Hierarchical Clustering of Hyperspectral Images using Rank-Two Nonnegative Matrix Factorization

Nicolas Gillis* Da Kuang† Haesun Park†

Abstract

In this paper, we design a hierarchical clustering algorithm for high-resolution hyperspectral images. At the core of the algorithm, a new rank-two nonnegative matrix factorizations (NMF) algorithm is used to split the clusters, which is motivated by convex geometry concepts. The method starts with a single cluster containing all pixels, and, at each step, (i) selects a cluster in such a way that the error at the next step is minimized, and (ii) splits the selected cluster into two disjoint clusters using rank-two NMF in such a way that the clusters are well balanced and stable. The proposed method can also be used as an endmember extraction algorithm in the presence of pure pixels. The effectiveness of this approach is illustrated on several synthetic and real-world hyperspectral images, and shown to outperform standard clustering techniques such as k-means, spherical k-means and standard NMF.

Keywords. nonnegative matrix factorization, rank-two approximation, convex geometry, high-resolution hyperspectral images, hierarchical clustering, endmember extraction algorithm.

1 Introduction

A hyperspectral image (HSI) is a set of images taken at many different wavelengths (usually between 100 and 200), not just the usual three visible bands of light (red at 650nm, green at 550nm, and blue at 450nm). An important problem in hyperspectral imaging is blind hyperspectral unmixing (blind HU): given a HSI, the goal is to recover the constitutive materials present in the image (the *endmembers*) and the corresponding abundance maps (that is, determine which pixel contains which endmember and in which quantity). Blind HU has many applications such as quality control in the food industry, analysis of the composition of chemical compositions and reactions, monitoring the development and health of crops, monitoring polluting sources, military surveillance, and medical imaging; see, e.g., [8] and the references therein.

Let us associate a matrix $M \in \mathbb{R}^{m \times n}_{+}$ to a given HSI with m spectral bands and n pixels as follows: the (i, j)th entry $M(i, j)$ of matrix M is the reflectance of the jth pixel at the ith wavelength (that is, the fraction of incident light that is reflected by the ith pixel at the jth wavelength). Hence each column of M is equal to the spectral signature of a pixel while each row is a vectorized image at a given wavelength. The linear mixing model (LMM) assumes that the spectral signature of each pixel is a linear combination of the spectral signatures of the endmembers, where the weights in the linear combination are the abundances of each endmember in that pixel. For example, if a pixel contains 40% of aluminum and 60% of copper, then its spectral signature will be 0.4 times the spectral signature of the aluminum plus 0.6 times the spectral signature of the copper. This is a rather natural model:

*Department of Mathematics and Operational Research, Faculté Polytechnique, Université de Mons, Rue de Houdain 9, B-7000 Mons, Email: nicolas.gillis@umons.ac.be. This work was carried on when NG was a postdoctoral researcher of the fonds de la recherche scientifique (F.R.S.-FNRS).

†School of Computational Science and Engineering, Georgia Institute of Technology, Atlanta, GA 30332-0765, USA. Emails: {da.kuang,hpark}@cc.gatech.edu. The work of these authors was supported in part by the National Science Foundation (NSF) grants CCF-0808863 and CCF-0732318.

we assume that 40% of the light is reflected by the aluminum while 60% is by the copper, while non-linear effects are neglected (such as the light interacting with multiple materials before reflecting off, or atmospheric distortions).

Assuming the image contains r endmembers, and denoting $W(:,k) \in \mathbb{R}^m$ $(1 \leq k \leq r)$ the spectral signatures of the endmembers, the LMM can be written as

$$M(:,j) = \sum_{k=1}^{r} W(:,k)H(k,j) \quad 1 \leq j \leq n,$$

where $H(k,j)$ is the abundance of the kth endmember in the jth pixel, hence $\sum_{k=1}^{r} H(k,j) = 1$ for all j, which is referred to as the abundance sum-to-one constraint. Under the LMM and given a HSI M, blind HU amounts to recovering the spectral signatures of the endmembers (matrix W) along with the abundances (matrix H). Since all matrices involved M, W and H are nonnegative, blind HU under the LMM is equivalent to nonnegative matrix factorization (NMF): Given a nonnegative matrix $M \in \mathbb{R}^{m \times n}_{+}$ and a factorization rank r, find two nonnegative matrices $W \in \mathbb{R}^{m \times r}_{+}$ and $H \in \mathbb{R}^{r \times n}_{+}$ such that $M \approx WH$. Unfortunately, NMF is NP-hard [35] and highly ill-posed [16]. Therefore, in practice, it is crucial to use the structure of the problem at hand to develop efficient numerical schemes for blind HU. This is usually achieved using additional constraints or regularization terms in the objective function, e.g., the sum-to-one constraint on the columns of H (see above), sparsity of the abundance matrix H (most pixels contain only a few endmembers), piecewise smoothness of the spectral signatures $W(:,k)$ [24], and spatial information [38] (that is, neighboring pixels are more likely to contain the same materials). Although these priors make the corresponding NMF problems more well-posed, the underlying optimization problems to be solved are still computationally difficult (and only local minimum are usually obtained). We refer the reader to the survey [8] for more details about blind HU.

In this paper, we make an additional assumption, namely that *most pixels are dominated mostly by one endmember*, and our goal is to cluster the pixels accordingly. In fact, clustering the pixels of a HSI only makes sense for relatively high resolution images. For such images, it is often assumed that, for each endmember, there exists at least one pixel containing only that endmember, that is, for all $1 \leq k \leq r$ there exists j such that $M(:,j) = W(:,k)$. This is the so-called *pure-pixel assumption*. The pure-pixel assumption is equivalent to the separability assumption (see [21] and the references therein) which makes the corresponding NMF problem tractable, even in the presence of noise [5]. Hence, blind HU can be solved efficiently under the pure-pixel assumption. Mathematically, a matrix $M \in \mathbb{R}^{m \times n}$ is r-separable if it can be written as

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

where $W \in \mathbb{R}^{m \times r}$, $H' \geq 0$ and Π is a permutation matrix. If M is a HSI, we have, as before, that

- ◇ The number r is the number of endmembers present in the HSI.
- ◇ Each column of W is the spectral signature of an endmember.
- ◇ Each column of H is the abundance vector of a pixel. More precisely, the entry $H(i,j)$ is the abundance of the ith endmember in the jth pixel.

Because the column of H sum to one, each column of M belongs to the convex hull of the columns of W, that is, $\text{conv}(M) \subseteq \text{conv}(W)$. The pure-pixel assumption requires that $\text{conv}(M) = \text{conv}(W)$, that is, that the vertices of the convex hull of the columns of M are the columns of W; see the top of Figure 1 for an illustration in the rank-three case. Hence, the separable NMF problem (or, equivalently, blind HU under the LMM and the pure-pixel assumption) reduces to identifying the vertices of the convex hull of the columns of M. However, in noisy settings, this problem becomes more difficult, and although some robust algorithms have been proposed recently (see, e.g., [17] and the references therein), they are typically rather sensitive to noise and outliers.

Motivated by the fact that in high-resolution HSI's, most pixels are mostly dominated by one endmember, we develop in this paper a practical and theoretically well-founded hierarchical clustering technique. Hierarchical clustering based on NMF has been shown to be faster than flat clustering and can often achieve similar or even better clustering quality [28]. At the core of the algorithm is the use of rank-two NMF that splits a cluster into two disjoint clusters. We study the unique property of rank-two NMF as opposed to a higher-rank NMF. We also propose an efficient algorithm for rank-two NMF so that the overall problem of hierarchical clustering of HSI's can be efficiently solved.

The paper is organized as follows. In Section 2, we describe our hierarchical clustering approach (see Algorithm 1 referred to as H2NMF). At each step, a cluster is selected (Section 2.1) and then split into two disjoint clusters (Section 2.2). The splitting procedure has a rank-two NMF algorithm at its core which is described in Section 2.3 where we also provide some sufficient conditions under which the proposed algorithm recovers an optimal solution. In Section 2.4, we analyze the geometric properties of the hierarchical clustering. In Section 3, we show that it outperforms k-means, spherical k-means (either if they are used in a hierarchical manner, or directly on the full image) and standard NMF on synthetic and real-world HSI's, being more robust to noise, outliers and absence of pure pixels. We also show that it can be used as an endmember extraction algorithms and outperforms vertex component analysis (VCA) [31] and the successive projection algorithm (SPA) [3], two standard and widely used techniques.

2 Hierarchical Clustering for HSI's using Rank-Two NMF

As mentioned in the introduction, for high-resolution HSI, one can assume that most pixels contain mostly one material. Hence, given a high-resolution HSI with r endmembers, it makes sense to cluster the pixels into r clusters, each cluster corresponding to one endmember. Mathematically, given the HSI $M \in \mathbb{R}^{m \times n}_{+}$, we want to find r disjoint clusters $\mathcal{K}_k \subset \{1, 2, \dots n\}$ for $1 \leq k \leq r$ so that $\cup_{k=1,2,\dots,r} \mathcal{K}_k = \{1, 2, \dots n\}$ and so that all pixels in $\mathcal{K}_k$ are dominated by the same endmember.

In this paper, we assume the number of endmembers is known in advance. In fact, the problem of determining the number of endmembers (also known as model order selection) is nontrivial and out of the scope of this paper; see, e.g., [7]. However, a crucial advantage of our approach is that it decomposes the data hierarchically and hence provides the user with a hierarchy of materials (see, e.g., Figures 7 and 14). In particular, the algorithm does not need to be rerun from scratch if the number of clusters required by the user is modified.

In this section, we propose an algorithm to cluster the pixels of a HSI in a hierarchical manner. More precisely, at each step, given the current set of clusters $\{\mathcal{K}_k\}_{k=1}^{p}$, we select one of the clusters and split it into two disjoint clusters. Hierarchical clustering is a standard technique in data mining that organizes a data set into a tree structure of items. It is widely used in text analysis for efficient browsing and retrieval [37, 28, 9], as well as exploratory genomic study for grouping genes participating in the same pathway [12]. Another example is to segment an image into a hierarchy of regions according to different cues in computer vision such as contours and textures [4]. In contrast to image segmentation problems, our focus is to obtain a hierarchy of materials from HSI's taken at hundreds of wavelengths instead of the three visible wavelengths.

At each step of a hierarchical clustering technique, one has to address the following two questions:

1. Which cluster should be split next?

2. How do we split the selected cluster?

These two building blocks for our hierarchical clustering technique for HSI's are described in the following sections.

2.1 Selecting the Leaf Node to Split

Eventually, we want to cluster the pixels into r disjoint clusters $\{\mathcal{K}_k\}_{k=1}^r$, each corresponding to a different endmember. Therefore, each submatrix $M(:,\mathcal{K}_k)$ should be close to a rank-one matrix since for all $j \in \mathcal{K}_k$, we should have $M(:,j) \approx W(:,k)$, possibly up to a scaling factor (e.g., due to different illumination conditions in the image), where $W(:,k)$ is the spectral signature of the endmember corresponding to the cluster $\mathcal{K}_k$. In particular, in ideal conditions, that is, each pixel contains exactly one material and no noise is present, $M(:,\mathcal{K}_k)$ is a rank-one matrix. Based on this observation, we define the error E_k corresponding to each cluster as follows

$$E_k \quad = \min_{X,\mathrm{rank}(X)=1} ||M(:,\mathcal{K}_k) - X||_F^2 \quad = \quad ||M(:,\mathcal{K}_k)||_F^2 - \sigma_1^2(M(:,\mathcal{K}_k)).$$

We also define the total error $E = \sum_{k=1}^r E_k$. If we decide to split the kth cluster $\mathcal{K}_k$ into $\mathcal{K}_k^1$ and $\mathcal{K}_k^2$, the error corresponding to the columns in $\mathcal{K}_k$ is given by

$$\begin{aligned}
\left(||M(:,\mathcal{K}_k^1)||_F^2 - \sigma_1^2(M(:,\mathcal{K}_k^1))\right) + \left(||M(:,\mathcal{K}_k^2)||_F^2 - \sigma_1^2(M(:,\mathcal{K}_k^2))\right)& \\
= \left(||M(:,\mathcal{K}_k^1)||_F^2 + ||M(:,\mathcal{K}_k^2)||_F^2\right) - \left(\sigma_1^2(M(:,\mathcal{K}_k^1)) + \sigma_1^2(M(:,\mathcal{K}_k^2))\right)& \\
= ||M(:,\mathcal{K}_k)||_F^2 - \left(\sigma_1^2(M(:,\mathcal{K}_k^1)) + \sigma_1^2(M(:,\mathcal{K}_k^2))\right).&
\end{aligned}$$

(Note that the error corresponding to the other clusters is unchanged.) Hence, if the kth cluster is split, the total error E will be reduced by $\sigma_1^2(M(:,\mathcal{K}_k^1)) + \sigma_1^2(M(:,\mathcal{K}_k^2) - \sigma_1^2(M(:,\mathcal{K}_k))$. Therefore, we propose to split the cluster k for which $\sigma_1^2(M(:,\mathcal{K}_k^1)) + \sigma_1^2(M(:,\mathcal{K}_k^2) - \sigma_1^2(M(:,\mathcal{K}_k))$ is maximized: this leads to the largest possible decrease in the total error E at each step.

2.2 Splitting a Leaf Node

For the splitting procedure, we propose to use rank-two NMF. Given a nonnegative matrix $M \in \mathbb{R}_+^{m \times n}$, rank-two NMF looks for two nonnegative matrices $W \in \mathbb{R}_+^{m \times 2}$ and $H \in \mathbb{R}_+^{2 \times n}$ such that $WH \approx M$. The motivation for this choice is two-fold:

- ◇ NMF corresponds to the linear mixing model for HSI's (see Introduction), and
- ◇ Rank-two NMF can be solved efficiently, avoiding the use of an iterative procedure as in standard NMF algorithms. In Section 2.3, we propose a new rank-two NMF algorithm using convex geometry concepts from HSI; see Algorithm 4.

