American Journal of Signal Processing 2018, 8(1): 1-8
DOI: 10.5923/j.ajsp.20180801.01

Performance Evaluation of Various Hyperspectral Nonlinear Unmixing Algorithms

Awabed Jibreen[1,*], Nourah Alqahtani[2], Ouiem Bchir[1]

[1]Department of Computer Science, College of Computer and Information Sciences, King Saud University, Riyadh, Saudi Arabia
[2]Department of Information Systems, College of Computer and Information Sciences, King Saud University, Riyadh, Saudi Arabia

Abstract Nonlinear unmixing of hyperspectral images has shown considerable attention in image and signal processing research areas. Hyperspectral unmixing identifies endmembers spectral signatures and the abundance fractions of each endmemeber within each pixel of an observed hyperspectral scene. Over the last few years, several nonlinear unmixing algorithms have been proposed. This paper presents an empirical comparison of several popular and recent algorithms of supervised nonlinear unmixing algorithms. Namely, we compared Kernel-based algorithms, graph Laplacian regularization algorithm, and nonlinear unmixing algorithm using a generalized bilinear model (GBM). These unmixing algorithms estimate the abundances of linear, bilinear and intimate mixtures of hyperspectral data. We assessed the performance of these algorithms using Root Mean SquareError on the same data sets.

Keywords Hyperspectral imagery, Unmixing, Linear model, Nonlinear model, Graph Laplacian regularization, GBM

1. Introduction

Hyperspectral image is a three-dimensional data cube which consists of one spectral and two spatial dimensions. Each pixel in the hyperspectral image is represented by a vector of reflectance values (also known as the pixel's spectrum) whose length is equal to the number of spectral bands considered. Thus, the pixel's spectrum corresponding to a sole material (such as soil, vegetation, or water) characterizes the material and it is called endmember.

Hyperspectral sensors generate high spectral resolution images, but they have a low spatial resolution, which causes mixed pixels within hyperspectral images. As another cause for mixed pixels is a homogeneous combination of different materials in one pixel. Therefore, the hyperspectral spectrum can be seen a mixture of the spectra of each component in the observed scene. This has lead up to linear mixing model and nonlinear mixing model.

In fact, the mixing model related to spectral unmixing imagery can be either linear or nonlinear, and that depends on the nature of the observed hyperspectral image. As the photons hit the detector coming from the source, the imaging system bins the observed photons according to their spatial location and wavelength [27]. Linear mixtures are used in case that the detected photons interact mainly with a single component of the observed scene before they reach the sensor. On the contrary, nonlinear mixture models are used when the photons interact with multiple components [4].

There are two types of nonlinear mixing: intimate mixing and bilinear mixing. In bilinear mixing, effects of multiple light scattering occur, i.e. the solar radiation scattered by a given material reflects off other materials prior reaching the sensor. In intimate nonlinear mixing, interactions occur at a microscopic level and the photons interact with all the materials concurrently as they are scattered [5, 6].

The mixture problem can be solved by applying an appropriate unmixing process. Hyperspectral unmixing is an important process in many fields such as agriculture, geography and geology. It has different applications such as surveillance applications, earth surfaces analysis application, pollution monitoring applications. Spectral unmixing is widely used for analyzing hyperspectral data. It includes two main steps. The first step is determining the pure components in the hyperspectral image (Endmembers). The second step is finding out these materials' abundances. Spectral unmixing can be either supervised unmixing or unsupervised unmixing. In supervised unmixing, the number of endmember is known while in unsupervised unmixing the number of endmemeber is unknown. Depending on the mixing type, the unimixing processes could be linear and nonlinear.

Linear spectral unmixing is to determine the relative proportion (abundance) of materials that are presented in hyperspectral imagery based on the spectral characteristics of materials. The reflectance at each pixel is assumed to be a linear combination of the reflectance of each material (or

* Corresponding author:
ajibreen@ksu.edu.sa (Awabed Jibreen)

Published online at http://journal.sapub.org/ajsp

endmember) existing in the pixel. Linear unmixing methods are used only with the linear mixing models [1].

Linear unmixing models cannot handle nonlinear mixing pixels. There are many algorithms and researches regarding linear unmixing, which assumes that pixels are linearly mixed by material signatures weighted by abundances.

Recently, nonlinear unmixing for hyperspectral images is receiving attention in remote sensing image exploitation [6] [20] [4] and [18]. Alternative approximation approaches have been proposed for handling the effects of nonlinearity leading to utilizing physics-based nonlinear mixing models [1]. The bilinear mixture model (BMM), has been studied in several researches which is used with second-order scattering of photons between two different materials [8].

In this paper, we consider a set of latest and well-known algorithms of supervised nonlinear unmixing of hyperspectral images. We compare empirically these algorithms using the same set of data. The performance is assessed in terms of accuracy.

Namely, we compare kernel-based algorithms [6], regularization algorithms [20], and nonlinear unmixing using a generalized bilinear model (GBM) [4]. These unmixing algorithms estimate the abundances of linear, bilinear and intimate mixtures of hyperspectral data.

More specifically, for Kernel-based methods we used the K-Hype [6] and Multiple Kernel Learning (SK-Hype) [6] algorithms. For Graph Laplacian Regularization approaches, we considered GLUP-Lap [20] (Group Lasso with Unit sum, Positivity constraints and graph Laplacian regularization) [20].

The outline of the rest of the paper is as follows: The state-of-the-art literature of different general approaches used for solving the hyperspectral unmixing problem is given in section 2. A description of well-known nonlinear unmixing algorithms is provided in section 3. Experimental results and discussion are highlighted in section 4. Conclusion is reported in section 5.

