Smoothed Separable Nonnegative Matrix Factorization

Nicolas Nadisic, Nicolas Gillis, Christophe Kervazo

Smoothed Separable Nonnegative Matrix Factorization

Nicolas Nadisic* Nicolas Gillis* Christophe Kervazo†

*Department of Mathematics and Operational Research
Faculté Polytechnique, Université de Mons
Rue de Houdain 9, 7000 Mons, Belgium

†LTCI, Télécom Paris, Institut Polytechnique de Paris
19 Place Marguerite Perey, 91120 Palaiseau, France

Abstract

Given a set of data points belonging to the convex hull of a set of vertices, a key problem in data analysis and machine learning is to estimate these vertices in the presence of noise. Many algorithms have been developed under the assumption that there is at least one nearby data point to each vertex; two of the most widely used ones are vertex component analysis (VCA) and the successive projection algorithm (SPA). This assumption is known as the pure-pixel assumption in blind hyperspectral unmixing, and as the separability assumption in nonnegative matrix factorization. More recently, Bhattacharyya and Kannan (ACM-SIAM Symposium on Discrete Algorithms, 2020) proposed an algorithm for learning a latent simplex (ALLS) that relies on the assumption that there is more than one nearby data point for each vertex. In that scenario, ALLS is probalistically more robust to noise than algorithms based on the separability assumption. In this paper, inspired by ALLS, we propose smoothed VCA (SVCA) and smoothed SPA (SSPA) that generalize VCA and SPA by assuming the presence of several nearby data points to each vertex. We illustrate the effectiveness of SVCA and SSPA over VCA, SPA and ALLS on synthetic data sets, and on the unmixing of hyperspectral images.

Keywords. blind hyperspectral unmixing, pure-pixel search algorithms, latent simplex, simplex-structured matrix factorization, nonnegative matrix factorization, separability

1 Introduction

Given a set of data points within the convex hull of a set of vertices, estimating these vertices in the presence of noise is a key problem in data analysis and machine learning; see below for some examples. This problem can be formulated as follows.

Problem 1. *Given $X = WH + N \in \mathbb{R}^{m\times n}$ where $H \in \mathbb{R}^{r\times n}_+$ is column stochastic and N is the noise, estimate $W \in \mathbb{R}^{m\times r}$.*

*Email: nicolas.{nadisic, gillis}@umons.ac.be. The authors acknowledge the support by the European Research Council (ERC starting grant no 679515), and by the Fonds de la Recherche Scientifique - FNRS and the Fonds Wetenschappelijk Onderzoek - Vlanderen (FWO) under EOS Project no O005318F-RG47.

†Email: christophe.kervazo@telecom-paris.fr

Note that W, H and N are unknown, only X is given. Once W is estimated, H can be estimated for example using a nonnegative least squares (NNLS) algorithm. Problem 1 is sometimes referred to as simplex-structured matrix factorization (SSMF), and generalizes nonnegative matrix factorization (NMF); see [1] and the references therein.

In Problem 1, the columns of W are the vertices, while the columns of X are noisy data points within the convex hull of the columns of W,

$$\text{conv}(W) = \{x \mid x = Wy, y \geq 0, e^\top y = 1\},$$

where e is the vector of all ones of appropriate dimension. In fact, for all j,

$$X(:,j) = WH(:,j) + N(:,j),$$

where $H(:,j) \geq 0$ and $e^\top H(:,j) = 1$ (since H is column stochastic). To be able to estimate W in Problem 1, appropriate assumptions on W, H and N are required. In particular, the following assumptions are necessary:

- No column of W is contained in the convex hull of the other columns of W, otherwise it is not possible to distinguish it from a data point.
- The data points must be sufficiently spread within $\text{conv}(W)$, having data points on each facet of $\text{conv}(W)$. This implies some degree of sparsity for H.
- The noise N must be bounded.

To obtain provable or practical algorithms, the above assumptions must be carefully and rigorously complemented. Different assumptions on W, H, and N lead to different models for which different algorithms can be designed; see Section 2 for a literature review. We discuss below some applications of such algorithms.

Blind hyperspectral unmixing In this paper, we focus on Problem 1 in the context of blind hyperspectral unmixing (HU), a key problem in remote sensing. Let us briefly describe this problem; see the survey papers [7, 21] and the references therein for more details. A hyperspectral image (HSI) is a picture of a scene acquired within a large number of spectral bands (usually between 100 and 200). Thus, for each pixel a precise electromagnetic spectrum is recorded, which gives an information concerning the materials present in the pixel; specifically, about their reflectances (fraction of incoming light they reflect) and/or their emissivity (which is due to the fact that the materials usually have non-zero temperatures). Unfortunately, despite their high spectral resolution, hyperspectral sensors generally have a low spatial resolution; as such, the spectrum recorded for each pixel might not correspond to the one of a single material, but rather to a mixture of the spectra of the different materials present within the pixel.

Given an HSI, blind HU thus aims to recover the set of materials present in the image, called endmembers, along with the abundances of each endmember in each pixel. The standard model used to solve blind HU is the linear mixing model. It assumes that the spectral signature of each pixel is a linear combination of the spectral signatures of the endmembers, where the weights of the linear combination are the abundances of the endmembers in the pixel. Typically, we represent a hyperspectal image as a matrix X, where the jth column, $X(:,j)$, corresponds to the spectral signature of the jth pixel in the scene. If the spectral signatures of the endmembers are also collected as the

columns of a matrix W, then according to the linear mixing model, the jth pixel can be written as $X(:,j) \approx \sum_{k=1}^{r} W(:,k)H(k,j) + N(:,j)$, where $H(k,j)$ is the abundance of the kth endmember in the jth pixel, and $N(:,j)$ represents the noise and model misfit. This is exactly the setup of Problem 1.

An important class of algorithms to solve blind HU are *pure-pixel search* algorithms. They rely on the assumption that for each endmember, there is at least one pixel in which this endmember appears almost alone, that is, purely, so that the endmember signature is close to the one of the corresponding pure pixel. In the NMF literature, the pure-pixel assumption is referred to as separability, and the corresponding algorithms are referred to as separable NMF algorithms, or near-separable NMF algorithms. The pure-pixel assumption is reasonable for most high-resolution HSIs. Moreover, even when it is violated, pure-pixel search algorithms can still be used to find a decent initialization to more sophisticated algorithms which do not rely on it.

Applications in data analysis and machine learning In [6, 5], the authors describe in details various applications of solving Problem 1; in particular topic modeling via latent Dirichlet allocation, adversarial clustering, and community detection via the mixed membership stochastic block model. Moreover, Problem 1 generalizes NMF and as such could be useful for all applications of NMF, such as feature extraction in sets of images, audio source separation, or chemometrics.

Contribution and outline As far as we know, none of the existing pure-pixel search algorithms for blind HU leverage the fact that, when the pure-pixel assumption holds, then there are typically more than one pixel close to each endmember. In this paper, we leverage this fact, and propose smoothed versions of VCA and SPA, namely SVCA and SSPA. SVCA is an adaption of the recent algorithm, ALLS, proposed by Bhattacharyya and Kannan [6]. The presence of multiple pure pixels allows SVCA to be much more tolerant to noise than VCA. The idea is to average several data points around an endmember to obtain a better estimate of that endmember. Mathematically, SVCA is equivalent to applying VCA on a smoothed data set containing $\binom{n}{p}$ data points which are the averages of every combination of p data points from the original data set. Similarly, SSPA adapts SPA in the presence of multiple pure pixels.

The paper is organized as follows. In Section 2, we summarize the literature for solving Problem 1, with a focus on the three algorithms VCA, SPA and ALLS. In Section 3, we propose SVCA which is equivalent to applying VCA on a smoothed data set. SVCA is similar to ALLS, but two key differences make it empirically more efficient than ALLS. In Section 4, we propose SSPA which adapts SPA in the presence of multiple pure pixels. In Section 5, we show on synthetic and real-world hyperspectral data sets that SVCA and SSPA outperform VCA, SPA and ALLS in the presence of multiple pure pixels.

Motivation of this paper One of the goals of this paper is to try to bridge the gap between the machine learning community and the remote sensing community. In fact, many researchers with background in theoretical computer science and machine learning are not aware of the developments in blind HU; see, e.g., [4, 3, 6, 5]. Conversely, many researchers from the remote sensing community have not been using the latest developed algorithms with strong theoretical guarantees from the machine learning community, in particular from [6, 5].

In this paper, we adapt the ideas of [6, 5] to design more effective algorithms for blind HU, which we empirically illustrate on various data sets. Our focus in not on providing theoretical guarantees for these new algorithms, which is left for further research.

2 Simplex-structured matrix factorization

In this section, we describe existing models and algorithms to solve Problem 1, that is, to solve SSMF, in order to identify the vertices of the convex hull of a set of data points, X. We focus on two key models, upon which our contribution is built:

1. Separable NMF: it is equivalent to blind HU under the pure-pixel assumption. It assumes there is one data point (in blind HU, one pixel) close to each vertex of $\text{conv}(W)$ (in blind HU, to each endmember), and that the noise added to each pixel is bounded; see Section 2.1. Some authors refer to this model as near-separable NMF.

2. Learning a latent simplex: it is motivated by machine learning applications. It assumes that there is more than one data point close to each vertex of $\text{conv}(W)$, but it allows much larger noise levels as it only requires the ℓ_2 norm of N to be bounded, instead of each individual column; see Section 2.2.

Other models and algorithms exist to tackle Problem 1 relying on different assumptions. Although detailing them is out of the scope of this article, it is worth mentioning minimum-volume NMF [9, 19], where W is regularized such that its convex hull $\text{conv}(W)$ has the smallest possible volume; facet-based identification algorithms that identify the facets of $\text{conv}(W)$ from which its vertices are recovered [10, 18, 20, 1]; or probabilistic simplex component analysis [26] that relies on a probabilistic model on the data (the columns of H are sampled using the Dirichlet distribution, and the entries of N using i.i.d. Gaussian noise).

