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

Nicolas Gillis, Da Kuang, and Haesun Park

***Abstract*—In this paper, we design a fast hierarchical clustering algorithm for high-resolution hyperspectral images (HSI). At the core of the algorithm, a new rank-two nonnegative matrix factorization (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, performs the following: 1) selects a cluster in such a way that the error at the next step is minimized and 2) 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 HSIs and is shown to outperform standard clustering techniques such as k-means, spherical k-means, and standard NMF.**

***Index Terms*—Blind unmixing, endmember extraction algorithm, hierarchical clustering, high-resolution hyperspectral images (HSIs), nonnegative matrix factorization (NMF).**

I. 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 650 nm, green at 550 nm, and blue at 450 nm). An important problem in hyperspectral imaging is blind hyperspectral unmixing (blind HU): Given an HSI, the goal is to recover the constitutive materials present in the image (the *endmembers*) and the corresponding abundance maps (i.e., 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, for example, [1] 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 (i.e., 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 aluminum plus 0.6 times the spectral signature of copper. This is a rather natural model: We assume that 40% of the light is reflected by aluminum while 60% is by copper, while nonlinear effects are neglected (such as the light interacting with multiple materials before reflecting off or atmospheric distortions).

Assuming that the image contains r endmembers and denoting $W(:, k) \in \mathbb{R}^m (1 \leq k \leq r)$ as 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 an 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 [2] and highly ill-posed [3]. 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 previous discussion), sparsity of the abundance matrix H (most pixels contain only a few endmembers), piecewise smoothness of the spectral signatures $W(:, k)$ [4], and spatial information [5] (i.e., 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 [1] for more details about blind HU.

In this paper, we make an additional assumption, i.e., that *most pixels are dominated mostly by one endmember*, and our goal is to cluster the pixels accordingly. In fact, clustering the

Manuscript received September 2, 2013; revised February 5, 2014 and July 2, 2014; accepted August 10, 2014.

N. Gillis is with the Department of Mathematics and Operational Research, Faculté Polytechnique, Université de Mons, 7000 Mons, Belgium (e-mail: nicolas.gillis@umons.ac.be).

D. Kuang and H. Park are with the School of Computational Science and Engineering, Georgia Institute of Technology, Atlanta, GA 30332-0765 USA (e-mail: da.kuang@cc.gatech.edu; hpark@cc.gatech.edu).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TGRS.2014.2352857

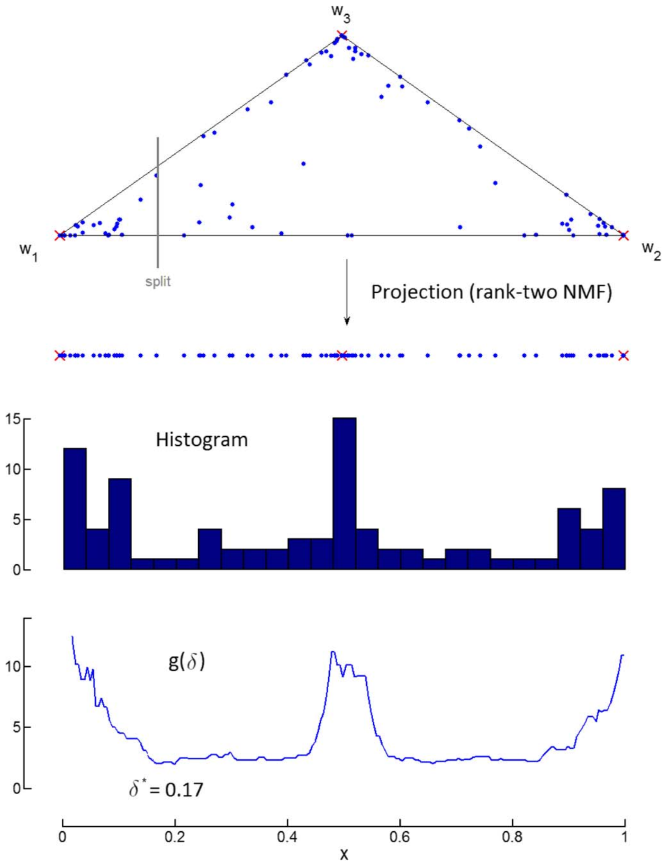


Fig. 1. Illustration of the splitting technique based on rank-two NMF.

pixels of an 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, i.e., 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 [6] and the references therein) which makes the corresponding NMF problem tractable, even in the presence of noise [7]. 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 an HSI, we have, as before, the following.

1) The number r is the number of endmembers present in the HSI.
2) Each column of W is the spectral signature of an endmember.
3) 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, i.e., $\text{conv}(M) \subseteq \text{conv}(W)$. The pure-pixel assumption requires that $\text{conv}(M) = \text{conv}(W)$, i.e., that the vertices of the convex hull of the columns of M are the columns of W (see the top of Fig. 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, for example, [8] and the references therein), they are typically rather sensitive to noise and outliers.

Motivated by the fact that, in high-resolution HSIs, most pixels are dominated mostly 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 [9]. 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 HSIs can be efficiently solved.

This paper is organized as follows. In Section II, we describe our hierarchical clustering approach (see Algorithm 1 referred to as H2NMF). At each step, a cluster is selected (Section II-A) and then split into two disjoint clusters (Section II-B). The splitting procedure has a rank-two NMF algorithm at its core which is described in Section II-C, where we also provide some sufficient conditions under which the proposed algorithm recovers an optimal solution. In Section II-D, we analyze the geometric properties of the hierarchical clustering. In Section III, 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 HSIs, being more robust to noise, outliers, and absence of pure pixels. We also show that it can be used as an endmember extraction algorithm and outperforms vertex component analysis (VCA) [10] and the successive projection algorithm (SPA) [11], two standard and widely used techniques.

II. Hierarchical Clustering for HSIs 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 that 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, for example, [12]). 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, for example, Figs. 6 and 9). 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 an 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 [9], [13], [14], as well as exploratory genomic study for grouping genes participating in the same pathway [15]. Another example is to segment an image into a hierarchy of regions according to different cues in computer vision such as contours and textures [16]. In contrast to image segmentation problems, our focus is to obtain a hierarchy of materials from HSIs 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 HSIs are described in the following sections.

A. 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 $M(:,j) = W(:,k)$, 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, i.e., 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:

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

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

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

(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

$$\sigma_1^2\left(M\left(:,\mathcal{K}_k^1\right)\right) + \sigma_1^2\left(M\left(:,\mathcal{K}_k^2\right) - \sigma_1^2\left(M(:,\mathcal{K}_k)\right)\right).$$

Therefore, we propose to split the cluster k for which the aforementioned value is maximized: This leads to the largest possible decrease in the total error E at each step.

