Regularization approaches to hyperspectral unmixing

Maria Toomik

A thesis submitted in partial fulfillment

of the requirements for the degree of

Doctor of Philosophy

University College London

Department of Statistical Science

December 20, 2019

In memory of Dr. James D.B. Nelson

Declaration

I, Maria Toomik confirm that the work presented in this thesis is my own. Where information has been derived from other sources, I confirm that this has been indicated in the thesis.

Abstract

We consider a few different approaches to hyperspectral unmixing of remotely sensed imagery which exploit and extend recent advances in sparse statistical regularization, handling of constraints and dictionary reduction. Hyperspectral unmixing methods often use a conventional least-squares based lasso which assumes that the data follows the Gaussian distribution, we use this as a starting point. In addition, we consider a robust approach to sparse spectral unmixing of remotely sensed imagery which reduces the sensitivity of the estimator to outliers. Due to water absorption and atmospheric effects that affect data collection, hyperspectral images are prone to have large outliers.

The framework comprises of several well-principled penalties. A non-convex, hyper-Laplacian prior is incorporated to induce sparsity in the number of active pure spectral components, and total variation regularizer is included to exploit the spatial-contextual information of hyperspectral images. Enforcing the sum-to-one and non-negativity constraint on the models parameters is essential for obtaining realistic estimates. We consider two approaches to account for this: an iterative heuristic renormalization and projection onto the positive orthant, and a reparametrization of the coefficients which gives rise to a theoretically founded method. Since the large size of modern spectral libraries cannot only present computational challenges but also introduce collinearities between regressors, we introduce a library reduction step. This uses the multiple signal classification (MUSIC) array processing algorithm, which both speeds up unmixing and yields superior results in scenarios where the library size is extensive.

We show that although these problems are non-convex, they can be solved by a properly defined algorithm based on either trust region optimization or iteratively reweighted least squares. The performance of the different approaches is validated in several simulated and real hyperspectral data experiments.

Impact statement

Satellite imagery is gaining popularity for monitoring of the Earth as more satellites are launched into the orbit every year. Thanks to recent international collaboration, remote sensing data is becoming ever more accessible and abundant. Timely automated analysis of the image data is of vital importance to future resource and climate risk management. Statistical image processing methodology must now catch up with rapid hardware advances and initiatives in remote sensing to deliver automatic, accurate, and robust data analysis.

The work presented in this thesis has applications for use both inside and outside academia. We motivate the use of statistical methods such as regularization, reparametrization and robust methods in image processing applications. We have produced well-functioning algorithms for the hyperspectral unmixing problem that can be used in a wide range of scenarios. This has implications for many industries, and can be adopted for example by military, government institutions, international organizations and multinational corporations. Satellite imagery has potential for monitoring land use, melting ice caps, oil spills etc.

The hyperspectral unmixing task is a statistical image processing problem that is relevant in the academic literature. The statistical regularization tools adopted for hyperspectral unmixing in this thesis can be adapted to various applied data sciences in which this kind of analysis is meaningful. Image processing methods can be developed further using the ideas and insights presented.

Publications

Chapters 2-5 in this thesis form four separate papers that have been submitted to journals for publishing. Their titles are as follows:

- Sparse hyperspectral unmixing with spatial total variation regularization and library pruning
- Sparse hyperspectral unmixing with spatial total variation regularization and constraint's reparametrization
- Robust sparse spatial total variation regularized hyperspectral unmixing
- Robust regularized hyperspectral unmixing with constraint's reparametrization

A shorter and less extensive version of Chapter 4, was published as a conference paper:

[1] Toomik, Maria, Shijian Lu, and James D. B. Nelson. "M-estimation for robust sparse unmixing of hyperspectral images." *Image and Signal Processing for Remote Sensing XXII*. Vol. 10004. p. 100040V. International Society for Optics and Photonics, 2016.

Contents

List of Tables

List of Figures

List of Algorithms

Notation

L	number of spectral bands
N	number of materials/endmembers
P	number of pixels in the image
$\boldsymbol{y}_n$	spectral reflectance of the nth pixel, size $L \times 1$
$\boldsymbol{M}_0$	spectral library/mixing matrix, size $L \times N$
$\boldsymbol{x}_n$	fractional abundance vector for the nth pixel, size $N \times 1$
$\boldsymbol{e}_n$	observational errors for the nth pixel, size $L \times 1$
λ	regularization parameter for the sparsity term
λ_{TV}	regularization parameter for the spatial term
p	quasi-norm with values in the range $0 < p < 1$
∇	operator that takes horizontal and vertical differences of the fractional abundances between neighboring pixels
$\boldsymbol{y}$	spectral reflectance for the whole image s.t. $\left[\boldsymbol{y}_1^\top, \boldsymbol{y}_2^\top, \ldots, \boldsymbol{y}_P^\top\right]^\top$, size $LP \times 1$
$\boldsymbol{M}$	block-diagonal mixing matrix with P entries of $\boldsymbol{M}_0$ on the main diagonal s.t. $I_p \otimes \boldsymbol{M}_0$, where $\otimes$ is the Kronecker product, size $LP \times NP$
$\boldsymbol{x}$	fractional abundance estimates for the whole image s.t. $\left[\boldsymbol{x}_1^\top, \boldsymbol{x}_2^\top, \ldots, \boldsymbol{x}_P^\top\right]^\top$, size $NP \times 1$
$\boldsymbol{e}$	observational errors for the whole image s.t. $\left[\boldsymbol{e}_1^\top, \boldsymbol{e}_2^\top, \ldots, \boldsymbol{e}_P^\top\right]^\top$, size $LP \times 1$
$\boldsymbol{d}$	weights for the sparsity term, size $NP \times 1$
$\boldsymbol{D}$	positive diagonal weight matrix containing the weights $\boldsymbol{d}$, size $NP \times NP$
$\boldsymbol{\epsilon}_n$	measure of error in the solution for the nth pixel, size $N \times 1$

$\boldsymbol{\epsilon}$	measure of error in the solution for the whole image s.t. $\left[\boldsymbol{\epsilon}_1^\top, \boldsymbol{\epsilon}_2^\top, \ldots, \boldsymbol{\epsilon}_P^\top\right]^\top$, size $NP \times 1$
α	iteration number
T	total number of differences taken between neighboring pixels in the image
T_h	number of horizontal differences taken for the image
T_v	number of vertical differences taken for the image
c	number of pixels in each row of the image
ϕ	weights for the spatial term, size $T \times 1$
Φ	positive diagonal matrix containing the weights ϕ for the spatial term, size $T \times T$
η	small positive integer of size 10^{-6}
$\boldsymbol{\eta}$	column vector filled with η values, size $NP \times 1$
t_i	unconstrained reparametrized parameter for endmember i
x_i	sum-to-one and non-negatively constrained parameter for endmember i
ρ	robust function
ψ	first derivative of ρ
ω	second derivative of ρ
ξ_a, ξ_b, ξ_c	threshold parameters for the robust function ρ
$\boldsymbol{w}$	weights for the robust function, size $LP \times 1$
$\boldsymbol{W}$	positive diagonal matrix containing the weights $\boldsymbol{w}$, size $LP \times LP$

1

Introduction

Understanding, protecting, and optimizing land-use and natural resources is a fundamental interest to the environmental sciences and society as a whole. It informs, if not drives, environmental policy both at the government and international level. Set against the backdrop of the current global climate crisis, its importance and timeliness cannot be underestimated. To this end, hyperspectral imaging offers a uniquely effective and potentially automated means of monitoring, often subtle, changes in the environment such as CO_2 emissions [2,3]; oil/gas leakage [4,5]; snow/ice melting/grain-size [6,7]; biomass [8,9]; dissolved organic materials [10, 11]; atmosphere composition [12]; etc [13–17]. Members of the *Committee on Earth Observation Satellites* (CEOS) have planned to conduct 268 satellite Earth observation missions between the years 2012 and 2027 [18]. Many leading space agencies have current space programs, or have scheduled launches over the next few years to place cutting edge hyperspectral remote sensors into low-Earth orbit. For example, the Italian Space Agency launched the PRISMA[1] (*Hyperspectral Precursor and Application Mission*), satellite system that carries a hyperspectral sensor, into space in March 2019 aboard the VEGA launcher (currently still in trial phase); and the Germans are planning to launch the hyperspectral satellite mission EnMAP[2] (*Environmental Mapping and Analysis Program*), that is currently in the development and production phase, in 2020 (the launch was previously planned for 2015 [19]). The spatial resolution of the planned sensors will stay the same as the previous and current ones, having around 20-30-meter resolution[3]. Hence spectral unmixing of hyperspectral images will be a important challenge to solve for years to come.

Timely automated analysis of the image data is of vital importance to future resource and climate risk management. Thanks to recent international collaboration, remote sensing data is becoming ever more accessible and abundant. Statistical image processing

[1]Further details about PRISMA at: http://www.asi.it/en/activity/observation-earth/prisma
[2]Further details about EnMAP at: http://www.enmap.org/?q=mission
[3]Further details about Hyperion at: https://archive.usgs.gov/archive/sites/eo1.usgs.gov/hyperion.html

methodology must now catch up with rapid hardware advances and initiatives in remote sensing to deliver automatic, accurate, and robust data analysis.

The standard red/green/blue and multispectral images only have a few spectral channels, but hyperspectral imagery often has information for over two hundred spectral bands in the visible light and near infrared spectrum that is collected in narrow wavebands. This extra data allows for the analysis of materials in the image scenes. Hyperspectral sensors record the spectral signature of the materials for each pixel separately. Hence if there is only one material in the area of the pixel, a pure pixel is observed. Concurrently, hyperspectral imaging sensors provide reflectance information from many different wavelengths simultaneously. With appropriate image modeling and estimation methodology, a rich description can be revealed. The high spectral resolution of the hyperspectral images is lessened by the low spatial resolution of the data, where each pixel in the image often covers a large area on the ground, e.g. 30m by 30m. The low spatial resolution of current sensors, together with the complexity presented by many environments, provides considerable challenges because each pixel often contains the spectrum of more than one distinct pure material, or *endmember*. This leads to the existence of mixed spectra in hyperspectral imagery. Mixed pixels tend to have greater spectral variance and are comparatively more difficult to interpret. This embodies the primary impediment to hyperspectral image analysis and creates the need for hyperspectral unmixing.

The unmixing task is a statistical image processing problem that has gained popularity. There has been a wave of interest in translating recent ideas from statistical regularization to advance hyperspectral unmixing capabilities. This has happened as tools from statistical regularization begin to mature and are adapted to applied data sciences. The main methodological thrust is to exploit any prior information, encode it into a penalty, or log-prior term, and use modern optimization machinery to perform maximum a posteriori estimation.

Unmixing can be thought of as classification at the sub-pixel level. It is used to decompose the spectral mixture in each pixel into endmembers and their respective proportions, so-called *fractional abundances*. Hence, considering the spectrum in each pixel is a mixture, unmixing then consists of two tasks: identifying the endmembers in the image, and estimating the abundances of endmembers in each pixel. We will concentrate on the abundance estimation problem in hyperspectral imagery. The endmember identification task is a well-established domain with numerous proposed methods, it is also rather straightforward due to geometry. The methods either assume that there are some pure pixels in the image scene, e.g. the N-FINDR [20], pixel purity index [21], independent component analysis (ICA) [22] and the vertex component analysis (VCA) [23], or they are are minimum volume based, e.g. minimum volume constrained non-negative matrix factorization [24], minimum-volume enclosing simplex [25], robust alternating volume maximization [26] and minimum simplex analysis [27]. Assuming that the observed spectral signatures can be expressed as a linear combination of pure spectral signatures known in advance, i.e. us-

ing spectral libraries that contain spectra collected in laboratory settings, unmixing then consists of finding the best subset of signatures in the spectral library [28]. This is an alternative to using the endmember identification algorithms for extracting the endmember signatures from the image.