2.1 Model 1: Separable NMF

As already mentioned, an important class of blind HU algorithm corresponds to pure-pixel search algorithms, that build on the following formal assumption:

Assumption 1 (pure-pixel, separability)**.** *In Problem 1, there exists an index set $\mathcal{K}$ of cardinality r such that $H(\mathcal{K},:) = I_r$ where I_r is the r-by-r identity matrix.*

Under this assumption, solving Problem 1 amounts to recover $\mathcal{K}$ such that

$$X(:,\mathcal{K}) \;=\; W + N(:,\mathcal{K}) \quad \approx \quad W.$$

In the NMF literature, such algorithms are referred to as (near-)separable NMF algorithms.

Building on this assumption, the early algorithms in blind HU include pure-pixel index (PPI) in 1995 [8], N-FINDR in 1999 [25], and vertex component analysis (VCA) in 2005 [23]. Most of these algorithms were developed based on convex geometry concepts. These early works however did not analyze noise robustness, and in fact they are not guaranteed to recover the endmembers in the presence of noise.

In analytical chemistry, Problem 1 is closely related to the problem of self-modeling curve resolution [16]. As in blind HU, several algorithms were developed based on geometry concepts; in particular the successive projection algorithm (SPA) [2].

More recently, and motivated by applications in machine learning (in particular, topic modeling where pure data points are referred to as anchor words), Arora et al. [4] introduced the first provably robust near-separable NMF algorithms. Their robustness is deterministic: under some conditions, their

algorithm is guaranteed to recover an approximation of the vertices. Arora et al. were not aware of the algorithms developed within the blind HU literature. Many provably robust algorithms have followed this seminal paper, including algorithms that use linear programming [24, 11, 14], a generalization of SPA [15], fast anchor words [3], and the successive nonnegative projection algorithm (SNPA) [12]. These deterministically robust algorithms guarantee that, in the presence of noise, the endmembers are recovered, up to some error bounds that depend on the noise level and the conditioning of W; see Section 2.1.2 for such a result for SPA. We refer the interested reader to [13, Chapter 7] for a detailed discussion and comparison of these algorithms.

In the following, we describe in more detail VCA and SPA that will be instrumental in proposing our new algorithms, smoothed VCA in Section 3 and smoothed SPA in Section 4.

2.1.1 Vertex component analysis

VCA [23] is a greedy near-separable NMF algorithm, that is, it identifies the indices of the subset $\mathcal{K}$ sequentially. The index set is initialized with $\mathcal{K} = \emptyset$. At each of the r iterations of VCA, a random direction belonging to the subspace spanned by the r top left singular vectors of X is generated (this is equivalent to working with the best rank-r approximation of X, and hence filters the noise). This direction is then projected onto the orthogonal complement of $X(:, \mathcal{K})$, and the index of the column of X that maximizes the absolute value of the inner product with that direction is added to $\mathcal{K}$. Algorithm 1 summarizes VCA.

Algorithm 1 Vertex Component Analysis (VCA) [23]

Input: The matrix $X \in \mathbb{R}^{m \times n}$, the number r of columns to extract.

Output: Index set $\mathcal{K}$ of cardinality r such that $X \approx X(:, \mathcal{K})H$ for some $H \geq 0$.

1: Let $\mathcal{K} = \emptyset$, $P^\perp = I_m$, $V = [\,]$.
2: Let $Y \in \mathbb{R}^{m \times r}$ be the vector space spanned by the top r left singular vectors of X.
3: **for** $k = 1 : r$ **do**
4: Pick a random direction $d_k \in \mathbb{R}^m$ in the subspace spanned by Y, e.g., $d_k \sim Y\mathcal{N}(0, I_r)$.
5: Compute $u_k = (d_k^T P^\perp)X \in \mathbb{R}^n$, and let $j_k = \operatorname{argmax}_{1 \leq j \leq n} |u_k(j)|$.
6: Let $\mathcal{K} = \mathcal{K} \cup \{j_k\}$.
7: Update the projector $P^\perp$ onto the orthogonal complement of $W = X(:, \mathcal{K})$:

$$v_k = \frac{P^\perp X(:, j_k)}{\|P^\perp X(:, j_k)\|_2}, \quad V = [V\, v_k], \quad P^\perp \leftarrow \left(I_m - VV^T\right).$$

8: **end for**

The computational cost of VCA is $\mathcal{O}(r\,\mathrm{nnz}(X))$ operations, where $\mathrm{nnz}(X)$ is the number of non-zero entries of X. The main cost is to compute Y which is can be done efficiently using the subspace power iteration, in $\mathcal{O}(\mathrm{nnz}(X)r)$ operations, and to compute the products $(d_k^T P^\perp)X$ at each of the r iterations.

An important drawback of VCA is that is it not guaranteed to be deterministically robust to noise. In other words, for any noise N such that a data point goes outside $\mathrm{conv}(W)$, there is a non-zero probability that VCA extract this point. The reason is that VCA uses a linear function to identify the vertices of $\mathrm{conv}(W)$; see [13, Chapter 7.4] for a numerical example.

2.1.2 Successive Projection Algorithm

SPA is very similar to VCA. The only difference is in the selection step, when adding an index to $\mathcal{K}$. SPA selects the column of $P^\perp X$ with maximum ℓ_2 norm; see Algorithm 2. To have an efficient implementation of SPA, in $\mathcal{O}(r\,\mathrm{nnz}(X))$ operations like VCA, one should use the following formula sequentially: for any vectors x and y with $\|y\|_2 = 1$,

$$\|(I - yy^\top)x\|_2^2 = \|x\|_2^2 - y^\top x.$$

Algorithm 2 Successive Projection Algorithm (SPA) [2]

Input: The matrix $X \in \mathbb{R}^{m\times n}$, the number r of columns to extract.

Output: Index set $\mathcal{K}$ of cardinality r such that $X \approx X(:,\mathcal{K})H$ for some $H \geq 0$.

1: Let $\mathcal{K} = \emptyset$, $P^\perp = I_m$, $V = [\,]$.
2: Let $u_1(j) = \|X(:,j)\|_2^2$ for all j.
3: **for** $k = 1 : r$ **do**
4: Let $j_k = \mathrm{argmax}_{1\leq j\leq n} u_k(j)$. (Break ties arbitrarily, if necessary.)
5: Let $\mathcal{K} = \mathcal{K} \cup \{j_k\}$.
6: Update the projector $P^\perp$ onto the orthogonal complement of $W = X(:,\mathcal{K})$:

$$v_k = \frac{P^\perp X(:,j_k)}{\|P^\perp X(:,j_k)\|_2}, \quad V = [V\, v_k], \quad P^\perp \leftarrow \left(I_m - VV^T\right).$$

7: Update the squared norms of the columns of $P^\perp X$: for all j,

$$u_{k+1}(j) = u_k(j) - v_k^\top X(:,j) = \|P^\perp X(:,j)\|_2^2.$$

8: **end for**

It is interesting to note that VCA is equivalent to SPA if the direction u_k randomly chosen at each step is instead taken as the column of the residual $P^\perp X$ with maximum ℓ_2 norm. We will use this observation for our proposed algorithm, smoothed SPA.

Robustness of SPA As opposed to VCA, SPA is deterministically robust to noise, provided the following assumption on top of separability (Assumption 1):

Assumption 2 (column-wise bounded noise)**.** *In Problem 1, the noise satisfies $\|N(:,j)\|_2 \leq \epsilon$ for all j for some $\epsilon > 0$ sufficiently small.*

Let us state the robustness result for SPA.

Theorem 1. *[15, Theorem 3] Let $X = WH + N$ as in Problem 1, and let Assumptions 1 and 2 be satisfied, that is, $W = X(:,\mathcal{K}^*)$ for some index set $\mathcal{K}^*$ of cardinality r, and $\|N(:,j)\|_2 \leq \epsilon$ for all j where $\epsilon \leq \mathcal{O}\left(\frac{\sigma_r^3(W)}{\sqrt{r}K(W)^2}\right)$. Let also the rth singular value of W be positive, that is, $\sigma_r(W) > 0$, meaning that W has rank r. Let $\mathcal{K}$ be the index set extracted by SPA. Then there exists a permutation π of*

$\{1, 2, \ldots, r\}$ *such that for all* $k = 1, 2, \ldots, r$,

$$\|X(:, \mathcal{K}(k)) - W(:, \pi(k))\|_2 \leq \mathcal{O}\left(\frac{\epsilon K(W)^2}{\sigma_r^2(W)}\right),$$

where $\mathcal{K}(k)$ *denotes the* k*th index in* $\mathcal{K}$*, and* $K(W) = \max_j \|W(:, j)\|_2$.

Note that the bounds in Theorem 1 are relatively weak: the noise level has to be rather small to guarantee SPA to recover W approximately.

2.2 Model 2: Learning a latent simplex

A drawback of near-separable algorithms, such as VCA and SPA, is that they assume that there is only one data point close to each column of W. Therefore, to estimate W, the column-wise bounded noise assumption (Assumption 2) is necessary; see Theorem 1. This is a rather strong assumption, often not met in practical situations as typically many data points are affected by large amounts of noise.

Bhattacharyya et al. [6] rather propose to leverage the fact that typically more than one data point are close to each column of W. This assumption, which is stronger than the pure-pixel one (requiring only a single pure-pixel), allows higher noise levels. It is called the *proximate latent points* assumption, and is defined as follows.

Assumption 3 (proximate latent points)**.** *In Problem 1, there exists* r *index sets,* $\mathcal{K}_k$ *for* $k = 1, 2, \ldots, r$*, of cardinality at least* $p = \delta n$ *such that*

$$\|WH(:, j) - W(:, k)\|_2 \leq \frac{4\sigma}{\delta} \quad \text{for all } j \in \mathcal{K}_k,$$

for some $\delta \in \left[\frac{1}{n}, \frac{1}{r}\right]$ *and* $\sigma > 0$.

Under this assumption, instead of looking for one column of X to represent each vertex, like in VCA and SPA, algorithms should look for p of them and then estimate each vertex as the average of these p data points. We will refer to such algorithms as *smoothed separable NMF algorithms*. The main contribution of this paper is to propose two new such algorithms; in Sections 3 and 4.

This assumption is often met in the machine learning applications mentioned in Section 1; see the discussions in [6, 5]. For high-resolution HSIs that satisfy the pure-pixel assumption, there are typically more than one pixel close to each endmember.

