

Implementing Hottopixx Methods for Endmember Extraction in Hyperspectral Images

Tomohiko Mizutani

Abstract—Hyperspectral imaging technology has a wide range of applications, including forest management, mineral resource exploration, and Earth surface monitoring. Endmember extraction of hyperspectral images is a key step in leveraging this technology for applications. It aims to identifying the spectral signatures of materials, i.e., the major components in the observed scenes. Theoretically speaking, Hottopixx methods should be effective on problems involving extracting endmembers from hyperspectral images. Yet, these methods are challenging to perform in practice, due to high computational costs. They require us to solve LP problems, called Hottopixx models, whose size grows quadratically with the number of pixels in the image. It is thus still unclear as to whether they are actually effective or not. This study clarifies this situation. We propose an efficient and effective implementation of Hottopixx. Our implementation follows the framework of column generation, which is known as a classical but powerful means of solving large-scale LPs. We show in experiments that our implementation is applicable to the endmember extraction from real hyperspectral images and can provide estimations of endmember signatures with higher accuracy than the existing methods can.

Index Terms—Hyperspectral imaging, endmember extraction, nonnegative matrix factorization, linear mixing model, pure pixel assumption, linear programming, column generation method.

I. INTRODUCTION

A hyperspectral camera is an optical instrument to measure the spectra of materials in a scene. Hyperspectral images (HSIs) are images acquired by it, and typically consist of several hundred spectral bands. Endmember extraction of HSIs aims to identifying the spectral signatures of materials, i.e., the major components in the observed scenes. This process is a key step for leveraging hyperspectral imaging for applications. Let us formulate the problem of endmember extraction of HSIs built on the assumptions that HSI data follow a linear mixing model (LMM) and pure pixels exist for every endmember. These assumptions are often used for tackling the problem [5], [20] and many algorithms have been developed under them; see Section IV-A of [5] for instance. The endmember extraction problem under these assumptions is known to be equivalent to the problem of computing a nonnegative matrix factorization (NMF) under the separability assumption.

Hottopixx methods were originally proposed by Bittorf et al. [6] in 2012 as way of solving separable NMF problems. Separable NMFs have been used for extracting topics from documents as well as endmembers from HSIs; see [10], [15], for instance. The Hottopixx of Bittorf et al. was developed in context of the former. Since then, several authors [13],

[16], [22] have refined Hottopixx and have examined the performance of their refinements in theory and in practice. Their algorithmic framework can be described as follows: first, solve a linear programming problem (LP), called a Hottopixx model; second, conduct clustering-based postprocessing using the optimal solution of the LP; and then, output one element from each obtained cluster. The theoretical results shown in [13], [16], [22] imply that Hottopixx methods are effective on endmember extraction problems.

Yet, it remains challenging to perform Hottopixx methods in practice, because solving Hottopixx models is costly; the size of the models grows quadratically with the number of pixels in the HSI. It is thus unclear whether these methods are actually effective on endmember extraction problems. This study clarifies this situation. We propose an efficient and effective implementation of Hottopixx, EEHT for short. We show in experiments that EEHT can significantly reduce the computational time for solving endmember extraction problems and can estimate endmember signatures with higher accuracy than existing methods can.

The contributions of this study are summarized as follows.

a) *Reducing the computational cost of Hottopixx*: Solving Hottopixx models is computationally costly, which is an obstacle to apply it to endmember extraction problems. To remedy this issue, we propose a row and column expansion (RCE) algorithm for solving Hottopixx models efficiently. The algorithm is designed by following the framework of column generation, which is known as a classical but powerful means of solving large-scale LPs.

Many zero elements could exist in the optimal solution of a Hottopixx model. This suggests that Hottopixx models can be solved by breaking them up into smaller subproblems and solving them. We show in Theorem 2 the conditions under which the optimal solution of a subproblem yields that of the Hottopixx model. We then develop an RCE algorithm on the basis of this theoretical result. The details are described in Section IV.

b) *Enhancing the endmember extraction performance of Hottopixx*: Hottopixx conducts clustering-based postprocessing and outputs one element from each cluster. In practice, the choice of elements affects the endmember extraction performance of Hottopixx. The methods of choice used in the previous studies [13], [16], [22] were based on the optimal solutions of Hottopixx models. Alternatively, we propose to use a method based on the shape of clusters; it computes the centroids of each cluster and chooses elements close to them. This method is simple but effective in enhancing the endmember extraction performance of Hottopixx. We will call it the cluster centroid choice. The details are described in

Section V.

c) *Demonstrating the performance of EEHT by experiments:* We developed EEHT by incorporating RCE and the cluster centroid choice. We then experimentally tested its performance on endmember extraction problems.

The first experiment (Section VI-A) examined the computational time of RCE. The results are summarized in Figure 1. It shows that RCE is faster than the direct method of solving Hottopixx models. In particular, if RCE uses dimensionality reduction based on a singular value decomposition (SVD) technique as preprocessing, it can significantly reduce the computational time.

The second experiment (Section VI-B) examined the endmember extraction performance of EEHT on semi-real HSI datasets. The datasets were constructed from two real HSIs, Jasper Ridge and Samson. We compared EEHT with six existing algorithms, SPA [1], [18], PSPA [19], ER [21], SNPA [14], VCA [23], and FGNSR [17]. The results, summarized in Figure 2, indicate that EEHT can often give more accurate estimations for endmember signatures than the existing methods can.

Finally, we conducted an experimental study (Section VI-C) on hyperspectral unmixing of the Urban HSI dataset. The results are summarized in Tables II, III, and Figure 4. They show that, compared with the existing methods, EEHT usually gives more accurate estimations for endmember signatures in Urban.

A. Organization of This Paper

This paper is organized as follows. Section II is devoted to preliminaries. Section III formulates the endmember extraction problems and explains the algorithm of Hottopixx. It also reviews related methods. Section IV and Section V present the details of RCE and EEHT. Section VI describes the experiments.

II. PRELIMINARIES

A. Notation

$\mathbf{0}$ denotes a vector of all zeros, $\mathbf{1}$ a vector of all ones, O a matrix of all zeros, I the identity matrix, and J a matrix of all ones. $\mathbb{R}_+^{d \times n}$ denotes the set of nonnegative matrices.

Let $\mathbf{a} \in \mathbb{R}^d$ and $A \in \mathbb{R}^{d \times n}$. The notation $\mathbf{a}(i)$ indicates the i th entry of $\mathbf{a}$. In a similar way, $A(i, j)$ indicates the (i, j) th entry of A , $A(:, j)$ the j th column, and $A(i, :)$ the i th row. We write $\text{tr}(A)$ for the trace of A . We write $\text{diag}(\mathbf{a})$ for a diagonal matrix of size d containing the elements of $\mathbf{a}$ in the diagonal positions. Here, assume that A is a square matrix, i.e., $d = n$. We write $\text{diag}(A)$ for a vector of size d consisting of the diagonal elements of A .

Let $A = [\mathbf{a}_1, \dots, \mathbf{a}_n] \in \mathbb{R}^{d \times n}$ and $\mathcal{I} \subset \{1, \dots, n\}$. We use $A(\mathcal{I})$ or $[\mathbf{a}_i : i \in \mathcal{I}]$ to denote the submatrix of A indexed by $\mathcal{I}$ in ascending order. That is, $A(\mathcal{I}) = [\mathbf{a}_i : i \in \mathcal{I}] = [\mathbf{a}_{i_1}, \dots, \mathbf{a}_{i_p}]$ for $\mathcal{I} = \{i_1, \dots, i_p\}$ with $i_1 < \dots < i_p$. In addition, we use $|\mathcal{I}|$ to denote the number of elements in $\mathcal{I}$.

The notation $\|\cdot\|_p$ indicates the L_p -norm of a vector or a matrix, and $\|\cdot\|_F$ the Frobenius norm of a matrix. Moreover,

$\langle A, B \rangle$ indicates the inner product of two matrices A, B of the same size, defined by $\langle A, B \rangle = \text{tr}(A^\top B)$.

For $A \in \mathbb{R}^{d \times n}$ (resp. $\mathbf{a} \in \mathbb{R}^d$), we define A^+ (resp. $\mathbf{a}^+$) to be the matrix (resp. vector) obtained by replacing all negative values of A (resp. $\mathbf{a}$) with zero. Moreover, we define A^- by $A^- = A^+ - A$. Hence, A^+ and A^- satisfy $A^+, A^- \geq O$ and $A = A^+ - A^-$. The L_1 norm of A can be written using A^+ and A^- as follows:

$$\|A\|_1 = \max_{j=1, \dots, n} \sum_{i=1}^d A^+(i, j) + A^-(i, j). \quad (1)$$

In addition, for an optimization problems S , we denote the optimal value of S by $\text{opt}(S)$.

B. Duality of LP Problems

The RCE algorithm for solving Hottopixx models is founded on the duality of LP problems. For $A \in \mathbb{R}^{d \times n}$, $\mathbf{b} \in \mathbb{R}^d$ and $\mathbf{c} \in \mathbb{R}^n$, consider the primal and dual pair of LP,

$$\begin{aligned} (\text{Primal}) \quad & \min \quad \mathbf{c}^\top \mathbf{x} \quad \text{s.t.} \quad A\mathbf{x} = \mathbf{b}, \mathbf{x} \geq \mathbf{0}. \\ (\text{Dual}) \quad & \max \quad \mathbf{b}^\top \mathbf{y} \quad \text{s.t.} \quad A^\top \mathbf{y} \leq \mathbf{c}. \end{aligned}$$

Here, $\mathbf{x} \in \mathbb{R}^n$ and $\mathbf{y} \in \mathbb{R}^d$ are the variables of the primal and dual problems, respectively.

- *The duality theorem:* If the primal and dual are feasible, then, there exist optimal solutions $\mathbf{x}$ and $\mathbf{y}$ to the primal and dual and they satisfy $\mathbf{c}^\top \mathbf{x} = \mathbf{b}^\top \mathbf{y}$.
- *Implication of the weak duality theorem:* Let $\mathbf{x}$ and $\mathbf{y}$ be feasible solutions to the primal and dual. If $\mathbf{c}^\top \mathbf{x} = \mathbf{b}^\top \mathbf{y}$, then, $\mathbf{x}$ and $\mathbf{y}$ are optimal solutions to the primal and dual.

For the details, refer to the textbooks [3], [4]. We will use them in developing RCE.

III. PROBLEM AND METHODS

A. Endmember Extraction of HSIs

Suppose that we are given an HSI data on some area consisting of n pixels acquired by a hyperspectral camera with d spectral bands. We represent the HSI by a matrix $A \in \mathbb{R}^{d \times n}$ such that the (i, j) th entry of A stores the measurement at the i th band and j th pixel. In this paper, we call such an A an *HSI matrix*. The j th column $\mathbf{a}_j$ of A corresponds to the observed spectral signature at the j th pixel.

Let $A = [\mathbf{a}_1, \dots, \mathbf{a}_n] \in \mathbb{R}^{d \times n}$ be an HSI matrix and consider a linear mixing model (LMM) on A . Under the LMM, the observed spectral signatures $\mathbf{a}_1, \dots, \mathbf{a}_n$ at each pixel are written as

$$\mathbf{a}_j = \sum_{i=1}^r h_{ij} \mathbf{w}_i + \mathbf{v}_j, \quad j = 1, \dots, n.$$

Here, $\mathbf{w}_i$ satisfies $\mathbf{w}_i \geq \mathbf{0}$ and represents the i th endmember signature; h_{ij} satisfies $h_{ij} \geq 0$ and represents the abundance fraction of the i th endmember signature at the j th pixel; $\mathbf{v}_j$ represents noise. The equation above can be expressed in matrix form as

$$A = WH + V \quad (2)$$

by letting $W = [\mathbf{w}_1, \dots, \mathbf{w}_r] \in \mathbb{R}_+^{d \times r}$, $H \in \mathbb{R}_+^{r \times n}$ such that $H(i, j) = h_{ij}$, and $V = [\mathbf{v}_1, \dots, \mathbf{v}_n] \in \mathbb{R}^{d \times n}$. We call W and H the *endmember matrix* and the *abundance matrix* of A , respectively.

Endmember signatures, represented by $\mathbf{w}_1, \dots, \mathbf{w}_r$, are the spectral signatures of certain materials contained in the image scene. The term *endmember* is used for referring to the corresponding material; hence, endmembers are materials contained in the image scene and are major components of the image scene.

We put assumptions on the HSIs. We say that a pixel is *pure* if it contains a single material corresponding to some endmember. We assume that there is at least one pure pixel for every endmember. This is called the *pure pixel assumption*. Accordingly, the HSI matrix A shown in (2) can be rewritten as

$$A = W[I, \bar{H}]\Pi + V \quad (3)$$

for $W \in \mathbb{R}_+^{d \times r}$, $\bar{H} \in \mathbb{R}_+^{r \times (n-r)}$, $V \in \mathbb{R}^{d \times n}$ and a permutation matrix Π of size n . If A is noiseless, i.e., $V = O$, then, we have $A = W[I, \bar{H}]$, which tells us that all columns of W appear in those of A . Furthermore, we assume that the number of endmembers in an HSI is known in advance. We can now formulate the endmember extraction problems as follows.

