Semidefinite Programming Based Preconditioning for More Robust Near-Separable Nonnegative Matrix Factorization*

Nicolas Gillis
Faculté Polytechnique, Université de Mons
Department of Mathematics and Operational Research
Rue de Houdain 9, 7000 Mons, Belgium
Email: nicolas.gillis@umons.ac.be

Stephen A. Vavasis
University of Waterloo
Department of Combinatorics and Optimization
Waterloo, Ontario N2L 3G1, Canada
Email: vavasis@math.uwaterloo.ca

Abstract

Nonnegative matrix factorization (NMF) under the separability assumption can provably be solved efficiently, even in the presence of noise, and has been shown to be a powerful technique in document classification and hyperspectral unmixing. This problem is referred to as near-separable NMF and requires that there exists a cone spanned by a small subset of the columns of the input nonnegative matrix approximately containing all columns. In this paper, we propose a preconditioning based on semidefinite programming making the input matrix well-conditioned. This in turn can improve significantly the performance of near-separable NMF algorithms which is illustrated on the popular successive projection algorithm (SPA). The new preconditioned SPA is provably more robust to noise, and outperforms SPA on several synthetic data sets. We also show how an active-set method allow us to apply the preconditioning on large-scale real-world hyperspectral images.

Keywords. nonnegative matrix factorization, semidefinite programming, preconditioning, separability, robustness to noise.

1 Introduction

Nonnegative matrix factorization (NMF) has become a standard technique in machine learning and data analysis. NMF is a linear dimensionality reduction technique for nonnegative data where both the basis elements and the weights of the linear combinations are imposed to be nonnegative: Given an m-by-n nonnegative matrix M and a factorization rank r, NMF produces an m-by-y nonnegative matrix U and an r-by-n nonnegative matrix V such that $UV \approx M$. The columns of M, which usually represent elements of a data set such as images or documents, are approximately reconstructed using

---

nonnegative linear combinations of the columns of U since $M(:,j) \approx \sum_{k=1}^{r} U(:,k)V(k,j)$ for all j. The advantage of the nonnegativity constraints is two fold. First, the nonnegativity of the basis elements (that is, the columns of U) allows us to interpret them in the same way as the data (e.g., as images or documents). Second, the nonnegativity of the weights in the linear combinations only allows an additive reconstruction of the data points from the basis elements leading to a parts-based and sparse representation of the data [24].

Unfortunately, NMF is NP-hard [33] and highly ill-posed [12]. Therefore, in practice, people usually use standard nonlinear optimization techniques to find locally optimal solutions. Hence most NMF algorithms come with no guarantee. However, NMF algorithms have been proved successful in a series of applications which suggests that some real-world NMF problems might not be as difficult as the general NMF problem. In fact, it was recently shown by Arora et al. [4] that there exists a subclass of nonnegative matrices, referred to as separable, for which the NMF problem can be solved in polynomial time. Separability requires that there exists an NMF (U, V) of the input matrix M such that $M = UV$ and where each column of U is equal to a column of M. In other terms, a matrix M is r-separable if there exists an index set $\mathcal{K} \subseteq \{1, 2, \dots, n\}$ with cardinality r and an r-by-n nonnegative matrix V such that $M = M(:, \mathcal{K})V$. This is equivalent to requiring the cone spanned by a subset of r columns of M to contain all columns of M. Although this condition is rather strong, it appears to make sense in several applications. For example,

- In document classification, separability requires that for each topic there exists a word used only by that topic (it is referred to as an 'anchor' work) [5, 3].

- In hyperspectral unmixing, separability requires that for each constitutive material present in the hyperspectral image there exists a pixel containing only that material (it is referred to as a 'pure' pixel). The separability assumption coincides with the so called pure-pixel assumption; see Section 5.1.2.

- In blind source separation, separability requires that for each source there exists a moment in time where only that source is active; see, e.g., [9, 10] and the references therein.

1.1   Near-Separable NMF

In practice, the input separable matrix is perturbed with some noise and it is important to design algorithms robust to noise. Note that, in the noiseless case, the separable NMF problem is relatively easy and reduces to identify the extreme rays of the cone spanned by a finite set of points (or the vertices of the convex hull of a set of points after normalization of the columns of input matrix); see, e.g., [22]. The separable NMF problem with noise is referred to as near-separable NMF, and can be defined as follows [14].

> (Near-Separable NMF) *Given a noisy r-separable matrix $\tilde{M} = M + N$ with $M = WH = W[I_r, H']\Pi$ where W and H' are nonnegative matrices, I_r is the r-by-r identity matrix, Π is a permutation matrix and N is the noise with $\max_i ||N(:,i)||_2 \leq \epsilon$ for some $\epsilon \geq 0$, find a set $\mathcal{K}$ of r indices such that $\tilde{M}(:, \mathcal{K}) \approx W$.*

Several algorithms have been proposed to solve this problem [4, 5, 7, 13, 15]. All these algorithms are sensitive to the conditioning of the input data matrix [13]. For example, it is necessary that $\epsilon < \sigma_{\min}(W)$ where $\sigma_{\min}(W)$ is the smallest singular value of W, otherwise some columns of W could potentially be set to zero (take for example $W = \sigma I_r$ and the noise corresponding to the columns of

W as $-\sigma I_r$). Hence the near-separable NMF problem would not really make sense since extracting some zero columns would solve the problem.

In this paper, we will assume without loss of generality that the columns of H sum to at most one (this can be obtained by normalizing the columns of the input matrix). More precisely we assume that the input data matrix has the following form [15].

Assumption 1 (Near-Separable Matrix). *The separable noiseless matrix M can be written as $M = WH \in \mathbb{R}^{m \times n}$ where $W \in \mathbb{R}^{m \times r}$ is full rank, $H = [I_r \; H'] \in \mathbb{R}_+^{r \times n}$ and the sum of the entries of each column of H' is at most one. The near-separable matrix $\tilde{M} = M + N$ is the perturbed matrix M, with*

$$||N(:,i)||_2 = ||\tilde{M}(:,i) - M(:,i)||_2 \leq \epsilon \quad \text{for all } i.$$

Note that this assumption on the sum of the entries of each column of H makes the presentation easier. In fact, otherwise the upper bound on the noise added to each column will have to be different since columns with small norm cannot tolerate as much noise (we would need to define something like $||n_i||_2 \leq \epsilon ||h_i||_2$).

Finally, given a matrix satisfying Assumption 1 whose columns have been arbitrarily permuted, our aim is to automatically identify the columns of M corresponding to the columns of W.

1.2 Successive Projection Algorithm (SPA)

The successive projection algorithm (SPA; see Algorithm 1) is a simple but fast and robust recursive algorithm for solving near-separable NMF. It was first introduced in [2], and later proved to be robust in [15].

Theorem 1 ([15], Th. 3). *Let $\tilde{M}$ satisfy Assumption 1. If $\epsilon \leq \mathcal{O}\left(\frac{\sigma_{\min}(W)}{\sqrt{r}\kappa^2(W)}\right)$, then SPA identifies all the columns of W up to error $\mathcal{O}\left(\epsilon\,\kappa^2(W)\right)$, that is, the index set $\mathcal{K}$ identified by SPA satisfies*

$$\max_{1 \leq j \leq r} \min_{k \in \mathcal{K}} ||W(:,j) - M(:,k)||_2 \leq \mathcal{O}\left(\epsilon\,\kappa^2(W)\right),$$

where $\kappa(W) = \frac{\sigma_{\max}(W)}{\sigma_{\min}(W)}$ is the condition number of W.

Note that the dependence in terms of the condition number $\kappa(W)$ of W is quite strong: for ill-conditioned matrices, ϵ has to be very close to zero to guarantee recovery. Note however that, in practice, SPA often performs well [15, 3] (in fact, the above robustness result considers a worst-case scenario).

Remark 1. *In Theorem 1, $\kappa(W)$ can actually be replaced with $\beta(W) = \frac{\max_i ||W(:,i)||_2}{\sigma_{\min}(W)}$; see [15]. We choose to work here with $\kappa(W)$ instead of $\beta(W)$ because it is more convenient and makes the presentation nicer. Note that $\beta(W) \leq \kappa(W) \leq \sqrt{r}\beta(W)$ since $\max_i ||W(:,i)||_2 \leq \sigma_{\max}(W) \leq \sqrt{r}\max_{1 \leq k \leq r} ||W(:,k)||_2$ so that $\beta(W)$ and $\kappa(W)$ only differ by a factor of at most $\sqrt{r}$ which is usually negligible (in practice, r is in general smaller than 100).*

In [3], Arora et al. proposed FastAnchorWords, an algorithm closely related to SPA: at each step, instead of picking the column whose projection onto the orthogonal complement of the columns extracted so far has maximum norm, they pick the column whose projection onto the affine hull of the columns extracted so far has maximum norm. This variant requires the entries of each column of H

Algorithm 1 Successive Projection Algorithm (SPA) [2, 15]

Require: Near-separable matrix $M = WH + N \in \mathbb{R}_+^{m \times n}$ (see Assumption 1), the number r of columns to be extracted.

Ensure: Set of indices $\mathcal{K}$ such that $M(:, \mathcal{K}) \approx W$ (up to permutation).

1: Let $R = M$, $\mathcal{K} = \{\}$, $k = 1$.
2: **while** $R \neq 0$ and $k \leq r$ **do**
3:     $k^* = \mathrm{argmax}_k \|R_{:k}\|_2$.
4:     $u_k = R_{:k^*}$.
5:     $R \leftarrow \left(I - \frac{u_k u_k^T}{\|u_k\|_2^2} \right) R$.
6:     $\mathcal{K} = \mathcal{K} \cup \{k^*\}$.
7:     $k = k + 1$.
8: **end while**

---

to sum to one so that it is less general than SPA[1]. However, their analysis extends to SPA[2], and they proposed a post-processing to make it more robust: Let $\mathcal{K}$ be the set of indices extracted by SPA, and denote $\mathcal{K}(k)$ the index extracted at step k. For $k = 1, 2, \ldots r$, the post-processing

- Projects each column of the data matrix onto the orthogonal complement of $M(:, \mathcal{K} \backslash \{\mathcal{K}(k)\})$.