2. Nonlinear Unmixing Approaches

Yoann Altmann et. al. [19] proposed a nonlinear unmixing method which is based on the Gaussian process. In [19], the abundances of all pixels are identified first, and then the endmembers are estimated using Gaussian regression. They consider a kernel-based method for unsupervised spectral unmixing based on the Gaussian latent variable model (GP-LVM) [29] which is a nonlinear dimension reduction method that has the ability to accurately model any nonlinearity.

Jie Chen et. al. [6] addressed the abundances estimation problem of the nonlinear mixed hyperspectral data. They propose a solution to an appropriate kernel-based regression problem. They propose the K-Hype mixture model which is a kernel-based hyperspectral mixture model. They also suggest associated abundance extraction algorithms. The disadvantage of k-Hype is that the balance between linear and nonlinear interactions is fixed. To manage this limitation they propose SK-Hype, which is a natural generalization of K-Hype. It depends on the Multiple Kernel Learning concept. Also, it can automatically adapt between linear and nonlinear contributions.

Rita Ammanouil et. al. [20] proposed a graph Laplacian regularization in the hyperspectral image unmixing. The proposed method depends on the construction of a graph representation of the hyperspectral image. Similar pixels are connected by edges both spectrally and spatially. Convex optimization problem is solved using the Alternating Direction Method of Multipliers (ADMM). Graph-cut methods are proposed in order to reduce the computational burden.

Yoann Altmann et. al. [18] suggested a Bayesian and two least squares optimization algorithms for nonlinear unmixing. These algorithms assumed that the pixels are mixed by a polynomial post-nonlinear mixing model. They also proposed in [4] a generalized bilinear Method (GBM) where a Bayesian algorithm is proposed to estimate the nonlinearity coefficients and the abundances values.

Chang Li et. al. [30] propose a general sparse unmixing method (SU-NLE) that depends on the estimation of noise level. They used the weighted approach in order to provide the matrix of the noise weighting. The proposed approach [30] is robust for different noise levels in different bands of the hyper spectral image.

Xiaoguang Mei et. al. [31] propose two methods for nonlinear unmixing of HIS. Robust GBM (RGBM) [32] and another new unmixing method with superpixel segmentation (SS) and low-rank representation (LRR) unmixing approach.

In this paper, we empirically compare four nonlinear unmixing algorithms. Namely, we assess the performance of the Kernel-Based Hyperspectral Unmixing Algorithms: the K-Hype Algorithm [6], the Super K-Hype (SK-Hype) Algorithm [6], the Group Lasso with Unit sum, Positivity constraints and graph Laplacian regularization (GLUP-Lap) [20], and the Generalized Bilinear Model (GBM) applied [4].

2.1. The K-Hype Algorithm

The K-Hype [6] is designed for both linear and nonlinear mixing models. It is based on the model defined in (1).

$$\mathrm{K_{nlin}}\left(\mathrm{m}_{\lambda_\ell}, \mathrm{m}_{\lambda_\mathrm{p}}\right) = (\,1 + \tfrac{1}{\mathrm{R}^2}(\mathrm{m}_{\lambda_\ell} - 1/2)^{\mathrm{T}}(\mathrm{m}_{\lambda_\mathrm{p}} - 1/2)\,)^2 \tag{1}$$

where $\mathrm{K_{nlin}}$ is the polynomial kernel of degree 2, m is endmember spectra. R is the number of endmembers. $\mathrm{m_i}$ is an endmember spectral signature, $\mathrm{m}_{\lambda_\mathrm{l}}$ is the ℓ -th (1XR) row of M Endmember matrix. The constants $1/\mathrm{R}^2$and 1/2 serve the purpose of normalization. It optimized using quadratic programming.

Jie Chen et al. [6] define the function in equation (2) to extract the mixing ratios of the endmembers.

$$\psi^* = \; arg_{\psi\in\mathcal{H}} \min \tfrac{1}{2}||\psi||^2_{\mathcal{H}} + \tfrac{1}{2\mu}\textstyle\sum_{\ell=1}^{L}(r_\ell - \psi(m_{\lambda_\ell}))^2 \tag{2}$$

where $\mathcal{H}$ is a given functional space, μ is a small positive parameter that controls the trade-off between regularization. ψ is unknown nonlinear function that defines the interactions between the endmembers in matrix. r_ℓ is dominated by a linear function.

This function is defined by a linear trend parameterized by the abundance vector α , combined with a nonlinear fluctuation term:

$$\psi(m_{\lambda\ell}) = \alpha^T m_{\lambda_\ell} + \psi_{nlin}(m_{\lambda_\ell}) \qquad (3)$$

subjected to $\alpha \geq 0$, and 1^T $\alpha=1$.

where ψ_{nlin} can be any real-valued functions on a compact $\mathcal{M}$ of a reproducing kernel Hilbert space $\mathcal{H}_{nlin}$. The corresponding Gram matrix K is given by:

$$K = MM^T + K_{nlin} \qquad (4)$$

where $K_{nlin}$is the Gram matrix associated with the nonlinear map ψ_{nlin}, with (ℓ, ρ)-thentry $K_{nlin}(\boldsymbol{m}_{\lambda\ell}\ ,\boldsymbol{m}_{\lambda \mathrm{p}}\ )$

The abundance vector can be estimated as follows:

$$\alpha^* = M^T\beta^* + \gamma^* - \lambda^* 1 \qquad (5)$$

The K-Hype algorithm [6] is described in algorithm 1.

