Algorithms and Comparisons of Nonnegative Matrix Factorizations with Volume Regularization for Hyperspectral Unmixing

Andersen Man Shun Ang, *Member, IEEE,* and Nicolas Gillis, *Member, IEEE*

***Abstract*—In this work, we consider nonnegative matrix factorization (NMF) with a regularization that promotes small volume of the convex hull spanned by the basis matrix. We present highly efficient algorithms for three different volume regularizers, and compare them on endmember recovery in hyperspectral unmixing. The NMF algorithms developed in this work are shown to outperform the state-of-the-art volume-regularized NMF methods, and produce meaningful decompositions on real-world hyperspectral images in situations where endmembers are highly mixed (no pure pixels). Furthermore, our extensive numerical experiments show that when the data is highly separable, meaning that there are data points close to the true endmembers, and there are a few endmembers, the regularizer based on the determinant of the Gramian produces the best results in most cases. For data that is less separable and/or contains more endmembers, the regularizer based on the logarithm of the determinant of the Gramian performs best in general.**

***Index Terms*—nonnegative matrix factorization, volume regularization, hyperspectral unmixing, blind source separation**

I. Introduction

Non-negative Matrix Factorization (NMF) is the following problem: given a matrix $\mathbf{X} \in \mathbb{R}^{m \times n}$ and an integer r, find two matrices $\mathbf{W} \in \mathbb{R}^{m \times r}_+$ and $\mathbf{H} \in \mathbb{R}^{r \times n}_+$ such that

$$(\text{NMF}): \ \mathbf{X} \cong \mathbf{WH}. \tag{1}$$

NMF has many applications; see, e.g., [4], [9], [6] and the references therein. In this work, we focus on blind hyperspectral unmixing (HU) [2] that aims at recovering the spectral signatures of the pure materials (called endmembers, represented as the columns of $\mathbf{W}$) and their abundances in each pixel (represented as the columns of $\mathbf{H}$) within a hyperspectral image $\mathbf{X}$ where each column of $\mathbf{X}$ is the spectral signature of a pixel; see Section II for more details. In order to tackle HU, we will use *volume-regularized NMF* (VRNMF) which can be formulated as follows

$$\begin{aligned} \min_{\mathbf{W},\mathbf{H}} \quad & f(\mathbf{W},\mathbf{H};\mathbf{X}) + \lambda V(\mathbf{W}) \\ \text{subject to} \quad & \mathbf{W} \geq 0, \ \mathbf{H} \geq 0, \ \mathbf{H}^\top \mathbf{1}_r \leq \mathbf{1}_n, \end{aligned} \tag{2}$$

where $\mathbf{1}_n$ denotes the vector of ones of length n and $\mathbf{W} \geq 0$ indicates that $\mathbf{W}$ is component-wise non-negative. The regularization parameter $\lambda \geq 0$ controls the trade off between the data fitting term $f(\mathbf{W},\mathbf{H};\mathbf{X})$ and the volume regularizer $V(\mathbf{W})$. The constraint $\mathbf{H}^\top \mathbf{1}_r \leq \mathbf{1}_n$ is a relaxation of the sum-to-one constraint $\mathbf{H}^\top \mathbf{1}_r = \mathbf{1}_n$ which requires that the abundances of the endmembers in each pixel sum to one. Because VRNMF minimizes the volume of $\mathbf{W}$, the constraint $\mathbf{H}^\top \mathbf{1}_r \leq \mathbf{1}_n$ will tend to be active for most pixels. It has been shown previously that this relaxation works better in practice as it allows to take into account for example different illumination conditions within the hyperspectral image; see for example [8]. The reason to consider VRNMF has a long history in blind HU and is motivated by geometric insights: the columns of $\mathbf{W}$ (the endmembers) are the vertices of a convex hull that contains the data points. In the absence of pure pixels, that is, pixels containing a single endmember, minimizing the volume of the columns of $\mathbf{W}$ allows to recover these endmemebers under mild conditions; see Figure 1 for an illustation, and Section II for more details.

In this paper, we will consider the most widely used data fitting term, namely the least squares error $f(\mathbf{W},\mathbf{H};\mathbf{X}) = \frac{1}{2}\|\mathbf{X} - \mathbf{WH}\|_F^2 = \frac{1}{2}\sum_{i,j}(\mathbf{X} - \mathbf{WH})_{ij}^2$. For the volume regularizer, we will consider the following three functions

$$\begin{aligned} \text{Detminant:} \quad & V_{\det}(\mathbf{W}) &&= \frac{1}{2}\det(\mathbf{W}^\top\mathbf{W}), \\ \text{Log-det:} \quad & V_{\text{logdet}}(\mathbf{W}) &&= \frac{1}{2}\log\det(\mathbf{W}^\top\mathbf{W} + \delta\mathbf{I}_r), \\ \text{Nuclear:} \quad & V_*(\mathbf{W}) &&= \|\mathbf{W}\|_*, \end{aligned}$$

where $\det(\mathbf{A})$ is the determinant of matrix $\mathbf{A}$, $\mathbf{I}_r$ is identity matrix of size r, $\delta > 0$ is a constant to lower bound V_{logdet}, and $\|\mathbf{A}\|_*$ is the nuclear norm of $\mathbf{A}$, that is, the sum of the singular values of $\mathbf{A}$. VRNMF aims at fitting the data points within the convex hull of the columns of $\mathbf{W}$ which should have a small volume. The reason to consider the functions $V_{\det}$ and V_{logdet} is that $\sqrt{\det(\mathbf{W}^\top\mathbf{W})}/r!$ is the volume of the convex hull of the columns of $\mathbf{W}$ and the origin. Hence, $V_{\det}$ is, up to a constant multiplicative factor, the square of that volume, while $V_{\det}$ is its logarithm. Let us write $V_{\det}$ and V_{logdet} as functions of the singular values of $\mathbf{W}$, denoted $\sigma_i(\mathbf{W})$ for $1 \leq i \leq r$:

$$V_{\det}(\mathbf{W}) = \frac{1}{2}\prod_{i=1}^{r}\sigma_i^2(\mathbf{W}),$$

and

$$V_{\text{logdet}}(\mathbf{W}) = \sum_{i=1}^{r}\log(\sigma_i^2(\mathbf{W}) + \delta).$$

A. Ang and N. Gillis are with the Department of Mathematics and Operational Research, Université de Mons, Belgium. E-mails: {manshun.ang,nicolas.gillis}@umons.ac.be. This work was supported by the European Research Council (ERC starting grant no 679515), and the Fonds de la Recherche Scientifique - FNRS and the Fonds Wetenschappelijk Onderzoek - Vlanderen (FWO) under EOS Project no O005318F-RG47.

Both functions V_{det} and V_{logdet} are non-decreasing functions of the singular values of $\mathbf{W}$. This motivates us to consider

$$V_*(\mathbf{W}) = \|\mathbf{W}\|_* = \sum_{i=1}^{r} \sigma_i(\mathbf{W}).$$

The reason is that this regularizer is also a non-decreasing function of the singular values of $\mathbf{W}$, and has been widely used in machine learning for several tasks such as matrix completion [20]. However, to the best of our knowledge, it has never been used in the context of VRNMF and it would be interesting to know how it compares to the standard choices V_{det} and V_{logdet}. As we will see, this regularizer also performs well in practice, although not as well as V_{det} and V_{logdet}.

The approach of using a volume regularization with NMF has a long history and has been considered for example in [15], [21], [24], [7], [1], [6], [13]. The key differences among these works are in the choice of V. Almost all previous works have focused on the two functions V_{det} and V_{logdet}. We believe it is important to design efficient algorithms for these regularizes, and to compare them on solving blind HU on highly mixed hyperspectral images in order to know which one performs better in which situations. These are the main goals of this paper.

The contribution of this work is twofold: the first part of this work is algorithm design, in which we implement and enhance the algorithms to solve VRNMF by using block coordinate descent and optimal first-order methods. Experimental results will show that our algorithms perform better than the current state-of-the-art volume-regularization based method from [7]. The second part of this work is focused on model comparisons. We will answer the question "which volume function is better suited for VRNMF to tackle HU?". We do so by performing extensive numerical comparisons of VRNMF with different volume functions under various settings. The summary of the findings are as follows:

- For data that is highly separable, meaning that there exists data points close to the true endmembers, and in the presence of a few endmemebers, VRNMF with V_{det} produces the best results in most cases.
- For data that is less separable or in the presence of a large number of endmemebers, VRNMF with V_{logdet} performs best in general.

Finally, as a proof of concept, we showcase the ability of VRNMF to produces a meaningful unmixing on hyperspectral images using real-world data.

This work is the continuation of the conference paper [1]. The additional contributions of this extended version are the following:

- We base our numerical experiments only on real endmembers, as opposed to randomly generated ones in [1].
- We use a fine grid search by bisection to tune the regularization parameter λ.
- We implement VRNMF with the nuclear norm regularizer.
- We have improved our implementations; they are available from https://angms.science/research.html.
- We compare our implementation of VRNMF with the state-of-the-art volume-regularization algorithm `RVolMin` [7].

The remaining of this paper is organized as follows. In Section II, we give a more complete introduction to HU, followed by the discussion of the pure-pixel assumption also known as the separability assumption. This motivates the use of VRNMF when this assumption is violated. Section III gives the details about the enhancement and implementation of the algorithms for VRNMF. Section IV presents the experiments on VRNMF and Section V concludes the paper.