- Identifies the column with maximum norm of the corresponding projected matrix (say the k'th),

- Updates $\mathcal{K} \leftarrow \mathcal{K} \backslash \{\mathcal{K}(k)\} \cup \{k'\}$.

See Algorithm 4 in [3] for more details.

Theorem 2 ([3], Th. 4.3). *Let $\tilde{M}$ satisfy Assumption 1. If $\epsilon \leq \mathcal{O}\left(\frac{\sigma_{\min}(W)}{r \kappa^2(W)} \right)$, then post-processed SPA identifies all the columns of W up to error $\mathcal{O}\left(\epsilon \, \kappa(W) \right)$.*

Theorem 2 improves upon the error bound[3] of Theorem 1 (to a factor $\kappa(W)$), however it still requires ϵ to be rather small to guarantee any error bound (especially for ill-conditioned matrices). Moreover, it was observed in practice that SPA and its post-processed version often give similar results [11] (see also Section 5 for some results on synthetic data sets).

Remark 2. *In [3], the bounds are only in terms of $\gamma(W)$, which is the minimum distance between a vertex and the affine hull of the other vertices. Because we need to add the origin in the data set to extend their analysis to SPA (see footnote 2), we have to replace $\gamma(W)$ with $\gamma([W, 0]) = \sigma_{\min}(W)$. Moreover the input matrix is assumed to be normalized in [3] (in particular, $\|W(:, j)\|_1 = 1$ for all j) hence $\mathcal{O}\left(\frac{\kappa(W)}{\sqrt{r}} \right) \leq \gamma^{-1}([W, 0]) \leq \mathcal{O}\left(\kappa(W) \right)$. Therefore, as for Theorem 1 (see Remark 1), we have slightly weakened the bound of Theorem 2 (of at most a factor of $\sqrt{r}$) to work with $\kappa(W)$.*

---

[1]For example, in hyperspectral imaging, the input matrix usually satisfies the assumption that the entries of each column of H sum to at most one (Assumption 1), but not to one (for example because of different illumination conditions in the image). Therefore, in this case, FastAnchorWords requires normalization of the input matrix (while SPA does not) and will be rather sensitive to columns of M with small norms (e.g., background pixels); see the discussion in [15].

[2] In fact, adding the origin in the data set and applying FastAnchorWords by imposing that the origin (which is a vertex) is extracted first makes FastAnchorWords and SPA equivalent.

[3]Note however that the bound on the noise level is slightly weaker than in Theorem 1 being proportional to r^{-1} instead of $r^{-1/2}$ (although we believe the bound could be improved, and r is small in practice).

1.3 Contribution and Outline of the Paper

In this paper, we present a preconditioning based on semidefinite programming reducing the condition number of the input near-separable matrix. In fact, for sufficiently small noise levels, the preconditioning makes the condition number smaller than some constant (Theorem 5). This in turn allows us to solve near-separable NMF problems in a more robust way since the error bounds provided by near-separable NMF algorithms depend on the conditioning of the input matrix. Hence, the preconditioning can be combined with any existing (robust) near-separable NMF algorithm to make it more robust.

In this paper, we focus on SPA, because it is rather simple and very effective in practice, and prove the following (see Theorem 6 for more details).

Theorem 3. *Let $\tilde{M}$ satisfy Assumption 1 with $m = r$. If $\epsilon \leq \mathcal{O}\left(\frac{\sigma_{\min}(W)}{r\sqrt{r}}\right)$, preconditioned SPA identifies all the columns of W up to error $\mathcal{O}\left(\epsilon \kappa(W)\right)$.*

Observe that, up to some factor in r, the upper bound on ϵ only depends on $\sigma_{\min}(W)$, and we cannot expect better; see Section 1.1. Hence preconditioned SPA can tolerate much higher noise levels than SPA and post-processed SPA.

The paper is organized as follows. In Section 2, we propose and analyze a way to precondition near-separable matrices using semidefinite programming (SDP), where we focus on the case $m = r$. This allows us to prove that the preconditioning makes SPA significantly more robust to noise (see Theorem 3). In Section 3, the preconditioning is combined with a linear dimensionality reduction technique to handle the case $m > r$. In Section 4, an active-set method is proposed to deal with large-scale problems (that is, for n large). In Section 5, we show on some synthetic data sets that combining the preconditioning with SPA leads to a significantly more robust algorithm. We also show that it can be applied to large-scale real-world hyperspectral images.

Remark 3. *While writing this paper, we noticed the very recent paper [26] where the same idea is used to perform near-separable NMF. However, the proposed algorithm is different: in fact, several SDP's might have to be solved while the SDP is not used for preconditioning but as a preprocessing to select a subset of the columns of M (see [26, Algorithm 2]). Because of that, the robustness result only holds with an additional condition on the input separable matrix [26, Theorem 2]: the author requires that there are no duplicates nor near duplicates of the columns of W in the data set which is a rather strong and not very reasonable assumption in practice; see the discussion in [13].*

Therefore, the main contributions of this paper are (i) the use of SDP for preconditioning near-separable matrices, and (ii) a general robustness analysis for preconditioned SPA.

1.4 Notation

The set $\mathbb{S}_+^r$ denotes the set of r-by-r symmetric positive-semidefinite matrices. The eigenvalue of $A \in \mathbb{S}_+^r$ will be denoted

$$\lambda_{\max}(A) = \lambda_1(A) \geq \lambda_2(A) \geq \cdots \geq \lambda_r(A) = \lambda_{\min}(A) \geq 0.$$

The singular values of a matrix $B \in \mathbb{R}^{r \times n}$ where $r \leq n$ are denoted

$$\sigma_{\max}(B) = \sigma_1(B) \geq \sigma_2(B) \geq \cdots \geq \sigma_r(B) = \sigma_{\min}(B) \geq 0.$$

2 Preconditioning

If a near-separable matrix $\tilde{M}$ satisfying Assumption 1 is pre-multiplied by a full rank matrix Q such that QW remains full rank, then the new matrix $Q\tilde{M}$ also satisfies Assumption 1 where the matrix W is replaced with QW, while the noise is replaced with QN. Therefore, if one can find a matrix Q such that QW is better conditioned than W (and QN is not too large), near-separable NMF algorithms should be more robust against noise. For simplicity, we focus our analysis on the case $m = r$ which implies, under Assumption 1, that W is a full rank r-by-r matrix. We show in Section 3 how to use the preconditioning when $m > r$ using linear dimensionality reduction.

A first straightforward and useful observation is that taking $Q = RW^{-1}$ where R is any r-by-r orthonormal matrix[4] (that is, $R^T R = I_r$) gives a matrix QW which is perfectly conditioned, that is, $\kappa(QW) = 1$. Therefore, the preconditioning Q should be as close as possible to a matrix of the form RW^{-1}. The main goal of this section is to show how to provably compute an approximation of $Q = RW^{-1}$ based on $\tilde{M}$ using semidefinite programming.

2.1 Motivation: Preconditioning SPA

It is rather straightforward to analyze the influence of a preconditioning on SPA.

Corollary 1. *Let $\tilde{M}$ satisfy Assumption 1 with $m = r$ and let $Q \in \mathbb{R}^{r \times r}$. If QW is full rank, and*

$$\epsilon \leq \mathcal{O}\left(\frac{\sigma_{\min}(QW)}{\sqrt{r}\sigma_{\max}(Q)\kappa^2(QW)}\right),$$

then SPA applied on matrix $Q\tilde{M}$ identifies indices corresponding to the columns of W up to error $\mathcal{O}\left(\epsilon\,\kappa(Q)\,\kappa(QW)^2\right)$.

Proof. Let us denote ϵ' the smallest value such that $||QN(:,i)||_2 \leq \epsilon'$ for all i, $\sigma' = \sigma_{\min}(QW)$ and $\kappa' = \kappa(QW)$. By Theorem 1, if $\epsilon' \leq \mathcal{O}\left(\frac{\sigma'}{\sqrt{r}\kappa'^2}\right)$, SPA applied on matrix $Q\tilde{M}$ identifies the columns of matrix QW up to error $\mathcal{O}\left(\epsilon'\kappa'^2\right)$, that is, it identifies r columns of $Q\tilde{M}$ such that for all $1 \leq k \leq r$ there exists i in the extracted set of indices such that

$$\sigma_{\min}(Q)||\tilde{M}(:,i) - W(:,k)||_2 \leq ||Q\tilde{M}(:,i) - QW(:,k)||_2 \leq \mathcal{O}\left(\epsilon'\kappa'^2\right).$$

This implies that the indices extracted by SPA allows us to identify the columns of W up to error $\mathcal{O}\left(\frac{\epsilon'\kappa'^2}{\sigma_{\min}(Q)}\right)$. Moreover, we have that

$$||QN(:,i)||_2 \leq \sigma_{\max}(Q)\epsilon,$$

hence $\epsilon' \leq \sigma_{\max}(Q)\epsilon$ so that $\epsilon \leq \mathcal{O}\left(\frac{\sigma'}{\sqrt{r}\kappa'^2\sigma_{\max}(Q)}\right) \Rightarrow \epsilon' \leq \mathcal{O}\left(\frac{\sigma'}{\sqrt{r}\kappa'^2}\right)$, while $\mathcal{O}\left(\frac{\epsilon'\kappa'^2}{\sigma_{\min}(Q)}\right) \leq \mathcal{O}\left(\epsilon\kappa'^2\kappa(Q)\right)$. $\qquad\square$

In particular, using $Q = W^{-1}$ (or any orthonormal transformation) gives the following result.

Corollary 2. *Let $\tilde{M}$ satisfy Assumption 1 with $m = r$. If $\epsilon \leq \mathcal{O}\left(\frac{\sigma_{\min}(W)}{\sqrt{r}}\right)$, then SPA applied on the matrix $W^{-1}\tilde{M}$ identifies indices corresponding to the columns of W up to error $\mathcal{O}\left(\epsilon\kappa(W)\right)$.*

---

[4]Actually, we only need $\kappa(QW) = 1$ so Q can be a scaling of RW^{-1}.