Algorithm 1 The K-Hype Algorithm [6]

```
1: Initialization.
2: Choose the kernel and the regularization constant
3: Calculate Gaussian kernel using
   $K_{(\boldsymbol{m}_{\lambda\ell}, \boldsymbol{m}_{\lambda \mathrm{p}})} = exp\left(\frac{[\![\boldsymbol{m}_{\lambda\ell}, \boldsymbol{m}_{\lambda\rho}]\!]}{2\sigma^2}\right)$
4: Extract the mixing ratios of the endmembers using the function in equation (2)
5: Solve the optimization problem using generic quadratic program (QP) solver.
6: Estimate the abudance using (5).
```

2.2. The SK-Hype Algorithm

The SK-Hype algorithm [6] is designed for both linear and nonlinear mixing models. The model in equation (3) has some limitations in that the balance between the linear component $\alpha^T m_{\lambda\ell}$ and the nonlinear component $\psi_{nlin}(m_{\lambda\ell}\ )$ cannot be tuned. As for K-Hype [6], the Gaussian kernel and the polynomial kernel were considered. Another difficulty in the model of equation (3) is that it cannot captures the dynamic of the mixture, which requires that r or the $m_{\lambda\ell}$ 'sbe locally normalized. Thus Gram matrice is considered as in equation (6)

$$K_u = \text{u } MM^T + (1\text{-u})\ K_{nlin} \qquad (6)$$

The abundance vector can be estimated as in (7).

$$\alpha^* = \frac{M^T\beta^* + \gamma^*}{1^T(M^T\beta^* + \gamma^*)} \qquad (7)$$

The SK-Hype algorithm [6] is described in Algorithm 2.

Algorithm 2 The SK-Hype Algorithm [6]

```
1: Initialization
2: Choose the kernel and the regularization constant.
3: Calculate the kernel matrix.
4: Repeat
5: Solve the optimization problem using generic QP solver.
6: Until stopping criteria is satisfied.
7: Estimate the abundance.
```

2.3. GLUP-Lap Algorithm

GLUP-Lap [20] stands for Group Lasse with Unit Sum, Positivity constraint and Laplacian regularization [20]. GLUP-Lap approach [20] is graph based. If two nodes are connected, then they are likely to have similar abundances. They incorporate this information in the unmixing problem using the graph Laplacian regularization. This leads to a convex optimization problem as defined in (8).

$$min_A \frac{1}{2} ||S - RA||_F^2 + \lambda \mathrm{tr}(A\mathcal{L}A^T) + \mu \sum_{k=1}^{N} ||a_k||_2 \quad (8)$$

Subject to $A_{ij} \geq 0, \forall i, j,\ \boldsymbol{\sum_{i=1}^{N} A_{ij}},\ \forall \boldsymbol{j}$

where $\mathcal{L}$ is the graph Laplacian matrix given by $\mathcal{L} = \mathrm{D} - \mathrm{W}$, D is diagonal matrix with $D_{ii} = \sum_{j=1}^{N} W_{ij}$, $\mu \geq 0$ and $\lambda \geq 0$ are two regularization parameters. The relevance of the regularization are expressed as in (9).

$$tr(A\mathcal{L}A^T) = \sum_{i=1}^{M} \sum_{j=1}^{N} \sum_{k\sim \mathrm{j}} W_{jk}(A_{ij} - A_{ik})^2 \qquad (9)$$

where k ~ j indicates that pixels j and k are similar and $W_{jk} \neq 0$, is the degree of similarity. The regularization parameter λ in (8) controls the extent to which similar pixels estimate similar abundances. R is a large dictionary of endmembers, and only few of these endmembers are present in the image.

Algorithm 3. The GLUP-Lap algorithm [20]

```
1: Generate W, the affinity matrix of size N x N.
If pixels i and j are similar then Wij is set to some positive value proportional to their degree of similarity as in (5).
Else Wij = zero.
2: Partition pixels into k clusters by applying ADMM Algorithm.
3: Solve GLUP-Lap for each cluster (8)
```

The first and second term of the cost function in (8) can be grouped in a single quadratic form. However the resulting Quadratic Problem has N × M non-separable variables. Its solution can be obtained using the Alternating Direction Method of Multipliers (ADMM) [21]. The GLUP-Lap algorithm steps are described in Algorithm 3.

2.4. GBM Algorithm

The GBM in [4] assumes that the mixture problem can be expressed as

$$r=M\,\alpha+\sum_{i=1}^{R-1}\sum_{j=i+1}^{R}\gamma_{ij}\alpha_i\alpha_j\,m_i\odot m_j+n \qquad (10)$$

under the following parameters constraints:

$$\alpha\ \alpha_k\geq 0,\ \forall\ k\in\{1,...,R\}\text{ and }\sum_{i=1}^{R-1}\alpha_k=1 \qquad (11)$$

$0\leq\gamma_{i,j}\leq 1\ \forall\ i\in\{1,\ldots,R-1\}\ \forall\ j\in\{i+1,\ldots,R\}$.

Where $\gamma_{i,j}$ is a coefficient that determines the interactions between endmembers #i and #j in the observed pixel. The unknown parameter vector θ that associated with the GBM [4] includes the nonlinearity coefficient vector $\boldsymbol{\gamma}=[\gamma_{1,2},\ldots,\gamma_{R-1,R}]^T$, the abundance vector α, and the noise variance σ^2.