Suppose for now we are given a rank-two NMF (W,H) of M. Such a factorization is a two-dimensional representation of the data; more precisely, it projects the columns of M onto a two-dimensional pointed cone generated by the columns of W. Hence, a naive strategy to cluster the columns of M is to choosing the clusters as follows

$$C_1 = \{\ i \mid H(1,i) \geq H(2,i)\ \} \quad \text{and} \quad C_2 = \{\ i \mid H(1,i) < H(2,i)\ \}.$$

Defining the vector $x \in [0,1]^n$ as

$$x(i) = \frac{H(1,i)}{H(1,i) + H(2,i)} \quad \text{for} \quad 1 \leq i \leq n,$$

the above clustering assignment is equivalent to taking

$$C_1 = \{\ i \mid x_i \geq \delta\ \} \quad \text{and} \quad C_2 = \{\ i \mid x_i < \delta\ \}, \tag{2.1}$$

with $\delta = 0.5$. However, the choice of $\delta = 0.5$ is by no means optimal, and often leads to a rather poor separation. In particular, if an endmember is located exactly between the two extracted endmembers, the corresponding cluster is likely to be divided into two which is not desirable (see Figure 1). In this section, we present a simple way to tune the threshold $\delta \in [0, 1]$ in order to obtain, in general, significantly better clusters C_1 and C_2.

Let us define the empirical cumulative distribution of x as follows

$$\hat{F}_X(\delta) = \frac{1}{n} \left|\{ \, i \mid x_i \leq \delta \, \}\right| \in [0,1], \qquad \text{for } \delta \in [0,1].$$

By construction, $\hat{F}_X(0) = 0$ and $\hat{F}_X(1) = 1$. Let us also define

$$\hat{G}_X(\delta) = \frac{1}{n(\bar{\delta} - \underline{\delta})} \left|\{ \, i \mid \underline{\delta} = \max(0, \delta - \hat{\delta}) \leq x_i \leq \min(1, \delta + \hat{\delta}) = \bar{\delta} \, \}\right| \in [0,1],$$

for $\delta \in [0, 1]$, and $\hat{\delta} \in (0, 0.5)$ is a small parameter. The function $\hat{G}_X(\delta)$ accounts for the number of points in a small interval around δ. Note that, assuming uniform distribution in the interval $[0, 1]$, the expected value of $\hat{G}_X(\delta)$ is equal to one. In fact, since the entries of x are in the interval $[0, 1]$, the expected number of data points in an interval of length L is nL. In this work, we use $\hat{\delta} = 0.05$.

Given δ, we obtain two clusters C_1 and C_2; see Equation (2.1). We propose to choose a value of δ such that

1. The clusters are balanced, that is, the two clusters contain, if possible, roughly the same number of elements. Mathematically, we would like that $\hat{F}_X(\delta) \approx 0.5$.

2. The clustering is stable, that is, if the value of δ is slightly modified, then only a few points are transfered from one cluster to the other. Mathematically, we would like that $\hat{G}_X(\delta) \approx 0$.

We propose to balance these two goals by choosing δ that minimizes the following criterion:

$$g(\delta) = \underbrace{-\log\left(\hat{F}_X(\delta)\left(1 - \hat{F}_X(\delta)\right)\right)}_{\text{balanced clusters}} + \underbrace{\exp\left(\hat{G}_X(\delta)\right)}_{\text{stable clusters}}. \tag{2.2}$$

The first term avoids skewed classes, while the second promotes a stable clustering. Note that the two terms are somewhat well-balanced since, for $\hat{F}_X(\delta) \in [0.1, 0.9]$,

$$-\log\left(\hat{F}_X(\delta)\left(1 - \hat{F}_X(\delta)\right)\right) \leq 2.5,$$

and the expected value of $\hat{G}_X(\delta)$ is one (see above). Note that depending on the application at hand, the two terms of $g(\delta)$ can be balanced in different ways; for example, if one wants to allow very small clusters to be extracted, then the first term of $g(\delta)$ should be given less importance.

Remark 1 (Sensitivity to δ)**.** *The splitting procedure is clearly very sensitive to the choice of δ. For example, as described above, choosing $\delta = 0.5$ can give very poor results. However, if the function $g(\delta)$ is chosen in a sensible way, then the corresponding splitting procedure generates in general good clusters. For example, we had first run all the experiments from Section 3 selecting δ minimizing the function*

$$g(\delta) = 4\left(\hat{F}_X(\delta) - 0.5\right)^2 + \left(\hat{G}_X(\delta)\right)^2,$$

and it gave very similar results (sometimes slightly better, sometimes slightly worse). The advantage of the function (2.2) *is that it makes sure no empty cluster is generated (since it goes to infinity when $\hat{F}_X(\delta)$ goes to 0 or 1).*

Remark 2 (Sensitivity to $\hat{\delta}$)**.** *The parameter $\hat{\delta}$ is the window size where the stability of a given clustering is evaluated. For δ corresponding to a stable cluster (that is, only a few pixels are transferred from one cluster to the other if δ is slightly modified), $\hat{G}_X(\delta)$ will remain small when $\hat{\delta}$ is slightly modified. For the considered data sets, most clusterings are stable (because the data is in fact constituted of several clusters of points) hence in that case the splitting procedure does not seem to be very sensitive to $\hat{\delta}$ as long as it is in a reasonable range. In fact, we have also run the numerical experiments for $\hat{\delta} = 0.01$ and $\hat{\delta} = 0.1$ and it gave very similar results (in particular, for the Urban, San Diego, Terrain and Cuprite HSI's from Section 3.4, it is hardly possible to distinguish the solutions with the naked eye).*

Figure 1 illustrates the geometric insight behind the splitting procedure in the case $r = 3$ (see also Section 2.4), while Algorithm 1 gives a pseudo-code of the full hierarchical procedure.

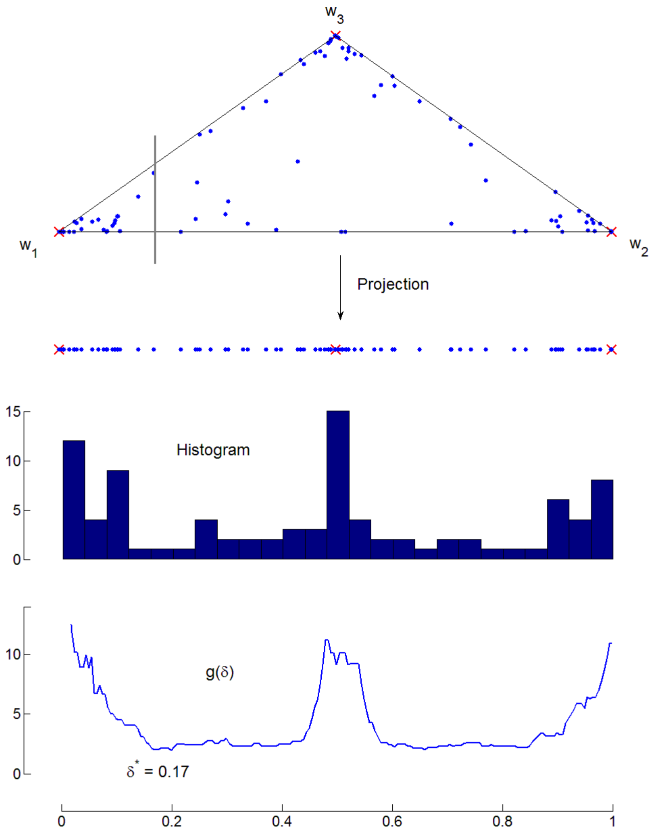

Figure 1: Illustration of the splitting technique based on rank-two NMF.

2.3 Rank-Two NMF for HSI's

In this section, we propose a simple and fast algorithm for the rank-two NMF problem tailored for HSI's (Section 2.3.1). Then, we discuss some sufficient conditions for the algorithm to be optimal (Section 2.3.2).

2.3.1 Description of the Algorithm

When a nonnegative matrix $M \in \mathbb{R}^{m \times n}_+$ has rank two, Thomas has shown [34] that finding two nonnegative matrices $(W, H) \in \mathbb{R}^{m \times 2}_+ \times \mathbb{R}^{2 \times n}_+$ such that $M = WH$ is always possible (see also [11]). This can be explained geometrically as follows: viewing columns of M as points in $\mathbb{R}^m_+$, the fact that M has rank two implies that the set of its columns belongs to a two-dimensional subspace. Furthermore,

Algorithm 1 Hierachical Clustering of a HSI based on Rank-Two NMF (H2NMF)

```
Input: A HSI $M \in \mathbb{R}^{m \times n}_{+}$ and the number $r$ of clusters to generate.
Output: Set of disjoint clusters $\mathcal{K}_i$ for $1 \leq i \leq r$ with $\cup_i \mathcal{K}_i = \{1, 2, \dots, n\}$.
% Initialization
$\mathcal{K}_1 = \{1, 2, \dots, n\}$ and $\mathcal{K}_i = \emptyset$ for $2 \leq i \leq r$.
$\left(\mathcal{K}_1^1, \mathcal{K}_1^2\right)$ = splitting$(M, \mathcal{K}_1)$. % See Algorithm 2 and Section 2.2
$\mathcal{K}_i^1 = \mathcal{K}_i^2 = \emptyset$ for $2 \leq i \leq r$.
for $k = 2 : r$ do
    % Select the cluster to split; see Section 2.1
    Let $j = \text{argmax}_{i=1,2,\dots r}\, \sigma_1^2(M(:, \mathcal{K}_i^1)) + \sigma_1^2(M(:, \mathcal{K}_i^2) - \sigma_1^2(M(:, \mathcal{K}_i))$.
    % Update the clustering
    $\mathcal{K}_j = \mathcal{K}_j^1$ and $\mathcal{K}_k = \mathcal{K}_j^2$.
    % Split the new clusters (Algorithm 2)
    $\left(\mathcal{K}_j^1, \mathcal{K}_j^2\right)$ = splitting$(M, \mathcal{K}_j)$ and $\left(\mathcal{K}_k^1, \mathcal{K}_k^2\right)$ = splitting$(M, \mathcal{K}_k)$.
end for
```

Algorithm 2 Splitting of a HSI using Rank-Two NMF

```
Input: A HSI $M \in \mathbb{R}^{m \times n}_{+}$ and a subset $\mathcal{K} \subseteq \{1, 2, \dots, n\}$.
Output: Set of two disjoint clusters $\mathcal{K}^1$ and $\mathcal{K}^2$ with $\mathcal{K}_1 \cup \mathcal{K}_2 = \mathcal{K}$.
Let $(W, H)$ be the rank-two NMF of $M(:, \mathcal{K})$ computed by Algorithm 4.
Let $x(i) = \frac{H(1,i)}{H(1,i)+H(2,i)}$ for $1 \leq i \leq |\mathcal{K}|$.
Compute $\delta^*$ as the minimum of $g(\delta)$ defined in (2.2).
$\mathcal{K}^1 = \{\ \mathcal{K}(i) \mid x(i) \geq \delta^*\ \}$ and $\mathcal{K}^2 = \{\ \mathcal{K}(i) \mid x(i) < \delta^*\ \}$.
```

because these columns are nonnegative, they belong to a two-dimensional pointed cone, see Figure 2. Since such a cone is always spanned by two extreme vectors, this implies that all columns of M can be represented exactly as nonnegative linear combinations of two nonnegative vectors, and therefore the exact NMF is always possible[1] for $r = 2$. Moreover, these two extreme columns can easily be identified. For example, if the columns of M are normalized so that their entries sum to one, then the columns of M belong to a line segment and it is easy to detect the two vertices. This can be

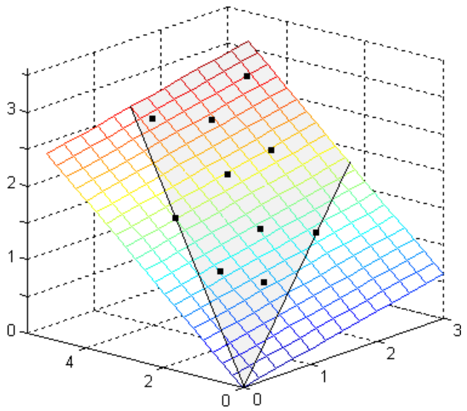

Figure 2: Illustration of exact NMF for a rank-two 3-by-10 nonnegative matrix [15, p.24].

[1] The reason why this property no longer holds for higher values of the rank r of matrix M is that a r-dimensional cone is not necessarily spanned by a set of r vectors when $r > 2$.

done for example using any endmember extraction algorithm under the linear mixing model and the pure-pixel assumption since they aim to detect the vertices (corresponding to the endmembers) of a convex hull of a set of points (see Introduction). In this paper, we use the successive projection algorithm (SPA) [3] which is a highly efficient and widely used algorithm; see Algorithm 3. Moreover, it has been shown to be robust to any small perturbation of the input matrix [21]. Note that SPA is closely related to the automatic target generation process algorithm (ATGP) [33] and the successive volume maximization algorithm (SVMAX) [10]; see [30] for a survey about these methods. Note that it would be possible to use more sophisticated endmember extraction algorithms for this step, e.g., RAVMAX [1] or WAVMAX [10] which are more robust variants of SPA (although computationally much more expensive).

Algorithm 3 Successive Projection Algorithm (SPA) [3, 21]

Input: Separable matrix $M = W[I_r, H']\Pi$ where $H' \geq 0$, the sum of the entries of each column of H' is smaller than one, W is full rank and Π is a permutation, and the number r of columns to be extracted.