II. Brief review on HU

We now give a brief review on HU; see [2], [14] and the references therein for more details. The goal of blind HU is to study the composition and the distribution of materials in a given scene being imaged. A scene usually consists of a few fundamental types of materials called *endmembers*, and the first goal of blind HU is to obtain the information of these endmembers from the observed hyperspectral image (HSI). HSI are images captured by sensors over different wavelengths in the electromagnetic spectrum. These images form a hyperspectral data cube of size $m \times \text{col} \times \text{row}$, where m is the number of spectral bands, and "col" and "row" are the dimensions of the images, with $n = \text{col} \times \text{row}$. The m-by-n data matrix $\mathbf{X}$ is obtained by stacking the m-dimensional spectral signature of the pixels as the columns of $\mathbf{X}$. Given the observed data matrix $\mathbf{X}$, the goal of blind HU is to (i) identify the number of endmembers r, (ii) obtain the spectral signature of these endmembers, (iii) identify which pixel contains which endmember and in which proportion. This work does not consider problem (i) by assuming r is known. In fact, model order selection is a topic of research on its own; see, e.g., [2] and the references therein. The focus of this work is (ii): recover the (ground truth) endmember spectral matrix, denoted by $\mathbf{W}^{\text{true}}$, from the observed data $\mathbf{X}$. In fact, assuming (ii) is solved, (iii) can be tackled by solving a (convex) nonnegative least squares problem [2], [14]. The most widely used model for blind HU is the linear mixing model for which a hidden low-rank linear mixing structure for the data, namely $\mathbf{X} = \mathbf{W}^{\text{true}}\mathbf{H}^{\text{true}} + \mathbf{N}$. The data $\mathbf{X}$ is hence generated by $\mathbf{W}^{\text{true}}$ (the basis, where each column of $\mathbf{W}^{\text{true}}$ is the spectral signature of an endmember), weighted by the matrix $\mathbf{H}^{\text{true}}$ plus some noise $\mathbf{N}$. The matrix $\mathbf{H}$ in the HU literature is also called the *abundance matrix* and encodes how much each endmember (columns of $\mathbf{W}^{\text{true}}$) is present in each pixel of the image: $\mathbf{H}_{k,j}$ is the abundance of the kth endmember in the jth pixel. There are two physical constraints in HU: the non-negativity constraints $\mathbf{W} \geq 0$ (spectral signatures are nonnegative), and the nonnegativity $\mathbf{H} \geq 0$ and sum-to-one constraint $\mathbf{H}^\top \mathbf{1}_r = \mathbf{1}_n$ (the abundances in each pixel are nonnegative and sum to one). In this paper, we consider a more general model, namely using $\mathbf{H}^\top \mathbf{1}_r \leq \mathbf{1}_n$, that allows to take into account different intensities of illumination among the pixels of the image. Finally, NMF is the right model to perform blind HU under the linear mixing model, that is, to learn the endmember matrix $\mathbf{W}$ and the abundance matrix $\mathbf{H}$

from the data matrix $\mathbf{X}$. However, NMF is a difficult problem in general [22].

A. Pure-pixel assumption

Separable NMF (SNMF) is able to solve blind HU when the data satisfies the *separability* condition, which is also known as the *pure-pixel assumption* in HU. It means the data $\mathbf{X}$ contains at least r pure pixels where each pure pixel contains only one endmember, and there is a one-to-one correspondence between the r pure pixels and the r endmembers. Mathematically, separability changes the NMF model (1) to

$$(\text{SNMF}): \ \mathbf{X} \cong \mathbf{W} \underbrace{[\mathbf{I}_r \ \mathbf{H}']\mathbf{\Pi}_n}_{\mathbf{H}} = [\mathbf{X}(:,\mathcal{A}) \ \mathbf{X}(:,\mathcal{A})\mathbf{H}']\mathbf{\Pi}_n,$$

where $\mathbf{\Pi}_n$ is a n-by-n permutation matrix, and $\mathbf{H}' \in \mathbb{R}_+^{r\times(n-r)}$ has its columns with l_1 norm smaller than one. In SNMF, we have that $\mathbf{W} = \mathbf{X}(:,\mathcal{A})$, that is, the index set $\mathcal{A}$ contains the indices of the pure pixels. Since all columns of $\mathbf{H}'$ have l_1 norm smaller than one, $\mathbf{X}(:,\mathcal{A})$ and the origin are the r extreme points of the data cloud $\mathbf{X}$, that is, the convex hull of $\mathbf{X}(:,\mathcal{A})$ and the origin encapsulates all the other points in $\mathbf{X}$. Hence, SNMF is geometrically a vertex identification problem: given $\mathbf{X}$, locate the extreme points $\mathbf{W} = \mathbf{X}(:,\mathcal{A})$ which will be exactly $\mathbf{W}^{\text{true}}$ if the separability condition holds and no noise is present. Many algorithms exist to perform this task (referred to as pure-pixel search algorithms), e.g., vertex component analysis (VCA) [16] or the successive projection algorithm (SPA) [11]; see [2], [14] and the references therein for more algorithms and discussions. In this paper, we will compare VRNMF to SPA, as SPA is a state-of-the-art provably robust pure-pixel search algorithm.

However, when the separability condition does not hold, pure-pixel search algorithms fail. In order to quantify how much separability is violated, we introduce the *non-separability parameter* $p \in (0,1]^r$. First, note that separability holds if and only if each row of $\mathbf{H}$ contains at least one entry equal to one, that is, $\|\mathbf{h}^j\|_\infty = 1$ for $j \in [r]$, where $\mathbf{h}^j$ denotes the j^{th} row of $\mathbf{H}$ and $[r] = \{1, 2, \ldots, r\}$. Therefore, to break the separability condition, we need $\|\mathbf{h}^j\| \leq p_j < 1$ for some j. Figure 1 gives an example with $r = 4$. Hence, having $\|\mathbf{h}^j\| \leq p_j$ means that the maximum abundance of the jth endmember in all pixels is at most p_j, which sets a minimal amount of separation between the data points (the black dots in Figure 1) to the vertex $\mathbf{w}_j$ (the black stars). In other words, p_j controls the gap between $\{\mathbf{x}_i\}_{i\in[n]}$ and $\mathbf{w}_j$, where $\mathbf{x}_i$ is the ith data point (ith column of $\mathbf{X}$). Note that the entries of p can be different in general meaning that we have *asymmetric non-separability*. For example, in Figure 1, data points are closer to vertex 1 than vertex 4. In the absence of pure pixels, that is, $p_j < 1$ for some j, it has been proved that, under mild conditions, minimizing the volume of the convex hull of the columns of $\mathbf{W}$ allows to recover the endmembers. These conditions require that the data points are well spread within the convex hull spanned by the columns of $\mathbf{W}$; see [6] for a recent survey on the topic.

Note that in Figure 1, the reconstructions given by SPA [11] (green vertices) and `RVolMin` [7] (deep blue vertices, a state-of-the-art minimum-volume NMF algorithm) are far away from the ground truth. VRNMF with V_{det} (cyan vertices) produces a perfect recovery of $\mathbf{W}^{\text{true}}$. The next section describes how we solve the VRNMF minimization problem to obtain such results.

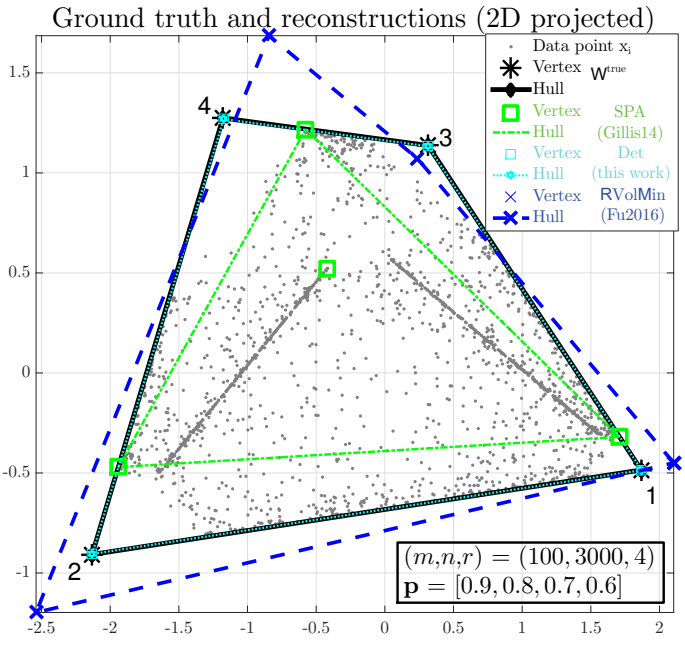


Fig. 1. A toy example with $(m, n, r) = (100, 3000, 4)$. The plot shows the projection of data points ($\mathbf{x}_i$, the black dots) in two dimensions using PCA. This data set was generated using $\mathbf{p} = [0.9, 0.8, 0.7, 0.6]$, meaning the maximum abundance of each ground truth vertex ($\mathbf{W}^{\text{true}}$, the black stars) in any pixel is at most 90% for vertex 1, 80% for vertex 2, 70% for vertex 3, and 60% for vertex 4.

III. Solving VRNMF

In this section, we describe how to solve (2). We use the framework of block coordinate descent (BCD) by solving the subproblems on $\mathbf{W}$ and $\mathbf{H}$ separately in an alternating scheme, as done in most NMF works [9]. Let us start with the subproblem for $\mathbf{H}$.

