ORIGINAL PAPER

Kronecker least angle regression for unsupervised unmixing of hyperspectral imaging data

Ahmed Elrewainy[1] · Sherif S. Sherif[1]

Abstract

Spectral unmixing is an important data processing task that is commonly applied to hyperspectral imaging data. It uses a set of spectral pixels, i.e., multichannel data, to separate each spectral pixel, a multicomponent signal comprised of a linear mixture of pure spectral signatures, into its individual spectral signatures commonly known as endmembers. When no prior information about the required endmembers is available, the resulting unsupervised unmixing problem will be underdetermined, and additional constraints become necessary. A recent approach to solving this problem required that these endmembers be sparse in some dictionary. Sparse signal recovery is commonly solved using a *basis pursuit* optimization algorithm that requires specifying a data-dependent regularization parameter. Least angle regression (LARS) is a very efficient method to simultaneously solve the *basis pursuit* optimization problem for all relevant regularization parameter values. However, despite this efficiency of LARS, it has not been applied to the spectral unmixing problem before. This is likely because the application of LARS to large multichannel data could be very challenging in practice, due to the need for generation and storage of extremely large arrays ($\sim 10^{10}$ bytes in a relatively small area of spectral unmixing problem). In this paper, we extend the standard LARS algorithm, using Kronecker products, to make it suitable for practical and efficient recovery of sparse signals from large multichannel data, i.e., without the need to construct or process very large arrays, or the need for trial and error to determine the regularization parameter value. We then apply this new Kronecker LARS (K-LARS) algorithm to successfully achieve spectral unmixing of both synthetic and *AVIRIS* hyperspectral imaging data. We also compare our results to ones obtained using an earlier *basis pursuit*-based spectral unmixing algorithm, generalized morphological component analysis (GMCA). We show that these two results are similar, albeit our results were obtained without trial and error, or arbitrary choices, in specifying the regularization parameter. More important, our K-LARS algorithm could be a very valuable research tool to the signal processing community, where it could be used to solve sparse least squares problems involving large multichannel data.

Keywords Kronecker least angle regression · Multichannel signal recovery · Hyperspectral imaging · Spectral unmixing · Kronecker products

1 Introduction

Hyperspectral imaging (HSI) acquires images using hundreds of contiguous narrow electromagnetic wavelength bands that result in spectral pixels that represent spectral signatures of materials in a scene [1]. As every material has its own spectral signature that is different from other materials, HSI could be used to discriminate between different materials that are present in an acquired scene. That is why HSI has been used in a wide variety of civilian, environmental and military applications [2, 3]. Most digital processing algorithms applied to HSI data include spectral unmixing [4], anomaly detection [3], spectra classification [5] and material identification [6]. Spectral unmixing, i.e., the separation of a spectral pixel into its individual pure spectral signatures, commonly referred to as endmembers, has been an important data processing task from the earliest days of hyperspectral imaging.

This is because hyperspectral cameras typically have low spatial resolutions and image acquisition usually takes place

✉ Sherif S. Sherif
  Sherif.Sherif@umanitoba.ca

1   Electrical and Computer Engineering Department, Faculty of Engineering, University of Manitoba, Winnipeg, MB R3T 5V6, Canada

from very far distances, e.g., in the case of satellite imaging. Therefore, an acquired spectral pixel is usually comprised of a linear mixture of endmembers that exist in the imaged region corresponding to it. The proportion of each endmember in this linear mixture is called its abundance. Spectral unmixing of a group of spectral pixels involves determining the endmembers, in addition to their abundances, that are available in each spectral pixel.

An overview of approaches to linear spectral unmixing in HSI is given in [4]. These approaches could be categorized into four broad categories. First, geometrical approaches that exploit the fact that acquired spectral pixels must lie in a linear simplex formed by the available endmembers [7]. Second, statistical approaches, e.g., Bayesian techniques that use probabilistic data models, utilize prior distributions to impose model constraints and estimate parameters of posterior probability distributions [8, 9]. Third, spatial–spectral contextual approaches that exploit both spatial and spectral features available in HSI data through computing spatial correlations between different spectral pixels [10]. Fourth, sparse signal recovery approaches assume that the unknown endmembers are sparse in some dictionary, where they could be obtained using a sparse signal recovery algorithm.

Sparse signal recovery-based spectral unmixing techniques could be implemented in a supervised fashion. This is done by assuming that each acquired spectral pixel is a linear combination of a priori known endmembers. These endmembers, i.e., pure spectral signatures, would have been obtained in a laboratory beforehand using a spectroradiometer. Supervised spectral unmixing then aims to find the optimal subset of endmembers from a very large library that can best fit each spectral pixel [11, 12].

Also, sparse signal recovery-based spectral unmixing could be solved using an unsupervised approach, where a priori knowledge of the endmembers would not be required. This unsupervised spectral unmixing could be viewed as a blind source separation (BSS) problem [13], where the unknown sources are assumed to be sparse signals. Two common mathematical formulations of this sparse BSS problem are sparse nonnegative matrix factorization (S-NMF) [14–19] and generalized morphological component analysis (GMCA) [20–22]. Both formulations are typically solved using a coordinate descent optimization approach [23], where one alternates between estimating endmembers while keeping their abundances constant (endmembers estimation step), and estimating abundances while keeping endmembers constant (abundances estimation step). Both steps are repeated till conversion to the sought optimal values of both endmembers and their abundances.