B. 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 twofold.

1) NMF corresponds to the LMM for HSIs (see the introduction).
2) Rank-two NMF can be solved efficiently, avoiding the use of an iterative procedure as in standard NMF algorithms. In Section II-C, we propose a new rank-two NMF algorithm using convex geometry concepts from HSI (see Algorithm 4).

Suppose for now that we are given a rank-two NMF (W, H) of M. Such a factorization is a 2-D representation of the data; more precisely, it projects the columns of M onto a 2-D pointed cone generated by the columns of W. Hence, a naive strategy to cluster the columns of M is to choose the clusters as follows:

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

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 aforementioned clustering assignment is equivalent to taking

$$C_1 = \{i | x_i \geq \delta\} \quad \text{and} \quad C_2 = \{i | x_i < \delta\} \tag{1}$$

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 Fig. 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:

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

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

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

where $\underline{\delta} = \max(0, \delta - \hat{\delta})$ and $\bar{\delta} = \min(1, \delta + \hat{\delta})$, $\delta \in [0,1]$, 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 paper, we use $\hat{\delta} = 0.05$.

Given δ, we obtain two clusters C_1 and C_2 [see (1)]. We propose to choose a value of δ such that the following goals are achieved.

1) The clusters are balanced, i.e., the two clusters contain, if possible, roughly the same number of elements. Mathematically, we would like to have $\hat{F}_X(\delta) \approx 0.5$.
2) The clustering is stable, i.e., if the value of δ is slightly modified, then only a few points are transferred from one cluster to the other. Mathematically, we would like to have $\hat{G}_X(\delta) \approx 0$.

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

$$g(\delta) = \underbrace{-\log\left(\hat{F}_X(\delta)\left(1-\hat{F}_X(\delta)\right)\right)}_{\text{balanced clusters}} + \underbrace{\exp\left(\hat{G}_X(\delta)\right)}_{\text{stable clusters}}. \quad (2)$$

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

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

and the expected value of $\hat{G}_X(\delta)$ is one (see previous discussion). 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 previously, 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 of the experiments from Section III selecting δ minimizing the function

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

and it gave very similar results (sometimes slightly better and sometimes slightly worse). The advantage of the function (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 (i.e., 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 are, 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 HSIs from Section III-D, it is hardly possible to distinguish the solutions with the naked eye).

Fig. 1 illustrates the geometric insight behind the splitting procedure in the case $r = 3$ (see also Section II-D), while Algorithm 1 gives a pseudocode of the full hierarchical procedure.

Algorithm 1 Hierarchical 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.
Output: Disjoint clusters $\mathcal{K}_i (1 \le i \le r)$ with $\cup_i \mathcal{K}_i = \{1,\dots,n\}$.
% Initialization
$\mathcal{K}_1 = \{1, 2, \dots, n\}$ and $\mathcal{K}_i = \emptyset$ for $2 \le i \le r$.
$(\mathcal{K}_1^1, \mathcal{K}_1^2) = $ split$(M, \mathcal{K}_1)$. % See Algorithm 2
$\mathcal{K}_i^1 = \mathcal{K}_i^2 = \emptyset$ for $2 \le i \le r$.
for $k = 2$: $r$ do
    % Select the cluster to split; see Section II-A
    Let $j = \arg\max_i \sigma_1^2(M(:,\mathcal{K}_i^1)) + \sigma_1^2(M(:,\mathcal{K}_i^2) - \sigma_1^2(M(:,\mathcal{K}_i))$.
    % Update the clustering
    $\mathcal{K}_j = \mathcal{K}_j^1$ and $\mathcal{K}_k = \mathcal{K}_j^2$.
    % Split the new clusters (Algorithm 2)
    $(\mathcal{K}_\ell^1, \mathcal{K}_\ell^2) = $ split$(M, \mathcal{K}_\ell)$ for $\ell = j, k$.
end for
```

Algorithm 2 Splitting of an 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: Two disjoint clusters $\mathcal{K}^1$ and $\mathcal{K}^2$ with $\mathcal{K}_1 \cup \mathcal{K}_2 = \mathcal{K}$.
$(W, H) = $ rank-twoNMF$(M(:,\mathcal{K}))$ (Algorithm 4).
Let $x(i) = H(1,i)/(H(1,i) + H(2,i))$ for $1 \le i \le |\mathcal{K}|$.
Compute $\delta^*$ as the minimum of $g(\delta)$ defined in (2).
$\mathcal{K}^1 = \{\mathcal{K}(i) \mid x(i) \ge \delta^*\}$ and $\mathcal{K}^2 = \{\mathcal{K}(i) \mid x(i) < \delta^*\}$.
```

C. *Rank-Two NMF for HSIs*

In this section, we propose a simple and fast algorithm for the rank-two NMF problem tailored for HSIs (Section II-C1). Then, we discuss some sufficient conditions for the algorithm to be optimal (Section II-C2).

1) Description of the Algorithm: When a nonnegative matrix $M \in \mathbb{R}^{m\times n}_+$ has rank two, Thomas has shown [17] 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 [18]). This can be explained geometrically as follows: Viewing the 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 2-D subspace. Furthermore, because these columns are nonnegative, they belong to a 2-D pointed cone. 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

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

$r = 2$. 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 done, for example, using any endmember extraction algorithm under the LMM 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 the introduction). In this paper, we use the SPA [11], 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 [6]. Note that SPA is closely related to the automatic target generation process algorithm [19] and the successive volume maximization algorithm [20] (see [21] 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 [22] or WAVMAX [20] which are more robust variants of SPA (although computationally much more expensive).