A. Subproblem for $\mathbf{H}$

Splitting $\mathbf{H}$ into columns yields n independent problems: for $1 \leq j \leq n$, solve

$$\min_{\mathbf{h}_j \in \Delta} \frac{1}{2}\|\mathbf{x}_j - \mathbf{W}\mathbf{h}_j\|_2^2, \ \Delta = \{\mathbf{h} \in \mathbb{R}_+^r | \mathbf{h}^\top \mathbf{1}_r \leq 1\}, \qquad (3)$$

where $\mathbf{h}_j$ is the jth column of $\mathbf{H}$ and Δ is the r-dimensional unit simplex that encodes the non-negativity and sum-to-one constraints in (2). Assuming $\text{rank}(\mathbf{W}) = r$ which is a standard assumption, this least squares problem over the unit simplex is a convex problem with strongly convex objective function. We use the accelerated projected gradient (APG) method from Nesterov [17] which requires $\mathcal{O}\left(\sqrt{\kappa}\log\frac{1}{\epsilon}\right)$ iterations to reach an ϵ-accurate solution, where κ is the condition number of $\mathbf{W}^\top\mathbf{W}$ which is the Hessian of the objective function in (3). The convergence rate of APG is optimal as no other first-order method can have a faster convergence rate [17]. We defer the explanation of the acceleration scheme to section III-C. To

compute the projection onto Δ, we use the implementation from [8] requiring $\mathcal{O}(r \log r)$ operations, and which uses the fact that the projection can be written as $P_\Delta(\mathbf{h}) = [\mathbf{h} - l\mathbf{1}_r]_+$ where l is a Lagrangian multiplier. Since r is small (usually $r \leq 20$), this implementation is numerically as good as the optimal method [5] with complexity $\mathcal{O}(r)$.

In summary, we solve (3) using an optimal first-order method. Note that (3) is parallelizable, we can solve the n problems (3) in parallel.

B. Subproblem for $\mathbf{W}$ with V_{det}

We follow the idea from [24] and perform a block coordinate descent method on the columns $\mathbf{w}_i$ of $\mathbf{W}$ ($1 \leq i \leq r$). We have

$$\|\mathbf{X} - \mathbf{W}\mathbf{H}\|_F^2 = \|\mathbf{h}^i\|_2^2 \|\mathbf{w}_i\|_2^2 - 2\langle \mathbf{X}_i \mathbf{h}^{i\top}, \mathbf{w}_i \rangle + c, \quad (4)$$

$$\det(\mathbf{W}^\top \mathbf{W}) = \gamma_i \mathbf{w}_i^\top \mathbf{Q}_i \mathbf{Q}_i^\top \mathbf{w}_i. \quad (5)$$

In (4), $\mathbf{X}_i = \mathbf{X} - \sum_{j \neq i} \mathbf{w}_j \mathbf{h}^j$ and c is a constant independent of $\mathbf{w}_i$. In (5), $\gamma_i = \det(\mathbf{W}_i^\top \mathbf{W}_i)$ and $\mathbf{Q}_i$ is the orthonormal basis of the null space of $\mathbf{W}_i^\top$, where $\mathbf{W}_i$ is $\mathbf{W}$ without the column $\mathbf{w}_i$; see [24, Appendix] for more details. Using (4) and (5), we obtain the following problem for each column of $\mathbf{W}$:

$$\min_{\mathbf{w}_i \geq 0} \frac{1}{2} \mathbf{w}_i^\top \big(\|\mathbf{h}^i\|_2^2 \mathbf{I}_m + \gamma_i \mathbf{Q}_i \mathbf{Q}_i^\top \big) \mathbf{w}_i - \langle \mathbf{X}_i \mathbf{h}^{i\top}, \mathbf{w}_i \rangle.$$

Unlike the problem on $\mathbf{h}$, here the objective function in $\mathbf{w}_i$ depends on the other columns of $\mathbf{W}$. We solve this quadratic program with nonnegativity constraints using APG, which is faster than the standard quadratic programming algorithms used in [24].

C. Subproblem for $\mathbf{W}$ with V_{logdet}

To solve for V_{logdet}, we use *majorization minimization*, similarly as in [7]. First we have

$$(\text{Lemma 2, [12]}) \quad \log\det(\mathbf{W}^\top \mathbf{W} + \delta \mathbf{I}_r) \leq \|\mathbf{W}\|_{\mathbf{D}}^2 + c, \quad (6)$$

where $\|\mathbf{W}\|_{\mathbf{D}} = \|\mathbf{D}^{\frac{1}{2}} \mathbf{W}^\top\|_F$ is a weighted norm with $\mathbf{D} = (\mathbf{Y}^\top \mathbf{Y} + \delta \mathbf{I}_r)^{-1} \succ 0$ for any matrix $\mathbf{Y}$. This expression (6) comes from performing the first-order Taylor expansion of the concave function $\log\det(.)$. The inequality (6) holds when $\mathbf{Y} = \mathbf{W}$. In other words, we minimize the *tight convex upper bound* on the non-convex logdet function

$$\min_{\mathbf{W}^t \geq 0} \Phi(\mathbf{W}) = \frac{1}{2} \langle \mathbf{W}^{t\top} \mathbf{W}^t, \mathbf{H}\mathbf{H}^\top \rangle - \langle \mathbf{X}, \mathbf{W}^t \mathbf{H} \rangle + \frac{\lambda}{2} \|\mathbf{W}^t\|_{\mathbf{D}^t}^2, \quad (7)$$

where $\mathbf{D}^t = \Big(\mathbf{W}^{t-1\top} \mathbf{W}^{t-1} + \delta \mathbf{I}_r \Big)^{-1}$. To solve (7), we again use APG. However, we embed the following two acceleration strategies to APG:

- The adaptive restart heuristic from [18]. The APG update of $\mathbf{W}$ at iteration t can be expressed as

  $$\mathbf{W}^{t+1} = [\mathbf{W}^t - \alpha_t \nabla \Phi(\mathbf{W}^t + \beta_t \Delta_t) + \beta_t \Delta_t]_+,$$

  where α is the step size, $\nabla\Phi$ is the gradient of the objective function and $[\,\cdot\,]_+$ is the projection onto the nonnegative orthant. Here $\beta_t \Delta_t$ is the momentum term added in Nesterov's acceleration that extrapolates $\mathbf{W}^t$ along the direction $\Delta_t = \mathbf{W}^t - \mathbf{W}^{t-1}$ and $\beta_t \in [0, 1]$ is a parameter with $\beta_0 = 0$. The extrapolation may increase the value of Φ, and when this happens, we "restart" the APG scheme by reinitializing β_t, which reduces the APG back to a standard projected gradient step that decreases the value of Φ.
- The general acceleration framework for NMF algroithms from [10]. There are several terms independent of $\mathbf{W}$, in particular $\mathbf{H}\mathbf{H}^\top$ and $\mathbf{X}\mathbf{H}^\top$, in the gradient term and are the main computational cost. The idea has two components: (i) pre-compute the terms independent of $\mathbf{W}$ outside the update to avoid repeated computation of the same terms, and (ii) perform the update on $\mathbf{W}$ multiple times to reuse these precomputed terms (in most standard NMF algorithms, $\mathbf{W}$ is only updated once before the update of $\mathbf{H}$).

Finally, due to space limit, we do not present another majorization for logdet which is based on eigenvalue inequality proposed in the conference work [1]. In short, that approach is another relaxation of (7) that unlocks the coupling between columns of $\mathbf{W}$ so that column-wise update similar to the one mentioned in section III-B can be used.

D. Subproblem for $\mathbf{W}$ with V_*

The nuclear norm regularized VRNMF problem on $\mathbf{W}$ is

$$\min_{\mathbf{W} \geq 0} \frac{1}{2} \langle \mathbf{W}^\top \mathbf{W}, \mathbf{H}\mathbf{H}^\top \rangle - \langle \mathbf{X}, \mathbf{W}\mathbf{H} \rangle + \lambda \|\mathbf{W}\|_*. \quad (8)$$

Ignoring for now the non-negativity constraints, we solve the resulting problem by proximal gradient which updates $\mathbf{W}$ as follows:

$$\mathbf{W}^{t+\frac{2}{3}} = \text{prox}_{\theta\|\,\|_*} \Big\{ \mathbf{W}^{t+\frac{1}{3}} \Big\}, \; \mathbf{W}^{t+\frac{1}{3}} := \mathbf{W}^t - \alpha \nabla f(\mathbf{W}^t),$$

where α is step size and ∇f is the gradient of $\frac{1}{2}\langle \mathbf{W}^\top \mathbf{W}, \mathbf{H}\mathbf{H}^\top \rangle - \langle \mathbf{X}, \mathbf{W}\mathbf{H} \rangle$. For the step size, we use $\alpha = 1/L$ where $L = \|\mathbf{H}\mathbf{H}^\top\|_2$ is the Lipschitz constant of the gradient. The proximal operator of $\|\cdot\|_*$ with step size θ has the closed-form expression given by the *singular value thresholding* (SVT) operator [3], we have

$$\mathbf{W}^{t+\frac{2}{3}} = \text{SVT}_\theta \Big\{ \mathbf{W}^{t+\frac{1}{3}} \Big\} := \mathbf{U}[\Sigma - \theta \mathbf{I}]_+ \mathbf{V}^\top,$$

where $\mathbf{U}\Sigma\mathbf{V}^\top$ is the SVD of $\mathbf{W}^{t+\frac{1}{3}}$. Finally, to take the nonnegativity constraint into account, we simply put a nonnegative projection after SVT and set $\mathbf{W}^{t+1} = [\mathbf{W}^{t+\frac{2}{3}}]_+$.

E. Summary of the algorithms

Algorithm 1 shows the general framework for solving VRNMF. Our implementations are faster than previous works because of the use of Nesterov's acceleration [17], adaptive restart [18] and the use of the acceleration strategy for NMF from [10]. Moreover, our update of $\mathbf{W}$ in VRNMF using V_* is new.

We refer to the algorithm using V_{det}, V_{logdet} and V_* as `Det`, `logdet` and `Nuclear`, respectively.