Spectral variability in blind HU In blind HU, an issue with separable NMF algorithms is that they identify a single pixel to represent a material. It is however well-known that the spectral signature of an endmember may vary across the pixels of the image, for example because of differences in light intensity or orientation. This is known as *spectral variability*. By construction, most separable NMF algorithms, such as VCA and SPA, will identify pure pixels that do not represent well the average behaviour of a material, but rather a pure pixel located at the boundary of the convex hull of the variations of the spectral signature of that endmember. Therefore, working on the smoothed data set, which averages every subset of p data points, allows to better represent this average behaviour.

2.2.1 Algorithm to learn a latent simplex

To solve Problem 1 under Assumption 3, Bhattacharyya and Kannan [6] proposed an algorithm similar to VCA, which we will refer to as the algorithm for learning a latent simplex (ALLS). The main difference between ALLS and VCA is the selection step. Instead of picking a single column of X, ALLS averages over p columns for some $p \in \left\{1, 2, \ldots, \lfloor \frac{n}{r} \rfloor\right\}$. More precisely, ALLS picks the p columns corresponding to the indices that maximize the absolute value of u_k; see Algorithm 3.

The idea behind ALLS is to apply VCA on a smoothed data set. This smoothed data set is made of $\binom{n}{p}$ data points which are the averages of all possible combinations of p data points, that is, p columns of X. Of course, constructing this smoothed data set explicitly is not practical, since $\binom{n}{p}$ grows exponentially. However, by the linearity of the selection step in VCA, this is not necessary: the smoothed data point that maximizes a linear function is the average of the p data points that have the p largest values for that function. Algorithm 3 summarizes ALLS.

Algorithm 3 Algorithm for Learning a Latent Simplex (ALLS) [6]

Input: The matrix $X \in \mathbb{R}^{m \times n}$, the number r of columns of W, the number p of columns of X to be averaged to obtain each column of W.

Output: A matrix W' such that $X \approx W'H$ for some $H \geq 0$.

1: Let $W' = [\,]$, $P^\perp = I_m$, $V = [\,]$.
2: Let $Y \in \mathbb{R}^{m \times r}$ be the vector space spanned by the top r left singular vectors of X.
3: **for** $k = 1 : r$ **do**
4: Pick a random direction $d_k \in \mathbb{R}^m$ in the subspace spanned by Y, e.g., $d_k \sim Y\mathcal{N}(0, I_r)$.
5: Compute $u_k = \left(d_k^T P^\perp\right) X \in \mathbb{R}^n$.
6: Let $\mathcal{S}_k$ be the set of p indices corresponding to the largest coordinates of u_k in absolute value.
7: Let $W'(:, k)$ be the average of the columns of $X(:, \mathcal{S}_k)$.
8: Update the projector $P^\perp$ onto the orthogonal complement of W':

$$v_k = \frac{P^\perp W'(:,k)}{\|P^\perp W'(:,k)\|_2}, \quad V = [V\, v_k], \quad P^\perp \leftarrow \left(I_m - VV^T\right).$$

9: **end for**

Note that ALLS with $p = 1$ is equivalent to VCA.

Computational cost The only additional cost of ALLS compared to VCA is to average p columns of X, which requires r times $\mathcal{O}(pm)$ operations, which is negligible since $p \ll n \leq \text{nnz}(X)$.

Remark 1 (ALLS in $O\big(\text{nnz}(Z)\big)$ time)**.** *In a more recent work, Bakshi et al. [5] improved ALLS, from a computational point of view, by providing an algorithm running in $\mathcal{O}(nnz(X))$ operations. To do so, Bakshi et al. rely on advanced low-rank matrix approximation algorithms. We will not focus on this rather technical aspect which is out of the scope of this paper. In fact, for HSI, this is not crucial as X is typically dense, while m is small (between 100 and 200, typically). Hence, in our implementation, we simply use to the rank-r truncated SVD.*

2.2.2 Probabilistic robustness of ALLS

Let us describe the assumptions needed to prove the probabilistic robustness of ALLS. The condition on W is defined as follows:

Assumption 4 (well-separatedness of W)**.** *In Problem 1, the matrix W satisfies*

$$\alpha(W) \quad = \quad \frac{\min_{k=1,2,\ldots,r} \min_x \|W(:,k) - W(:,\bar{k})x\|_2}{K(W)} \quad > \quad 0, \tag{1}$$

where $\bar{k} = \{1, 2, \ldots, r\} \backslash \{k\}$.

Assumption 4 holds if and only if $\text{rank}(W) = r$, in which case $\text{conv}(W)$ is a simplex, that is, a polytope of dimension $r-1$ with r vertices (hence the name of the algorithm).

The condition on the noise is as follows.

Assumption 5 (Spectrally bounded perturbations)**.** *In Problem 1,*

$$\|N\|_2 = \sigma_{\max}(N) \ \leq \ \sigma\sqrt{n},$$

where there exists some constant c such that

$$\sigma \ \leq \ \frac{\alpha^2\sqrt{\delta}}{c\, r^9} \min_j \|W(:,j)\|_2, \tag{2}$$

where $\alpha = \alpha(W)$ is defined in Assumption 4, and δ and σ in Assumption 3 (recall, $p = \delta n$ is the number of data points close to each column of W).

It is key to note here that the noise allowed is not column wise as in Assumption 2, but on the spectral norm of N, which is rather different.

We can now state the robustness theorem for ALLS.

Theorem 2. *Let us consider Problem 1 under Assumptions 3 (proximate latent points), 4 (well-separatedness of W) and 5 (spectrally bounded perturbations). Then, with probability at least $1 - c/r^{3/2}$, ALLS computes a matrix W' such that upon permutation of its columns, for all $k = 1, 2, \ldots, r$,*

$$\|W(:,k) - W'(:,k)||_2 \ \leq \ O\left(\frac{r^4\sigma}{\alpha\sqrt{\delta}}\right). \tag{3}$$

Note that substituting (2) in (3) gives

$$\|W(:,k) - W'(:,k)||_2 \ \leq \ O\left(\frac{\alpha}{c\, r^5}\right) \min_j \|W(:,j)\|_2.$$

Interestingly, since ALLS for $p = 1$ coincide with VCA, Theorem 2 provides a probabilistic robustness result for VCA which is unknown in the blind HU literature.

2.2.3 Bounds of SPA versus ALLS

Theorem 2 might look somewhat weak because of the dependence in r^9 in the bound (2) for σ. However, it is not known whether this bound is tight, although it is believed it could be improved [6, 5]. A similar comment applies to SPA. Moreover, these bounds assume an adversarial settings, and noise robustness under particular generative models is also an interesting direction of research, as in [26].

In any case, Theorem 2 only requires a bound on $\|N\|_2$ while SPA requires each column of the noise matrix N to be bounded, indicating that ALLS should perform better, in general, when p is sufficiently large. Since the theory is still not fully developed and the tightness of the theoretical bounds should be carefully studied, it is important to compare these algorithm empirically to shed light on their differences on practical problems; this will be done in Section 5.

3 Smoothed VCA

Inspired by VCA, and ALLS, we now propose smoothed VCA (SVCA); see Algorithm 4. SVCA has two key important differences compared to ALLS:

1. At step k, ALLS selects the p entries maximizing the absolute value of the vector of u_k, obtained as the inner product of X and a randomly generated direction $d_k^\top P^\perp$; see steps 5-6 of Algorithm 3. This is not equivalent to maximizing (or minimizing) the linear function $l(x) = d_k^\top P^\perp x$ over the smoothed polytope. In fact, by using the absolute value, this approach could select data points in opposite directions. For example, take the simple case with two vertices $w_1 = (-1, 0)$ and $w_2 = (0, 1)$. For any direction d, we have $|d^\top w_1| = |d^\top w_2|$ and hence it is very likely that data points close to both vertices will maximize $|d^\top x|$, and their average will be a poor approximation of both vertices.

   Instead, to maximize (or minimize) $l(x)$, one should select the p indices maximizing u_k (or $-u_k$). In SVCA, we therefore propose to select the p indices that maximize (resp. minimize) u_k if the median of the p largest values is larger (resp. smaller) than the absolute value of the median of the p smallest values of u_k. We have observed in practical experiments that this modification of ALLS is crucial to obtain competitive results in real-world hyperspectral images. In fact, we will see that SVCA outperforms ALLS, and the main reason is this modified selection step.

2. Instead of averaging p columns of X at each step, we will also consider taking their median. This allows SVCA to be much more tolerant to gross corruptions and outliers, which are often present in HSI. (In the presence of Gaussian noise, using the average is better.)

   It also allows SVCA to be more tolerant to a misspecified value of p. For example, assume a very simplistic scenario where there are $n = rp'$ data points and no noise, with exactly p' data points close to each endmember (that is, all data points are pure). For $p < p'$, one does not leverage optimally the presence of multiple pure pixels. On the other side, as soon as p is larger than p', ALLS will perform rather badly because it will average p' data points close to a vertex and $p - p'$ data points corresponding to another vertex. If instead one takes the median, the algorithm remains able to extract the endmembers for any $p < 2p'$. In practice, as we will show in Section 5, using the median performs significantly better on real data sets.

Note that SVCA has the same computational cost as VCA, SPA and ALLS, namely $\mathcal{O}(r\text{nnz}(X))$ operations.

Algorithm 4 Smoothed Vertex Component Analysis (SVCA)

Input: The matrix $X \in \mathbb{R}^{m \times n}$, the number r of columns of $W \in \mathbb{R}^{m \times r}$, the number p of columns of X to be averaged to obtain each column of W, the aggregation method (median or mean).

Output: A matrix W such that $X \approx WH$ for some $H \geq 0$.