In its endmember estimation step, GMCA solves an l_1-norm minimization problem, i.e., a *basis pursuit* problem [24] using thresholding methods [25] to obtain sparse estimates of the endmembers from the given multichannel

hyperspectral data. Any *basis pursuit* problem requires specifying a data-dependent regularization parameter to specify the relative importance of sparsity levels of unknown variables and the error in their estimates, also this parameter sets the threshold value in the GMCA algorithm. Therefore, computationally inefficient trial and error is typically needed to find a suitable value for this parameter. GMCA, however, exploits the fact that it solves the *basis pursuit* problem only as a step in the larger coordinate descent optimization procedure. In its first iteration of coordinate descent, GMCA solves the required *basis pursuit* problem with a large initial value of this regularization parameter (obtained using trial and error), corresponding to endmembers with a high level of sparsity. In its following iterations of coordinate descent, GMCA solves the required *basis pursuit* problem with a lower value of this regularization parameter, thereby increasing details of the estimated endmembers. However, trial and error is a computationally inefficient manner to choose the initial value of the regularization parameter, and the arbitrary manner, in choosing the schedule used to lower its value, is not optimal and could affect convergence and spectral unmixing results obtained using GMCA.

Least angle regression (LARS) could solve the *basis pursuit* minimization problem efficiently [26], by simultaneously obtaining solutions corresponding to all relevant values of λ, with a computational complexity comparable to solving a single unconstrained least squares problem. However, despite this clear computational advantage of LARS, it has not been applied to the spectral unmixing problem before. This is likely the case because the LARS algorithm is not directly suitable for multichannel and multicomponent HSI data, where data vectorization [27] would be necessary before its application. Such HSI data vectorization would result in extremely large matrices (~ 10^{10} elements) and vectors that would be very challenging to store and process using a typical computer.

In this paper, we exploit the properties of Kronecker products [28] to extend the LARS algorithm to handle large multichannel and multicomponent data without the need to construct or process very large arrays. This extension of the LARS algorithm would make the application of LARS to HSI spectral unmixing practical. Also, this extension of LARS would be of general importance, where it could be used for practical and efficient sparse signal recovery from large multichannel data. We refer to our extended LARS method as the Kronecker LARS (K-LARS) algorithm.

This paper is arranged as follows: Sect. 2 presents a mathematical formulation of the HSI spectral unmixing problem. Section 3 describes the sparsity-based approach to solve this problem. Section 4 presents our new K-LARS algorithm. In Sect. 5, we apply K-LARS to successfully achieve unsupervised spectral unmixing of both synthetic and *AVIRIS* HSI data. We also compare our results to ones obtained using an

earlier *basis pursuit*-based algorithm (GMCA). We show that these two results are similar, albeit our results were obtained without arbitrary choices related to specifying the regularization parameter.

2 Spectral unmixing problem definition

2.1 Linear mixture model (LMM)

A linear mixing model (LMM) is widely used to represent the mixing of spectra in HSI [4]. Any spectral pixel, y_i, could be represented as a superposition of pure endmembers that are available in the imaged scene,

$$y_i = \sum_{j=1}^{M} s_j a_{ji} + e_i \tag{1}$$

where $s_j = \left[s_{j1}, \ldots s_{jl} \ldots, s_{jL}\right]^T$ is the absorption spectrum of the jth endmember of the available M endmembers and L is the number of HSI wavelengths. The abundance, $a_{ji} \geq 0$, is the proportion of the jth endmember in the ith spectral pixel that should be constrained as

$$\sum_{j=1}^{M} a_{ji} = 1 \tag{2}$$

The vector e_i represents additive noise. Equation (1) could be written in matrix form as

$$Y = SA + E \tag{3}$$

where $Y = \left[y_1, \ldots, y_N\right]$ is an $L \times N$ HSI data matrix, N is the number of spectral pixels, $S = [s_1, \ldots, s_M]$ is an $L \times M$ matrix whose columns represent the available endmembers, and A is an $M \times N$ matrix that contains all the abundances of the endmembers in all spectral pixels. The problem of spectral unmixing is to estimate the endmember matrix S and the abundance matrix A given the spectral pixels matrix Y. It could be considered a blind source separation problem [29].

2.2 Sparsity-constrained multichannel data fitting

For the LMM in (3), spectral unmixing could be formulated as a matrix least squares-based multichannel data fitting problem,

$$\min_{S,A} \frac{1}{2} \|Y - SA\|_2^2 \tag{4}$$

This problem is equivalent to decomposing the multichannel data matrix Y into the sum of rank-1 matrices. Typically,

such decomposition would not be unique, so additional constraints would be needed to obtain a unique solution. In sparsity-based spectral unmixing, such constraints could be added as a penalty term on the l_1-norm of S

$$\min_{S,A} \frac{1}{2} \|Y - SA\|_2^2 + \lambda \|S\|_1 \tag{5}$$

2.3 Sparse endmember representation

An endmember, s_l, is considered sparse in a dictionary, $\boldsymbol{\Phi} = \left[\boldsymbol{\varphi}_1, \ldots, \boldsymbol{\varphi}_T\right]$, if it can be represented as a superposition

$$s_l = \boldsymbol{\Phi}\boldsymbol{\alpha}_l = \sum_{i=1}^{T} \alpha_l[i]\boldsymbol{\varphi}_i \tag{6}$$

where only a small number, compared to the dimension of s_l, of the coefficients $\alpha_l[i]$ are nonzero. This sparsity could be quantified exactly by the l_0-norm of $\boldsymbol{\alpha}_l$, $\|\boldsymbol{\alpha}_l\|_0$ or quantified approximately by the l_1-norm of $\boldsymbol{\alpha}_l$, $\|\boldsymbol{\alpha}_l\|_1$. We note that any lp-norm with $p < 1$ does not fulfill the triangle inequality, thus is not strictly a norm [30].