Algorithm 1 VRNMF

Input: $\mathbf{X} \in \mathbb{R}^{m\times n}$, an integer r, $\lambda \geq 0$
Output: $\mathbf{W} \in \mathbb{R}_+^{m\times r}$, $\mathbf{H} \in \mathbb{R}_+^{r\times n}$ for problem (2).
Initialize $(\mathbf{W}^0, \mathbf{H}^0)$ by SPA [11]
for $t = 1, 2, \ldots$ **do**
  Update of $\mathbf{W}$
  Compute and store $\mathbf{H}\mathbf{H}^\top$ and $\mathbf{X}\mathbf{H}^\top$
  if $V = V_{\text{det}}$ **then**
    Update $\mathbf{W}$ as stated in section III-B.
  else if $V = V_{\text{logdet}}$ **then**
    Update $\mathbf{W}$ as stated in section III-C.
    Update $\mathbf{D} = (\mathbf{W}^\top\mathbf{W} + \delta\mathbf{I}_r)^{-1}$.
  else if $V = V_*$ **then**
    Update $\mathbf{W}$ as stated in section III-D.
  end if
  Update of $\mathbf{H}$ by APG [8].
end for

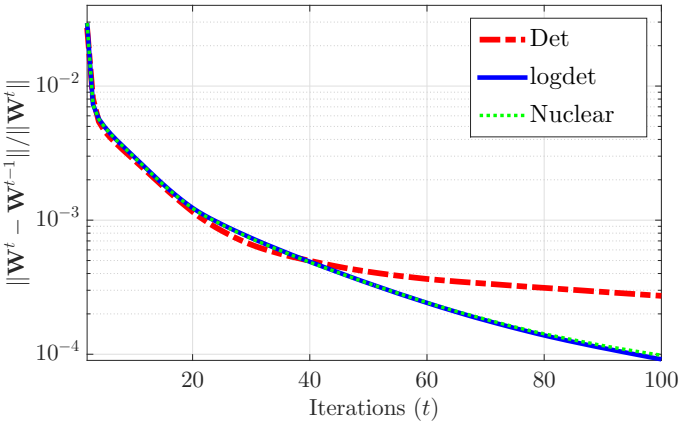


Fig. 2. The typical convergence curve of the algorithms `Det`, `logdet` and `Nuclear`.

We end this section by briefly mentioning the convergence of these algorithms. VRNMF is a non-convex problem and it can be shown that the sequence $\{\mathbf{W}^t, \mathbf{H}^t\}_{t\geq 1}$ produced by Algorithm 1 converges to a first-order stationary point: For `Det`, the convergence comes from the standard result of coordinate descent [23]. For `logdet`, which involves the use of the upper bound Φ', convergence comes from the theory of [19]. For `Nuclear`, convergence result of proximal gradient applies [3]. Figure 2 shows a typical convergence curve of the algorithms.

IV. Experiments

In this section we compare the VRNMF models on solving HU. We first describe the general experimental setting in section IV-A and then report the experimental results on recovering the ground truth $\mathbf{W}^{\text{true}}$ under different asymmetric non-separability and different noise levels in the subsequent sub-sections. Finally we present results on two real-world data sets. All the experiments are run with `MATLAB` (v.2015a) on a laptop with 2.4GHz CPU and 16GB RAM. The codes available from https://angms.science/research.html.

A. Settings

a) Data generation: For synthetic experiments, we generate the observed data matrix as $\mathbf{X} = [\mathbf{X}^{\text{clean}} + \mathbf{N}]_+$ with white Gaussian noise $\mathbf{N} \in \mathbb{R}^{m\times n}$ with zero mean and variance $\sigma \geq 0$. We generate $\mathbf{X}^{\text{clean}} = \mathbf{W}^{\text{true}}\mathbf{H}^{\text{true}}$, where $\mathbf{W}^{\text{true}}$ comes from several datasets available from http://lesun.weebly.com/hyperspectral-data-set.html [25] (unlike the conference version [1] that generated $\mathbf{W}^{\text{true}}$ at random); see Figure 3. We generate each column of $\mathbf{H} \in \mathbb{R}_+^{r\times n}$ using the Dirichlet distribution of parameter 0.1. If $\mathbf{p} = \mathbf{1}_r$, that is, separability condition holds, we take $\mathbf{H}$ as $\mathbf{H}^{\text{true}}$. Otherwise, we remove the columns of $\mathbf{H}$ with at least one element in the jth coordinate that exceeds the value p_j, and resample again until $\mathbf{H}$ satisfies the condition $\|\mathbf{h}^j\|_\infty \leq p_j$ for all j. In this paper, we will use $p_j \in (0.5, 0.99]$ for all j. In all the experiments, we set the number of data points $n = 1000$ and the maximum number of iterations to 300.

For experiments on real data, we take the real data as $\mathbf{X}$, without any preliminary dimension reduction, and without any pre-processing to suppress noise or to remove outliers.

b) Parameters of the algorithms: We will compare our three proposed algorithms `Det`, `logdet` and `Nuclear` with SPA [11] and `RVolMin` which is a state-of-the-art volume-regularized method [7]. For `RVolMin`, we use the same data fitting term (namely, the Frobenius norm), the same number of iterations, the same parameter search scheme and the same initialization as for our algorithms.

Given an observed data matrix $\mathbf{X}$, all VRNMF algorithms have two main parameters: r and λ. They also require an initialization $(\mathbf{W}^{\text{ini}}, \mathbf{H}^{\text{ini}})$. We assume r is known. We generate $\mathbf{W}^{\text{ini}}$ from $\mathbf{X}$ using SPA [11], and generate $\mathbf{H}^{\text{ini}}$ using the method described in Section III-A. The regularization parameter λ should usually be chosen small. In fact, a large λ forces the vertices of the convex hull of $\mathbf{W}$ to be very close to each other, making $\mathbf{W}$ ill-conditioned and/or rank-deficient. In particular, for VRNMF with V_*, the SVT operator set singular values smaller than λ to zero, so a large λ makes $\mathbf{W}$ rank-deficient. The following describes how we tune λ. The goal here is to tune λ so that each algorithm performs as best as possible for the considered problems. To achieve this goal, we use the ground truth $\mathbf{W}^{\text{true}}$. We set $\lambda = \tilde{\lambda}\frac{f(\mathbf{W}^{\text{ini}}, \mathbf{H}^{\text{ini}})}{|V(\mathbf{W}^{\text{ini}})|}$, where $(\mathbf{W}^{\text{ini}}, \mathbf{H}^{\text{ini}})$ are the initial solutions obtained by SPA, and $\tilde{\lambda}$ is a tuning variable within the search interval $\mathcal{I}_0 = [10^{-6}, 0.5]$. We perform grid search by bisection in a greedy way to tune $\tilde{\lambda}$:

- (step-1) Take $\tilde{\lambda}_a = 10^{-6}$, $\tilde{\lambda}_b = 0.5$ initially.
- (step 2) Run the algorithm with $\tilde{\lambda}_a$, $\tilde{\lambda}_b$ and $\tilde{\lambda}_c$ where $\tilde{\lambda}_c = 0.5(\tilde{\lambda}_a + \tilde{\lambda}_b)$. Denote the performance of the solution $\mathbf{W}$ produced under a specific $\tilde{\lambda}$ as MRSA($\tilde{\lambda}$), where MRSA will be defined in the next paragraph.
- (step-3) Split the search interval $\mathcal{I}_0 = [\tilde{\lambda}_a, \tilde{\lambda}_b]$ into two intervals $\mathcal{I}_0^1 = [\tilde{\lambda}_a, \tilde{\lambda}_c]$ and $\mathcal{I}_0^2 = [\tilde{\lambda}_c, \tilde{\lambda}_b]$. For each interval there are two MRSA values, we let the MRSA value of an interval be the sum of these values.
- (step-4) We repeat step-1 to step-3 on the interval with the lowest MRSA value. That is, at iteration k, we shrink the