1.1 Classical unmixing methods

The linear mixing model (LMM) is widely used for signal processing and spectral unmixing tasks. It assumes that the spectrum of each pixel is a linear combination of the endmembers. The task then involves finding the best subset of signatures to describe the spectra. The LMM has proven to be a good approximation of the physical process generating the spectra and has produced good results for certain applications. It models the spectral mixture in the nth pixel in the image $\boldsymbol{y}_n \in \mathbb{R}^{L\times 1}$, for $n = 1, \ldots, P$, where P is the number of pixels, such that

$$\boldsymbol{y}_n = \boldsymbol{M}_0 \boldsymbol{x}_n + \boldsymbol{e}_n \; . \tag{1.1}$$

Here, $\boldsymbol{M}_0 \in \mathbb{R}^{L\times N}$ is the spectral library containing the signatures of the N endmembers over the L spectral bands, $\boldsymbol{x}_n$ is the $N \times 1$ fractional abundance vector, and $\boldsymbol{e}_n = [e_n[1], \ldots, e_n[L]]^\top$ is the vector of observational errors over the L spectral bands. The errors are assumed to follow a Gaussian distribution. As the fractional abundances define proportions, we then need to enforce some constraints in order to get realistic results. We want the fractional abundance vector to be restrained by the following abundance non-negativity (ANC) and abundance sum-to-one (ASC) constraints: $\boldsymbol{x}_n \geq 0$, $\boldsymbol{a}\boldsymbol{x}_n = 1$, where $\boldsymbol{a} = [1, \ldots, 1]$ of size N. Early examples of this approach encoded the sum-to-one and/or non-negativity constraints of the abundances as penalties to form constrained least-squares solutions [29].

Although most pixels in a hyperspectral image contain a mixture spectrum of different materials, the number of endmembers in each pixel is generally very low [30]. This has motivated the use of sparse regression methods like the *least absolute shrinkage and selection operator* (LASSO) [31] to enforce sparsity on the fractional abundance vector [28, 32, 33]. For example, Guo et al [32] reported significantly superior results when enforcing sparsity on the fractional abundance vector. The ridge regression with an ℓ_2 norm was quickly replaced with the ℓ_1 norm to enforce sparsity on the fractional abundance vector $\boldsymbol{x}_n$ as it enforces greater sparsity. The LASSO is defined as

$$\arg\min_{\boldsymbol{x}_n} \|\boldsymbol{y}_n - \boldsymbol{M}_0 \boldsymbol{x}_n\|_2^2 + \lambda \|\boldsymbol{x}_n\|_1 \; , \tag{1.2}$$

where $\lambda \geq 0$ is the regularization parameter. For a greater value for λ, the constraint $\|\boldsymbol{x}_n\|_1$ is given more weight which increases the sparsity of the result. The ℓ_1 norm is a

combinatorial problem with a finite set of possibilities, but determining the best possible solution can be difficult as it is computer-intensive and time-consuming. The ℓ_2 norm with $\|\boldsymbol{x}_n\|_2^2$ is frequently used to reduce the size of the set of possible solutions and to make the computation faster. However, it is not as efficient at enforcing sparsity on the solution [34, 35]. The sparsity of the unmixing result can be increased further by imposing the ℓ_p norm, with $0 < p < 1$, on the fractional abundances instead of the ℓ_1 norm used in the Lasso [34]. This would improve the unmixing performance further [35]. The illustration of the different norms can be seen in Fig. 1.1. Shown are the contours of the three constraint functions. It can be easily seen that if one was to draw an error function onto the ℓ_2 norm plot, it would most likely hit a point in the constraint region where the value is not equal to zero. In fact, for ℓ_2 it is very difficult to get an exact zero. The probability of getting zeros increases with the ℓ_1 norm and even more so with ℓ_p norm for $p < 1$, as the constraint's contours are concave. The more zeros in the result, the sparser the solution.

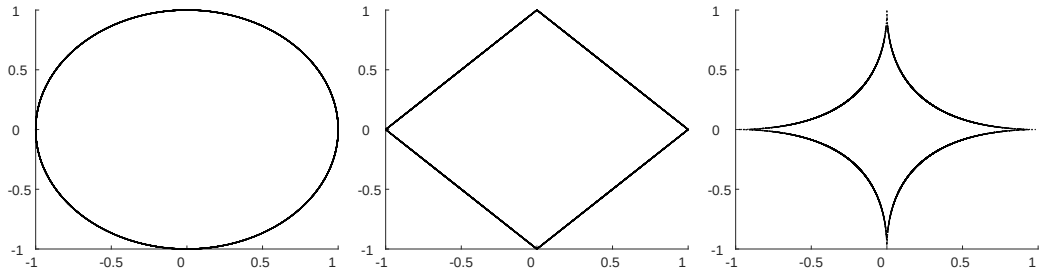


Figure 1.1: Illustration of different norms for regularization (a) ℓ_2 norm. (b) ℓ_1 norm. (c) ℓ_p norm with $p = 0.5$.

In many unmixing scenarios it is reasonable to assume that a high proportion of neighboring pixels will contain similar fractional abundances. We can expect further improvement in the unmixing results if we incorporate the spatial information available in the hyperspectral images, although many methods have ignored such information [36]. We need to add another regularization term to the optimization problem with a constraint that ensures that the neighboring pixels in the image differ very little. A total variation regularizer is optimal for this task [37–39]. The optimization problem in Eq. (1.2) can be extended without difficulty to include a total variation term:

$$\arg\min_{\boldsymbol{x}_n} \|\boldsymbol{y}_n - \boldsymbol{M}_0\boldsymbol{x}_n\|_2^2 + \lambda \|\boldsymbol{x}_n\|_1 + \lambda_{TV} \|\nabla \boldsymbol{x}_n\|_1 \ , \tag{1.3}$$

where ∇ is an operator which takes both horizontal and vertical differences between neighboring pixels [40]. This statistical problem is more generally known as the fused lasso [41] or sparse 2d fused lasso [42], which is a type of sparse, total variation denoising.

1.2 Related Work

Alternating direction method of multipliers (ADMM) and gradient decent are commonly used in literature for solving sparse regression optimization problems. They have slow convergence, but due to the low complexity of each calculation the total computation time required is fairly low. They perform well when the optimization problem is convex. However, these algorithms are not guaranteed to converge or they may find a suboptimal solution when the optimization problem is non-convex [43, 44]. We are interested in using the ℓ_p norm in the sparse regression problem, which makes the problem non-convex. Iteratively reweighted least squares (IRLS) is a fast solver, that offers a great alternative [45]. Instead of directly minimizing the non-convex ℓ_p norm that may result in a local minimum, IRLS solves a sequence of smoothed sub-problems [46]. It has been successfully used in compressive sensing and signal reconstruction [47], and is also used for sparse spectral unmixing and other optimization problems [35, 48, 49] in the recent years [50]. In this thesis, we employ the ℓ_p norm in the sparse regression problem, which makes the problem non-convex. We also use the trust region optimization that is based on first and second order analytical derivatives. It is generally more stable than its line search counterparts, particularly for problems that are, for example, non-concave and/or exhibit regions that are close to flat [51]. This approach has been used in a number of different applications [52–54], however, to the best of our knowledge, it has not been used for spectral unmixing so far.

When unmixing is phrased as an optimization problem, a priori knowledge of the abundances' features such as non-negativity and additivity can be accounted for by constraints or penalties such as the sum-to-one or abundance non-negativity constraint. Similarly, a relatively small numbers of endmembers active in each pixel can be found by adopting a sparsity penalty term. None of these terms, however, reflect the observation that abundance vectors of neighboring pixels are often very similar. Such spatial association in the abundance vectors is due to the fact that materials in nature such as minerals, grass, trees, etc, often form in contiguous clumps. Most hyperspectral unmixing algorithms ignore the spatial-contextual information that is inherent in the hyperspectral data [36]. Taking such information into account during the unmixing process would benefit the performance of the algorithms significantly.

In the recent years, more methods have been published that take into account the spatial information within the hyperspectral data in the form of correlations between spatial and spectral neighbors, which improves the results. These are mostly performing image classification [55–59]. However, classification ignores the fact that, due to the low resolution of hyperspectral images, several endmembers often jointly occupy a pixel. Spatial characteristics can be taken into account by using several existing techniques and models such as Markov random fields [60], loopy belief propagation [57], edge-preserving filtering [56], and morphological profiles [58]. The authors of [61] develop a spatial-spectral pre-

processing step prior to unmixing. In [62], a Laplacian constraint on the nearest neighbors of a pixel is used to incorporate contextual information. A weighted quadratic function on the differences between neighbors and a regularizer that uses the weighted proportion values of neighboring pixels are integrated into the optimization problem in [63] and [64], respectively. Similar ways of using spatial information can be found in [65–67].

Spatial features can be captured in a probabilistically well-principled way using the total variation (TV) regularizer [68]. It can be used to incorporate spatial information as it accounts well for spatial homogeneity [38, 39]. In [32], TV is used to unmix and improve the resolution of hyperspectral images. The TV regularizer is incorporated into the alternating projected sub-gradient descent for hyperspectral unmixing in [69]. The TV penalty fits rather naturally into a statistical regularization framework used to encourage sparsity and other constraints on the abundances, and has been recently combined with sparse hyperspectral unmixing [37, 70, 71].

Although sparse unmixing has overcome many problems, the high mutual coherence (often close to one [72]) of the always expanding spectral libraries limits the success of sparse unmixing methods. The mutual coherence between the columns of the design matrix and the sparseness of the original signals affect the sparseness of the solutions for an under-determined system of equations [28, 73]. Sparse unmixing can be thought of as a multiple measurement vector (MMV) problem. Using a linear mixing model, the MMV problem can be described as a sparse regression with the objective of estimating the fractional abundance of vectors with the same sparse support. The best performing method for MMV problems is the multiple signal classification (MUSIC) algorithm [74,75]. The MUSIC algorithm was recently used in [76] to introduce a library reduction step prior to spectral unmixing. This method simplifies the computation and attempts to overcome the high mutual coherence problem. However, it does not take spatial information into account. This was further developed in [77] and [49] to incorporate the spectral signature mismatches in the MUSIC formulation to create a robust version of MUSIC. The library reduction step by MUSIC is useful in situations when the initial library size is large.

The unmixing problem comes with certain physical limitations that we must follow to obtain realistic estimates. As our goal is to estimate the fractional abundances of materials in an image scene, we need to impose the abundance sum-to-one (ASC) and abundance non-negativity (ANC) constraints on each pixel. Both these constraints were enforced using a traditional least squares spectral mixture analysis (FCLS) in [78]. The same optimization problem is tackled using the alternating direction method of multipliers (ADMM) with an algorithm called sparse unmixing by variable splitting and augmented Lagrangian (SUnSAL) [28]. A renormalization heuristic and projection onto the positive orthant was used to enforce the constraints in [62]. These are examples of constrained optimization. However, if one wishes to perform unconstrained optimization, then reparametrisation with respect to the constraints is necessary. Although reparametrization of coefficients is a prevalent method in statistics, it has not been used in spectral unmixing until recently [79].

The reparametrisation of [80] is equivalent to the reparametrisation for independent Poisson mixture models in [81]. A general form of reparametrization with the logarithmic function as one option for the function for reparametrization was proposed in [82]. Multivariate logistic transformation was used for reparametrization in [83], whereas [84] used a Dirichlet distribution as a prior to enforce the constraints. In [85] the Dirichlet distribution was used in combination with reparametrization.

The traditional least squares based LMM is very sensitive to impulsive noise and outliers [86], whereas robust methods provide protection from noise [87, 88]. Outliers can affect data-based methods used to determine the tuning or penalty constants involved in the smoothing steps [89], hence robust extensions to generalized additive models were presented in [90, 91], and a robust mixing model was introduced to describe the hyperspectral data in [92]. Bilinear methods with mismodeling effects can consider the effect of outliers and cope with some types of non-linearities [93]. The non-Gaussian heavy-tailed relaxation motivated in hyperspectral image segmentation [94] and modeling papers [95] can be accommodated by the recently proposed regularization models and approaches for unmixing with a few modifications to the objective function and algorithmic framework. To reduce the effect of outliers, [96] replaced the maximum likelihood based weights in their iteratively weighted least squares (IRLS) algorithm with others derived from quasi likelihood equations. More recently robust methods have also been applied to spectral unmixing [97–99]. We also consider a robust approach to sparse spectral unmixing of remotely sensed imagery data which reduces the sensitivity of the estimator to outliers.

1.3 Overview

In Chapter 2, we extend the standard linear mixing model with sparsity and spatial total variation regularization using the ℓ_p norm with $0 < p < 1$. We use a simple heuristic method to handle the abundance sum-to-one and non-negativity constraints in an iterative manner. We introduce a library reduction step that allows us to handle situations when the initial library size is huge and mutual coherence very high. Hence we create a fast new algorithm based on iteratively reweighted least squares to perform hyperspectral unmixing. In Chapter 3, we develop and derive an analytical way of taking account of the ANC and ASC constraints for hyperspectral unmixing using reparametrization of coefficients. In addition, we introduce an original algorithm based on trust region estimation that can handle the non-convex optimization problem by solving a sequence of smoothed sub-problems instead of directly minimizing the non-convex one. Chapter 4 begins with reasons for the extension of the least-squares model, and highlights the necessity for robust methods. It then continues to introduce a new algorithm based on a robust loss function that can adaptively assign reduced or even zero weights to outliers. In Chapter 5, we create a novel algorithm using the robust loss function and reparametrization of coefficients. The performance of the algorithms in Chapters 2-5 is tested on both synthetic and real

hyperspectral images and compared with other state-of-the-art hyperspectral unmixing methods. Chapter 6 summarizes the thesis and offers some concluding remarks.