Algorithm 3 Successive Projection Algorithm (SPA) [6], [11]

```
Input: Separable matrix $M = W[I_r, H']\Pi$ where $H' \geq 0$,
    $\|H(:,j)\|_1 \leq 1 \; \forall j$, $W$ is full rank and $\Pi$ is a permutation,
    and $r$.
Output: Set of indices $K$ such that $M(:,K) = W$ (up to
    permutation).
Let $R = M$, $K = \{\}$.
for $i = 1$: $r$ do
    $k = \arg\max_j \|R_{:j}\|_2$.
    $R \leftarrow (I - (R_{:k}R_{:k}^T / \|R_{:k}\|_2^2))R$.
    $K = K \cup \{k\}$.
end for
```

We can now describe our proposed rank-two NMF algorithm for HSI: It first projects the columns of M into a 2-D linear space using the SVD (note that, if the rank of the 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 HSIs

```
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: $[U, S, V^T]$ = svds$(M, 2)$; % MATLAB function svds
3: Let $X = SV (= U^T USV = U^T M)$;
4: % Extract two indices using SPA
5: $K$ = SPA$(X, 2)$; % See Algorithm 3
6: $W = \max(0, USV(:, K))$;
7: $H = \arg\min_{Y \geq 0} \|M - WY\|_F^2$; % See Algorithm 5
```

Algorithm 5 Nonnegative Least Squares with Two Variables [9]

```
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 least squares problem
$x = \arg\min_x \|Ax - b\|_2$.
if $x \geq 0$, then return.
% Compute the solutions for $x(1) = 0$ and $x(2) = 0$ (the
    two possible active sets)
Let $y = (0, \max(0, (A(:,1)^T b / \|A(:,1)\|_2^2)))$,
    and $z = (\max(0, (A(:,2)^T b / \|A(:,2)\|_2^2)), 0)$.
if $\|Ay - b\|_2 < \|Az - b\|_2$ then
    $x = y$, else $x = z$.
end if
```

Let us analyze the computational cost of Algorithm 4. The computation of the rank-two SVD of M is $\mathcal{O}(mn)$ operations [23]. (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 HSIs, m is much smaller than n (usually $m \sim 200$, while $n \sim 10^6$); hence, it is faster to compute the SVD of M using the SVD of MM^T which requires $2mn + O(m^2)$ operations (see, for example, [10]). 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 [6], while computing the optimal H requires solving n linear systems in two variables for a total computational cost of $\mathcal{O}(mn)$ operations [9]. 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 [9] 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) Theoretical Motivations: As mentioned previously, 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 $M = FG = \sum_{k=1}^{2} F(:,k)G(k,:)$ [17]. Moreover, since the entries of each column of M sum to one, we can assume without loss of generality that the entries of 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

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

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 [6, 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 LMM 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 the introduction). The rest of the proof follows the second part of the proof of Theorem 1 (note that the pure-pixel assumption is not necessary). ■

In practice, the sum-to-one constraint 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 contain "background" pixels with zero spectral signatures or taking into account different intensities of light among the pixels in the image (see, for example, [1]). 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 LMM 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 $W(:,2)$ [6, 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 2-D 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

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

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

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

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 HSIs usually have mostly positive entries, and in fact, we have observed that the best rank-two approximation of real-world HSIs 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 III-D 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 that 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 III 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, although the theory in this paper is developed for rank-two NMF only, it has great significance in clustering HSIs with more than two endmembers.

D. Geometric Interpretation of the Splitting Procedure

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

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

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 the introduction). A well-known fact in convex geometry is that the projection of any polytope P into an affine subspace generates another polytope, e.g., 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 [10] uses three kinds of projections. First, it projects the data into an r-dimensional space using the SVD (in order to reduce the noise). Then, at each step, the following operations are performed.

1) In order to identify a vertex (i.e., an endmember), VCA projects conv(M) onto a 1-D subspace. More precisely, it randomly generates a vector $c \in \mathbb{R}^m$ and then selects the columns of M maximizing $c^T M(:, i)$.
2) It projects all columns of M onto the orthogonal complement of the extracted vertex so that, if W is full rank (i.e., if conv(M) has r vertices and has dimension $r - 1$), the projection of 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.

1) At the first step, the data points are projected into a 2-D subspace so that the maximum variance is preserved.
2) At the second step, two vertices are extracted by SPA.
3) At the third step, the data points are projected onto the 2-D convex cone generated by these two vertices.

E. 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 [24], 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, which could discover clusters not corresponding to any vertex of the conv(M) but lying inside conv(M), and an algorithm based on FastMap [26] was used. In [9], hierarchical clustering based on rank-two NMF was used for document classification. The rank-two subproblems were solved using alternating nonnegative least squares [27], [28], i.e., 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:

1) 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 II-A);
2) the way the clusters are split based on a trade-off between having balanced clusters and stable clusters (see Section II-B);
3) the use of a rank-two NMF technique tailored for HSIs (using their convex geometry properties) to design a splitting procedure (see Section II-C).

III. 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 performed using MATLAB on a laptop Intel CORE i5-3210M CPU @2.5 GHz 2.5 GHz 6 Go RAM.

A. Tested Algorithms

We will compare the following algorithms.

1) **H2NMF**: hierarchical clustering based on rank-two NMF (see Algorithm 1 and Section II).
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 II-B (we used the `kmeans` function of MATLAB).
3) **HSPKM**: hierarchical clustering based on spherical k-means [29]. 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 [30]. 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 that for H2NMF, i.e., 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 identify the clusters perfectly in noiseless conditions). Recall that SPA is optimal for HSIs satisfying the pure-pixel assumption [6]; hence, it is a reasonable initialization.

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

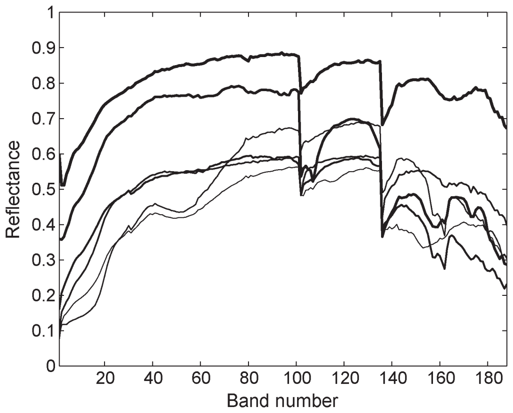


Fig. 2. Endmembers from the Cuprite HSI used for the synthetic data sets.

B. 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.

1) We use six endmembers, i.e., $r = 6$.
2) The spectral signatures of the six endmembers, i.e., the columns of W, are taken as the spectral signatures of materials from the Cuprite HSI (see Section III-E3), and we have $W \in \mathbb{R}_+^{188 \times 6}$ (see Fig. 2). Note that W is rather poorly conditioned ($\kappa(W) = 91.5$) as the spectral signatures look very similar to one another.
3) 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).
4) Once a pixel, e.g., the ith, has been assigned to a cluster, e.g., the kth, the corresponding column of H is generated as follows: $H(:, i) = 0.9e_k + 0.1x$, 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.
5) 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.
6) If $b = 1$, then 10 outliers and 40 background pixels with zero spectral signatures are added to M, i.e., $Z = [z_1, z_2, \ldots, z_{10}, 0_{m \times 40}]$, 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_ps are scaled as follows:
$$z_p \leftarrow K_W \frac{z_p}{\|z_p\|_2} \quad 1 \leq p \leq 10$$
where $K_W = (1/r) \sum_{k=1}^{r} \|W(:, k)\|_2$ 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, i.e., Z is the empty matrix.
7) 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 once it has generated r clusters $\mathcal{K}'_k (1 \leq k \leq r)$, its performance is evaluated using the following criterion:

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

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).