2.4 Representation to promote endmember sparsity

To facilitate solving the minimization problem (5), we could transform it into a different domain, represented by, e.g., an $L \times L$ dictionary $\boldsymbol{\Phi}$, where the transformed endmembers $\boldsymbol{\alpha} = \boldsymbol{\Phi}^T S$ would likely be more sparse. The optimization problem (5) would then become

$$\min_{\boldsymbol{\alpha},\mathbf{A}} \frac{1}{2} \|\boldsymbol{\beta} - \boldsymbol{\alpha}A\|_2^2 + \lambda \|\boldsymbol{\alpha}\|_1 \tag{7}$$

where $\boldsymbol{\beta} = \boldsymbol{\Phi}^T Y$ is an $L \times N$ matrix whose columns represent coefficients of each transformed pixel y_i. After solving minimization problem (7), the original endmemebers S could be restored as $S = \tilde{\boldsymbol{\Phi}}\boldsymbol{\alpha}$, where $\tilde{\boldsymbol{\Phi}}$ represents the dual of dictionary $\boldsymbol{\Phi}$ [31].

3 Sparsity-based spectral unmixing

3.1 Solving the spectral unmixing problem

The minimization problem (7) is nonconvex, but it could be solved using a coordinate descent method [32]. Therefore, it could be divided into two convex problems, $\min_{\boldsymbol{\alpha}}(\cdot)|_A$ and $\min_{A}(\cdot)|_{\boldsymbol{\alpha}}$, and solved alternately until convergence to optimal values for both $\boldsymbol{\alpha}$ and A. The first of these two convex problems,

$$\min_{\boldsymbol{\alpha}} \frac{1}{2} \|\boldsymbol{\beta} - \boldsymbol{\alpha}A\|_2^2 + \lambda \|\boldsymbol{\alpha}\|_1 \tag{8}$$
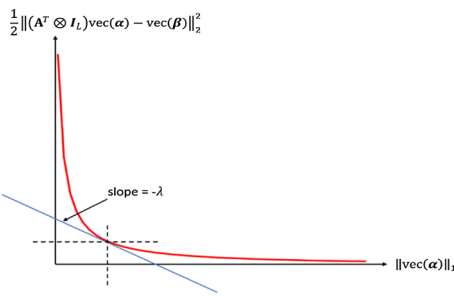
Fig. 1 Typical Pareto curve for an arbitrary vector *basis pursuit* problem

obtains an optimal value of $\boldsymbol{\alpha}$ by solving a matrix least squares minimization problem with an l_1-norm sparsity constraint on $\boldsymbol{\alpha}$, i.e., a matrix *basis pursuit* problem [24]. The second convex problem

$$\min_{A} \frac{1}{2}\|\boldsymbol{\beta} - \boldsymbol{\alpha}A\|_2^2 \tag{9}$$

obtains an optimal value of A by solving a matrix least squares minimization problem, under the constraints given by (2) [33].

3.2 Vectorizing of HSI multichannel data

To solve minimization problem (8), one would write the matrix linear system of equations, $\boldsymbol{\beta} = \boldsymbol{\alpha}A$, as a vector linear system of equations

$$\left(A^T \otimes I_L\right)\text{vec}(\boldsymbol{\alpha}) = \text{vec}(\boldsymbol{\beta}) \tag{10}$$

where $\text{vec}(\cdot)$ denotes the vectorization of a matrix obtained by stacking its columns on top of one another, $\otimes$ denotes Kronecker product, and I_L is an $L \times L$ identity matrix. Therefore, the matrix *basis pursuit* in (8) could be written in vectorized form as

$$\min_{\boldsymbol{\alpha}} \frac{1}{2}\left\|\left(A^T \otimes I_L\right)\text{vec}(\boldsymbol{\alpha}) - \text{vec}(\boldsymbol{\beta})\right\|_2^2 + \lambda\|\text{vec}(\boldsymbol{\alpha})\|_1 \tag{11}$$

On the other hand, solving minimization problem (9) in its matrix linear systems of equations form is straightforward.

3.3 Choice of regularization parameter λ value

Different solutions for the vector *basis pursuit* problem (11), corresponding to different values of the regularization parameter λ, could be obtained [26]. As shown by the Pareto curve in Fig. 1, different values of λ quantify the trade-off between the sparsity level of the unknown endmembers, $\|\text{vec}(\boldsymbol{\alpha})\|_1$, and the data fitting error resulting from using their estimates, $\frac{1}{2}\left\|\left(A^T \otimes I_L\right)\text{vec}(\boldsymbol{\alpha}) - \text{vec}(\boldsymbol{\beta})\right\|_2^2$.

From Fig. 1, we note that when the value of λ is high (low), and the sparsity of the endmembers would be high (low), i.e., $\|\text{vec}(\boldsymbol{\alpha})\|_1$ would be small (large) and the data fitting error, $\frac{1}{2}\left\|\left(A^T \otimes I_L\right)\text{vec}(\boldsymbol{\alpha}) - \text{vec}(\boldsymbol{\beta})\right\|_2^2$, would be high (low). Therefore, a traditional way to solve the vector *basis pursuit* problem (11) is to solve it many times using different values of λ, and then pick the solution that would successfully solve the spectral unmixing problem. This traditional way however is not practical as it could be both time and resource consuming.

3.4 Least angle regression (LARS)

Least angle regression (LARS) is a computationally efficient method to solve the vector *basis pursuit* problem (11) for all important values of the regularization parameter λ [34]. To avoid rewriting the cumbersome mathematical expressions in (11), we will explain the basic steps of LARS using a generic vector *basis pursuit* problem,

$$\min_{x} \frac{1}{2}\|Hx - b\|_2^2 + \lambda\|x\|_1 \tag{12}$$

which we will consider in this subsection only.