Proof. Taking $Q = W^{-1}$ in Corollary 1 gives $\sigma' = \kappa' = 1$ while $\sigma_{\min}(Q) = \sigma_{\max}(W)^{-1}$, $\sigma_{\max}(Q) = \sigma_{\min}(W)^{-1}$, and $\kappa(Q) = \kappa(W)$. $\qquad\qquad\square$

Of course, the matrix W is unknown, otherwise the near-separable NMF problem would be solved. In this section, we will show how to approximately compute W^{-1} from $\tilde{M}$ using semidefinite programming.

Remark 4 (Combining preconditioning with post-processing). *Note that combining the preconditioning with post-processed SPA (see Theorem 2) would not improve the error bound significantly (only up to a factor $\kappa(QW)$ which will be shown to be constant for our preconditioning); see Section 5 for some numerical experiments.*

2.2 Minimum Volume Ellipsoid and SDP Formulation for Approximating W^{-1}

We have seen in the previous section that, if we were given RW^{-1} where R is any orthonormal matrix, we could significantly improve the error bounds for SPA, leading to a significantly more robust near-separable NMF algorithm. Equivalently, we need to find a matrix Q such that $\kappa(QW)$ is close to one. Let $A = Q^T Q \succ 0$. We have $W^T A W = (QW)^T (QW)$ so that $\sigma_i(W^T A W) = \sigma_i^2(QW)$ for all i, hence it is also equivalent to find a matrix A such that $\kappa(W^T A W)$ is close to one since it will imply that $\kappa(QW)$ is close to one, while we can compute a factorization of $A = Q^T Q$ (e.g., a Cholesky decomposition). Ideally, we would like that $A = (WW^T)^{-1} = W^{-T}W^{-1}$, that is, $Q = RW^{-1}$ for some orthonormal transformation R.

The central step of our algorithm is to compute the minimum volume ellipsoid centered at the origin containing all columns of $\tilde{M}$. An ellipsoid $\mathcal{E}$ centered at the origin in $\mathbb{R}^r$ is described via a positive definite matrix $A \in \mathbb{S}_{++}^r$:

$$\mathcal{E} = \{\ x \in \mathbb{R}^r \mid x^T A x \leq 1\ \}.$$

The axes of the ellipsoid are given by the eigenvectors of matrix A, while their length is equal to the inverse of the square root of the corresponding eigenvalue. The volume of $\mathcal{E}$ is equal to $\det(A)^{-1/2}$ times the volume of the unit ball in dimension r. Therefore, given a matrix $\tilde{M} \in \mathbb{R}^{r \times n}$ of rank r, we can formulate the minimum volume ellipsoid centered at the origin and containing the columns $\tilde{m}_i$ $1 \leq i \leq n$ of matrix $\tilde{M}$ as follows

$$\min_{A \in \mathbb{S}_+^r} \quad \log \det(A)^{-1}$$
$$\text{such that} \quad \tilde{m}_i{}^T A \tilde{m}_i \leq 1 \quad \text{for } i = 1, 2, \ldots, n. \tag{1}$$

This problem is SDP representable [8, p.222] (see also Remark 8). Note that if $\tilde{M}$ is not full rank, that is, the convex hull of the columns of $\tilde{M}$ and the origin is not full dimensional, then the objective function value is unbounded below. Otherwise, the optimal solution of the problem exists and is unique [18].

Theorem 4. *For a separable matrix $\tilde{M}$ satisfying Assumption 1 with $m = r$ and in the noiseless case (that is, $N = 0$ and $M = \tilde{M}$), the optimal solution of (1) is given by $A^* = (WW^T)^{-1}$.*

Proof. The matrix $A^* = (WW^T)^{-1}$ is a feasible solution of the primal (1): In fact, for all i,

$$m_i{}^T A m_i = h_i{}^T W^T W^{-T} W^{-1} W h_i = ||h_i||_2^2 \leq ||h_i||_1^2 \leq 1.$$

7

The dual of (1) is given by [8, p.222]

$$\max_{y \in \mathbb{R}^n} \ \log \det \left(\sum_{i=1}^{n} y_i \tilde{m}_i \tilde{m}_i^T \right) - e^T y + r \quad \text{such that} \ \ y \geq 0. \tag{2}$$

One can check that $y^* = [e_r; 0]$ is a feasible solution of the dual (with $\sum_{i=1}^{n} y_i \tilde{m}_i \tilde{m}_i^T = \sum_{i=1}^{r} w_i w_i^T = WW^T$ and $e^T y = r$) whose objective function value coincides with the one of the primal solution $A^* = (WW^T)^{-1}$ which is therefore optimal by duality. $\qquad \square$

Theorem 4 shows that, in the noiseless case, the optimal solution of (1) provides us with an optimal preconditioning for the separable NMF problem.

We now show that for sufficiently small noise, the optimal solution A^* of (1) still provides us with a good preconditioning for $\tilde{M}$, that is, $\kappa(W^T A^* W) \approx 1$. Intuitively, the noise N perturbs 'continuously' the feasible domain of (1), hence the optimal solution is only slightly modified for small enough perturbations N; see, e.g., [30]. In the following, we quantify this statement precisely.

Let us perform a change of variable on the SDP (1) using $A = W^{-T} C W^{-1}$ to obtain the following equivalent problem

$$C^* = \text{argmin}_{C \in \mathbb{S}_+^r} \ \log \det(C)^{-1} + \log \det(WW^T)$$

$$\text{such that} \quad \tilde{m}_i^T \left(W^{-T} C W^{-1} \right) \tilde{m}_i \leq 1 \quad \text{for } i = 1, 2, \ldots, n. \tag{3}$$

We have that $A^* = W^{-T} C^* W^{-1}$ where A^* is the optimal solution of (1). Since our goal is to bound $\kappa(W^T A^* W)$ and $W^T A^* W = C^*$, it is equivalent to show that C^* is well-conditioned, that is, $\kappa(C^*) \approx 1$.

2.3 Upper Bounding $\kappa(C^*)$

It is interesting to notice that the case $r = 1$ is trivial since C^* is a scalar hence $\kappa(C^*) = 1$. In fact, the near-separable NMF problem is trivial when $r = 1$ since all columns of $\tilde{M}$ are multiple of the column of $W \in \mathbb{R}^{m \times 1}$.

Otherwise, since $\kappa(C^*) \geq 1$, we only need to provide an upper bound for $\kappa(C^*)$. Let us first provide a lower bound on $\det(C^*)$ which will be useful later.

Lemma 1. *If $\tilde{M}$ satisfies Assumption 1 with $m = r$, then the optimal solution C^* of (3) satisfies*

$$\det(C^*) \geq \left(1 + \frac{\epsilon}{\sigma_{\min}(W)} \right)^{-2r}. \tag{4}$$

Proof. Let us consider the matrix

$$A = \alpha(WW^T)^{-1}, \text{ where } 0 < \alpha \leq 1.$$

Let us denote $\sigma = \sigma_{\min}(W)$. Note that $\sigma_{\max}((WW^T)^{-1}) = \sigma_{\max}^2(W^{-1}) = \frac{1}{\sigma^2}$. We have

$$h_i^T W^T A W h_i + 2 n_i^T A W h_i + n_i^T A n_i \leq \alpha \left(1 + 2\epsilon \|W^{-T} h_i\|_2 + \frac{\epsilon^2}{\sigma^2} \right)$$

$$\leq \alpha \left(1 + 2\frac{\epsilon}{\sigma} + \frac{\epsilon^2}{\sigma^2} \right) = \alpha \left(1 + \frac{\epsilon}{\sigma} \right)^2,$$

8

since $\tilde{m}_i = W h_i + n_i$, $||n_i||_2 \leq \epsilon$ and $||h_i||_2 \leq ||h_i||_1 \leq 1$ for all i. Therefore, taking

$$\alpha = \left(1 + \frac{\epsilon}{\sigma}\right)^{-2}$$

makes A a feasible solution of (1). From the change of variable $A = W^{-T} C W^{-1}$, we have that

$$C = W^T A W = \left(1 + \frac{\epsilon}{\sigma}\right)^{-2} I_r$$

is a feasible solution of (3) hence $\det(C^*) \geq \left(1 + \frac{\epsilon}{\sigma}\right)^{-2r}$. $\qquad\square$

We can now provide a first upper bound on $\lambda_{\max}(C^*)$.

Lemma 2. *If $\tilde{M}$ satisfies Assumption 1 with $m = r$ then*

$$\left(1 - \frac{\sqrt{r}\epsilon}{\sigma_{\min}(W)}\right) \sqrt{\lambda} \leq \sqrt{r},$$

where $\lambda = \lambda_{\max}(C)$ and C is any feasible solution of (3).

Proof. Let us denote $N' = W^{-1} N$ so that $||N'(:,i)||_2 \leq \sigma_1(W^{-1}) ||N(:,i)||_2 \leq \frac{\epsilon}{\sigma}$ where $\sigma = \sigma_{\min}(W)$. Any feasible solution C of (3) must satisfy for all i

$$(W h_i + n_i)^T W^{-T} C W^{-1} (W h_i + n_i) = (h_i + n_i')^T C (h_i + n_i') \leq 1.$$

In particular, for all $1 \leq k \leq r$,

$$(e_k + n_k')^T C (e_k + n_k') \leq 1, \tag{5}$$

where e_k's are the columns of the identity matrix (since $m_k = W e_k + n_k = w_k + n_k$ for $1 \leq k \leq r$ by Assumption 1). Letting $C = B^T B$ so that $\sigma_{\max}(B) = \sqrt{\lambda}$, we have for all $1 \leq k \leq r$

$$||B(:,k)||_2 - \sigma_{\max}(B)\frac{\epsilon}{\sigma} \leq ||B(e_k + n_k')||_2 \leq 1 \quad \Rightarrow \quad ||B(:,k)||_2 = ||Be_k||_2 \leq 1 + \sigma_{\max}(B)\frac{\epsilon}{\sigma}. \tag{6}$$

Therefore,

$$\sqrt{\lambda} = \sigma_{\max}(B) \leq \sqrt{r} \max_k ||B(:,k)||_2 \leq \sqrt{r}\left(1 + \sigma_{\max}(B)\frac{\epsilon}{\sigma}\right) = \sqrt{r}\left(1 + \sqrt{\lambda}\frac{\epsilon}{\sigma}\right),$$

which gives the result. $\qquad\square$

We now derive an upper bound on $\lambda_{\max}(C^*)$ independent of r.