Let $W = [\,]$, $P^\perp = I_m$, $V = [\,]$.
Let $Y \in \mathbb{R}^{m \times r}$ be the vector space spanned by the top r left singular vectors of X.
for $k = 1 : r$ **do**
    Pick a random direction $d_k \in \mathbb{R}^m$ in the subspace spanned by Y, e.g., $d_k \sim Y\mathcal{N}(0, I_r)$.
    Compute $u_k = \left(d_k^T P^\perp\right) X \in \mathbb{R}^n$.
    if the median of the p largest values of u_k is larger than the absolute value of the median of the p smallest values of u_k **then**
        Let $\mathcal{S}_k$ be the set of p indices maximizing u_k.
    else
        Let $\mathcal{S}_k$ be the set of p indices minimizing u_k.
    end if
    Let $W(:, k)$ be the median (or the mean) of the columns of $X(:, \mathcal{S}_k)$.
    Update the projector $P^\perp$ onto the orthogonal complement of $W = X(:, \mathcal{K})$:

$$v_k = \frac{P^\perp X(:, j_k)}{\|P^\perp X(:, j_k)\|_2}, \quad V = [V\, v_k], \quad P^\perp \leftarrow \left(I_m - VV^T\right).$$

end for

Recovery Guarantees for SVCA SVCA is very similar to ALLS, and in fact the robustness analysis of ALLS applies to SVCA, that is, Theorem 2 applies to SVCA. The reason is that we guarantee SVCA to extract the data point in the smoothed data set that maximizes the absolute value $l(x) = d_k^\top P^\perp x$. Note that, interestingly, SVCA with $p = 1$ coincides with VCA, and hence Theorem 2 also applies to VCA, although the bound is rather weak; see the discussion in section 2.2.3.

4 Smoothed SPA

Since SVCA is equivalent to VCA for $p = 1$, it is not guaranteed to be deterministically robust to noise. This motivates us to propose smoothed SPA. Unfortunately, it is not practical to apply SPA directly on the smoothed data set. Indeed, it would require to find the p columns of the smoothed data set with the largest ℓ_2 norm. The ℓ_2 norm being a nonlinear function, it would require to explicitly compute the $\binom{n}{p}$ data points of the smoothed data set, which is computationally prohibitive.

Instead, we replace the random selection of $u_k = Y\mathcal{N}(0, 1)$ in SVCA by the column of the residual $P^\perp X$ with maximum ℓ_2 norm, that is, $u_k = P^\perp X(:, j_k)$ for some j_k so that $\|u_k\|_2 \geq \|P^\perp X(:, j)\|_2$ for all j. This allows us to combine the best of 'both worlds': deterministic robustness under separability when $p = 1$, and the proximate latent point assumption (Assumption 3).

Recovery Guarantees for SSPA For $p = 1$, SSPA coincides with SPA, and hence Theorem 1 applies to SSPA for $p = 1$, that is, it is deterministically robust to column-wise bounded noise.

Algorithm 5 Smoothed Successive Projection Algorithm (SSPA)

Input: The matrix $X \in \mathbb{R}^{m\times n}$, the number r of columns of $W \in \mathbb{R}^{m\times r}$, the number p of columns of X to be averaged to obtain each column of W, the aggregation method (median or mean).

Output: A matrix W such that $X \approx WH$ for some $H \geq 0$.

Let $W = [\,]$, $P^\perp = I_m$, $V = [\,]$.
Let $u_1(j) = \|X(:,j)\|_2^2$ for all j.
for $k = 1 : r$ **do**
    Let $j_k = \operatorname{argmax}_{1\leq j\leq n} u_k(j)$. (Break ties arbitrarily, if necessary.)
    Let $d_k = X(:,j_k)$.
    Compute $u_k = (d_k^T P^\perp)X \in \mathbb{R}^n$.
    if $\max_i u_k(i) \geq -\min_i u_k(i)$ **then**
        Let $\mathcal{S}_k$ be the set of p indices maximizing u_k.
    else
        Let $\mathcal{S}_k$ be the set of p indices minimizing u_k.
    end if
    Let $W(:,k)$ be the median (or the mean) of the columns of $X(:,\mathcal{S}_k)$.
    Update the projector $P^\perp$ onto the orthogonal complement of $W = X(:,\mathcal{K})$:

$$v_k = \frac{P^\perp X(:,j_k)}{\|P^\perp X(:,j_k)\|_2}, \quad V = [V\, v_k], \quad P^\perp \leftarrow \left(I_m - VV^T\right).$$

    Update the squared norms of the columns of $P^\perp X$: for all j,

$$u_{k+1}(j) = u_k(j) - v_k^\top X(:,j) = \|P^\perp X(:,j)\|_2^2.$$

end for

However, since the selection step of SSPA is deterministic, Theorem 2 does not apply to SSPA. A promising direction of further research would be to analyze noise robustness of SSPA for $p > 1$.

Should you use SVCA or SSPA? SVCA has the advantage to be a randomized algorithm, and hence can be run multiple times and the best solution, according to some criterion, can be kept. SSPA is deterministic and has the advantage to have stronger theoretical guarantees for $p = 1$. From a practical point of view, one could run SVCA several times, and SSPA once, and then keep the best solution. In this paper, when the ground truth W^* is not available, we will use the following criterion

$$Q_F(W) \;=\; \frac{\min_{H\geq 0} \|X - WH\|_F}{\|X\|_F} \;\in\; [0,1],$$

to evaluate the quality of a solution W. Note that we do not use the sum-to-one constraint, $H^\top e = e$, because, in many practical situations, including hyperspectral imaging, this constraint is not satisfied for all columns of H, e.g., for pixels with low luminosity; see a discussion in [12].

In matrix factorization, it could be argued that in general $Q_F(W)$ is not a good measure to assess the quality of a solution W, since any W such that $\operatorname{conv}(X) \subset \operatorname{conv}(W)$ implies $Q_F(W) = 0$, even

if W is very different from the ground truth W^*. In our case though, $Q_F(W)$ is relevant, since all the considered algorithms generate solutions W which columns are close to $\text{conv}(X)$. Indeed, VCA, SPA and ALLS generate solutions within $\text{conv}(X)$. This is not always the case for SVCA and SSPA, because of the use of the median (a non-linear operator) for the aggregation of the columns of X. In practice though, they find solutions close to $\text{conv}(X)$.

5 Numerical Experiments

In this section, we study and compare the performance of ALLS, SVCA, and SSPA. We first consider synthetic datasets and then the unmixing of real-world hyperspectral images. The code and data are available online[1]. All algorithms are implemented in Matlab and run on a computer with an i5-8350U processor.

5.1 Synthetic Data Sets

In this section, we study the behavior of smooth separable NMF algorithms in several experimental setups. To build synthetic data sets, we first build $W \in \mathbb{R}_+^{224\times 10}$ by selecting 10 columns from the USGS hyperspectral library[2] using SPA. The condition number of the corresponding matrix was $\kappa(W) = 33.88$. Then, we generate a random $H \in \mathbb{R}_+^{10\times 1000}$ such that $H = [I_{10}, H']$, meaning there is at least one pure pixel for every endmember. The coefficients of H' follow a Dirichlet distribution, which is usually a good model for the abundances in HSI [22], of parameters αe, where α controls the proportion of pixels close to the endmembers, see Table 1. The larger α, the denser the columns of H and the less likely the 'proximal latent points' assumption is to be satisfied for large p.

Table 1: Generating the columns of $H \in \mathbb{R}^{10\times n}$ using the Dirichlet distribution of parameter αe, this table reports the expected percentage of pure pixels close to each endmember. The pixel j is considered close to the endmember i when $H(i,j) > 0.95$, hence this table reports the expected value of $\frac{1}{n}\big|\{j \mid H(i,j) > 0.95\}\big|$ for all i. Since the Dirichlet distribution is uniform with parameters αe, this expected value is the same for all i.

α	0.01	0.02	0.05	0.1	0.2	0.5
δ	7.7%	5.9%	2.7%	0.75%	0.06%	0%

Finally, we let $X = WH + N$ where N is a normalized Gaussian noise: Given a noise level ϵ, we first generate $N(i,j) \sim \mathcal{N}(0,1)$ for all (i,j), then set

$$N \leftarrow \epsilon \frac{\|WH\|_F}{\|N\|_F} N,$$

so that ϵ is the norm of the noise relative to WH: $\|N\|_F = \epsilon \|WH\|_F$.

Given the noisy data matrix X and a parameter p, we can compute W' with the algorithms ALLS, SVCA, and SSPA. We note ALLS(p), SVCA(p) and SSPA(p) these algorithms run with parameter p. Given the computed solution W', we report the mean removed spectral angle (MRSA) to assess its quality. Given two spectral signatures $x, y \in \mathbb{R}^m$, the MRSA is defined as follows:

$$\phi(x,y) = \frac{1}{\pi} \arccos\left(\frac{(x-\bar{x})^T(y-\bar{y})}{\|x-\bar{x}\|_2 \|y-\bar{y}\|_2}\right) \in [0,1],$$

[1] `https://gitlab.com/nnadisic/smoothed-separable-nmf`

[2] `https://www.usgs.gov`

where for a vector $z \in \mathbb{R}^m$, $\bar{z} = (\sum_{i=1}^m z_i)e$ and e is the vector of all ones. Given two matrices, here the groundtruth W and the estimate W', we define the MRSA as

$$\text{MRSA}(W, W') = \sum_{j=1}^{r} \phi(W(:,j), W'(:,j)),$$

after the columns of W' have been reordered so as to minimize the MRSA. The smaller the MRSA, the better the solution. For ALLS and SVCA, on a given data set, we run 30 trials and keep the median of the results. SSPA is deterministic so we only run it once. Unless stated otherwise, SVCA and SSPA are equipped with the median aggregation. In the following, we consider several experimental setups to highlight the property of these algorithms.

In fig. 1, we test ALLS, SVCA, and SSPA with different values of the parameter p, when the noise ϵ varies. Note that SVCA(1) and SSPA(1) are equivalent to their non-smoothed version VCA and SPA. Also note that ALLS(1) is equivalent to SVCA(1) and thus VCA. In this experiment, we observe that smoothing improves the algorithm performances. However, for ALLS, a parameter p set too large can in fact worsen the solution, especially when the noise level is small. Also, ALLS is outperformed by SVCA and SSPA; this will be confirmed in experiments on hyperspectral images in section 5.2.