Algorithm 4 The GBM algorithm [4]

```
1: Initialization
2: Initialize parameters
        $k^*=R,\sigma^{2(0)},\alpha_k^{(0)},\forall k\neq k^8$
3: $\alpha_{K^*}^{(0)}=1-\sum_{i=1}^{R-1}\alpha_i^{(0)}$
4: Sample Generation
5: for t=1: $N_{bi}+N_r$ do
8: sample $\alpha_k^{(t)}\sim f(\alpha_k\,|\sigma^{2(t-1)},y,\gamma,\alpha_{\{k,k^*\}}^{(t)}$ using Algorithm5
6: choose a random K* from {1,..R}
7: for k=1: R and k ≠ K* do
9: end for
10: $\alpha_{k^*}^{(t)}=1-\sum_{i\neq k^*}\alpha_i^{(t)}$
11: Sample $\gamma_{1,2}\sim f(\gamma_{1,2}|\sigma^{2(t-1)},y,\gamma_{(1,2)}^{(t)},\alpha^{(t)})$
 .. .. …
 .. .. …
12: .. .. …
13: sample
        $\gamma_{R-1,R}\sim f(\gamma_{R-1,R}\,|\sigma^{2(t-1)},y,\gamma_{(R-1,R)}^{(t)},\alpha^{(t)})$
14: sample $\sigma^{2(t)}\sim f(\sigma^2|y,\gamma^{(t)},\alpha^{(t)})$
15: end for
```

The hierarchical Bayesian model is used to calculate the unknown parameter vector $\theta=(\alpha^T,\boldsymbol{\gamma}^T,\sigma^2)^T$ that is associated with the GBM [4]. Metropolis-within-Gibbs algorithm is also used in order to generate sample distribution according to the posterior distribution f(θ|y). The generated samples are then used to estimate the unknown parameters. Metropolis-within-Gibbs algorithm is described in algorithm 5.

Algorithm 5 The Metropolis-within-Gibbs algorithm [4]

```
Sampling $\boldsymbol{\alpha}_k$ with “Metropolis-within-Gibbs”
1: Draw candidate $\boldsymbol{\alpha}_k^{(*)}$ from the proposal $\boldsymbol{\pi}(\boldsymbol{\alpha}_k^{(*)})$ in
        $\boldsymbol{\alpha}_k^{(*)}\sim\boldsymbol{N}_{[01]}(\boldsymbol{W}_{k,k^*},\frac{2\boldsymbol{\sigma}^2}{\boldsymbol{h}_k^T\boldsymbol{h}})$
2: Compute the acceptance probability
        $\rho=\min=\left\{1,\frac{f\left(\alpha_k^{(*)}|\sigma^{2(t-1)},y,\Upsilon,\alpha_{\{k,k^*\}}^{(t)}\right)\pi\left(\alpha_k^{(t-1)}\right)}{f\left(\alpha_k^{(t-1)}\,|\,\sigma^{2(t-1)}y,\Upsilon,\alpha_{\{k,k^*\}}^{(t)}\right)\pi\left(\alpha_k^{(*)}\right)}\right\}$
    3: sample w $\sim U_{[0,1]}$
    4: if w< $\rho$ then
    5: $\alpha_k^{(t)}=\alpha_k^{(*)}$
    6: else
    7: $\alpha_k^{(t)}=\alpha_k^{(t-1)}$
    8: end if
```

3. Experiments

In this paper, we compare empirically four supervised unmixing algorithms; The K-Hype Algorithm, Super K-Hype (SK-Hype) Algorithm [6], GLUP-Lap [20], and the GBM Algorithms [4]. We use the same endmemer sets and the mixing models that have been used to experiment the considered unmixing approaches as reported in [4], [6], [18], [20]. and [24]. We should mention that these approaches did not use the same data. However, in our experiment, we convey the same data as input to all unmixing approaches in order to compare their performances. Thus, we used 7 input data with different mixture model, number of endmembers, number of signatures (spectra), number of pixels, Signal to Noise Ratio SNR, and abundances. In the follwing, we describe these datasets.

Data A is a synthetic image data with 900 pixels that are mixed by linear mixing model using the endmembers, the abundances in [6]. The linear mixing model is defined as:

$$Y=MA+n \qquad (12)$$

where Y is the pixel vector, M is the endmember matrix, A is the abundance one and n is a noise vector. For this data, the SNR is set to 30 and the number of pixels N to 900. The endmember matrix is 420X5, meaning that number of endmembers is 5 with 420 spectral bands.

Data B is a synthetic image data with 900 pixels that is mixed by bilinear mixing model. The bilinear mixture model is defined as:

$$\mathbf{r}=\mathrm{M}\alpha+\sum_{i=1}^{R-1}\sum_{j=i+1}^{R-1}\alpha_j\left(m_i\otimes m_j\right)+n \qquad (13)$$

where M is the endmember matrix and α is the abundance matrix.

Data C is a synthetic image data with 900 pixels that is mixed by intimate mixing model. The mixing model is defined as:

$$\mathrm{r}=(M\alpha)^{\xi}+n \qquad (14)$$

where ξ is set to 0.7, n is the noise, M is the endmember matrix, and α is the abundance matrix.

Data D is a 75x75 image that is mixed by linear mixing model. The endmember matrix size is 224X5 and is generated as in [20]. For this data, the SNR is set to 30 and number of pixels N is 75*75. The image contains 25 squares in a 5x5 grid pattern. Each square is a homogeneous surface, where all pixels inside the square have the same abundances. The first 20 squares are distinct from each other; i.e. each one has a different mixture of the endmembers, while the last 5 squares are identical [20].

The linear mixing model is expressed as:

$$Y = MA + N \qquad (15)$$

where N is an additive Gaussian noise. M is the endmember matrix, and A is the abundance matrix.

Data E is 75x75 image that is mixed by linear mixing model. Data E is generated similarly to Data D, except that it is created using 15 different endmembers, and has identical squares in each row. Each pixel has local similar neighbors and distant similar ones.

Data F is a set of pixels that are mixed by bilinear mixing model using the endmember matrix and the abundance defined in [4]. The endmember matrix size is 826X3. The three endmembers are extracted from the ENVI software library [28] [18], with 826 spectral bands and SNR=15. The bilinear mixture model defined as:

$$r = M\,\alpha + \sum_{i=1}^{R-1} \sum_{j=i-1}^{R} \gamma i,j\ \alpha_i \alpha_j\,(m_i \otimes m_j) + n \qquad (16)$$

Note that γi,j is a coefficient that controls the interactions between endmembers #i and #j in the considered pixel, R is the number of endmembers, M is the endmember matrix, α is the abundance and n is the noise.