As the sub-differential of the objective function in (12) must contain the zero vector, the optimal solution $\tilde{x}_\lambda$ must satisfy the following optimality conditions [35]

$$H_I^T\left(H\tilde{x}_\lambda - b\right) = -\lambda z \tag{13}$$

$$\left\|H_{I^c}^T\left(H\tilde{x}_\lambda - b\right)\right\|_\infty \leq \lambda \tag{14}$$

where I denotes an active set, i.e., the indices where $\tilde{x}_\lambda$ is nonzero, I^c denotes the corresponding inactive set, z is the sign sequence of $\tilde{x}_\lambda$ on I, and H_I is a matrix whose columns are the columns of H whose indices comprise the active set I. For a given active set, using (11), $\tilde{x}_\lambda$ could be written as

$$\tilde{x}_\lambda = \begin{cases} \left(H_I^T H_I\right)^{-1}\left(H_I^T b - \lambda z\right), & \text{on } I \\ 0, & \text{otherwise} \end{cases} \tag{15}$$

LARS is a homotopy algorithm [36] that obtains an optimal solution $\tilde{x}_\lambda$ for every critical value of λ, i.e., values of λ where elements should be added or removed from the active set. LARS starts with a very high value of λ, an empty active set and a zero vector for $\tilde{x}_\lambda$. As λ is decreased, using (15), the update direction that keeps $\tilde{x}_\lambda$ optimal will be

$$d = \begin{cases} \left(H_I^T H_I\right)^{-1}z & \text{on } I \\ 0, & \text{otherwise} \end{cases} \tag{16}$$

Thus, the optimal solution $\tilde{x}_\lambda$ is moved in direction d, until a constraint in either (13) or (14) is violated. In this

case, either an additional column of H must be added to H_I, thereby a nonzero value would be added to $\tilde{x}_\lambda$, or a column of H_I must be removed, thereby a nonzero value of $\tilde{x}_\lambda$ would be set to zero.

The smallest step size that would violate one of these constraints is given by $\delta^* = \min(\delta^+, \delta^-)$, where

$$\delta^+ = \min_{i \in I^c}^{+} \left\{ \frac{\lambda - p(i)}{1 + q(i)}, -\frac{\lambda + p(i)}{1 - q(i)} \right\} \tag{17}$$

$$\delta^- = \min_{i \in I}^{+} \left\{ -\frac{\tilde{x}_\lambda(i)}{d(i)} \right\} \tag{18}$$

$p = H_{I^c}^T (H\tilde{x}_\lambda - b)$ and $q = H_{I^c}^T H d$. We note that min + denotes the minimum of positive components only.

At every step of this homotopy algorithm, the new critical value of λ would become $\lambda - \delta^*$, the new optimal solution $\tilde{x}_\lambda$ would become $\tilde{x}_\lambda + \delta^* d$, and both active set I and sign sequence z would be updated accordingly. This procedure is repeated until λ has been lowered to a required minimum value [35].

3.5 Limitations of LARS for HSI large multichannel data

In the optimization problem (11), we note that vec(β) has dimensions $(L * N) \times 1$, vec(α) has dimensions $(L * M) \times 1$, and $A^T \otimes I_L$ has dimensions $(L * N) \times (L * M)$. Therefore, even relatively small regions of interest in hyperspectral data cubes, e.g., $N = 75 \times 75$ spectral pixels, $L = 512$ wavelengths, and $M = 3$ endmembers, would result in vec(β) with dimensions $(2.88 \times 10^6) \times 1$, vec($\alpha$) with dimensions $(1.54 \times 10^3) \times 1$ and $A^T \otimes I_L$ with dimensions $(2.88 \times 10^6) \times (1.54 \times 10^3)$. Such extremely large arrays ($\sim 10^9$ elements) would be very challenging to process and would require very large computer storage ($\sim 10^{10}$ bytes).

4 Kronecker least angle regression (K-LARS)

In this section, we use properties of Kronecker products to extend the standard LARS algorithm to solve the *basis pursuit* in (11) without actually constructing, storing or processing very large arrays that would typically result from Kronecker products. This would allow practical and efficient recovery of sparse multicomponent signals from multichannel data, i.e., without constructing or processing very large arrays. We call this extension of standard LARS the Kronecker LARS (K-LARS) algorithm.

4.1 Steps of our K-LARS algorithm

Initial steps:

- Divide all elements of β by $\sqrt{\text{trace}(\beta^T \beta)}$. This is equivalent to normalizing vec(β) to have unit l_2-norm, i.e., $\|\text{vec}(\beta)\|_2$. Subtract the mean of all elements of β from all the elements of β. A simple post-processing step could easily offset this normalization and centering of β.
- Normalize all columns of matrix A^T to have unit l_2-norm. A simple post-processing step could easily offset this normalization.
- Set all elements of matrix α to zero.
- Define an initial residual matrix R_0 and set it equal to β.
- Define an $L \times M$ active set matrix Z_0, where the indices of its nonzero elements would specify active columns selected from $(A^T \otimes I_L)$, and set all its elements to zero.
- Obtain correlations between all columns of $(A^T \otimes I_L)$ and vec(R_0) as $C_1 = R_0 A^T$. Select the column that is most correlated with the residual R_0, i.e.,

$$(i_1, j_1) = \max_{\substack{1 \le i \le L \\ 1 \le j \le M}} |C_1(i, j)| \tag{19}$$

- Generate an $L \times M$ mask matrix, $K^{i,j} = (I_L)_{:,i} \times (I_M)_{j,:}$, whose elements are all zeros, except a single element of unit value at (i, j). Therefore, any column of $(A^T \otimes I_L)$ specified by (i, j) would be equal to vec($K^{i,j} A$).
- Update the active set matrix, $Z_1 = Z_0 + K^{i_1, j_1}$, to indicate that the column of $(A^T \otimes I_L)$ specified by (i_1, j_1) now belongs to the active set.