C. Results

For each noise level ϵ and each value of s and b, we generate 25 synthetic HSIs as described in Section III-B. Fig. 3 reports the average accuracy; hence, the higher the curve, the better.

We observe the following.

1) In almost all cases, the hierarchical clustering techniques consistently outperform the plain clustering approaches.
2) Without scaling nor outliers (top left of Fig. 3), HKM performs the best, while H2NMF is second best.
3) With scaling but without outliers (top right of Fig. 3), H2NMF performs the best, slightly better than SPKM, while HKM performs rather poorly. This shows that HKM is sensitive to scaling (i.e., to different illumination conditions in the image), which will be confirmed on the real-world HSIs.
4) With outliers but without scaling (bottom left of Fig. 3), 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.
5) With scaling and outliers (bottom right of Fig. 3), HKM performs even worse. H2NMF still outperforms all other algorithms, while HSPKM extracts relatively good clusters compared to the other approaches.

Table I 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.

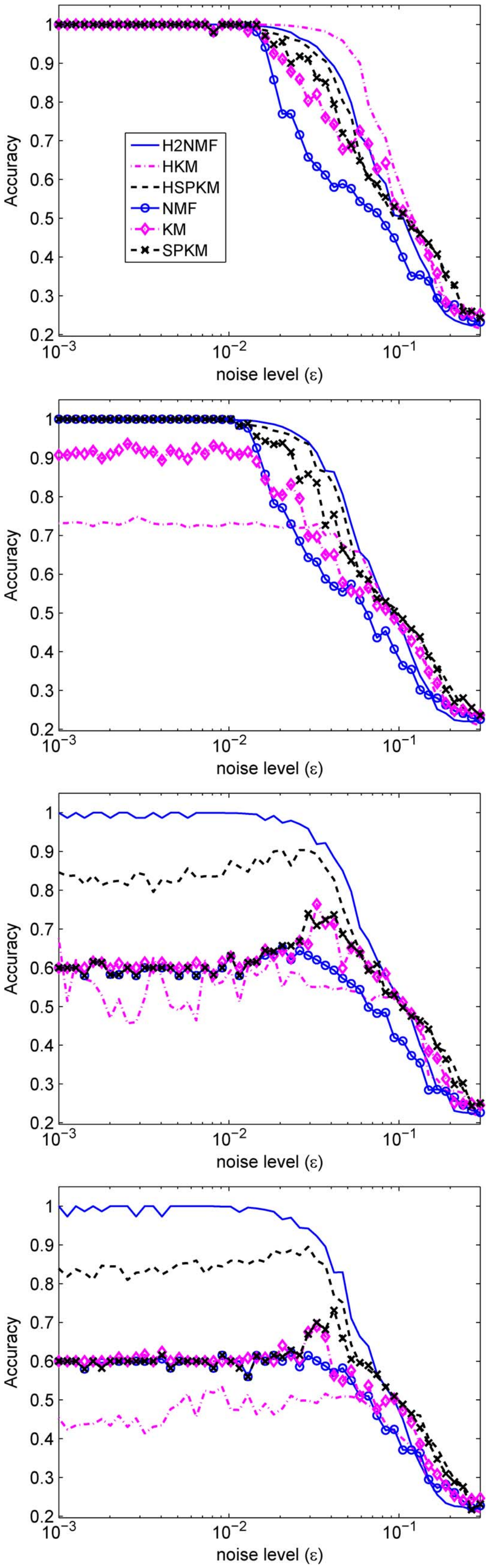


Fig. 3. Performance of the different algorithms on synthetic data sets. From top to bottom: $(s, b) = (0, 0)$, $(1, 0)$, $(0, 1)$, and $(1, 1)$.

TABLE I
Average Running Time in Seconds for the Different Algorithms on the Synthetic Data Sets

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

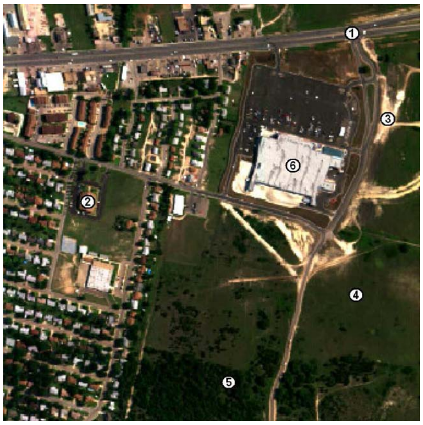


Fig. 4. Urban HSI set taken from an aircraft (Army Geospatial Center).

D. Real-World HSIs

In this section, we show that H2NMF is able to perform very good clustering of high-resolution real-world HSIs. This section will focus on illustrating three important contributions: 1) H2NMF performs better than standard clustering techniques on real-world HSI; 2) although H2NMF has been designed to deal with HSIs with pixels dominated mostly by one endmember, it can provide meaningful and useful results in more difficult settings; and 3) H2NMF can be used as an endmember extraction algorithm in the presence of pure pixels (we compare it to VCA [10] and the SPA [11]). Note that, because the ground truth of these HSIs is not known precisely, it is difficult to provide an objective quantitative measure for the cluster quality.

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 II-A, it makes sense to approximate these matrices with their best rank-one approximation: For $1 \leq k \leq r$

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

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, for example, [31]). Given two spectral signatures $(x, y \in \mathbb{R}^m)$, it is defined as

$$\phi(x, y) = \frac{1}{\pi} \arccos\left(\frac{(x - \bar{x})^T (y - \bar{y})}{\|x - \bar{x}\|_2 \|y - \bar{y}\|_2}\right) \in [0, 1] \qquad (3)$$

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.