Similarly, Data G is a set of pixels that are mixed by polynomial post-nonlinear mixing model using the endmember matrix and the abundance defined in [18, 24]. The mixture model is PNMM [18, 24] and defined as:

$$y = g_b(\mathrm{Ma}) + n = \mathrm{Ma} + b(Ma) \odot (Ma) + n \qquad (17)$$

where ⊙ denotes the Hadamard (term-by-term) product, M is the endmember matrix, a is the abundance and n is the noise. Note that the resulting PPNMM includes bilinear terms. However, the nonlinear terms are characterized by single amplitude parameter b. Table 1 summarises the characteristics of the used data sets.

Table 1. Data sets Characteristics

Data	A	D	E	B	F	C	G
Endmembers	5	5	15	5	3		3
pixels	900	75x75		900	1	900	1
SNR	30	30		30	15	30	15
Mixing type	linear			bilinear		Intimate	polynomial post-nonlinear

In order to assess the performance of the considered non-linear unmixing approach, we use the Root Mean Square Error (RMSE). It has been used as an evaluation measure in [4], [6] [18] and [20]. It is defined as in (19)

$$RMSE = \sqrt{\frac{1}{NR} \sum_{n=1}^{N} ||\alpha_n - \alpha_n^*||^2} \qquad (18)$$

We should mention that when RMSE is small, it reflects a good unmixing result. On the other hand, when it is large, it reflects a bad unmixing result. Figure 1, Figure 2, Figure 3, Figure 4, Figure 5, Figure 6, and Figure 7, shows the performance of the considered approaches when varying the experimental parameters on data set A, data set B, data set C, data set D, data set E, data set F, and data set G, respectively.

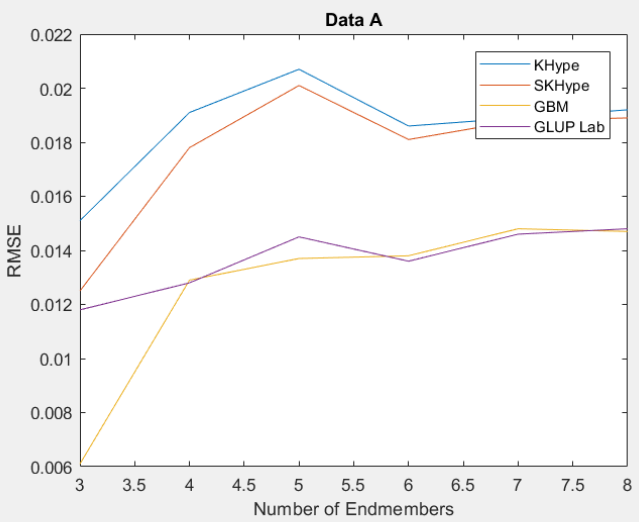

Figure 1. RMSE of Unmixing Data Awhen varying the number of endmembers

As shown in Figures 2, 3, 4, and 5, for each considered data set, one unmixing approach outperforms all the others regardless of the different considered parameters. However, in figure 1, we notice that although KHype achieves better results than the others on Data A when varying the number of endmembers, it is slightly worse than GLUP-Lap [20] when the number of endmembers is equal to 5.