For each iteration, $t = 1, 2 \ldots$

- Obtain correlations between all columns of $(A^T \otimes I_L)$ and R_{t-1} as $C_t = R_{t-1} A^T$.
- Set λ_t to the maximum value of $|C_t(i, j)|$.
- Obtain the Gram matrix G of the active columns of $(A^T \otimes I_L)$ corresponding to all nonzero entries of Z_t specified by (i, j). Each element $G(m, n)$ is given by the inner product of the mth and nth active columns specified by (i_m, j_m) and (i_n, j_n), respectively,

$$G_t(m, n) = \text{trace}\left(A^T \left(K^{i_m, j_m} \right)^T K^{i_n, j_n} A \right). \tag{20}$$

- Obtain a vector, q_t, indexed by k where (i_k, j_k) corresponds to all nonzero elements of Z_t, and whose length is equal to the number of all active columns of $(A^T \otimes I_L)$. Its elements are the inner products between these active columns and vec(R_t),

$$q_t(k) = \text{trace}\left(A^T \left(K^{i_k, j_k}\right)^T R_t\right).$$

- Obtain the nonzero elements of the matrix $D_t(i, j)$ that represents the update direction of the solution α, as the vector

$$u_t = \frac{1}{\lambda_t} G_t^{-1} q_t. \tag{21}$$

Using (i_k, j_k) update the matrix $D_t(i, j)$ accordingly.

- For all (i, j) where $Z_t(i, j) = 0$, i.e., indices of the inactive set, obtain the positive step size δ_t^+ using

$$\delta_t^+ = \min_{(i,j)} \left\{ \frac{\lambda_t - C_t(i, j)}{1 - V_t(i, j)}, \frac{\lambda_t + C_t(i, j)}{1 + V_t(i, j)} \right\} \tag{22}$$

where $V_t = D_t A A^T$. Set $(i_{\text{add}}, j_{\text{add}})$ to the indices corresponding to δ_t^+ in (22).

- For all (i, j) where $Z_t(i, j) = 1$, i.e., indices of the active set, obtain the negative step size $\boldsymbol{\delta_t^-}$ using

$$\delta_t^- = \min_{(i,j)}^+ \left\{ \frac{\alpha_{t-1}(i, j)}{D_t(i, j)} \right\} \tag{23}$$

Set $(i_{\text{remove}}, j_{\text{remove}})$ to the indices corresponding to δ_t^- in (23).

- Obtain the step size δ_t^* using

$$\delta_t^* = \min\left\{\delta_t^+, \delta_t^-\right\} \tag{24}$$

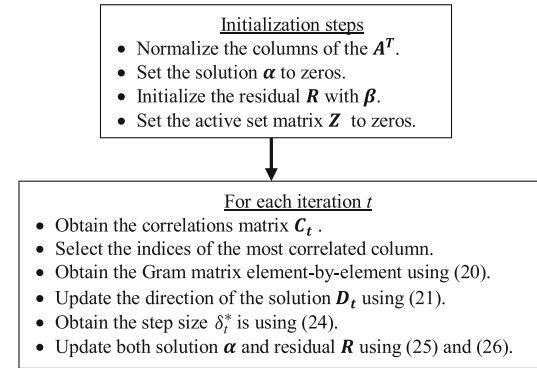- Update both solution matrix α and residual matrix using

$$\alpha_t = \alpha_{t-1} + \delta_t^* D_t \tag{25}$$

$$R_t = R_{t-1} - \delta_t^* D_t A \tag{26}$$

- If $\delta_t^* = \delta_t^+$, set $Z_{t+1} = Z_t + K^{i_{\text{add}}, j_{\text{add}}}$, to indicate that the column of $\left(A^T \otimes I_L\right)$ specified by $(i_{\text{add}}, j_{\text{add}})$ now belongs to the active set.
- Else if $\delta_t^* = \delta_t^-$, set $Z_{t+1} = Z_t - K^{i_{\text{remove}}, j_{\text{remove}}}$, to indicate that the column of $\left(A^T \otimes I_L\right)$ specified by $(i_{\text{remove}}, j_{\text{remove}})$ now belongs to the inactive set.

Continue iterations until all columns of $\left(A^T \otimes I_L\right)$ are included in the active set, all values of the correlation matrix C become equal, λ reaches its minimum value, and all entries of the active set matrix Z are ones. This is equivalent to reaching the full least squares solution. However, typically one would terminate the iterations when the desired sparsity level is reached or when the norm of the residual becomes within a prescribed tolerance. The K-LARS algorithm steps can be summarized in the following steps.

K-LARS algorithm steps:

Initialization steps
- Normalize the columns of the A^T.
- Set the solution α to zeros.
- Initialize the residual R with β.
- Set the active set matrix Z to zeros.

For each iteration t
- Obtain the correlations matrix C_t.
- Select the indices of the most correlated column.
- Obtain the Gram matrix element-by-element using (20).
- Update the direction of the solution D_t using (21).
- Obtain the step size δ_t^* is using (24).
- Update both solution α and residual R using (25) and (26).

5 Spectral unmixing results

To demonstrate the validity of our K-LARS-based spectral unmixing method, we applied it on two HSI datasets. The first is a synthetic dataset with spectral pixels that are linear mixtures of endmembers, from the *ASTER* spectral library [37], with uniformly random abundances. The second is an actual HSI dataset from *NASA's AVIRIS* website [38]. We also compare our results to ones obtained using GMCA. We show that these two results are similar, albeit our results were obtained without arbitrary choices related to specifying the regularization parameter.

5.1 Preprocessing of HSI multichannel data

Assuming the presence of white noise in the HSI data, then a data denoising step would be necessary before solving optimization problem (7). This is because white noise could not be sparsely represented in any dictionary [39]. Therefore, in this paper, we denoised each raw spectral pixel using a wavelet (*Symmlet*) thresholding method with a universal threshold [40].

As discussed in Sect. 2.4, to promote endmember sparsity, we transformed the used HSI datasets to the wavelet domain, where the transformed endmembers α would likely be more sparse than in the original measurement domain. We used the *Coiflet 5* wavelet, because both its scaling and wavelet functions have vanishing moments, thereby significantly promoting sparsity of piecewise smooth functions, e.g., endmember spectra.

