(1-\beta)}{(r-1)(1-\beta)+1}. \tag{2.5}$$

Proof. See Appendix B. □

Theorem 2.4 shows that the sufficient condition derived in Theorem 2.3 is close to being tight. In particular, if r is assumed to be bounded above, then it is tight up to some constant multiplicative factor. (In practice r is often assumed to be small.) We believe it is possible to improve the bound of Theorem 2.3 to match the one of Theorem 2.4 (up to some constant multiplicative factor). Unfortunately, we were not able to derive such a sufficient condition; this is a topic for further research.

Remark 3 (cases $r = 1, 2$). Theorem 2.4 does not apply when $r = 1, 2$ because of the following:

- The rank-one separable NMF problem is trivial. In fact, if M admits a rank-one separable factorization wh^T and $\tilde{M} = M + N$ with $||N||_1 \leq \epsilon$, then $||\tilde{M}(:,j) - w||_1 \leq \epsilon$ for all j.
- The rank-two case is particular because it is not possible to construct very bad instances. In fact, all rank-two separable NMF problems are essentially equivalent to each other because the columns of M belong to the line segment $[W(:,1), W(:,2)]$.

To conclude this section, we provide a necessary condition for Proposition 2.

COROLLARY 2.5. *For Proposition* 2 *to hold for any* $\delta < \frac{\kappa}{2}$ *and* $r \geq 3$, *it is necessary that*

$$\epsilon < \frac{\kappa}{r-1}.$$

Proof. In fact,

$$\epsilon < \frac{\kappa(1-\beta)}{(1-\beta)(r-1)+1} \leq \frac{\kappa(1-\beta)}{(1-\beta)(r-1)} = \frac{\kappa}{r-1},$$

while the matrix $\tilde{M} = WH + N$ constructed in the proof of Theorem 2.4 satisfies $||W - \tilde{W}(:,P)||_1 \geq \frac{r-2}{r-1}\kappa \geq \frac{\kappa}{2}$, where $\tilde{W}$ is the matrix extracted by Algorithm 1 and P is any permutation. □

3. Dealing with duplicates and near duplicates using postprocessing.

In this section, we investigate Proposition 2 and propose a variant of Hottopixx (see Algorithm 3) which is provably robust for *any noisy separable matrix.* In section 3.1, we present a simple necessary condition for Proposition 2 to hold. In section 3.2, we show that for each column of W, there is a subset of the columns of $\tilde{M}$ close to that column of W such that the sum of the corresponding diagonal entries of any feasible solution X of (1.2) is larger than $\frac{r}{r+1}$. Therefore, using an appropriate postprocessing of the solution X of (1.2) (see Algorithm 3), we can approximately recover the columns of W, given that the noise level ϵ is smaller than some upper bound. In section 3.3, we show that Hottopixx (Algorithm 1) cannot achieve this bound which proves that Algorithm 3 is more robust. Finally, we compare Algorithm 1 with the algorithm of Arora et al. [1] in section 3.4.

3.1. Preliminary necessary conditions. Recall that the aim is to identify, among the columns of $\tilde{M}$, r columns gathered in the matrix $\tilde{W}$ in such a way that $||W - \tilde{W}(:,P)||_1 \leq \delta$ for some permutation P and some $\delta \geq 0$. Since $||W||_1 = 1$, we will assume that $\delta < 1$; otherwise the separable NMF problem is trivial since the solution $\tilde{W} = 0$ gives the result. It actually makes sense to impose $\delta < \kappa \leq \alpha \leq 1$: this guarantees a solution $\tilde{W}$ to have distinct columns since two columns of W can potentially be at distance κ; for example, with

$$W = \begin{pmatrix} \frac{\kappa}{2} & 0 \\ 0 & \frac{\kappa}{2} \\ 1-\frac{\kappa}{2} & 1-\frac{\kappa}{2} \end{pmatrix},$$

extracting twice the first column would give the result with $\delta = \kappa$, which is not desirable. Moreover, as shown in section 1.1, it is necessary that $\epsilon < \frac{\alpha}{2} \leq \kappa$ for *any* separable NMF algorithm to be able to extract approximately the columns of W.

THEOREM 3.1. *For any $0 \leq \epsilon < \frac{\alpha}{2}$, it is necessary that $\delta \geq \left(3\frac{\epsilon}{\alpha} + \frac{3}{2}\epsilon\right)$ for Proposition 2 to hold.*

Proof. Let us consider $\tilde{M} = M + N = WH + N$, where

$$W = \begin{pmatrix} 1 & 0 & \frac{1}{2}-\frac{\alpha}{4} \\ 0 & 1 & \frac{1}{2}-\frac{\alpha}{4} \\ 0 & 0 & \frac{\alpha}{2} \end{pmatrix}, \quad H = \begin{pmatrix} 1 & 0 & 0 & 1-\lambda & 0 \\ 0 & 1 & 0 & 0 & 1-\lambda \\ 0 & 0 & 1 & \lambda & \lambda \end{pmatrix},$$

$$\text{and } N = \begin{pmatrix} 0 & 0 & \frac{\epsilon}{4} & 0 & 0 \\ 0 & 0 & \frac{\epsilon}{4} & 0 & 0 \\ 0 & 0 & -\frac{\epsilon}{2} & 0 & 0 \end{pmatrix},$$

where W is α-robustly simplicial (and $\frac{\alpha}{2}$-robustly conical) and where λ is such that the middle point between $M(:,4)$ and $M(:,5)$ is $\left(\tilde{M}(:,3) + 2N(:,3)\right)$, that is,

$$\tilde{M}(:,3) + 2N(:,3) = \begin{pmatrix} \frac{1}{2}-\frac{\alpha}{4}+\frac{3\epsilon}{4} \\ \frac{1}{2}-\frac{\alpha}{4}+\frac{3\epsilon}{4} \\ \frac{\alpha}{2}-\frac{3\epsilon}{2} \end{pmatrix} = \begin{pmatrix} \frac{1}{2}-\frac{\lambda\alpha}{4} \\ \frac{1}{2}-\frac{\lambda\alpha}{4} \\ \frac{\lambda\alpha}{2} \end{pmatrix} = \frac{1}{2}(M(:,4) + M(:,5)),$$

which requires $\lambda = 1 - 3\frac{\epsilon}{\alpha} \geq 0$. Let $p = (-K, -K, K^2, -1, 0)^T$ for any K sufficiently large. It can be checked that

$$X = \begin{pmatrix} 1 & 0 & 0 & \mu & 0 \\ 0 & 1 & 0 & 0 & \mu \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.5 & 0.5 & 0.5-\mu \\ 0 & 0 & 0.5 & 0.5-\mu & 0.5 \end{pmatrix}, \qquad \text{where } \mu = \frac{1-\lambda}{2-\lambda},$$

is a feasible solution of (1.2). By Lemma B.1, there exists K sufficiently large such that $X^*(3,3) = 0$ for any optimal solution X^*. Using Lemma B.1 again we have $X^*(1,1) = X^*(2,2) = 1$ for any optimal solution X^* for K sufficiently large. Hence, for K sufficiently large, the third column of M will not be extracted and the fourth or fifth will be, hence,

$$\begin{aligned} ||\tilde{W} - W||_1 = ||\tilde{W}(:,3) - W(:,3)||_1 &= ||M(:,4) - W(:,3)||_1 = ||M(:,5) - W(:,3)||_1 \\ &= ||(1-\lambda)W(:,1) - (1-\lambda)W(:,3)||_1 \\ &= 3\frac{\epsilon}{\alpha}\left\|W(:,1) - W(:,3)\right\|_1 \\ &= 3\frac{\epsilon}{\alpha}\left(1 + \frac{\alpha}{2}\right) = 3\frac{\epsilon}{\alpha} + \frac{3}{2}\epsilon. \qquad \square \end{aligned}$$