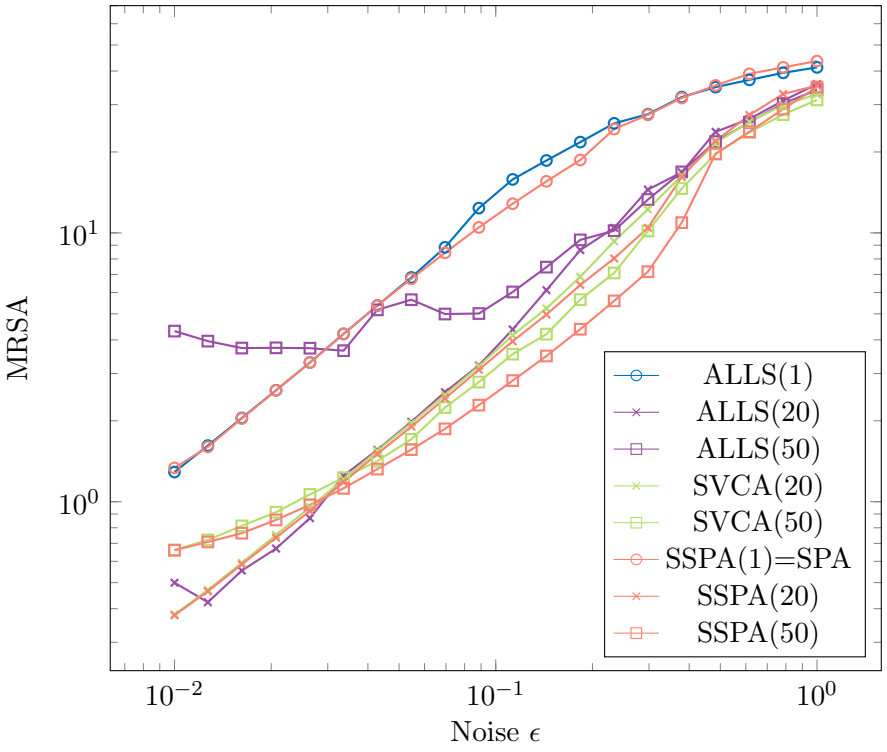


Figure 1: Results for ALLS, SVCA, and SSPA for different values of p, when ϵ varies, for fixed $n = 1000$ and purity $\alpha = 0.05$ ($\delta = 2.7\%$). Values for ALLS and SVCA are the medians over 30 trials. Note that ALLS(1)=SVCA(1)=VCA.

In fig. 2, we compare the stability of ALLS and SVCA when ϵ varies by showing the best, median, and worst result among 30 runs. We also compare them to the MRSA of the result of SVCA that has the smallest reconstruction error, $Q_F(W')$, and to SSPA. We see that the best result from ALLS

is slightly better than other results. This is due to the use of the mean as an aggregation method, which works better with the centered Gaussian noise of the synthetic data[3], see fig. 4. However, the algorithm is less stable, as the median and worst result are worst than SVCA. With SVCA, the median results are close to the best. Also, the best results in terms of reconstruction error generally coincides with the best one in terms of MRSA, showing than the reconstruction error is a good proxy for the tested algorithms, as evoked in Section 4 (this will be useful when the groundtruth is unknown and the MRSA cannot be computed, for example in section 5.2). The deterministic SSPA is better than the median result of SVCA, but not always better than its best result.

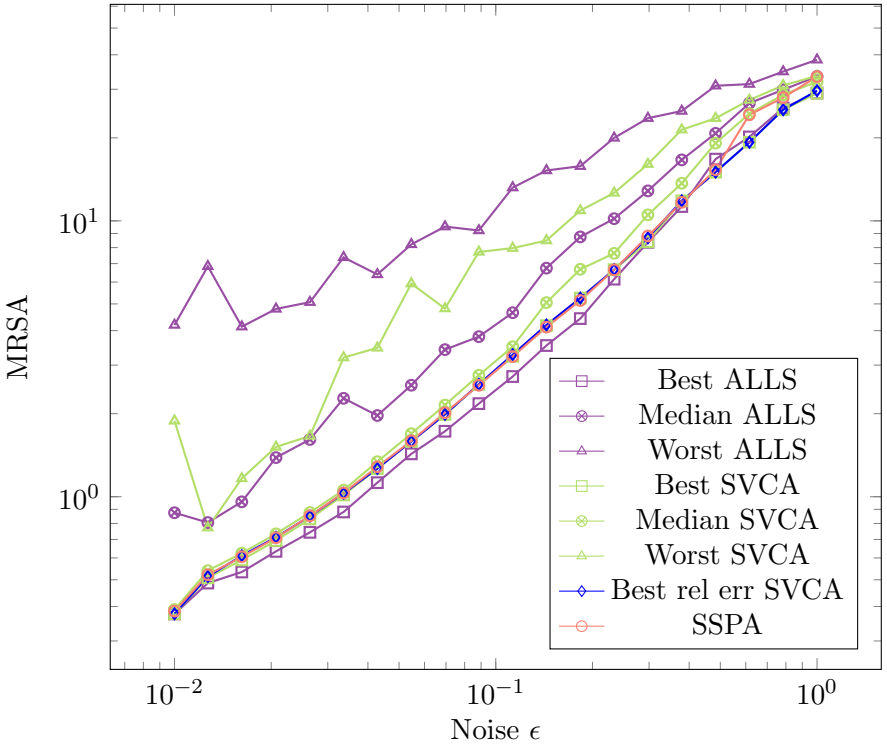


Figure 2: Comparison of SSPA with best, median, and worst result for ALLS and SVCA among 30 runs, when ϵ varies, for fixed $n = 1000$, purity $\alpha = 0.05$ ($\delta = 2.7\%$) and parameter $p = 25$. "Best rel err SVCA" represents the MRSA of the solution of SVCA that has the smallest relative reconstruction error.

In fig. 3, we compare SVCA and SSPA with higher values of p when ϵ varies. Again, we observe that the smoothing improves the algorithm performances, but an overestimated p worsens it. Interestingly, when the noise is very high (above 10%), smoothed algorithms outperform their non-smoothed counterpart even when p is overestimated. This is due to the fact that the value of p required to obtain the best estimation of W is not only determined by the purity but also by the noise level. For instance, consider a toy example with four data points and $r = 2$:

$$X = WH + N = W \begin{bmatrix} 1 & 0 & 0.99 & 0.01 \\ 0 & 1 & 0.01 & 0.99 \end{bmatrix} + N.$$

[3]Using the mean, instead of the median in SVCA, its best MRSA result are always better than that of ALLS in this experiment.

That is, x_3 and x_4 are not pure-pixel but are almost pure. Let us further assume N to follow a centered Gaussian law. If $\epsilon = 0$ (noiseless mixing), the best estimation of $W(:,1)$ (resp. $W(:,2)$) from X is to extract $X(:,1)$ (resp. $X(:,2)$), yielding a perfect estimation. On the other hand, if ϵ is large relatively to the distance of $W(:,1)$ and $W(:,2)$, it is better to choose as an estimate of $W(:,1)$ (resp. $W(:,2)$) the average – or median – of $X(:,1)$ and $X(:,3)$ (resp. $X(:,2)$ and $X(:,4)$), as the noise power is then divided by two.

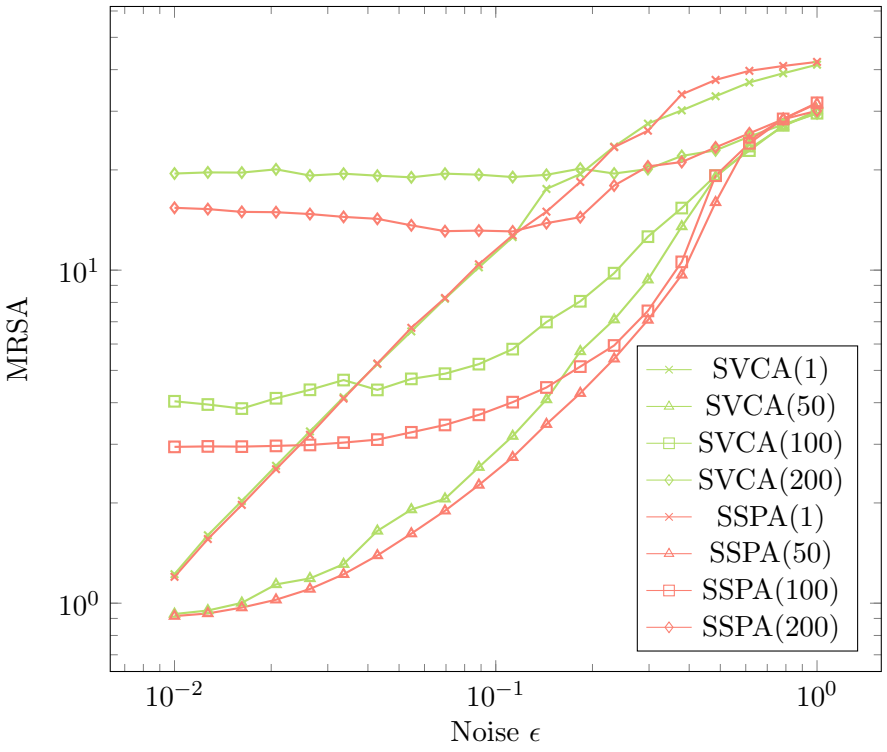


Figure 3: Results for SVCA and SSPA for different values of p, when ϵ varies, for fixed $n = 1000$ and purity $\alpha = 0.05$ ($\delta = 2.7\%$), Values for SVCA are the medians over 30 trials.

In fig. 4, we compare SVCA and SSPA equipped with either the median or the mean aggregation, for fixed data setup and when p varies. The reverse bell curve shows that the performance of the algorithms improves gradually as p grows, until it reaches an optimal value, after which the performance gradually worsen. We observe that the algorithms equipped with the median are more robust to an overestimation of p, but with the mean they are slightly better for smaller p. However the difference is small; this is expected as this synthetic data is generated with centered Gaussian noise, and as such the mean is expected to give the best estimation when p is well chosen. Note that the results would be different with different kind of noises, and the median could for instance be better with sparse noises. The difference is more obvious in hyperspectral images, see section 5.2 and fig. 6.

In fig. 5, we compare SVCA and SSPA when p varies in setups with different values of purity α. As expected, we observe that the shapes of the curves are similar, and that for a fixed noise level the value of the parameter p leading to the lowest MRSA decreases as the parameter α increases.