2

Sparse total variation regularization

This chapter focuses on regularization extensions to the spectral unmixing problem. Two different regularization terms are proposed: sparsity regularization and spatial regularization. These extra terms are shown to be beneficial in the unmixing process. The regularization terms are combined into a new unmixing algorithm for hyperspectral imagery. The performance of the algorithm is tested on simulated and real hyperspectral images and compared with other state-of-the-art algorithms.

2.1 Introduction

In this chapter we propose a new constrained sparse unmixing technique with spatial total variation regularization that induces sparsity in the number of active pure spectral components via a hyper-Laplacian prior, the ℓ_p norm with $0 < p < 1$. Here, we use the non-convex ℓ_p norm because it produces sparser results than the ℓ_1 norm [34]. As such, since it can shrink the fractional abundances of inactive endmembers closer to zero, it offers greater selectivity than the more traditional ℓ_1 norm. In turn this enables the method to be used on large, semi-supervised spectral libraries or in unsupervised scenarios where the number of pure signatures present in a typical pixel is much smaller than the number present in the image scene. We propose to estimate the fractional abundances $\boldsymbol{x}$ as a solution to the non-convex optimization problem

$$\arg\min_{\boldsymbol{x}} \|\boldsymbol{y} - \boldsymbol{M}\boldsymbol{x}\|_2^2 + \lambda \|\boldsymbol{x}\|_p^p + \lambda_{TV} \|\nabla \boldsymbol{x}\|_p^p \ , \tag{2.1}$$

subject to $\boldsymbol{x} \geq 0, \boldsymbol{a}\boldsymbol{x}_n = 1$, where $\boldsymbol{a} = [1, \ldots, 1] \in \mathbb{R}^N$, and $\boldsymbol{y}, \boldsymbol{x}, \boldsymbol{M}$ with the respective dimensions $LP \times 1$, $NP \times 1$, $LP \times NP$ describe the unmixing problem over all pixels

jointly. They are defined as

$$\begin{aligned} \boldsymbol{y} &= \left[\boldsymbol{y}_1^\top, \boldsymbol{y}_2^\top, \ldots, \boldsymbol{y}_P^\top\right]^\top, \\ \boldsymbol{x} &= \left[\boldsymbol{x}_1^\top, \boldsymbol{x}_2^\top, \ldots, \boldsymbol{x}_P^\top\right]^\top, \\ \boldsymbol{M} &= I_p \otimes \boldsymbol{M}_0 \,. \end{aligned} \tag{2.2}$$

Here $\otimes$ denotes the Kronecker product; $\boldsymbol{y}_n \in \mathbb{R}^{L\times 1}$ is the hyperspectral reflectance of the nth pixel; $\boldsymbol{x}_n$ of size $N \times 1$ is the estimated fractional abundance of the N endmembers at pixel n; and $\boldsymbol{M}_0 \in \mathbb{R}^{L\times N}$ is the spectral library that forms a block-diagonal matrix $\boldsymbol{M}$ with P blocks of $\boldsymbol{M}_0$ on the main diagonal and where the rest of the matrix $\boldsymbol{M}$ consists of zeros.

We propose a two-step algorithm to solve the minimization problem (2.1). The first step performs library reduction using hyperspectral subspace identification by minimum error [100], and a binary test similar to MUSIC to identify the endmembers. In an environment with no noise the MUSIC step is able to identify the active endmembers correctly [76]. The second step solves the optimization problem (2.1) via iteratively reweighted least squares (IRLS). IRLS with ℓ_p norm has local super-linear convergence, and although the problem (2.1) is non-convex, it is not necessarily intractable for IRLS [34]. We call the algorithm MUSIC-IRLSTV. Additionally, we include another version of the proposed algorithm that uses robust dictionary reduction. The method is called RMUSIC-IRLSTV and it accounts for extra noise in the library reduction step.

Table 2.1: Overview of methods

Method	Library pruning	Sparsity $\|.\|_p$	Spatial $\|\nabla.\|_p$	Sum-to-one	Non-negativity
NCLS	No	No	No	No	Yes
IRLS [35]	No	Yes	No	Normalize	Yes
SUnSAL [28]	No	p=1	No	No	Yes
SUnSAL-TV [37]	No	p=1	p=1	No	Yes
SUnSAL-CSR [101]	No	p=1	No	No	Yes
MUSIC-CSR [76]	Yes	p=1	No	No	Yes
RMUSIC-DANCER [49]	Robust	Yes	No	No	Yes
MUSIC-IRLSTV	Yes	Yes	Yes	Normalize	Yes
MUSIC-IRLSTV2	Yes	Yes	Yes	Lagrangian	Yes
RMUSIC-IRLSTV	Robust	Yes	Yes	Normalize	Yes

Through the $|\nabla \boldsymbol{x}|$ term, we introduce a total variation regularizer. However, unlike [37] that uses the ADMM method and the ℓ_1 norm, we achieve greater sparsity in both the estimated abundances and the spatial differences of the abundances by setting up a constrained $\ell_p - \ell_2$ optimization problem and solving it with an IRLS algorithm. The proposed algorithm MUSIC-IRLSTV performs constrained $\ell_p - \ell_2$ optimization with spatial

total variation regularization and library reduction. Experiments on both synthetic and real data show that the proposed method obtains better hyperspectral unmixing performance compared to other recent state-of-the-art statistical regularization methods such as: IRLS [35], SUnSAL [102], SUnSAL-TV [37], MUSIC-CSR [76], RMUSIC-DANCER [49]. A simulation study on the convergence of the MUSIC-IRLSTV algorithm shows that the algorithm converges in less than 50 iterations with $p < 1$. Unlike the renormalization heuristics used in [35, 78, 103], we derive and incorporate a well-principled analytical sum-to-one constraint for the optimization problem (2.1), and find that it offers further performance enhancement in our experiments. The difference between other methods and our approach is summarised in Table 2.1.

2.2 Iteratively Reweighted Least Squares with ℓ_p Norm

The constrained, sparse $l_p - l_2$ hyperspectral unmixing task, with $0 < p < 1$, is defined as the optimization problem

$$\min_{\boldsymbol{x}} \|\boldsymbol{y} - \boldsymbol{M}\boldsymbol{x}\|_2^2 + \lambda \|\boldsymbol{x}\|_p^p, \quad \text{s.t. } \boldsymbol{x} \geq 0,\ \boldsymbol{a}\boldsymbol{x}_n = 1\ , \tag{2.3}$$

where $\boldsymbol{y}, \boldsymbol{x}, \boldsymbol{M}$ are defined as in (2.2), λ is the Lagrange multiplier that regulates the sparsity of the solution. It is possible to use the IRLS method to rewrite the Lasso objective as a weighted ridge regression one (e.g. [104]). Then (2.3) becomes

$$\min_{\boldsymbol{x}} \|\boldsymbol{y} - \boldsymbol{M}\boldsymbol{x}\|_2^2 + \lambda \|\boldsymbol{D}\boldsymbol{x}\|_2^2\ , \tag{2.4}$$

$$\text{where} \quad \boldsymbol{D} = \operatorname{diag}\left(\boldsymbol{d}^{\frac{1}{2}}\right),\ \boldsymbol{d} = \left(\left(\boldsymbol{x}^{(\alpha-1)}\right)^2 + \boldsymbol{\epsilon}^2\right)^{\frac{p}{2}-1}\ .$$

where the weights $\boldsymbol{d}$ are calculated using the $\boldsymbol{x}^{(\alpha-1)}$ results from the previous iteration $\alpha - 1$. Hence the weighted ℓ_2 norm in (2.4) is a first-order approximation to the ℓ_p norm in (2.3). Here, $\boldsymbol{D}$ is a positive diagonal matrix; $\boldsymbol{\epsilon}$ measures the error in the solution for the pixels in the whole image, and goes to zero as the IRLS algorithm converges. $\boldsymbol{\epsilon}_n$ for the nth pixel is a vector of size $N \times 1$, $\boldsymbol{\epsilon}$ covers the whole image s.t. $\left[\boldsymbol{\epsilon}_1^\top, \boldsymbol{\epsilon}_2^\top, \ldots, \boldsymbol{\epsilon}_P^\top\right]^\top$, size $NP \times 1$. When the algorithm converges then $\boldsymbol{\epsilon} \to 0$ and $\boldsymbol{x}^{(\alpha-1)} \cong \boldsymbol{x}^{(\alpha)}$. It follows that $\|\boldsymbol{D}\boldsymbol{x}\|_2^2 = \sum_n \left(\left(x_n^2\right)^{\frac{p-2}{2}}\right)^{\frac{1}{2}\cdot 2} \cdot x_n^2 = \sum_n x_n^p = \|\boldsymbol{x}\|_p^p$ if the two conditions hold.

The IRLS algorithm proceeds by initializing the weights with ones and then iterating between: (i) solving the quadratic problem in (2.4) and (ii) updating the weight matrix $\boldsymbol{D}$. This is a fixed-point iteration for solving the optimization problem (2.3). The iterative update leads to an IRLS algorithm that guarantees convergence [45]. The IRLS enjoys two very attractive properties. It is simple but it is also very flexible [105]. As will be seen in the next section, it can very easily accommodate additional p-norm penalty terms, such as a total-variation regularizer, in the objective function.

Although the ℓ_p minimization problem is non-convex, the minimization can still be carried out by an IRLS algorithm. Daubechies showed that IRLS can solve such a non-convex problem and successfully recover sparse solutions [34]. In addition, she also showed that the local convergence of the algorithm is super-linear when certain null space properties on the matrix $\boldsymbol{M}$ hold. Indeed, the convergence is very fast compared to other algorithms like ADMM and gradient decent. The traditional IRLS algorithms contain a large scale inversion operation at every iteration, which makes the method computationally expensive. The complexity of the least squares problem at each iteration is $\mathcal{O}(LN^2)$ [106]. However, due to the extremely fast convergence of the algorithm, IRLS needs to do only a small number of iterations, and hence often requires less computational time than alternative methods.

2.3 Total Variation for Sparse Unmixing

The least squares objective function implies that the dependencies between pixels and between spectral bands were ignored. Therefore, adding total variation regularization to the objective function can explain these spatial dependencies and improve the performance of unmixing. However, the assumption of homogenous background noise levels remains.

We propose the constrained, sparse, total variation regularization problem

$$\arg\min_{\boldsymbol{x}} \|\boldsymbol{y} - \boldsymbol{M}\boldsymbol{x}\|_2^2 + \lambda \|\boldsymbol{x}\|_p^p + \lambda_{TV} \|\nabla \boldsymbol{x}\|_p^p \ , \tag{2.5}$$

with the constraints $\boldsymbol{x} \geq 0$ and $\boldsymbol{a}\boldsymbol{x}_n = 1$ for $n = 1, \ldots, P$, where $\boldsymbol{a} = [1, \ldots, 1] \in \mathbb{R}^N$ is a vector of N ones , and where $\boldsymbol{y}, \boldsymbol{x}, \boldsymbol{M}$ are defined as in (2.2).