Problem 1: Given the HSI matrix A shown in (3) and the number r of endmembers, find r columns of A close to those of the endmember matrix W .

The pure pixel assumption is believed to be reasonable. Meanwhile, it is difficult to get the true value for the number of endmembers in an HSI. Thus, our experiments estimated it in the way shown in the previous studies. Problem 1 is known to be equivalent to a problem of computing a nonnegative matrix factorization (NMF) under the separability assumption. Arora et al. [2] studied the separable NMF problems from a theoretical point of view. Their results tell us that we can find W from A exactly if A is noiseless; otherwise, if A contains noise V so that the amount of V is smaller than some level, we can find r columns of A close to those of W . Given A as shown in (3), we say that an algorithm for Problem 1 is *robust to noise* if it can find r such columns of A .

Once Problem 1 is solved, we can compute the abundance matrix H of A from the estimated endmember signatures. Here, let $\mathcal{I}$ be the index set of r columns that was found in the problem. Then, H is the optimal solution of a convex optimization problem in the variable $X \in \mathbb{R}^{r \times n}$, as follows:

$$\begin{aligned} Q(A, \mathcal{I}) : \quad & \min \quad \|A - A(\mathcal{I})X\|_F^2 \\ & \text{s.t.} \quad \mathbf{1}^\top X = \mathbf{1}^\top, X \geq O. \end{aligned}$$

After solving Problem 1, we should evaluate the accuracy of the estimated endmember signatures. In this paper, we use the mean-removed spectral angle (MRSA) of two spectral signatures for this evaluation. For $\mathbf{c} \in \mathbb{R}^d$, we set $\text{ave}(\mathbf{c}) =$

$(\mathbf{1}^\top \mathbf{c}/d) \cdot \mathbf{1} \in \mathbb{R}^d$. For spectral signature vectors $\mathbf{a}, \mathbf{b} \in \mathbb{R}^d$, the MRSA is defined as

$$\text{MRSA}(\mathbf{a}, \mathbf{b}) = \frac{1}{\pi} \arccos \frac{(\mathbf{a} - \text{ave}(\mathbf{a}))^\top (\mathbf{b} - \text{ave}(\mathbf{b}))}{\|\mathbf{a} - \text{ave}(\mathbf{a})\|_2 \|\mathbf{b} - \text{ave}(\mathbf{b})\|_2},$$

which takes any value in the interval from 0 to 1. The small MRSA value for $\mathbf{a}, \mathbf{b}$ means that $\mathbf{a}$ is similar to $\mathbf{b}$.

B. Hottopixx Methods

Hottopixx methods were originally proposed by Bittorf et al. [6] in 2012 for solving separable NMF problems. Several refinements [13], [16], [22] have been developed since then. In this section, we mainly focus on explaining the algorithm by Mizutani [22]. Hottopixx is considered to be a convex relaxation approach to Problem 1. Here, the problem is rewritten as a combinatorial optimization problem as follows: find $X \in \mathbb{R}^{n \times n}$ that minimizes the gap of A and AX under the condition that the number of nonzero rows in X is just r . Hottopixx uses its LP relaxation for this purpose. The algorithmic framework of Hottopixx is described as follows: compute an optimal solution of the LP; then, choose r columns from those of HSI matrix based on information obtained from the optimal solution.

The algorithm of Hottopixx in [22] is as follows. For given A and r in Problem 1, construct an optimization problem,

$$\text{H} : \min \quad \|A - AX\|_1 \quad \text{s.t.} \quad X \in \mathcal{F}$$

where $X \in \mathbb{R}^{n \times n}$ is the variable. The feasible region $\mathcal{F}$ is defined as

$$\begin{aligned} \sum_{i=1}^n X(i, i) &= r, \\ 0 \leq X(i, j) &\leq X(i, i) \leq 1, \quad i, j = 1, \dots, n. \end{aligned}$$

As shown in Section IV-B, H can be reduced to an LP. In this paper, we call H the *Hottopixx model*, although it is different from the original Hottopixx model proposed by Bittorf et al. [6].

Choose r columns of A by using the optimal solution X of H. A simple way of doing so is to find a set $\mathcal{I}$ of r indices corresponding to the r largest elements of $\text{diag}(X)$ and, then, output columns $\mathbf{a}_i$ with $i \in \mathcal{I}$. Theorem 3.1 of [22] ensures that the algorithm is robust to noise. However, the theorem is invalid and does not hold in a case where there are overlaps of pure pixels in HSIs. To cope with this issue, we incorporate a clustering-based postprocessing procedure. Note that the use of postprocessing was first proposed by Gillis [13].

Let us introduce the notation and terminology for describing the postprocessing. Choose a column $\mathbf{a}_i$ of A and then sort the columns $\mathbf{a}_1, \dots, \mathbf{a}_n$ of A by their distance to $\mathbf{a}_i$ in ascending order so that

$$\|\mathbf{a}_i - \mathbf{a}_{u_1}\|_1 \leq \|\mathbf{a}_i - \mathbf{a}_{u_2}\|_1 \leq \dots \leq \|\mathbf{a}_i - \mathbf{a}_{u_{n-1}}\|_1$$

where $\{i, u_1, \dots, u_{n-1}\} = \{1, \dots, n\}$. Then, construct

$$\Omega_i = \{\{i\}, \{i, u_1\}, \{i, u_1, u_2\}, \dots, \{i, u_1, u_2, \dots, u_{n-1}\}\}.$$

We use the term *cluster* to refer to an element $\mathcal{S}$ of $\Omega_1 \cup \dots \cup \Omega_n$. Let $\mathcal{S} \in \Omega_i$ and $\mathbf{p} \in \mathbb{R}_+^n$. We regard $\mathbf{p}$ as a *point list* for $\mathcal{S}$. We define the score of $\mathcal{S}$ according to the point list $\mathbf{p}$ by

$$\text{score}(\mathcal{S}, \mathbf{p}) = \sum_{u \in \mathcal{S}} \mathbf{p}(u).$$

Also, define the diameter of $\mathcal{S}$ in Ω_i by

$$\text{diam}(\mathcal{S}) = \max_{u \in \mathcal{S}} \|\mathbf{a}_i - \mathbf{a}_u\|_1.$$

Then, let $\Psi(\mathbf{p}) = \Psi_1(\mathbf{p}) \cup \dots \cup \Psi_n(\mathbf{p})$ for

$$\Psi_i(\mathbf{p}) = \left\{ \mathcal{S} \in \Omega_i \mid \text{score}(\mathcal{S}, \mathbf{p}) > \frac{r}{r+1} \right\}.$$

Algorithm 1 describes each step of Hottopixx with postprocessing [22].

Algorithm 1 Hottopixx with postprocessing (Algorithm 5.1 of [22])

Input: $A \in \mathbb{R}^{d \times n}$ and a positive integer r .

Output: r columns of A .

- 1: Compute the optimal solution X of H.
- 2: Set $\mathbf{p}_1 = \text{diag}(X)$, $\mathcal{I} = \emptyset$ and $\ell = 1$. Perform the following procedure.
 - 2-1: Find a cluster $\mathcal{S}_\ell$ such that

$$\mathcal{S}_\ell = \arg \min_{\mathcal{S} \in \Psi(\mathbf{p}_\ell)} \text{diam}(\mathcal{S}).$$

- 2-2: Choose one element from $\mathcal{S}_\ell$ and add it to $\mathcal{I}$. Increase ℓ by 1.
- 2-3: If $\ell = r$, return $\mathbf{a}_i$ with $i \in \mathcal{I}$ and terminate; otherwise, construct $\mathbf{p}_\ell$ as

$$\mathbf{p}_\ell(u) = \begin{cases} 0 & \text{if } u \in \mathcal{S}_1 \cup \dots \cup \mathcal{S}_{\ell-1}, \\ \mathbf{p}_1(u) & \text{otherwise,} \end{cases}$$

and go to step 2-1.

Theorem 3.2 of [22] ensures that Algorithm 1 is robust to noise. Even if there are overlaps of pure pixels in the HSI, the theorem remains valid and the algorithm is robust to noise. Algorithm 1 has an advantage over the other algorithms [6], [13], [16] in that it does not require us to give the noise level involved in A as input. In general, the noise-level estimation is not an easy task. We thus use Algorithm 1 for solving Problem 1.

In most cases, there are a number of pixels in the HSIs that are close to pure pixels. Specifically, we will see this feature in the Urban dataset in Section VI-C. Hence, when solving Problem 1, we should incorporate postprocessing into Hottopixx. In fact, our experimental results, shown in Sections VI-B, VI-C, should convince the reader of the need for postprocessing.

C. Related Methods

So far, many methods have been developed for tackling Problem 1. As described in the textbook [15] by Gillis, they can be categorized into two groups: convex methods and

greedy methods. Convex methods have more robustness to noise but have high computational costs, while greedy methods are the opposite. Hottopixx is a convex method.

1) *Convex Methods*: Solving Hottopixx models is costly. For that reason, Gillis and Luce [17] proposed an algorithm, called FGNSR, that solves alternative convex optimization models as follows. For a penalty parameter $\theta \geq 0$ and some $\mathbf{c} \in \mathbb{R}^n$ close to $\mathbf{1}$, consider the following model,

$$\min \frac{1}{2} \|A - AX\|_F^2 + \theta \cdot \mathbf{c}^\top \text{diag}(X) \quad \text{s.t.} \quad X \in \mathcal{G}.$$

Here, $X \in \mathbb{R}^{n \times n}$ is the variable. The feasible set $\mathcal{G}$ is defined by

$$\begin{aligned} X(i, i) &\leq 1, & i &= 1, \dots, n, \\ c_i X(i, j) &\leq c_j X(i, i), & i, j &= 1, \dots, n, \\ X &\geq O \end{aligned}$$

where $c_i = \|A(:, i)\|_1$. Gillis and Luce showed that a projection of a matrix onto $\mathcal{G}$ can be done in $O(n^2 \log n)$. Taking that into account it, they developed a first-order method equipped with Nesterov's acceleration. FGNSR solves the models by using the method and extracts r columns of A by using the optimal solutions.

Self-dictionary sparse regression is common approach to Problem 1. The model can be described as

$$\min \frac{1}{2} \|A - AX\|_F^2 + \theta \cdot \psi(X) \quad \text{s.t.} \quad \mathbf{1}^\top X = \mathbf{1}^\top, \quad X \geq O.$$

The regularization term $\psi(X)$ aims to minimize the number of nonzero rows in X . In [8], [9], [12], it was chosen to be $\psi(X) = \|X\|_{1,q}$, whereas in [24], it was set as a smooth approximation of $\|X\|_{1,\infty}$. Here, we define $\|X\|_{1,q} = \sum_{i=1}^n \|X(i, :)\|_q$. ADMM was used to solve the models in [8], [9], [12], and the Frank-Wolfe method in [24]. Fu and Ma [11] proved that an algorithm based on self-dictionary sparse regression with $\psi(X) = \|X\|_{1,\infty}$ is robust to noise.

2) *Greedy Methods*: Greedy methods choose the columns of the HSI matrix one-by-one based on certain criteria. One of the most popular greedy methods for Problem 1 is SPA. It is said to date back to the work of Araújo et al. [1] in 2001. They used SPA in the context of chemometrics. After that, Gillis and Vavasis [18] gave a theoretical justification for the robustness of SPA to noise. SPA is considered to be same as QR factorization with column pivoting. It recursively chooses the columns of the input matrix. For a column chosen one iteration before, it projects columns onto the orthogonal complement of the chosen column and then chooses one with the maximum L_2 norm. As way of enhancing the robustness of SPA to noise, Gillis and Vavasis [19] developed PSPA that is SPA with preconditioning, and Mizutani [21] ER that is SPA with preprocessing. Ellipsoids with the minimum volume play a key role in the preconditioning and preprocessing. Gillis [14] developed the SNPA, which is similar to that of SPA but it uses a projection onto the convex hull of the origin and the columns chosen in previous iterations. Nascimento and Bioucas-Dias [23] proposed VCA, which is popular in the remote sensing community. It applies an SVD-based dimensionality reduction to the input matrix, and then, runs the same algorithm as SPA except for the criterion for choosing columns.

IV. SOLVING HOTTOPIXX MODELS EFFICIENTLY

A. Algorithm Outline

Solving Hottopixx models H arising from endmember extraction problems is challenging. We show in Section IV-B that H can be reduced into an LP with $O(n^2)$ variables and $O(n^2)$ constraints for HSI matrices $A \in \mathbb{R}^{d \times n}$ with $d \leq n$. The size of the LP increases quadratically with the number n of pixels in HSIs, and n often exceeds 10,000 ($= 100 \times 100$).

To remedy the computational issue, we develop an algorithm for solving H efficiently based on the framework of column generation, which is known as a classical but powerful means of solving large-scale LPs; see, for instance, Chapter 6 of the textbook [4] for details of the column generation methods. Our algorithm is based on the observation that there could be many zero rows in the optimal solution of H. Indeed, the constraints $0 \leq X(i, j) \leq X(i, i)$ and $\sum_{i=1}^n X(i, i) = r$ of H imply that the i th row of X is zero if $X(i, i) = 0$, and many of $X(1, 1), \dots, X(n, n)$ could be zero. We can significantly reduce the size of the H by exploiting the sparsity of the optimal solution.