Lemma 3. *If $\tilde{M}$ satisfies Assumption 1 with $m = r \geq 2$ and*

$$\epsilon \leq \frac{\sigma_{\min}(W)}{8r\sqrt{r}},$$

then $\lambda = \lambda_{\max}(C^*) \leq 4$ where C^* is the optimal solution of (3).

Proof. Let us denote $\epsilon' = \frac{\epsilon}{\sigma_{\min}(W)}$ and let $C^* = B^T B$ for some B. By Lemma 2, we have

$$\sqrt{\lambda} \le \frac{\sqrt{r}}{1 - \sqrt{r}\epsilon'} \le 2\sqrt{r}$$

since $\epsilon' \le \frac{1}{8r\sqrt{r}} \le \frac{1}{2\sqrt{r}}$. Using Equation (6), we obtain

$$\sum_{k=1}^r \lambda_k(C^*) = \sum_{k=1}^r \sigma_k^2(B) = \|B\|_F^2 = \sum_{k=1}^r \|B(:,k)\|_2^2 \le r\left(1 + \sqrt{\lambda}\,\epsilon'\right)^2 \le r\left(1 + 2\sqrt{r}\,\epsilon'\right)^2 \le r\left(1 + \frac{1}{r}\right),$$

where the last inequality follows from $\epsilon' \le \frac{1}{8r\sqrt{r}}$: in fact,

$$\left(1 + 2\sqrt{r}\,\epsilon'\right)^2 \le \left(1 + \frac{1}{4r}\right)^2 = 1 + \frac{1}{2r} + \frac{1}{16r^2} \le 1 + \frac{1}{r}.$$

Since the maximum x^* of the problem

$$\max_{x \in \mathbb{R}_+^r} \prod_{k=1}^r x_k \quad \text{such that} \quad \sum_{k=1}^r x_k = \beta \ge 0,$$

is unique and given by $x_k^* = \frac{\beta}{r}$ for all k, we have that the optimal solution C^* of (3) satifies

$$\left(1 + \frac{1}{8r}\right)^{-2r} \le \left(1 + \epsilon'\right)^{-2r} \le \det(C^*) = \prod_{k=1}^r \lambda_k(C^*) \le \lambda \left(\frac{r\left(1 + \frac{1}{r}\right) - \lambda}{r - 1}\right)^{r-1}, \tag{7}$$

where the left-hand side inequality follows from Lemma 1 and $\epsilon' \le \frac{1}{8r}$. Equivalently,

$$(r - 1)\left(1 + \frac{1}{8r}\right)^{-\frac{2r}{r-1}} \le \lambda^{\frac{1}{r-1}}\left(r + 1 - \lambda\right).$$

In Appendix A, we prove that this inequality implies $\lambda \le 4$ for any integer $r \ge 2$. $\qquad\square$

We have now an upper bound on $\lambda_{\max}(C^*)$. In order to bound $\kappa(C^*)$, it is therefore sufficient to find a lower bound for $\lambda_{\min}(C^*)$.

Lemma 4. *If $\tilde{M}$ satisfies Assumption 1 with $m = r \ge 2$ and*

$$\epsilon \le \frac{\sigma_{\min}(W)}{8r\sqrt{r}},$$

then $\lambda_{\min}(C^*) \ge \frac{1}{10}$ where C^* is the optimal solution of (3).

Proof. Using the same trick as for obtaining Equation (7) and denoting $\delta = \lambda_{\min}(C^*)$, we have

$$e^{-1/4} \le \left(1 + \frac{1}{8r}\right)^{-2r} \le \det(C^*) \le \delta \left(\frac{r\left(1 + \frac{1}{r}\right) - \delta}{r - 1}\right)^{r-1} \le \delta \left(1 + \frac{2}{r - 1}\right)^{r-1} \le e^2 \delta,$$

which implies $\delta \ge \frac{e^{-1/4}}{e^2} \ge \frac{1}{10}$. The first inequality follows from the nonincreasingness of $\left(1 + \frac{1}{8r}\right)^{-2r}$ and the limit to infinity being $e^{-1/4}$. The last inequality follows from the nondecreasingness of $\left(1 + \frac{2}{r-1}\right)^{r-1}$ and the limit to infinity being e^2. $\qquad\square$

Remark 5. *The lower bound for $\lambda_{\min}(C^*)$ can be improved to $\frac{1}{4}$ showing that*

$$0.84 \le \left(1 + \frac{1}{8r\sqrt{r}}\right)^{-2r} \le \det(C^*) \le \delta\left(\frac{r}{r-1}\left(1 + 2\frac{\epsilon}{\sigma}\right)^2\right)^{r-1} \le \delta\left(\frac{r}{r-1}\left(1 + \frac{1}{4r\sqrt{r}}\right)^2\right)^{r-1} \le 3\delta$$

for any $r \ge 2$. The third inequality follows from $\sum_{k=1}^{r} \lambda_k(C^*) \le r\left(1 + \sqrt{\lambda_{\max}(C^*)}\frac{\epsilon}{\sigma}\right)^2 \le r(1 + 2\frac{\epsilon}{\sigma})^2$.

For sufficiently small noise level, we can now provide an upper bound for the condition number of C^*.

Theorem 5. *If $\tilde{M}$ satisfies Assumption 1 with $m = r$ and*

$$\epsilon \le \frac{\sigma_{\min}(W)}{8r\sqrt{r}},$$

then $\kappa(C^*) \le 40$ where C^* is the optimal solution of (3).

Proof. This follows from Lemmas 3 and 4. $\qquad\square$

2.4 Robustness of Preconditioned SPA

The upper bound on $\kappa(C^*)$ (Theorem 5) proves that the preconditioning generates a well-conditioned near-separable matrix. Hence the preconditioning allows to obtain more robust near-separable NMF algorithms. In particular, this allows us to prove that preconditioned SPA is significantly more robust to noise than SPA.

Theorem 6. *Let M satisfy Assumption 1 with $m = r$ and let $Q \in \mathbb{R}^{r \times r}$ be such that $A^* = Q^T Q$ where A^* is the optimal solution of (1). If $\epsilon \le \mathcal{O}\left(\frac{\sigma_{\min}(W)}{r\sqrt{r}}\right)$, SPA applied on matrix QM identifies indices corresponding to the columns of W up to error $\mathcal{O}\left(\epsilon\kappa(W)\right)$.*

Proof. This follows from Corollary 1, Lemmas 3 and 4, and Theorem 5. In fact, let Q be such that $A^* = Q^T Q$ where A^* is the optimal solution of (1). Let also $C^* = W^T A^* W$ be the optimal solution of (3). We have that

$$(QW)^T QW = W^T Q^T QW = W^T A^* W = C^*,$$

hence $\sigma_i(QW) = \sqrt{\lambda_i(C^*)}$ for all i implying

$$\sigma' = \sigma_{\min}(QW) = \sqrt{\lambda_{\min}(C^*)} \ge \frac{1}{\sqrt{10}}, \qquad (Lemma\ 4)$$

$$\kappa' = \kappa(QW) = \sqrt{\kappa(C^*)} \le \sqrt{40}, \qquad (Theorem\ 5)$$

$$\sigma_{\min}(Q) = \sigma_{\min}(QWW^{-1}) \ge \sigma_{\min}(QW)\sigma_{\min}(W^{-1}) \ge \frac{1}{\sqrt{10}\sigma_{\max}(W)},$$

$$\sigma_{\max}(Q) = \sigma_{\max}(QWW^{-1}) \le \sigma_{\max}(QW)\sigma_{\max}(W^{-1}) \le \frac{2}{\sigma_{\min}(W)}, \qquad (Lemma\ 3)$$

hence

$$\kappa(Q) \le 2\sqrt{10}\kappa(W).$$

$\qquad\square$

3 Linear Dimensionality Reduction

In practice, the input matrix $M \in \mathbb{R}^{m \times n}$ of rank r usually has more than r rows (that is, $m > r$) and the matrix M is rank deficient, or, equivalently, the columns of matrix the M are contained in a polytope which is not full dimensional. A simple and natural way to handle this situation is to use a linear dimensionality reduction technique as a pre-processing. This is a standard trick in signal processing and in particular in hyperspectral unmixing; see, e.g., [25] and the references therein.

For example, on can use the truncated singular value decomposition (SVD). Given $\tilde{M} \in \mathbb{R}^{m \times n}$, an optimal rank-$r$ approximation of $\tilde{M}$ with respect to the Frobenius norm (and the 2-norm) is given by

$$\tilde{M}_r = U \underbrace{\begin{pmatrix} \sigma_1(\tilde{M}) & & & \\ & \sigma_2(\tilde{M}) & & \\ & & \ddots & \\ & & & \sigma_r(\tilde{M}) \end{pmatrix}}_{\Sigma} V,$$

where $U \in \mathbb{R}^{m \times r}$ (resp. $V \in \mathbb{R}^{r \times n}$) has orthonormal columns (resp. rows) which are the first r left (resp. right) singular vectors of $\tilde{M}$. Given the truncated SVD, we can therefore use as the input near-separable matrix the matrix $\Sigma V \in \mathbb{R}^{r \times n}$. This linear dimensionality reduction amounts to replace the data points with their projections onto the linear space that minimizes the sum of the squared residuals.