Using the same construction[3] as in Theorem 3.1 but taking $\lambda = 1 - \frac{\epsilon}{\alpha}$, we have

$$\tilde{M}(:,3) = W(:,3) + N(:,3) = \frac{1}{2}\left(M(:,4) + M(:,5)\right),$$

for which $||\tilde{W}(:,P) - W||_1 \geq \frac{\epsilon}{\alpha} + \frac{\epsilon}{2}$ for any permutation P, where $\tilde{W}$ is the matrix extracted by Hottopixx. We notice that the corresponding matrix $\tilde{M}$ can also be obtained from a 4-separable matrix $M_4 = W_4 H_4$, where

$$W_4 = \big(M(:,[1\,2]\big) \quad M(:,4) - v \quad M(:,5) - v), H_4 = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0.5 & 1 & 0 \\ 0 & 0 & 0.5 & 0 & 1 \end{pmatrix},$$

$v = (\epsilon/4, \epsilon/4, -\epsilon/2)^T$, and

$$N_4 = \begin{pmatrix} 0_{3\times 2} & v & v & v \end{pmatrix},$$

and we have $\tilde{M} = WH + N = W_4 H_4 + N_4 = \tilde{M}_4$. Therefore, *no algorithm to which only the noisy separable matrix $\tilde{M}$ and the noise level ϵ are given as input can approximately extract the columns of W among the columns of M with error smaller than $\mathcal{O}\left(\frac{\epsilon}{\alpha}\right)$.* In fact, the matrix $\tilde{M}$ above has two solutions to the noisy separable NMF problem and there is no way to discriminate between them (the original matrix could be 3- or 4- separable):

- If the algorithm returns a matrix $\tilde{W}$ with three columns, then if the original matrix was M_4 we have $\max_{1\leq j\leq 4}\min_{1\leq k\leq 3}||W_4(:,j) - \tilde{W}(:,k)||_1 \geq \frac{\epsilon}{\alpha}$.

[3] A MATLAB code is available at https://sites.google.com/site/nicolasgillis/code and contains this construction, along with the one of Theorem 3.1.

- Similarly, if the algorithm returns a matrix $\tilde{W}$ with four columns, then
  - if the third column is not extracted and the original matrix was M, we have

$$\max_{1\leq j\leq 3} \min_{1\leq k\leq 4} ||W(:,j) - \tilde{W}(:,k)||_1 \geq \frac{\epsilon}{\alpha}, \quad \text{while}$$

  - if the third column is extracted and the original matrix was M_4, we have

$$\max_{1\leq j\leq 4} \min_{1\leq k\leq 4} ||W_4(:,j) - \tilde{W}(:,k)||_1 \geq \frac{\epsilon}{\alpha}.$$

The reason is that the distance between each pair of columns of M is at least $\frac{\epsilon}{\alpha}$.

Note that the algorithm of Arora et al. [1] achieves this optimal error bound $\mathcal{O}\left(\frac{\epsilon}{\alpha}\right)$; see Theorem 3.7. However, *it requires the parameter α as an input* so that the construction above does not prove their algorithm is optimal up to some constant multiplicative factor. In fact, for the 3-separable matrix M, W is α-robustly simplicial, while for the 4-separable matrix M_4, W_4 is α'-robustly simplicial with $\alpha' \leq 2\frac{\epsilon}{\alpha} = ||W_4(:,3) - W_4(:,4)||_1$. It is possible to adjust the construction so that W and W_4 have the same condition number, proving that the algorithm of Arora et al. [1] is optimal. It suffices to add a row and a column to the input matrices as follows:

$$M' = \begin{pmatrix} M & \left(1 - \frac{\alpha}{\epsilon}\right) We \\ 0 & 1 - \frac{\alpha}{\epsilon} \end{pmatrix}, \quad \text{and} \quad M_4' = \begin{pmatrix} M_4 & \left(1 - \frac{\alpha}{\epsilon}\right) W_4 e \\ 0 & 1 - \frac{\alpha}{\epsilon} \end{pmatrix}$$

(and updating W, W_4, H, and H_4 accordingly), where M' is 4-separable with conditioning $\frac{\epsilon}{\alpha}$, while M_4' is 5-separable with the same conditioning.

3.2. Cluster identification. We now prove that there is a cluster of columns of $\tilde{M}$ around each column of W for which the sum of the corresponding diagonal entries of any feasible solution X of (1.2) is large. More formally, defining the clusters around the columns of W as

$$\Omega_k^\rho = \left\{ j \;\middle|\; ||\tilde{M}(:,j) - W(:,k)||_1 \leq \rho \right\}, \quad 1 \leq k \leq r, \tag{3.1}$$

we are going to prove that $c_k = \sum_{j\in\Omega_k^\rho} X(j,j)$ is large for any feasible solution X of (1.2), given that ϵ is sufficiently small.

Lemma 3.2. *Let $W \in \mathbb{R}^{m\times r}_+$ have its columns sum to one, and let $h \in \Delta^m$. Then, denoting $k = \operatorname{argmax}_{1\leq i\leq r} h(i)$, we have*

$$||h||_\infty = h(k) \geq 1 - \frac{\rho}{2} \quad \Rightarrow \quad ||W(:,k) - Wh||_1 \leq \rho.$$

Proof. Let us denote denote $\mathcal{R} = \{1, 2, \dots, r\} \backslash \{k\}$, we have

$$\begin{aligned} ||W(:,k) - Wh||_1 &= ||(1 - h(k))W(:,k) - W(:,\mathcal{R})h(\mathcal{R})||_1 \\ &\leq (1 - h(k))||W(:,k)||_1 + (||h||_1 - h_k)||W(:,\mathcal{R})||_1 \\ &\leq 2(1 - h(k)) \leq \rho. \qquad \square \end{aligned}$$

Lemma 3.3. *Let $M = WH$ be a normalized r-separable matrix where W is κ-robustly conical with $\kappa > 0$. Let also $\tilde{M} = M + N$, where $||N||_1 \leq \epsilon < 1$, and let X be a feasible solution of (1.2). Then, the total weight $c_k = \sum_{j\in\Omega_k^\rho} X(j,j)$ assigned to the columns of $\tilde{M}$ in Ω_k^ρ defined in (3.1) satisfies*

$$c_k \geq 1 - \frac{16\epsilon}{\kappa\rho(1-\epsilon)} \qquad \textit{for all } 1 \leq k \leq r.$$

Proof. Let $1 \leq k \leq r$ and $\mathcal{R} = \{1, 2, \ldots, r\} \backslash \{k\}$, and let us denote the indices corresponding to the columns of $\tilde{M}$ not in Ω_k^ρ as

$$\bar{\Omega}_k^\rho = \{1, 2, \ldots, n\} \backslash \Omega_k^\rho.$$

Let also j be such that $W(:,k) = M(:,j)$. By Lemma 3.2, $\max_{j \in \bar{\Omega}_k^\rho} ||H(:,j)||_\infty < 1 - \frac{\rho}{2} = \beta$. The rest of the proof is similar to that of Lemma 2.2. By Lemma 2.1, $||W(:,k) - WHX(:,j)||_1 \leq \frac{4\epsilon}{1-\epsilon}$ and $||X(:,j)||_1 \leq 1 + \frac{4\epsilon}{1-\epsilon}$. We have

$$\begin{aligned} WHX(:,j) &= W(:,k)H(k,:)X(:,j) + W(:,\mathcal{R})H(\mathcal{R},:)X(:,j) \\ &= W(:,k)\Big(H(k,\Omega_k^\rho)X(\Omega_k^\rho,j) + H(k,\bar{\Omega}_k^\rho)X(\bar{\Omega}_k^\rho,j)\Big) + W(:,\mathcal{R})y, \end{aligned}$$

where $y = H(\mathcal{R},:)X(:,j) \geq 0$, and

$$\begin{aligned} \eta &= H(k,\Omega_k^\rho)X(\Omega_k^\rho,j) + H(k,\bar{\Omega}_i^\rho)X(\bar{\Omega}_i^\rho,j) \\ &\leq ||X(\Omega_k^\rho,j)||_1 + \beta(||X(:,j)||_1 - ||X(\Omega_k^\rho,j)||_1) \leq c_k + \beta\left(1 + \frac{4\epsilon}{1-\epsilon} - c_k\right). \end{aligned}$$