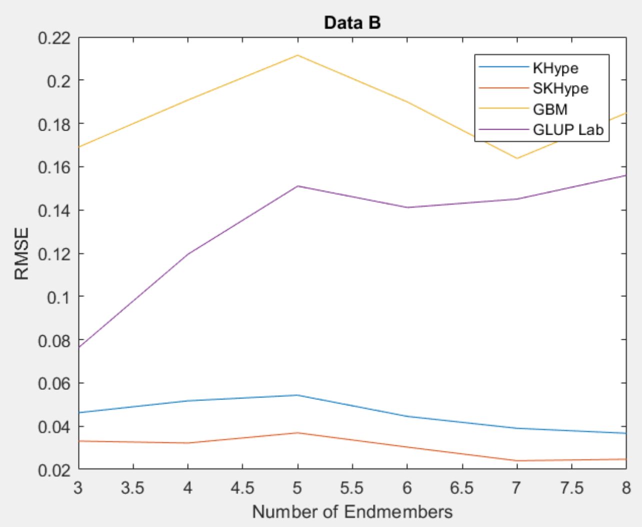

Figure 2. RMSE of Unmixing Data Bwhen varying the number of endmembers

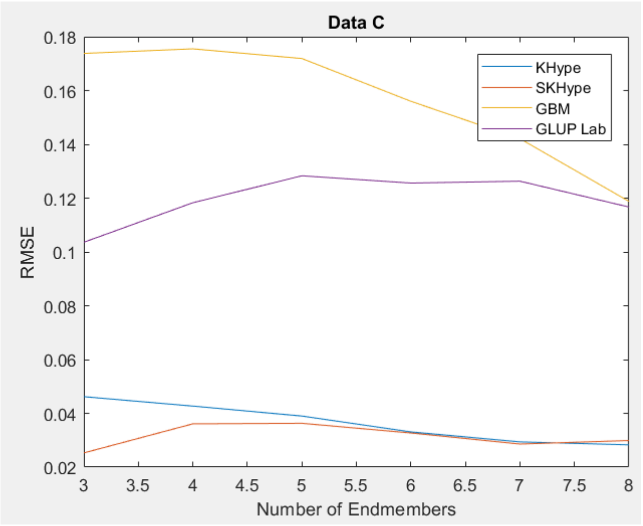

Figure 3. RMSE of Unmixing Data Cwhen varying the number of endmembers

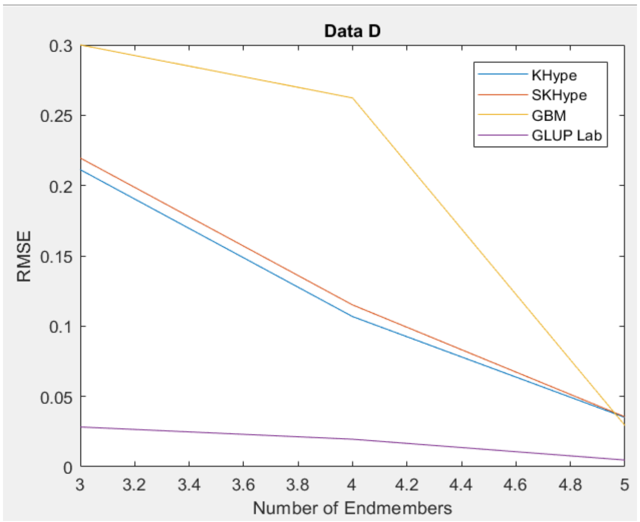

Figure 4. RMSE of Unmixing Data Dwhen varying the number of endmembers

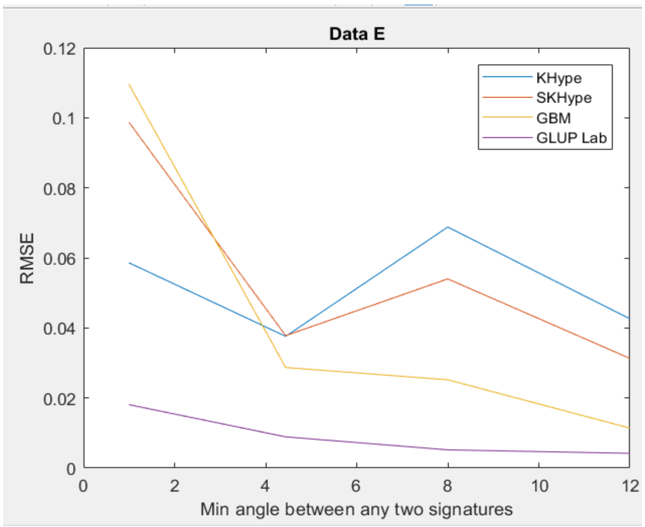

Figure 5. RMSE of Unmixing Data Ewhen varying the angle between any two signature

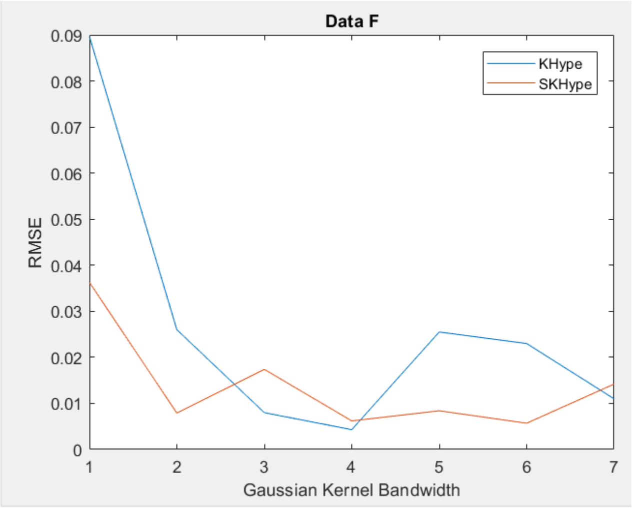

Figure 6. RMSE of Unmixing Data F when varying the Gussian Kernel bandwith

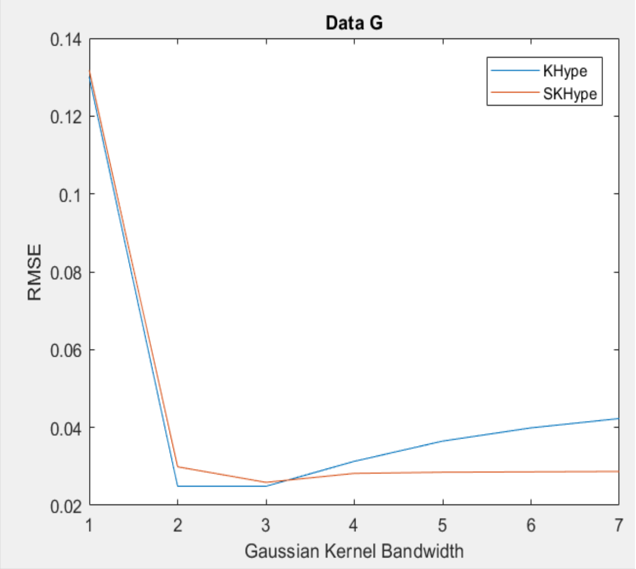

Figure 7. RMSE of Unmixing Data G when varying the Gussian Kernel bandwith

Table 2. Performance results of the Considered unmixing approaches

RMSE± STD	GBM	KHype	SKHype	GLUP-Lap
Data A	0.0134 ± 0.0002	0.0304 ± 0.0007	0.0206 ± 0.0003	0.0142 ± 0.0001
Data B	0.2134 ± 0.0217	0.0284 ± 0.0006	0.0294 ± 0.0006	0.1518 ± 0.0070
Data C	0.1716 ± 0.0061	0.0453 ± 0.0015	0.0362 ± 0.0011	0.1278 ± 0.0029
Data D	0.0293 ± 0.0009	0.0410 ± 0.0016	0.0361 ± 0.0012	0.0050 ± 0.00003
Data E	0.0284 ± 0.0009	0.0495 ± 0.0014	0.0405 ± 0.0011	0.0091 ± 0.00002
Data F	0.0052	0.0288	0.0083	-
Data G	0.0075	0.0424	0.0299	-

Similarly, KHype outperforms SKHype on Data F (refer to figure 6) when varying the Gaussian parameters except when the bandwidth is equal to 3. Figure 7 displays similar results for data G. In fact, KHype achieves better results than SKHype regardless of the Gaussian bandwidth expect for a slight decrease in the performance when this parameter is equal to 2.