To summarize, our synthetic experiments highlight that the two proposed algorithms seem to obtain better results than ALLS. They furthermore show that SVCA and SSPA outperform VCA and SPA when p is well chosen. Although we illustrate that the choice of p is important, as a bad value

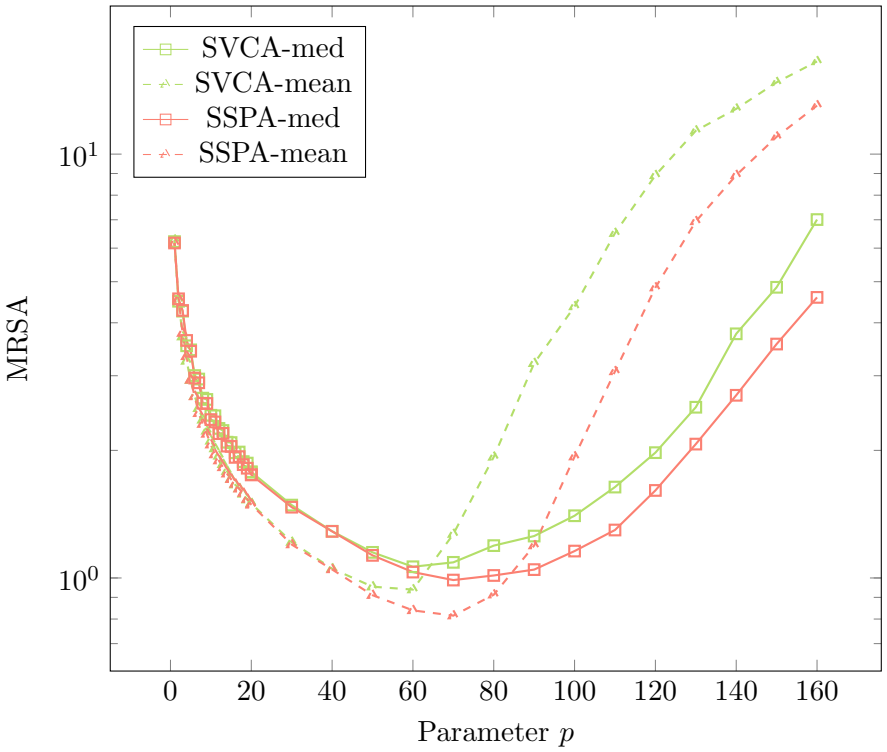


Figure 4: Results for SVCA and SSPA using either the median or the mean to average points, when p varies, for fixed $n = 1000$, purity $\alpha = 0.05$ ($\delta = 2.7\%$), and noise $\epsilon = 0.05$. Values for SVCA are the medians over 30 trials.

can worsen the results compared to the non-smoothed algorithms, the use of the median instead of the mean makes this choice easier.

5.2 Hyperspectral images

In this section, we apply ALLS, SVCA, and SSPA to the unmixing of hyperspectral images, as described in section 1. We consider three commonly used hyperspectral images[4], San Diego, Urban, and Terrain. In hyperspectral data sets, extremely large values are commonly associated with sensor noise or interference. To avoid overfitting the factorizations to this interference, the pixels corresponding to the 10 largest values of any wavelength range are zeroed out. Extreme pixels generally have extreme values in many wavelenght ranges at once, so this preprocessing results in the removal of less than 0.1% pixels. The characteristics of these images are summarized in table 2.

Table 2: Summary of the hyperspectral images studied in this work.

Dataset	m	n	r	Extreme pixels removed
San Diego	158	$400 \times 400 = 160000$	8	19
Urban	162	$307 \times 307 = 94249$	6	68
Terrain	188	$500 \times 307 = 153500$	6	107

[4]Downloaded from `http://lesun.weebly.com/hyperspectral-data-set.html`

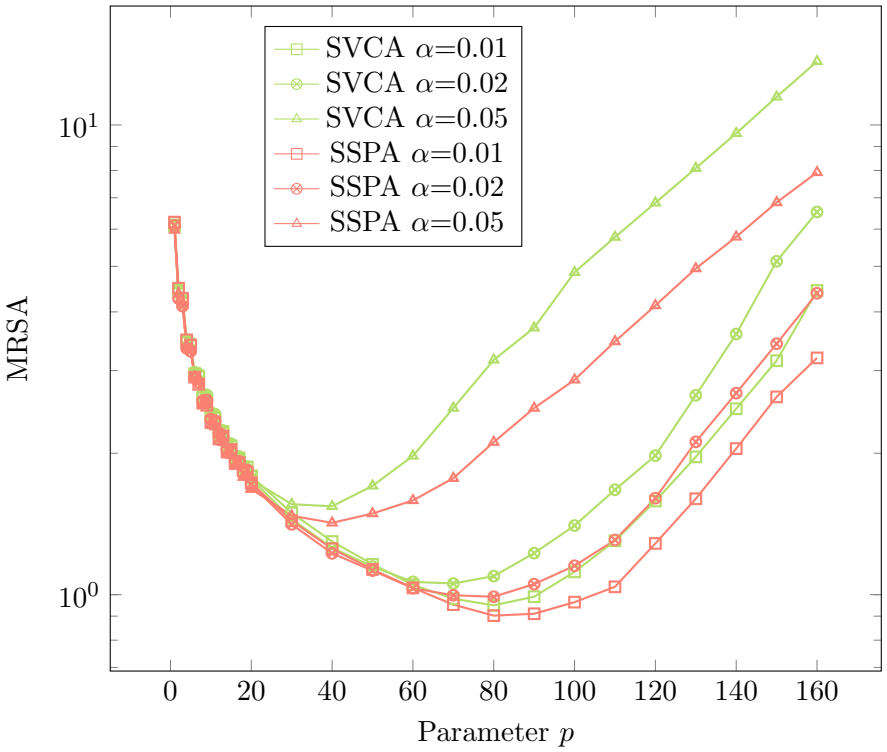


Figure 5: Results for SVCA and SSPA for different values of purity α, when p varies, for fixed $n = 1000$ and noise $\epsilon = 0.05$. Values for SVCA are the medians over 30 trials.

Given a data matrix $X \in \mathbb{R}^{m \times n}$, we compute $W \in \mathbb{R}^{m \times r}$ with the three algorithms. We then compute for each algorithm $H \in \mathbb{R}^{r \times n}$ with a standard coordinate descent algorithm, and measure the relative reconstruction error $\|X - WH\|_F / \|X\|_F$. The smaller the error, the better the solution.

Some works such as [27] proposed groundtruths for these hyperspectral images, but they are computed using numerical methods and as such do not necessarily represent reality. Therefore, we lack a reference to assess the quality of the reconstruction, for example by measuring the MRSA. This is why we use the relative reconstruction error as the criteria for the quality of a solution. This is a satisfying criteria, as illustrated in fig. 2.

In table 3, we report results from the experiments. We observe that, when $p > 1$, the result is always improved. When p is too large, however, the solution can be worse. We observe that the best p varies between the algorithms. For example, with Terrain, ALLS and SVCA perform best with $p = 1000$ while SSPA performs best with $p = 100$. This difference is expected, as the algorithms have different robustness to noise, they need to average on different numbers of data points. Also, different endmembers generally have different numbers of nearby points, so there is no value of p that would be always ideal for a given data set. Larger values of p seem to give more stable results, as it produces solutions with smaller deviations. SVCA outperforms ALLS in all cases. SSPA performance is comparable to SVCA, and generally produces a better result than the median of SVCA, but never better than the best result obtained by SVCA.

In a few cases, SSPA produces solutions with a large error, for example in San Diego for $p = 100$ and Terrain for $p = 1000$. We believe this behavior to originate from small groups points with a very large norm, that could correspond to a rare material or to interference.

Table 3: Relative reconstruction errors ($\min_{H\geq 0} \|X - WH\|_F / \|X\|_F$) resulting from the unmixing of hyperspectral images with ALLS, SVCA, and SSPA, with different values of parameter p. SVCA(1) and SSPA(1) are equivalent to VCA and SPA. For non-deterministic algorithms ALLS and SVCA, we show the minimum, median, standard deviation, and maximum of the error over 30 trials.

		SanDiego			Urban			Terrain		
	p	Min	Med ± std	Max	Min	Med ± std	Max	Min	Med ± std	Max
ALLS	1	4.72	5.60 ± 0.67	8.25	5.39	9.14 ± 1.93	12.26	3.94	4.88 ± 0.73	7.08
	100	**4.27**	**5.35** ± 1.72	10.91	**6.37**	**9.28** ± 3.18	19.40	**3.84**	4.87 ± 0.87	6.85
	1000	4.64	6.14 ± 1.12	8.68	6.78	9.71 ± 2.16	14.20	**3.84**	**4.71** ± 1.19	8.81
	2000	4.87	5.91 ± 1.62	11.79	6.96	9.93 ± 1.60	12.85	3.96	4.89 ± 0.88	7.63
	5000	5.51	7.42 ± 2.64	13.88	7.68	10.37 ± 1.89	14.98	4.28	5.26 ± 0.80	6.88
SVCA	1	3.95	5.42 ± 0.61	6.90	6.25	9.13 ± 1.78	12.23	4.03	5.11 ± 1.25	8.70
	100	**3.44**	4.92 ± 0.77	6.96	**5.08**	**6.10** ± 1.27	10.13	3.52	4.04 ± 0.67	6.52
	1000	3.82	4.95 ± 0.59	6.82	5.82	6.77 ± 1.23	10.84	**3.18**	**3.92** ± 0.38	4.70
	2000	3.73	**4.40** ± 0.51	5.81	5.66	6.36 ± 0.67	7.83	3.38	4.12 ± 0.45	4.95
	5000	4.01	4.66 ± 0.73	7.01	5.69	6.94 ± 1.21	11.74	3.70	4.19 ± 0.30	4.82
SSPA	1		5.90			9.46			5.01	
	100		9.29			6.65			**4.03**	
	1000		5.82			6.22			8.05	
	2000		**4.32**			6.11			7.86	
	5000		4.65			**5.91**			5.38	

In fig. 6, we compare SVCA and SSPA using either the median or the mean for the unmixing of the hyperspectral image Urban, with a varying p. We observe that $p > 1$ always improves the results. Also, the median aggregation almost always gives better results than the mean. While the curves are not as regular as with synthetic data, we observe a similar tendency that the solution improves when p grows, until a certain point or zone after which it worsens again. However, SSPA-med has an irregular behaviour for $p = 200$. This can be explained by the fact that SSPA is a greedy algorithm, so if it makes a bad choice in the first iterations, it will likely never compensate. Also, it is deterministic, so the error is not averaged over several runs.