The first inequality follows from $H(i,j) \leq 1$ for all i,j and $||H(k,\bar{\Omega}_k^\rho)||_\infty \leq \beta$ and the second from $X(i,j) \leq X(i,i)$ for all i,j (hence $c_k \geq ||X(\Omega_k^\rho,j)||_1$), and $\beta \leq 1$. Finally, $(1-\eta)\kappa \leq ||W(:,k) - WHX(:,j)||_1 \leq \frac{4\epsilon}{1-\epsilon}$, leading to $c_k = \sum_{j \in \Omega_k^\rho} X(j,j) \geq 1 - \frac{8\epsilon}{\kappa(1-\beta)(1-\epsilon)} = 1 - \frac{16\epsilon}{\kappa\rho(1-\epsilon)}$. □

If we can guarantee that $c_k > \frac{r}{r+1}$ for all $1 \leq k \leq r$, then the sum of the diagonal entries of X corresponding to columns of $\tilde{M}$ not in any Ω_k^ρ will be smaller than $\frac{r}{r+1}$. Therefore, if instead of picking the r largest diagonal entries of X, we cluster the diagonal entries of X depending on the distances between the corresponding columns of $\tilde{M}$, we should be able to identify the columns of W approximately; see Algorithm 3.

LEMMA 3.4. *Let $m_j \in \mathbb{R}^m$ $1 \leq j \leq n$, $x \in \mathbb{R}^n_+$ be such that $\sum_{j=1}^n x = r$, and $\rho \geq 0$. Let also $\Omega_k = \{m_j \mid ||m_j - w_k||_1 \leq \rho\}$ for $1 \leq k \leq r$, where $w_k \in \mathbb{R}^m$ $1 \leq k \leq r$. Suppose*

- $\sum_{j \in \Omega_k} x_j > \frac{r}{r+1}$,
- $\omega = \min_{i \neq j} ||w_i - w_j||_1 > 6\rho$, *and*
- *For all $1 \leq k \leq r$, there exists $1 \leq j \leq n$ such that $||m_j - w_k||_1 \leq \epsilon \leq \rho$.*

ALGORITHM 3. Extracting columns of a separable matrix by linear programming and clustering.

Input: An r-separable matrix $\tilde{M} = WH + N$ with W κ-robustly conical, the noise level $||N||_1 \leq \epsilon$, and the factorization rank r.

Output: A matrix $\tilde{W}$ such that $||\tilde{W}(:,P) - W||_1$ is small for some permutation P.

```
Compute the optimal solution $X$ of (1.2).
Initialize $\mathcal{K} = \{k \mid X(k,k) > \frac{r}{r+1}\}$ and $\nu = 2\epsilon$.
while $|\mathcal{K}| < r$ and $\nu \leq 2||\tilde{M}||_1$ do
    Compute $\mathcal{K}$ with Algorithm 4 using input $m_j = \tilde{M}(:,j)$ $1 \leq j \leq n$, $x = \mathrm{diag}(X)$
    and $\nu$;
    $\nu \leftarrow 2\nu$;
end while
$\tilde{W} = \tilde{M}(:,\mathcal{K})$ ;
```

```
Algorithm 4. Cluster extraction.
Input: A set of points $m_j$ $1 \leq j \leq n$, a vector of weights $x \in \mathbb{R}^n_+$ such that
    $\sum_{i=j}^n x = r$, and $\nu \geq 0$.
Output: A index set $\mathcal{K}$ of centroids corresponding to clusters with weight strictly
    larger than $\frac{r}{r+1}$.
$D(i,j) = ||m_i - m_j||_1$ for $1 \leq i,j \leq n$.
$\mathcal{S}_i = \{j \mid D(i,j) \leq \nu\}$ for $1 \leq i \leq n$;
$w(i) = \sum_{j \in \mathcal{S}_i} x(j)$ for $1 \leq i \leq n$;
$\mathcal{K} = \emptyset$;
while $\max_{1 \leq i \leq n} w(i) > \frac{r}{r+1}$ do
    $k = \operatorname{argmax} w(i)$;
    $\mathcal{K} \leftarrow \mathcal{K} \cup \{k\}$;
    For all $j \in \mathcal{S}_k$ : $w(j) \leftarrow 0$;
    For all $i \notin \mathcal{S}_k$ and $j \in \mathcal{S}_k$ such that $j \in \mathcal{S}_i$ : $w(i) \leftarrow w(i) - x(j)$;
end while
```

Then, for any $(\rho + \epsilon) \leq \nu \leq 2(\rho + \epsilon)$, Algorithm 4 *identifies a set $\mathcal{K}$ with r indices such that*

$$\max_{1 \leq k \leq r} \min_{j \in \mathcal{K}} ||m_j - w_k||_1 \leq 3\rho + 2\epsilon. \tag{3.2}$$

Moreover, if Algorithm 4 *identifies a set $\mathcal{K}$ with r indices for some $\nu < \rho + \epsilon$, then $\mathcal{K}$ satisfies* (3.2).

Proof. First notice that the index set $\mathcal{K}$ extracted by Algorithm 4 cannot contain more than r indices. In fact, Algorithm 4 only identifies clusters with weight strictly larger than $\frac{r}{r+1}$, while the total weight $\sum_{i=1}^n x$ is equal to r. It remains to show that $\mathcal{K}$ contains at least r indices and satisfies (3.2).

First consider the case $(\rho + \epsilon) \leq \nu \leq 2(\rho + \epsilon)$. Let $\mathcal{S}_i$ $1 \leq i \leq n$ be the sets computed by Algorithm 4 before entering the while loop. We observe the following:

- For $m_j \in \Omega_k$ and $m_{j'} \in \Omega_{k'}$, where $j \neq j'$ and $k \neq k'$, we have $m_j \notin \mathcal{S}_{j'}$ and $m_{j'} \notin \mathcal{S}_j$. In fact,
$$||m_j - m_{j'}||_1 = ||(m_i - w_k) + (w_k - w_{k'}) + (w_{k'} - m_{j'})||_1 \geq \omega - 2\rho > 4\rho \geq \nu.$$
- For all $1 \leq k \leq r$, there exists $m_j \in \Omega_k$ such that $w(j) > \frac{r}{r+1}$. By assumption, for all $1 \leq k \leq r$, there exists $m_j \in \Omega_k$ such that $||m_j - w_k||_1 \leq \epsilon$, hence for all $m_i \in \Omega_k$ we have $||m_j - m_i||_1 = ||(m_j - w_k) + (w_k - m_i)||_1 \leq \rho + \epsilon \leq \nu$, while $\sum_{i \in \Omega_k} x(i) > \frac{r}{r+1}$.
- If $m_i \notin \cup_{1 \leq k \leq r} \Omega_k$ and $w(i) > \frac{r}{r+1}$, then $||m_i - w_k||_1 \leq 3\rho + 2\epsilon$ for some $1 \leq k \leq r$. Suppose $||m_i - w_k||_1 > 3\rho + 2\epsilon$ for all k; then for all $m_j \in \cup_{1 \leq k \leq r} \Omega_k$
$$||m_i - m_j||_1 \geq ||(m_i - w_k) + (w_k - m_j)||_1 > 3\rho + 2\epsilon - \rho \geq \nu.$$
Therefore, $\sum_{j \in \mathcal{S}_i} x(j) \leq r - \sum_k \sum_{j \in \Omega_k} x_i < r - r\frac{r}{r+1} < \frac{r}{r+1}$, a contradiction. Let then k be such that $||m_i - w_k||_1 \leq 3\rho + 2\epsilon$. This implies that if $m_j \in \mathcal{S}_i$, then either $m_j \in \Omega_k$, or $m_j \notin \cup_{k' \neq k} \Omega_{k'}$. In fact, if $m_j \in \Omega_{k'}$ for some $k' \neq k$, then
$$\begin{aligned} ||m_i - m_j||_1 &\geq ||(m_i - w_k) + (w_k - w_{k'}) + (w_{k'} - m_j)||_1 \\ &\geq \omega - 3\rho - 2\epsilon - \rho > 2\rho - 2\epsilon \geq \nu, \end{aligned}$$
a contradiction.