Table 2 reports the abundance RMSE and the standard deviation of the considered unmixing approaches on the seven data sets. For a better understanding of the reults, table 3 shows the rank of all four nonlinear unmixing algorithms according to the goodness of RMSEs results from the first (the best) to the fifth order.

Table 3. Ranking Algorithms According to Their Performance

Mixing Model	Data	GBM	KHype	SKHype	GLUP-Lap
Linear	Data A	1st	4th	3rd	2nd
Bilinear	Data B	4th	1st	2nd	3rd
Intimate	Data C	4th	2nd	1st	3rd
Linear	Data D	2nd	4th	3rd	1st
Linear	Data E	2nd	4th	3rd	1st
Non-linear	Data F	1st	3rd	2nd	-
Non- linear	Data G	1st	3rd	2nd	-

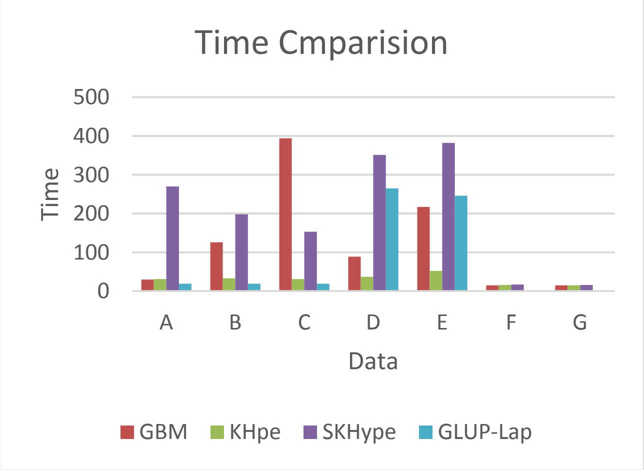

Figure 8. Time Comparison of the Considered Unmixing approaches

GBM algorithm [4] has the best performance in Data A which is linearly mixed image. Moreover, the results show that it has the second best performance in case of Data D, E, F and G. However, it performs poorly on both Data B and Data C which are mixed using bilinear and intimate mixing model. On the other hand, K-Hype algorithm has the best performance on unmixing Data B and has the second best performance on unmixing Data C. However, it performs poorly on Data A, D and E, which are mixed by linear mixing model. Besides, SK-Hype algorithm has the best performance on unmixing Data C which is mixed by intimate mixing model. It also has the second best performance on unmixing Data B which is mixed by bilinear mixing model. Finally, GLUP-Lap [20] algorithm outperforms the other considered algorithms on Data D and E which are mixed by linear mixing model. It also has the second best performance in case of Data A which is mixed by linear mixing model too.

In summary, we notice that the performance of the considered unmixing approaches vary with respect to the data. However, we conclude that GBM [4] gives good performance on linear and non-linear data, and KHype [6] and S-KHype are doing better on bilinear and intimate mixing models.

Figure 8 illustrates the running time in seconds of the considered approaches. We notice that the running time differs from one data to another, but generally GLUP-Lap [20] and KHype have the smallest running times.

4. Conclusions

In this paper, we compared empirically a set of non-linear unmixing algorithms. Seven input data sets have been mixed using different mixing models. The results show that GBM [4] is able to unmix linear and non-linear mixed models. However, it is not able to unmix bilinear or intimate mixed models. On the other hand, KHype [6] and S-KHype [6] give better performance results on bilinear and intimate unmixing models while they are not unmixing linear and non-linear mixing models. Thus, there is no universal unmixing approach that is able to unmix all the considered scenarios.

As the performance varies with respect to the data and therfore it is difficult to decide on the unmixing approach to adopt, we plan as future work to investigate combining fusion techniques on these approaches in order to obtain betterunmixing result regardless of the data set. We aim also to consider more approaches and other data sets.

REFERENCES

[1] Y. Altmann, "Nonlinear unmixing of hyperspectral images," PhD. Thesis, INP Toulouse, October 2013.

[2] Fauvel, M.; Tarabalka, Y.; Benediktsson, J.; Chanussot, J.; Tilton, J., “Advances in Spectral-Spatial Classification of Hyperspectral Images,” Proceedings of the IEEE, vol.101, no.3, pp.652-675, March 2013.

[3] Rajabi, R.; Ghassemian, H., “Hyperspectral Data Unmixing Using Gnmf Method and Sparseness Constraint,” Geoscience and Remote Sensing Symposium (IGARSS), 2013 IEEE International, pp. 1450-1453, July 2013.

[4] Abderrahim Halimi, Yoann Altmann, Nicolas Dobigeon and Jean-Yves Tourneret, “Nonlinear Unmixing of Hyperspectral Images Using a Generalized Bilinear Model”, Geoscience and Remote Sensing, IEEE Transactions on, vol.49, no.11, pp.4153-4162, Nov. 2011.

[5] Dobigeon, N.; Tourneret, J.-Y.; Richard, C.; Bermudez, J.C.M.; McLaughlin, S.; Hero, A.O., "Nonlinear Unmixing of Hyperspectral Images: Models and Algorithms," Signal Processing Magazine, IEEE, vol.31, no.1, pp.82,94, Jan. 2014.

[6] Chen, J.; Richard, C.; Honeine, P., "Nonlinear Unmixing of Hyperspectral Data Based on a Linear-Mixture/ Nonlinear-Fluctuation Model," Signal Processing, IEEE Transactions on , vol.61, no.2, pp.480, 492, Jan.15, 2013.