Remark 6 (Huge Scale Problems). *If it is too expensive to compute the SVD, it is possible to use cheaper linear dimensionality reduction techniques such as random projections [19, 28], or picking randomly a subset of the rows and then performing the SVD on the subset; see [6] and the references therein.*

Remark 7 (Other Noise Models). *Using the SVD for dimensionality reduction is ideal when the noise is Gaussian. When this is not the case (in particular if we have some information about the noise model), it would be beneficial to use other dimensionality reduction techniques.*

Heuristic Preconditioning with SVD

It is interesting to observe that

1. The pre-processing using the SVD described above is equivalent to pre-multiplying $\tilde{M}$ with U^T because $U^T \tilde{M} = \Sigma V$.

2. Given the SVD (U_W, Σ_W, V_W) of W, an optimal preconditioning is given by $Q = \Sigma_W^{-1} U_W^T$: in fact, $\kappa(QW) = \kappa(\Sigma_W^{-1} U_W^T W) = \kappa(V_W) = 1$.

3. The SVD of M and W are closely related (in fact, they share the same column space). Intuitively, the best fitting linear subspace (w.r.t. to the Frobenius norm) for the columns of $\tilde{M}$ should be close to the column space of W.

Therefore, if one wants to avoid solving the SDP (1), a heuristic to approximately estimate $\Sigma_W^{-1} U_W^T$ is to use the SVD of $\tilde{M}$, that is, given the truncated SVD (U, Σ, V) of $\tilde{M}$, use the preconditioning $\Sigma^{-1} U^T$. We have observed that U_W and U are usually close to one another (up to orthogonal transformations), while Σ_W^{-1} and Σ are if the data points are well distributed in conv(W); see some numerical experiments in Section 5. However, in the next section, we show how to solve the SDP (1) for large n which makes this heuristic less attractive (unless r is large).

4 Active-Set Algorithm for Large-Scale Problems

John [18] proved that there exists a subset of the columns of $\tilde{M}$, say $\mathcal{S}$, with $r \leq |\mathcal{S}| \leq \frac{r(r+1)}{2}$ so that the problem (1) is equivalent to

$$\min_{A \in \mathbb{S}_+^r} \quad \log \det(A)^{-1}$$
$$\text{such that} \qquad \tilde{m}_i^T A \tilde{m}_i \leq 1 \ \text{ for } \ i \in \mathcal{S}. \tag{8}$$

For example, in the noiseless separable case, we have seen that there exists such a set with $|\mathcal{S}| = r$; see Theorem 4. (John [18] also showed that if $\tilde{m}_i$ belongs to the convex hull of the other columns of $\tilde{M}$, then $i \notin \mathcal{S}$.)

Although the set $\mathcal{S}$ is unknown in advance, we can try to make a good guess for $\mathcal{S}$, solve (8), and then check whether all remaining columns of $\tilde{M}$ satisfy $\tilde{m}_i^T A \tilde{m}_i \leq 1$. If not, we can add some of them to the current set $\mathcal{S}$, remove columns with $\tilde{m}_i^T A \tilde{m}_i < 1$, and resolve (8). A similar active-set approach is described in [31]. We propose to use SPA in order to find a good initial guess for $\mathcal{S}$; see Algorithm 3. Since SPA cannot extract more than $\min(m, n)$ indices of an m-by-n matrix (because the residual has to be equal to zero after $\min(m, n)$ steps), we reinitialize SPA as many times as necessary, setting the columns corresponding to the indices already extracted to zero.

Algorithm 2 Preconditioned SPA for Near-Separable NMF using SVD and an Active-Set Method

---

Require: Near-separable matrix $\tilde{M} = WH + N \in \mathbb{R}_+^{m \times n}$ (see Assumption 1), number of columns to be extracted r, number of active constraints $\frac{r(r+1)}{2} < \eta \leq n$, precision $\delta \geq 0$.

    % (1) Linear Dimensionality Reduction using SVD
1: Compute the truncated SVD $[U, \Sigma, V] \in \mathbb{R}^{m \times r} \times \mathbb{R}^{r \times r} \times \mathbb{R}^{n \times r}$ of $\tilde{M} \approx U \Sigma V^T$.
2: Replace $\tilde{M} \leftarrow U^T \tilde{M} = \Sigma V^T$;
    % (2) Solve the SDP in the reduced space using an active-set method
3: $\mathcal{S} = \text{Algorithm } 3(\tilde{M}, \eta)$.   *% Initialization using SPA*
4: Compute the optimal solution A of (1) for matrix $\tilde{M}(:, \mathcal{S})$.
5: **while** $\max_i(\tilde{m}_i^T A \tilde{m}_i) \geq 1 + \delta$ **do**
6:     $\mathcal{S} \leftarrow \mathcal{S} \setminus \{i \mid \tilde{m}_i^T A \tilde{m}_i < 1\}$.   *% Remove inactive constraints*
7:     *% Keep $\frac{r(r+1)}{2}$ 'good' active constraints using SPA and $\max_i(\tilde{m}_i^T A \tilde{m}_i)$*
8:     **if** $|\mathcal{S}| > \frac{r(r+1)}{2}$ **then**
9:         $\mathcal{S} = \text{indices extracted by SPA}(M(:, \mathcal{S}), r)$;
10:         Add $\frac{r(r+1)}{2} - r$ indices to $\mathcal{S}$ corresponding to the largest $\tilde{m}_i^T A \tilde{m}_i$, $i \in \mathcal{S}$.
11:     **end if**
12:     Add $(\nu - |\mathcal{S}|)$ indices to the set $\mathcal{S}$ corresponding to the largest $\tilde{m}_i^T A \tilde{m}_i$, $i \notin \mathcal{S}$.
13:     Compute the optimal solution A of (1) for matrix $\tilde{M}(:, \mathcal{S})$.
14: **end while**
    % (3) Extract r indices using SPA on the preconditioned matrix
15: Compute Q such that $A = Q^T Q$.
16: $\mathcal{K} = \text{SPA}(Q\tilde{M}, r)$ (see Algorithm 1).

Remark 8 (Solving the SDP with CVX). *In this work, we use CVX [17, 16] to solve the SDP (1). It turns out that the problem can be written directly in terms of the matrix Q (see below) hence we do not need to perform a factorization at step 15 of Algorithm 2:*

Algorithm 3 Selection of an Initial Active Set using SPA

Require: Matrix $M \in \mathbb{R}_+^{m \times n}$, number of indices to be extracted $K \leq n$.
Ensure: Set of indices $\mathcal{K} \subseteq \{1, 2, \ldots, n\}$ with $|\mathcal{K}| = K$.

1: $R = M$; $\mathcal{K} = \{\}$.
2: **while** $|\mathcal{K}| < K$ **do**
3:     $\mathcal{I} = \mathrm{SPA}(R, r)$.
4:     $R(:, \mathcal{I}) = 0$.
5:     $\mathcal{K} = \mathcal{K} \cup \mathcal{I}$.
6: **end while**

```
% Create and solve the minimum volumns ellipsoid centered at the origin
% and containing the columns of the r—by—n matrix Mtilde
cvx_begin quiet
    variable Q(r,r) symmetric
    maximize( det_rootn( Q ) )
    subject to
        norms( Q * Mtilde , 2 ) <= 1;
cvx_end
```

5   Numerical Experiments

In this section, we compare the following near-separable NMF algorithms on some synthetic data sets:

1. **SPA**: the successive projection algorithm; see Algorithm 1.

2. **Post-SPA**: SPA post-processed with the procedure proposed by Arora et al. [3]; see Section 1.2.

3. **Prec-SPA**: preconditioned SPA; see Algorithm 2 (we use $\delta = 10^{-6}$ and $\nu = \frac{r(r+1)}{2} + r$ for the remaining of the paper).

4. **Post-Prec-SPA**: Prec-SPA post-processed with the procedure proposed by Arora et al. [3]; see Section 1.2.

5. **Heur-SPA**: SPA preconditioned with the SVD-based heuristic; see Section 3.

6. **VCA**: vertex component analysis, a popular endmember extraction algorithm proposed in [27].

7. **XRAY**: recursive algorithm similar to SPA, but taking into account the nonnegativity constraints for the projection step [22]. (We use in this paper the variant referred to as max.)

We also show that Prec-SPA can be applied to large-scale real-world hyperspectral images.

The Matlab code is available at `https://sites.google.com/site/nicolasgillis/`. All tests are preformed using Matlab on a laptop Intel CORE i5-3210M CPU @2.5GHz 2.5GHz 6Go RAM.

5.1   Synthetic Data Sets

In this section, we compare the different algorithms on some synthetic data sets.

5.1.1 Middle Points

We take $m = r = 20$, and $n = 210$. The matrix W is generated using the `rand(.)` function of Matlab, that is, each entry is drawn uniformly at random in the interval $[0, 1]$. The matrix $H = [I_r, H']$ is such that H' has exactly two non-zero entries in each row equal to 0.5. Hence, the 190 data points are in the middle of two different columns of W. The noise is chosen such that the columns of W (that is, the first 20 columns of M) are not perturbed, while the 190 data points are moved towards the outside of the convex hull of the columns of W:

$$N = \epsilon\,[0_{20 \times 20}, M(:, 21) - \bar{w}, M(:, 22) - \bar{w}, \dots, M(:, n) - \bar{w}],$$

where $M = WH$ and $\bar{w} = \frac{1}{r} \sum_i w_i$ is the vertex centroid of the convex hull of the columns of W. These are the same near-separable matrices as in [15].