These observations imply that there are at least r disjoint sets $\mathcal{S}_i$ with weight larger than $\frac{r}{r+1}$, each corresponding to a different cluster Ω_k. Therefore, Algorithm 4 will identify them individually and (3.2) will be satisfied.

For the case $\nu < \rho + \epsilon$, the result follows directly from the observations above: any point m_i with $w(i) > \frac{r}{r+1}$ must satisfy $||m_i - w_k||_1 \leq 3\rho + 2\epsilon$ for some $1 \leq k \leq r$. Moreover, for all k there must exist $j \in \mathcal{K}$ such that $||m_j - w_k||_1 \leq 3\rho + 2\epsilon$. In fact, suppose there exists k such that $||m_j - w_k|| > 3\rho + 2\epsilon$ for all $j \in \mathcal{K}$. Then, $m_i \notin \cup_{j \in \mathcal{K}} \mathcal{S}_j$ for all $i \in \Omega_k$ (see above) and hence

$$\sum_{i \in \mathcal{S}_j, j \in \mathcal{K}} x(i) < r - \frac{r}{r+1} = r\frac{r}{r+1},$$

which implies that $\mathcal{K}$ cannot contain more than $r - 1$ indices, a contradiction. □

THEOREM 3.5. *Let $M = WH$ be a normalized r-separable matrix with W κ-robustly conical. Let also $\tilde{M} = M + N$ with $||N||_1 \leq \epsilon$. If*

$$\epsilon < \frac{\omega\kappa}{99(r+1)}, \tag{3.3}$$

where $\omega = \min_{i \neq j} ||W(:,i) - W(:,j)||_1$, then Algorithm 3 will extract a matrix $\tilde{W}$ such that

$$||W - \tilde{W}(:,P)||_1 \leq \delta = 49(r+1)\frac{\epsilon}{\kappa} + 2\epsilon \quad \textit{for some permutation } P.$$

Proof. Let X be a feasible solution of (1.2), let the r clusters Ω_k^ρ $1 \leq k \leq r$ be defined as in (3.1), and let $c_k = \sum_{j \in \Omega_k^\rho} X(j,j)$. If $\rho < \frac{\omega}{6}$ and $c_k > \frac{r}{r+1}$, then by Lemma 3.4, Algorithm 4 will identify a set $\mathcal{K}$ with r indices such that

$$\max_{1 \leq k \leq r} \min_{j \in \mathcal{K}} ||W(:,k) - \tilde{M}(:,j)||_1 \leq \delta = 3\rho + 2\epsilon$$

for any $\nu \in [\rho + \epsilon, 2\rho + 2\epsilon]$. Therefore, starting with $\nu = 2\epsilon \leq (\rho + \epsilon)$ and multiplying it by two at each iteration will eventually give a value of ν in $[\rho + \epsilon, 2\rho + 2\epsilon]$. (Note that Algorithm 4 could return a set $\mathcal{K}$ with r indices for ν smaller than $\rho + \epsilon$; see Lemma 3.4. Note also that the number of iterations performed by Algorithm 3 is at most $\log_2\left(\frac{\rho+\epsilon}{\epsilon}\right)$.) If $\epsilon = 0$, then $c_k = 1$ for all $1 \leq k \leq r$, while $\rho = 0 < \frac{\omega}{6}$, and the loop is entered at most once. (If the entries of p are distinct, then it is not entered because exactly r diagonal entries of an optimal solution of (1.2) will be equal to one, each corresponding to a different column of W [3, Prop. 3.1].) Otherwise $\epsilon > 0$ and it remains to guarantee that $\rho < \frac{\omega}{6}$ and $c_k > \frac{r}{r+1}$. By Lemma 3.3,

$$\frac{\epsilon}{1-\epsilon} < \frac{\rho\kappa}{16(r+1)} \quad \Rightarrow \quad c_k > \frac{r}{r+1}.$$

Taking $\epsilon < \frac{\omega\kappa}{99(r+1)}$ and $\rho = \frac{98}{6}(r+1)\frac{\epsilon}{\kappa} < \frac{\omega}{6}$ completes the proof since

$$\rho = \frac{98}{6}(r+1)\frac{\epsilon}{\kappa} > 16(r+1)\frac{\epsilon}{\kappa}\left(\frac{1}{1-\epsilon}\right),$$

because $\frac{96}{1-\epsilon} < 98$ for any $0 \leq \epsilon < \frac{1}{49}$. □

It can be checked that all the results from section 2 apply to Algorithm 3. In fact, by assumption, all the matrices considered did not contain duplicates or near duplicates of the columns of matrix W, in which case we showed that r diagonal entries of X have weight at least $\frac{r}{r+1}$. This implies that Algorithm 3 will not enter the while loop; hence it is equivalent to Algorithm 1. In particular, Corollary 2.5 also applies to Algorithm 3, that is, it is necessary that

$$\epsilon < \frac{\kappa}{r-1} \qquad \text{for any } \delta < \frac{\kappa}{2}.$$

This shows that the bound of Theorem 3.5 for ϵ is tight up to a factor ω (and some constant multiplicative factor). Moreover, by Theorem 3.1, the bound for δ is tight up to a factor r (and some constant multiplicative factor).

Remark 4 (computational cost). The main additional cost of Algorithm 3 compared to Algorithm 1 is computing and storing the distance matrix D. This requires $\mathcal{O}(mn^2)$ floating point operations and $\mathcal{O}(n^2)$ space in memory. This is negligible as computing MX already requires $\mathcal{O}(mn^2)$ operations, while storing X requires $\mathcal{O}(n^2)$ space in memory.

Remark 5 (choice of the vector p). Because of the postprocessing procedure in Algorithm 3, it is not necessary for Theorem 3.5 to hold that the vector p has distinct entries. However, it will still be useful in practice to impose this condition. In fact, this will incite the weights to be concentrated in fewer diagonal entries of X so that typically fewer loops will have to be performed to obtain a set $\mathcal{K}$ containing r indices. In particular, in the exact case (that is, $\epsilon = 0$) or in the case in which there is no duplicate and near duplicate in the data set (see above), the loop will not be entered.

Remark 6 (more sophisticated postprocessing strategies). It is possible to design better postprocessing procedures but we wanted here to keep the analysis simple. In particular, if the input matrix $\tilde{M}$ does not satisfy the conditions of Theorem 3.5, it may happen that no set $\mathcal{K}$ computed in the loop of Algorithm 3 contains r elements. Therefore, one should keep in memory the largest set extracted so far or design more sophisticated strategies. For example, if fewer than r clusters have been extracted, the condition that the weight of each extracted cluster must larger than $\frac{r}{r+1}$ can be relaxed; this variant has been implemented in the MATLAB code available at https://sites.google.com/site/nicolasgillis/code.

Remark 7 (preprocessing). Another possible way to deal with duplicates and near duplicates would be to use an appropriate preprocessing. In [5], k-means is used to reduce the number of data points and get rid of the duplicates. In fact, their algorithm cannot deal with duplicates, even in the noiseless case. (Note that their robustness result is only asymptotical, that is, it holds only when the noise level ϵ goes to zero.) Arora et al. [1] also use some preprocessing in their algorithm (before processing any data point, its neighbors have to be discarded). However, it seems difficult to combine a robustness analysis with a preprocessing strategy (in fact, Arora et al. [1] need the conditioning α as an input to do so); this is a topic for further research.