From here on, we will use the following notation: $\mathcal{N} = \{1, \dots, n\}$; $\ell = |\mathcal{L}|$ and $m = |\mathcal{M}|$ for the subsets $\mathcal{L}, \mathcal{M}$ of $\mathcal{N}$. Below, we consider the subproblems of H in a variable $X \in \mathbb{R}^{\ell \times m}$. Each one has a feasible region $\mathcal{F}(\ell, m)$ defined by

$$\begin{aligned} \sum_{i=1}^{\ell} X(i, i) &= r, \\ 0 \leq X(i, j) &\leq X(i, i) \leq 1, \quad i = 1, \dots, \ell, \quad j = 1, \dots, m. \end{aligned}$$

First, let us consider the subproblem of H,

$$H' : \min \|A - A(\mathcal{L})X\|_1 \quad \text{s.t.} \quad X \in \mathcal{F}(\ell, n).$$

There is an $\mathcal{L} \subset \mathcal{N}$ such that $\text{opt}(H) = \text{opt}(H')$. If many rows in the optimal solution of H are zero, then, ℓ (i.e., the size of $\mathcal{L}$) is much smaller than n . Next, let us consider a smaller subproblem,

$$H'' : \min \|A(\mathcal{L}) - A(\mathcal{L})X\|_1 \quad \text{s.t.} \quad X \in \mathcal{F}(\ell, \ell).$$

It is easy to see that $\text{opt}(H') = \text{opt}(H'')$ if some condition is satisfied. We formally describe it in Theorem 2 (i) of Section IV-C.

Our algorithm solves H'' and checks the conditions as to whether the optimal solution of H'' yields the optimal solution of H; if does not do so, some elements in $\mathcal{N} \setminus \mathcal{L}$ are added to $\mathcal{L}$ and H'' is solved again. We call the algorithm “row and column expansion”, RCE, since it starts from the optimal solution X^* of H'' , and then expands the rows and the columns of X^* step-by-step. In Section IV-C, we present the conditions such that the optimal solution of H'' yields the optimal solution of H' and also that of H, and give a precise description of each step of the RCE algorithm.

B. Subproblems of H

Here, we formally describe the subproblems of H. For given subsets $\mathcal{L}$ and $\mathcal{M}$ of $\mathcal{N}$ satisfying $\mathcal{L} \subset \mathcal{M}$, choose a permutation matrix $\Pi \in \mathbb{R}^{m \times m}$ such that

$$A(\mathcal{M})\Pi = [A(\mathcal{L}), A(\mathcal{M} \setminus \mathcal{L})].$$

Then, construct the subproblem of H,

$$\begin{aligned} H(\mathcal{L}, \mathcal{M}) : \quad &\min \|A(\mathcal{M})\Pi - A(\mathcal{L})X\|_1 \\ &\text{s.t.} \quad X \in \mathcal{F}(\ell, m) \end{aligned}$$

where $X \in \mathbb{R}^{\ell \times m}$ is the variable. The permutation matrix Π is chosen to be the identity matrix when $\mathcal{L} = \mathcal{M}$. Hence, H is equivalent to $H(\mathcal{N}, \mathcal{N})$. We can reduce $H(\mathcal{L}, \mathcal{M})$ to the following LP problem,

$$\begin{aligned} P(\mathcal{L}, \mathcal{M}) : \quad &\min \quad u \\ &\text{s.t.} \quad A(\mathcal{M})\Pi - A(\mathcal{L})X = F - G, \\ &\quad \sum_{i=1}^d F(i, j) + G(i, j) \leq u, \quad j = 1, \dots, m, \\ &\quad F \geq O, \quad G \geq O, \quad X \in \mathcal{F}(\ell, m) \end{aligned}$$

where $(X, F, G, u) \in \mathbb{R}^{\ell \times m} \times \mathbb{R}^{d \times m} \times \mathbb{R}^{d \times m} \times \mathbb{R}$ is the variable. We will show in Theorem 1 that solving $P(\mathcal{L}, \mathcal{M})$ is equivalent to solving $H(\mathcal{L}, \mathcal{M})$. We add slack variables to the inequality constraints and convert $P(\mathcal{L}, \mathcal{M})$ to a standard form LP. Then, the dual problem becomes

$$\begin{aligned} D(\mathcal{L}, \mathcal{M}) : \quad &\max \quad \langle A(\mathcal{M})\Pi, Y \rangle + rv - \mathbf{1}^\top t \\ &\text{s.t.} \quad A(\mathcal{L})^\top Y + [vI, O] - [\text{diag}(t), O] \\ &\quad - Z^\top + [\text{diag}(Z^\top \mathbf{1}), O] \leq O, \\ &\quad -J \cdot \text{diag}(s) \leq Y \leq J \cdot \text{diag}(s), \\ &\quad \mathbf{1}^\top s \leq 1, \\ &\quad s \geq O, \quad t \geq O, \quad Z \geq O \end{aligned}$$

where $(Y, Z, s, t, v) \in \mathbb{R}^{d \times m} \times \mathbb{R}^{m \times \ell} \times \mathbb{R}^m \times \mathbb{R}^\ell \times \mathbb{R}$ is the variable.

It is obvious that there exist feasible solutions to $P(\mathcal{L}, \mathcal{M})$ and $D(\mathcal{L}, \mathcal{M})$. Hence, the duality theorem, which we reviewed in Section II-B, holds and ensures that there are optimal solutions to $P(\mathcal{L}, \mathcal{M})$ and $D(\mathcal{L}, \mathcal{M})$, and $\text{opt}(P(\mathcal{L}, \mathcal{M})) = \text{opt}(D(\mathcal{L}, \mathcal{M}))$ holds.

Theorem 1: For any subsets $\mathcal{L}$ and $\mathcal{M}$ of $\mathcal{N}$ satisfying $r \leq \ell \leq m$, the following hold.

- (i) $\text{opt}(P(\mathcal{L}, \mathcal{M})) = \text{opt}(D(\mathcal{L}, \mathcal{M})) = \text{opt}(H(\mathcal{L}, \mathcal{M}))$.
- (ii) If (X^*, F^*, G^*, u^*) is the optimal solution of $P(\mathcal{L}, \mathcal{M})$, then X^* is the optimal solution of $H(\mathcal{L}, \mathcal{M})$.
- (iii) If X^* is the optimal solution of $H(\mathcal{L}, \mathcal{M})$, then, $(X^*, R^+, R^-, \|R\|_1)$ for $R = A\Pi - A(\mathcal{L})X^*$ is the optimal solution of $P(\mathcal{L}, \mathcal{M})$.

The proof is in Appendix B of the supplemental material.

C. Algorithm Details

The optimal solution of H, which is equivalent to $H(\mathcal{N}, \mathcal{N})$, can be obtained by solving $H(\mathcal{L}, \mathcal{L})$ if certain conditions are satisfied. Below, we show conditions such that

$\text{opt}(\mathcal{H}(\mathcal{L}, \mathcal{L})) = \text{opt}(\mathcal{H}(\mathcal{L}, \mathcal{N})) = \text{opt}(\mathcal{H}(\mathcal{N}, \mathcal{N}))$ and then describe each step of RCE at the end of this section. In order to describe the conditions, we construct auxiliary problems from the optimal solution $X^* \in \mathbb{R}^{\ell \times \ell}$ of $\mathcal{H}(\mathcal{L}, \mathcal{L})$, as follows:

$$\begin{aligned} \mathcal{R}_j(\mathcal{L}, X^*) : \quad & \min \quad \|\mathbf{a}_j - A(\mathcal{L})\gamma_j\|_1 \\ \text{s.t.} \quad & \mathbf{0} \leq \gamma_j \leq \text{diag}(X^*) \end{aligned}$$

for $j \in \mathcal{N} \setminus \mathcal{L}$ where $\gamma_j \in \mathbb{R}^\ell$ is the variable.

Theorem 2: Let $\mathcal{L} \subset \mathcal{N}$. Let $\alpha^* = (X^*, F^*, G^*, u^*)$ be the optimal solution of $\mathcal{P}(\mathcal{L}, \mathcal{L})$ and $\beta^* = (Y^*, Z^*, \mathbf{s}^*, \mathbf{t}^*, v^*)$ be the optimal solution of $\mathcal{D}(\mathcal{L}, \mathcal{L})$. Consider the following conditions.

(C1) X^* of α^* satisfies

$$\text{opt}(\mathcal{R}_j(\mathcal{L}, X^*)) \leq \text{opt}(\mathcal{P}(\mathcal{L}, \mathcal{L}))$$

for every $j \in \mathcal{N} \setminus \mathcal{L}$.

(C2) Y^* and v^* of β^* satisfy

$$v^* + \mathbf{1}^\top ((Y^*)^\top \mathbf{a}_j)^+ \leq 0$$

for every $j \in \mathcal{N} \setminus \mathcal{L}$.

The following hold.

- (i) Assume that condition (C1) holds. Let $\Gamma^* = [\gamma_j^* : j \in \mathcal{N} \setminus \mathcal{L}]$ for the optimal solution γ_j^* of $\mathcal{R}_j(\mathcal{L}, X^*)$. Then,
 - $\text{opt}(\mathcal{H}(\mathcal{L}, \mathcal{L})) = \text{opt}(\mathcal{H}(\mathcal{L}, \mathcal{N}))$, and
 - the matrix $[X^*, \Gamma^*]$ is the optimal solution of $\mathcal{H}(\mathcal{L}, \mathcal{N})$.
- (ii) Assume that conditions (C1) and (C2) hold. Let $\Gamma^* = [\gamma_j^* : j \in \mathcal{N} \setminus \mathcal{L}]$ for the optimal solution γ_j^* of $\mathcal{R}_j(\mathcal{L}, X^*)$, and let Π be a permutation matrix of size n such that $A\Pi = [A(\mathcal{L}), A(\mathcal{N} \setminus \mathcal{L})]$. Then,
 - $\text{opt}(\mathcal{H}(\mathcal{L}, \mathcal{L})) = \text{opt}(\mathcal{H}(\mathcal{N}, \mathcal{N}))$, and
 - the matrix

$$\Pi \begin{bmatrix} X^* & \Gamma^* \\ O & O \end{bmatrix} \Pi^\top$$

is the optimal solution of $\mathcal{H}(\mathcal{N}, \mathcal{N})$.

The proof is in Appendix A. Note that $((Y^*)^\top \mathbf{a}_j)^+$ in condition (C2) is equivalent to $\mathbf{b}_j^+$ for $\mathbf{b}_j = (Y^*)^\top \mathbf{a}_j$. We design an efficient algorithm for solving $\mathcal{H}$ on the basis of Theorem 2. One might be concerned that condition (C2) rarely holds; i.e., it never holds as long as v^* takes a positive number. However, this concern is alleviated by Lemma 1, which ensures that v^* always takes a nonpositive number.

Lemma 1: Let $(Y^*, Z^*, \mathbf{s}^*, \mathbf{t}^*, v^*)$ be the optimal solution of $\mathcal{D}(\mathcal{L}, \mathcal{L})$. If $\mathcal{L}$ is chosen to satisfy $\ell \geq r$, then $v^* \leq 0$.

The proof is in Appendix C of the supplemental material. Algorithm 2 describes each step of RCE.

V. IMPLEMENTATION OF HOTTOPIXX

We propose EEHT that is the abbreviation for an efficient and effective implementation of Hottopixx [22]. This section explains the details of EEHT and then summarizes the overall algorithm.

Algorithm 2 RCE : Row and column expansion algorithm for computing the optimal solution of $\mathcal{H}$

Input: $A \in \mathbb{R}^{d \times n}$ and a positive integer r .

Output: $X \in \mathbb{R}^{n \times n}$.

- 1: Choose $\mathcal{L} \subset \mathcal{N}$ satisfying $r \leq \ell$.
- 2: Repeat the following procedure until the stopping condition at step 2-3 is satisfied.
 - 2-1: Compute the optimal solution (X^*, F^*, G^*, u^*) of $\mathcal{P}(\mathcal{L}, \mathcal{L})$, and the optimal solution $(Y^*, Z^*, \mathbf{s}^*, \mathbf{t}^*, v^*)$ of $\mathcal{D}(\mathcal{L}, \mathcal{L})$.
 - 2-2: Compute the optimal solution γ_j of $\mathcal{R}_j(\mathcal{L}, X^*)$ for every $j \in \mathcal{N} \setminus \mathcal{L}$.
 - 2-3: If

$$\text{opt}(\mathcal{R}_j(\mathcal{L}, X^*)) \leq \text{opt}(\mathcal{P}(\mathcal{L}, \mathcal{L}))$$

for every $j \in \mathcal{N} \setminus \mathcal{L}$, go to step 3; otherwise, update $\mathcal{L}$ by

$$\mathcal{L} \cup \{j \in \mathcal{N} \setminus \mathcal{L} \mid \text{opt}(\mathcal{R}_j(\mathcal{L}, X^*)) > \text{opt}(\mathcal{P}(\mathcal{L}, \mathcal{L}))\},$$

and go back to step 2-1.