For each noise level (from 0 to 0.6 with step 0.01), we generate 100 such matrices, and Figure 1 reports the percentage of correctly extracted columns of W.
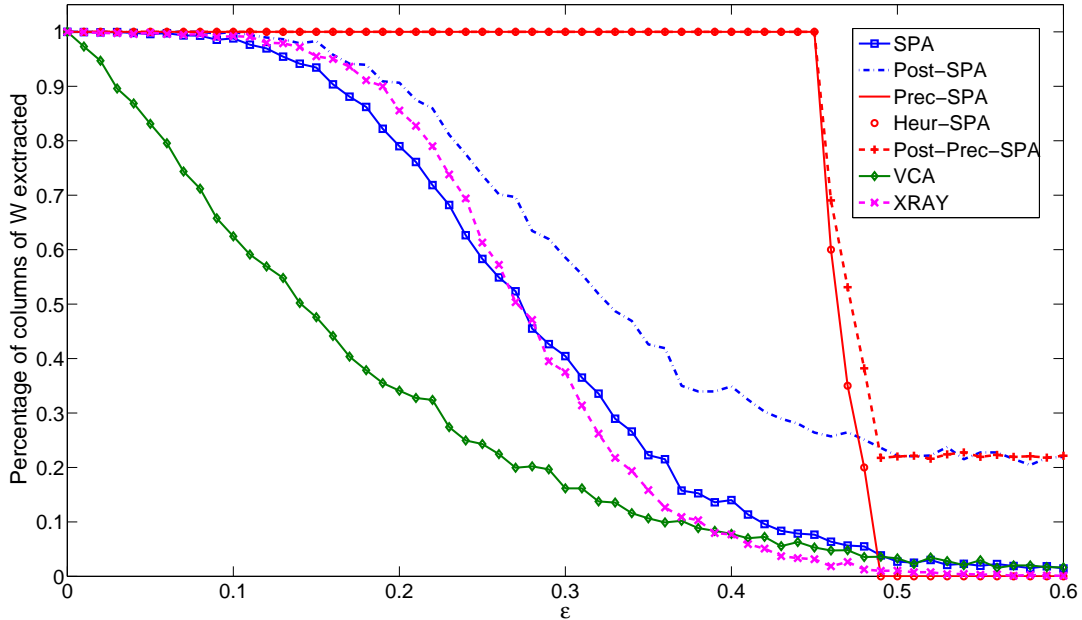

Figure 1: Comparison of the different near-separable NMF algorithms on the 'Middle Points' experiment.

We observe that

- As observed in [15], VCA is not robust to noise.

- As observed in [14], XRAY and SPA perform similarly.

- Heur-SPA and Prec-SPA perform exactly the same. As explained in Section 3, the reason is that the data points are rather well distributed and the SVD's of W and M are close to one another.

- Prec-SPA, Heur-SPA and Post-Prec-SPA are significantly more robust to noise than all other algorithms as they identify correctly all columns of W for the largest noise levels; see also Table 1. This confirms our theoretical findings that the preconditioning makes SPA significantly more robust to noise. This can be explained as follows: As long as the middle points remain inside the minimum volume ellipsoid containing the columns of W, the preconditioned algorithms perform perfectly (the solution of the SDP (1) actually remains unchanged). Moreover, when the middle points hit the boarder of the ellipsoid, there is a sharp change in the extracted columns.

- For large noise levels, Prec-SPA and Heur-SPA do not extract any column of W. The reason is the following: when the noise level becomes large enough, the columns of W will be contained in the convex hull of the other columns of M hence we should not expect an algorithm to identify columns of W for large noise levels. In some sense, Prec-SPA and Heur-SPA perform ideally in this particular situation.

- As predicted by Theorem 2, Post-SPA (resp. Post-Prec-SPA) performs slightly better than SPA (resp. Prec-SPA). However, it is surprising that both post-processed variants of SPA extract columns of W for large noise levels as the columns of W are contained in the convex hull of the other columns.

Table 1 gives the robustness and the average running time of the different algorithms. Prec-SPA and Post-Prec-SPA are the slowest because they have to solve the SDP (1). XRAY is the second slowest because it has to solve nonnegative least squares problems at each step.

Table 1: Robustness (that is, largest value of ϵ for which all columns of W are correctly identified) and average running time in seconds of the different near-separable NMF algorithms.

	SPA	Post-SPA	Prec-SPA	Heur-SPA	Post-Prec-SPA	VCA	XRAY
Robustness	0.01	0.03	0.45	0.45	0.45	0	0.01
Time (s.)	3.4^*10^{-3}	1.8^*10^{-2}	2.8	2.4^*10^{-2}	2.8	2.3^*10^{-2}	0.69

5.1.2 Hubble Telescope

In hyperspectral imaging, the columns of the input matrix are the spectral signatures of the pixels present in the image. Under the linear mixing model, the spectral signature of each pixel is equal to a linear combination of the spectral signatures of the materials present in the image (referred to as endmembers). The weights in the linear combinations represent the abundances of the endmembers in the pixels. If for each endmember, there is a least one pixel containing only that endmember, the separability assumption is satisfied: this is the so-called pure-pixel assumption un hyperspectral imaging. Therefore, under the linear mixing model and the pure-pixel assumptions, hyperspectral unmixing is equivalent to near-separable NMF and the aim is to identify one pure pixel per endmember; see [15] and the references therein for more details.

In this section, we analyze the simulated Hubble telescope hyperspectral image with 100 spectral bands and 128×128 pixels [29] (that is, $m = 100$ and $n = 16384$); see Figure 2. It is composed of eight materials (that is, $r = 8$); see Figure 3.
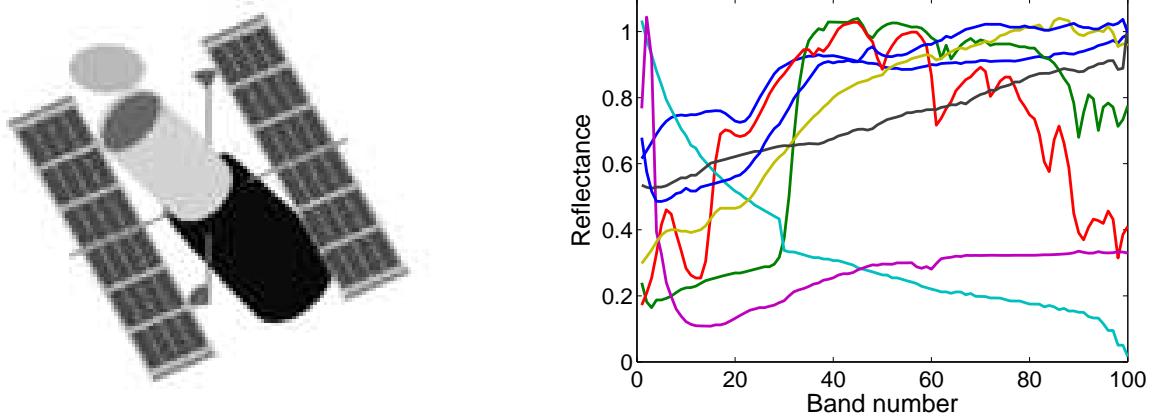
16

Figure 2: On the left: sample image from the Hubble telescope hyperspectral image. On the right: spectral signatures of the eight endmembers.
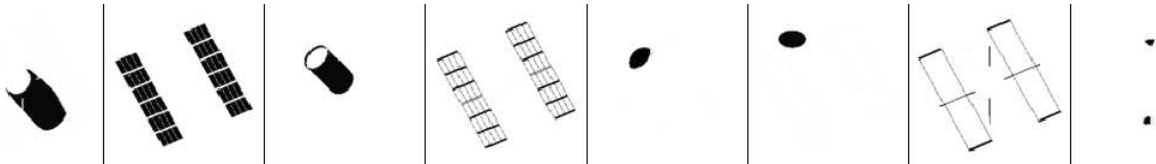

Figure 3: The eight materials for the Hubble telescope data provided by the NASA Johnson Space Center. From left to right: aluminum, solar cell, green glue, copper stripping, honeycomb side, honeycomb top, black rubber edge and bolts.

On the clean image, all the SPA variants listed above are able to recover the eight materials perfectly[5]. We add blur and noise as in [29] (point spread function on 5-by-5 pixels and with standard deviation of 1, and white Gaussian noise $\sigma = 1\%$ of the values of M and Poisson noise $\sigma = 1\%$ of the mean value of M), and run all the algorithms. We then compute the mean-removed spectral angle (MRSA) between the true endmembers (of the clean image) and the extracted endmembers. Given two spectral signatures, $x, y \in \mathbb{R}^m$, the MRSA is defined as

$$\phi(x,y) = \frac{100}{\pi} \arccos\left(\frac{(x - \bar{x})^T(y - \bar{y})}{||x - \bar{x}||_2 ||y - \bar{y}||_2}\right) \quad \in \quad [0, 100],$$

where, for a vector $z \in \mathbb{R}^m$, $\bar{z} = \frac{1}{m}\left(\sum_{i=1}^m z_i\right)e$ and e is the vector of all ones. The MRSA measures how close two endmembers are (neglecting scaling and translation); 0 meaning that they match perfectly, 100 that they do not match at all.

Table 2 reports the results, along with the running time of the different algorithms. Note that (i) we match the extracted endmembers with the true endmembers in order to minimize the average MRSA[6], and (ii) we do not include Post-Prec-SPA because it gave exactly the same solution as Prec-SPA.

---

[5]Note that VCA and XRAY are not able to identify all materials perfectly.

[6]We use http://www.mathworks.com/matlabcentral/fileexchange/20328-munkres-assignment-algorithm.

Table 2: MRSA of the identified endmembers with the true endmembers, and running time in seconds of the different near-separable NMF algorithms.

	SPA	Post-SPA	Prec-SPA	Heur-SPA	VCA	XRAY
Honeycomb side	**6.51**	6.94	6.94	6.94	48.70	45.78
Copper Stripping	26.83	7.46	7.46	**7.44**	48.98	46.93
Green glue	2.09	2.09	**2.03**	**2.03**	48.30	42.70
Aluminum	**1.71**	1.80	1.80	1.80	42.31	44.54
Solar cell	**4.96**	5.48	5.48	5.48	48.74	46.31
Honeycomb top	2.34	**2.07**	2.30	2.30	49.90	53.23
Black rubber edge	27.09	45.94	**13.16**	**13.16**	47.21	46.55
bolts	**2.65**	**2.65**	**2.65**	**2.65**	47.20	45.81
Average	9.27	9.30	**5.23**	**5.23**	47.88	46.48
Time (s.)	0.05	1.45	4.74	2.18	0.37	1.53

We also computed the abundance maps corresponding to the extracted endmembers, that is, we solved the nonnegative least squares problem $H = \text{argmin}_{X \geq 0} ||M - M(:, \mathcal{K})X||_F$ where $\mathcal{K}$ is the extracted index set by a given algorithm. This allows to visually asses the quality of the extracted endmembers; see Figure 5.