5.2 Applying K-LARS to HSI spectral unmixing

As discussed in Sect. 3.1, the sparsity-based spectral unmixing problem, problem (7), could be solved by an iterative and alternate estimation of α, problem (8), and A, problem (9).
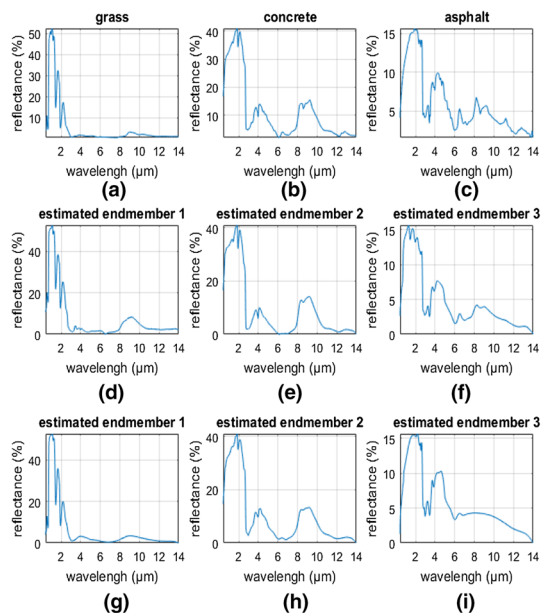
Fig. 2 **a–c** Exact spectra of grass, concrete, and asphalt, respectively. **d–f** Estimated spectra obtained using GMCA-based spectral unmixing algorithm. **g–i** Estimated spectra obtained using our K-LARS-based spectral unmixing algorithm

Initially, α is set to zero and A is set to random abundance values. At each iteration, we fixed A and estimated α using our K-LARS algorithm, then we fixed α and estimated A by solving a simple constrained linear least squares problem.

The sparsity levels of any of the estimates of α are crucial to the accuracy of our final spectral unmixing results. Therefore, in our first spectral unmixing iteration, we terminated the K-LARS algorithm when the data fitting error corresponding to the sparse solution was within 0.5% of the data fitting error corresponding to the full least squares solution. We then used this sparsity level of α as the K-LARS termination condition for all subsequent spectral unmixing iterations.

5.3 Spectral unmixing of synthetic HSI multichannel data

We generated a synthetic HSI dataset of 3000 spectral pixels. These pixels are linear mixtures of the spectra of grass, concrete and asphalt, obtained from the *ASTER* spectral library [36], with uniformly random abundances. Additive Gaussian noise was added to these spectral pixels to obtain a signal-to-noise ratio (SNR) of 30 dB. Therefore, in this dataset, the number of endmembers, $M = 3$, the number of spectral pixels, $N = 3000$ and the number of wavelengths, $L = 491$ (0.42–14 µm). Figure 2a–c shows the spectra of grass, concrete and asphalt, whose different linear mixtures make up different spectral pixels in this dataset.

Figure 2d–f shows the estimated spectra obtained by applying the GMCA-based spectral unmixing algorithm to this dataset. To obtain these GMCA-based results, we used trial and error that consumed more computational time and effort to obtain both initial and final values of the regularization parameter, $\lambda_{initial} = 15$, $\lambda_{final} = 4$, and we arbitrarily chose a linear schedule for its reduction with each coordinate descent iteration. The correlation coefficients between the exact spectra of grass, concrete, asphalt and their GMCA-estimated counterparts are 0.9673, 0.9942 and 0.9583, respectively.
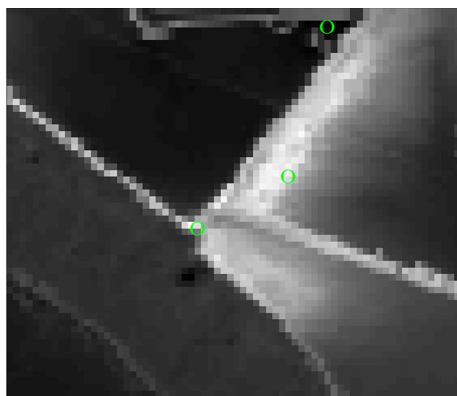
Figure 2g–i shows the estimated spectra obtained by applying our K-LARS-based spectral unmixing algorithm to this dataset. The correlation coefficients between the exact spectra of grass, concrete, asphalt and their K-LARS-estimated counterparts are 0.9970, 0.9993 and 0.9727, respectively. These quantitative results demonstrate the validity of our new K-LARS algorithm and its successful application to HSI spectral unmixing. Compared to spectral unmixing results obtained by GMCA, our results obtained by K-LARS are very close, albeit they were obtained without any trial and error, or arbitrary choices, in specifying the regularization parameter.

### 5.4 Spectral unmixing of *AVIRIS* HSI multichannel data

We downloaded an actual HSI dataset called the Moffet Field from *NASA's AVIRIS* website [37]. This dataset is comprised of images obtained using 224 wavelengths (0.365–2.497 µm). We selected an arbitrary region (70×60 pixels) as our region of interest, and then used Google Maps to estimate that our selected region had three endmembers: water, grass and land. Therefore, in the dataset corresponding to our selected region, the number of endmembers, $M = 3$, the number of spectral pixels, $N = 4200$ and the number of wavelengths, $L = 224$.

To validate spectral unmixing results obtained by applying our K-LARS-based unmixing algorithm to this dataset, we need to know its ground truth, i.e., spectra of endmembers present in our selected region. For this purpose, we used N-FINDR, a software package that uses a geometrical approach to find approximately pure spectral pixels, i.e., spectral pixels comprised of a single endmember only, in the given HSI data. The locations of these approximately pure spectral pixels in our selected region obtained by N-FINDR [41] are shown in Fig. 3.