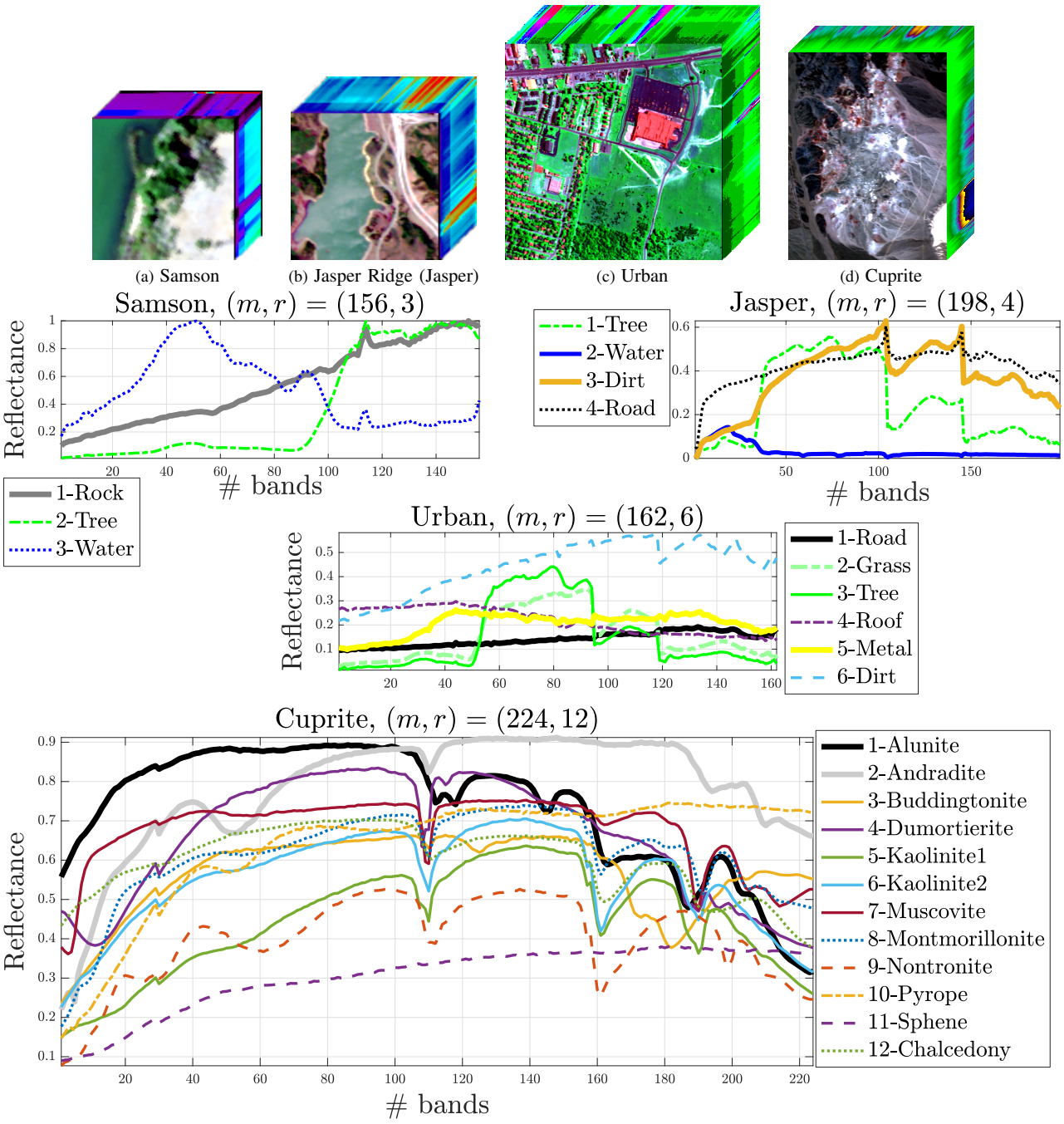


Fig. 3. **Top row** The datasets from [25]. **Bottom subplots** The endmembers $\mathbf{W}^{\text{true}}$ of the datasets. Best viewed in color.

search interval by half by defining the new search interval $\mathcal{I}_{k+1}$ as

$$\mathcal{I}_{k+1} \leftarrow \min_{\mathcal{I}} \left\{ \text{MRSA}(\mathcal{I}_k^1), \text{MRSA}(\mathcal{I}_k^2) \right\}.$$

- If a draw happens in step-4, we perform two bisections on each of the interval $\mathcal{I}_k^1$ and $\mathcal{I}_k^2$ and set the next interval to be the one with the smallest MRSA.
- We repeat this process (step-1 to step-4) at most 20 times, or if the improvement from one iteration to the next is negligible, namely if

$$\left| \text{MRSA}\left(\tilde{\lambda}_{k+1}\right) - \text{MRSA}\left(\tilde{\lambda}_k\right) \right| \leq 10^{-4}.$$

Note that such greedy bisection search does not guarantee that $\tilde{\lambda}_k$ will converge to the best value, that is, the value that corresponds to the lowest MRSA. However, we have observed in extensive numerical experiments the effectiveness of this scheme, which will be illustrated in IV-B.

c) Performance metric: We measure the quality of a solution $\mathbf{W}$ produced by VRNMF algorithms using the *mean removed spectral angle* (MRSA) between $\mathbf{W}$ and $\mathbf{W}^{\text{true}}$.

TABLE I
COMPARISON OF MRSA VALUES OF THE BISECTION AND THE GRID SEARCH TO OBTAIN λ^b AND λ^*, RESPECTIVELY. THE SECOND COLUMN IS THE NUMBER OF ITERATIONS NEEDED FOR THE BISECTION SEARCH TO TERMINATE.

p	$\frac{\text{MRSA}(\lambda^b) - \text{MRSA}(\lambda^*)}{\lvert\text{MRSA}(\lambda^*)\rvert}$	# of iterations
0.93	-0.0016±0.0002	2±0.00
0.89	-0.0143±0.0023	6.4±0.52
0.86	-0.0235±0.0029	7.9±0.32
0.83	-0.0429±0.0032	9±0.00
0.79	-0.0815±0.0089	11.1±0.32
0.76	-0.0419±0.0032	10.5±0.53

MRSA between two vectors $\mathbf{x}, \mathbf{y} \in \mathbb{R}^m \setminus \{\mathbf{0}_m\}$ is defined as

$$\frac{100}{\pi} \cos^{-1}\left(\frac{\langle \mathbf{x} - \bar{\mathbf{x}}, \mathbf{y} - \bar{\mathbf{y}} \rangle}{\|\mathbf{x} - \bar{\mathbf{x}}\|_2 \|\mathbf{y} - \bar{\mathbf{y}}\|_2}\right) \in [0, 100]. \quad (9)$$

MRSA gives a better measurement than relative percentage error as it purely depends on the shapes of $\mathbf{x}$ and $\mathbf{y}$, where the effects of shifts and scaling are removed. A low MRSA value means a good matching between $\mathbf{x}$ and $\mathbf{y}$. We measure the performance of algorithms by calculating the MRSA between $\{\mathbf{w}_i\}_{i\in[r]}$ and $\{\mathbf{w}_i^{\text{true}}\}_{i\in[r]}$, which is defined as the mean of MRSA between each vector pair $\{\mathbf{w}_i, \mathbf{w}_i^{\text{true}}\}$, such value is within [0, 100].

B. Effectiveness of the bisection search on λ

In this section, we illustrate the effectiveness of our tuning strategy for λ. We perform experiments on the synthetic dataset where $\mathbf{W}^{\text{true}}$ comes from the Samson dataset with $r = 3$ as follows. We consider 6 different values of the symmetric non-separability vector $\mathbf{p} = [p_1, p_2, p_3]$ where $p_1 = p_2 = p_3$ are selected from the set $\{0.93, 0.89, 0.86, 0.83, 0.79, 0.76\}$. We run `Det` with two parameter tuning schemes: (1) the bisection search mentioned in IV-A, and (2) a brute-force grid search: we run `Det` with all 100 equally spaced steps values of $\tilde{\lambda}$ in the interval $\mathcal{I}_0 = [10^{-6}, 0.5]$.

For each value of $\mathbf{p}$, 10 data sets are generated randomly. Let us denote λ^b the value of λ obtained by the bisection search and λ^* by the grid search. Table I shows the results between comparing the bisection search and the grid search displayed as $\frac{\text{MRSA}(\lambda^b) - \text{MRSA}(\lambda^*)}{\lvert\text{MRSA}(\lambda^*)\rvert}$ in the format of (mean±std) and the average number of iterations performed by the bisection search to reach λ^b. From Table I, we make the following two interesting observations:

1) As the purity goes down, more bisections are need to identify the best λ. This was expected since, for a high purity, the initialization (SPA) provides a good initial solution.
2) In all 50 cases, the value of λ^b lead to a slightly smaller MRSA value than λ^* while requiring less computations. This illustrates the fact that the bisection is able to identify the right value of λ and to refine the search around that value with more precision than an expensive exhaustive search.

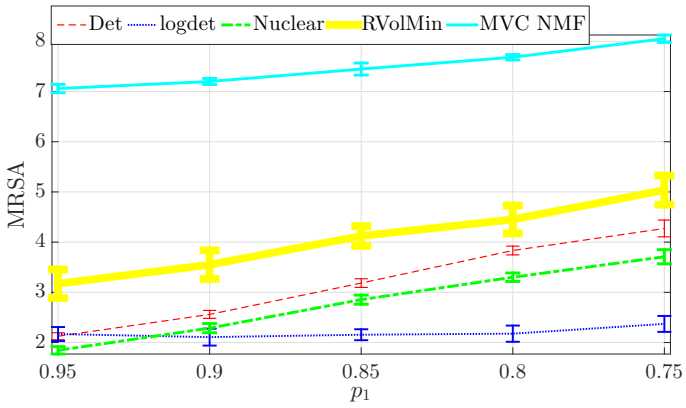


Fig. 4. MRSA curves of `Det`, `logdet`, `Nuclear`, `RVolMin` and `MVC` across different values of p_1 ($p_2 = 0.79$, $p_3 = 0.69$) for synthetic Samson data sets. Best viewed in color.

C. Comparison with MVC-NMF

Minimum volume constrained nonnegative matrix factorization (MVC-NMF) is one of the first minimum-volume NMF algorithm[1] [15]. Let us run a small experiment to show that MVC-NMF does not compete with `Det`, `logdet` and `Nuclear` and `RVolMin` [7]. Similarly as in the previous section, we use synthetic datasets where $\mathbf{W}^{\text{true}}$ comes from the Samson dataset with $r = 3$. However, we use more difficult scenarios using the non-separability vectors $\mathbf{p} = [p_1, p_2, p_3]$ where p_1 is selected from $\{0.95, 0.9, 0.85, 0.8, 0.75\}$, $p_2 = 0.79$, and $p_3 = 0.69$. For each value of $\mathbf{p}$, 10 data sets are generated randomly. All algorithms take the same initialization (SPA), the same number of iterations (100) and the regularization parameter λ is tuned using the bisection search. Figure 4 shows that `MVC-NMF` produces significantly worse results than the four other algorithms, hence will not be considered in the extensive numerical experiments the subsequent comparisons. It is difficult to pinpoint the reasons of the poor results of MVC-NMF; we see at least two of them: (1) MVC-NM uses another regularizer which is the squared volume of the convex hull of the columns of $\mathbf{W}$ without the origin, and (2) it does not use optimal optimization methods (for example, it uses a fixed step size of 0.001).