Output: Set of indices K such that $M(:,K) = W$ (up to permutation).

```
Let $R = M$, $K = \{\}$.
for $i = 1 : r$ do
    $k = \text{argmax}_j \, ||R_{:j}||_2$.
    $R \leftarrow \left( I - \frac{R_{:k}R_{:k}^T}{||R_{:k}||_2^2} \right) R$.
    $K = K \cup \{k\}$.
end for
```

We can now describe our proposed rank-two NMF algorithm for HSI: It first projects the columns of M into a two-dimensional linear space using the SVD (note that if the rank of input matrix is two, this projection step is exact), then identifies two important columns with SPA and projects them onto the nonnegative orthant, and finally computes the optimal weights solving a nonnegative least squares problem (NNLS); see Algorithm 4.

Algorithm 4 Rank-Two NMF for HSI's

Input: A nonnegative matrix $M \in \mathbb{R}^{m \times n}_+$.

Output: A rank-two NMF $(W,H) \in \mathbb{R}^{m \times 2}_+ \times \mathbb{R}^{2 \times n}_+$.

```
1: % Compute an optimal rank-two approximation of $M$
2: $[U,S,V^T] = \text{svds}(M,2)$; % See the Matlab function svds
3: Let $X = SV$ $(= U^TUSV = U^TM)$;
4: % Extract two indices using SPA
5: $K = \text{SPA}(X,2)$; % See Algorithm 3
6: $W = \max\big(0, USV(:,K)\big)$;
7: $H = \text{argmin}_{Y \geq 0} ||M - WY||_F^2$; % See Algorithm 5
```

Let us analyze the computational cost of Algorithm 4. The computation of the rank-two SVD of M is $\mathcal{O}(mn)$ operations [22]. (Note that this operation scales well for sparse matrices as there exist SVD methods that can handle large sparse matrices, e.g., the `svds` function of Matlab.) For HSI's, m is much smaller than n (usually $m \sim 200$ while $n \sim 10^6$) hence it is faster to computing the SVD of M using the SVD of MM^T which requires $2mn + O(m^2)$ operations; see, e.g., [31]. Note however that this is numerically less stable as the condition number of the corresponding problem is squared. Extracting the two indices in step 5 with SPA requires $\mathcal{O}(n)$ operations [21], while computing the optimal H requires solving n linear systems in two variables for a total computational cost of $\mathcal{O}(mn)$

Algorithm 5 Nonnegative Least Squares with Two Variables [28]

Input: A matrix $A \in \mathbb{R}^{m \times 2}$ and a vector $b \in \mathbb{R}^m$.
Output: A solution $x \in \mathbb{R}^2_+$ to $\min_{x \geq 0} ||Ax - b||_2$.

```
% Compute the solution of the unconstrained least squares problem
$x = \text{argmin}_x ||Ax - b||_2$ (e.g., solve the normal equations $(A^TA)x = A^Tb$).
if $x \geq 0$ then
    return.
else
    % Compute the solutions for $x(1) = 0$ and $x(2) = 0$ (the two possible active sets)
    Let $y = \left(0, \max\left(0, \frac{A(:,1)^T b}{||A(:,1)||_2^2}\right)\right)$ and $z = \left(\max\left(0, \frac{A(:,2)^T b}{||A(:,2)||_2^2}\right), 0\right)$.
    if $||Ay - b||_2 < ||Az - b||_2$ then
        $x = y$.
    else
        $x = z$.
    end if
end if
```

operations [28]. In fact, the NNLS

$$\min_{X \in \mathbb{R}^{2 \times n}_+} ||M - WX||_F^2$$

where $W \in \mathbb{R}^{m \times 2}_+$ can be decoupled into n independent NNLS in two variables since

$$||M - WX||_F^2 = \sum_{i=1}^{n} ||M(:,i) - WX(:,i)||_2^2.$$

Algorithm 5 implements the algorithm in [28] to solve these subproblems.

Finally, Algorithm 4 requires $\mathcal{O}(mn)$ operations which implies that the global hierarchical clustering procedure (Algorithm 1) requires at most $\mathcal{O}(mnr)$ operations. Note that this is rather efficient and developing a significantly faster method would be difficult. In fact, it already requires $\mathcal{O}(mnr)$ operations to compute the product of $W \in \mathbb{R}^{m \times r}$ and $H \in \mathbb{R}^{r \times n}$, or to assign optimally n data points in dimension m to r cluster centroids using the Euclidean distance. Note however that in an ideal case, if the largest cluster is always divided into two clusters containing the same number of pixels (hence we would have a perfectly balanced tree), the number of operations reduces to $\mathcal{O}(mn \log(r))$. Hence, in practice, if the clusters are well balanced, the computational cost is rather in $\mathcal{O}(mn \log(r))$ operations.

2.3.2 Theoretical Motivations

An mentioned above, rank-two NMF can be solved exactly for rank-two input matrices. Let us show that Algorithm 4 does.

Theorem 1. *If M is a rank-two nonnegative matrix whose entries of each column sum to one, then Algorithm 4 computes an optimal rank-two NMF of M.*

Proof. Since M has rank-two and is nonnegative, there exists an exact rank-two NMF (F, G) of $M = FG = F(:,1)G(1,:) + X(:,2)Y(2,:)$ [34]. Moreover, since the entries of each column of M sum to one, we can assume without loss of generality that the entries of the each column of F and G sum to one as well. In fact, we can normalize the two columns of F so that their entries sum to one while scaling the rows of G accordingly:

$$M = \underbrace{\frac{F(:,1)}{||F(:,1)||_1}}_{F'(:,1)} \underbrace{||F(:,1)||_1 G(1,:)}_{G'(1,:)} + \underbrace{\frac{F(:,2)}{||F(:,2)||_1}}_{F'(:,2)} \underbrace{||F(:,2)||_1 G(2,:)}_{G'(2,:)} = F'G'.$$

Since the entries of each column of M and F' sum to one and $M = F'G'$, the entries of each column of G' have to sum to one as well. Hence, the columns of M belong to the line segment $[F'(:,1), F'(:,2)]$.

Let (U, S, V^T) be the rank-two SVD of M computed at step 2 of Algorithm 4, we have $SV = U^T M = (U^T F')G'$. Hence, the columns of SV belong to the line segment $[U^T F'(:,1)$,$U^T F'(:,2)]$ so that SPA applied on SV will identify two indices corresponding to two columns of M being the vertices of the line segment defined by its columns [21, Th.1]. Therefore, any column of M can be reconstructed with a convex combination of these two extracted columns and Algorithm 4 will generate an exact rank-two NMF of M. □

Corollary 1. *Let M be a noiseless HSI with two endmembers satisfying the linear mixing model and the sum-to-one constraint, then Algorithm 4 computes an optimal rank-two NMF of M.*

Proof. By definition, $M = WH$ where the columns of W are equal to the spectral signatures of the two endmembers and the columns of H are nonnegative and sum to one (see Introduction). The rest of the proof follows from the second part of the proof of Theorem 1. (Note that the pure-pixel assumption is not necessary.) □

In practice, the sum-to-one constraints assumption is sometimes relaxed to the following: the sum of the entries of each column of H is at most one. This has several advantages such as allowing the image to containing 'background' pixels with zero spectral signatures, or taking into account different intensities of light among the pixels in the image; see, e.g., [8]. In that case, Algorithm 4 works under the additional pure-pixel assumption:

Corollary 2. *Let M be a noiseless HSI with different illumination conditions, with two endmembers, and satisfying the linear mixing model and the pure-pixel assumption, then Algorithm 4 computes an optimal rank-two NMF of M.*

Proof. By assumption, $M = W[I_2, H']\Pi$ where H' is nonnegative and the entries of each column sum to at most one, and Π is a permutation. This implies that the columns of M are now in the triangle whose vertices are $W(:,1)$, $W(:,2)$ and the origin. Following the proof of Theorem 1, after the SVD, the columns of SV are in the triangle whose vertices are $U^T W(:,1)$, $U^T W(:,2)$ and the origin. Hence SPA will identify correctly the indices corresponding to $W(:,1)$ and $W(:,2)$ [21, Th.1] so that any column of M can be reconstructed using these two columns. □

At the first steps of the hierarchical procedure, rank-two NMF maps the data points into a two-dimensional subspace. However, the input matrix does not have rank-two if it contains more than two endmembers. In the following, we derive some simple sufficient conditions to support the fact that the rank-two SVD of a nonnegative matrix is nonnegative (or at least has most of its entries nonnegative). Let us refer to an optimal rank-two approximation of a matrix M as an optimal solution of

$$\min_{A \in \mathbb{R}^{m \times n}} ||M - A||_F^2 \quad \text{such that} \quad \text{rank}(A) = 2.$$

We will also refer to rank-two NMF as the following optimization problem

$$\min_{U \in \mathbb{R}^{m \times 2}, V \in \mathbb{R}^{2 \times n}} ||M - UV||_F^2 \quad \text{such that} \quad U \geq 0 \text{ and } V \geq 0.$$

Lemma 1. *Let $M \in \mathbb{R}^{m\times n}_{+}$, $A \in \mathbb{R}^{m\times n}$ be an optimal rank-two approximation of M, and $R = M - A$ be the residual error. If*

$$L = \min_{i,j}(M_{ij}) \geq \max_{i,j} R_{ij},$$

then every entry of A is nonnegative.

Proof. If $A_{kl} < 0$ for some (k,l), then $L \leq M_{kl} < M_{kl} - A_{kl} = R_{kl} \leq \max_{ij} R_{ij}$, a contradiction. □

Corollary 3. *Let $M \in \mathbb{R}^{m\times n}_{+}$ satisfy*

$$L = \min_{i,j}(M_{ij}) \geq \sigma_3(M).$$

Then any optimal rank-two approximation of M is nonnegative.

Proof. This follows from Lemma 1 since, for any optimal rank-two approximation A of M with $R = M - A$, we have $\max_{ij} R_{ij} \leq ||R||_2 = \sigma_3(M)$. □

Corollary 3 shows that a positive matrix close to having rank two and/or only containing relatively large entries is likely to have an optimal rank-two approximation which is nonnegative. Note that HSI's usually have mostly positive entries and, in fact, we have observed that the best rank-two approximation of real-world HSI's typically contains mostly nonnegative entries (e.g., for the Urban HSI more than 99.5%, for the San Diego HSI more than 99.9%, for the Cuprite HSI more than 99.98%, and for the Terrain HSI more than 99.8%; see Section 3.4 for a description of these data sets). It would be interesting to investigate further sufficient and necessary conditions for the optimal rank-two approximations of a nonnegative matrix to be nonnegative; this is a topic for further research. Note also Theorem 1 only holds for rank-two NMF and cannot be extended to more general cases with an arbitrary r. Consequently, we designed Algorithm 4 specifically for rank-two NMF. However, Algorithm 4 is important in the context of hierarchical clustering where rank-two NMF is the core computation. We will show in Section 3 that our overall method achieves high efficiency compared to other hyperspectral unmixing methods. Moreover, if we flatten the obtained tree structure and look at the clusters corresponding to the leaf nodes, we will see that H2NMF achieves much better cluster quality compared to the flat clustering methods including k-means and spherical k-means. Thus, though the theory in this paper is developed for rank-two NMF only, it has great significance in clustering hyperspectral images with more than two endmembers.

2.4 Geometric Interpretation of the Splitting Procedure

Given a HSI $M \in \mathbb{R}^{m\times n}$ containing r endmembers, and given that the pure-pixel assumption holds, we have that

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

where $W \in \mathbb{R}^{m\times r}$, $H' \geq 0$ and Π is a permutation matrix. This implies that the convex hull conv(M) of the columns of M coincides with the convex hull of the columns of W and has r vertices; see Introduction. A well-known fact in convex geometry is that the projection of any polytope P into an affine subspace generates another polytope, say P'. Moreover, each vertex of P' results from the projection of at least one vertex of P (although it is unlikely, it may happen that two vertices are projected onto the same vertex, given that the projection is parallel to the segment joining these two vertices). It is interesting to notice that this fact has been used previously in hyperspectral imaging: for example, the widely used VCA algorithm [31] uses three kinds of projections: First, it projects the data into a r-dimensional space using the SVD (in order to reduce the noise). Then, at each step,

- ◇ In order to identify a vertex (that is, an endmember), VCA projects conv(M) onto a one-dimensional subspace. More precisely, it randomly generates a vector $c \in \mathbb{R}^m$ and then selects the columns of M maximizing $c^T M(:,i)$.

- ◇ It projects all columns of M onto the orthogonal complement of the extracted vertex so that, if W is full rank (that is, if $\text{conv}(M)$ has r vertices and has dimension $r-1$), the projection of $\text{conv}(M)$ has $r-1$ vertices and has dimension $r-2$ (this step is the same as step 4 of SPA; see Algorithm 3).

In view of these observations, Algorithm 4 can be geometrically interpreted as follows:

- ◇ At the first step, the data points are projected into a two-dimensional subspace so that the maximum variance is preserved.
- ◇ At the second step, two vertices are extracted by SPA.
- ◇ At the third step, the data points are projected onto the two-dimensional convex cone generated by these two vertices.

2.5 Related Work

It has to be noted that the use of rank-two NMF as a subroutine to solve classification problems has already been studied before. In [29], a hierarchical NMF algorithm was proposed (namely, hierarchical NMF) based on rank-two NMF, and was used to identify tumor tissues in magnetic resonance spectroscopy images of the brain. The rank-two NMF subproblems were solved via standard iterative NMF techniques. In [25], a hierarchical approach was proposed for convex-hull NMF, that could discover clusters not corresponding to any vertex of the $\text{conv}(M)$ but lying inside $\text{conv}(M)$, and an algorithm based on FastMap [13] was used. In [28], hierarchical clustering based on rank-two NMF was used for document classification. The rank-two subproblems were solved using alternating nonnegative least squares [26, 27], that is, by optimizing alternatively W for H fixed, and H for W fixed (the subproblems being efficiently solved using Algorithm 5).

However, these methods do not take advantage of the nice properties of rank-two NMF, and the novelty of our technique is threefold:

- ◇ The way the next cluster to be split is chosen based on a greedy approach (so that the largest possible decrease in the error is obtained at each step); see Section 2.1.
- ◇ The way the clusters are split based on a trade off between having balanced clusters and stable clusters; see Section 2.2.
- ◇ The use of a rank-two NMF technique tailored for HSI's (using their convex geometry properties) to design a splitting procedure; see Section 2.3.