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

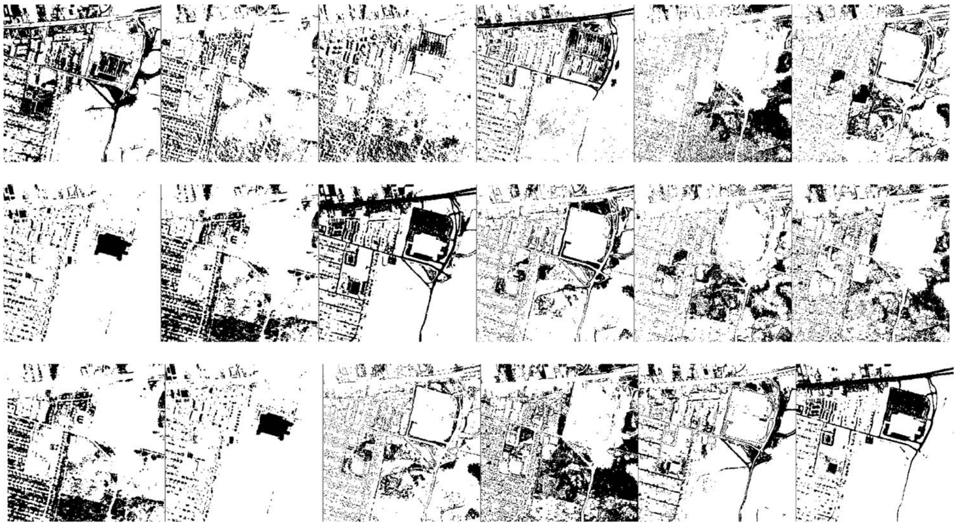


Fig. 5. Clustering of the Urban HSI. From top to bottom: HKM, HSPKM, and H2NMF.

pure-pixel assumption, while H2NMF requires that most pixels are dominated mostly by one endmember).

2) Urban HSI: The Urban HSI[3] from the HYperspectral Digital Imagery Collection Experiment (HYDICE) contains 162 clean spectral bands, and the data cube has a dimension of $307 \times 307 \times 162$. The Urban data set is a rather simple and well-understood data set: It is mainly composed of six types of materials (road, dirt, trees, roof, grass, and metal) as reported in [32] (see Figs. 4 and 7). Fig. 5 displays the clusters obtained with H2NMF, HKM, and HSPKM.[4] We observe the following.

1) 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 III-B).
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.
3) 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: 1) The grass is the largest cluster and actually contains two subclasses with slightly different spectral signatures (as reported in [34]; see also Fig. 8); 2) the metal is a very small cluster; and 3) the spectral signatures of the road and metal are not so different (see Fig. 4). 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 tend to be larger when the matrix contains more columns (see Section II-A). Although it works well in many situations (particularly when clusters are relatively well balanced), other criterions might be preferable in some cases; this is a topic for further research.

Fig. 6 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 identify all materials separately (see Fig. 7; note that this is not possible with HKM and HSPKM).

Using the strategy described in Section III-D1, we now compare the different algorithms when they are used for endmember extraction. Fig. 8 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 $w_k (1 \leq k \leq r)$ obtained using the N-FINDR5 algorithm [35] plus manual adjustment [32] so that $\sum_{k=1}^{r} \phi(w_k, w'_k)$ is minimized [see (3)]. Table II 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.

E. Additional Experiments on Real-World HSIs

In this section, our goal is not to compare the different clustering strategies (due to the space limitation) but rather to show that H2NMF can give good results for other real-world and widely used data sets.

1) San Diego Airport HSI: The San Diego airport is a HYDICE HSI containing 158 clean bands and 400×400 pixels for each spectral image (i.e., $M \in \mathbb{R}_{+}^{158 \times 160000}$). There are mainly four types of materials: road surfaces, roof, trees, and grass (see, for example, [36]). There are three types of road surfaces, including boarding and landing zones, parking lots, and streets, and two types of roof tops.[6] H2NMF took 33.6 s, and Fig. 9

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

[4] The clustering obtained with KM and SPKM can be found in [33]; the clustering obtained with KM is rather poor, while the one obtained with SPKM is similar to the one obtained with HSPKM.

[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 [36], only one type of roof top is identified.

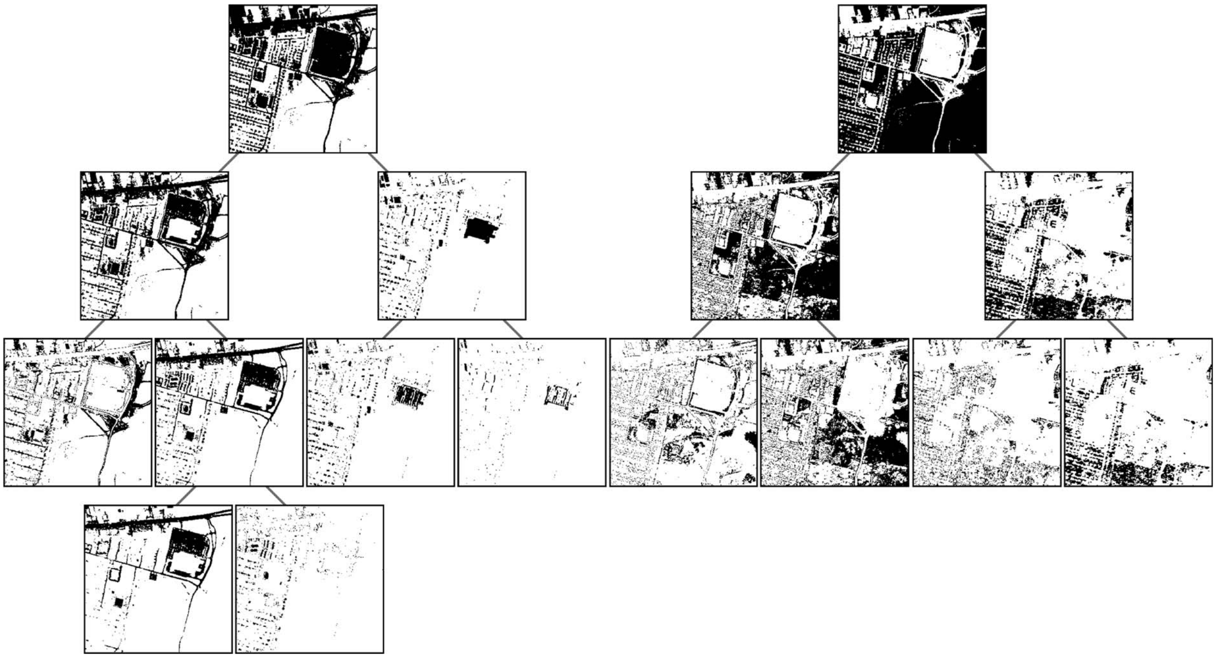


Fig. 6. Hierarchical structure of H2NMF for the Urban HSI.