3: If

$$v^* + \mathbf{1}^\top ((Y^*)^\top \mathbf{a}_j)^+ \leq 0$$

for every $j \in \mathcal{N} \setminus \mathcal{L}$, then construct the matrix,

$$X = \Pi \begin{bmatrix} X^* & \Gamma^* \\ O & O \end{bmatrix} \Pi^\top \in \mathbb{R}^{n \times n}$$

for $\Gamma^* = [\gamma_j^* : j \in \mathcal{N} \setminus \mathcal{L}]$ and a permutation matrix Π such that $A\Pi = [A(\mathcal{L}), A(\mathcal{N} \setminus \mathcal{L})]$, and return the result; otherwise, update $\mathcal{L}$ by

$$\mathcal{L} \cup \{j \in \mathcal{N} \setminus \mathcal{L} \mid v^* + \mathbf{1}^\top ((Y^*)^\top \mathbf{a}_j)^+ > 0\},$$

and go back to step 2.

a) Preprocessing: Before running RCE, our implementation applies a dimensionality reduction technique to the input matrix. For a given $A \in \mathbb{R}^{d \times n}$ and positive integer r as input, compute the top- r truncated SVD,

$$A_r = U_r \Sigma_r V_r^\top,$$

of A . Here, $\Sigma_r \in \mathbb{R}^{r \times r}$ is diagonal so that the top- r singular values $\sigma_1, \dots, \sigma_r$ of A are in the diagonal positions and the columns of $U_r \in \mathbb{R}^{d \times r}$ (resp. $V_r \in \mathbb{R}^{n \times r}$) are the left-singular (resp. right-singular) vectors of A corresponding to $\sigma_1, \dots, \sigma_r$. Then, construct a size-reduced matrix $A' = \Sigma_r V_r^\top \in \mathbb{R}^{r \times n}$ of A and run RCE on input (A', r) . It should be noted that, if A is an HSI matrix of size $d \times n$ with r endmembers, as shown in (3), then its size-reduced matrix A' is also an HSI matrix, written as $A' = W'[I, \bar{H}]\Pi + V'$ for $W' \in \mathbb{R}^{r \times r}$ and $V' \in \mathbb{R}^{r \times n}$.

b) Step 2-2 of Algorithm 1: In theory, we are allowed to choose any one of elements in a cluster $\mathcal{S}_\ell$ at step 2-2 of Algorithm 1. Indeed, Theorem 3.2 of [22] ensures that Algorithm 1 is robust to noise, and this result holds no matter how we choose an element from $\mathcal{S}_\ell$ at step 2-2. Yet, in practice,

the method of choice affects its robustness to noise. For that reason, our implementation employs two methods of choice in step 2-2 of Algorithm 1. The first one, called the *max-point choice*, chooses an element $u \in \mathcal{S}_\ell$ with the highest score according to the point list $\mathbf{p}_\ell$:

$$u = \arg \max_{u \in \mathcal{S}_\ell} \mathbf{p}_\ell(u). \quad (4)$$

The max-point choice is not new; it was used in [13], [16], [22] in their postprocessing for Hottopixx. The second one, called the *cluster centroid choice*, defines the centroid $\mathbf{c}_\ell$ of a cluster $\mathcal{S}_\ell$ by

$$\mathbf{c}_\ell = \frac{1}{|\mathcal{S}_\ell|} \sum_{u \in \mathcal{S}_\ell} \mathbf{a}_u$$

and then chooses an element $u \in \mathcal{S}_\ell$ such that $\mathbf{a}_u$ is closest to $\mathbf{c}_\ell$ in terms of MRSA:

$$u = \arg \min_{u \in \mathcal{S}_\ell} \text{MRSA}(\mathbf{c}_\ell, \mathbf{a}_u). \quad (5)$$

The cluster centroid choice is our proposal.

c) *Step 1 of RCE*: The initial choice of $\mathcal{L}$ at step 1 of RCE affects scaling up of the size of $\mathcal{L}$ during the iterations. If the initial choice of $\mathcal{L}$ covers many column indices of A corresponding to pure pixels, we can expect an increase in the size of $\mathcal{L}$ to be inhibited. Our implementation uses SPA [1], [18] for this purpose. SPA is fast and the accuracy of its output is reasonable.

Algorithm 3 is our procedure for constructing an initial set $\mathcal{L}$. Step 1 uses SPA to find columns $\mathbf{a}_{\ell_1}, \dots, \mathbf{a}_{\ell_r}$ of A that are expected to be close to pure pixels. Step 2 finds the top- λ nearest neighbors to each $\mathbf{a}_{\ell_1}, \dots, \mathbf{a}_{\ell_r}$ and constructs a set $\mathcal{L}_{\text{NN}}$ of their indices. Step 3 chooses μ extra elements from the complement of $\mathcal{L}_{\text{NN}}$ and constructs a set $\mathcal{L}_{\text{EX}}$ of the extras. Step 4 takes the union of $\mathcal{L}_{\text{NN}}$ and $\mathcal{L}_{\text{EX}}$ and returns it as $\mathcal{L}$.

Algorithm 4 is the overall procedure for implementing Hottopixx.

Algorithm 3 Constructing an initial index set $\mathcal{L}$

Input: $A \in \mathbb{R}^{d \times n}$ and positive integers r, λ and μ .

Output: $\mathcal{L} \subset \mathcal{N}$.

- 1: Perform SPA on input (A, r) . Let $\mathbf{a}_{\ell_1}, \dots, \mathbf{a}_{\ell_r}$ be the columns of A output by it.
- 2: Set $\mathcal{L}_{\text{NN}} = \emptyset$. Perform the following procedure for $i = 1, \dots, r$.
 - 2-1: Sort columns $\mathbf{a}_1, \dots, \mathbf{a}_n$ of A by their distance to $\mathbf{a}_{\ell_i}$ in ascending order so that

$$\|\mathbf{a}_{\ell_i} - \mathbf{a}_{u_1}\|_2 \leq \dots \leq \|\mathbf{a}_{\ell_i} - \mathbf{a}_{u_n}\|_2.$$

- 2-2: Update $\mathcal{L}_{\text{NN}}$ by $\mathcal{L}_{\text{NN}} \cup \{u_1, \dots, u_\lambda\}$.
 - 3: Choose μ arbitrary elements from $\mathcal{N} \setminus \mathcal{L}_{\text{NN}}$, and construct a set $\mathcal{L}_{\text{EX}}$ of them.
 - 4: Set $\mathcal{L} = \mathcal{L}_{\text{NN}} \cup \mathcal{L}_{\text{EX}}$ and return $\mathcal{L}$.
-

VI. EXPERIMENTS

We conducted experiments to assess the efficiency and effectiveness of EEHT on Problem 1. For the experiments,

Algorithm 4 EEHT: Efficient and effective implementation of Hottopixx

Input: $A \in \mathbb{R}^{d \times n}$ and positive integers r, λ , and μ .

Output: r columns of A .

- 1: Compute the top- r truncated SVD $A_r = U_r \Sigma_r V_r^\top$ and construct a size-reduced matrix $A' = \Sigma_r V_r^\top$.
 - 2: Run RCE on (A', r) and obtain the output X , where step 1 runs Algorithm 3 on (A', r, λ, μ) .
 - 3: Construct an index set $\mathcal{I}$ by performing one of the following methods A-C and return $\mathbf{a}_i$ with $i \in \mathcal{I}$.
 - A. Find a set $\mathcal{I}$ of r indices corresponding to the r largest elements of $\text{diag}(X)$, and return $\mathbf{a}_i$ with $i \in \mathcal{I}$.
 - B. Run step 2 of Algorithm 1 where step 2-2 adopts the max-point choice of (4).
 - C. Run step 2 of Algorithm 1 where step 2-2 adopts the cluster centroid choice of (5).
-

we coded EEHT in MATLAB. Here, step 2 of EEHT runs RCE whose step 2-1 needs to solve LP problems $P(\mathcal{L}, \mathcal{L})$ and $D(\mathcal{L}, \mathcal{L})$. Here, we employed CPLEX. The MATLAB function `cplexlp` is available in the CPLEX package and it enables us to run CPLEX in the MATLAB environment. Applying the `cplexlp` function to an LP returns the optimal solutions for the LP and also for its dual. Our code used `cplexlp` for $D(\mathcal{L}, \mathcal{L})$ in order to obtain both optimal solutions of $P(\mathcal{L}, \mathcal{L})$ and $D(\mathcal{L}, \mathcal{L})$. In what follows, EEHT with the choice of method A (resp. B, C) at step 3 is referred to as EEHT-A (resp. -B, -C).

We used MRSA to assess the endmember extraction performances. Let $\mathbf{w}_1^{\text{ref}}, \dots, \mathbf{w}_r^{\text{ref}}$ be reference signatures and $\mathbf{a}_{i_1}, \dots, \mathbf{a}_{i_r}$ be r columns of A returned by the algorithm. We computed a permutation σ of size r such that

$$\min_{\sigma} \sum_{j=1}^r \text{MRSA}(\mathbf{a}_{i_j}, \mathbf{w}_{\sigma(j)}^{\text{ref}}).$$

Note that the minimum is taken over all permutations σ of size r . We next evaluated the MRSA values between estimated endmember signatures and reference ones

$$\text{MRSA}(\mathbf{a}_{i_j}, \mathbf{w}_{\sigma(j)}^{\text{ref}}) \text{ for } j = 1, \dots, r \quad (6)$$

and their average,

$$\frac{1}{r} \sum_{j=1}^r \text{MRSA}(\mathbf{a}_{i_j}, \mathbf{w}_{\sigma(j)}^{\text{ref}}). \quad (7)$$

The experiments were done in MATLAB on dual Intel Xeon Gold 6336Y processors with 256 GB of memory.

A. Computational Efficiency

RCE is a key component of EEHT in solving Hottopixx models efficiently. We thus examined the computational time of RCE on synthetic datasets. The datasets contained HSI matrices $A = W[I, \bar{H}] + V$ with r endmembers that were generated by the following procedure:

- *As for W* : Draw the entries of W from a uniform distribution on the interval $[0, 1]$, and replace them with

their absolute values. Finally, normalize the columns of W to have unit L_1 norm.

- *As for $\bar{H}$:* Draw the columns of $\bar{H}$ from a Dirichlet distribution with r parameters that are uniformly distributed in the interval $[0, 1]$.
- *As for V :* Draw the entries of V from a standard normal distribution, and normalize V such that $\nu = \|V\|_1$ for a noise intensity level ν specified in advance.

We chose ten equally spaced points between 0 and 1 and set them as noise intensity levels ν . One dataset contained ten HSI matrices $A = W[I, \bar{H}] + V$ of size $50 \times n$ with $r = 10$ satisfying $\nu = \|V\|_1$ for each ν . We set n from 500 to 2500 in 500 increments and, thus, constructed five datasets in total.

We performed four algorithms, RCE-SR, RCE-DIR, CPLEX-SR, and CPLEX-DIR. Here, RCE-DIR and CPLEX-DIR solved the Hottopixx models H for the original matrices A , while RCE-SR and CPLEX-SR solved theirs for size-reduced ones A' obtained from the top- r truncated SVD of A . The details are as follows.

- **RCE-SR** constructed a size-reduced matrix A' of A and then applied RCE to A' , where step 1 runs Algorithm 3 with $(\lambda, \mu) = (10, 100)$.
- **RCE-DIR** directly applied RCE to A , where step 1 runs Algorithm 3 with $(\lambda, \mu) = (10, 100)$.
- **CPLEX-SR** constructed a size-reduced matrix A' of A and used `cplexlp` to solve the LP problem $P(\mathcal{N}, \mathcal{N})$ for A' .
- **CPLEX-DIR** directly used `cplexlp` to solve the LP problem $P(\mathcal{N}, \mathcal{N})$ for A .

We compared RCE-SR and -DIR with CPLEX-SR and -DIR, since the implementation of Hottopixx methods in [16], [22] uses CPLEX and solves Hottopixx models directly. Figure 1 summarizes the average elapsed time for each dataset. We can see from the figure that RCE is faster than CPLEX for $n \geq 1000$. In particular, RCE-SR is significantly faster than the other algorithms and its average elapsed time increases more slowly. RCE-DIR is faster than CPLEX-DIR, but its average elapsed time increases more rapidly than RCE-SR. The results suggest that RCE-SR should be used for the implementation of Hottopixx when dealing with large matrices.

B. Endmember Extraction Performance

Next, we examined the endmember extraction performance of EEHT on semi-real HSI datasets. The datasets were constructed from the Jasper Ridge and Samson HSI datasets. We extracted the subimages in the manner described in Section V of [26]:

- Jasper Ridge consists of 100×100 pixels with 198 bands and contains 4 endmembers: Tree, Soil, Water, Road, and
- Samson consists of 95×95 pixels with 156 bands and contains 3 endmembers: Soil, Tree, Water.