Figure 4a–c shows the ground truth spectra of the three endmembers present in our selected region, and whose different linear mixtures make up different spectral pixels in

Fig. 3 Locations of approximately pure spectral pixels obtained by N-FINDR
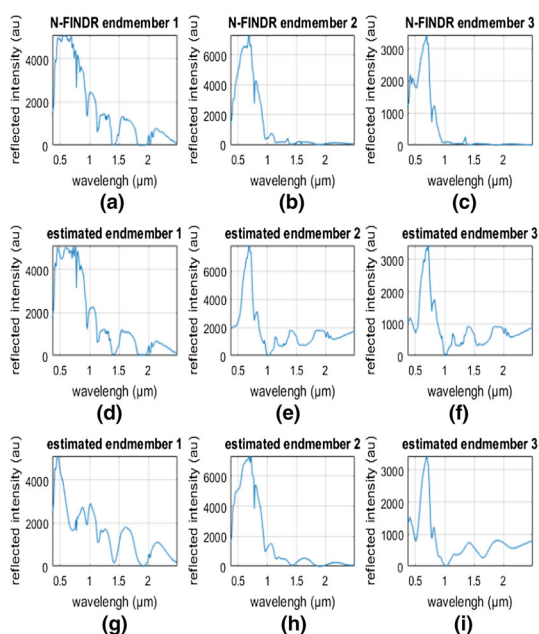
Fig. 4 a–c Ground truth spectra of three endmembers obtained by N-FINDR. **d–f** Estimated spectra obtained using GMCA-based spectral unmixing algorithm. **g–i** Estimated spectra obtained using our K-LARS-based spectral unmixing algorithm

this dataset. Figure 4d–f shows the estimated spectra of these three endmembers obtained by applying the GMCA-based spectral unmixing algorithm to this dataset. To obtain these GMCA-based results, we used trial and error that consumed more computational time and effort to obtain both initial and final values of the regularization parameter, $\lambda_{\text{initial}} = 15$, $\lambda_{\text{final}} = 2.8$, and we arbitrarily chose a linear schedule for its reduction with each coordinate descent iteration. The correlations between the ground truth spectra and their corresponding estimated endmember spectra are 0.9923, 0.8514 and 0.8607, respectively. Figure 4g–i shows the estimated spectra of these three endmembers obtained by

applying our K-LARS-based spectral unmixing algorithm to this dataset. The correlations between the ground truth spectra and their corresponding estimated endmember spectra are 0.8358, 0.9892 and 0.8878, respectively. These quantitative results further demonstrate the validity of our new K-LARS algorithm and its successful application to HSI spectral unmixing. Compared to spectral unmixing results obtained by GMCA, our results obtained by K-LARS are very close, albeit they were obtained without any trial and error, or arbitrary choices, in specifying the regularization parameter.

6 Discussion and conclusions

A recent approach to solve the unsupervised spectral unmixing of HSI data requires that these endmembers be sparse in some dictionary. This requires the solution of a sparse signal recovery optimization problem using *basis pursuit* that requires the specification of a data-dependent regularization parameter. LARS is a recent method for sparse signal recovery that efficiently recovers all sparse signals corresponding to all relevant regularization parameter values. However, the application of LARS to large multichannel data could be very challenging in practice, due to the need for generation and storage of extremely large arrays ($\sim 10^{10}$ bytes in a relatively small spectral unmixing problem). In this paper, we extended the standard LARS algorithm, using Kronecker products, to achieve practical and efficient recovery of sparse multicomponent signals from multichannel data, i.e., without the need to construct and process very large arrays or the need for trial and error to specify regularization parameters. We then applied our new K-LARS algorithm to successfully achieve unsupervised spectral unmixing of both synthetic and *AVIRIS* hyperspectral imaging data. Our quantitative spectral unmixing results for both synthetic HSI multichannel data (Sect. 5.3) and *AVIRIS* HSI multichannel data (Sect. 5.4) demonstrate the validity of our new K-LARS algorithm and its successful application to HSI spectral unmixing. Compared to spectral unmixing results obtained by GMCA, our results obtained by K-LARS are very close, albeit they were obtained without any trial and error, or arbitrary choices, in specifying the regularization parameter. This lack of trial and error that could consume significant computational time and effort is the main advantage of using K-LARS, instead of GMAC, for sparse spectral unmixing. More important, our K-LARS algorithm could be a very valuable research tool to the signal processing community, where it could be used to solve sparse least squares problems involving large multichannel data. On the other hand, K-LARS has a disadvantage of being an intricate algorithm that would require careful programming. As future work, we are currently generalizing our K-LARS algorithm

that we introduced in this paper to efficiently represent and recover multidimensional signals using Kronecker dictionaries.