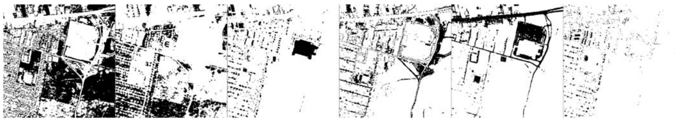


Fig. 7. Interactive H2NMF (I-H2NMF) of the Urban HSI (see also Fig. 6). From left to right: grass, trees, roof, dirt, road, and metal.

displays the first levels of the cluster hierarchy of H2NMF. It is interesting to notice that roads 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). More computational results with comparison with HKM, HSPKM, SPA, and VCA can be found in the arXiv version [37] of this paper.

2) Terrain HSI: The Terrain HSI is available from http://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, and thin and tick grass (see, for example, http://www.way2c.com/rs2.php). H2NMF took 20.3 s to perform the clustering shown in Fig. 10. H2NMF is able to identify the five clusters extremely well, while HKM and HSPKM are not able to separate bare soil and thick and thin grass properly.

3) Cuprite HIS: Cuprite is a mining area in southern Nevada with mostly mineral and very little vegetation, located approximately 200 km northwest of Las Vegas (see, for example, [10] and [31] for more information and http://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 a mixture of several endmembers. Hence, this experiment illustrates the usefulness of H2NMF to analyze more difficult data sets, where the assumption that most pixels are dominated mostly by one endmember is only roughly satisfied (see Fig. 11). We run H2NMF with $r = 15$, which took 11.6 s.

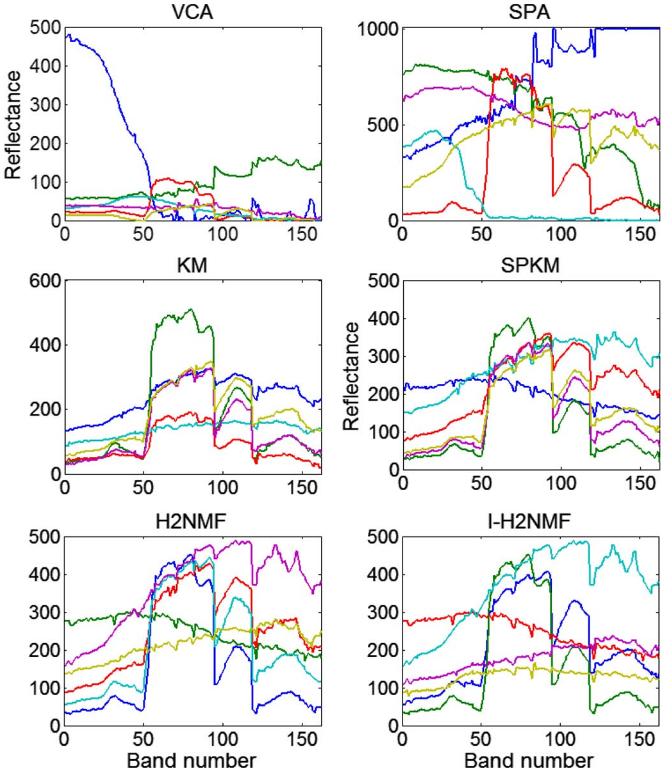


Fig. 8. Spectral signatures extracted for the Urban HSI.

TABLE II
RUNNING TIMES AND MRSA (IN PERCENT) FOR THE URBAN HSI

	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**

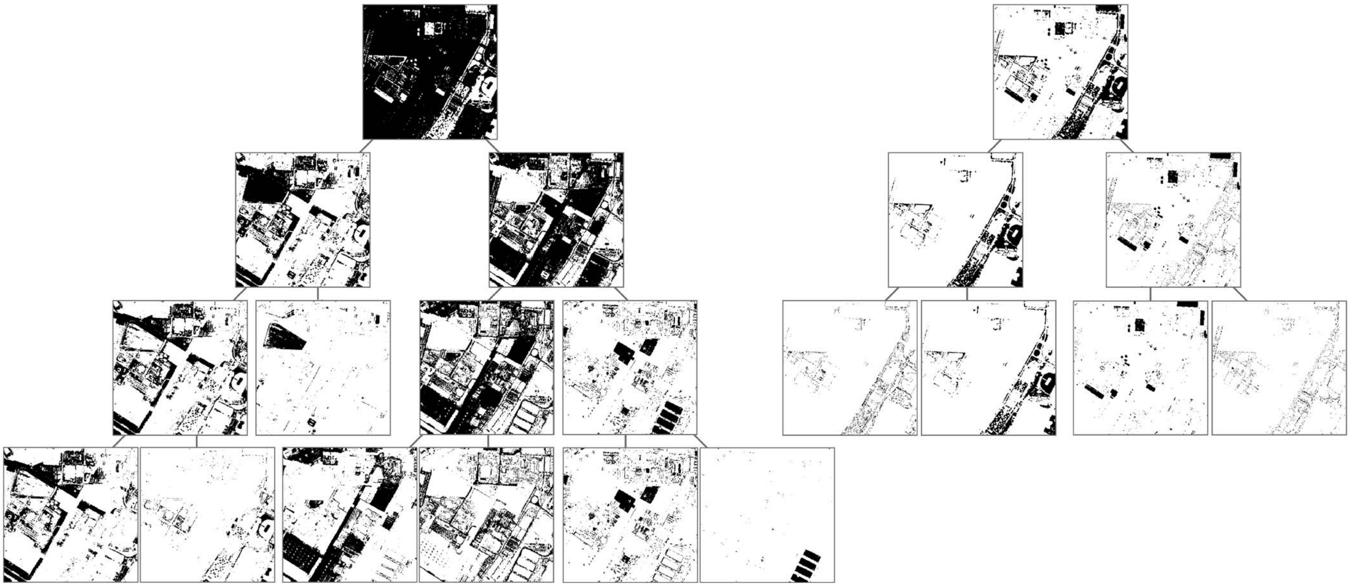


Fig. 9. Hierarchical structure of H2NMF for the San Diego airport HSI.

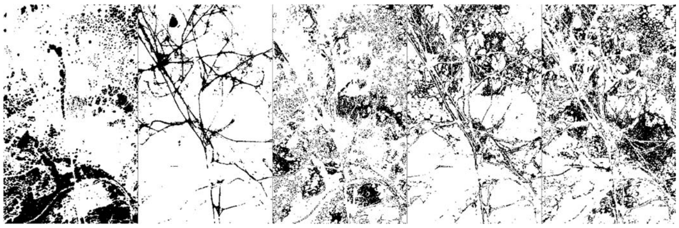


Fig. 10. Five clusters obtained automatically with H2NMF on the Terrain HSI. From left to right: tree, road, thick grass, bare soil, and thin grass.