The top of Figure 5 in the supplemental material displays the RGB images of them.

The endmember signatures in the regions were already identified in [26]. Using them as reference signatures, we constructed W, H and V as follows. Let A^{real} be the HSI

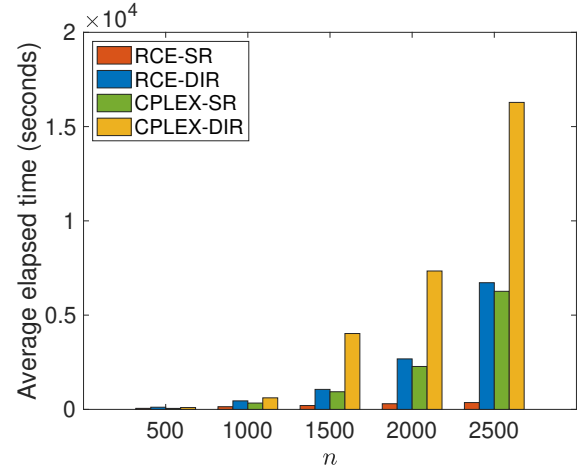


Fig. 1. Average elapsed times of algorithms for five datasets.

matrix for a dataset, and $w_1^{\text{ref}}, \dots, w_r^{\text{ref}}$ be the reference signature vectors. Perform the following procedure:

- 1: Normalize all columns of A^{real} to have unit L_1 norm.
- 2: For $i = 1, \dots, r$, find

$$j_i = \arg \min_{j=1, \dots, n} \text{MRSA}(a_j^{\text{real}}, w_i^{\text{ref}}),$$

and then construct $\mathcal{J} = \{j_1, \dots, j_r\}$.

- 3: Set $W = A^{\text{real}}(:, \mathcal{J})$.
- 4: Compute the optimal solution X of problem $Q(A^{\text{real}}, \mathcal{J})$. Set $X(:, \mathcal{J}) = I$ and then $H = X$.
- 5: Set $V = A^{\text{real}} - WH$.

We constructed two datasets using the W, H and V obtained above: dataset 1 from Jasper Ridge and dataset 2 from Samson. Each dataset contained 20 HSI matrices that were generated as follows: choose 20 equally spaced points between 0 and 1 as noise intensity levels ν ; and construct an HSI matrix $A = WH + (\nu/\|V\|_1) \cdot V$ for each ν . The HSI matrices A were of size 198×10000 with 4 endmembers for dataset 1 and of size 156×9025 with 3 endmembers for dataset 2. Note that $A = A^{\text{real}}$ holds if $\nu = \|V\|_1$, where $\|V\|_1 \approx 0.61$ for dataset 1 and $\|V\|_1 \approx 0.15$ for dataset 2.

We compared EEHT-A, -B, -C, with six existing algorithms: SPA, PSPA, ER, SNPA, VCA and FGNSR. We developed MATLAB codes for SPA, PSPA and ER and used publicly available codes for SNPA, VCA and FGNSR, which were developed by the authors in [14], [17], [23]. We set the input parameters λ and μ of EEHT as $(\lambda, \mu) = (10, 100)$. We used the average MRSA given in (7) as the performance metric.

Figure 2 plots the average MRSA values of the algorithms for datasets 1 and 2. We can see from the figure that, in most cases, EEHT-C had better average MRSA values than the other algorithms had. It was superior to EEHT-A and -B for 15 (16) out of 20 intensity levels on dataset 1 (2). As well, it was superior to the six existing algorithms for 17 (15) out of 20 intensity levels on dataset 1 (2).

In addition, we examined the average elapsed time of the algorithms on 20 HSI matrices in each dataset. The average elapsed time of SPA, PSPA, ER, SNPA and VCA was less than a second on both two datasets. EEHT-A, -B and -C took

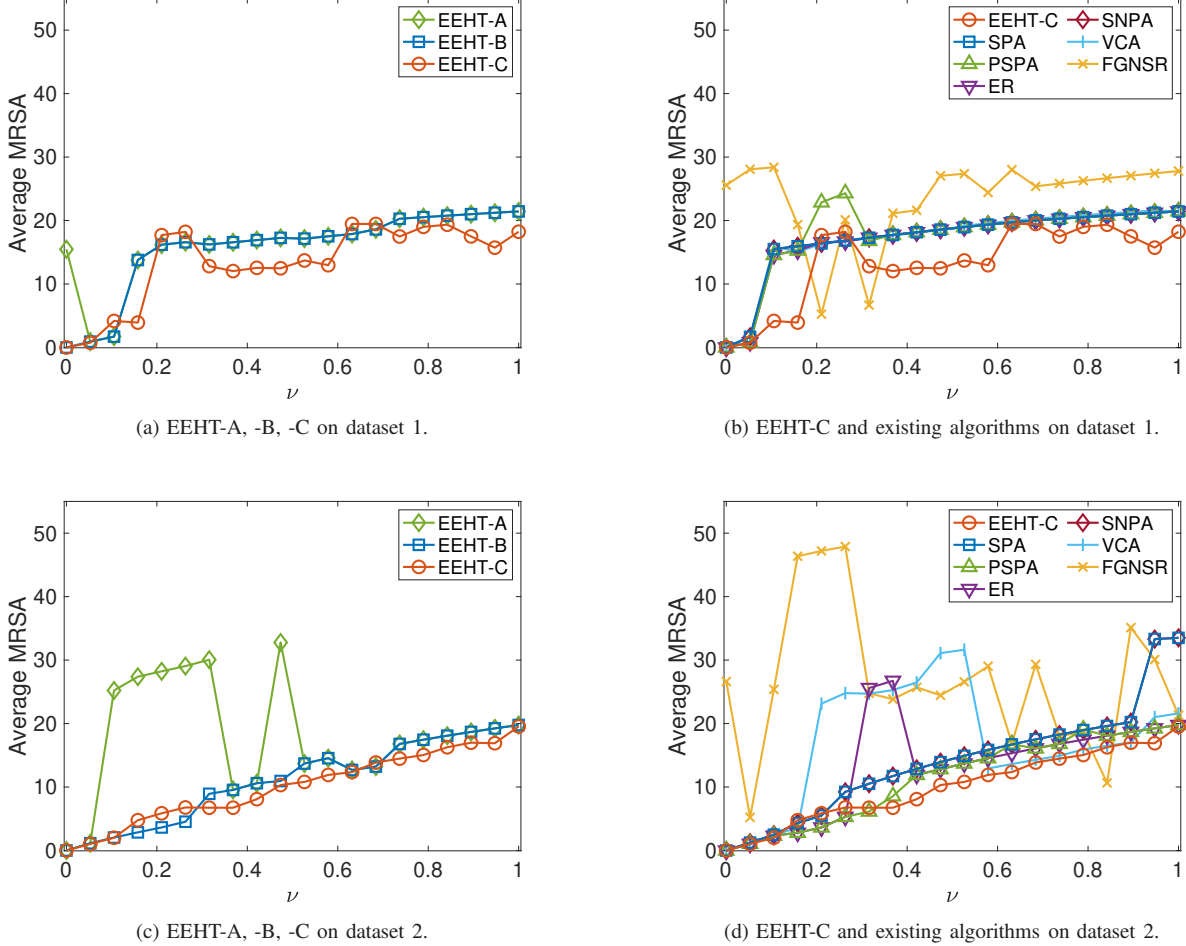


Fig. 2. Average MRSA value ($\times 10^2$) of EEHT-A, -B, -C and existing algorithms on dataset 1 (top) and 2 (bottom).

about 30 minutes, and FGNSR about 17 hours. EEHT was significantly faster than FGNSR. Here, recall that EEHT and FGNSR are categorized as convex methods, while the others greedy methods.

Remark 1: Let us go back to the results of EEHT-A, -B, and -C on dataset 1 (top-left of Figure 2). One may wonder why EEHT-A took positive MRSA values at $\nu = 0$. Remark 4.1 of [22] predicts that this can happen if overlapping columns exist in the HSI matrices. Indeed, EEHT-A took a zero MRSA value for every endmember at $\nu = 0$ after eliminating the duplicate columns.

C. Hyperspectral Unmixing of Urban HSI

Finally, we conducted an experimental study on hyperspectral unmixing of the Urban HSI dataset. The image was taken over Copperas Cove, Texas, USA by the HYDICE sensor. It consists of 307×307 pixels with 210 bands from 400 nm to 2500 nm. Following the procedure in [25], [26], we removed dirty bands from the image. The resulting image had 162 clean bands. The experimental studies in [25], [26] showed that there are mainly 4-6 endmembers in the image. Following the settings used in [17], our experiments set the number of endmembers to 6: Asphalt, Grass, Tree, Roof 1, Roof 2 and

Soil. These 6 endmember signatures were identified in [25]. We used them as reference signatures. The bottom of Figure 5 in the supplemental material displays the RGB image of Urban.

TABLE I
NUMBER OF PIXELS UNDER AN MRSA VALUE OF 0.05 RELATIVE TO THE REFERENCE SIGNATURES OF ENDMEMBERS FOR URBAN.

Asphalt	Grass	Tree	Roof 1	Roof 2	Soil
939	16,435	14,122	1,493	70	1,491

We did not achieve satisfactory results for Urban by applying the algorithms that were tested on the semi-real HSI datasets described in Section VI-B. The estimated endmember signatures were not sufficiently close to the reference ones. Consequently, we conducted data-specific preprocessing aimed at enhancing the endmember extraction performance of them. The preprocessing was based on a reasonable assumption that there are a number of pixels close to each pure pixel in an HSI. Indeed, it holds in the case of the Urban dataset: see Table I that summarizes the number of pixels under an MRSA value of 0.05 relative to the reference signatures of endmembers. Here, we shall say that pixels are *isolated* if there are almost no pixels close to them. Under the assumption, even if we

remove isolated pixels, there would be little effect on the number of pure pixels. It is possible that some of the isolated pixels contain large amounts of noise, which would worsen the endmember extraction performance of the algorithms.

To find isolated pixels, we introduced the *neighborhood density* of each pixel, defined by

$$\rho(i; \phi) = \frac{1}{n} |\text{NH}(i; \phi)| \in [0, 1]$$

for the neighborhood of the i th pixel $\mathbf{a}_i$ under an MRSA value of ϕ ,

$$\text{NH}(i; \phi) = \{j \in \mathcal{N} \mid \text{MRSA}(\mathbf{a}_i, \mathbf{a}_j) \leq \phi\},$$

and the total number n of pixels in an HSI. For Urban, we examined the neighborhood density ρ while varying ϕ . Figure 3 displays a histogram of ρ with a bin size of 0.01 when $\phi = 0.4$. We can see from the figure that there is a small peak in the low-density range from $\rho = 0.08$ to 0.09.

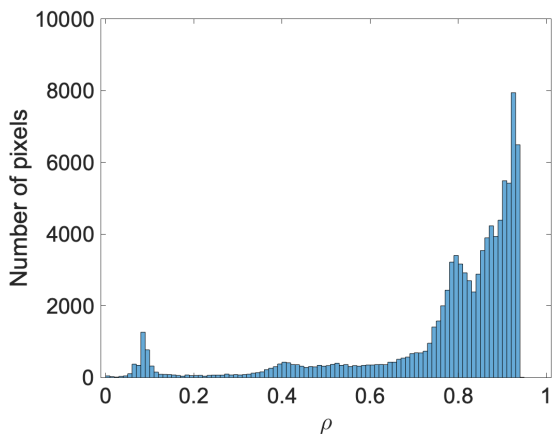


Fig. 3. Histogram of neighborhood density ρ versus the number of pixels for Urban with $\phi = 0.4$ (bin size 0.01).

In the experiments, we set $\phi = 0.4$ and removed all pixels $\mathbf{a}_i$ satisfying $\rho(i; \phi) \leq \omega$ for a truncation parameter ω specified in advance. We ran the same algorithms as in Section VI-B for Urban while varying ω from 0 to 0.15. We set the input parameters λ and μ of EEHT as $(\lambda, \mu) = (50, 300)$. We did not obtain any results from FGNSR, because a forced termination of MATLAB occurred due to out of memory.

First, let us examine the results for Urban with $\phi = 0.4$ and $\omega = 0.1$. Under the setting of ϕ and ω , there were 2955 pixels $\mathbf{a}_i$ (3% for total number of pixels) satisfying $\rho(i; \phi) \leq \omega$. We removed all of those pixels. Table II summarizes the MRSA values for each endmember given in (6) and their averages given in (7). We can see from the table that the MRSA values of EEHT-C are smaller than those of the other algorithms, except in the case of Roof 2. It suggests that we should use EEHT-C rather than EEHT-A, -B for Urban. As for computational time, EEHT-A, -B, -C took about 10 hours, while the other algorithms finished within a few seconds.

To assess the effectiveness of EEHT-C further, we computed abundance maps by using the estimated endmember signatures as well as the ground truth by using the reference signatures. Figure 6 in the supplemental material displays them. We can

see that the abundance maps obtained by EEHT-C are similar to those obtained by the reference signatures except Roof 2.