3 Numerical Experiments

In the first part, we compare different algorithms on synthetic data sets: this allows us to highlight their differences and also shows that our hierarchical clustering approach based on rank-two NMF is rather robust to noise and outliers. In the second part, we apply our technique to real-world hyperspectral data sets. This in turn shows the power of our rank-two NMF approach for clustering, but also as a robust hyperspectral unmixing algorithm for HSI. The Matlab code is available at `https://sites.google.com/site/nicolasgillis/`. All tests are preformed using Matlab on a laptop Intel CORE i5-3210M CPU @2.5GHz 2.5GHz 6Go RAM.

3.1 Tested Algorithms

We will compare the following algorithms:

1. **H2NMF**: hierarchical clustering based on rank-two NMF; see Algorithm 1 and Section 2.

2. **HKM**: hierarchical clustering based on k-means. This is exactly the same algorithm as H2NMF except that the clusters are split using k-means instead of the rank-two NMF based technique described in Section 2.2 (we used the `kmeans` function of Matlab).

3. **HSPKM**: hierarchical clustering based on spherical k-means [6]. This is exactly the same algorithm as H2NMF except that the clusters are split using spherical k-means (we used a Matlab code available online[2]).

4. **NMF**: we compute a rank-r NMF (U, V) of the HSI M using the accelerated HALS algorithm from [18]. Each pixel is assigned to the cluster corresponding to the largest entry of the columns of V.

5. **KM**: k-means algorithm with $k = r$.

6. **SPKM**: spherical k-means algorithm with $k = r$.

Moreover, the cluster centroids of HKM and HSPKM are initialized the same way as for H2NMF, that is, using steps 2-5 of Algorithm 4. NMF, KM and SPKM are initialized in a similar way: the rank-r SVD of M is first computed (which reduces the noise) and then SPA is applied on the resulting low-rank approximation of M (this is essentially equivalent to steps 2-5 of Algorithm 4 but replacing 2 by r). Note that we have tried using random initializations for HKM, HSPKM, NMF, KM and SPKM (which is the default in Matlab) but the corresponding clustering results were very poor (for example, NMF, KM and SPKM were in general not able to identifying the clusters perfectly in noiseless conditions). Recall that SPA is optimal for HSI's satisfying the pure-pixel assumption [21] hence it is a reasonable initialization.

3.2 Synthetic Data Sets

In this section, we compare the six algorithms described in the previous section on synthetic data sets, so that the ground truth labels are known. Given the parameters $\epsilon \geq 0$, $s \in \{0, 1\}$ and $b \in \{0, 1\}$, the synthetic HSI $M = [WH,\ Z] + N$ with $W \in \mathbb{R}^{m \times r}_{+}$, $H \in \mathbb{R}^{r \times (n-z)}_{+}$, $Z \in \mathbb{R}^{m \times z}_{+}$ and $N \in \mathbb{R}^{m \times n}$ is generated as follows:

- ◇ We use six endmembers, that is, $r = 6$.

- ◇ The spectral signatures of the six endmembers, that is, the columns of W, are taken as the spectral signatures of materials from the Cuprite HSI (see Section 3.5.2) and we have $W \in \mathbb{R}^{188 \times 6}_{+}$; see Figure 3. Note that W is rather poorly conditioned ($\kappa(W) = 91.5$) as the spectral signatures look very similar to one another.

- ◇ The pixels are assigned to the six clusters $\mathcal{K}_k$ $1 \leq k \leq r$ where each cluster contains a different number of pixels with $|\mathcal{K}_k| = 500 - (k - 1)50$, $1 \leq k \leq r$ (for a total of 2250 pixels).

- ◇ Once a pixel, say the ith, has been assigned to a cluster, say the kth, the corresponding column of H is generated as follows:
$$H(:, i) = 0.9\, e_k + 0.1\, x,$$

[2] `http://www.mathworks.com/matlabcentral/fileexchange/28902-spherical-k-means/content/spkmeans.m`

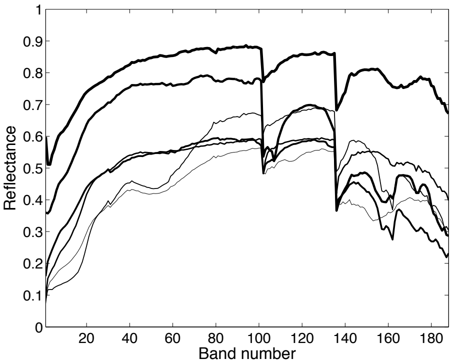

Figure 3: Spectral signatures from the Cuprite HSI used for the synthetic data sets.

where e_k is the kth column of the identity matrix, and $x \in \mathbb{R}^r_+$ is drawn from a Dirichlet distribution where all parameters are equal to 0.1. Note that the Dirichlet distribution generates a vector x whose entries sum to one (hence the entries of $H(:,i)$ also do), while the weight of the entries of x is concentrated only in a few components (hence each pixel usually contains only a few endmembers in large proportions). In particular, each pixel contains at least 90% of a single endmember.

◇ If $s = 1$, each column of H is multiplied by a constant drawn uniformly at random between 0.8 and 1. This allows us to take into account different illumination conditions in the HSI. Otherwise, if $s = 0$, then H is not modified.

◇ If $b = 1$, then ten outliers and forty background pixels with zero spectral signatures are added to M, that is, $Z = [z_1, z_2, \dots, z_{10}, 0_{m\times 40}]$ where $0_{p\times q}$ is the p-by-q all zero matrix. Each entry of an outlier $z_p \in \mathbb{R}^m_+$ $(1 \leq p \leq 10)$ is drawn uniformly at random in the interval $[0,1]$ (using the `rand` function of Matlab), and then the z_p's are scaled as follows:

$$z_p \leftarrow K_W \frac{z_p}{||z_p||_2} \quad 1 \leq p \leq 10,$$

where $K_W = \frac{1}{r}\sum_{k=1}^r ||W(:,k)||_2$ is the average of the norm of the columns of W. If $b = 0$, no outliers nor background pixels with zero spectral signatures are added to M, that is, Z is the empty matrix.

◇ The jth column of the noise matrix N is generated as follows: each entry is generated following the normal distribution $N(i,j) \sim \mathcal{N}(0,1)$ for all i (using the `randn` function of Matlab) and is then scaled as follows

$$N(:,j) \leftarrow \epsilon\, K_W\, u\, N(:,j),$$

where $\epsilon \geq 0$ is the parameter controlling the noise level, and u is drawn uniformly at random between 0 and 1 (hence the columns are perturbed with different noise levels which is more realistic).

Finally, the negative entries of $M = [WH,\, Z] + N$ are set to zero (note that this can only reduce the noise).

Once an algorithm was run on a data set and has generated r clusters $\mathcal{K}'_k$ $(1 \leq k \leq r)$, its performance is evaluated using the following criterion

$$\text{Accuracy} = \max_{P \in [1,2,\dots,r]} \frac{1}{n}\left(\sum_{k=1}^r |\mathcal{K}_k \cap \mathcal{K}'_{P(k)}|\right) \in [0,1],$$

where $[1, 2, \ldots, r]$ is the set of permutations of $\{1, 2, \ldots, r\}$, and $\mathcal{K}_k$ are the true clusters. Note that if a data point does not belong to any cluster (such as an outlier), it does not affect the accuracy. In other words, the accuracy can be equal to 1 even in the presence of outliers (as long as all other data points are properly clustered together).

3.3 Results

For each noise level ϵ and each value of s and b, we generate 25 synthetic HSI's as described in Section 3.2. Figure 4 reports the average accuracy; hence the higher the curve, the better.

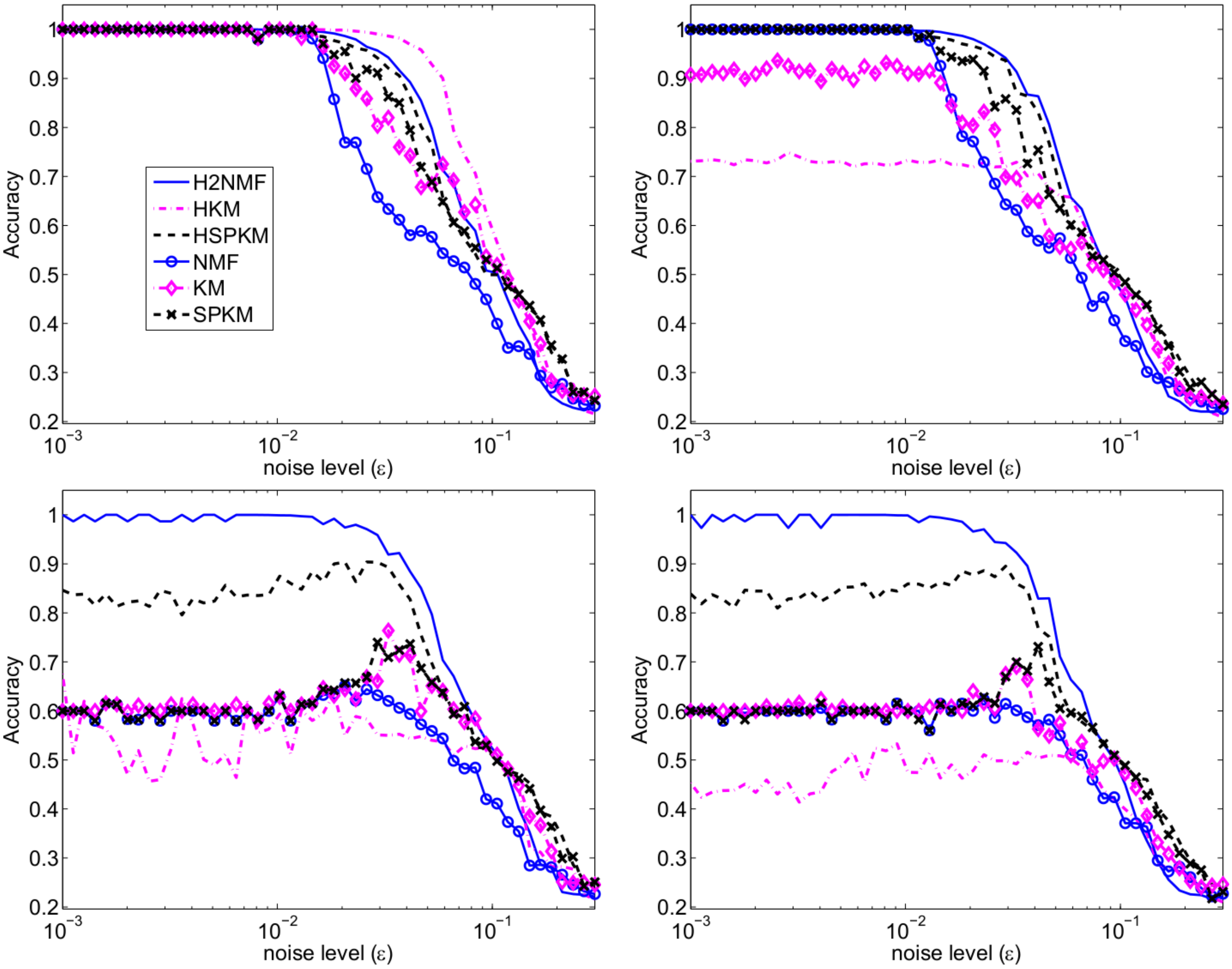

Figure 4: Performance of the different algorithms on synthetic data sets. From top to bottom, left to right: $(s, b) = (0,0)$, (1,0), (0,1), and (1,1).

We observe that:

- ◇ In almost all cases, the hierarchical clustering techniques consistently outperform the plain clustering approaches.
- ◇ Without scaling nor outliers (top left of Figure 4), HKM performs the best, while H2NMF is second best.
- ◇ With scaling but without outliers (top right of Figure 4), H2NMF performs the best, slightly better than SPKM while HKM performs rather poorly. This shows that HKM is sensitive to

scaling (that is, to different illumination conditions in the image), which will be confirmed on the real-world HSI's.

- With outliers but without scaling (bottom left of Figure 4), H2NMF outperforms all other algorithms. In particular, H2NMF has more than 95% average accuracy for all $\epsilon \leq 0.3$. HSPKM behaves better than other algorithms but is not able to perfectly cluster the pixels, even for very small noise levels.

- With scaling and outliers (bottom right of Figure 4), HKM performs even worse. H2NMF still outperforms all other algorithms, while HSPKM extracts relatively good clusters compared to the other approaches.

Table 1 gives the average computational time (in seconds) of all algorithms for clustering a single synthetic data set. We observe that SPKM is significantly faster than all other algorithms while HKM is slightly slower.

H2NMF	HKM	HSPKM	NMF	KM	SPKM
1.68	2.77	1.78	2.25	3.73	0.19

Table 1: Average running time in seconds for the different algorithms on the synthetic data sets.

3.4 Real-World Hyperspectral Images

In this section, we show that H2NMF is able to perform very good clustering of high resolution real-world HSI's. This section will focus on illustrating two important contributions: (i) H2NMF performs better than standard clustering techniques on real-world HSI, (ii) although H2NMF has been design to deal with HSI's with pixels dominated mostly by one endmember, it can provide meaningful and useful results in more difficult settings, and (iii) H2NMF can be used as an endmember extraction algorithm in the presence of pure pixels (we compare it to vertex component analysis (VCA) [31] and the successive projection algorithm (SPA) [3]). Note that because the ground truth of these HSI's is not known precisely, it is difficult to provide an objective quantitative measure for the cluster quality.