D. Using the Samson dataset with r = 3

Now we run more comprehensive experiments on the synthetic dataset where $\mathbf{W}^{\text{true}}$ comes from the Samson data set with $r = 3$ as follows. We perform $10 \times 10 \times 10 = 1000$ experiments where the data in each experiment is generated using the non-separability vector $\mathbf{p} = [p_1, p_2, p_3]$, where p_i is selected from the set $\mathbb{P} = \{0.99, 0.96, 0.93, 0.89, 0.86, 0.83, 0.79, 0.76, 0.73, 0.69\}$. Figure 5 shows the results in form of MRSA cubes for the algorithms `SPA`, `Det`, `logdet`, `Nuclear` and `RVolMin`, where each pixel in the cube is the result (in MRSA) over one trial. We then construct the recovery curves by counting the number of cases (pixels) in the cube that the MRSA value

[1]The code available from https://github.com/aicip/MVCNMF

is less than a threshold; see Figure 6. In general, the results in Figures 5 and 6 show that

- All VRNMF algorithms perform better than the state-of-the-art SNMF algorithm `SPA`, as the MRSA cubes of VRNMF have a wider blue region (lower MRSA) and their recovery curves in Figure 6 dominates that of `SPA`.
- When the data is highly non-separable (low p_is), VRNMF algorithms perform worse than `SPA`. In fact, the region of the cube corresponding to highly non-separable data in `SPA` is not as red as for the VRNMF approaches. The reason is that `SPA` always extract points from the data could hence these points are never too far from the vertices. However, when the data is highly non-separable, VRNMF may generate points further away than the vertices.
- Compared with `RVolMin`, `logdet` and `Nuclear` are consistently better, while `Det` performs similarly.
- `logdet` performs better than `Det` and `Nuclear` for highly non-separable data, as its red region is much more concentrated around the highly non-separable corner of the cube.

In terms of computational time, `Det` takes 2.1±0.2 seconds, `logdet` takes 1.2±0.1 seconds, `Nuclear` takes 1.3±0.2 seconds, and `RVolMin` takes 2.1±0.0 seconds. As expected, due to the inner loop and the computation of $\mathbf{Q}_i$, `Det` is slower than the other algorithms.

E. Using the Jasper dataset with r = 4

In Figure 7, we perform the same experiment as in the previous section on the dataset Jasper with $r = 4$, where we fix $p_4 = 0.75$. Furthermore, Table II shows the result in MRSA on the dataset Jasper across three sets of predefined $\mathbf{p}$ values (highly separable with $\mathbf{p}_{\text{high}} = [0.9, 0.8, 0.7, 0.6]$, less separable with $\mathbf{p}_{\text{mid}} = [0.8, 0.7, 0.6, 0.51]$, and even less separable with $\mathbf{p}_{\text{low}} = [0.7, 0.65, 0.55, 0.51]$) and three noise levels ($\sigma = 0.001, 0.005$ and 0.01). Each item in the table (in mean±std) is the average over 20 trails.

We observe the following

- Figure 7 show that VRNMF algorithms are competitive with the state-of-the-art minimum volume based method `RVolMin`, where `Det` performs slightly better than `RVolMin` while `logdet` is significantly better.
- Tables II shows that all the methods improve the fitting accuracy of SPA, with `Det` has the best performance in all cases and `RVolMin` has the worst performances in all cases.

In terms of computational time, `Det` takes 6.62±0.5 seconds, `logdet` takes 2.22±0.2 seconds, `Nuclear` takes 1.45±0.0 seconds, and `RVolMin` takes 2.7±0.0 seconds. Hence, for the same reasons as in the previous experiment, `Det` is slower.

F. Using the Urban dataset with r = 6

Table III shows the result of the same experiment as in the previous section performed with the Urban dataset. Here $\mathbf{p}_{\text{high}} = [0.9, 0.75, 0.7, 0.65, 0.8, 0.85]$, $\mathbf{p}_{\text{mid}} = [0.8, 0.7, 0.65, 0.6, 0.75, 0.8]$, and $\mathbf{p}_{\text{low}} = [0.7, 0.6, 0.55, 0.51, 0.65, 0.7]$. The noise levels are $\sigma = 0.001, 0.005$ and 0.01.

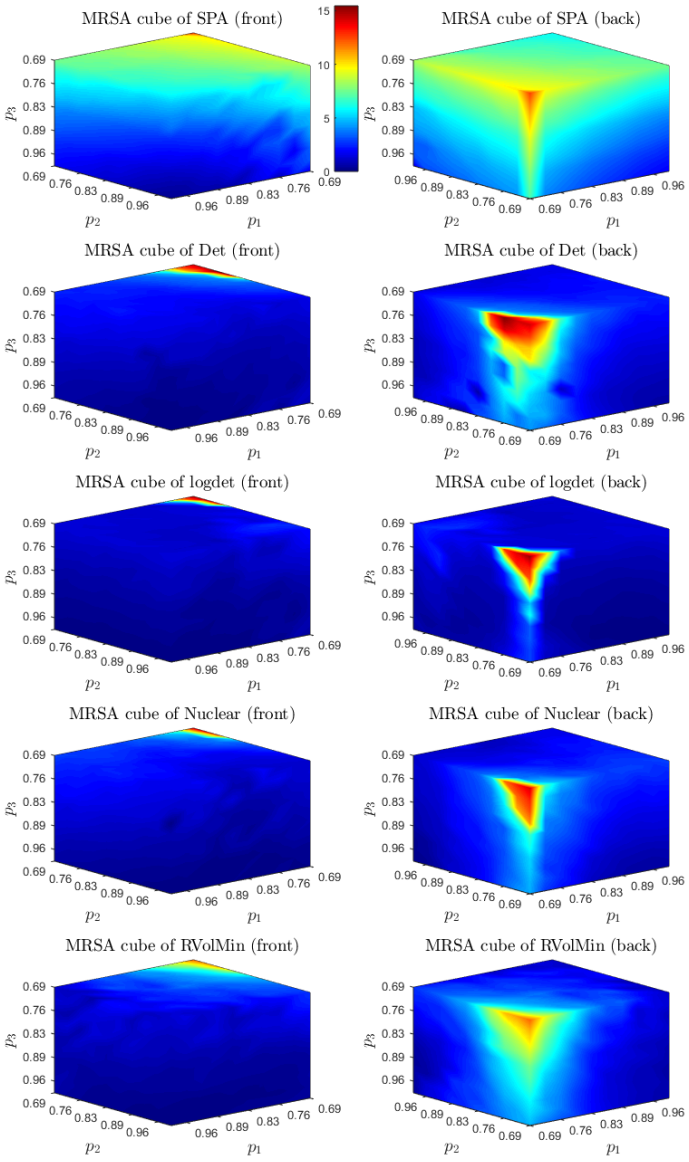


Fig. 5. MRSA cubes for the different NMF algorithms viewed from two different angles, for the synthetic Samson data sets with $r = 3$. All the cubes share the same color scheme. In general, MRSA cubes for VRNMF have a wider blue region than that of `SPA`. However, when the vector $\mathbf{p}$ has a low value, VRNMF gives less accurate fitting than `SPA` (the red regions). VRNMF with `logdet` provides the best results (widest dark blue region). Best viewed in color.

The results show that

- All the methods improve the fitting accuracy of SPA.
- `Det` has the best performance in most cases, with `logdet` as the first-runner up.
- `Det` performs well for $\mathbf{p}_{\text{high}}$ and $\mathbf{p}_{\text{mid}}$. With $\mathbf{p}_{\text{low}}$, it is not as good as `logdet`.
- `Nuclear` and `RVolMin` have the worst performances in all cases.
- The computational times are: `Det` 7.5±0.2, `logdet` 1.8±0.1, `Nuclear` 1.5±0.0 and `RVolMin` 2.3±0.0,

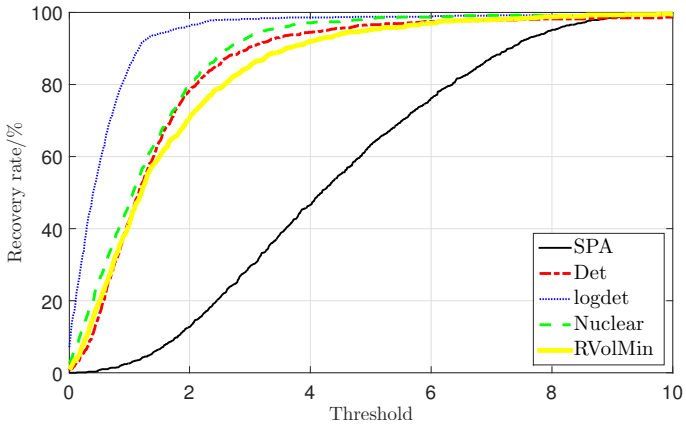


Fig. 6. Curves of recovery corresponding to Figure 5 for the synthetic Samson data sets with $r = 3$.

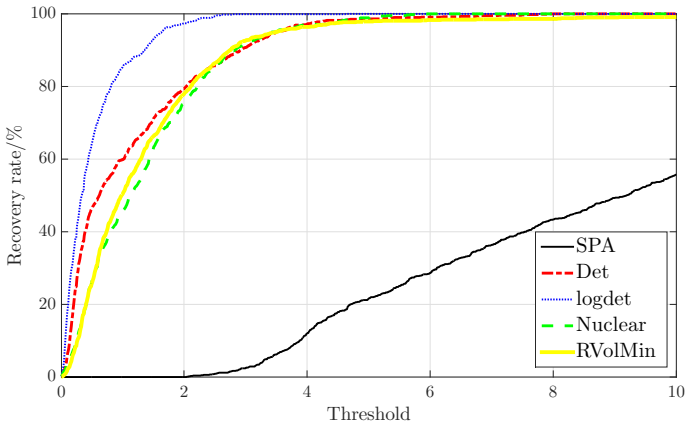


Fig. 7. Curves of recovery corresponds for the dataset Jasper with $r = 4$. Same experimental set up as the one in Figure 6, but where p_4 fixed at 0.75.