TABLE II
MRSA VALUES ($\times 10^2$) OF ALGORITHMS FOR URBAN WITH $\phi = 0.4$ AND $\omega = 0.1$. THE BEST SCORES ARE HIGHLIGHTED IN BOLD.

	EEHT			SPA	PSPA	ER	VCA	SNPA
	-A	-B	-C					
Asphalt	11.2	27.4	9.2	20.5	20.5	20.5	16.2	12.7
Grass	15.7	15.7	5.7	35.4	6.4	18.4	41.0	27.0
Tree	44.2	5.8	3.4	4.7	4.7	4.7	15.3	4.7
Roof 1	12.5	24.8	7.1	12.5	12.5	12.5	13.7	12.5
Roof 2	17.1	57.5	18.9	17.1	17.1	29.2	50.8	44.6
Soil	10.4	10.4	3.2	16.7	16.7	16.7	31.1	16.7
Average	18.5	23.6	7.9	17.8	13.0	17.0	28.0	19.7

Next, let us examine the results for Urban with $\phi = 0.4$ and varying ω from 0 to 0.15 in 0.025 increments. Figure 4 plots the average MRSA given in (7). We can see from the figure that EEHT-C is better than the other algorithms in average MRSA except $\omega = 0.15$. The MRSA values of EEHT-C take a minimum at $\omega = 0.1$. If ω is set to be a large value, some of the pixels $\mathbf{a}_i$ satisfying $\rho(i; \phi) \leq \omega$ are no longer isolated and may be close to pure pixels. Removing such pixels can affect the number of pure pixels in an HSI. We set $\phi = 0.4$ and $\omega = 0.125$ and removed all pixels $\mathbf{a}_i$ satisfying $\rho(i; \phi) \leq \omega$. Then, no pixels existed with an MRSA value of 0.05 relative to the reference signature of Roof 1. Accordingly, when $\omega \geq 0.125$, it was impossible for EEHT-C to identify pixels containing the spectral signature of Roof 1 with high purity. For that reason, the average MRSA values of EEHT-C increased. We can see from the figure that our data-specific preprocessing was effective for EEHT-C and also the five existing algorithms and it could improve their average MRSA values with an appropriate choice of ω .

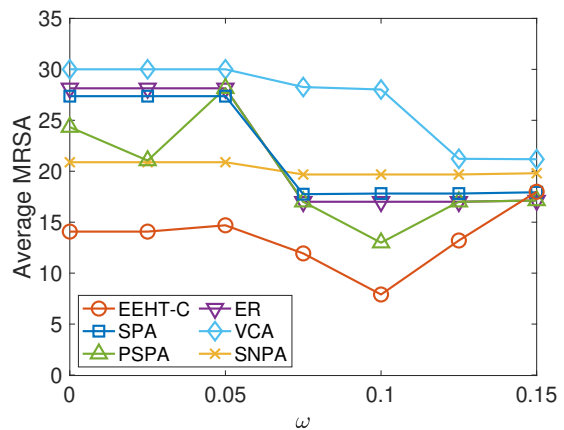


Fig. 4. Average MRSA values ($\times 10^2$) of algorithms for Urban with $\phi = 0.4$ and varying ω from 0 to 0.15 in 0.025 increments.

Our data-specific preprocessing requires us to choose two parameters ϕ, ω . In particular, as shown in Figure 4, the endmember extraction performance of the algorithms depends on the choice of ω . When drawing the histogram of density ρ versus the number of pixels while varying ϕ from 0.45 to 0.6 in 0.05 increments, as is the case of $\phi = 0.4$, a small peak

appeared in a low-density range. Hence, on the basis of the experimental results for $\phi = 0.4$, we can describe the criteria on the choice of ω in the cases of $\phi = 0.45$ to 0.6 as follows.

- (1) When drawing the histogram of density ρ , the peak in a low-density range is cut off by ω .
- (2) The value of ω is small to prevent pixels close to pure pixels from being removed.

We chose ω by following criteria (1) and (2) and ran the algorithms on Urban. Table III summarizes the average MRSA values. We see that their average MRSA values are nearly unchanged for ϕ from 0.4 to 0.6 . These results imply that criteria (1) and (2) on the choice of ω can yield consistent results on the endmember extraction performance of algorithms for Urban if ϕ belongs to some range.

TABLE III

AVERAGE MRSA VALUES ($\times 10^2$) OF ALGORITHMS FOR URBAN WHILE VARYING ϕ AND USING CRITERIA (1) AND (2) ON THE CHOICE OF ω . THE BEST SCORES ARE HIGHLIGHTED IN BOLD.

ϕ	ω	EEHT-C	SPA	PSPA	ER	VCA	SNPA
0.4	0.1	7.9	17.8	13.0	17.0	28.0	19.7
0.45	0.15	8.4	17.8	17.0	17.0	28.3	19.7
0.5	0.3	9.3	17.8	17.0	17.0	28.3	19.7
0.55	0.45	8.2	17.8	17.0	17.0	28.0	19.7
0.6	0.6	8.2	17.8	17.0	17.0	28.3	19.7

VII. CONCLUDING REMARKS

This study was motivated by the question: although the theoretical results shown in [13], [16], [22] imply that Hottopixx methods are effective for endmember extraction problems, is this really true? Hottopixx methods require us to solve costly LP problems, called Hottopixx models. Thus, these methods are challenging to perform in practice. To answer the question, we developed an efficient and effective implementation of Hottopixx, EEHT, and found numerical evidence that EEHT can estimate the endmember signatures of real HSIs with higher accuracy than the existing methods can.

We close this paper with remarks on directions for future research. As we saw in Section VI-C, EEHT can be applied to the endmember extraction of HSIs with around one hundred thousand pixels, but its computational time is much longer than that of greedy methods such as SPA and its refinements. In order to reduce it, we need to focus on the computational cost of RCE, which is computing the optimal solutions of problems $P(\mathcal{L}, \mathcal{L})$ and $D(\mathcal{L}, \mathcal{L})$ in step 2-1, since the problem size could be large. Augmented Lagrangian methods are often used for solving large-scale LPs; see [7], for instance. These methods may be of help in reducing the computational time of EEHT.

REFERENCES

- [1] U. M. C. Araújo, B. T. C. Saldanha, R. K. H. Galvão, T. Yoneyama, H. C. Chame, and V. Visani. The successive projections algorithm for variable selection in spectroscopic multicomponent analysis. *Chemometrics and Intelligent Laboratory Systems*, 57(2):65–73, 2001.
- [2] S. Arora, R. Ge, R. Kannan, and A. Moitra. Computing a nonnegative matrix factorization – Provably. In *Proceedings of the 44th symposium on Theory of Computing (STOC)*, pages 145–162, 2012.
- [3] A. Ben-Tal and A. Nemirovski. *Lectures on Modern Convex Optimization*. SIAM, 2001.
- [4] D. Bertsimas and J. N. Tsitsiklis. *Introduction to Linear Optimization*. Athena Scientific, 1997.
- [5] J. M. Bioucas-Dias, A. Plaza, N. Dobigeon, M. Parente, Q. Du, P. Gader, and J. Chanussot. 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.
- [6] V. Bittorf, B. Recht, C. Re, and J. A. Tropp. Factoring nonnegative matrices with linear programs. In *Proceedings of Advances in Neural Information Processing Systems 25 (NIPS)*, pages 1223–1231, 2012.
- [7] S. Burer and D. Vandenbussche. Solving lift-and-project relaxations of binary integer programs. *SIAM Journal on Optimization*, 16(3):726–750, 2006.
- [8] E. Elhamifar, G. Sapiro, and R. Vidal. See all by looking at a few: Sparse modeling for finding representative objects. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pages 1600–1608, 2012.
- [9] E. Esser, M. Möller, S. Osher, G. Sapiro, and J. Xin. A convex model for nonnegative matrix factorization and dimensionality reduction on physical space. *IEEE Transactions on Image Processing*, 21(7):3239–3252, 2012.
- [10] X. Fu, K. Huang, N. D. Sidiropoulos, and W.-K. Ma. Nonnegative matrix factorization for signal and data analytics: Identifiability, algorithms, and applications. *IEEE Signal Processing Magazine*, 36(2):59–80, 2019.
- [11] X. Fu and W.-K. Ma. Robustness analysis of structured matrix factorization via self-dictionary mixed-norm optimization. *IEEE Signal Processing Letters*, 23(1):60–64, 2016.
- [12] X. Fu, N. D. Sidiropoulos, and W.-K. Ma. Power spectra separation via structured matrix factorization. *IEEE Transactions on Signal Processing*, 64(17):4592–4605, 2016.
- [13] N. Gillis. Robustness analysis of Hottopixx, a linear programming model for factoring nonnegative matrices. *SIAM Journal on Matrix Analysis and Applications*, 34(3):1189–1212, 2013.
- [14] N. Gillis. Successive nonnegative projection algorithm for robust nonnegative blind source separation. *SIAM Journal on Imaging Sciences*, 7(2):1420–1450, 2014.
- [15] N. Gillis. *Nonnegative Matrix Factorization*. SIAM, 2020.
- [16] N. Gillis and R. Luce. Robust near-separable nonnegative matrix factorization using linear optimization. *Journal of Machine Learning Research*, 15:1249–1280, 2014.
- [17] N. Gillis and R. Luce. A fast gradient method for nonnegative sparse regression with self-dictionary. *IEEE Transactions on Image Processing*, 27(1):24–37, 2018.
- [18] N. Gillis and S. A. Vavasis. Fast and robust recursive algorithms for separable nonnegative matrix factorization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(4):698–714, 2014.
- [19] N. Gillis and S. A. Vavasis. Semidefinite programming based preconditioning for more robust near-separable nonnegative matrix factorization. *SIAM Journal on Optimization*, 25(1):677–698, 2015.
- [20] W.-K. Ma, J. M. Bioucas-Dias, T.-H. Chan, N. Gillis, P. Gader, A. J. Plaza, A. Ambikapathi, and C.-Y. Chi. A signal processing perspective on hyperspectral unmixing. *IEEE Signal Processing Magazine*, 31(1):67–81, 2014.
- [21] T. Mizutani. Ellipsoidal rounding for nonnegative matrix factorization under noisy separability. *Journal of Machine Learning Research*, 15:1011–1039, 2014.
- [22] T. Mizutani. Refinement of Hottopixx method for nonnegative matrix factorization under noisy separability. *SIAM Journal on Matrix Analysis and Applications*, 43(3):1029–1057, 2022.
- [23] J. M. P. Nascimento and J. M. B. Dias. Vertex component analysis: A fast algorithm to unmix hyperspectral data. *IEEE Transactions on Geoscience and Remote Sensing*, 43(4):898–910, 2005.
- [24] T. Nguyen, X. Fu, and R. Wu. Memory-efficient convex optimization for self-dictionary separable nonnegative matrix factorization: A frank-wolfe approach. *IEEE Transactions on Signal Processing*, 70:3221–3236, 2022.
- [25] F. Zhu. Hyperspectral unmixing: Ground truth labeling, datasets, benchmark performances and survey. arXiv:1708.05125, 2017.
- [26] F. Zhu, Y. Wang, B. Fan, S. Xiang, G. Meng, and C. Pan. Spectral unmixing via data-guided sparsity. *IEEE Transactions on Image Processing*, 23(12):5412–5427, 2014.

APPENDIX A
PROOF OF THEOREM 2

First, we will prove part (i) of Theorem 2 and then address part (ii).