3.4.1 H2NMF as an Endmember Extraction Algorithm

Once a set of clusters $\mathcal{K}_k$ $(1 \leq k \leq r)$ has been identified by H2NMF (or any other clustering technique), each cluster of pixels should roughly correspond to a single material hence $M(:, \mathcal{K}_k)$ $(1 \leq k \leq r)$ should be close to rank-one matrices. Therefore, as explained in Section 2.1, it makes sense to approximate these matrices with their best-rank one approximation: For $1 \leq k \leq r$,

$$M(:, \mathcal{K}_k) \approx u_k v_k^T, \qquad \text{where } u_k \in \mathbb{R}^m, v_k \in \mathbb{R}^n.$$

Note that, by the Perron-Frobenius and Eckart-Young theorems, u_k and v_k $(1 \leq k \leq r)$ can be taken nonnegative since M is nonnegative. Finally, u_k should be close (up to a scaling factor) to the spectral signature of the endmember corresponding to the kth cluster. To extract a (good) pure pixel, a simple strategy is therefore to extract a pixel in each $\mathcal{K}_k$ whose spectral signature is the closest, with respect to some measure, to u_k. In this paper, we use the mean-removed spectral angle (MRSA) between u_k and the pixels present in the corresponding cluster (see, e.g., [2]). Given two spectral signatures, $x, y \in \mathbb{R}^m$, it is defined as

$$\phi(x, y) = \frac{1}{\pi} \arccos\left(\frac{(x - \bar{x})^T (y - \bar{y})}{||x - \bar{x}||_2 ||y - \bar{y}||_2} \right) \quad \in \quad [0, 1], \tag{3.1}$$

where, for a vector $z \in \mathbb{R}^m$, $\bar{z} = \left(\sum_{i=1}^{m} z_i\right) e$ and e is the vector of all ones.

As we will see, this approach is rather effective for high-resolution images, and much more robust to noise and outliers than VCA and SPA. This will be illustrated later in this section. (It is important to keep in mind that SPA and VCA require the pure-pixel assumption while H2NMF requires that most pixels are dominated mostly by one endmember.)

3.4.2 Urban HSI

The Urban HSI[3] from the HYper-spectral Digital Imagery Collection Experiment (HYDICE) contains 162 clean spectral bands, and the data cube has dimension 307 × 307 × 162. The Urban data set is a rather simple and well understood data set: it is mainly composed of 6 types of materials (road, dirt, trees, roof, grass and metal) as reported in [23]; see Figure 5 and Figure 8. Figure 6 displays the clusters obtained with H2NMF, HKM and HSPKM[4]. We observe that

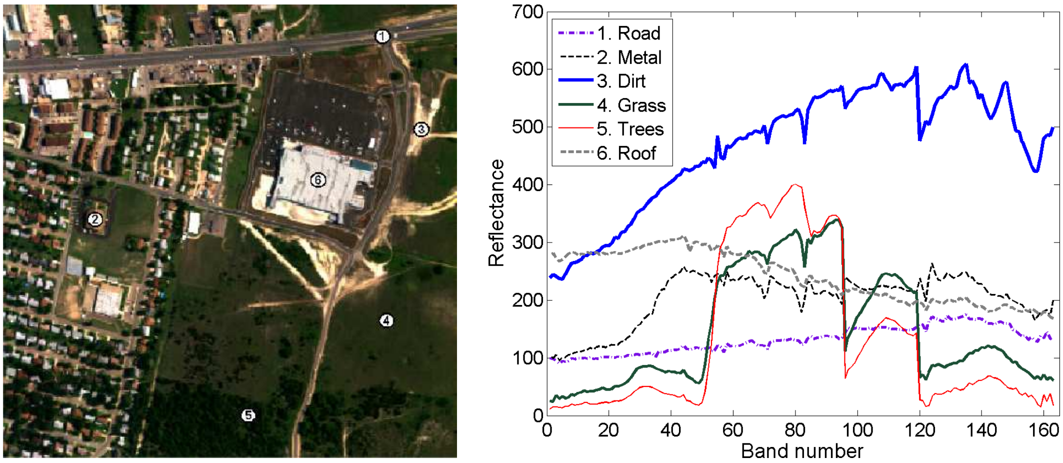

Figure 5: Urban HSI set taken from an aircraft (army geospatial center). The spectral signatures of the six endmembers on the right-hand side were obtained using the N-FINDR5 algorithm [36] plus manual adjustment [23].

- HKM performs very poorly. This is due to the illumination which is uneven among the pixels in the image (which is very damaging for HKM as shown on the synthetic data sets in Section 3.2).
- HSPKM properly extracts the trees and roof, but the grass is extracted as three separate clusters, while the road, metal and dirt form a unique cluster.
- H2NMF properly extracts the trees, roof and dirt, while the grass is extracted as two separate clusters, and the metal and road form a unique cluster.

The reason why H2NMF separates the grass before separating the road and metal is threefold: (i) the grass is the largest cluster and actually contains two subclasses with slightly different spectral signatures (as reported in [19]; see also Figure 9), (ii) the metal is a very small cluster, and (iii) the spectral signature of the road and metal are not so different (see Figure 5). Therefore, splitting the cluster containing the road and metal does not reduce the error as much as splitting the cluster containing the grass. It is important to note that our criterion used to choose the cluster to split at each step favors larger clusters as the singular values of a matrix tends to be larger when the matrix contains more

[3] Available at `http://www.agc.army.mil/`.

[4] The clustering obtained with KM and SPKM can be found in [32]; the clustering obtained with KM is rather poor while the one obtained with SPKM is similar to the one obtained with HSPKM.

columns (see Section 2.1). Although it works well in many situations (in particular, when clusters are relatively well balanced), other criterion might be preferable in some cases; this is a topic for further research.

Figure 7 displays the first levels of the cluster hierarchy generated by H2NMF. We see that if we were to split the cluster containing the road and metal, they would be properly separated. Therefore, we have also implemented an interactive version of H2NMF (denoted I-H2NMF) where, at each step, the cluster to split is visually selected[5]. Hence, selecting the right clusters to split (namely, splitting the road and metal, and not splitting the grass into two clusters) allows us to identifying all materials separately, see Figure 8. (Note that this is not possible with HKM and HSPKM.)

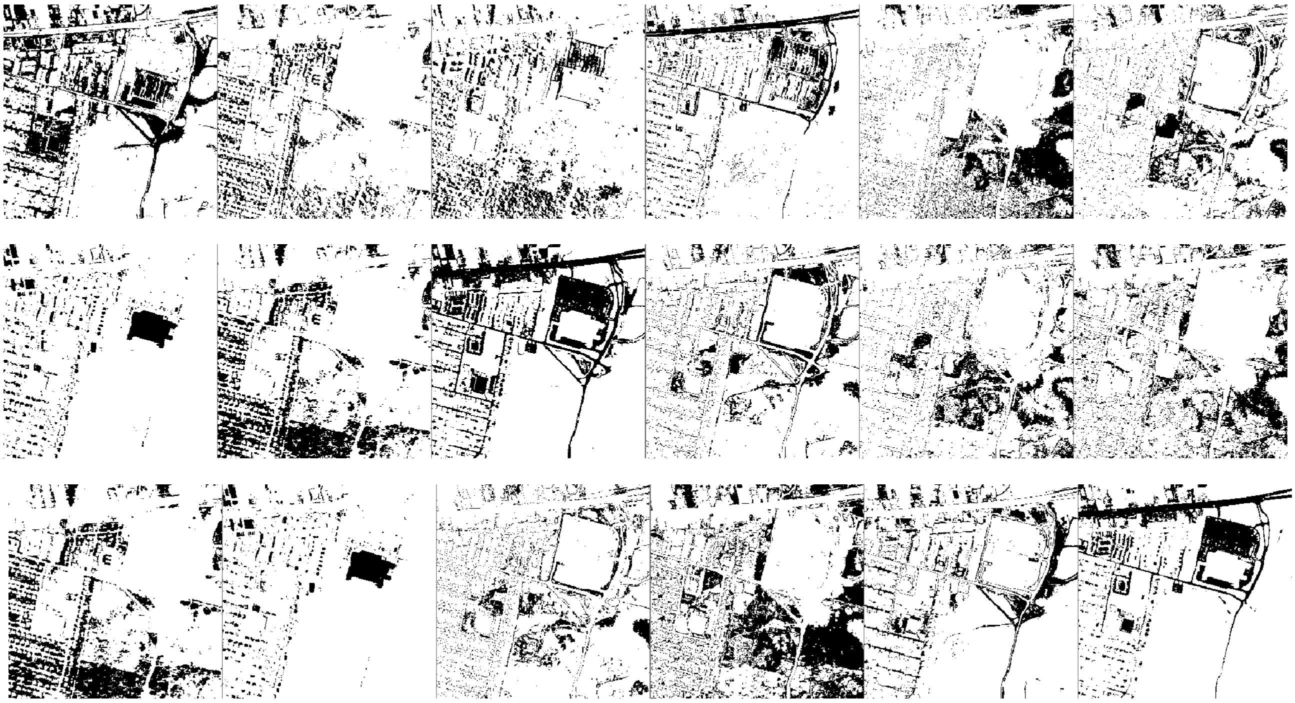

Figure 6: Clustering of the Urban HSI. From top to bottom: HKM, HSPKM, and H2NMF.

Using the strategy described in Section 3.4.1, we now compare the different algorithms when they are used for endmember extraction. Figure 9 displays the spectral signatures of the pixels extracted by the different algorithms. Letting w'_k $(1 \leq k \leq r)$ be the spectral signatures extracted by an algorithm, we match them with the 'true' spectral signatures w_k $(1 \leq k \leq r)$ from [23] so that $\sum_{k=1}^{r} \phi(w_k, w'_k)$ is minimized; see Equation (3.1). Table 2 reports the MRSA, along with the running time of all methods. Although the hierarchical clustering methods are computationally more expensive, they perform much better than both VCA and SPA.

3.4.3 San Diego Airport

The San Diego airport is a HYDICE HSI containing 158 clean bands, and 400 × 400 pixels for each spectral image (i.e., $M \in \mathbb{R}_{+}^{158 \times 160000}$), see Figure 10. There are mainly four types of materials: road surfaces, roof, trees and grass; see, e.g., [20]. There are three types of road surfaces including boarding and landing zones, parking lots and streets, and two types of roof tops[6]. In this section, we perform exactly the same experiment as in the previous section for the Urban HSI. The spectral signatures of

[5] This is also available at `https://sites.google.com/site/nicolasgillis/`. The user can interactively choose which cluster to split, when to stop the recursion and, if necessary, which clusters to fuse.

[6] Note that in [20], only one type of roof top is identified.

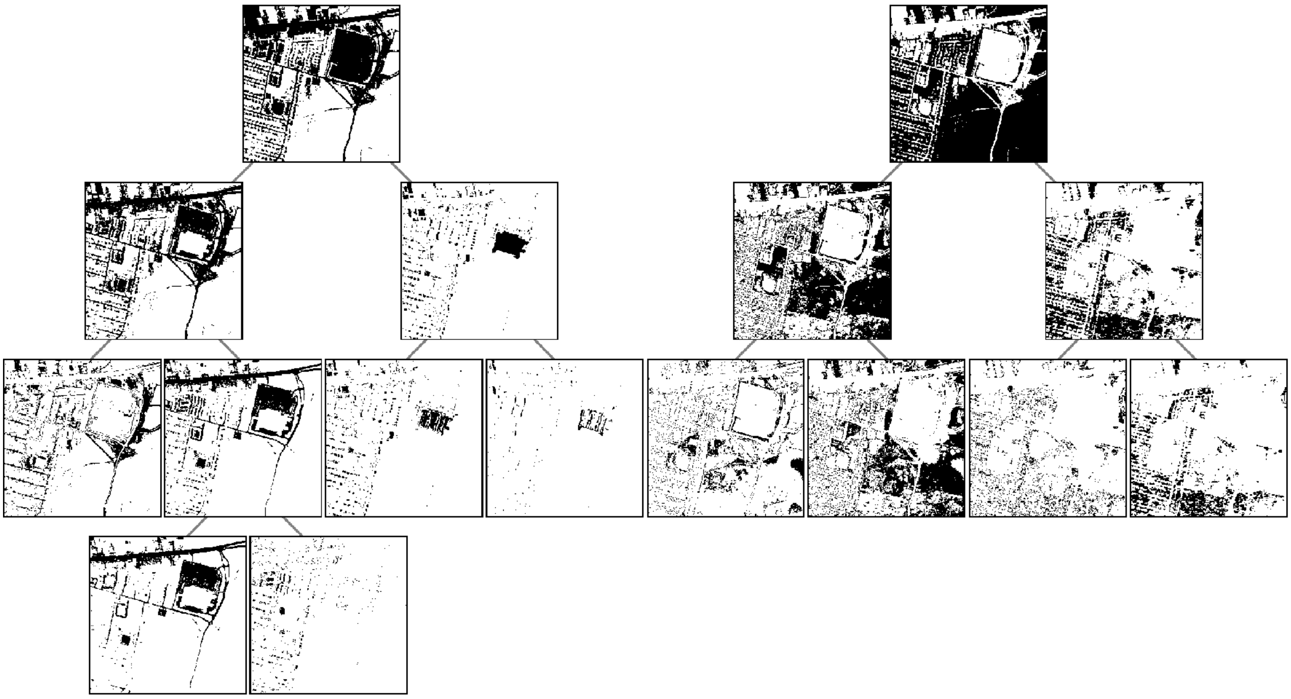

Figure 7: Hierarchical structure of H2NMF for the Urban HSI.

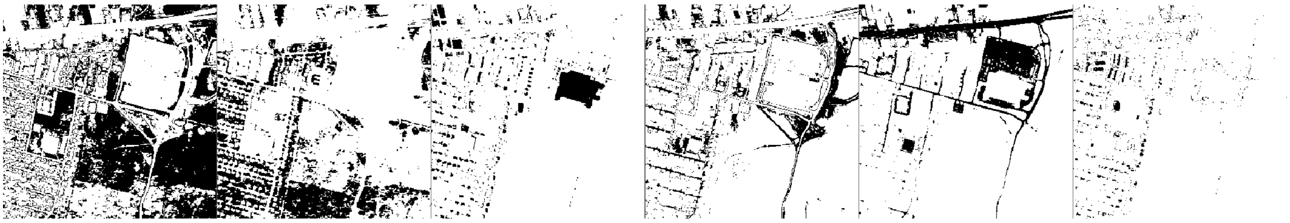

Figure 8: Interactive H2NMF (I-H2NMF) of the Urban HSI (see also Figure 7). From left to right: grass, trees, roof, dirt, road, and metal.