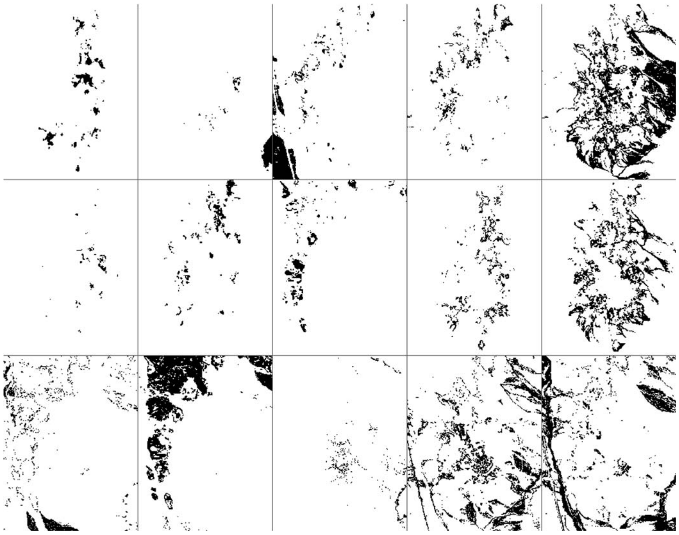


Fig. 11. Fifteen clusters obtained automatically with H2NMF on the Cuprite HSI. Some materials can be distinguished, e.g., (1) alunite, (2) montmorillonite, (3) goethite, (5) hematite, (8)–(12) desert varnish, (11) iron oxydes, and (15) kaolinite (counting from left to right and top to bottom).

IV. Conclusion and Further Work

In this paper, we have introduced a way to perform hierarchical clustering of high-resolution HSIs using the geometry of such images and the properties of rank-two NMF [see Algorithm 1 (referred to as H2NMF)]. We have 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 bands, n is the number of pixels, and r is the number of clusters). Although high-resolution HSIs usually have low noise levels, one of the reasons 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 [1] (hence, robustness to noise is a desirable property even for high-resolution HSIs). Moreover, we have 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 HSIs to perform the clustering. In particular, using the spatial information (i.e., 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).

Acknowledgment

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

References

[1] J. Bioucas-Dias *et al.*, “Hyperspectral unmixing overview: Geometrical, statistical, sparse regression-based approaches,” *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 5, no. 2, pp. 354–379, Apr. 2012.

[2] S. Vavasis, “On the complexity of nonnegative matrix factorization,” *SIAM J. Optim.*, vol. 20, no. 3, pp. 1364–1377, Aug. 2009.

[3] N. Gillis, “Sparse and unique nonnegative matrix factorization through data preprocessing,” *J. Mach. Learn. Res.*, vol. 13, no. 1, pp. 3349–3386, Jan. 2012.

[4] S. Jia and Y. Qian, “Constrained nonnegative matrix factorization for hyperspectral unmixing,” *IEEE Trans. Geosci. Remote Sens.*, vol. 47, no. 1, pp. 161–173, Jan. 2009.

[5] A. Zymnis, S.-J. Kim, J. Skaf, M. Parente, and S. Boyd, “Hyperspectral image unmixing via alternating projected subgradients,” in *Conf. Rec. 41st Asilomar Conf. Signals, Syst. Comput.*, 2007, pp. 1164–1168.

[6] N. Gillis and S. Vavasis, “Fast and robust recursive algorithms for separable nonnegative matrix factorization,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36, no. 4, pp. 698–714, Apr. 2014.

[7] S. Arora, R. Ge, R. Kannan, and A. Moitra, “Computing a nonnegative matrix factorization—Provably,” in *Proc. 44th Annu. ACM Symp. Theory Comput., STOC’12*, 2012, pp. 145–162.

[8] N. Gillis, “Robustness analysis of Hottopixx, a linear programming model for factoring nonnegative matrices,” *SIAM J. Mat. Anal. Appl.*, vol. 34, no. 3, pp. 1189–1212, 2013.

[9] D. Kuang and H. Park, “Fast rank-2 nonnegative matrix factorization for hierarchical document clustering,” in *Proc. 19th ACM SIGKDD Conf. Knowl. Discov. Data Mining, KDD’13*, 2013, pp. 739–747.

[10] J. Nascimento and J. Bioucas-Dias, “Vertex component analysis: A fast algorithm to unmix hyperspectral data,” *IEEE Trans. Geosci. Remote Sens.*, vol. 43, no. 4, pp. 898–910, Apr. 2005.

[11] U. Araújo *et al.*, “The successive projections algorithm for variable selection in spectroscopic multicomponent analysis,” *Chemometrics Intell. Lab. Syst.*, vol. 57, no. 2, pp. 65–73, Jul. 2001.

[12] J. Bioucas-Dias and J. Nascimento, “Estimation of signal subspace on hyperspectral data,” in *Proc. SPIE Image Signal Process. Remote Sens.*, 2005, p. 59820L.

[13] Y. Zhao, G. Karypis, and U. Fayyad, “Hierarchical clustering algorithms for document datasets,” *Data Mining Knowl. Discov.*, vol. 10, no. 2, pp. 141–168, Mar. 2005.

[14] D. Cai, X. He, Z. Li, W.-Y. Ma, and J.-R. Wen, “Hierarchical clustering of WWW image search results using visual, textual and link information,” in *Proc. 12th Annu. ACM Int. Conf. Multimedia*, 2004, pp. 952–959.

[15] M. Eisen, P. Spellman, P. Brown, and D. Botstein, “Cluster analysis and display of genome-wide expression patterns,” in *Proc. Nat. Academy Sci. USA*, 1998, vol. 95, no. 25, pp. 14 863–14 868.

[16] P. Arbelaez, M. Maire, C. Fowlkes, and J. Malik, “Contour detection and hierarchical image segmentation,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 5, pp. 898–916, May 2011.

[17] L. Thomas, “Rank factorization of nonnegative matrices,” *SIAM Rev.*, vol. 16, no. 3, pp. 393–394, 1974.

[18] J. Cohen and U. Rothblum, “Nonnegative ranks, decompositions and factorization of nonnegative matrices,” *Linear Algebra Appl.*, vol. 190, pp. 149–168, Sep. 1993.