3.3. Repartition of the weights inside a cluster. In this section, we show that Hottopixx (Algorithm 1) cannot provide better bounds than Algorithm 3. The reason is the following: inside a cluster Ω_k^ρ, there is no guarantee that all the weight will be assigned to a single diagonal entry of X. In the proof of Theorem 3.6, we show that the weight may be equally distributed inside a cluster. This construction allows us to show that $\epsilon \leq \frac{\kappa}{(r-1)^2}$ is necessary for Proportion 2 to hold for any $\delta < \kappa + \epsilon$, which proves our claim.

THEOREM 3.6. *For any $r \geq 3$ and $\delta < \kappa + \epsilon$, it is necessary for Proposition* 2 *to hold that*

$$\epsilon \leq \frac{\kappa}{(r-1)^2}.$$

Proof. See Appendix C. □

Remark 8. It remains an open question whether there exists a bound on the noise level to guarantee Hottopixx to be robust for any separable matrix, that is, one that also contains duplicates and near duplicates. (Note that by Theorem 3.6, this bound, if it exists, has to be smaller than $\frac{\kappa}{(r-1)^2}$.)

3.4. Comparison with the algorithm of Arora et al. [1]. In this section, we compare the theoretical bounds for Algorithm 3 obtained in Theorem 3.5 with the ones of the algorithm of Arora et al. for which the following holds.

THEOREM 3.7 (see [1], Thm. 5.7). *Let $M = WH$ be a normalized r-separable matrix with W α-robustly simplicial. Let also $\tilde{M} = M + N$ with $||N||_1 \leq \epsilon$. If*

$$\epsilon < \frac{\alpha^2}{20 + 13\alpha}, \tag{3.4}$$

then the algorithm proposed by Arora et al. [1] *extracts a matrix $\tilde{W}$ such that*

$$||W - \tilde{W}(:, P)||_1 \leq 10\frac{\epsilon}{\alpha} + 6\epsilon \quad \textit{for some permutation } P.$$

There are two bounds to compare. First, there is the bound on noise level ϵ allowed to have any error guarantee. The one from (3.4) does not dominate the one from Theorem 3.5; see (3.3). In fact, α and κ differ by only a factor of at most two (Theorem 1.3), while ω ($\geq \kappa \geq \frac{\alpha}{2}$) can potentially be arbitrarily larger than α (take, for example, the columns of W as the vertices of a flat triangle). Hence, for some highly ill-conditioned matrices, Algorithm 3 can tolerate much higher noise levels.

Second, there is the bound on the error: the algorithm of Arora et al. dominates the one of Algorithm 3, but only up to a factor r (which is usually small in practice). This is not very surprising since the algorithm of Arora et al. is optimal in terms of the error bound; see section 3.1. However, *the algorithm of Arora et al. requires the parameter α as an input,* which, we believe, is highly impractical. At least, we do not know of an efficient way to compute α (and this issue is not discussed in their paper). Moreover, it was observed in [3] that Hottopixx performs better than the algorithm of Arora et al. on some synthetic data sets.

To conclude, Algorithm 3 is, to the best of our knowledge, the provably most robust algorithm for separable NMF for which the condition number α is not required as an input.

4. Numerical experiments.

In this section, we present some numerical experiments to show the superiority of Algorithm 3 over Hottopixx in the case in which there are duplicates and near duplicates of the columns of W in the data set; otherwise both algorithms coincide since the postprocessing will not be entered (cf. the discussion after Theorem 3.5).

All experiments were run on a two-core machine with 2.99 GHz and 2 GB of RAM using a CPLEX implementation to solve the LP (1.2); the code is available at https://sites.google.com/site/nicolasgillis/code and was developed in [6]. The constructions of Theorems 3.6 and the postprocessing procedure (Algorithm 3) can be found on the same web page.

We use the constructions from the proof of Theorem 3.6 with the parameters $\kappa = 0.1$ and $K = 5$, while we vary the value of the rank r and the noise level ϵ. Moreover, we duplicate each column of W twice (that is, each column of W is present three times in the data set) and permute the columns of $\tilde{M}$ at random in order to avoid a bias toward the natural ordering. Finally, we slightly perturb the vector p in the objective function to make its entries distinct (since we also duplicated the entries of p) by adding to each entry a value drawn from the normal distribution with mean zero and standard deviation 0.1.

Figure 4.1 displays the percentage of columns of W correctly extracted by the two algorithms for different values of the rank r and of the noise level ϵ (hence the higher the curve, the better). As shown in Theorem 3.6, Hottopixx cannot extract one of the columns of W, even for small noise levels. More interestingly, Algorithm 3 clearly outperforms Hottopixx for larger noise levels. For example, for $r = 40$ and $\epsilon = 0.046$ (see the plot in the bottom right corner of Figure 4.1), Hottopixx identifies only 35% of the 40 columns of W, while Algorithm 3 identifies 95% of them. The reason for this behavior is the following: when the noise is large, the set of feasible solutions of (1.2) is typically larger (because the constraint $||M - MX||_1 \leq 2\epsilon$ is relaxed). For ϵ sufficiently large, this allows all duplicates of some columns of W corresponding to small entries of p to be given a large weight, and hence some columns of W are extracted more than once. This is not possible with the postprocessing which clusters these entries

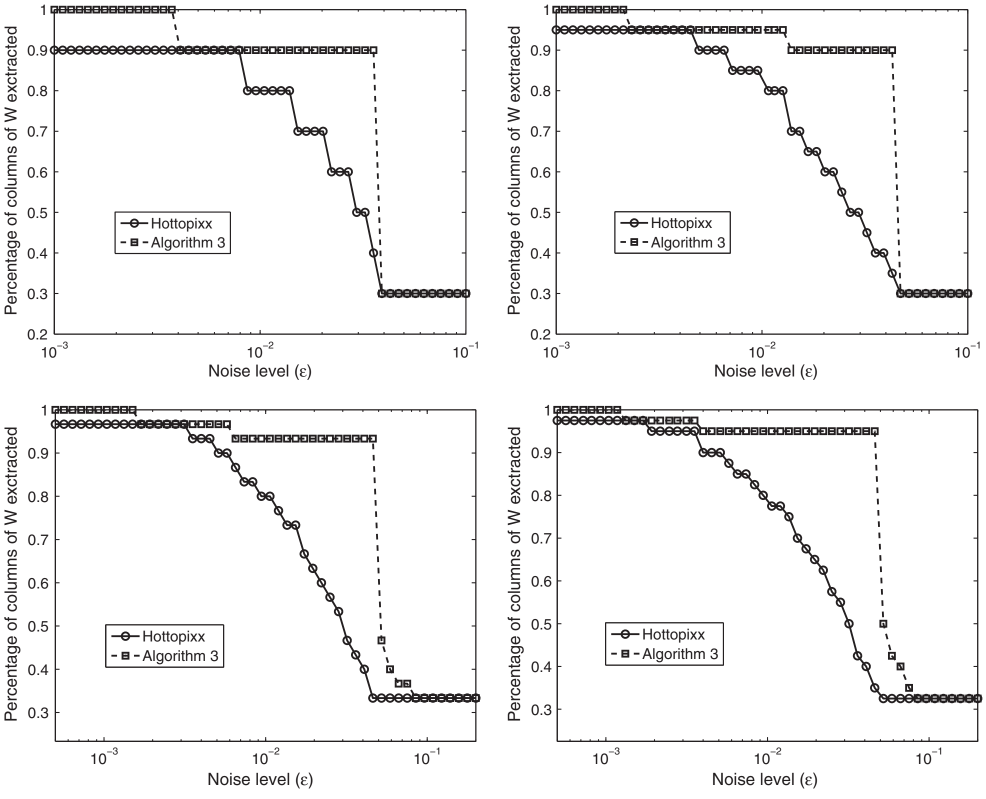

FIG. 4.1. *Comparison of Hottopixx and Algorithm* 3 *on the near-separable matrices from Theorem* 3.6 *with duplicates of the columns of* W*. From left to right, top to bottom:* $r = 10, 20, 30, 40$.

together and is then able to extract other columns of W whose corresponding diagonal entries of X are smaller.