	VCA	SPA	HKM	HSPKM	H2NMF	I-H2NMF
Time (s.)	3.62	1.10	113.79	47.30	41.00	43.18
Road	13.09	11.54	14.97	11.54	7.62	**7.27**
Metal	51.53	62.31	30.72	31.42	28.61	**12.74**
Dirt	60.81	17.35	10.97	13.56	**5.08**	**5.08**
Grass	16.65	47.32	**2.46**	2.87	3.39	5.36
Trees	53.38	4.21	2.16	1.86	**1.63**	**1.63**
Roof	26.44	27.40	45.68	8.84	**7.30**	**7.30**
Average	36.98	28.36	17.82	11.68	8.94	**6.56**

Table 2: Running times and MRSA (in percent) for the Urban HSI.

the endmembers are shown on Figure 10 and have been extracted manually using the HYPERACTIVE toolkit [14].

Figure 11 displays the clusters obtained with H2NMF, HKM and HSPKM. We observe that

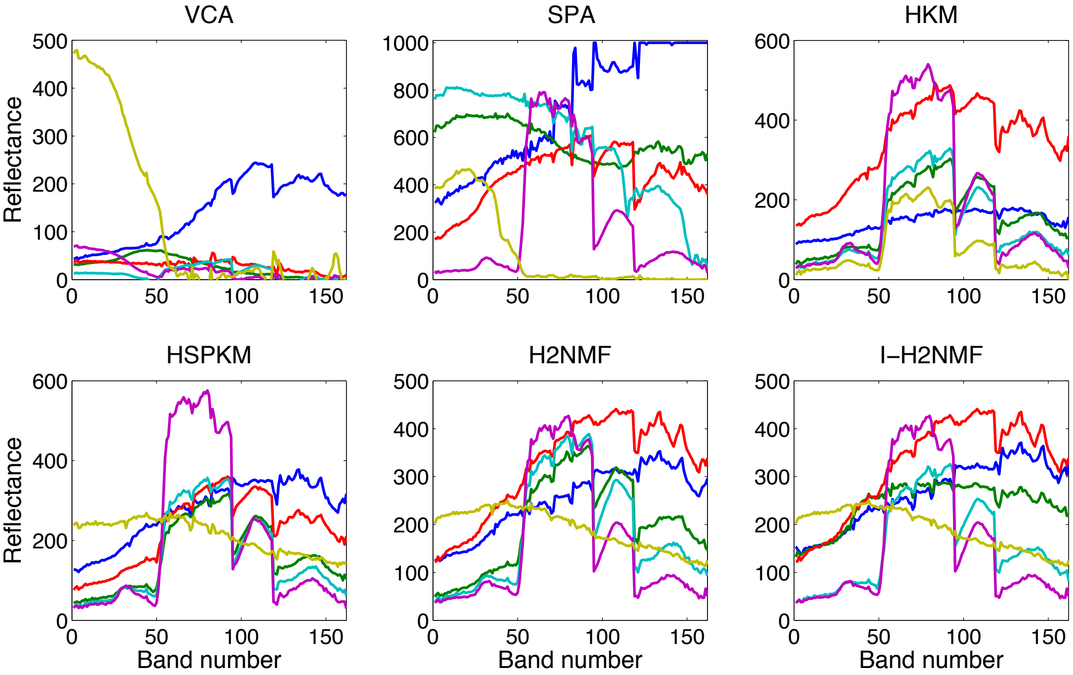

Figure 9: Spectral signatures extracted by the different algorithms for the Urban HSI.

- ◇ HKM performs rather poorly. It cannot identify any roof tops, and four clusters only contain the vegetation. The reason is that the spectral signatures of the pixels containing grass and trees differ by scaling factors.
- ◇ HSPKM properly extracts the grass, two of the three road surfaces and the roof tops (although roof 2 is mixed with some road surfaces).
- ◇ H2NMF extracts all materials very effectively, except for the trees.

The reason why H2NMF and HSPKM do not identify the trees properly is because the spectral signature of the trees and grass are almost the same: they only differ by a scaling factor (see Figure 10). Hence, for this data set, it is more effective to split the cluster containing the vegetation with k-means (which is the only one able to identify the trees, see the sixth abundance map of the first row of Figure 11). It would therefore be interesting to combine different methods for the splitting strategy; this is a topic for further research.

Figure 12 displays the spectral signatures of the pixels extracted by the different algorithms. Table 3 reports the running time of all methods, and the MRSA where the ground 'truth' are the manually selected spectral signatures. The hierarchical clustering methods perform much better than VCA and SPA. In fact, as it can be observed on Figure 12, VCA and SPA are very sensitive to outliers and both extract four of them (the San Diego airport HSI contains several outliers with very large spectral signatures).

Figure 13 displays the abundance maps[7] corresponding to the spectral signatures displayed in Figure 12. The abundance maps identified by H2NMF are the best: in fact, HSPKM does not identify road 1 which is mixed with roof 1. The reason is that the spectral signature extracted by HSPKM for road 1 is rather poor (see Table 3). Also, H2NMF actually identified better spectral signatures for

[7] We used the solution of $\min_{H \geq 0} ||M - WH||_F^2$ obtained with the NNLS solver from [27].

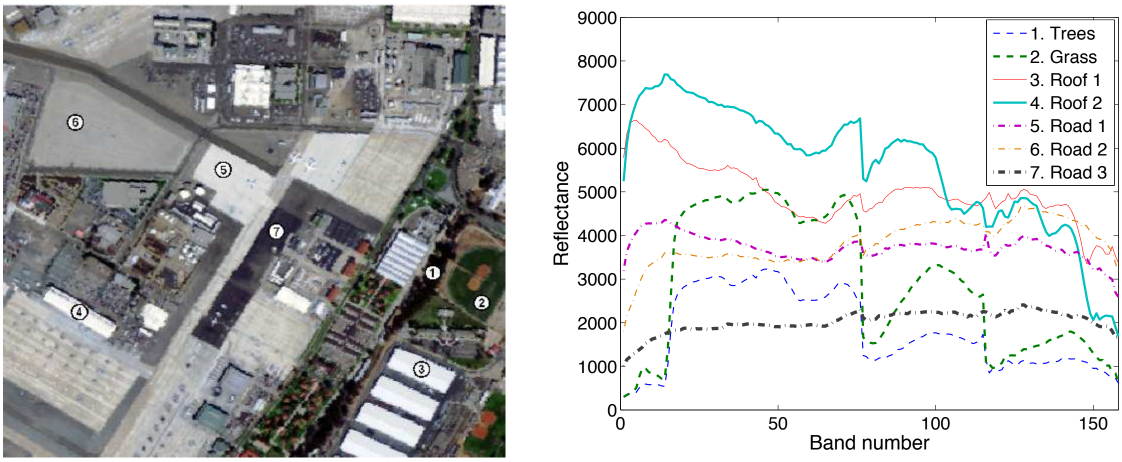

Figure 10: San Diego airport HSI (left) and manually selected spectral signatures (right).

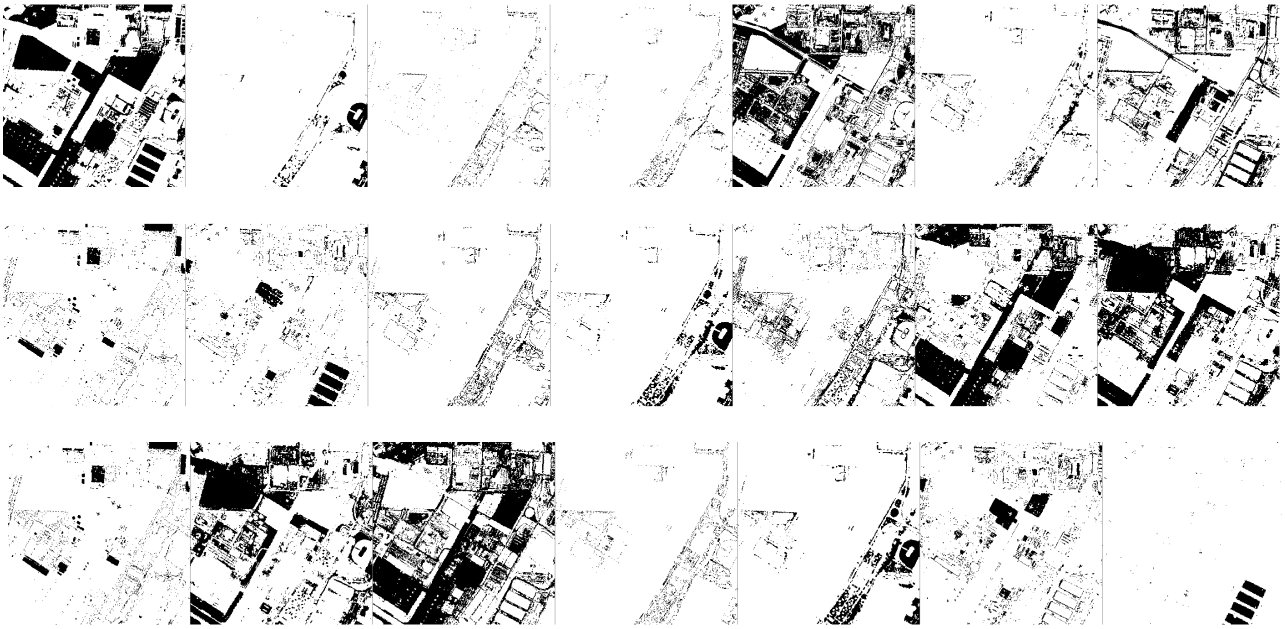

Figure 11: Clustering of the San Diego airport HSI. From top to bottom: HKM, HSPKM, and H2NMF.

roof 1 and road 1 than the manually selected ones for which the corresponding abundance maps on the first row of Figure 13 contains more mixture.

Figure 14 displays the first levels of the cluster hierarchy of H2NMF. It is interesting to notice that road 2 and 3 can be further split up into two meaningful subclasses. Moreover, another new material is identified (unknown to us prior to this study), it is some kind of roofing material/dirt (note that HKM and HSPKM are not able to identify this material). Figure 15 displays the clusters obtained using I-H2NMF, that is, manually splitting and fusing the clusters (more precisely, after having identified the new endmember by splitting road 2, we refuse the two clusters corresponding to road 2); see also Figure 14.

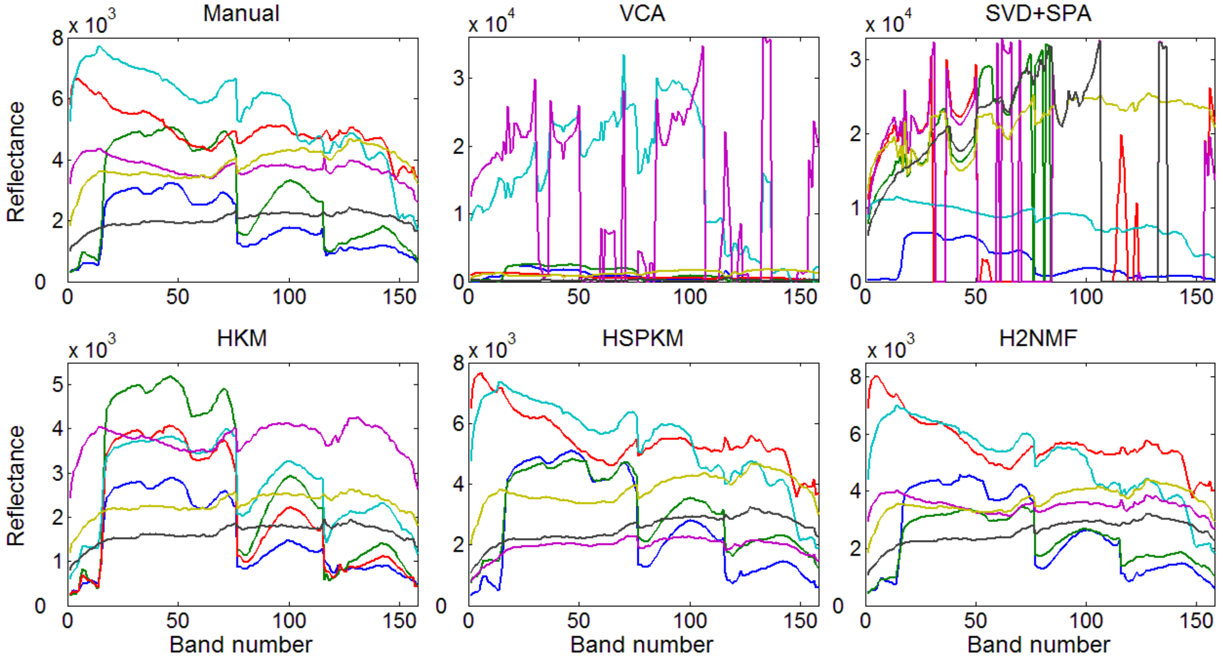

Figure 12: Spectral signatures extracted by the different algorithms for the San Diego airport HSI.

	VCA	SVD+SPA	HKM	HSPKM	H2NMF
Running time (s.)	5.13	1.95	145.75	98.09	68.91
Trees	21.52	9.20	**2.97**	3.57	3.42
Grass	8.18	32.36	**2.37**	2.79	7.09
Roof 1	16.06	38.30	47.43	**3.55**	4.29
Roof 2	27.36	3.01	35.21	**1.64**	2.45
Road 1	41.11	42.78	29.91	51.19	**9.58**
Road 2	19.28	21.13	13.84	5.24	**3.77**
Road 3	46.15	48.76	**5.32**	9.25	7.85
Average	25.67	27.93	19.58	11.03	**5.49**

Table 3: Running times and MRSA (in percent) for the San Diego airport HSI.