The TV term is equivalent to placing a hyper-Laplacian prior on the horizontal and vertical increments of the abundances. The parameter λ_{TV} balances the influence of this prior against the sparsity and likelihood terms; larger values will result in smoother solutions. The differencing operator ∇ computes the spatial increments of the fractional abundances, i.e. differences of abundances over neighboring pixels. It can be decomposed into horizontal ∇_h and vertical ∇_v differences so that $\nabla = [\nabla_h^\top, \nabla_v^\top]^\top$. The differences between horizontal neighbors are $\nabla_h \mathbf{x} = [\mathbf{a_1}, \mathbf{a_2}, ..., \mathbf{a_{T_h}}]^\top$, where $\mathbf{a_n} = \mathbf{x_n} - \mathbf{x_{n_h}}$, with n and n_h indicating a pixel and its horizontal neighbor, and T_h is the number of horizontal differences taken for the image. ∇_v performs similarly for vertical neighbors taking T_v differences. The difference operator ∇ is mostly composed of zeros, with only one 1 and -1 in each row for the respective vertical or horizontal endmembers the differences are taken in between. The distance between the 1 and -1 in the rows of ∇_h is N, hence the difference is taken in between $x_n[i]$ and $x_{n+1}[i]$ for endmember i and pixels n and $n+1$. For ∇_v the distance is cN, where c indicates the number of pixels in each row of the image. In order to solve the optimization problem in (2.5) with the IRLS algorithm, the p-norm

is rewritten in terms of a weighted 2-norm, to get:

$$\begin{aligned} \arg\min_{\boldsymbol{x}} \quad & \|\boldsymbol{y}-\boldsymbol{M}\boldsymbol{x}\|_2^2+\lambda\|\boldsymbol{D}\boldsymbol{x}\|_2^2+\lambda_{TV}\left\|\boldsymbol{\Phi}\nabla\boldsymbol{x}\right\|_2^2, \\ \text{with} \quad & \boldsymbol{D}=\operatorname{diag}(\boldsymbol{d}^{\frac{1}{2}}), \quad \boldsymbol{d}=\left(\left(\boldsymbol{x}^{(\alpha-1)}\right)^2+\boldsymbol{\epsilon}^2\right)^{\frac{p}{2}-1}, \\ & \boldsymbol{\Phi}=\operatorname{diag}(\boldsymbol{\phi}^{\frac{1}{2}}), \quad \boldsymbol{\phi}=\left(\left(\nabla\boldsymbol{x}^{(\alpha-1)}\right)^2+\boldsymbol{\eta}^2\right)^{\frac{p}{2}-1}, \end{aligned} \tag{2.6}$$

where the weights $\boldsymbol{d}$ and $\boldsymbol{\phi}$ are calculated using the $\boldsymbol{x}^{(\alpha-1)}$ results from the previous iteration $\alpha-1$. Here, $\boldsymbol{\Phi}$ is a positive diagonal matrix containing the weights, the vector $\boldsymbol{\eta}$ is a column vector of size $NP\times 1$ filled with the constant $\eta=10^{-6}$. It is included in order to avoid division by zeros. When the algorithm converges, then $\boldsymbol{\epsilon}\to 0$, $\boldsymbol{\eta}\to 0$ and $\boldsymbol{x}^{(\alpha-1)}\approx\boldsymbol{x}^{(\alpha)}$, consequently we have that the weighted ℓ_2 norm is equivalent to ℓ_p norm: $\left\|\boldsymbol{\Phi}\nabla\boldsymbol{x}\right\|_2^2=\|\nabla\boldsymbol{x}\|_p^p$.

2.4 IRLSTV unmixing algorithm

2.4.1 Numerical algorithm

The spatial regularization extension is incorporated into the IRLS algorithm via a total variation regularizer. The resulting IRLSTV pseudocode is presented in Algorithm 1. The IRLSTV uses a weighted ℓ_2 norm to estimate the ℓ_p norm with $0<p<1$. The weights are defined as positive diagonal matrices with the values $\left(\left(\boldsymbol{x}^{(\alpha-1)}\right)^2+\boldsymbol{\epsilon}^2\right)^{(p/2-1)/2}$ and $\left(\left(\nabla\boldsymbol{x}^{(\alpha-1)}\right)^2+\boldsymbol{\eta}^2\right)^{(p/2-1)/2}$ respectively on the diagonal. This is fixed-point iteration using the iterate $\boldsymbol{x}^{(\alpha-1)}$ to update the weights for iteration α. When $\boldsymbol{\epsilon}=\boldsymbol{\eta}=0$ and $\boldsymbol{x}^{(\alpha-1)}=\boldsymbol{x}^{(\alpha)}$then we have that $\|\boldsymbol{x}\|_p^p+\|\nabla\boldsymbol{x}\|_p^p=\|\boldsymbol{D}\boldsymbol{x}\|_2^2+\left\|\boldsymbol{\Phi}\nabla\boldsymbol{x}\right\|_2^2$.

The function $\operatorname{sort}(\boldsymbol{x}_n)_{q+1}$ in step 11 of the algorithm rearranges the absolute values of each pixel $\boldsymbol{x}_n$ into a decreasing sequence of numbers and selects the $q+1^{th}$ value for some fixed integer q. Hence $\operatorname{sort}(\boldsymbol{x}_i)_{q+1}$ outputs the $q+1^{th}$ largest value in $\boldsymbol{x}_i$, $i=1,\ldots,P$. By definition, a vector $\boldsymbol{x}$ is *q-sparse* if and only if $r(\boldsymbol{x})_q=0$ [34]. Thus the function $\operatorname{sort}(\boldsymbol{x}_i)_{q+1}$ conveys information about how sparse each estimated solution vector $\boldsymbol{x}_n$ is into the value of $\boldsymbol{\epsilon}_n$ and to the updated weights d_n at each iteration.

The vector of residuals $\boldsymbol{\epsilon}$, together with the estimated solution, are used to update the weights for the sparsity inducing term $\|\boldsymbol{D}\boldsymbol{x}\|_2^2$. In addition, the estimated solution is used to calculate the differences between the fractional abundances of neighboring pixels and give weights to the spatial regularization term in steps 8 and 9. Both of these new weights are used in the following iteration for the estimation of the solution (step 10). This process is repeated until all $\boldsymbol{\epsilon}_n$ for $n=1,\ldots,P$ become smaller than the set threshold $\boldsymbol{\epsilon}_{thr}$. In that case, the algorithm has found an optimal solution of the optimization problem (2.5).

Algorithm 1 Pseudocode of the IRLS $\ell_p - \ell_2$ optimization algorithm with spatial total variation regularization (IRLSTV).

Task: Solve $\|\boldsymbol{y} - \boldsymbol{M}\boldsymbol{x}\|_2^2 + \lambda \|\boldsymbol{x}\|_p^p + \lambda_{TV} \|\nabla \boldsymbol{x}\|_p^p$, s.t. $\boldsymbol{x} \geq 0, \boldsymbol{a}\boldsymbol{x}_n = 1$. Find $\boldsymbol{x}$.
Parameters: ϵ_{thr} : convergence threshold, j_{max} : maximum number of iterations.
Input: $\boldsymbol{M}_0$: $L \times N$ spectral library, $\boldsymbol{Y}$: $L \times P$ hyperspectral data, λ : sparsity regularization parameter, λ_{TV} : spatial regularization parameter, p : quasi-norm.
Output: $\boldsymbol{X}$: $N \times P$ fractional abundance matrix w.r.t. $\boldsymbol{M}_0$.

Initialization:
1: $\boldsymbol{\epsilon} \leftarrow [1, ..., 1], \quad \boldsymbol{\epsilon}_{thr} \leftarrow [\epsilon_{thr}, ..., \epsilon_{thr}]$.
2: $\boldsymbol{d} \leftarrow [1, ..., 1], \quad \boldsymbol{D} = \text{diag}(\boldsymbol{d}^{\frac{1}{2}})$.
3: $\boldsymbol{x} \leftarrow \arg\min_{\boldsymbol{x}} \|\boldsymbol{y} - \boldsymbol{M}\boldsymbol{x}\|_2^2 + \lambda \|\boldsymbol{D}\boldsymbol{x}\|_2^2$
s.t. $\boldsymbol{x} \geq 0, \quad \boldsymbol{a}\boldsymbol{x}_n = 1$ for $n = 1, ..., P$.
4: $j = 1$.
Main iteration for iteration α:
5: **while** $\text{sum}(\boldsymbol{\epsilon} > \boldsymbol{\epsilon}_{thr}) > 0$ **do**
6: $\quad \boldsymbol{d} = \left(\left(\boldsymbol{x}^{(\alpha-1)} \right)^2 + \boldsymbol{\epsilon}^2 \right)^{\frac{p}{2}-1}$.
7: $\quad \boldsymbol{D} = \text{diag}(\boldsymbol{d}^{\frac{1}{2}})$.
8: $\quad \boldsymbol{\phi} = \left(\left(\nabla \boldsymbol{x}^{(\alpha-1)} \right)^2 + \boldsymbol{\eta}^2 \right)^{\frac{p}{2}-1}$.
9: $\quad \boldsymbol{\Phi} = \text{diag}(\boldsymbol{\phi}^{\frac{1}{2}})$.
10: $\quad \boldsymbol{x}^{(\alpha)} \leftarrow \arg\min_{\boldsymbol{x}} \|\boldsymbol{y} - \boldsymbol{M}\boldsymbol{x}\|_2^2 + \lambda \|\boldsymbol{D}\boldsymbol{x}\|_2^2 + \lambda_{\text{TV}} \left\|\boldsymbol{\Phi}\nabla \boldsymbol{x}\right\|_2^2$
s.t. $\boldsymbol{x} \geq 0, \quad \boldsymbol{a}\boldsymbol{x}_n = 1$ for $n = 1, ..., P$.
11: $\quad \boldsymbol{\epsilon}_n = \min\left(\boldsymbol{\epsilon}_n, \frac{\text{sort}(\boldsymbol{x}_n)_{q+1}}{N}\right)$.
12: $\quad$ **if** $j > j_{max}$ **then**
13: $\quad\quad$ break;
14: $\quad$ **end if**
15: $\quad j = j + 1$.
16: **end while**

Please note that in the first iteration of the IRLSTV algorithm, the solution is estimated without the spatial information (step 3) as we cannot calculate the differences between neighboring pixels without an initial estimate of the abundances.

Step 10 in Algorithm 1 is an optimization problem with a quadratic objective and linear constraints. We can adapt standard quadratic programming techniques, such as those described in [51], to solve it. We rewrite $\|\boldsymbol{y} - \boldsymbol{M}\boldsymbol{x}\|_2^2 + \lambda \|\boldsymbol{D}\boldsymbol{x}\|_2^2 + \lambda_{TV} \left\|\boldsymbol{\Phi}\nabla \boldsymbol{x}\right\|_2^2$ as

$$\begin{aligned} f(\boldsymbol{x}) &= (\boldsymbol{y} - \boldsymbol{M}\boldsymbol{x})^\top (\boldsymbol{y} - \boldsymbol{M}\boldsymbol{x}) + \lambda (\boldsymbol{D}\boldsymbol{x})^\top (\boldsymbol{D}\boldsymbol{x}) + \lambda_{TV} (\boldsymbol{\Phi}\nabla \boldsymbol{x})^\top (\boldsymbol{\Phi}\nabla \boldsymbol{x}) \\ &= \boldsymbol{y}^\top \boldsymbol{y} + \boldsymbol{x}^\top \left(\boldsymbol{M}^\top \boldsymbol{M}\right) \boldsymbol{x} - 2\boldsymbol{y}^\top \boldsymbol{M}\boldsymbol{x} + \lambda \boldsymbol{x}^\top \boldsymbol{D}^\top \boldsymbol{D}\boldsymbol{x} \\ &\quad + \lambda_{TV} \boldsymbol{x}^\top \nabla^\top \boldsymbol{\Phi}^\top \boldsymbol{\Phi}\nabla \boldsymbol{x} \,, \end{aligned} \tag{2.7}$$

where $\boldsymbol{\Phi}$ is a diagonal matrix containing the weights of the TV term. Here the weight matrices $\boldsymbol{D}$ and $\boldsymbol{\Phi}$ are kept fixed since they are computed using the previous iteration

value $\boldsymbol{x}^{(\alpha-1)}$. Thus the partial derivatives of $f(\boldsymbol{x})$ with respect to $\boldsymbol{x}$ are

$$\begin{aligned}\frac{\partial f(\boldsymbol{x})}{\partial \boldsymbol{x}} &= \left(\boldsymbol{M}^\top\boldsymbol{M} + \left(\boldsymbol{M}^\top\boldsymbol{M}\right)^\top\right)\boldsymbol{x} - 2\left(\boldsymbol{y}^\top\boldsymbol{M}\right)^\top + \lambda\left(\boldsymbol{D}^\top\boldsymbol{D} + \left(\boldsymbol{D}^\top\boldsymbol{D}\right)^\top\right)\boldsymbol{x} \\ &\quad + \lambda_{TV}\left(\nabla^\top\boldsymbol{\Phi}^\top\boldsymbol{\Phi}\nabla + \left(\nabla^\top\boldsymbol{\Phi}^\top\boldsymbol{\Phi}\nabla\right)^\top\right)\boldsymbol{x} \\ &= 2\boldsymbol{M}^\top\boldsymbol{M}\boldsymbol{x} - 2\boldsymbol{M}^\top\boldsymbol{y} + 2\lambda\boldsymbol{D}^\top\boldsymbol{D}\boldsymbol{x} + 2\lambda_{TV}\nabla^\top\boldsymbol{\Phi}^\top\boldsymbol{\Phi}\nabla\boldsymbol{x}\,. \end{aligned} \tag{2.8}$$

After equating (2.8) to zero and expressing $\boldsymbol{x}$ we get that:

$$\hat{\boldsymbol{x}} = \left(\boldsymbol{M}^\top\boldsymbol{M} + \lambda\boldsymbol{D}^\top\boldsymbol{D} + \lambda_{TV}\nabla^\top\boldsymbol{\Phi}^\top\boldsymbol{\Phi}\nabla\right)^+\boldsymbol{M}^\top\boldsymbol{y}, \tag{2.9}$$

where $\boldsymbol{A}^+$ denotes a pseudo-inverse of matrix $\boldsymbol{A}$ and $\hat{\boldsymbol{x}}$ is the estimated solution to step 10 in the IRLSTV algorithm. The non-negativity constraint is enforced on the solution by a projection onto the non-negative orthant by keeping only the positive values, and the sum-to-one constraint is enforced by renormalization of the estimated solution for all pixels at each iteration. Although the constraints are applied in a heuristic manner, they integrate well into our iterative framework and give the correct feedback to the algorithm producing great results. Following the derivation to reach Eq. (2.9) for the full model, we can express $\boldsymbol{x}$ with only the sparsity regularization in order to do the initialization in step 3. We get that

$$\hat{\boldsymbol{x}} = \left(\boldsymbol{M}^\top\boldsymbol{M} + \lambda\boldsymbol{D}^\top\boldsymbol{D}\right)^+\boldsymbol{M}^\top\boldsymbol{y}. \tag{2.10}$$

Here, the constraints are enforced in the same manner as for the full model.