In fig. 7, we show the abundances maps corresponding to the unmixing of Urban. They indicate the proportion of every of the 6 extracted endmembers in the pixels of the image. We see that the smoothed algorithms perform a better separation than the non-smoothed ones. For example, the fourth endmember extracted by SVCA and SSPA corresponds to grass, and it is well separated by these algorithms, while VCA and SPA mix it with asphalt and dirt. The second endmember extracted by SVCA and SSPA corresponds to metallic rooftops, and it is well separated while VCA mixes it with other materials and SPA does not clearly identify it and produces a blurred picture.

In appendix A, we provide the results for the hyperspectral images Terrain and San Diego.

5.3 Discussion

SVCA allows to generate different solutions, among which the best solution w.r.t. reconstruction error can be found. Therefore, in practice and when time and resources allow, we recommend running SSPA once and SVCA several times, with different values of p, and keep the best solution.

Apart from the average and the median, other aggregation methods could perform better depending

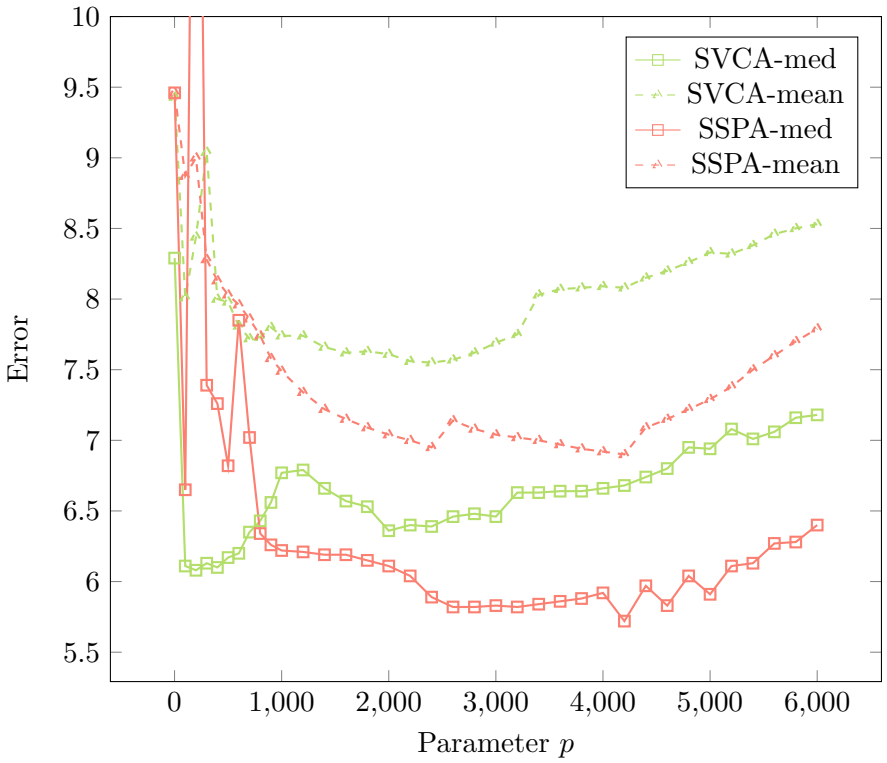


Figure 6: Results of the unmixing of the hyperspectral image Urban. Values for SVCA are the medians over 30 trials. One point is out of the plot; for $p = 200$, SVCA-med has an error of 14.55%.

on the noise law and data set at hand; see for instance [17], in which the authors use an aggregation on manifold in the different context of sparse matrix factorization to better take into account the structure of the columns of W.

Choosing a value for the parameter p is crucial and not trivial; a strategy to determine it is an interesting direction of further research. It could also be useful to consider a different value of p for every endmember, as the number of proximal latent points typically varies between materials. Also, the proximal latent points assumption could be used to generalize other pure-pixel search algorithms.

6 Conclusion

In this work, we introduced the smoothed separable NMF model, that strengthen the separability assumption by assuming the presence of several near-pure data points. Inspired by the existing algorithm ALLS, we developed smoothed variants of two separable NMF algorithms, namely VCA and SPA. Empirically, we showed that our smoothed methods outperform both the non-smoothed ones and ALLS, for both synthetic data sets and for the unmixing of real-world hyperspectral images. This shows that the proximal latent points assumption is verified in hyperspectral images, and that smoothed separable NMF algorithms are a more effective tool for hyperspectral unmixing.

References

[1] Abdolali, M., Gillis, N.: Simplex-structured matrix factorization: Sparsity-based identifiability and provably correct algorithms. SIAM Journal on Mathematics of Data Science **3**(2), 593–623 (2021)

[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: Proceedings of the 30th International Conference on Machine Learning, pp. 280–288 (2013)

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

[5] Bakshi, A., Bhattacharyya, C., Kannan, R., Woodruff, D.P., Zhou, S.: Learning a latent simplex in input sparsity time. In: Proceedings of the International Conference on Learning Representations (ICLR) (2021)

[6] Bhattacharyya, C., Kannan, R.: Finding a latent k–simplex in $o^*(k \cdot \text{nnz}(\text{data}))$ time via subset smoothing. In: Proceedings of the Fourteenth Annual ACM-SIAM Symposium on Discrete Algorithms, pp. 122–140. SIAM (2020)

[7] Bioucas-Dias, J.M., Plaza, A., Dobigeon, N., Parente, M., Du, Q., Gader, P., Chanussot, J.: Hyperspectral unmixing overview: Geometrical, statistical, and sparse regression-based approaches. IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing **5**(2), 354–379 (2012)

[8] Boardman, J.W., Kruse, F.A., Green, R.O.: Mapping target signatures via partial unmixing of AVIRIS data. In: Proc. Summary JPL Airborne Earth Science Workshop, Pasadena, CA, pp. 23–26 (1995)

[9] Fu, X., Ma, W.K., Huang, K., Sidiropoulos, N.D.: Blind separation of quasi-stationary sources: Exploiting convex geometry in covariance domain. IEEE Transactions on Signal Processing **63**(9), 2306–2320 (2015)

[10] Ge, R., Zou, J.: Intersecting faces: Non-negative matrix factorization with new guarantees. In: Proceedings of the 32nd International Conference on Machine Learning, pp. 2295–2303 (2015)

[11] Gillis, N.: Robustness analysis of Hottopixx, a linear programming model for factoring nonnegative matrices. SIAM Journal on Matrix Analysis and Applications **34**(3), 1189–1212 (2013)

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

[13] Gillis, N.: Nonnegative Matrix Factorization. SIAM, Philadelphia (2020)

[14] Gillis, N., Luce, R.: Robust near-separable nonnegative matrix factorization using linear optimization. Journal of Machine Learning Research **15**(1), 1249–1280 (2014)

[15] Gillis, N., Vavasis, S.A.: Fast and robust recursive algorithms for separable nonnegative matrix factorization. IEEE Transactions on Pattern Analysis and Machine Intelligence **36**(4), 698–714 (2013)

[16] Jiang, J.H., Liang, Y., Ozaki, Y.: Principles and methodologies in self-modeling curve resolution. Chemometrics and Intelligent Laboratory Systems **71**(1), 1–12 (2004)

[17] Kervazo, C., Liaudat, T., Bobin, J.: Faster and better sparse blind source separation through mini-batch optimization. Digital Signal Processing **106**, 102,827 (2020)

[18] Lin, C.H., Chi, C.Y., Wang, Y.H., Chan, T.H.: A fast hyperplane-based minimum-volume enclosing simplex algorithm for blind hyperspectral unmixing. IEEE Transactions on Signal Processing **64**(8), 1946–1961 (2015)

[19] Lin, C.H., Ma, W.K., Li, W.C., Chi, C.Y., Ambikapathi, A.: Identifiability of the simplex volume minimization criterion for blind hyperspectral unmixing: The no-pure-pixel case. IEEE Transactions on Geoscience and Remote Sensing **53**(10), 5530–5546 (2015)

[20] Lin, C.H., Wu, R., Ma, W.K., Chi, C.Y., Wang, Y.: Maximum volume inscribed ellipsoid: A new simplex-structured matrix factorization framework via facet enumeration and convex optimization. SIAM Journal on Imaging Sciences **11**(2), 1651–1679 (2018)

[21] 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)

[22] Nascimento, J.M., Bioucas-Dias, J.M.: Hyperspectral unmixing based on mixtures of dirichlet components. IEEE Transactions on Geoscience and Remote Sensing **50**(3), 863–878 (2011)

[23] 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)

[24] Recht, B., Re, C., Tropp, J., Bittorf, V.: Factoring nonnegative matrices with linear programs. In: Advances in Neural Information Processing Systems (NIPS), pp. 1214–1222 (2012)

[25] Winter, M.E.: N-FINDR: An algorithm for fast autonomous spectral end-member determination in hyperspectral data. In: Proc. SPIE Conf. Imaging Spectrometry V, vol. 3753, pp. 266–276. International Society for Optics and Photonics (1999)

[26] Wu, R., Ma, W.K., Li, Y., So, A.M.C., Sidiropoulos, N.D.: Probabilistic simplex component analysis. arXiv preprint arXiv:2103.10027 (2021)

[27] Zhu, F.: Hyperspectral unmixing: Ground truth labeling, datasets, benchmark performances and survey. preprint arXiv:1708.05125 (2017)

A Additional experiments

In this appendix, we provide the results from our experiments on the hyperspectral images Terrain (figs. 8 and 10) and San Diego (figs. 9 and 11).

(a) VCA, error= 6.24%

(b) SVCA p=200, error= 5.24%

(c) SPA, error= 9.46%

(d) SSPA p=4200, error= 5.72%

Figure 7: Abundance maps of the unmixing of the Urban hyperspectral images (that is, reshaped rows of H) with different algorithms. Endmembers have been reordered for easier comparison. Parameters p have been chosen as the best from fig. 6. Error corresponds to $\min_{H \geq 0} \|X - WH\|_F / \|X\|_F$. For VCA and SVCA, we show the best solution over 30 trials.