Proof of part (i) of Theorem 2: First, we show that $\text{opt}(\mathcal{H}(\mathcal{L}, \mathcal{L})) \geq \text{opt}(\mathcal{H}(\mathcal{L}, \mathcal{N}))$ holds under condition (C1). The matrix $[X^*, \Gamma^*]$ belongs to $\mathcal{F}(\ell, n)$, and thus, it is a feasible solution to $\mathcal{H}(\mathcal{L}, \mathcal{N})$. In addition, we obtain $\|A(\mathcal{L}) - A(\mathcal{L})X^*\|_1 = \text{opt}(\mathcal{H}(\mathcal{L}, \mathcal{L}))$ by Theorem 1 (ii), and $\|A(\mathcal{N} \setminus \mathcal{L}) - A(\mathcal{L})\Gamma^*\|_1 \leq \text{opt}(\mathcal{H}(\mathcal{L}, \mathcal{L}))$ by condition (C1) and Theorem 1 (i). Hence, the objective function value of $\mathcal{H}(\mathcal{L}, \mathcal{N})$ at $[X^*, \Gamma^*]$ satisfies

$$\begin{aligned} \text{opt}(\mathcal{H}(\mathcal{L}, \mathcal{N})) &\leq \|A\Pi - A(\mathcal{L})[X^*, \Gamma^*]\|_1 \\ &= \|[A(\mathcal{L}), A(\mathcal{N} \setminus \mathcal{L})] - A(\mathcal{L})[X^*, \Gamma^*]\|_1 \\ &= \|[A(\mathcal{L}) - A(\mathcal{L})X^*, A(\mathcal{N} \setminus \mathcal{L}) - A(\mathcal{L})\Gamma^*]\|_1 \\ &= \text{opt}(\mathcal{H}(\mathcal{L}, \mathcal{L})). \end{aligned}$$

Next, we show the reverse inequality. Let $\hat{X}^* = [\hat{x}_1^*, \dots, \hat{x}_n^*]$ denote the optimal solution of $\mathcal{H}(\mathcal{L}, \mathcal{N})$. We find that

$$\begin{aligned} \text{opt}(\mathcal{H}(\mathcal{L}, \mathcal{N})) &= \|A\Pi - A(\mathcal{L})\hat{X}^*\|_1 \\ &= \|[A(\mathcal{L}), A(\mathcal{N} \setminus \mathcal{L})] - A(\mathcal{L})\hat{X}^*\|_1 \\ &\geq \|[A(\mathcal{L}) - A(\mathcal{L})[\hat{x}_1^*, \dots, \hat{x}_\ell^*]]\|_1 \\ &\geq \text{opt}(\mathcal{H}(\mathcal{L}, \mathcal{L})). \end{aligned}$$

The last inequality comes from that the matrix $[\hat{x}_1^*, \dots, \hat{x}_\ell^*]$ belongs to $\mathcal{F}(\ell, \ell)$, and thus, it is a feasible solution to $\mathcal{H}(\mathcal{L}, \mathcal{L})$. The two inequalities we showed above yield

$$\text{opt}(\mathcal{H}(\mathcal{L}, \mathcal{L})) = \text{opt}(\mathcal{H}(\mathcal{L}, \mathcal{N})) = \|A\Pi - A(\mathcal{L})[X^*, \Gamma^*]\|_1.$$

The desired result follows upon recalling that $[X^*, \Gamma^*]$ is a feasible solution of $\mathcal{H}(\mathcal{L}, \mathcal{N})$. ■

Next, we prove part (ii) of Theorem 2. To do so, we need Lemmas 2 and 3.

- *Lemma 2:* Let α^* be the optimal solution of $\mathcal{P}(\mathcal{L}, \mathcal{L})$. If α^* satisfies condition (C1), then we can construct α from α^* such that α is a feasible solution of $\mathcal{P}(\mathcal{N}, \mathcal{N})$ and the objective function value of $\mathcal{P}(\mathcal{N}, \mathcal{N})$ at α is $\text{opt}(\mathcal{P}(\mathcal{L}, \mathcal{L}))$.
- *Lemma 3:* Let β^* be the optimal solution of $\mathcal{D}(\mathcal{L}, \mathcal{L})$. If β^* satisfies condition (C2), then we can construct β from β^* such that β is a feasible solution of $\mathcal{D}(\mathcal{N}, \mathcal{N})$ and the objective function value of $\mathcal{D}(\mathcal{N}, \mathcal{N})$ at β is $\text{opt}(\mathcal{D}(\mathcal{L}, \mathcal{L}))$.

Theorem 1 (i) tells us that $\text{opt}(\mathcal{P}(\mathcal{L}, \mathcal{L})) = \text{opt}(\mathcal{D}(\mathcal{L}, \mathcal{L}))$. Accordingly, as we reviewed in Section II-B, the weak duality theorem implies that α and β are optimal solutions of $\mathcal{P}(\mathcal{N}, \mathcal{N})$ and $\mathcal{D}(\mathcal{N}, \mathcal{N})$, respectively. Furthermore, it means that $\text{opt}(\mathcal{P}(\mathcal{N}, \mathcal{N})) = \text{opt}(\mathcal{P}(\mathcal{L}, \mathcal{L}))$. As a result, using parts (i) and (ii) of Theorem 1, we can establish part (ii) of Theorem 2.

Here, we give formal descriptions of Lemmas 2 and 3 and then prove them.

Lemma 2: Let $\mathcal{L} \subset \mathcal{N}$. Let $\alpha^* = (X^*, F^*, G^*, u^*)$ be the optimal solution of $\mathcal{P}(\mathcal{L}, \mathcal{L})$. Assume that condition (C1) holds.

By letting $\Gamma^* = [\gamma_j^* : j \in \mathcal{N} \setminus \mathcal{L}]$ for the optimal solution γ_j^* of $\mathcal{R}_j(\mathcal{L}, X^*)$ and letting Π be a permutation matrix of size n satisfying $A\Pi = [A(\mathcal{L}), A(\mathcal{N} \setminus \mathcal{L})]$, construct $\alpha = (X, F, G, u) \in \mathbb{R}^{n \times n} \times \mathbb{R}^{d \times n} \times \mathbb{R}^{d \times n} \times \mathbb{R}$ as follows:

$$X = \Pi \begin{bmatrix} X^* & \Gamma^* \\ O & O \end{bmatrix} \Pi^\top, \quad F = R^+, \quad G = R^-, \quad \text{and } u = \|R\|_1$$

where $R = A - AX$ for X as constructed above. Then, the following hold.

- (i) α is a feasible solution of $\mathcal{P}(\mathcal{N}, \mathcal{N})$.
- (ii) The objective function value of $\mathcal{P}(\mathcal{N}, \mathcal{N})$ at α is $\text{opt}(\mathcal{P}(\mathcal{L}, \mathcal{L}))$, i.e., $u = \text{opt}(\mathcal{P}(\mathcal{L}, \mathcal{L}))$.

Proof: As a simplification, we can write $X = \Pi \hat{X} \Pi^\top$ for X above by letting

$$\hat{X} = \begin{bmatrix} X^* & \Gamma^* \\ O & O \end{bmatrix}.$$

Part (i) It is sufficient to prove that α satisfies the first, second, and fifth constraints of $\mathcal{P}(\mathcal{N}, \mathcal{N})$, since $F = R^+ \geq O$ and $G = R^- \geq O$. Let us look at the first constraint. Since R , F and G are constructed as $R = A - AX$, $F = R^+$ and $G = R^-$, we have

$$A - AX = R = R^+ - R^- = F - G.$$

Next, let us look at the second constraint. Equality (1) tells us that

$$u = \|R\|_1 = \max_{j=1, \dots, n} \sum_{i=1}^d R^+(i, j) + R^-(i, j).$$

This implies that $\sum_{i=1}^d F(i, j) + G(i, j) \leq u$ holds for $j = 1, \dots, n$. Finally, let us look at the fifth constraint. Since $X^* \in \mathcal{F}(\ell, \ell)$ and $\mathbf{0} \leq \gamma_j^* \leq \text{diag}(X^*)$, we see that $\hat{X} \in \mathcal{F}(n, n)$. In addition, since X is constructed as $X = \Pi \hat{X} \Pi^\top$, the (i, j) th entry of X is given by $\hat{X}(\sigma(i), \sigma(j))$ for some permutation σ of size n . Hence, $X \in \mathcal{F}(n, n)$. Thus, the desired result follows.

Part (ii) The objective function value of $\mathcal{P}(\mathcal{N}, \mathcal{N})$ at α can be rewritten as

$$u = \|R\|_1 = \|A - AX\|_1 = \|A\Pi - A\Pi\hat{X}\|_1.$$

In light of the relation $A\Pi = [A(\mathcal{L}), A(\mathcal{N} \setminus \mathcal{L})]$, it can be further expressed as

$$\|A\Pi - A\Pi\hat{X}\|_1 = \|[S, T]\|_1$$

for $S = A(\mathcal{L}) - A(\mathcal{L})X^*$ and $T = A(\mathcal{N} \setminus \mathcal{L}) - A(\mathcal{L})\Gamma^*$. It follows from parts (i) and (ii) of Theorem 1 that $\text{opt}(\mathcal{P}(\mathcal{L}, \mathcal{L})) = \text{opt}(\mathcal{H}(\mathcal{L}, \mathcal{L})) = \|A(\mathcal{L}) - A(\mathcal{L})X^*\|_1 = \|S\|_1$. It follows from condition (C1) that $\|T\|_1 = \|A(\mathcal{N} \setminus \mathcal{L}) - A(\mathcal{L})\Gamma^*\|_1 \leq \text{opt}(\mathcal{P}(\mathcal{L}, \mathcal{L}))$. Consequently, we obtain $u = \text{opt}(\mathcal{P}(\mathcal{L}, \mathcal{L}))$. ■

Lemma 3: Let $\mathcal{L} \subset \mathcal{N}$. Let $\beta^* = (Y^*, Z^*, \mathbf{s}^*, \mathbf{t}^*, v^*)$ be the optimal solution of $\mathcal{D}(\mathcal{L}, \mathcal{L})$. Assume that condition (C2) holds. By letting Π be a permutation matrix of size n satisfying

$A\Pi = [A(\mathcal{L}), A(\mathcal{N} \setminus \mathcal{L})]$, and $\Delta = (Y^*)^\top A(\mathcal{N} \setminus \mathcal{L})$, construct $\beta = (Y, Z, \mathbf{s}, \mathbf{t}, v) \in \mathbb{R}^{d \times n} \times \mathbb{R}^{n \times n} \times \mathbb{R}^n \times \mathbb{R}^n \times \mathbb{R}$ as follows:

$$Y = [Y^*, O] \Pi^\top, \quad Z = \Pi \begin{bmatrix} Z^* & \Delta^+ \\ O & O \end{bmatrix} \Pi^\top, \quad \mathbf{s} = \Pi \begin{bmatrix} \mathbf{s}^* \\ \mathbf{0} \end{bmatrix},$$

$$\mathbf{t} = \Pi \begin{bmatrix} \mathbf{t}^* \\ \mathbf{0} \end{bmatrix} \text{ and } v = v^*.$$

Then, the following hold.

- (i) β is a feasible solution of $D(\mathcal{N}, \mathcal{N})$.
- (ii) The objective function value of $D(\mathcal{N}, \mathcal{N})$ at β is $\text{opt}(D(\mathcal{L}, \mathcal{L}))$.

Proof: The proof uses the following relation. Let $\mathbf{a} \in \mathbb{R}^n$ and $P \in \mathbb{R}^{n \times n}$ be a permutation matrix. Then, a straightforward calculation gives

$$\text{diag}(P\mathbf{a}) = P\text{diag}(\mathbf{a})P^\top. \quad (8)$$

As a simplification, we can write $Y = \hat{Y}\Pi^\top, Z = \Pi\hat{Z}\Pi^\top, \mathbf{s} = \Pi\hat{\mathbf{s}}$ and $\mathbf{t} = \Pi\hat{\mathbf{t}}$ for $Y, Z, \mathbf{s}$ and $\mathbf{t}$ above by letting

$$\hat{Y} = [Y^*, O], \quad \hat{Z} = \begin{bmatrix} Z^* & \Delta^+ \\ O & O \end{bmatrix}, \quad \hat{\mathbf{s}} = \begin{bmatrix} \mathbf{s}^* \\ \mathbf{0} \end{bmatrix} \text{ and } \hat{\mathbf{t}} = \begin{bmatrix} \mathbf{t}^* \\ \mathbf{0} \end{bmatrix}.$$

Part (i) It is sufficient to prove that β satisfies the first and the second constraints of $D(\mathcal{N}, \mathcal{N})$, since, obviously, it satisfies the other constraints. First, look at the second constraint. We find that β satisfies the second constraint of $D(\mathcal{N}, \mathcal{N})$ if and only if β^* satisfies that of $D(\mathcal{L}, \mathcal{L})$. Indeed,

$$\begin{aligned} & -J \cdot \text{diag}(\mathbf{s}) \leq Y \leq J \cdot \text{diag}(\mathbf{s}) \\ \Leftrightarrow & -J \cdot \text{diag}(\Pi\hat{\mathbf{s}}) \leq \hat{Y}\Pi^\top \leq J \cdot \text{diag}(\Pi\hat{\mathbf{s}}) \\ \Leftrightarrow & -J\Pi \cdot \text{diag}(\hat{\mathbf{s}}) \cdot \Pi^\top \leq \hat{Y}\Pi^\top \leq J\Pi \cdot \text{diag}(\hat{\mathbf{s}}) \cdot \Pi^\top \text{ (by (8))} \\ \Leftrightarrow & -J \cdot \text{diag}(\hat{\mathbf{s}}) \leq \hat{Y} \leq J \cdot \text{diag}(\hat{\mathbf{s}}) \\ \Leftrightarrow & -J \cdot \text{diag}(\mathbf{s}^*) \leq Y^* \leq J \cdot \text{diag}(\mathbf{s}^*). \end{aligned}$$

Next, look at the first constraint. By plugging in β on the left-hand side of the constraint, we get the following matrix $M \in \mathbb{R}^{n \times n}$:

$$M = A^\top Y + vI - \text{diag}(\mathbf{t}) - Z^\top + \text{diag}(Z^\top \mathbf{1}).$$