The parameters p, λ and λ_{TV} are chosen using five-fold cross-validation over a range of possible parameter values. Note that the solutions for $\boldsymbol{M}^\top\boldsymbol{M}$ and $\boldsymbol{M}^\top\boldsymbol{y}$ do not change within the algorithm, hence these calculations are precomputed at the start of the algorithm, outside the while loop. There is an analytical way of including the sum-to-one constraint in the optimization (see Section 2.5); however, it makes the optimization problem more complex.

2.4.2 MUSIC library reduction

Sparse unmixing has become very popular recently as it overcomes the problem of needing pure spectral pixels in a given scene [28,36]. Nevertheless, the high mutual coherence (often close to one [72]) of the always expanding spectral libraries limits the success of the sparse unmixing methods considerably [76]. In general, the higher the mutual coherence is, the lower the level of sparseness, which affects the performance of unmixing. This method will attempt to overcome the high mutual coherence problem and complexity of computation by introducing a library reduction step before performing sparse unmixing via IRLS with TV regularization. We will use the characteristic quality of hyperspectral data that the

amount of endmembers in a scene is much smaller than the number of signatures in a spectral library. This allows us to reduce the spectral library size, decrease the size of the estimated fractional abundances vector $\boldsymbol{x}$, lessen the mutual coherence of the spectral library, lower the computational complexity and shorten the computational time at the same time.

We propose a sparse hyperspectral unmixing method using library pruning via MUSIC-Hysime, where the HySime algorithm estimates the number of endmembers in the image and MUSIC determines the endmembers from a spectral library, and unmixing via IRLSTV, see Alg. 2.

Algorithm 2 Pseudocode of the MUSIC-HySime library reduction algorithm.

Parameters: ϵ_{thr} : convergence threshold, j_{max} : maximum number of iterations.
Input: $\boldsymbol{Y}$: $L \times P$ hyperspectral data,$\boldsymbol{M}_0$: $L \times N$ spectral library, λ : sparsity regularization parameter, λ_{TV} : spatial regularization parameter, p : quasi-norm.
Output: $\boldsymbol{X}$: $N \times P$ fractional abundance with respect to $\boldsymbol{M}_0$.

1: $\hat{\boldsymbol{\xi}} = \text{Noise estimation}(\boldsymbol{Y})$
2: $\hat{\boldsymbol{U}} = \text{HySime}(\boldsymbol{Y}, \hat{\boldsymbol{\xi}})$
Main iteration:
3: $\boldsymbol{P}^{\perp}_{\boldsymbol{M}_S} = \boldsymbol{I} - \hat{\boldsymbol{U}}\hat{\boldsymbol{U}}^{\top}$
4: **for** $j = 1$ to N **do**
5: $\quad \varepsilon_j = \frac{\left\| \boldsymbol{P}^{\perp}_{\boldsymbol{M}_S} \boldsymbol{m}_j \right\|_2}{\|\boldsymbol{m}_j\|_2}$
6: **end for**
7: $\pi = \text{permutation } \left\{1, ..., n : \varepsilon_{\pi(i)} \leq \varepsilon_{\pi(j)}, i \leq j\right\}$
8: $R = \{\pi(i), i = 1, ..., r\}$.
9: Solve $\|\boldsymbol{y} - \boldsymbol{M}_R \boldsymbol{x}_R\|_2^2 + \lambda \|\boldsymbol{x}_R\|_p^p + \lambda_{TV} \|\nabla \boldsymbol{x}_R\|_p^p$,
subject to $\quad \boldsymbol{x}_R \geq 0, \quad \boldsymbol{a}\boldsymbol{x}_n = 1$ for $n = 1, ..., P$
using the IRLSTV algorithm in Alg. 1.

Steps 1-8 of Alg. 2 are the pruning part of MUSIC-CSR [76]. Step 1 and 2 estimate an orthonormal basis for range($\boldsymbol{M}_S$) using HySime algorithm [100]. Step 3 computes the projection matrix on range($\boldsymbol{M}_S$). Step 5 calculates the distance from each member of the library to the estimated subspace using the normalized Euclidean distance. Steps 7 and 8 sort the normalized projection errors in an increasing order and retain the indexes of the first r in the set R. The reduced library size r is chosen manually using experience in unmixing hyperspectral images. It should be as small as possible while making sure that all endmembers in the image are retained. Step 9 performs the sparse unmixing process via IRLS with spatial total variation regularization introduced in Section 2.3. The sum-to-one constraint is enforced by renormalization and the non-negativity constraint by hard-thresholding. The IRLSTV algorithm takes the reduced spectral library $\boldsymbol{M}_R \in \mathbb{R}^{L \times r}$ as input instead of the full spectral library $\boldsymbol{M}_0$ of size $L \times N$.

2.5 Full additivity

There is an alternative way of enforcing the sum-to-one constraint on the regression problem with the spatial TV regularization displayed in (2.5). Similarly to the way it is done for only the likelihood function [78, 107–109], we can use standard Lagrangian analysis to minimize (2.5) with the generalized linear constraint $g(\boldsymbol{x}) = 0$, where $g(\boldsymbol{x}) = \boldsymbol{A}\boldsymbol{x} - \boldsymbol{b}$, s.t. $\boldsymbol{b} \in \mathbb{R}^{P\times 1}$ consists of ones. For the purpose of calculating the sum-to-one constraint in a vectorized form, we need to define $\boldsymbol{A}$ as a block-diagonal matrix of ones, where $\boldsymbol{A} \in \mathbb{R}^{P\times NP}$ has P blocks of the vector $\boldsymbol{a} = [1, 1, ...1] \in \mathbb{R}^N$ on the diagonal. The sum-to-one constraint will be enforced on each pixel $\boldsymbol{x}_n$ with $n = 1, ..., P$. We define the function $F(\boldsymbol{x}, \delta) = f(\boldsymbol{x}) + \delta g(\boldsymbol{x})$, where δ is the Lagrange multiplier. We get the equations:

$$\begin{aligned} f(\boldsymbol{x}) &= (\boldsymbol{y} - \boldsymbol{M}\boldsymbol{x})^\top (\boldsymbol{y} - \boldsymbol{M}\boldsymbol{x}) + \lambda (\boldsymbol{D}\boldsymbol{x})^\top (\boldsymbol{D}\boldsymbol{x}) + \lambda_{TV} (\boldsymbol{\Phi}\nabla\boldsymbol{x})^\top (\boldsymbol{\Phi}\nabla\boldsymbol{x}) \ , \\ g(\boldsymbol{x}) &= \boldsymbol{A}\boldsymbol{x} - \boldsymbol{b} = 0 \ , \\ F(\boldsymbol{x}, \delta) &= \boldsymbol{y}^\top\boldsymbol{y} + \boldsymbol{x}^\top\boldsymbol{M}^\top\boldsymbol{M}\boldsymbol{x} - 2\boldsymbol{y}^\top\boldsymbol{M}\boldsymbol{x} + \lambda\boldsymbol{x}^\top\boldsymbol{D}^\top\boldsymbol{D}\boldsymbol{x} \\ &\quad + \lambda_{TV}\boldsymbol{x}^\top\nabla^\top\boldsymbol{\Phi}^\top\boldsymbol{\Phi}\nabla\boldsymbol{x} + \delta\boldsymbol{A}\boldsymbol{x} - \delta\boldsymbol{b} \ . \end{aligned} \tag{2.11}$$

First, we find the partial derivatives with respect to $\boldsymbol{x}$ using $\frac{\partial f(\boldsymbol{x})}{\partial \boldsymbol{x}}$ from (2.8),

$$\frac{\partial F(\boldsymbol{x}, \delta)}{\partial \boldsymbol{x}} = 2\boldsymbol{M}^\top\boldsymbol{M}\boldsymbol{x} - 2\boldsymbol{M}^\top\boldsymbol{y} + 2\lambda\boldsymbol{D}^\top\boldsymbol{D}\boldsymbol{x} + 2\lambda_{TV}\nabla^\top\boldsymbol{\Phi}^\top\boldsymbol{\Phi}\nabla\boldsymbol{x} + \boldsymbol{A}^\top\delta \ . \tag{2.12}$$

Then we equate the result seen in (2.12) to zero in order to locate the stationary points of $F(\boldsymbol{x})$. The equation $2\boldsymbol{M}^\top\boldsymbol{M}\boldsymbol{x} + 2\lambda\boldsymbol{D}^\top\boldsymbol{D}\boldsymbol{x} + 2\lambda_{TV}\nabla^\top\boldsymbol{\Phi}^\top\boldsymbol{\Phi}\nabla\boldsymbol{x} = 2\boldsymbol{M}^\top\boldsymbol{y} - \boldsymbol{A}^\top\delta$ gives:

$$\begin{aligned} \boldsymbol{x} &= \left(\boldsymbol{M}^\top\boldsymbol{M} + \lambda\boldsymbol{D}^\top\boldsymbol{D} + \lambda_{TV}\nabla^\top\boldsymbol{\Phi}^\top\boldsymbol{\Phi}\nabla\right)^+ \boldsymbol{M}^\top\boldsymbol{y} \\ &\quad - \left(\boldsymbol{M}^\top\boldsymbol{M} + \lambda\boldsymbol{D}^\top\boldsymbol{D} + \lambda_{TV}\nabla^\top\boldsymbol{\Phi}^\top\boldsymbol{\Phi}\nabla\right)^+ \boldsymbol{A}^\top\frac{\delta}{2} \\ &= \boldsymbol{Z}\boldsymbol{M}^\top\boldsymbol{y} - \boldsymbol{Z}\boldsymbol{A}^\top\frac{\delta}{2} \ . \end{aligned} \tag{2.13}$$

where $\boldsymbol{Z} = \left(\boldsymbol{M}^\top\boldsymbol{M} + \lambda\boldsymbol{D}^\top\boldsymbol{D} + \lambda_{TV}\nabla^\top\boldsymbol{\Phi}^\top\boldsymbol{\Phi}\nabla\right)^+$ and $\boldsymbol{B}^+$ indicates a pseudo-inverse of the matrix $\boldsymbol{B}$. We do the same for δ, and substitute (2.13) into the equation for $\boldsymbol{x}$:

$$\begin{aligned} \frac{\partial F(\boldsymbol{x}, \delta)}{\partial \delta} &= -g(\boldsymbol{x}) = -\boldsymbol{A}\boldsymbol{x} + \boldsymbol{b} = 0 \ , \\ \frac{\delta}{2} &= \left[\boldsymbol{A}\boldsymbol{Z}\boldsymbol{A}^\top\right]^+ \left(\boldsymbol{A}\boldsymbol{Z}\boldsymbol{M}^\top\boldsymbol{y} - \boldsymbol{b}\right) \ . \end{aligned} \tag{2.14}$$

We can find the sum-to-one constrained estimate for $\boldsymbol{x}$ by substituting (2.14) into (2.13):

$$\hat{\boldsymbol{x}} = \boldsymbol{Z}\boldsymbol{M}^\top \boldsymbol{y} - \boldsymbol{Z}\boldsymbol{A}^\top \left[\boldsymbol{A}\boldsymbol{Z}\boldsymbol{A}^\top\right]^+ \left(\boldsymbol{A}\boldsymbol{Z}\boldsymbol{M}^\top \boldsymbol{y} - \boldsymbol{b}\right) , \tag{2.15}$$

where $\boldsymbol{Z} = \left(\boldsymbol{M}^\top \boldsymbol{M} + \lambda \boldsymbol{D}^\top \boldsymbol{D} + \lambda_{TV} \nabla^\top \boldsymbol{\Phi}^\top \boldsymbol{\Phi} \nabla\right)^+$.