As in for the synthetic data sets in Section 5.1.1, Prec-SPA and Heur-SPA perform the best. Moreover, for this data set, the running time of Prec-SPA is comparable to the one of Heur-SPA (because computing the SVD is more expensive; see next section). Figure 4 displays the spectral signatures extracted by SPA and Prec-SPA, which shows that SPA is not able to identify one of the endmembers (in fact, one is extracted twice). It is also interesting to notice that, in this example,
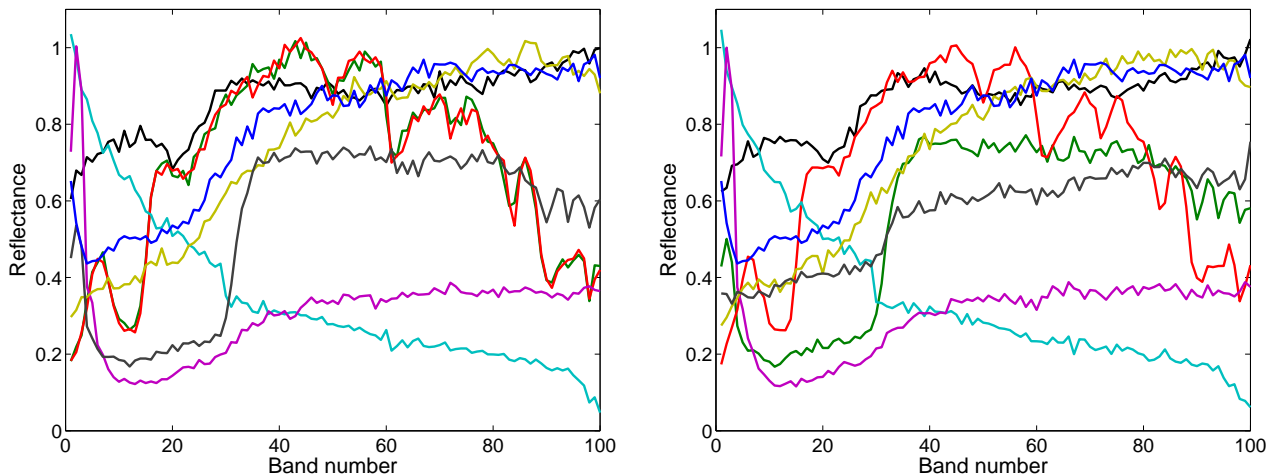

Figure 4: Extracted endmembers by SPA (left) and Prec-SPA (right).

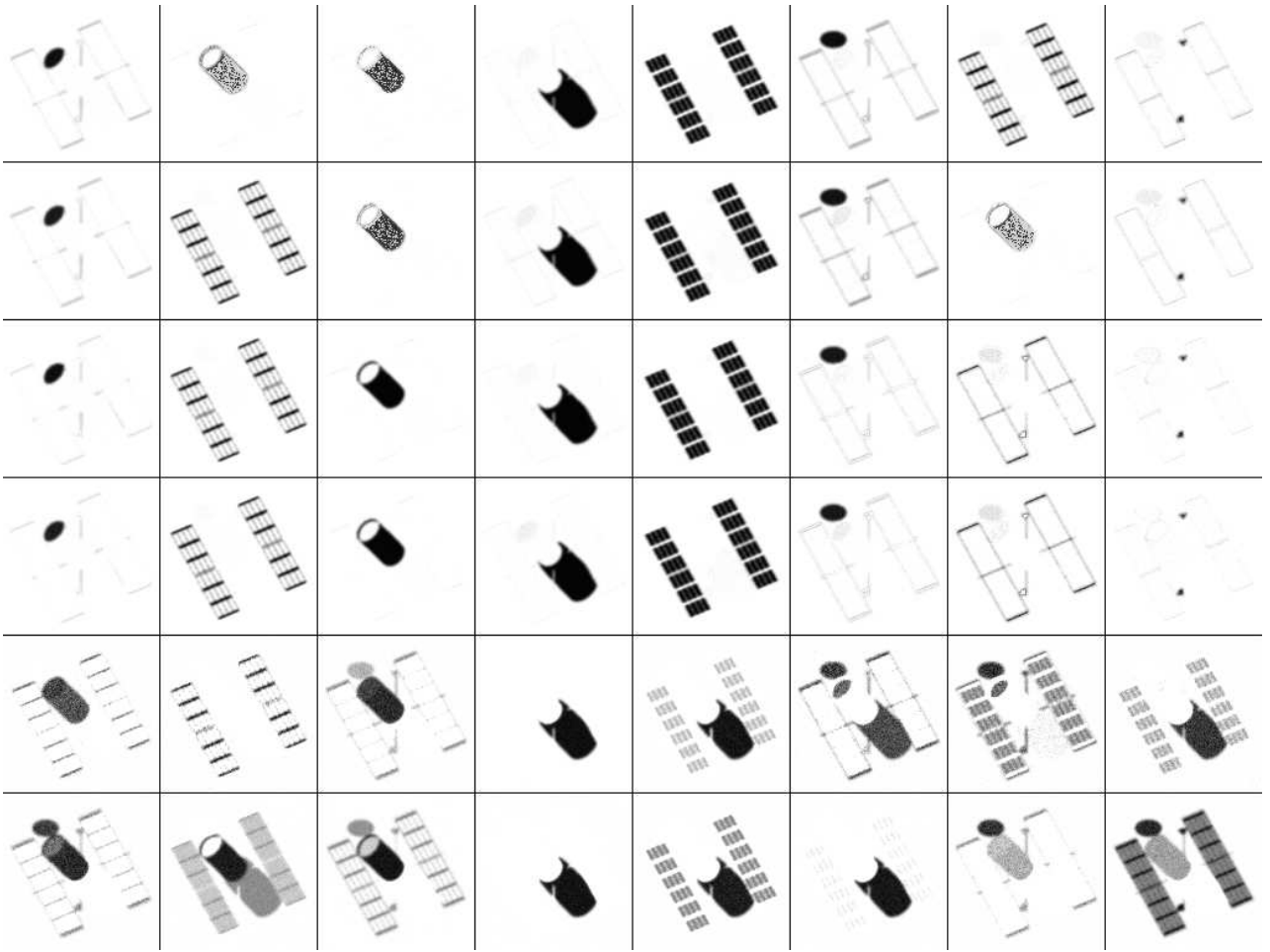SPA performs slightly better than Post-SPA.

Figure 5: The abundance maps corresponding to the extracted spectral signatures by the different algorithms. From top to bottom: SPA, Post-SPA, Prec-SPA, Heur-SPA, VCA and XRAY.

5.2 Real-World Hyperspectral Images

Assessing the quality of the extracted endmembers is rather challenging for real-world HSI because the true endmembers are unknown (in most cases, even the number of endmembers is unknown) while outliers and other artifacts in the image should be handled separately; see, e.g, [25] and the references therein. The aim of this section is therefore not to assess the performance of Prec-SPA on real-world hyperspectral images (this is out of the scope of this paper) but rather to show that the combination of the linear dimensionality reduction technique with the active-set method (that is, Algorithm 2) is applicable to this type of large-scale data sets. We report the running time of Algorithm 2 for several popular hyperspectral images. We also report the difference between the solutions obtained with Prec-SPA and Heur-SPA (to illustrate the fact that they might behave quite differently in some cases); see Table 3 for the numerical results.

In all cases, less than five changes in the active set are necessary for Algorithm 2 to terminate, which explains why the computational cost to solving the SDP is comparable to the one of computing

19

Table 3: Running time in seconds for Prec-SPA (Algorithm 2) on several hyperspectral images: *SVD* is the time in seconds needed to compute the truncated SVD of rank r, *SDP* the time in seconds needed to solve the SDP (1) using the active-set method, *Heur ∩ Prec* is the number of common indices extracted by Prec-SPA and Heur-SPA, and *# Active sets* is the number of changes in the active set needed for Algorithm 2 to terminate.

	m	n	r	SVD	SDP	Prec ∩ Heur	# Active sets
Urban*	162	94249	6	7.3	1.7	6	3
			12	21.9	2.4	9	3
San Diego**	158	160000	8	17.1	1.5	8	2
			16	32.1	7.2	13	4
Cuprite**	188	47750	15	11.0	6.6	6	5
			30	18.2	58.3	3	4

* http://www.agc.army.mil.
** http://aviris.jpl.nasa.gov.

the SVD. (Note that this will only be true for relatively small r.) It is interesting to observe that for some of these data sets Prec-SPA and Heur-SPA perform rather differently, especially when r increases (for example, for the Cuprite data set with $r = 30$ they only extract three common indices).

6 Conclusion and Further Work

In this paper, we have proposed a way to precondition near-separable NMF matrices using semidefinite programming. This is turn allowed us to robustify near-separable NMF algorithms. In particular, the preconditioning makes the popular successive projection algorithm (SPA) provably more robust to noise, which we have illustrated on some synthetic data sets. Moreover, we showed how to apply the preconditioned SPA on real-world hyperspectral images using the SVD and an active-set method.

For now, the preconditioning can be computed as long as (i) the SVD can be computed, and (ii) r is not too large[7] (say $r \sim 50$). For larger data sets, several alternatives to the SVD are possible, such as random projections, and are interesting directions for further research; see also Remark 6. For larger r, it would be useful to develop faster SDP solvers, e.g., using first-order methods and/or using the particular structure of the minimum volume ellipsoid problem; see [21, 20, 23, 32, 1] and the references therein.

In the future, it would also be particularly interesting to assess the performance of preconditioned SPA on real-world problems (e.g., in hyperspectral unmixing and document classification), and evaluate the effect of the preconditioning on other algorithms (theoretically and/or practically).

---

[7]With CVX, it took about 3 minutes to solve a problem with $m = r = 50$ and $n = 1000$ on a randomly generated matrix. For $r = 100$, Matlab runs out of memory.