Then, we rewrite M as

$$\begin{aligned} M &= A^\top \hat{Y}\Pi^\top + v^*I - \Pi \cdot \text{diag}(\hat{\mathbf{t}}) \cdot \Pi^\top - \Pi\hat{Z}^\top\Pi^\top \\ &\quad + \Pi \cdot \text{diag}(\hat{Z}^\top \mathbf{1}) \cdot \Pi^\top \text{ (by (8))} \\ &= \Pi(A\Pi)^\top \hat{Y}\Pi^\top + v^*I - \Pi \cdot \text{diag}(\hat{\mathbf{t}}) \cdot \Pi^\top - \Pi\hat{Z}^\top\Pi^\top \\ &\quad + \Pi \cdot \text{diag}(\hat{Z}^\top \mathbf{1}) \cdot \Pi^\top, \end{aligned}$$

which yields

$$\Pi^\top M \Pi = (A\Pi)^\top \hat{Y} + v^*I - \text{diag}(\hat{\mathbf{t}}) - \hat{Z}^\top + \text{diag}(\hat{Z}^\top \mathbf{1}).$$

Here,

$$\begin{aligned} (A\Pi)^\top \hat{Y} &= \begin{bmatrix} A(\mathcal{L})^\top Y^* & O \\ A(\mathcal{N} \setminus \mathcal{L})^\top Y^* & O \end{bmatrix}, \\ \text{diag}(\hat{\mathbf{t}}) &= \begin{bmatrix} \text{diag}(\mathbf{t}^*) & O \\ O & O \end{bmatrix}, \\ \hat{Z}^\top &= \begin{bmatrix} (Z^*)^\top & O \\ (\Delta^+)^\top & O \end{bmatrix}, \\ \text{diag}(\hat{Z}^\top \mathbf{1}) &= \begin{bmatrix} \text{diag}((Z^*)^\top \mathbf{1}) & O \\ O & \text{diag}((\Delta^+)^\top \mathbf{1}) \end{bmatrix}. \end{aligned}$$

We set $\tilde{M} = \Pi^\top M \Pi$ and partition $\tilde{M}$ into four blocks $\tilde{M}_{11} \in \mathbb{R}^{\ell \times \ell}, \tilde{M}_{12} \in \mathbb{R}^{\ell \times (n-\ell)}, \tilde{M}_{21} \in \mathbb{R}^{(n-\ell) \times \ell}$ and $\tilde{M}_{22} \in \mathbb{R}^{(n-\ell) \times (n-\ell)}$ in the following manner:

$$\tilde{M} = \begin{bmatrix} \tilde{M}_{11} & \tilde{M}_{12} \\ \tilde{M}_{21} & \tilde{M}_{22} \end{bmatrix}$$

where each block is given as

$$\begin{aligned} \tilde{M}_{11} &= A(\mathcal{L})^\top Y^* + v^*I - \text{diag}(\mathbf{t}^*) - (Z^*)^\top + \text{diag}((Z^*)^\top \mathbf{1}), \\ \tilde{M}_{12} &= O, \\ \tilde{M}_{21} &= A(\mathcal{N} \setminus \mathcal{L})^\top Y^* - (\Delta^+)^\top, \\ \tilde{M}_{22} &= v^*I + \text{diag}((\Delta^+)^\top \mathbf{1}). \end{aligned}$$

We observe the following: $\tilde{M}_{11} \leq O$ since β^* satisfies the first constraint of $D(\mathcal{L}, \mathcal{L})$; $\tilde{M}_{21} \leq O$ since

$$\tilde{M}_{21} = A(\mathcal{N} \setminus \mathcal{L})^\top Y^* - (\Delta^+)^\top = \Delta^\top - (\Delta^+)^\top \leq O;$$

and, $\tilde{M}_{22} \leq O$ since

$$\begin{aligned} \tilde{M}_{22} &= v^*I + \text{diag}((\Delta^+)^\top \mathbf{1}) \\ &= \text{diag}(v^* + \mathbf{1}^\top \Delta^+(:, 1), \dots, v^* + \mathbf{1}^\top \Delta^+(:, n-\ell)) \\ &\leq O \text{ (by condition (C2))}. \end{aligned}$$

Accordingly, $\tilde{M} = \Pi^\top M \Pi \leq O \Leftrightarrow M \leq O$, which means that β satisfies the first constraint of $D(\mathcal{N}, \mathcal{N})$. Consequently, we obtain the desired result.

Part (ii) By a straightforward calculation, we find that

$$\begin{aligned} \langle A, Y \rangle + rv - \mathbf{1}^\top \mathbf{t} &= \langle A\Pi, Y\Pi \rangle + rv - \mathbf{1}^\top \mathbf{t} \\ &= \langle [A(\mathcal{L}), A(\mathcal{N} \setminus \mathcal{L})], [Y^*, O] \rangle + rv^* \\ &\quad - \mathbf{1}^\top \Pi \hat{\mathbf{t}} \\ &= \langle A(\mathcal{L}), Y^* \rangle + rv^* - \mathbf{1}^\top \mathbf{t}^* \\ &= \text{opt}(D(\mathcal{L}, \mathcal{L})). \end{aligned}$$

Now, we are ready to prove part (ii) of Theorem 2. ■

Proof of part (ii) of Theorem 2: Let α and β be as constructed in Lemmas 2 and 3. From the lemmas and Theorem 1 (i), we find that α and β are feasible solutions of $P(\mathcal{N}, \mathcal{N})$ and $D(\mathcal{N}, \mathcal{N})$ and the objective function value of $P(\mathcal{N}, \mathcal{N})$ at α is equal to that of $D(\mathcal{N}, \mathcal{N})$ at β . Hence, the weak duality theorem implies that α and β are optimal solutions of $P(\mathcal{N}, \mathcal{N})$ and $D(\mathcal{N}, \mathcal{N})$. Furthermore, this means that $\text{opt}(P(\mathcal{L}, \mathcal{L})) = \text{opt}(P(\mathcal{N}, \mathcal{N}))$. Consequently, it follows from parts (i) and (ii) of Theorem 1 that $\text{opt}(H(\mathcal{L}, \mathcal{L})) = \text{opt}(H(\mathcal{N}, \mathcal{N}))$ holds and X of α is the optimal solution of $H(\mathcal{N}, \mathcal{N})$. ■

Supplemental Material of “Implementing Hottopixx Methods for Endmember Extraction in Hyperspectral Images”

Tomohiko Mizutani

APPENDIX B PROOF OF THEOREM 1

Proof: Let (X^*, F^*, G^*, u^*) be the optimal solution of $P(\mathcal{L}, \mathcal{M})$. Then, X^* is a feasible solution of $H(\mathcal{L}, \mathcal{M})$. Thus, we get $\text{opt}(H(\mathcal{L}, \mathcal{M})) \leq \|A(\mathcal{M})\Pi - A(\mathcal{L})X^*\|_1$. Furthermore, taking into account that (X^*, F^*, G^*, u^*) satisfies all constraints of $P(\mathcal{L}, \mathcal{M})$, we find that

$$\begin{aligned} \|A(\mathcal{M})\Pi - A(\mathcal{L})X^*\|_1 &= \|F^* - G^*\|_1 \\ &= \max_{j=1,\dots,m} \|F^*(:,j) - G^*(:,j)\|_1 \\ &\leq \max_{j=1,\dots,m} \|F^*(:,j)\|_1 + \|G^*(:,j)\|_1 \\ &= \max_{j=1,\dots,m} \sum_{i=1}^d F^*(i,j) + G^*(i,j) \\ &\leq u^* = \text{opt}(P(\mathcal{L}, \mathcal{M})). \end{aligned}$$

Accordingly,

$$\text{opt}(H(\mathcal{L}, \mathcal{M})) \leq \|A(\mathcal{M})\Pi - A(\mathcal{L})X^*\|_1 \leq \text{opt}(P(\mathcal{L}, \mathcal{M})). \quad (9)$$

Let $\hat{X}^*$ be the optimal solution of $H(\mathcal{L}, \mathcal{M})$, and let us use it to construct $R = A(\mathcal{M})\Pi - A(\mathcal{L})\hat{X}^*$. Note that R satisfies the relation $\|R\|_1 = \text{opt}(H(\mathcal{L}, \mathcal{M}))$, which will be used below. We show that $(\hat{X}^*, R^+, R^-, \|R\|_1)$ is a feasible solution of $P(\mathcal{L}, \mathcal{M})$. Obviously, it satisfies all the constraints except the second one. Here, recall equality (1) shown in Section II-A. For the second constraint, this equality implies that

$$\sum_{i=1}^d R^+(i,j) + R^-(i,j) \leq \|R\|_1.$$

Accordingly,

$$\text{opt}(H(\mathcal{L}, \mathcal{M})) = \|R\|_1 \geq \text{opt}(P(\mathcal{L}, \mathcal{M})). \quad (10)$$

In addition, as explained just before this theorem, the duality theorem ensures

$$\text{opt}(P(\mathcal{L}, \mathcal{M})) = \text{opt}(D(\mathcal{L}, \mathcal{M})). \quad (11)$$

From (9), (10) and (11), we obtain the equalities,

$$\begin{aligned} \text{opt}(P(\mathcal{L}, \mathcal{M})) &= \text{opt}(D(\mathcal{L}, \mathcal{M})) = \text{opt}(H(\mathcal{L}, \mathcal{M})) \\ &= \|A(\mathcal{M})\Pi - A(\mathcal{L})X^*\|_1 = \|R\|_1. \end{aligned}$$

As shown above, X^* and $(\hat{X}^*, R^+, R^-, \|R\|_1)$ are feasible solutions of $H(\mathcal{L}, \mathcal{M})$ and $P(\mathcal{L}, \mathcal{M})$, respectively. Hence, this completes the proof of parts (i)-(iii). ■

APPENDIX C PROOF OF LEMMA 1

Proof: We plug in the optimal solution of $D(\mathcal{L}, \mathcal{L})$ for the first constraint and then express the resulting inequality as $M \leq O$ by letting

$$M = A(\mathcal{L})^\top Y^* + v^* I - \text{diag}(\mathbf{t}^*) - (Z^*)^\top + \text{diag}((Z^*)^\top \mathbf{1}).$$

The trace of M is given as

$$\text{tr}(M) = \langle A(\mathcal{L}), Y^* \rangle + \ell v^* - \mathbf{1}^\top \mathbf{t}^* + \sum_{i \neq j} Z^*(i, j).$$

Here, the sum of $Z^*(i, j)$ run over all pairs (i, j) except $i = j$. Let $c = \langle A(\mathcal{L}), Y^* \rangle + \ell v^* - \mathbf{1}^\top \mathbf{t}^*$. Then, $\text{opt}(D(\mathcal{L}, \mathcal{L})) - c = (r - \ell)v^*$. We use this relation to prove the lemma. We have $c \leq 0$, since

$$c = \langle A(\mathcal{L}), Y^* \rangle + \ell v^* - \mathbf{1}^\top \mathbf{t}^* = \text{tr}(M) - \sum_{i \neq j} Z^*(i, j) \leq 0$$

by $M \leq O$ and $Z^* \geq O$. We also have $\text{opt}(D(\mathcal{L}, \mathcal{L})) \geq 0$. Indeed, $\text{opt}(D(\mathcal{L}, \mathcal{L})) = \text{opt}(H(\mathcal{L}, \mathcal{L}))$ by Theorem 1 (i), and $\text{opt}(H(\mathcal{L}, \mathcal{L})) \geq 0$, since the objective function value of $H(\mathcal{L}, \mathcal{L})$ always takes a nonnegative value. We thus get

$$\text{opt}(D(\mathcal{L}, \mathcal{L})) - c \geq 0 \Leftrightarrow (r - \ell)v^* \geq 0.$$

This implies that $v^* \leq 0$, since $\mathcal{L}$ is chosen to be $\ell \geq r \Leftrightarrow r - \ell \leq 0$. ■

APPENDIX D RGB IMAGES OF HSI DATASETS



Fig. 5. RGB images of Jasper Ridge (top-left), Samson (top-right), and Urban (bottom).

APPENDIX E ABUNDANCE MAPS FOR URBAN

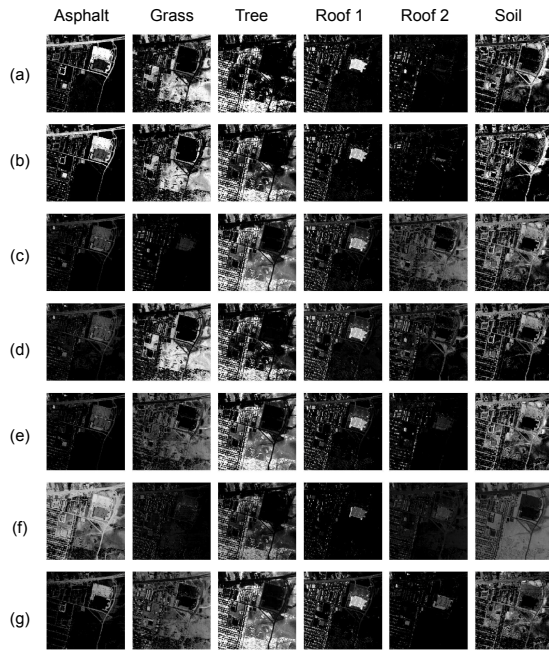


Fig. 6. Ground truth and abundance maps obtained by the algorithms for Urban with $\phi = 0.4$ and $\omega = 0.1$: (a) ground truth obtained by using the reference signatures, (b) EEHT-C, (c) SPA, (d) PSPA, (e) ER, (f) VCA, and (g) SNPA.