This result adds sparsity and spatial total variation extension to the solutions of sum-to-one constrained least squares seen in [78, 107–109]. We propose an algorithm with the above analytic sum-to-one constraint. We call it MUSIC-IRLSTV2 and use Alg. 2 to calculate it. The difference is that in Step 10 of Alg. 1 we use (2.15) to estimate $\hat{\boldsymbol{x}}$ instead of (2.9). The non-negativity constraint will be enforced on $\hat{\boldsymbol{x}}$ by projecting the result onto the non-negative orthant.

The proposed algorithms MUSIC-IRLSTV and MUSIC-IRLSTV2 differ from other sparse unmixing algorithms used in the literature. The comparison of the use of sparsity and spatial regularizer, library pruning and constraints can be seen in Table 2.1.

2.6 Experiments

In this section, we use the method of sparse unmixing via IRLS with library pruning and total variation spatial regularization on both simulated and real world data to evaluate the performance of the approach. The simulated data enables us to measure the performance quantitatively, and the real world data shows qualitative performance results.

2.6.1 Performance Discriminators

The classical performance discriminator often used in the literature is the root mean square error (RMSE) [110]. The RMSE is defined as RMSE= $\sqrt{\frac{1}{NP}\sum_{n=1}^{P} \|\boldsymbol{x}_n - \hat{\boldsymbol{x}}_n\|^2}$, where $\boldsymbol{x}$ denotes the true endmember fractional abundance values and $\hat{\boldsymbol{x}}$ the estimated endmember fractional abundance values. Here, a lower value indicates a better reconstruction of the image. A better way to assess the quality of reconstruction in the image is to use the signal to reconstruction error (SRE). SRE is measured in decibels: SRE(dB) $\equiv 10\log_{10}$(SRE), where SRE is defined as SRE $= \mathbb{E}\left[\|\boldsymbol{x}\|_2^2\right] / \mathbb{E}\left[\|\boldsymbol{x} - \hat{\boldsymbol{x}}\|_2^2\right]$. Higher values of the SRE(dB) indicate a superior unmixing performance. Note that in the rest of this paper we will use the term SRE for SRE(dB). We report both SRE and RMSE for comparison here. However, Iordache showed that the SRE error measure gives more information regarding the power of the error in relation with the power of the signal [28].

2.6.2 Experimental results

Simulated data

To test how well the proposed algorithm works for material identification in hyperspectral images, we used the USGS digital spectral library released in September 2007. It can be retrieved from the Spectroscopy Lab web page under the name splib06.[1] It consists of the spectral reflectance of hundreds of materials measured in the lab. These spectral reflectance values have 224 bands that lie uniformly in the interval 0.4-2.5 μm. Many spectral signatures in the library represent the same material with minor differences between them. We therefore followed the common preprocessing steps of e.g. [28] and pruned the library such that the minimum angle between any two signatures would not be smaller than 4.44°. The pruned library $\boldsymbol{M} \in \mathbb{R}^{224\times 240}$ includes 240 different signatures.

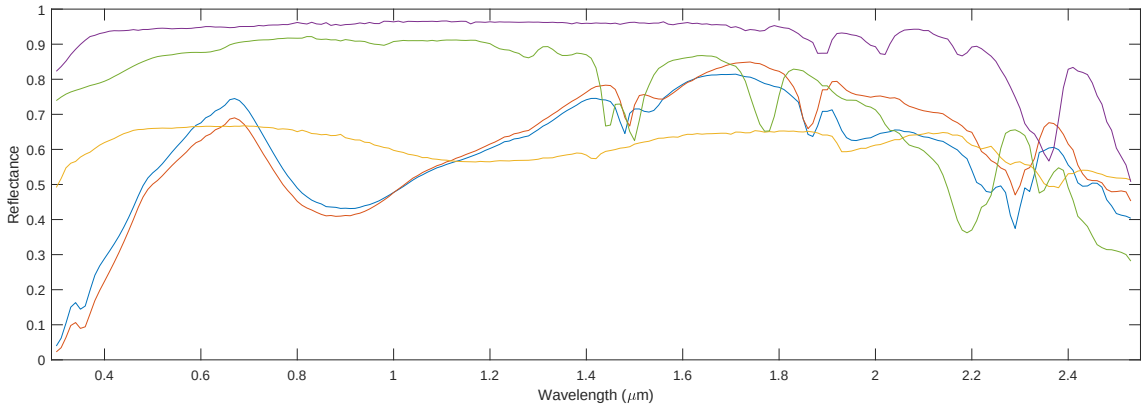


Figure 2.1: Five spectral endmember signatures chosen for the simulated hyperspectral image.

As in [28, 37] we created a 75×75 simulated image with 224 spectral bands for each pixel with five randomly selected spectral signatures from the library $\boldsymbol{M}$, Fig. 2.1. The data was generated using a linear mixing model, and the abundance to sum constraint was enforced on each pixel. The true abundances of the five endmembers in the simulated image can be seen in Fig. 2.2. The image has distinct square areas where there is only one endmember present (pure pixels) and areas that contain mixtures of two (fractional abundance of 1/2 each), three (1/3 each), four, and five endmembers. The background pixels surrounding the distinct square areas have abundance values of 0.1149, 0.0741, 0.2003, 0.2055 and 0.4051 respectively for the five endmembers. The simulated data were contaminated with white noise as well as spectrally correlated noise. The signal-to-noise ratio (SNR) was set to 40 dB [28](SNR $\equiv \mathbb{E}\left\|\boldsymbol{M}_0\boldsymbol{x}\right\|^2 / \mathbb{E}\left\|\boldsymbol{n}\right\|_2^2$) . To ensure that $\left\|\boldsymbol{D}\boldsymbol{x}\right\|_2^2$ is a good approximation of $\left\|\boldsymbol{x}\right\|_p^p$ we want $\boldsymbol{\epsilon}$ to approach zero. Therefore, our stopping rule ϵ_{thr} should be very small. We set the convergence threshold to $\epsilon_{thr} = 10^{-8}$, the maximum number of iterations to $j_{max} = 50$, and the number of endmembers after library pruning

[1] Available online at: http://speclab.cr.usgs.gov/spectral.lib06/.

to be 20 for these experiments.

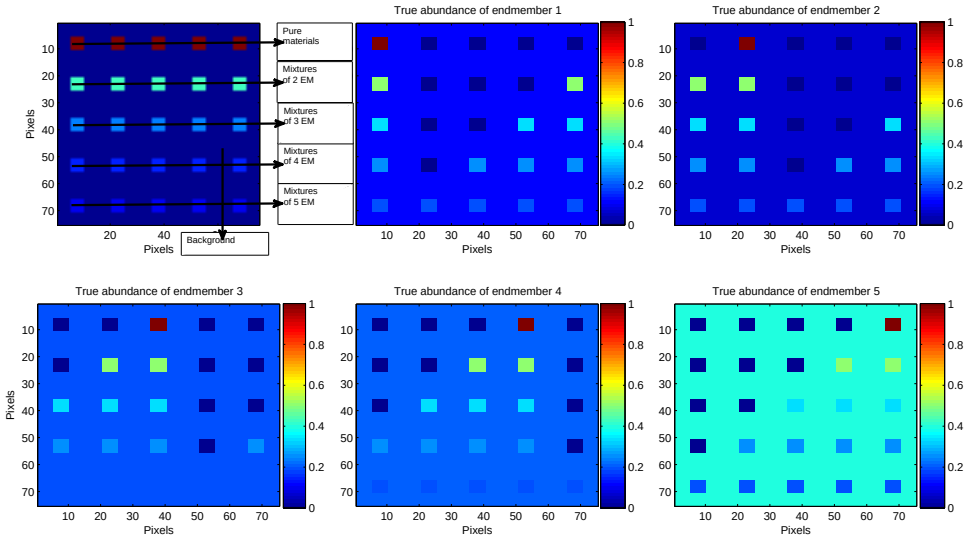


Figure 2.2: True fractional abundances of the five endmembers in the simulated image.

Parameter Study

The performance of the unmixing algorithm is affected by the choice of the parameter values. We have several adjustable regularization parameters in our model: λ, λ_{TV} and p. Parameters λ and p control the sparsity of the solution whereas λ_{TV} regulates the dependence of the abundance values in neighboring pixels. We have used the simulated data to assess the performance of the proposed unmixing algorithm MUSIC-IRLSTV for different parameter settings.

Fig. 2.3 shows the performance curve for a variety of λ values. Only a subset of p values are shown in the image for clearer understanding of the results. The MUSIC-IRLSTV algorithm did not produce good results for large λ ($\log_{10}(\lambda) = -1, -2$), when sparsity is strongly enforced. The sub-figures in Fig. 2.3 for λ smaller or equal to 10^{-3} show that there is a clear peak for λ_{TV} around 10^{-3} (for all p values), hence this may be the optimal λ_{TV} setting. Performance worsened for very large or small λ_{TV} values. Another valuable observation from Fig. 2.3 is that the proposed method acquired better performance and higher SRE values for p less than one. Hence using the ℓ_p norm instead of the traditional ℓ_1 norm is beneficial.

The Fig. 2.4a and 2.4b show the mean sparsity and SRE results calculated for $\log_{10}(\lambda)$ and $\log_{10}(\lambda_{TV})$ from -2 to -7 with an interval of -1. The mean SRE and mean sparsity are calculated over all pixels and over all different λ and λ_{TV} values jointly. Average sparsity in the results decreased as the p value got smaller and the highest SRE values were achieved with p close to 0.5. This extends similar convergence behavior and sparsity

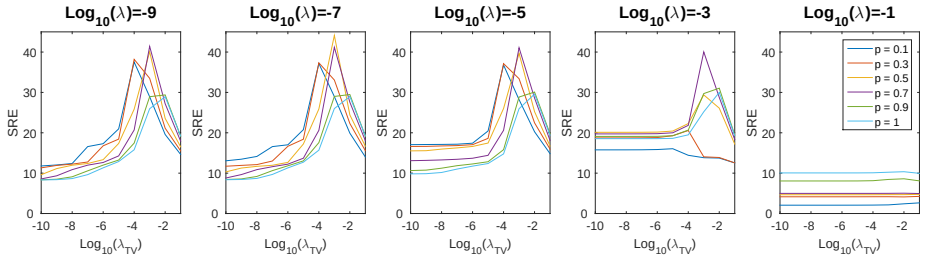
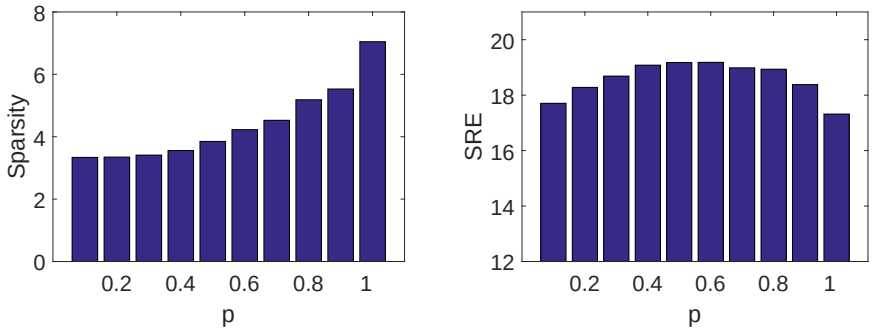


Figure 2.3: Performance of the MUSIC-IRLSTV algorithm with various λ and λ_{TV} values.

results of [35] to the spatially regularized case .