[7] Xiao, H.; Liu, H.; Chen, J., "Joint Supervised-Unsupervised Nonlinear Unmixing of Hyperspectral Images Using Kernel Method," Proceedings of the 2014 Fifth International Conference on Intelligent Systems Design and Engineering Applications, pp. 582-585, 2014.

[8] Yokoya, N.; Chanussot, J.; Iwasaki, A., "Generalized bilinear model based nonlinear unmixing using semi-nonnegative matrix factorization," in Geoscience and Remote Sensing Symposium (IGARSS), 2012 IEEE International, pp.1365-1368, 22-27 July 2012.

[9] M. E. Winter, "N-FINDR: An algorithm for fast autonomous spectral end-member determination in hyperspectral data," in Proc. SPIE Spectrom. V, 1999, vol. 3753, pp. 266–277.

[10] J. Boardman, "Atomatic spectral unmixing of AVIRIS data using convex geometry concepts," in Proc. AVIRIS Workshop, 1993, vol. 1, pp. 11–14.

[11] A. Zare, "Hyperspectral endmember detection and band selection using bayesian methods", Ph.D. dissertation, Univ. Florida, Gainesville, 2008.

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

[13] J. Li and J. M. Bioucas-Dias, "Minimum volume simplex analysis: A fast algorithm to unmix hyperspectral data," in Proc. IEEE IGARSS, 2008, pp. III-250–III-253.

[14] T.-H. Chan, C.-Y. Chi, Y.-M. Huang, and W.-K. Ma, "A convex analysis- based minimum-volume enclosing simplex algorithm for hyperspectral unmixing," IEEE Trans. Signal Process., vol. 57, no. 11, pp. 4418–4432, 2009.

[15] L. Miao and H. Qi, "Endmember extraction from highly mixed data using minimum volume constrained nonnegative matrix factorization," IEEE Trans. Geosci. Remote Sens., vol. 45, no. 3, pp. 765–777, 2007.

[16] Broadwater, J. and Banerjee, A. "A comparison of kernel functions for intimate mixture models." In Proc. IEEE Workshop on Hyperspectral Image and Signal Processing: Evolution in Remote Sensing (WHISPERS), pages 1-4, 2009.

[17] Broadwater, J., Chellappa, R., Banerjee, A., and Burlina, P. (2007). "Kernel fully constrained least squares abundance estimates." In Proc. IEEE Int. Conf. Geosci. and Remote Sensing (IGARSS), pages 4041_4044, Barcelona, Spain.

[18] Y. Altmann, A. Halimi, N. Dobigeon and J.-Y. Tourneret, "Supervised nonlinear spectral unmixing using a post-nonlinear mixing model for hyperspectral imagery," IEEE Trans. Image Processing, vol. 21, no. 6, pp. 3017-3025, June 2012.

[19] Yoann Altmann, Nicolas Dobigeon, Steve McLaughlin and Jean-Yves Tourneret, "Unsupervised nonlinear unmixing of hyperspectral images using gaussian processes", IEEE, 2012.

[20] Rita Ammanouil, Andr´e Ferrari, and C´edric Richard, "A graph laplacian regularization for hyperspectral data unmixing", Acoustics, Speech and Signal Processing (ICASSP), 2015 IEEE International Conference, 2015.

[21] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, "Distributed optimization and statistical learning via the alternating direction method of multipliers," Foundations and Trends in Machine Learning, vol. 3, no. 1, pp. 1–122, 2011.

[22] M. Babaie-Zadeh, C. Jutten, and K. Nayebi, "Blind Separating convolutive post non-linear mixtures," in Proc. 3rd ICA Workshop, San Diego, CA, 2001, pp. 138–143.

[23] C. Jutten and J. Karhunen, "Advances in nonlinear blind source separation," in Proc. Int. Symp. Independ. Compon. Anal. Blind Signal Separat. (ICA), 2003, pp. 245–256.

[24] Y. Altmann, A. Halimi, N. Dobigeon and J.-Y. Tourneret, "Supervised nonlinear spectral unmixing using a polynomial post nonlinear model for hyperspectral imagery," in Proc. IEEE Int. Conf. Acoust., Speech, and Signal Processing (ICASSP), Prague, Czech Republic, May 2011, pp. 1009-1012.

[25] M.-D. Iordache, J. Bioucas-Dias, and A. Plaza, "Total variation spatial regularization for sparse hyperspectral unmixing," IEEE Transactions on Geoscience and Remote Sensing, vol. 50, no. 11, pp. 4484–4502, 2012.

[26] J. Chen, C. Richard, and P. Honeine, "Nonlinear estimation of material abundances in hyperspectral images with ℓ1-norm spatial regularization," IEEE Transactions on Geoscience and Remote Sensing, vol. 52, no. 5, pp. 2654 – 2665, 2014.

[27] K. KRISHNAMURTHYy, M. RAGINSKY, AND R. WILLETTy, "Multiscale Photon-Limited Spectral Image Reconstruction".

[28] "ENVI User's Guide Version 4.0," RSI, Boulder, CO, Sep. 2003

[29] N. D. Lawrence, "Gaussian process latent variable models for visualisation of high dimensional data," in NIPS, Vancouver, Canada, 2003.

[30] Chang Li, Yong Ma, Xiaoguang Mei, Fan Fan, Jun Huang and Jiayi Ma, "Sparse Unmixing of Hyperspectral Data with Noise Level Estimation" 2017.

[31] Xiaoguang Mei, Yong Mei, Chang Li , Fan Fan a, Jun Huang, Jiayi Ma, "Robust GBM hyperspectral image unmixing with superpixel segmentation based low rank and sparse representation" 2018.