TABLE II
MRSA VALUES FOR THE JASPER DATASET WITH $r = 4$

Across different $\mathbf{p}$ ($\sigma = 0.001$)

Method	$\mathbf{p}_{\text{high}}$	$\mathbf{p}_{\text{mid}}$	$\mathbf{p}_{\text{low}}$
SPA	5.40±0.60	12.62±0.18	20.76±0.23
Det	**0.41±0.08**	**0.40±0.06**	**10.99±1.68**
logdet	0.48±0.54	3.03±0.28	12.57±1.49
Nuclear	0.64±0.07	2.12±0.13	19.90±1.40
RVolMin	1.04±0.38	4.99±0.22	13.67±2.99

Across different noise levels ($\mathbf{p} = \mathbf{p}_{\text{high}}$)

Method	$\sigma = 0.001$	$\sigma = 0.01$	$\sigma = 0.05$
SPA	5.40±0.60	7.29±0.51	24.59±1.43
Det	**0.41±0.08**	**0.74±0.06**	**4.90±3.27**
Taylor	0.48±0.54	1.40±0.08	9.00±3.88
Nuclear	0.64±0.07	1.23±0.06	6.78±5.78
RVolMin	1.04±0.38	2.31±0.28	10.23±2.05

respectively.

G. On the Cuprite dataset with $r = 12$

Table IV shows the result on the same experiments conducted with the Cuprite dataset. Here we concatenate the $\mathbf{p}$ vector used in Urban two times to form the $\mathbf{p}$ vector with

TABLE III
MRSA VALUES FOR THE URBAN DATASET WITH $r = 6$

Across different $\mathbf{p}$ ($\sigma = 0.001$)

Method	$\mathbf{p}_{\text{high}}$	$\mathbf{p}_{\text{mid}}$	$\mathbf{p}_{\text{low}}$
SPA	7.83±0.93	10.32±1.53	16.22±2.00
Det	**0.54±0.11**	**2.45±1.25**	10.08±5.71
logdet	1.27±0.68	3.09±2.04	**8.78±2.58**
Nuclear	3.79±0.62	6.39±1.54	13.48±4.53
RVolMin	3.03±0.90	5.64±1.29	13.05±4.28

Across different noise levels ($\mathbf{p} = \mathbf{p}_{\text{high}}$)

Method	$\sigma = 0.001$	$\sigma = 0.005$	$\sigma = 0.01$
SPA	7.83±0.93	8.56±0.86	15.95±3.57
Det	**0.54±0.11**	**1.25±0.48**	**6.85±3.47**
logdet	1.27±0.68	4.41±0.93	9.85±3.36
Nuclear	3.79±0.62	4.58±0.95	11.75±3.35
RVolMin	3.03±0.9	4.95±1.01	15.32±9.35

TABLE IV
MRSA VALUES FOR THE CUPRITE DATASET WITH $r = 12$

Across different $\mathbf{p}$ ($\sigma = 0.001$)

Method	$\mathbf{p}_{\text{high}}$	$\mathbf{p}_{\text{mid}}$	$\mathbf{p}_{\text{low}}$
SPA	6.59±0.98	8.87±1.42	11.53±1.39
Det	2.59±0.74	4.40±1.51	9.71±2.43
logdet	**2.51±0.59**	**3.85±0.96**	**8.41±2.04**
Nuclear	2.55±0.70	4.33±1.56	10.07±2.73
RVolMin	3.72±2.41	5.39±2.29	10.82±3.31

Across different noise levels ($\mathbf{p} = \mathbf{p}_{\text{high}}$)

Method	$\sigma = 0.001$	$\sigma = 0.005$	$\sigma = 0.01$
SPA	6.59±0.98	7.24±1.07	11.53±1.39
Det	2.59±0.74	4.34±1.74	9.71±2.43
logdet	**2.51±0.59**	**4.01±1.22**	**8.41±2.04**
Nuclear	2.55±0.70	4.38±1.72	10.07±2.73
RVolMin	3.72±2.41	4.66±1.73	10.82±3.31

length 12 for the data Cuprite. For example, for $\mathbf{p}_{\text{high}}^{\text{Cuprite}}$, we use $\mathbf{p}_{\text{high}}^{\text{Cuprite}} = [\mathbf{p}_{\text{high}}^{\text{Urban}}\ \mathbf{p}_{\text{high}}^{\text{Urban}}]$.

The result from the two tables show that

- All the methods improve the fitting accuracy of SPA.
- `logdet` has the best performance in all situations, with `Det` and `Nuclear` as the first-runner ups.
- `RVolMin` has the worst performances in most cases.

In terms of computational time, `Det` takes 28.1±0.9 seconds, `logdet` takes 3.8±1.1 seconds, `Nuclear` takes 3.4±0.1 seconds, and `RVolMin` takes 3.2±0.0 seconds. As expected, when r increase, `Det` takes much more time and it is not recommended for large r.

In general, the results show that the VRNMF algorithms with `Det` and `logdet` performs very well, better than `RVolMin` and `Nuclear` in terms of fitting accuracy. As `RVolMin` consistently produce inferior results, we do not include it in the subsequent sections.

H. On image segmentation on real data

In this section we run the algorithms `Det`, `logdet` and `Nuclear` on real HU data Samson and Jasper. As stated in section IV-A, no pre-processing is performed on the raw

TABLE V
NUMERICAL RESULTS ON SAMSON AND JASPER DATASETS.

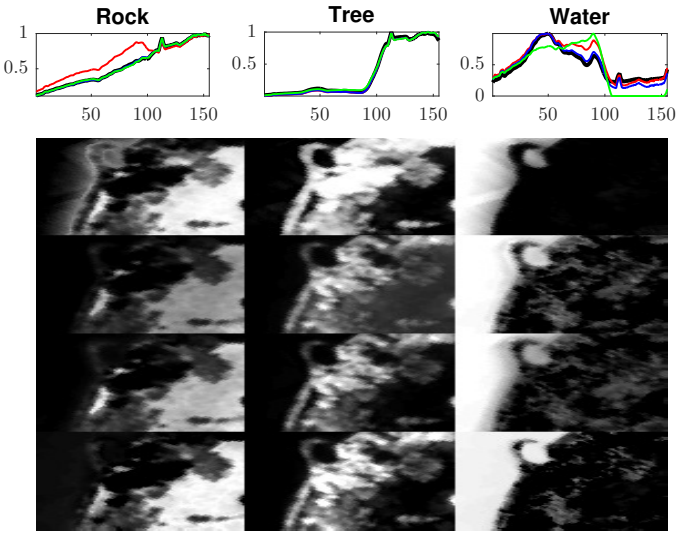
Method	Time (s.)	MRSA	$\frac{\|\mathbf{X}-\mathbf{WH}\|_F}{\|\mathbf{X}\|_F}$
Samson			
Det	8.53	7.13	2.86%
logdet	**6.22**	**2.58**	**2.69%**
Nuclear	8.94	6.99	7.13%
Jasper			
Det	15.13	**5.51**	**4.14%**
logdet	12.67	6.03	6.09%
Nuclear	**12.43**	8.96	4.51%

data and we use the raw data directly. The following states the specifications of the datasets. For the dataset Samson, $(m, n, r) = (156, 95^2, 3)$. For the dataset Jasper, $(m, n, r) = (198, 100^2, 4)$. We have tuned λ in the same way as for the synthetic datasets using the endmembers $\mathbf{W}^{\text{Ref}}$ from [25]. Figure 8 shows the decompositions. In the same figure, we also show the references provided by [25], and we list the numerical results in Table V. In Table V, the MRSA is calculated with respect to the reference $\mathbf{W}^{\text{Ref}}$ of [25]. The results show that all three VRNMF algorithms produce meaningful decomposition.

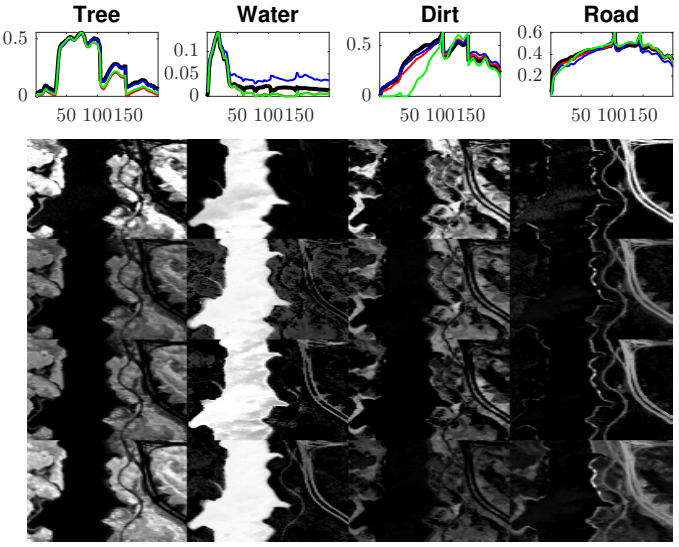
Remark The reference [25] used a sparsity regularization on $\mathbf{H}$ and hence the abundance map of [25] looks cleaner. It is out of the scope of this paper to consider a sparsity regularization on $\mathbf{H}$ as we are focusing on the volume regularization on $\mathbf{W}$. This is a direction for further research.