3.5 Additional experiments on real-world HSI's

In this section, our goal is not to compare the different clustering strategies (due to the space limitation) but rather show that H2NMF can give good results for other real-world and widely used data sets; in particular the Cuprite data set which is rather complicated with many endmembers and highly mixed pixels. We also take this opportunity to show that our Matlab code is rather easy to use and fast:

3.5.1 Terrain HSI

The Terrain hyperspectral image is available from `ttp://www.agc.army.mil/Missions/Hypercube.aspx`. It is constituted of 166 cleans bands, each having 500 × 307 pixels, and is composed of about 5 different materials: road, tree, bare soil, thin and tick grass; see, e.g., `http://www.way2c.com/rs2.php`. The Matlab code to run H2NMF is the following:

```
>> load Terrain; % Load HSI as matrix x
>> tic; [IDX, C] = hierclust2nmf(x,5); toc
```

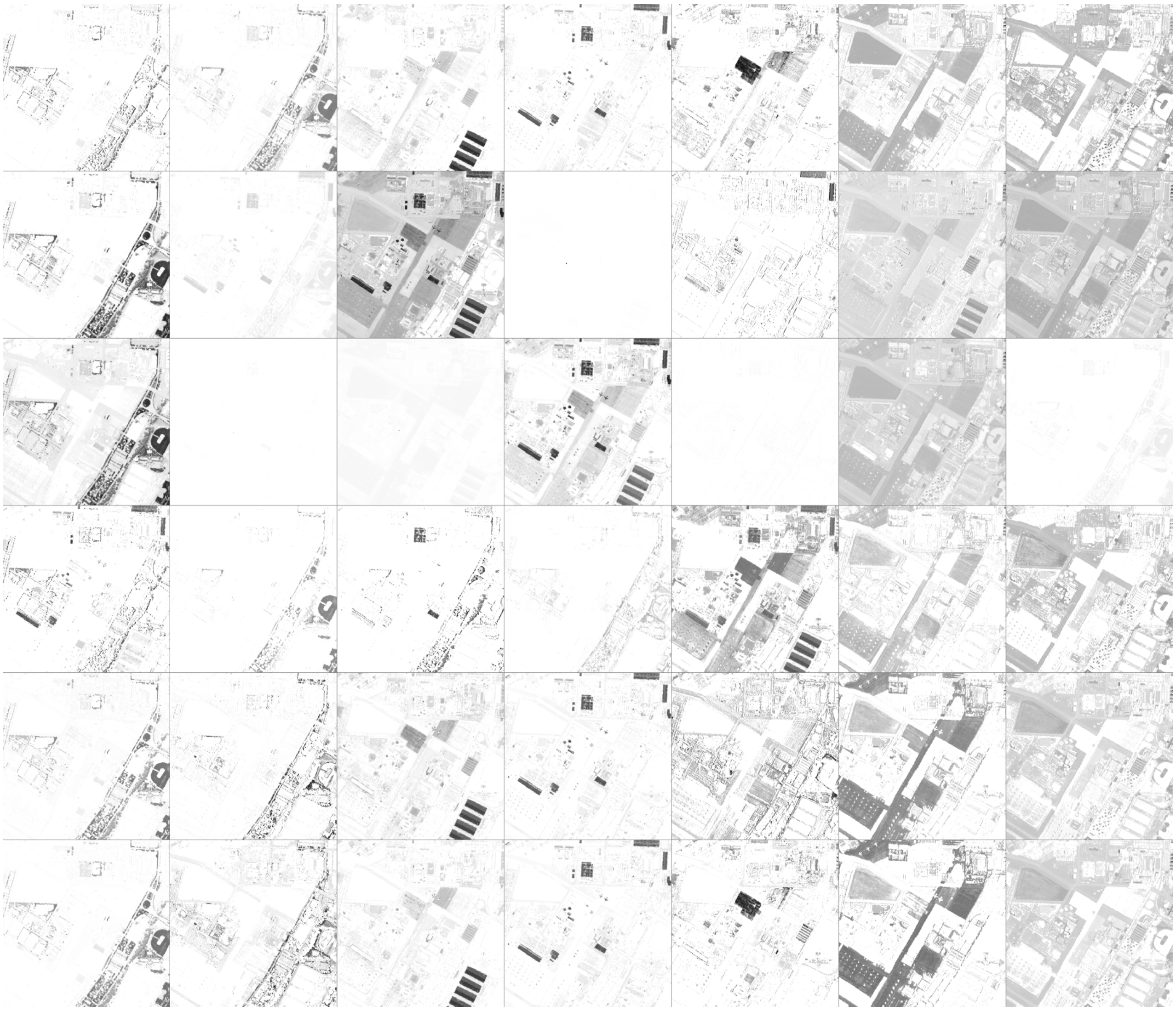

Figure 13: Comparison of the abundance maps obtained using the spectral signatures displayed in Figure 12. From left to right: trees/grass, roof 1, roof 2, road 1, road 2, road 3. From top to bottom: Manual, VCA, SVD+SPA, HKM, HSPKM, H2NMF.

```
Hierarchical clustering started...
1...2...3...4...Done.
Elapsed time is 20.261847 seconds.
>> affclust(IDX,500,307,5); % Display the clusters; see Figure 16
>> figure; plot(C);  % Display endmembers; see Figure 17
>> H = nnlsm_blockpivot(C,x); % Compute abundance maps
>> affichage(H',5,500,307); % Display abundance maps; see Figure 18
```

H2NMF is able to identify the five clusters extremely well, while HKM and HSPKM are not able to separate bare soil, thick and thin grass properly.

3.5.2 Cuprite HSI

Cuprite is a mining area in southern Nevada with mostly mineral and very little vegetation, located approximately 200km northwest of Las Vegas, see, e.g., [31, 2] for more information and `http://`

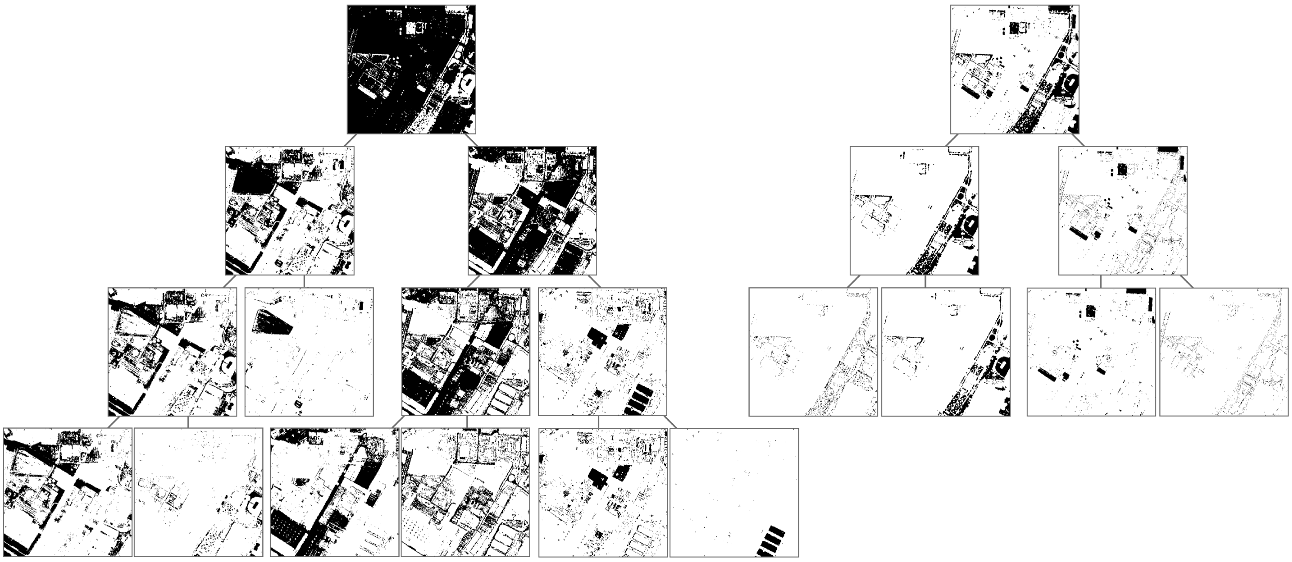

Figure 14: Hierarchical structure of H2NMF for the San Diego airport HSI.

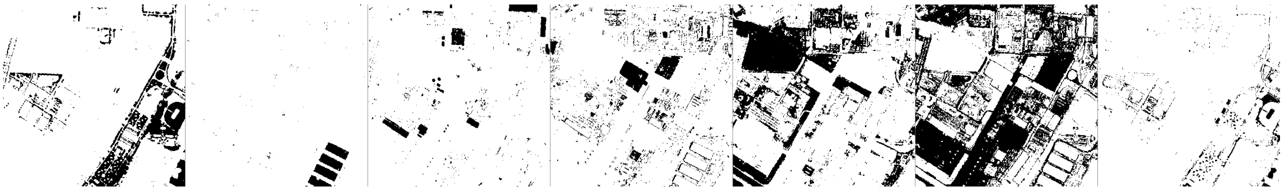

Figure 15: Clustering of San Diego airport HSI with I-H2NMF. From left to right, top to bottom: vegetation (grass and trees), roof 1, roof 2, road 1, road 2, road 3, dirt/roofing.

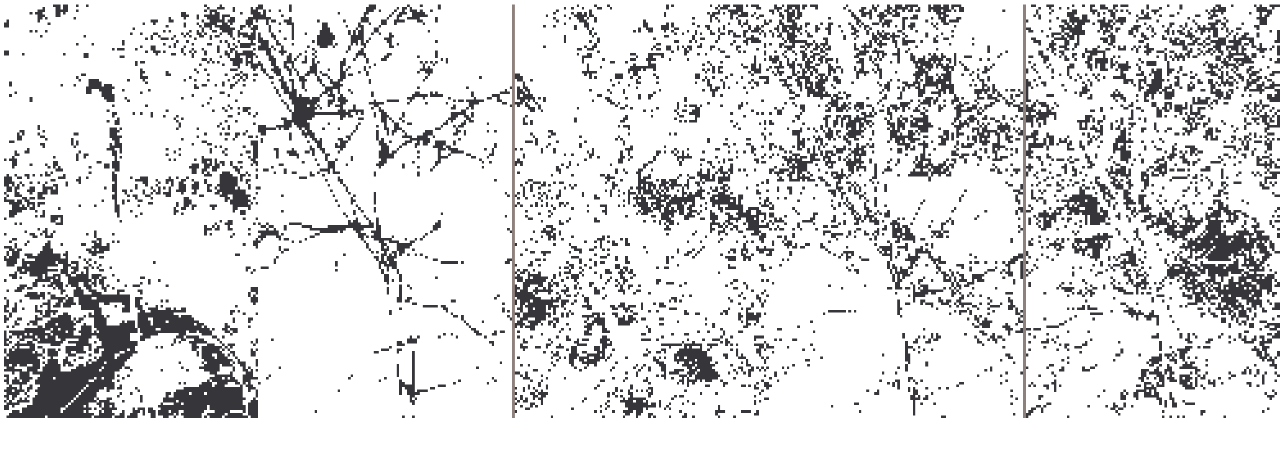

Figure 16: Five clusters obtained automatically with H2NMF on the Terrain HSI. From left to right: tree, road, thick grass, bare soil and thin grass.

`speclab.cr.usgs.gov/PAPERS.imspec.evol/aviris.evolution.html`. It consists of 188 images, each having 250 × 191 pixels, and is composed of about 20 different minerals. The Cuprite HSI is rather noisy and many pixels are mixture of several endmembers. Hence this experiment illustrates the usefulness of H2NMF to analyze more difficult data sets, where the assumption that most pixels

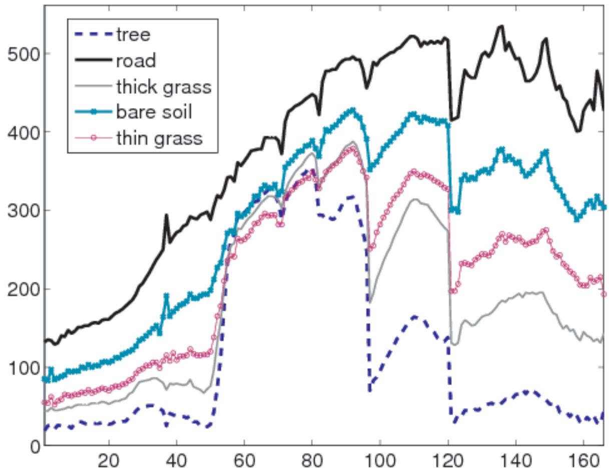

Figure 17: Five endmembers obtained with H2NMF on the Terrain HSI.