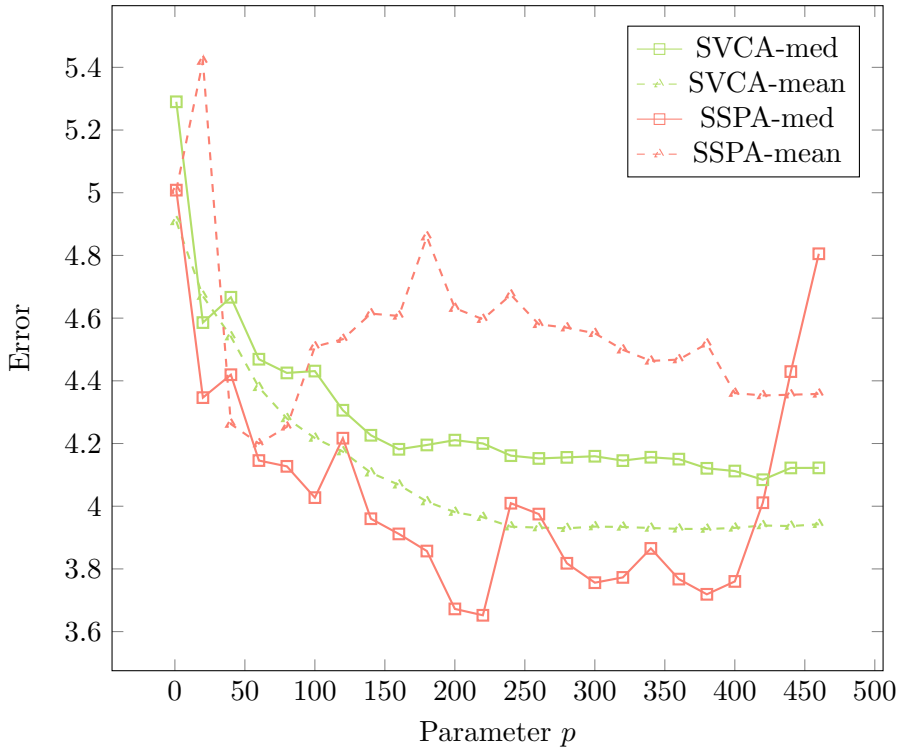


Figure 8: Results of the unmixing of the hyperspectral image Terrain. Values for SVCA are the medians over 30 trials.

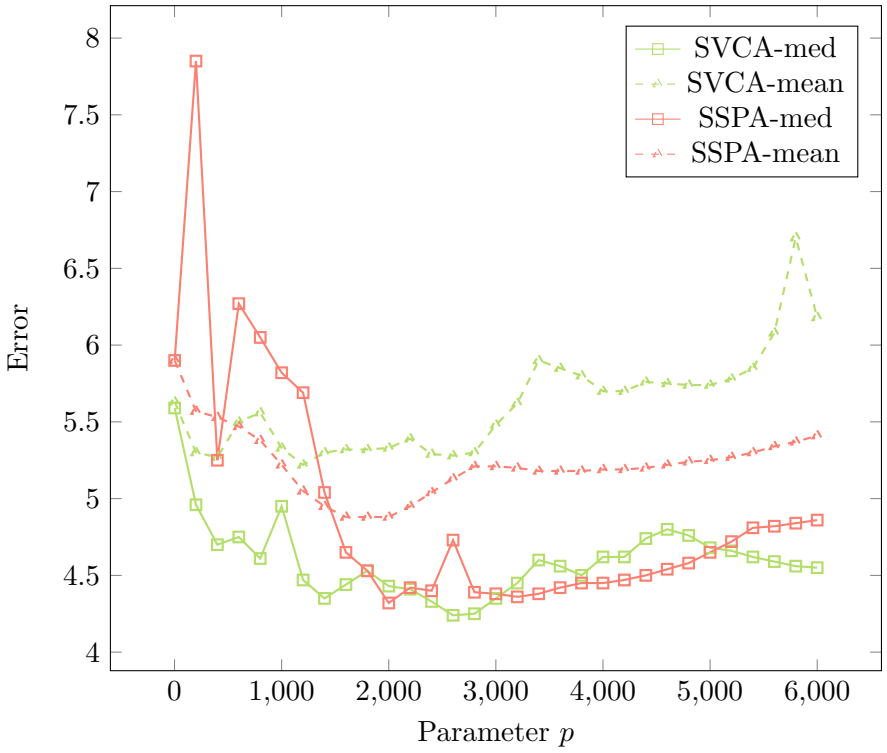
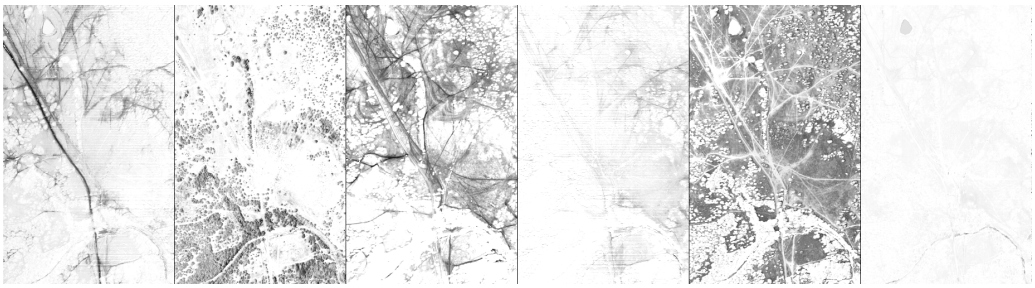
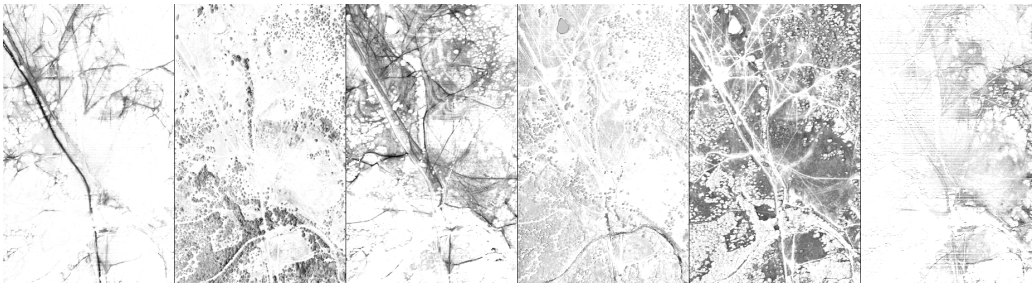


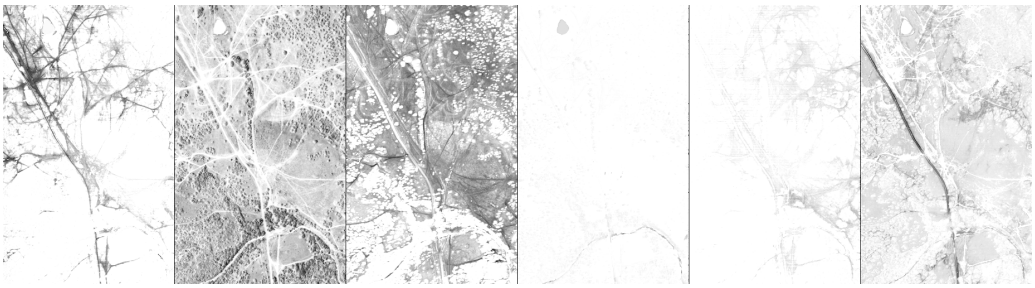
Figure 9: Results of the unmixing of the hyperspectral image San Diego. Values for SVCA are the medians over 30 trials.



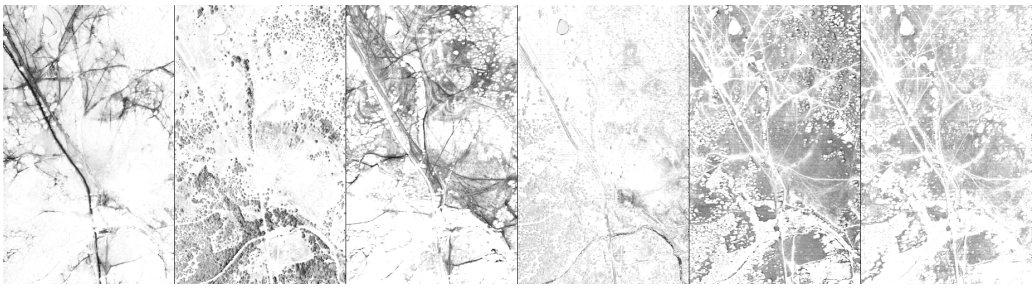
(a) VCA, error= 4.03%



(b) SVCA p=420, error= 3.25%



(c) SPA, error= 5.01%

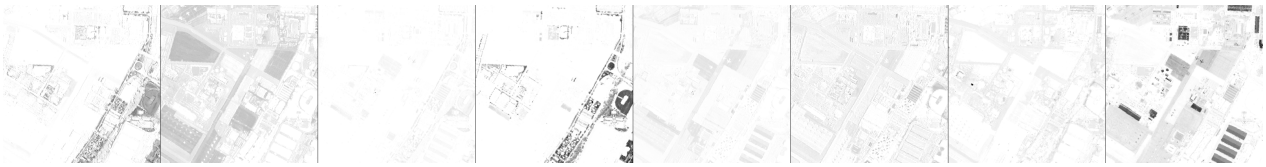


(d) SSPA p=220, error= 3.65%

Figure 10: Abundance maps of the unmixing of the Terrain hyperspectral images (that is, reshaped rows of H) with different algorithms. Endmembers have been reordered for easier comparison. Parameters p have been chosen as the best from fig. 8. Error corresponds to $\min_{H \geq 0} \|X - WH\|_F / \|X\|_F$. For VCA and SVCA, we show the best solution over 30 trials.

(a) VCA, error= 3.95%

(b) SVCA p=2600, error= 3.74%



(c) SPA, error= 5.90%

(d) SSPA p=2000, error= 4.32%

Figure 11: Abundance maps of the unmixing of the San Diego hyperspectral images (that is, reshaped rows of H) with different algorithms. Endmembers have been reordered for easier comparison. Parameters p have been chosen as the best from fig. 9. Error corresponds to $\min_{H \geq 0} \|X - WH\|_F / \|X\|_F$. For VCA and SVCA, we show the best solution over 30 trials.