V. CONCLUSION

In this paper, NMF models with different volume regularizations (VRNMF) were investigated. We have developed highly efficient algorithms for these VRNMF models. The VRNMF algorithms are shown to be able to outperform both the state-of-the-art separable NMF algorithm SPA, and the volume-based methods `MVC-NMF` [15] and `RVolMin` [7]. Furthermore, extensive experimental results using real hyperspectral data showed that, when the data has a small rank r ($r \leq 4$) and is highly separable, the volume regularizer based on the determinant (`Det`) provides the best results, although the the regularizer based on the logarithm of the determinant (`logdet`) provides almost as good decompositions. When the rank increases and/or the data becomes less separable, `logdet` performs best in most cases, while being computationally faster than `Det`. Therefore, in practice, we recommend the use of `logdet`.



(a) Samson - spectral signatures and abundance maps of each component



(b) Jasper - spectral signatures and abundance maps of each component

Fig. 8. The decomposition of the datasets Samson and Jasper. In the subfigures (a) and (b), the upper subplots are the identified endmembers $\mathbf{W}$: $\mathbf{W}^{\text{ref}}$ (black), $\mathbf{W}^{\text{Det}}$ (red), $\mathbf{W}^{\text{logdet}}$ (blue) and $\mathbf{W}^{\text{Nuclear}}$ (green); and the lower subplots are the corresponding abundances matrices $\mathbf{H}$; from the first row to the fourth row: $\mathbf{H}^{\text{ref}}$ (provided by [25]), $\mathbf{H}^{\text{Det}}$, $\mathbf{H}^{\text{logdet}}$ and $\mathbf{H}^{\text{Nuclear}}$. Best viewed in color.

REFERENCES

[1] Ang, M.A., Gillis, N.: Volume regularized non-negative matrix factorisations. In: IEEE WHISPERS (2018)

[2] Bioucas-Dias, J.M., Plaza, A., Camps-Valls, G., Scheunders, P., Nasrabadi, N., Chanussot, J.: Hyperspectral remote sensing data analysis and future challenges. IEEE Geoscience and remote sensing magazine **1**(2), 6–36 (2013)

[3] Cai, J.F., Candès, E.J., Shen, Z.: A singular value thresholding algorithm for matrix completion. SIAM Journal on Optimization **20**(4), 1956–1982 (2010)

[4] Cichocki, A., Zdunek, R., Phan, A.H., Amari, S.i.: Nonnegative matrix and tensor factorizations: applications to exploratory multi-way data analysis and blind source separation. John Wiley & Sons (2009)

[5] Condat, L.: Fast projection onto the simplex and the l_1 ball. Mathematical Programming **158**(1-2), 575–585 (2016)

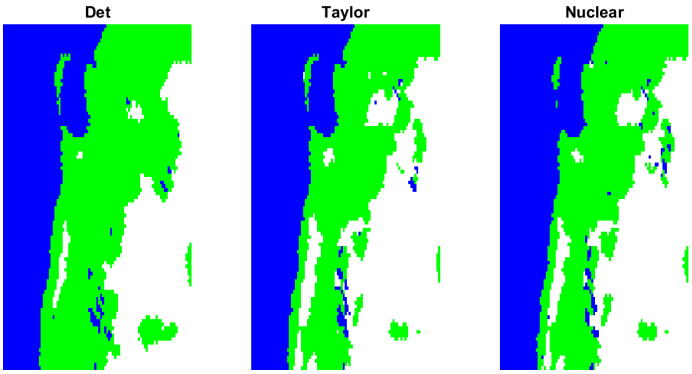
[6] Fu, X., Huang, K., Sidiropoulos, N.D., Ma, W.K.: Nonnegative matrix factorization for signal and data analytics: Identifiability, algorithms, and applications. IEEE Signal Processing Magazine **36**(2), 59–80 (2019)

[7] Fu, X., Huang, K., Yang, B., Ma, W.K., Sidiropoulos, N.D.: Robust volume minimization-based matrix factorization for remote sensing and document clustering. IEEE Transactions on Signal Processing **64**(23), 6254–6268 (2016)

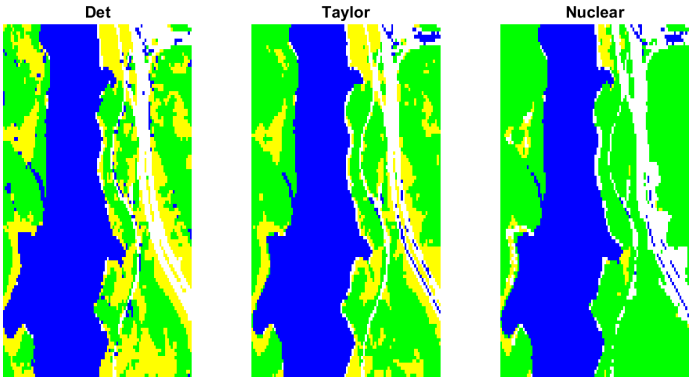
[8] Gillis, N.: Successive nonnegative projection algorithm for robust nonnegative blind source separation. SIAM Journal on Imaging Sciences **7**(2), 1420–1450 (2014)

[9] Gillis, N.: The why and how of nonnegative matrix factorization. Regularization, Optimization, Kernels, and Support Vector Machines **12**, 257–291 (2014)

[10] Gillis, N., Glineur, F.: Accelerated multiplicative updates and hier-



(a) Samson



(b) Jasper

Fig. 9. The distribution maps of the material in the whole scene obtained from the VRNMF algorithms. In (a), the colour code is: white – rock, blue – water, and green – tree. In (b), the colour code is: green – tree, blue – water, yellow – dirt and white – road. The map is constructed by taking the largest component of $\mathbf{h}_j$ as the material of that pixel. As the factorization rank r here is small and these datasets are highly separable, `Det` produces the best segmentation results. For example, the road in Jasper produced by `Det` is better identified than for the other two. Best viewed in color.

archical als algorithms for nonnegative matrix factorization. Neural computation **24**(4), 1085–1105 (2012)
[11] Gillis, N., Vavasis, S.A.: Fast and robust recursive algorithmsfor separable nonnegative matrix factorization. IEEE transactions on pattern analysis and machine intelligence **36**(4), 698–714 (2014)
[12] Jose, J., Prasad, N., Khojastepour, M., Rangarajan, S.: On robust weighted-sum rate maximization in mimo interference networks. In: Communications (ICC), 2011 IEEE International Conference on, pp. 1–6. IEEE (2011)
[13] Leplat, V., Ang, A.M., Gillis, N.: Minimum-volume rank-deficient nonnegative matrix factorizations. In: ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 3402–3406. IEEE (2019)
[14] Ma, W.K., Bioucas-Dias, J.M., Chan, T.H., Gillis, N., Gader, P., Plaza, A.J., Ambikapathi, A., Chi, C.Y.: A signal processing perspective on hyperspectral unmixing: Insights from remote sensing. IEEE Signal Processing Magazine **31**(1), 67–81 (2014)
[15] Miao, L., Qi, H.: Endmember extraction from highly mixed data using minimum volume constrained nonnegative matrix factorization. IEEE Transactions on Geoscience and Remote Sensing **45**(3), 765–777 (2007)
[16] Nascimento, J.M., Dias, J.M.: Vertex component analysis: A fast algorithm to unmix hyperspectral data. IEEE transactions on Geoscience and Remote Sensing **43**(4), 898–910 (2005)
[17] Nesterov, Y.: Introductory lectures on convex optimization: A basic course, vol. 87. Springer Science & Business Media (2013)
[18] Odonoghue, B., Candes, E.: Adaptive restart for accelerated gradient schemes. Foundations of computational mathematics **15**(3), 715–732 (2015)
[19] Razaviyayn, M., Hong, M., Luo, Z.Q.: A unified convergence analysis of block successive minimization methods for nonsmooth optimization. SIAM Journal on Optimization **23**(2), 1126–1153 (2013)
[20] Recht, B., Fazel, M., Parrilo, P.A.: Guaranteed minimum-rank solutions of linear matrix equations via nuclear norm minimization. SIAM review **52**(3), 471–501 (2010)
[21] Schachtner, R., Pöppel, G., Tomé, A.M., Lang, E.W.: Minimum determinant constraint for non-negative matrix factorization. In: International Conference on Independent Component Analysis and Signal Separation, pp. 106–113. Springer (2009)
[22] Vavasis, S.A.: On the complexity of nonnegative matrix factorization. SIAM Journal on Optimization **20**(3), 1364–1377 (2010)
[23] Wright, S.J.: Coordinate descent algorithms. Mathematical Programming **151**(1), 3–34 (2015)
[24] Zhou, G., Xie, S., Yang, Z., Yang, J.M., He, Z.: Minimum-volume-constrained nonnegative matrix factorization: Enhanced ability of learning parts. IEEE Trans. on Neural Networks **22**(10), 1626–1637 (2011)
[25] Zhu, F., Wang, Y., Fan, B., Xiang, S., Meng, G., Pan, C.: Spectral unmixing via data-guided sparsity. IEEE Transactions on Image Processing **23**(12), 5412–5427 (2014)

Andersen Man Shun Ang received the B.Eng and M.Phil degrees in engineering from the University of Hong Kong, Hong Kong, in 2014 and 2016, respectively. He is currently working towards a Ph.D. in the Department of Mathematics and Operational Research, Faculté polytechnique, Université de Mons, Mons, Belgium. His research interests are optimization, matrix factorization and tensor factorization.

Nicolas Gillis received the Master's and Ph.D. degrees in applied mathematics from the Université catholique de Louvain, Louvain-la-Neuve, Belgium, in 2007 and 2011, respectively. He is currently an Associate Professor with the Department of Mathematics and Operational Research, Faculté polytechnique, Université de Mons, Mons, Belgium. His research interests include optimization, numerical linear algebra, machine learning, and data mining.

Dr. Gillis received the Householder award in 2014, and an ERC starting grant in 2015. He is currently serving as an Associate Editor of the IEEE Transactions on Signal Processing and of the SIAM Journal on Matrix Analysis and Applications.