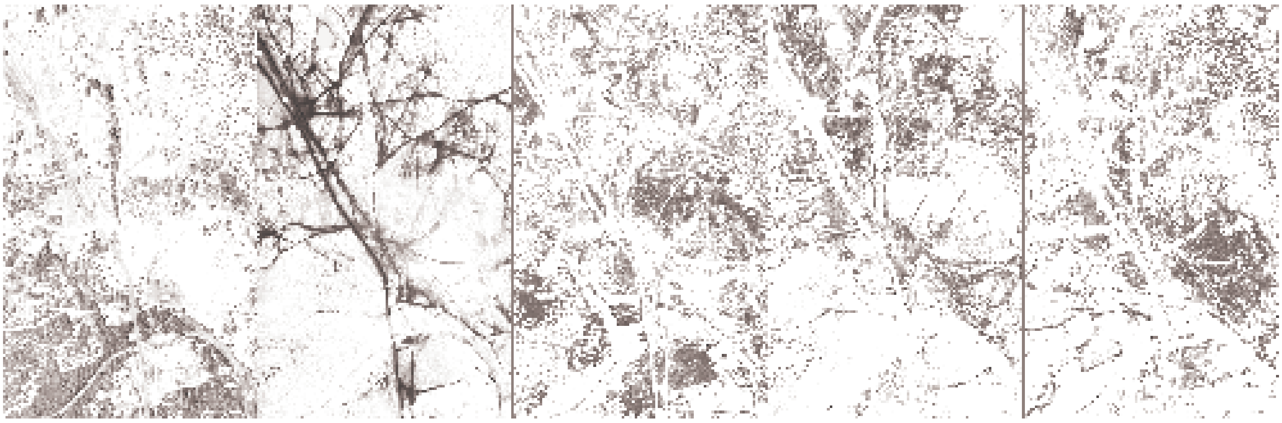

Figure 18: Five abundance maps corresponding to the endmembers extracted with H2NMF.

are dominated mostly by one endmember is only roughly satisfied; see Figure 19. We run H2NMF with $r = 15$:

```
>> load cuprite_ref; %From www.lx.it.pt/~bioucas.
>> tic; [IDX, C] = hierclust2nmf(x,15); toc
Hierarchical clustering started...
1...2...3...4...5...6...7...8...9...10...
11...12...13...14...Done.
Elapsed time is 11.632038 seconds.
>> affclust(IDX,250,191,5); %See Figure 19 displaying the 15 clusters.
```

4 Conclusion and Further Work

In this paper, we have introduced a way to perform hierarchical clustering of high-resolution HSI's using the geometry of such images and the properties of rank-two NMF; see Algorithm 1 (referred to as H2NMF). We showed that the proposed method outperforms k-means, spherical k-means and standard NMF on several synthetic and real-world data sets, being more robust to noise and outliers, while being computationally very efficient, requiring $\mathcal{O}(mnr)$ operations (m is the number of spectral

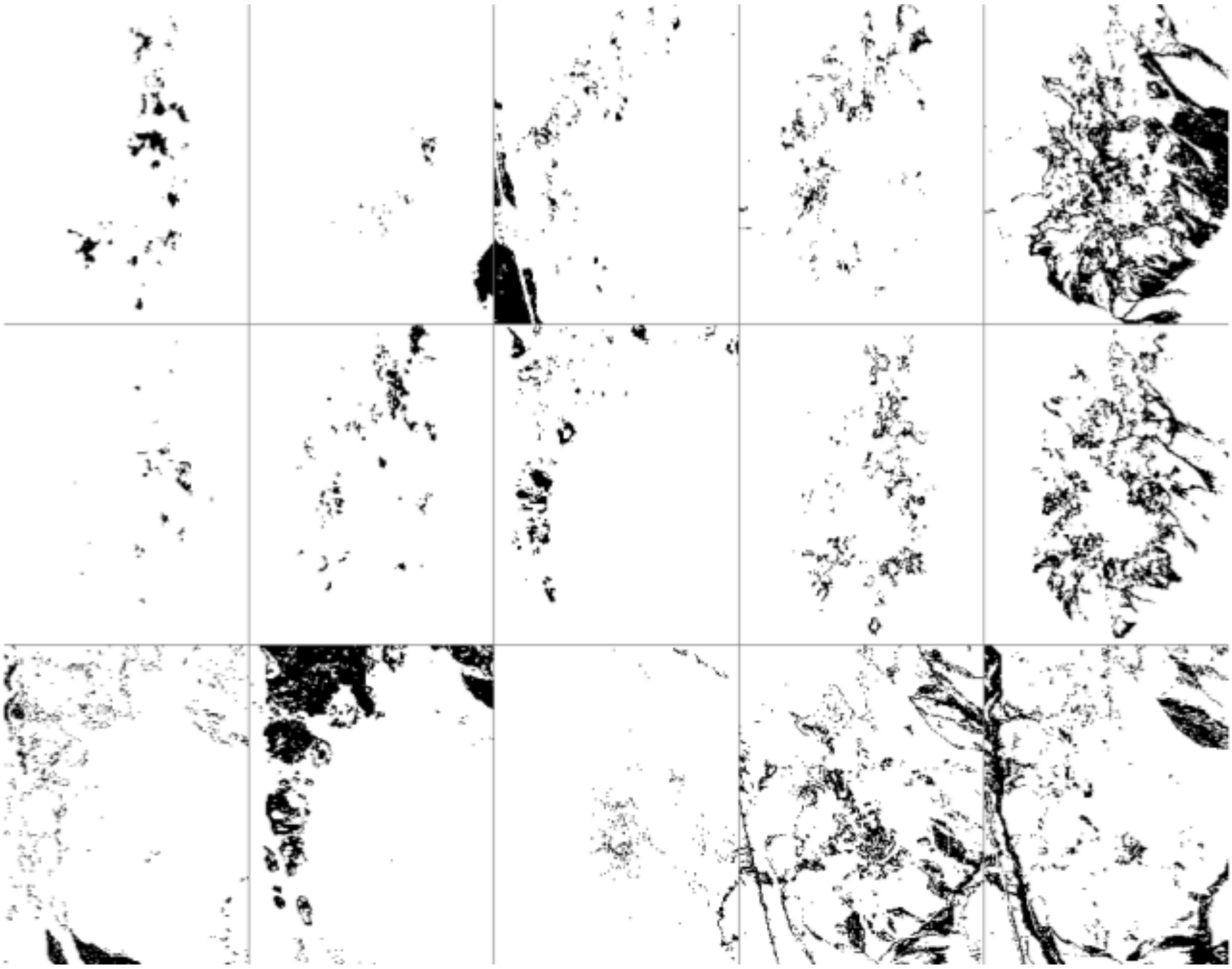

Figure 19: Fifteen clusters obtained automatically with H2NMF on the Cuprite HSI. Some materials can be distinguished, e.g., (1) Alunite, (2) Montmorillonite, (3) Goethite, (5) Hematite, (8)-(12) Desert Varnish, (11) Iron oxydes, and (15) Kaolinite (counting from left to right, top to botttom).

bands, n the number of pixels and r the number of clusters). Although high resolution HSI's usually have low noise levels, one of the reason H2NMF performs well is that it can handle better background pixels and outliers. There might also be some materials present in very small proportion that are usually modeled as noise [8] (hence robustness to noise is a desirable property even for high resolution HSI's). Moreover, we also showed how to use H2NMF to identify pure pixels which outperforms standard endmember extraction algorithms such as VCA and SPA.

It would be particularly interesting to use other priors of HSI's to perform the clustering. In particular, using the spatial information (that is, the fact that neighboring pixels are more likely to contain the same materials) could certainly improve the clustering accuracy. Also, the same technique could be applied to other kinds of data (e.g., in medical imaging, or document classification).

Acknowledgments

The authors would like to thank the editor and the reviewers for their insightful comments which helped improve the paper.

References

[1] Ambikapathi, A., Chan, T.H., Ma, W.K., Chi, C.Y.: A robust alternating volume maximization algorithm for endmember extraction in hyperspectral images. In: WHISPERS, Reykjavik, Iceland (2010)

[2] Ambikapathi, A., Chan, T.H., Ma, W.K., Chi, C.Y.: Chance-constrained robust minimum-volume enclosing simplex algorithm for hyperspectral unmixing. IEEE Trans. Geosci. Remote Sens. **49**(11), 4194–4209 (2011)

[3] 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)

[4] Arbelaez, P., Maire, M., Fowlkes, C., Malik, J.: Contour detection and hierarchical image segmentation. IEEE Trans. on Pattern Analysis and Machine Intelligence **33**(5), 898–916 (2011)

[5] Arora, S., Ge, R., Kannan, R., Moitra, A.: Computing a nonnegative matrix factorization – provably. In: STOC ’12, pp. 145–162 (2012)

[6] Banerjee, A., Dhillon, I., Ghosh, J., Sra, S.: Generative model-based clustering of directional data. In: Proceedings of the ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD-03), pp. 19–28. ACM Press (2003)

[7] Bioucas-Dias, J., Nascimento, J.: Estimation of signal subspace on hyperspectral data. In: Remote Sensing, p. 59820L. International Society for Optics and Photonics (2005)

[8] Bioucas-Dias, J., 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)

[9] Cai, D., He, X., Li, Z., Ma, W.Y., Wen, J.R.: Hierarchical clustering of www image search results using visual, textual and link information. In: Proc. of the 12th annual ACM Int. Conf. on Multimedia, pp. 952–959 (2004)

[10] Chan, T.H., Ma, W.K., Ambikapathi, A., Chi, C.Y.: IEEE Trans. Geosci. Remote Sens., title=A Simplex Volume Maximization Framework for Hyperspectral Endmember Extraction, year=2011, volume=49, number=11, pages=4177-4193,

[11] Cohen, J., Rothblum, U.: Nonnegative ranks, Decompositions and Factorization of Nonnegative Matrices. Linear Algebra and its Applications **190**, 149–168 (1993)

[12] Eisen, M., Spellman, P., Brown, P., Botstein, D.: Cluster analysis and display of genome-wide expression patterns. Proc. Natl. Acad. Sci. USA **95**(25), 14,863–14,868 (1998)

[13] Faloutsos, C., Lin, K.I.: Fastmap: a fast algorithm for indexing, data-mining and visualization of traditional and multimedia datasets. In: SIGMOD ’95: Proc. of the 1995 ACM SIGMOD Int. Conf. on Mgmt. of Data, pp. 163–174 (1995)

[14] Fong, M., Hu, Z.: Hyperactive: A matlab tool for visualization of hyperspectral images (2007). `http://www.math.ucla.edu/~wittman/lambda/software.html`

[15] Gillis, N.: Nonnegative matrix factorization: Complexity, algorithms and applications. Ph.D. thesis, Université catholique de Louvain (2011)

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

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

[18] Gillis, N., Glineur, F.: Accelerated multiplicative updates and hierarchical ALS algorithms for nonnegative matrix factorization. Neural Computation **24**(4), 1085–1105 (2012)

[19] Gillis, N., Plemmons, R.: Dimensionality reduction, classification, and spectral mixture analysis using nonnegative underapproximation. Optical Engineering **50, 027001** (2011)

[20] Gillis, N., Plemmons, R.: Sparse nonnegative matrix underapproximation and its application to hyperspectral image analysis. Linear Algebra and its Applications **438**(10), 3991–4007 (2013)

[21] Gillis, N., Vavasis, S.: Fast and robust recursive algorithms for separable nonnegative matrix factorization. IEEE Trans. Pattern Anal. Mach. Intell. **36**(4), 698–714 (2014)

[22] Golub, G., Van Loan, C.: Matrix Computation, 3rd Edition. The Johns Hopkins University Press Baltimore (1996)

[23] Guo, Z., Wittman, T., Osher, S.: L1 unmixing and its application to hyperspectral image enhancement. In: Proc. SPIE Conference on Algorithms and Technologies for Multispectral, Hyperspectral, and Ultraspectral Imagery XV (2009)

[24] Jia, S., Qian, Y.: Constrained nonnegative matrix factorization for hyperspectral unmixing. IEEE Trans. Geosci. Remote Sens. **47(1)**, 161–173 (2009)

[25] Kersting, K., Wahabzada, M., Thurau, C., Bauckhage, C.: Hierarchical convex nmf for clustering massive data. In: ACML '10: Proc. of 2nd Asian Conf. on Machine Learning, pp. 253–268 (2010)

[26] Kim, H., Park, H.: Sparse non-negative matrix factorizations via alternating non-negativity-constrained least squares for microarray data analysis. Bioinformatics **23**(12), 1495–1502 (2007)

[27] Kim, J., Park, H.: Fast nonnegative matrix factorization: An active-set-like method and comparisons. SIAM J. on Scientific Computing **33**(6), 3261–3281 (2011)

[28] Kuang, D., Park, H.: Fast rank-2 nonnegative matrix factorization for hierarchical document clustering. In: 19th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD '13), pp. 739–747 (2013)

[29] Li, Y., Sima, D., Van Cauter, S., Croitor Sava, S., Himmelreich, U., Pi, Y., Van Huffel, S.: Hierarchical non-negative matrix factorization (hnmf): a tissue pattern differentiation method for glioblastoma multiforme diagnosis using mrsi. NMR in Biomedicine **26**(3), 307–319 (2012)

[30] Ma, W.K., Bioucas-Dias, J., Chan, T.H., Gillis, N., Gader, P., Plaza, A., Ambikapathi, A., Chi, C.Y.: Signal processing perspective on hyperspectral unmixing. IEEE Signal Processing Magazine **31**(1), 67–81 (2014)

[31] Nascimento, J., Bioucas-Dias, J.: Vertex component analysis: a fast algorithm to unmix hyperspectral data. IEEE Trans. Geosci. Remote Sens. **43**(4), 898–910 (2005)

[32] Pompili, F., Gillis, N., Absil, P.A., Glineur, F.: Two algorithms for orthogonal nonnegative matrix factorization with application to clustering. Neurocomputing **141**, 15–25 (2014)

[33] Ren, H., Chang, C.I.: Automatic spectral target recognition in hyperspectral imagery. IEEE Trans. on Aerospace and Electronic Systems **39**(4), 1232–1249 (2003)

[34] Thomas, L.: Rank factorization of nonnegative matrices. SIAM Review **16**(3), 393–394 (1974)

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

[36] Winter, M.: N-findr: an algorithm for fast autonomous spectral end-member determination in hyperspectral data. In: Proc. SPIE Conference on Imaging Spectrometry V (1999)

[37] Zhao, Y., Karypis, G., Fayyad, U.: Hierarchical clustering algorithms for document datasets. Data Min. Knowl. Discov. **10**(2), 141–168 (2005)

[38] Zymnis, A., Kim, S.J., Skaf, J., Parente, M., Boyd, S.: Hyperspectral image unmixing via alternating projected subgradients. In: Signals, Systems and Computers, pp. 1164–1168 (2007)