Note that the computational time of the postprocessing strategy is negligible and takes in average about 5% percent of the total time needed to solve the LP (1.2).

5. Conclusion and further work.

In this paper, we have proposed a provably more robust variant of Hottopixx based on an appropriate postprocessing of the solution of the linear program (1.2) (see Algorithm 3). In particular, we proved that Algorithm 3 is robust for any input separable matrix M (Theorem 3.5), while our analysis is close to being tight.

It would be interesting to improve the bound of Theorem 3.5 or show that the bound is tight. It would also be particularly interesting to design more robust or computationally more effective (or both?) separable NMF algorithms. In particular, the following question seems to be open: does there exist a polynomial-time algorithm to which only the noisy separable matrix $\tilde{M}$ and the noise level ϵ are given as input and that achieves an error of order $\mathcal{O}(\frac{\epsilon}{\alpha})$? Such an algorithm would be optimal (see section 3.1). Note that the algorithm of Arora et al. [1] achieves this bound but requires the parameter α as an input, which is highly impractical; see section 3.4.

Appendix A. Proof of Theorem 1.3.

Proof of Theorem 1.3. Let

$$k = \text{argmin}_{1\leq j \leq r} \min_{x\in\mathbb{R}^{r-1}_+} ||W(:,j) - W(:,\mathcal{J})x||_1, \quad \text{ where } \mathcal{J} = \{1,2,\dots,r\}\backslash\{j\},$$

$w = W(:,k)$, and y$= W(:,\mathcal{R})x^*$, where

$$x^* = \text{argmin}_{x\in\mathbb{R}^{r-1}_+} ||W(:,k) - W(:,\mathcal{R})x||_1, \quad \text{ where } \mathcal{R} = \{1,2,\dots,r\}\backslash\{k\},$$

so that by definition, $||y-w||_1 = \kappa$. If $y = 0$, we are done since $\kappa = ||w||_1 = 1 \geq \frac{1}{2}\alpha$ as $\alpha \leq 2$. Otherwise $y \neq 0$ and we define $z = \frac{y}{||y||_1} = \lambda^{-1}y$. By definition, $||w-z||_1 \geq \alpha$ since z belongs to the convex hull of the columns of W. We have

$$\alpha \leq ||w-z||_1 = ||w-(\lambda+1-\lambda)z||_1 \leq ||w-\lambda z||_1 + (1-\lambda)||z||_1 = \kappa + (1-\lambda) \leq 2\kappa$$

since $\kappa = ||w-\lambda z||_1 \geq ||w||_1 - ||\lambda z||_1 = 1-\lambda$, and the proof is complete. □

Appendix B. Proof of Theorem 2.4.

The following lemma shows that if one of the coefficients in the objective function of a linear program is much larger than all the other ones, then the corresponding entry of any optimal solution must be smaller than the corresponding entry of any feasible solution. Although the result is clear intuitively, we provide here a simple proof.

LEMMA B.1. *Consider the linear program*

$$\min_{x\in\mathbb{R}^n} c_K^T x \quad \textit{such that } \; Ax = b \; \textit{ and } \; l \leq x \leq u \tag{B.1}$$

with $l, u \in \mathbb{R}^n$, $l \leq u$, and $c_K = (K, \tilde{c}) \in \mathbb{R}^n$, where $K \in \mathbb{R}$ is a parameter. Let us denote x_K^* an optimal solution of (B.1) *depending on K. Assume there exists a feasible solution x^f of* (B.1) *such that $x^f(1) = s$. Then, for any K sufficiently large, $x_K^*(1) \leq s$.*

Similarly, if $c_K(1) = -K$ and there exists a feasible solution such that $x(1) = t$, then for any K sufficiently large, $x_K^*(1) \geq t$.

Proof. Let $\mathcal{V} \neq \emptyset$ be the set of vertices of the feasible set of (B.1) and $\bar{\mathcal{V}} = \{x \in \mathcal{V} \mid x(1) > s\}$. Notice that because the feasible set of (B.1) is a polytope, there always exists an optimal solution in $\mathcal{V}$. Let us denote $d = \min_{x \in \bar{\mathcal{V}}} x(1) > s$. Assume there exists an optimal solution x^*_K such that $x^*_K(1) > s$. This implies that there exists an optimal solution $\bar{x}^*_K \in \bar{\mathcal{V}}$ (since any optimal solution is a convex combination of optimal vertices in $\mathcal{V}$). Therefore,

$$Kd - ||\tilde{c}||_2||u||_2 \leq c_K^T x^*_K = c_K^T \bar{x}^*_K = K\bar{x}^*_K(1) + \tilde{c}^T \bar{x}^*_K(2{:}n) \leq c_K^T x^f \leq Ks + ||\tilde{c}||_2||u||_2,$$

which is absurd for any $K > \frac{2||\tilde{c}||_2||u||_2}{d-s}$. □

The linear program (1.2) can be written in the form of (B.1); in fact, $0 \leq X \leq 1$, while the mn additional variables necessary to express the constraint $||M - MX||_1 \leq 2\epsilon$ linearly will be in the interval $[0, 2\epsilon]$. Therefore, Lemma B.1 applies to (1.2).

Proof of Theorem 2.4. We prove the result with the following construction: Let

$$W = \begin{pmatrix} \frac{\kappa}{2} I_r \\ (1 - \frac{\kappa}{2}) e^T \end{pmatrix},$$

which is κ-robustly conical. Let also

$$H = \begin{pmatrix} I_r & \beta I_r + (E_r - I_r)\frac{1-\beta}{r-1} \end{pmatrix},$$

so that $\max_{i,j} H'_{ij} = \beta$ (note that β must be larger than $\frac{1}{r}$ since the columns of H' sum to one), $N = 0$, $\tilde{M} = WH + N$, $p = (1, 2, 3, \ldots, r-1, -K, -1, -2, \ldots, -(r-1), -K^2)^T$ for K sufficiently large, and

$$\epsilon = \frac{\kappa(1-\beta)}{(r-1)(1-\beta)+1} \leq \frac{\kappa}{r-1}.$$

Assume that

$$X = \begin{pmatrix} (1-\delta)I_{r-1} + \frac{\delta-\omega}{r-1} J_{r-1} & 0 & (1-\delta)\left(\beta I_{r-1} + \frac{1}{r-1} J_{r-1}\left(\frac{1-\omega}{1-\delta} - \beta\right)\right) & 0 \\ \frac{\delta-\omega}{r-1} e^T & 1 & \frac{1-\delta}{r-1}\left(\frac{1-\omega}{1-\delta} - \beta\right) e^T & 0 \\ \omega I_{r-1} & 0 & \omega I_{r-1} & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix},$$

where $J_{r-1} = E_{r-1} - I_{r-1}$,

$$\delta = (2-\beta)\omega \quad \text{and} \quad \omega = \frac{\epsilon}{\kappa(1-\beta)}, \qquad \text{implying } X(n,n) = \omega + (r-1)(\delta - \omega) = 1,$$

is a feasible solution of (1.2) (note that $n = 2r$). By Lemma B.1, there exists K sufficiently large such that any optimal solution X^* must satisfy $X^*(n,n) = 1$. Using Lemma B.1 again, there exists K sufficiently large such that $X^*(r,r) = 1$. Therefore, for K sufficiently large, the rth and nth columns of $\tilde{M}$ will be extracted, implying

$$\begin{aligned} ||W - \tilde{W}(:,P)||_1 &= \min_{1 \leq j \leq r-1} ||W(:,j) - M(:,n)||_1 \\ &= ||W(:,1) - M(:,n)||_1 = \kappa\frac{r-2+\beta}{r-1} > \frac{r-2}{r-1}\kappa \geq \epsilon, \end{aligned}$$

and the proof will be complete.