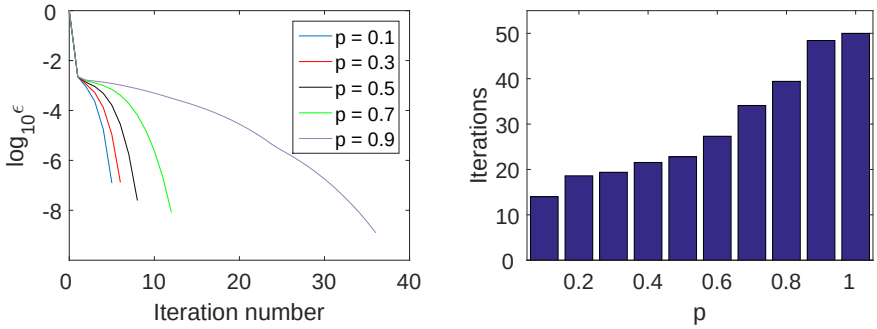
Convergence Analysis

The convergence of the proposed algorithm is of great interest to us. However the mathematical convergence of the algorithm is not straightforward. Hence we created a simulation to show empirical convergence of the algorithm. We performed the unmixing on the simulated image using the proposed algorithm for p values ranging from 0.1 to 1 with an interval of 0.1. We observed that the algorithm often converges before it reaches the maximum number of iterations $j_{max} = 50$. The average number of iterations it takes for the algorithm to converge is shown in Fig. 2.5b. The smaller the p value is the fewer iterations the algorithm takes on average to converge. The algorithm converged the slowest for $p = 1$. Fig. 2.5a depicts a convergence curve for the proposed unmixing method, i.e. the change of $\log_{10}(\boldsymbol{\epsilon})$ from one iteration to the next, where $\boldsymbol{\epsilon}$ is defined as in step 11 of Alg. 1. Epsilon conveys information about the sparsity of the result. We can see that the value of $\log_{10}(\boldsymbol{\epsilon})$ decreased faster for smaller p.



(a) (b)

Figure 2.4: Average results of the performance of MUSIC-IRLSTV algorithm on the simulated image for a subset of λ & λ_{TV} values, 10^{-2} to 10^{-7}. (a) Mean sparsity. (b) Mean SRE.



(a) (b)

Figure 2.5: Convergence behavior of MUSIC-IRLSTV algorithm. (a) Convergence curve over the number of iterations. (b) Average number of iterations for $0 < p \leq 1$.

Comparison with other algorithms

IRLS algorithm with constrained $\ell_p - \ell_2$ optimization has been shown to perform better than the Sparse Unmixing via variable Splitting and Augmented Lagrangian (SUnSAL) algorithm [35]. They both significantly outperform the traditional unmixing algorithms like fully constrained least squares [78], because they are based on sparse regression. The SUnSAL algorithm was later improved by adding a TV regularizer to the algorithm [37]. The performance of SUnSAL-TV was shown to perform better than SUnSAL and non-negatively constrained least squares (NCLS) algorithm with TV [37].

The simulated hyperspectral image in Fig. 2.2 was used to assess the performance of different sparse unmixing algorithms. Fig. 2.6 shows the abundance maps calculated using the optimal parameter values for λ, λ_{TV} and p by IRLS, SUnSAL-TV, MUSIC-CSR, RMUSIC-DANCER,MUSIC-IRLSTV, and RMUSIC-IRLSTV algorithms. We used five-fold cross-validation to find the optimal parameter values for all algorithms that are compared. Parameter values from -10 to 1 (with an interval of 1) were used for $\log_{10}(\lambda)$ and $\log_{10}(\lambda_{TV})$. Values from 0.1 to 1 (with an interval of 0.1) were used for p. The optimal parameter values can be seen in Table 2.3. Note that the names of the algorithms are shortened due to space constraints.

From Fig. 2.6 it can be seen that the use of TV norm improves the results. SUnSAL-TV, MUSIC-IRLSTV and RMUSIC-IRLSTV were visibly more accurate in finding the correct fractional abundances of the endmembers. The areas with high fractional abundance values were found more easily by the unmixing algorithms, whereas the regions with lower and very similar abundance fractions were more difficult to determine correctly. For endmember four, the difference of the true concentration of the background and the square boxes on the fifth line is only 0.0055. MUSIC-IRLSTV and RMUSIC-IRLSTV managed to find the correct fractional abundances in that region better than other methods.

The difference in the performance of algorithms with the increase in noise can be seen in Fig. 2.7 and Fig. 2.8. The signal to noise ratio (SNR) was lowered to 30dB

Table 2.2: Comparison of unmixing performance of MUSIC-IRLSTV at different noise levels

Method	SNR=40		SNR=30		SNR=20	
	SRE	RMSE	SRE	RMSE	SRE	RMSE
NCLS	5.50	16.79	0.91	24.74	-6.61	73.39
IRLS	8.59	11.48	5.60	15.68	1.53	24.60
SUnSAL	11.05	10.17	6.03	18.07	2.19	23.86
SUnSAL-TV	19.02	3.92	12.35	8.97	4.19	20.56
SUnSAL-CSR	12.85	8.57	6.45	16.96	2.68	23.32
MUSIC-CSR	22.11	2.48	12.35	7.27	4.74	16.85
RMUSIC-CSR	21.97	2.56	12.39	8.06	4.70	16.92
RMUSIC-DANSER	21.99	2.56	11.23	8.06	5.09	20.16
MUSIC-IRLSTV	**40.48**	**0.34**	**28.93**	**1.25**	19.25	3.20
MUSIC-IRLSTV2	40.08	0.34	25.80	1.87	18.33	4.92
RMUSIC-IRLSTV	39.22	0.38	27.66	1.54	**19.76**	**3.07**

Table 2.3: Parameter values

Method	SNR=40			SNR=30			SNR=20		
	λ	λ_{TV}	p	λ	λ_{TV}	p	λ	λ_{TV}	p
M-IRLSTV	10^{-4}	10^{-3}	0.6	10^{-3}	10^{-2}	0.8	10^{-5}	10^{-2}	0.5
M-IRLSTV2	10^{-7}	10^{-3}	0.6	10^{-6}	10^{-3}	0.6	10^{-6}	10^{-2}	0.5
RM-IRLSTV	10^{-3}	10^{-3}	0.6	10^{-3}	10^{-2}	0.7	10^{-6}	10^{-2}	0.5

and 20dB in these images respectively. As the noise increases the performance of all algorithms worsened as expected. However, the largest change was in IRLS and SUnSAL-TV. The result of MUSIC-CSR and RMUSIC-DANCER became very noisy. Whereas MUSIC-IRLSTV and RMUSIC-IRLSTV still performed the best. This can also be seen in quantitative measures of SRE and RMSE in Table 2.2.

Note that the signal-to-reconstruction error (SRE) that was calculated for each method of unmixing is truly an average. We used a 75×75 image for testing, which means that we estimated fractional abundance of the endmembers for 5625 pixels. In general, we calculated the SRE for the whole image, which is an average SRE of the 5625 pixels. However, to see the variance of the performance from pixel to pixel we calculated the SRE for individual pixels and hence acquired the sample standard deviation for each method. These can be seen in Fig. 2.9 together with the best SRE results for each method.

The traditional non-negatively constrained least squares (NCLS) method offered a poor solution compared to other algorithms. The results for the NCLS method were calculated using the SUnSAL algorithm with $\lambda = 0$. SUnSAL, SUnSAL-CSR and IRLS performed similarly, with IRLS giving a slightly higher mean SRE score but having a wider spread. SUnSAL-TV, MUSIC-CSR, RMUSIC-CSR, and RMUSIC-DANCER outperformed them. The proposed MUSIC-IRLSTV algorithm increased the mean individual SRE by another 35 and the MUSIC-IRLSTV2 by 40 points compared to SUnSAL-TV. RMUSIC-IRLSTV

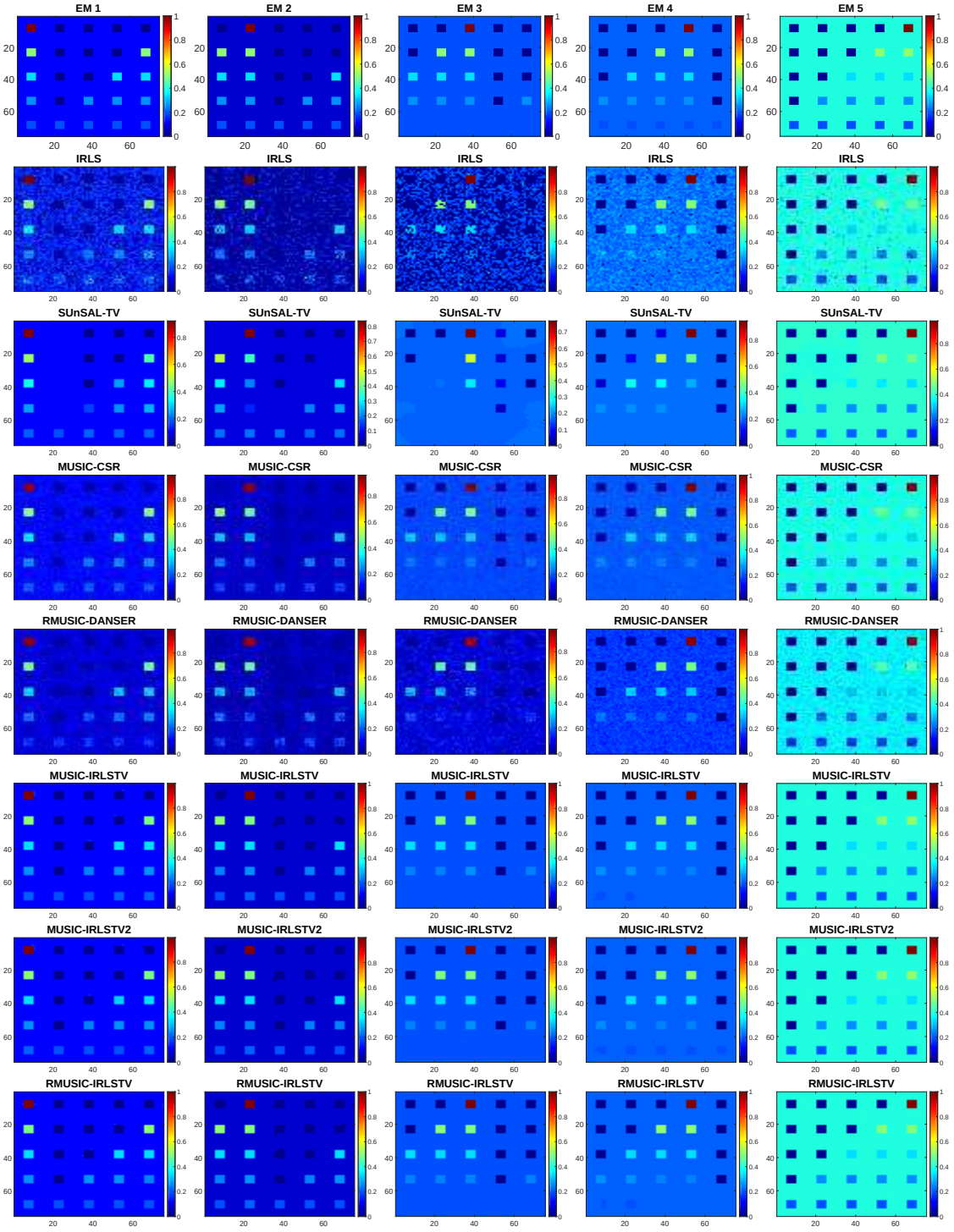


Figure 2.6: Fractional abundance maps obtained by IRLS, SUnSAL-TV, MUSIC-CSR, RMUSIC-DANCER, MUSIC-IRLSTV, MUSIC-IRLSTV2, and RMUSIC-IRLSTV for endmember (EM) 1 in (a), EM 2 in (b), EM 3 in (c), and EM 4 in (d). SNR=40dB.