References

[1] Ahipasaoglu, S., Sun, P., Todd, M.: Linear convergence of a modified Frank-Wolfe algorithm for computing minimum-volume enclosing ellipsoids. Optimization Methods and Software **23**(1), 5–19 (2008)

[2] Araújo, U., Saldanha, B., Galvão, R., Yoneyama, T., Chame, H., Visani, V.: The successive projections algorithm for variable selection in spectroscopic multicomponent analysis. Chemometrics and Intelligent Laboratory Systems **57**(2), 65–73 (2001)

[3] Arora, S., Ge, R., Halpern, Y., Mimno, D., Moitra, A., Sontag, D., Wu, Y., Zhu, M.: A practical algorithm for topic modeling with provable guarantees. In: International Conference on Machine Learning (ICML '13), vol. 28, pp. 280–288 (2013)

[4] Arora, S., Ge, R., Kannan, R., Moitra, A.: Computing a nonnegative matrix factorization – provably. In: Proceedings of the 44th symposium on Theory of Computing, STOC '12, pp. 145–162 (2012)

[5] Arora, S., Ge, R., Moitra, A.: Learning topic models - going beyond SVD. In: Proceedings of the 53rd Annual IEEE Symposium on Foundations of Computer Science, FOCS '12, pp. 1–10 (2012)

[6] Avron, H., Boutsidis, C.: Faster Subset Selection for Matrices and Applications (2011). ArXiv:1201.01272

[7] Bittorf, V., Recht, B., Ré, E., Tropp, J.: Factoring nonnegative matrices with linear programs. In: Advances in Neural Information Processing Systems, NIPS '12, pp. 1223–1231 (2012)

[8] Boyd, S., Vandenberghe, L.: Convex Optimization. Cambridge University Press, Cambridge (2004)

[9] Chan, T.H., Ma, W.K., Chi, C.Y., Wang, Y.: A convex analysis framework for blind separation of non-negative sources. IEEE Trans. on Signal Processing **56**(10), 5120–5134 (2008)

[10] Chen, L., Choyke, P., Chan, T.H., Chi, C.Y., Wang, G., Wang, Y.: Tissue-specific compartmental analysis for dynamic contrast-enhanced MR imaging of complex tumors. IEEE Trans. on Medical Imaging **30**(12), 2044–2058 (2011)

[11] Ge, R.: private communication (2013)

[12] Gillis, N.: Sparse and unique nonnegative matrix factorization through data preprocessing. Journal of Machine Learning Research **13**(Nov), 3349–3386 (2012)

[13] Gillis, N.: Robustness analysis of Hottopixx, a linear programming model for factoring nonnegative matrices. SIAM J. Mat. Anal. Appl. **34**(3), 1189–1212 (2013)

[14] Gillis, N., Luce, R.: Robust near-separable nonnegative matrix factorization using linear optimization (2013). ArXiv:1302.4385

[15] Gillis, N., Vavasis, S.: Fast and robust recursive algorithms for separable nonnegative matrix factorization (2012). ArXiv:1208.1237v3

[16] Grant, M., Boyd, S.: Graph implementations for nonsmooth convex programs. Recent Advances in Learning and Control (a tribute to M. Vidyasagar), V. Blondel, S. Boyd, and H. Kimura, editors, pages 95-110, Lecture Notes in Control and Information Sciences, Springer (2008). `http://stanford.edu/~boyd/graph_dcp.html`

[17] Grant, M., Boyd, S.: CVX: Matlab software for disciplined convex programming, version 1.21. `http://cvxr.com/cvx/` (April 2011)

[18] John, F.: Extremum problems with inequalities as subsidiary conditions. In J. Moser, editor, *Fritz John, Collected Papers* pp. 543–560 (1985). First published in 1948

[19] Johnson, W., Lindenstrauss, J.: Extensions of Lipschitz mappings into a Hilbert space. In: Conference in Modern Analysis and Probability, Contemporary Mathematics 26, pp. 189–206 (1984)

[20] Khachiyan, L.: Rounding of polytopes in the real number model of computation. Mathematics of Operations Research **21**, 307320 (1996)

[21] Khachiyan, L., Todd, M.: On the complexity of approximating the maximal inscribed ellipsoid for a polytope. Mathematical Programming **61**, 137–159 (1993)

[22] Kumar, A., Sindhwani, V., Kambadur, P.: Fast conical hull algorithms for near-separable non-negative matrix factorization. In: International Conference on Machine Learning (ICML '13), vol. 28, pp. 231–239 (2013)

[23] Kumar, P., Yildirim, E.: Minimum-Volume Enclosing Ellipsoids and Core Sets. Journal of Optimization Theory and Applications **126**(1), 1–21 (2005)

[24] Lee, D., Seung, H.: Learning the Parts of Objects by Nonnegative Matrix Factorization. Nature **401**, 788–791 (1999)

[25] Ma, W.K., Bioucas-Dias, J., Gader, P., Chan, T.H., Gillis, N., Plaza, A., Ambikapathi, A., Chi, C.Y.: A signal processing perspective on hyperspectral unmixing (2014). To appear in IEEE Signal Processing Magazine, Special Issue on Signal and Image Processing in Hyperspectral Remote Sensing

[26] Mizutani, T.: Ellipsoidal Rounding for Nonnegative Matrix Factorization Under Noisy Separability (September 2013). ArXiv:1309.5701

[27] Nascimento, J., Bioucas-Dias, J.: Vertex component analysis: a fast algorithm to unmix hyperspectral data. IEEE Transactions on Geoscience and Remote Sensing **43**(4), 898–910 (2005)

[28] Papadimitriou, C., Raghavan, P., Tamaki, H., Vempala, S.: Latent semantic indexing: A probabilistic analysis. In: Proceedings of the 17th ACM SIGACT-SIGMOD Symposium on Principles of Database Systems, pp. 159–168 (1998)

[29] Pauca, P., Piper, J., Plemmons, R.: Nonnegative matrix factorization for spectral data analysis. Linear Algebra and its Applications **406**(1), 29–47 (2006)

[30] Robinson, S.: Generalized equations and their solutions, Part II: Applications to nonlinear programming. Mathematical Programming Study **19**, 200–221 (1982)

[31] Sun, P., Freund, R.: Computation of minimum-volume covering ellipsoids. Operations Research **52**(5), 690–706 (2004)

[32] Todd, M., Yildirim, E.: On Khachiyan's algorithm for the computation of minimum-volume enclosing ellipsoids. Discrete Applied Mathematics **155**, 17311744 (2007)

[33] Vavasis, S.: On the complexity of nonnegative matrix factorization. SIAM J. on Optimization **20**(3), 1364–1377 (2009)

A   Proof for Lemma 3

In this appendix, we prove the following: Let r be any integer larger than 2 and $\lambda \geq 0$ be such that

$$(r-1)\left(1+\frac{1}{8r}\right)^{-\frac{2r}{r-1}} \leq \lambda^{\frac{1}{r-1}}(r+1-\lambda). \tag{9}$$

Then, $\lambda \leq 4$.

Proof. The function $f(\lambda) = \lambda^{\frac{1}{r-1}}(r+1-\lambda)$ is nonincreasing for any $r \geq 2$ and for $\lambda \geq 2$; in fact,

$$\frac{d}{d\lambda}f(\lambda) = \frac{r+1-\lambda}{r-1}\lambda^{\frac{2-r}{r-1}} - \lambda^{\frac{1}{r-1}} = \lambda^{\frac{1}{r-1}}\left(\underbrace{\frac{r+1-\lambda}{r-1}}_{\leq 1}\underbrace{\lambda^{2-r}}_{\leq 1}-1\right) \leq 0.$$

Since the left-hand side of (9) does not depend on λ, showing that the inequality above is violated for $\lambda = 4$ will imply that it is violated for any $\lambda \geq 4$ (hence $\lambda < 4$). It remains to show that

$$(r-1)\left(1+\frac{1}{8r}\right)^{-\frac{2r}{r-1}} > 4^{\frac{1}{r-1}}(r-3) \qquad \text{for any integer } r \geq 2.$$

It can be checked numerically that the inequality holds for any integer $r \leq 12$ hence it remains to show it holds for any $r \geq 13$. The inequality is equivalent to

$$\left(\frac{r-1}{r-3}\right)^{r-1} > 4\left(1+\frac{1}{8r}\right)^{2r}.$$

We have $\left(1+\frac{1}{8r}\right)^2 = 1 + \frac{1}{4r} + \frac{1}{64r^2} \leq 1 + \frac{1}{3r}$ while $\frac{r-1}{r-3} < \frac{5}{4}$ for any $r \geq 13$ hence it is sufficient to show that

$$\left(\frac{r-1}{r-3}\right)^r > 5\left(1+\frac{1}{3r}\right)^r \qquad \text{for } r \geq 13.$$

The right-hand side is an increasing function whose limit is given by $5e^{1/3} \approx 6.978$. In fact, $e^x \geq \left(1+\frac{x}{n}\right)^n$ for all $x \geq 0$ and $n > 0$. The left-hand side is decreasing for $r \geq 4$. In fact, its derivative is given by

$$\frac{1}{r-3}\left(\frac{r-1}{r-3}\right)^{r-1}\underbrace{\left((r-1)\ln\left(\frac{r-1}{r-3}\right) - \frac{2r}{(r-3)}\right)}_{h(r)}.$$

while using $\ln(x) \leq x - 1$ we have $\ln\left(\frac{r-1}{r-3}\right) \leq \frac{r-1}{r-3} - 1 = \frac{2}{r-3}$ hence

$$h(r) \leq (r-1)\frac{2}{r-3} - \frac{2r}{r-3} = \frac{-2}{r-3} \leq 0 \qquad \text{for any } r \geq 4.$$

Finally, the limit of the left-hand side for $r \to +\infty$ is given by

$$\lim_{r\to+\infty}\left(\frac{r-1}{r-3}\right)^r = \lim_{r\to+\infty}\left(1+\frac{2}{r-3}\right)^r = \lim_{r\to+\infty}\left(\frac{r-1}{r-3}\right)^3 \lim_{r\to+\infty}\left(1+\frac{2}{r-3}\right)^{r-3} = e^2 \approx 7.389 > 5e^{1/3}.$$

$\square$