It remains to show that X is feasible: Clearly, $\operatorname{tr}(X) = r$. For the constraints $0 \leq X \leq 1$, we check that

$$0 \leq \omega = \frac{\epsilon}{\kappa(1-\beta)} = \frac{1}{r(1-\beta)+1} \leq \delta = (2-\beta)\omega = \frac{(1-\beta)+1}{r(1-\beta)+1} \leq 1,$$

and

$$0 \leq \frac{1}{r-1}\left(\frac{1-\omega}{1-\delta} - \beta\right) \leq 1 \quad \text{ since } \frac{1-\omega}{1-\delta} = \frac{r-1}{r-2} \geq 1.$$

For $X(i,j) \leq X(i,i)$ for all i, j, we only have to check that

$$1 - \delta \geq \frac{\delta-\omega}{r-1} \iff (r-1)(r-2)(1-\beta) \geq (1-\beta).$$

It remains to verify that $||M(:,j) - MX(:,j)||_1 \leq 2\epsilon$ for all $1 \leq j \leq 2r$:

- $1 \leq j \leq r-1$. Letting $\mathcal{J} = \{1, 2, \dots, r\} \backslash \{j\}$, we have

$$\begin{aligned} ||M(:,j) - MX(:,j)||_1 &= \left\| M(:,j) - (1-\delta)M(:,j) - \frac{\delta-\omega}{r-1} M(:,j+r) \right\|_1 \\ &= \left\| \delta M(:,j) - \omega M(:,j+r) - \frac{\delta-\omega}{r-1} M(:,\mathcal{J})e \right\|_1 \\ &= \omega \left\| M(:,j) - M(:,j+r) \right\|_1 \\ &\quad + (\delta-\omega) \left\| M(:,j) - \frac{1}{r-1} M(:,\mathcal{J})e \right\|_1 \\ &= \omega\kappa(1-\beta) + (\delta-\omega)\kappa = 2\omega\kappa(1-\beta) = 2\epsilon. \end{aligned}$$

- $r+1 \leq j \leq 2r-1$. Letting $\mathcal{R} = \{1, 2, \dots, r\} \backslash \{j-r\}$ and $w_j = W(:,\mathcal{R})\frac{e}{r-1}$, we have

$$\begin{aligned} &||M(:,j) - MX(:,j)||_1 \\ &\quad = \left\| M(:,j) - \omega M(:,j) - (1-\delta)\beta W(:,j-r) - (1-\omega-\beta(1-\delta))\, w_j \right\|_1 \\ &\quad = (1-\omega) \left\| M(:,j) - \frac{r-2}{r-1}\beta M(:,j-r) - \left(1 - \beta\frac{r-2}{r-1}\right) w_j \right\|_1 \\ &\quad = (1-\omega) \left\| M(:,j) - \left(1 - \frac{1}{r-1}\right)\beta W(:,j-r) - \left(1 - \beta + \beta\frac{1}{r-1}\right) w_j \right\|_1 \\ &\quad = \frac{\beta(1-\omega)}{r-1} \left\| W(:,j-r) - w_j \right\|_1 = \frac{\beta(1-\omega)\kappa}{r-1} \leq \frac{(1-\omega)\kappa}{r} \\ &\quad = \frac{(r-1)(1-\beta)\kappa}{r((r-1)(1-\beta)+1)} \leq \frac{(1-\beta)\kappa}{(r-1)(1-\beta)+1} = \epsilon. \end{aligned}$$

In fact, $\frac{1-\delta}{1-\omega} = \frac{r-2}{r-1}$, $\beta \leq \frac{1}{r}$, and, by construction, $M(:,j) = \beta W(:,j-r) + (1-\beta)w_j$. □

Appendix C. Proof of Theorem 3.6.

Proof of Theorem 3.6. We prove the result with the following construction: Let

$$W = \begin{pmatrix} \frac{\kappa}{2} I_r \\ (1-\frac{\kappa}{2})e^T \\ 0_{r \times r} \end{pmatrix}, \quad H = \begin{pmatrix} I_{r-1} & 0 & \lambda I_{r-1} & \frac{1}{r-1}e \\ 0 & 1 & (1-\lambda)e^T & 0 \end{pmatrix},$$

where $\lambda = 2\frac{\epsilon}{\kappa}$,

$$N = \begin{pmatrix} 0_{(r+1)\times r} & 0_{(r+1)\times 1} & 0_{(r+1)\times(r-1)} & 0_{(r+1)\times 1} \\ \epsilon e^T & 0 & 0_{1\times(r-1)} & \epsilon \\ 0_{(r-1)\times(r-1)} & 0_{(r-1)\times 1} & Z & 0_{(r-1)\times 1} \end{pmatrix},$$

where $Z = xI_{r-1} + y(E_{r-1} - I_{r-1})$ with $x = \frac{1}{r-1}\epsilon$ and $y = \frac{-x}{r-2}$. The matrix Z has been constructed so that $||Z(:,j)||_1 \leq \epsilon$ for all j, $\sum_j Z(i,j) = 0$ for all i, and $\|\tilde{M}(:,j) - \frac{1}{r-1}\tilde{M}(:,\mathcal{I})e\|_1 = 2\epsilon$ for all $j \in \mathcal{J} = \{r+1, r+2, \dots 2r-1\}$ and $\mathcal{I} = \mathcal{J}\backslash\{i\}$. Let also $\tilde{M} = WH + N$,

$$\frac{\kappa}{(r-1)^2} < \epsilon \leq \frac{\kappa}{2(r-1)} \quad \text{ so that } \lambda \leq \frac{1}{r-1},$$

and

$$p = (1, 2, \dots, r-1, K^3, K^2, K^2+1, \dots, K^2+r-1, -K)^T$$

for K sufficiently large. Assume

$$X = \begin{pmatrix} (1 - \frac{2\epsilon}{\kappa})I_{r-1} & 0 & \lambda\frac{r-2}{r-1}I_{r-1} & \left(1 - \frac{2\epsilon(r-1)}{\kappa}\right) e^T \\ 0 & 0 & 0 & 0 \\ 0 & \frac{1}{r-1}e & \frac{1}{r-1}E_{r-1} & 0 \\ \frac{2\epsilon}{\kappa}e^T & 0 & 0 & (r-1)\frac{2\epsilon}{\kappa} \end{pmatrix}$$

is feasible for (1.2). Letting X^* be any optimal solution, by Lemma B.1 there exists K sufficiently large such that $X^*(r,r) = 0$. By Lemma C.1 (see below), this implies that $X^*(j,j) \geq \frac{1}{r-1}$ for $j \in \mathcal{J}$. Using Lemma B.1 again, we have that for K sufficiently large $X^*(j,j) = \frac{1}{r-1}$ for all for $j \in \mathcal{J}$, and $X^*(n,n) \geq (r-1)\frac{2\epsilon}{\kappa}$. Therefore, since

$$\frac{2\epsilon(r-1)}{\kappa} > \frac{1}{r-1} \iff \epsilon > \frac{\kappa}{2(r-1)^2}$$

and

$$1 - \frac{2\epsilon}{\kappa} > \frac{1}{r-1} \iff \epsilon < \left(\frac{r-2}{r-1}\right)\frac{\kappa}{2} \leq \frac{\kappa}{4},$$

the first $r-1$ columns and the last column of $\tilde{M}$ will be extracted so that

$$||W - \tilde{W}||_1 = ||W(:,r) - \tilde{M}(:,n)||_1 = \kappa + \epsilon,$$

and the proof will be complete.

It remains to show that X is feasible. We clearly have $\text{tr}(X) = r$, $0 \leq X \leq 1$, and $X(i,j) \leq X(i,i)$ for all i,j because $\epsilon \leq \frac{\kappa}{2(r-1)}$, while for $||\tilde{M}(:,j) - \tilde{M}X(:,j)||_1 \leq 2\epsilon$ for all j, we have the following:

- $1 \leq j \leq r-1$,

$$||\tilde{M}(:,j) - \tilde{M}X(:,j)||_1 = \frac{2\epsilon}{\kappa}||\tilde{M}(:,j) - \tilde{M}(:,n)||_1 = 2\frac{r-2}{r-1}\epsilon \leq 2\epsilon.$$

- $j = r$. This follows from Lemma C.1.