[19] H. Ren and C.-I. Chang, “Automatic spectral target recognition in hyperspectral imagery,” *IEEE Trans. Aerosp. Electron. Syst.*, vol. 39, no. 4, pp. 1232–1249, Oct. 2003.

[20] T.-H. Chan, W.-K. Ma, A. Ambikapathi, and C.-Y. Chi, “A simplex volume maximization framework for hyperspectral endmember extraction,” *IEEE Trans. Geosci. Remote Sens.*, vol. 49, no. 11, pp. 4177–4193, Nov. 2011.

[21] W.-K. Ma *et al.*, “Signal processing perspective on hyperspectral unmixing,” *IEEE Signal Process. Mag.*, vol. 31, no. 1, pp. 67–81, 2014.

[22] A. Ambikapathi, T.-H. Chan, W.-K. Ma, and C.-Y. Chi, “A robust alternating volume maximization algorithm for endmember extraction in hyperspectral images,” in *Proc. WHISPERS*, Reykjavik, Iceland, 2010, pp. 1–4.

[23] G. Golub and C. Van Loan, *Matrix Computation*, 3rd ed. Baltimore, MD, USA: The Johns Hopkins Univ. Press, 1996.

[24] Y. Li *et al.*, “Hierarchical non-negative matrix factorization (HNMF): A tissue pattern differentiation method for glioblastoma multiforme diagnosis using MRSI,” *NMR Biomed.*, vol. 26, no. 3, pp. 307–319, Mar. 2012.

[25] K. Kersting, M. Wahabzada, C. Thurau, and C. Bauckhage, “Hierarchical convex NMF for clustering massive data,” in *Proc. 2nd ACML*, 2010, pp. 253–268.

[26] C. Faloutsos and K.-I. Lin, “FastMap: A fast algorithm for indexing, data-mining and visualization of traditional and multimedia datasets,” in *Proc. ACM SIGMOD Int. Conf. Manag. Data, SIGMOD ’95*, 1995, pp. 163–174.

[27] H. Kim and H. Park, “Sparse non-negative matrix factorizations via alternating non-negativity-constrained least squares for microarray data analysis,” *Bioinformatics*, vol. 23, no. 12, pp. 1495–1502, Jun. 2007.

[28] J. Kim and H. Park, “Fast nonnegative matrix factorization: An active-set-like method and comparisons,” *SIAM J. Sci. Comput.*, vol. 33, no. 6, pp. 3261–3281, 2011.

[29] A. Banerjee, I. Dhillon, J. Ghosh, and S. Sra, “Generative model-based clustering of directional data,” in *Proc. 19th ACM SIGKDD Int. Conf. KDD Mining*, 2003, pp. 19–28, ACM Press.

[30] N. Gillis and F. Glineur, “Accelerated multiplicative updates and hierarchical ALS algorithms for nonnegative matrix factorization,” *Neural Comput.*, vol. 24, no. 4, pp. 1085–1105, Apr. 2012.

[31] A. Ambikapathi, T.-H. Chan, W.-K. Ma, and C.-Y. Chi, “Chance-constrained robust minimum-volume enclosing simplex algorithm for hyperspectral unmixing,” *IEEE Trans. Geosci. Remote Sens.*, vol. 49, no. 11, pp. 4194–4209, Nov. 2011.

[32] Z. Guo, T. Wittman, and S. Osher, “L1 unmixing and its application to hyperspectral image enhancement,” presented at the SPIE Conf. Algorithms Technol. Multispectral, Hyperspectral, Ultraspectral Imagery XV, 2009.

[33] F. Pompili, N. Gillis, P.-A. Absil, and F. Glineur, “Two algorithms for orthogonal nonnegative matrix factorization with application to clustering,” *Neurocomputing*, vol. 141, pp. 15–25, Oct. 2014.

[34] N. Gillis and R. Plemmons, “Dimensionality reduction, classification, spectral mixture analysis using nonnegative underapproximation,” *Opt. Eng.*, vol. 50, no. 2, p. 027001, Feb. 2011.

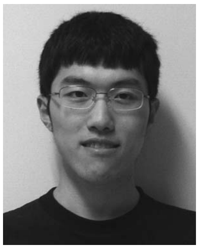
[35] M. Winter, “N-FINDR: An algorithm for fast autonomous spectral endmember determination in hyperspectral data,” presented at the SPIE Conf. Imag. Spectrometry V, 1999.

[36] N. Gillis and R. Plemmons, “Sparse nonnegative matrix underapproximation and its application to hyperspectral image analysis,” *Linear Algebra Appl.*, vol. 438, no. 10, pp. 3991–4007, May 2013.

[37] N. Gillis, D. Kuang, and H. Park, “Hierarchical clustering of hyperspectral images using rank-two nonnegative matrix factorization,” 2013, pp. 1–29, arXiv:1310.7441.

Nicolas Gillis received the Master’s degree and the Ph.D. degree in applied mathematics from Université catholique de Louvain, Louvain-la-Neuve, Belgium, in 2007 and 2011, respectively.

He is currently an Assistant Professor with the Department of Mathematics and Operational Research, Faculté Polytechnique, Université de Mons, Mons, Belgium. His research interests lie in optimization, numerical linear algebra, machine learning, and data mining.



Da Kuang received the Bachelor’s degree in computer science from Tsinghua University, Beijing, China, in 2009 and the Ph.D. degree in computational science and engineering from Georgia Institute of Technology, Atlanta, GA, USA, in 2014.

His research interests are numerical methods and high-performance computing for large-scale machine learning and data analytics.

Haesun Park received the B.S. degree in mathematics from Seoul National University, Seoul, Korea, with the University President’s Medal for the top graduate and the M.S. and Ph.D. degrees in computer science in 1985 and 1987 from Cornell University, Ithaca, NY, USA.

She is a Professor with the School of Computational Science and Engineering, Georgia Institute of Technology (Georgia Tech), Atlanta, GA, USA. Her research areas include numerical algorithms, data analysis, visual analytics, text mining, and parallel computing. She has played major leadership roles in these areas as the Executive Director of the Center for Data Analytics, Georgia Tech, and the Director of the NSF/DHS-funded FODAVA (Foundations of Data and Visual Analytics) Center.

Dr. Park is a Fellow of the Society for Industrial and Applied Mathematics (SIAM). She is the General Chair of the SIAM Conference on Data Mining, an editorial board member of SIAM and IEEE journals, and has been a plenary keynote speaker at major conferences.