References

1. Manolakis, D., Marden, D., Shaw, G.A.: Hyperspectral image processing for automatic target detection applications. Linc. Lab. J. **14**(1), 79–116 (2003)
2. Manolakis, D., et al.: Detection algorithms in hyperspectral imaging systems: an overview of practical algorithms. IEEE Signal Process. Mag. **31**(1), 24–33 (2014)
3. Nasrabadi, N.M.: Hyperspectral target detection: an overview of current and future challenges. IEEE Signal Process. Mag. **31**(1), 34–44 (2014)
4. Bioucas-Dias, J.M., et al.: Hyperspectral unmixing overview: geometrical, statistical, and sparse regression-based approaches. IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens. **5**(2), 354–379 (2012)
5. Rajan, S., Ghosh, J., Crawford, M.M.: An active learning approach to hyperspectral data classification. IEEE Trans. Geosci. Remote Sens. **46**(4), 1231–1242 (2008)
6. Keshava, N.: Distance metrics and band selection in hyperspectral processing with applications to material identification and spectral libraries. IEEE Trans. Geosci. Remote Sens. **42**(7), 1552–1565 (2004)
7. Parente, M., Plaza, A.: Survey of geometric and statistical unmixing algorithms for hyperspectral images. In: 2010 2nd Workshop on Hyperspectral Image and Signal Processing: Evolution in Remote Sensing (WHISPERS), pp. 1–4 (2010)
8. Dobigeon, N., et al.: Bayesian separation of spectral sources under non-negativity and full additivity constraints. Signal Process **89**(12), 2657–2669 (2009)
9. Amiri, F., Kahaei, M.: A sparsity-based Bayesian approach for hyperspectral unmixing using normal compositional model. SIViP **12**(7), 1361–1367 (2018)
10. Shi, C., Wang, L.: Incorporating spatial information in spectral unmixing: a review. Remote Sens. Environ. **149**, 70–87 (2014)
11. Iordache, M., Plaza, A., Bioucas-Dias, J.: On the use of spectral libraries to perform sparse unmixing of hyperspectral data. In: 2010 2nd Workshop on Hyperspectral Image and Signal Processing: Evolution in Remote Sensing (WHISPERS), pp. 1–4
12. Iordache, M., Bioucas-Dias, J.M., Plaza, A.: Sparse unmixing of hyperspectral data. IEEE Trans. Geosci. Remote Sens. **49**(6), 2014–2039 (2011)
13. Keshava, N., Mustard, J.F.: Spectral unmixing. IEEE Signal Process. Mag. **19**(1), 44–57 (2002)
14. Yang, Z., et al.: Blind spectral unmixing based on sparse nonnegative matrix factorization. IEEE Trans. Image Process. **20**(4), 1112–1125 (2011)
15. Qian, Y., et al.: Hyperspectral unmixing via l_1-sparsity-constrained nonnegative matrix factorization. IEEE Trans. Geosci. Remote Sens. **49**(11), 4282–4297 (2011)
16. Wu, Z., et al.: Sparse non-negative matrix factorization on GPUs for hyperspectral unmixing. IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens. **7**(8), 3640–3649 (2014)
17. Tong, L., et al.: Dual graph regularized NMF for hyperspectral unmixing. In: 2014 International Conference on Digital Lmage Computing: Techniques and Applications (DlCTA), pp. 1–8 (2014)
18. Wang, W., Qian, Y., Tang, Y.Y.: Hypergraph-regularized sparse NMF for hyperspectral unmixing. IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens. **9**(2), 681–694 (2016)
19. Li, X., Cui, J., Zhao, L.: Blind nonlinear hyperspectral unmixing based on constrained kernel nonnegative matrix factorization. Signal Image Video Process. **8**(8), 1555 (2014)
20. Bobin, J., et al.: Sparsity and morphological diversity in blind source separation. IEEE Trans. Image Process. **16**(11), 2662–2674 (2007)
21. Rapin, J., et al.: Sparse and non-negative BSS for noisy data. IEEE Trans. Signal Process. **61**(22), 5620–5632 (2013)
22. Chenot, C., Bobin, J., Rapin, J.: Robust sparse blind source separation. IEEE Signal Process. Lett. **22**(11), 2172–2176 (2015)
23. Bioucas-Dias, J.M., Figueiredo, M.A.: Alternating direction algorithms for constrained sparse regression: application to hyperspectral unmixing. In: 2010 2nd workshop on hyperspectral image and signal processing: evolution in remote sensing (WHISPERS), pp. 1–4 (2010)
24. Chen, S.S., Donoho, D.L., Saunders, M.A.: Atomic decomposition by basis pursuit. SIAM Rev. **43**(1), 129–159 (2001)
25. Beck, A., Teboulle, M.: A fast iterative shrinkage-thresholding algorithm for linear inverse problems. SIAM J. Imaging Sci. **2**(1), 183–202 (2009)
26. Rish, I., Grabarnik, G.: Sparse Modeling: Theory, Algorithms, and Applications. CRC Press, Boca Raton (2014)
27. Caiafa, C.F., Cichocki, A.: Computing sparse representations of multidimensional signals using kronecker bases. Neural Comput. **25**(1), 186–220 (2013)
28. Bernstein, D.S.: Matrix Mathematics: Theory, Facts, and Formulas with Application to Linear Systems Theory. Princeton University Press, Princeton (2005)
29. Comon, P., Jutten, C., Herault, J.: Blind separation of sources, Part II: problems statement. Signal Process **24**(1), 11–20 (1991)
30. Klement, E.P., Mesiar, R., Pap, E.: Triangular norms. Position paper I: basic analytical and algebraic properties. Fuzzy Sets Syst. **143**(1), 5–26 (2004)
31. Mallat, S.: A Wavelet Tour of Signal Processing. Academic press, Cambridge (1999)
32. Li, Y., Osher, S.: Coordinate descent optimization for l_1-minimization with application to compressed sensing; a greedy algorithm. Inverse Probl. Imaging **3**(3), 487–503 (2009)
33. Lawson, C.L., Hanson, R.J.: Solving Least Squares Problems. SIAM, Philadelphia (1995)
34. Efron, B., et al.: Least angle regression. Ann. Stat. **32**(2), 407–499 (2004)
35. Asif, M.S., Romberg, J.: Dynamic updating for l_1-minimization. IEEE J. Sel. Top. Signal Process. **4**(2), 421–434 (2010)
36. Donoho, D.L., Tsaig, Y.: Fast Solution of l_1-norm minimization problems when the solution may be sparse. IEEE Trans. Inf. Theory **54**(11), 4789–4812 (2008)
37. ASTER Spectral Library, http://speclib.jpl.nasa.gov. Accessed 16 Sept 2018
38. AVIRIS NASA Website, http://aviris.jpl.nasa.gov/alt_locator/. Accessed 16 Sept 2018
39. Elad, M., Aharon, M.: Image denoising via sparse and redundant representations over learned dictionaries. IEEE Trans. Image Process. **15**(12), 3736–3745 (2006)
40. Dewangan, N., Goswami, A.D.: Image denoising using wavelet thresholding methods. Int. J. Eng. Sci. Manag. **2**(2), 271–275 (2012)
41. Winter, M.E.: N-FINDR: an algorithm for fast autonomous spectral end-member determination in hyperspectral data. In: SPIE's International Symposium on Optical Science, Engineering, and Instrumentation, pp. 266–275 (1999)