- $r+1 \le j \le 2r-1$. This follows from the construction of matrix Z.
- $j = 2r$. $\tilde{M}(:,j) = \tilde{M}X(:,j)$ since $\tilde{M}(:,j) = \frac{1}{r-1}W(:,1{:}r-1)e$. $\square$

LEMMA C.1. *Let W, H, N and $\tilde{M} = WH + N$ be the matrices constructed in Theorem* 3.6. *Let also $\mathcal{R} = \{1,2,\dots,n\}\backslash\{r\}$. Then*

$$\min_{x\geq 0} ||\tilde{M}(:,r) - \tilde{M}(:,\mathcal{R})x||_1 = 2\epsilon, \tag{C.1}$$

and the unique *optimal solution of* (C.1) *is given by*

$$x^\dagger = \begin{pmatrix} 0_{(r-1)\times 1} \\ \frac{1}{r-1}e \\ 0_{1\times 1} \end{pmatrix} \in \mathbb{R}^{2r-1}.$$

Proof. Let $x^* = (y,z,w)$ be an optimal solution of (C.1), where $y, z \in \mathbb{R}^{r-1}_+$ and $w \in \mathbb{R}_+$. We have to show that $x^* = x^\dagger$. From x^*, let us construct another optimal solution $x' = (y', z', 0)$ such that all the entries of y' and z' are equal to each other. Because $\tilde{M}(:,n) = \frac{1}{r-1}\tilde{M}(:,1{:}r-1)e$, we take $w = 0$, replace $z \leftarrow z + \frac{w}{r-1}$, and obtain an equivalent solution. Let us denote $\bar{\mathcal{R}} = \mathcal{R}\backslash\{n\}$ and

$$g(y,z) = \left\| \tilde{M}(:,r) - \tilde{M}(:,\bar{\mathcal{R}})\begin{pmatrix} y \\ z \end{pmatrix} \right\|_1 .$$

By symmetry, one can check that $g(y,z) = g(y(P), z(P))$ for any permutation P of $\{1,2,\dots,r-1\}$. (This simply amounts to permuting the first and last $r-1$ columns of $M(:,\bar{\mathcal{R}})$.) By convexity, $(y',z') = \frac{1}{|\Pi|}\sum_{P\in\Pi}(y(P), z(P))$, where Π is the set of all possible permutations of $\{1,2,\dots,r-1\}$, is also an optimal solution of (C.1); hence all entries of y' and z' are equal to each other, and $||y'||_1 = ||y||_1$ and $||z'||_1 = ||z||_1$.

Therefore, denoting $y'(i) = \frac{a}{r-1}$ and $z'(i) = \frac{b}{r-1}$ for all $1 \le i \le r-1$, the optimization problem (C.1) can be reduced to

$$\min_{a,b\geq 0} \left\| \begin{pmatrix} 0_{(r-1)\times 1} \\ \frac{\kappa}{2} \\ 1-\frac{\kappa}{2} \\ 0 \\ 0_{(r-1)\times 1} \end{pmatrix} - a \begin{pmatrix} \frac{\kappa}{2(r-1)}e \\ 0 \\ 1-\frac{\kappa}{2} \\ \epsilon \\ 0_{(r-1)\times 1} \end{pmatrix} - b \begin{pmatrix} \frac{\lambda\kappa}{2(r-1)}e \\ (1-\lambda)\frac{\kappa}{2} \\ 1-\frac{\kappa}{2} \\ 0 \\ 0_{(r-1)\times 1} \end{pmatrix} \right\|_1 \tag{C.2}$$

$$\equiv \min_{a,b\geq 0}\ h(a,b) = \frac{\kappa}{2}|a+\lambda b| + \frac{\kappa}{2}|1-(1-\lambda)b| + \left(1-\frac{\kappa}{2}\right)|1-a-b| + \epsilon\,|a|\,.$$

Let us show that $(a^*, b^*) = (0,1)$ is the unique optimal solution, for which $h(0,1) = \kappa\lambda = 2\epsilon$. First, note that $(0,0)$ cannot be optimal since $h(0,0) = 1$. For $a+b > 1$, the subdifferential of h in a is larger than $1-\epsilon > 0$, while for $0 < a+b < 1$, the subdifferential of h in b is $\kappa\lambda - 1 = 2\epsilon - 1 < 0$ (recall that $\lambda = \frac{2\epsilon}{\kappa}$, $\epsilon \le \frac{\kappa}{2(r-1)}$, $\kappa \le 1$, and $r \ge 3$); hence $a^* + b^* = 1$ at optimality. Substituting $a = 1-b$ above, we obtain

$$b^* = \operatorname{argmin}_{0\le b\le 1}\ 2|1-(1-\lambda)b| = 1,$$

which is unique as the slope at $b = 1$ is negative (since $0 \le \lambda \le 1/2$).

Finally, we have $b^* = 1$, $a^* = 0$ is the unique solution of (C.2), implying that $y' = y = 0$ and that the minimal objective function value of (C.1) is 2ϵ. Moreover, this implies $||z'||_1 = ||z||_1 = 1$. It remains to show that the entries of z are equal to each other, that is, show that the unique solution to the system

$$\left\| \begin{pmatrix} 0_{(r-1)\times 1} \\ \frac{\kappa}{2} \\ 0_{(r-1)\times 1} \end{pmatrix} - \begin{pmatrix} \frac{\lambda\kappa}{2} I_{r-1} \\ (1-\lambda)\frac{\kappa}{2} \\ Z \end{pmatrix} z \right\|_1 = 2\epsilon$$

is $z^* = \frac{1}{r-1}e$, which is clearly the case as the only z such that $Zz = 0$ and $||z||_1 = 1$ is z^*. This completes the proof. □

Acknowledgments. The author would like to thank Robert Luce (T.U. Berlin) for some insightful discussions and for pointing out an error in Lemma 2.1 in a previous draft. The author would also like to thank the reviewers for their feedback, which helped improve the paper significantly.

REFERENCES

[1] S. Arora, R. Ge, R. Kannan, and A. Moitra, *Computing a nonnegative matrix factorization—provably*, in Proceedings of the 44th Symposium on Theory of Computing, STOC'12, 2012, pp. 145–162.

[2] S. Arora, R. Ge, and A. Moitra, *Learning topic models—going beyond SVD*, in Proceedings of the 53rd Annual IEEE Symposium on Foundations of Computer Science, FOCS'12, 2012, pp. 1–10.

[3] V. Bittorf, B. Recht, E. Ré, and J. Tropp, *Factoring nonnegative matrices with linear programs*, in Adv. Neural Inform. Process. Syst., 25 (2012), pp. 1223–1231.

[4] E. Elhamifar, G. Sapiro, and R. Vidal, *See all by looking at a few: Sparse modeling for finding representative objects*, in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2012.

[5] E. Esser, M. Moller, S. Osher, G. Sapiro, and J. Xin, *A convex model for nonnegative matrix factorization and dimensionality reduction on physical space*, IEEE Trans. Image Process., 21 (2012), pp. 3239–3252.

[6] N. Gillis and R. Luce, *Robust Near-Separable Nonnegative Matrix Factorization Using Linear Optimization*, arXiv:1302.4385, 2013.

[7] N. Gillis and S. Vavasis, *Fast and Robust Recursive Algorithms for Separable Nonnegative Matrix Factorization*, arXiv:1208.1237, 2012.

[8] A. Kumar, V. Sindhwani, and P. Kambadur, *Fast conical hull algorithms for near-separable non-negative matrix factorization*, in Proceedings of the International Conference on Machine Learning (ICML), 2013.

[9] D. Lee and H. Seung, *Learning the parts of objects by nonnegative matrix factorization*, Nature, 401 (1999), pp. 788–791.

[10] S. Vavasis, *On the complexity of nonnegative matrix factorization*, SIAM J. Optim., 20 (2009), pp. 1364–1377.