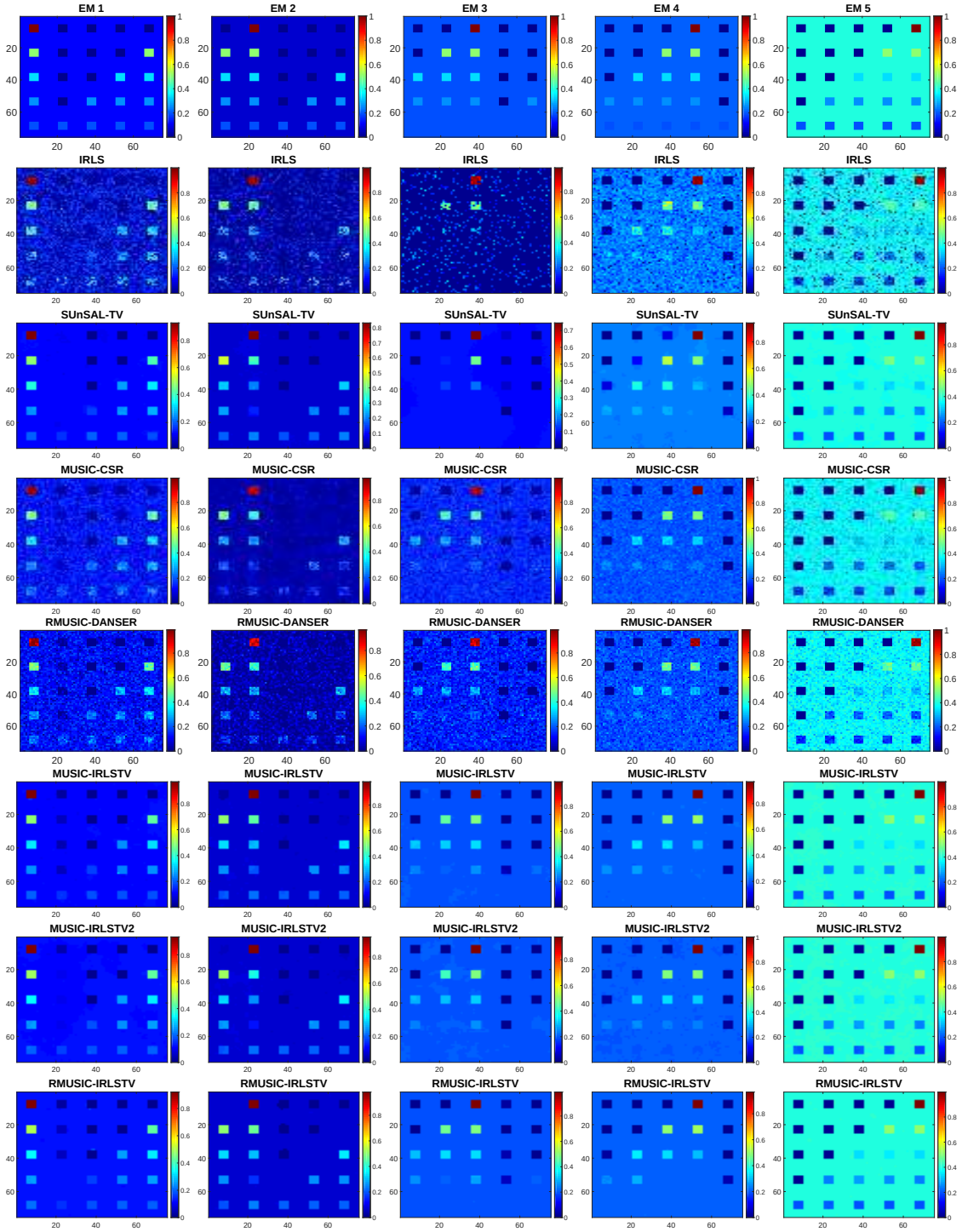
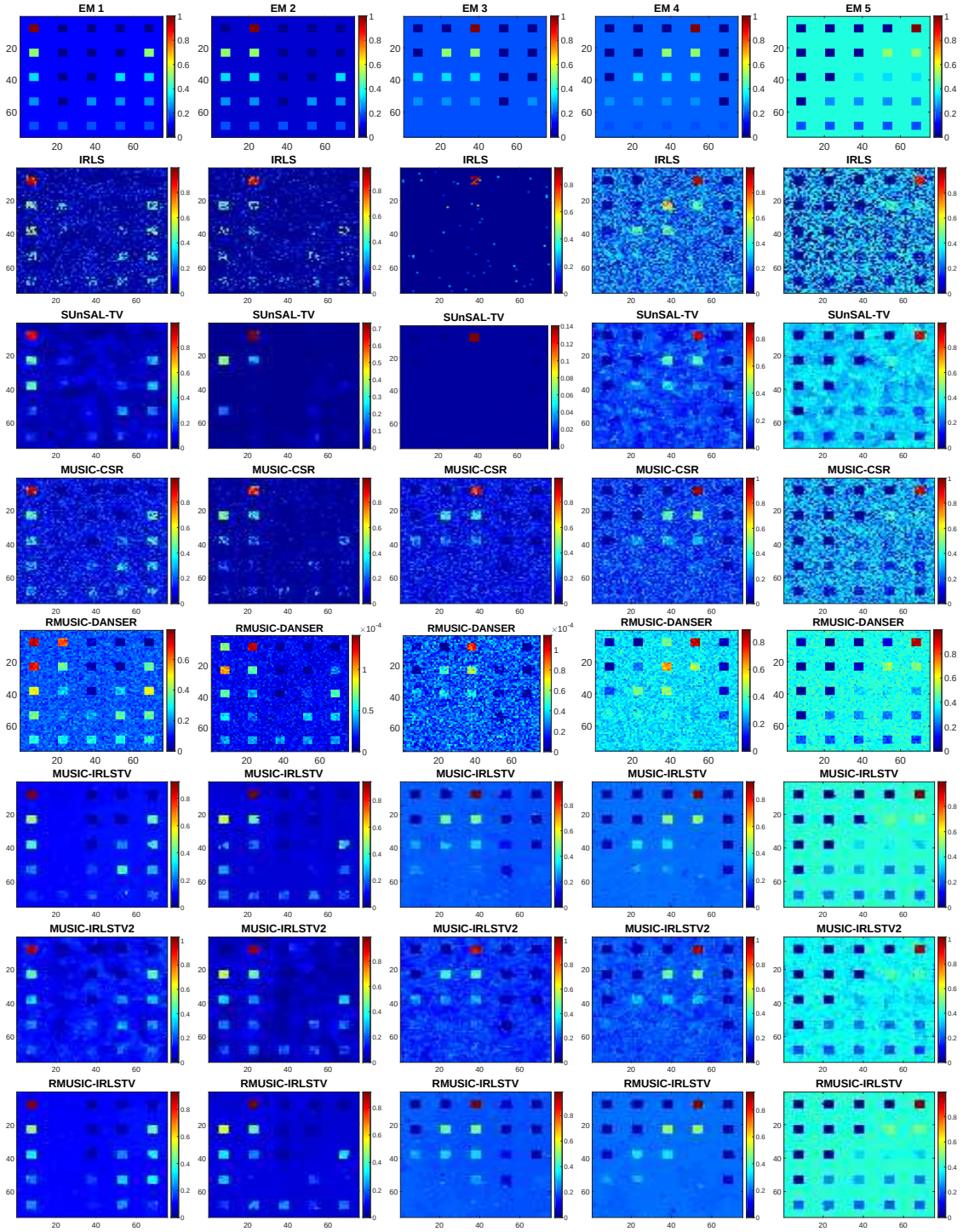


Figure 2.7: Fractional abundance maps obtained by IRLS, SUnSAL-TV, MUSIC-CSR, RMUSIC-DANCER, MUSIC-IRLSTV, MUSIC-IRLSTV2, and RMUSIC-IRLSTV for endmember (EM) 1 in (a), EM 2 in (b), EM 3 in (c), and EM 4 in (d). SNR=30dB.



(a) (b) (c) (d) (e)

Figure 2.8: Fractional abundance maps obtained by IRLS, SUnSAL-TV, MUSIC-CSR, RMUSIC-DANCER, MUSIC-IRLSTV, MUSIC-IRLSTV2,and RMUSIC-IRLSTV for end-member (EM) 1 in (a), EM 2 in (b), EM 3 in (c), and EM 4 in (d). SNR=20dB.

showed slightly worse results than MUSIC-IRLSTV in this case, but it was not far behind. Although the IRLS based methods had larger variance of SRE values over the image than the SUnSAL based methods (Fig. 2.9), the MUSIC-IRLSTV algorithm still provided a better performance in sparse spectral unmixing. The difference in the performance of MUSIC-IRLSTV and MUSIC-IRLSTV2 algorithm was not significant. MUSIC-IRLSTV had a higher mean SRE, but larger variance in the unmixing results from pixel to pixel.

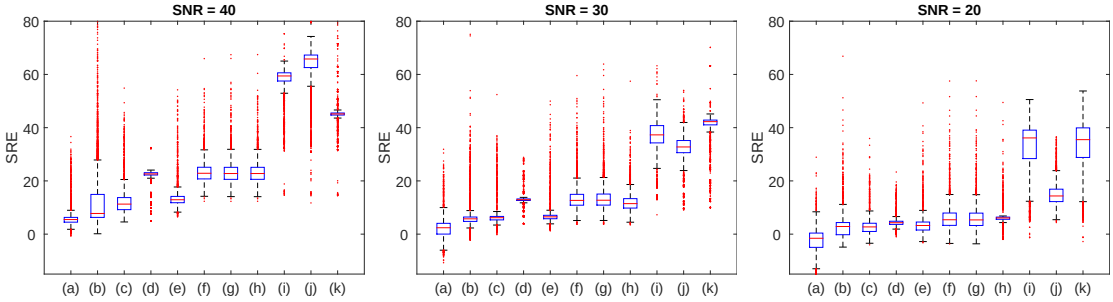


Figure 2.9: Individual SRE boxplots for (a) NCLS, (b) SUnSAL, (c) SUnSAL-TV, (d) SUnSAL-CSR, (e) MUSIC-CSR, (f) RMUSIC-CSR, (g) RMUSIC-DANCER, (h) IRLS, (i) MUSIC-IRLSTV, (j) MUSIC-IRLSTV2, (k) RMUSIC-IRLSTV.

Monte Carlo simulation

Since the signal to noise ratio (SNR) in the images generally affects the unmixing results, we performed the experiments using four different noise levels SNR $\in \{10, 20, 30, 40\}$. The number of endmembers present in the spectrum of a pixel can also influence the outcome. Hence, we considered three different levels of endmember quantities mixed in the pixel such that N $\in \{3, 6, 9\}$. We ran 10 Monte Carlo simulations for all these experiments to account for the randomness in the noise.

For the simulated data experiments, we used the endmember signatures from the USGS digital spectral library. It consists of the spectral reflectance of hundreds of materials measured in the lab. These spectral reflectance values have 224 bands that lie uniformly in the interval $0.4 - 2.5\mu$m. The chosen signatures can be seen in Fig. 2.10. We created a 50×50 image using the LMM in Eq.(1.1). Then we added Gaussian noise such that the noise level varied across the spectral bands because this kind of noise pattern is the most prevalent in real hyperspectral images [111]. The Dirichlet distribution was used for the uniform generation of the fractional abundance vectors $\boldsymbol{x}$ as in [97, 98]. For this experiment we did not use the MUSIC library reduction step for the IRSLTV algorithm as the number of endmembers was already low. Hence we will call the algorithm IRLSTV from now on.

The unmixing results for various SNR levels and different numbers of endmembers can be found in Tables 2.4−2.6. Both the SRE and RMSE values are reported together with their respective standard deviations for the simulation. The displayed performance

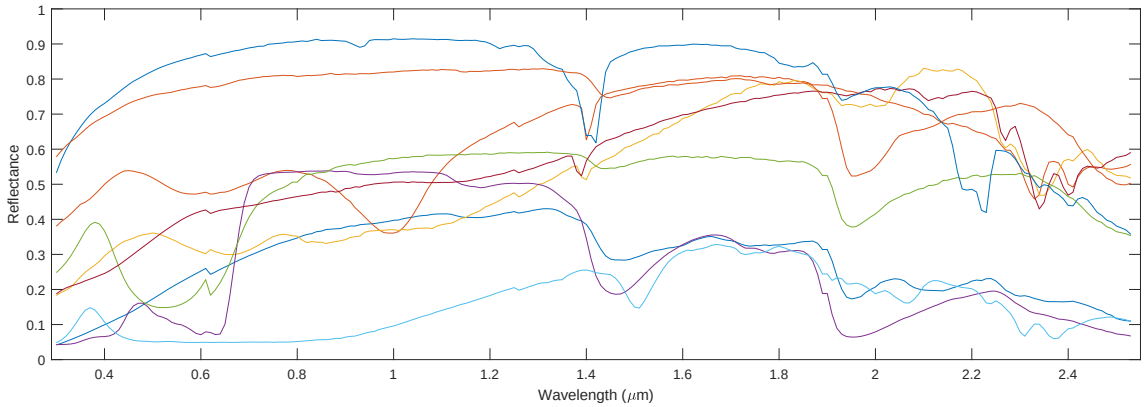


Figure 2.10: The USGS endmember signatures used in simulated data experiments.

measures are averages over 10 Monte-Carlo realizations for the simulated image. We can observe from the results that as the number of endmembers in a pixel went up and the amount of noise in the image increased, the performance of all algorithms deteriorated. SUnSAL and SUnSAL-TV gave consistently good unmixing results for different noise levels and endmember numbers. The performance of the IRLS and RMUSIC-DANSER algorithms was in general the lowest. However, RMUSIC-DANSER produced good results for high SNR and high number of endmembers. The proposed IRLSTV algorithm generated mostly the best reconstruction of the simulated hyperspectral image as measured by SRE and RMSE. The performance was very good in all noise environments and material densities. The average running times over 10 Monte-Carlo realizations of the simulated image can be seen in Table 2.7. It can be seen that the speed of IRLSTV is comparable to SUnSAL-TV, another method with spatial regularization.

German Alpine foothills image

A sub-image of the EnMAP data of German Alpine foothills image scene (Fig. 2.11) was used to test the performance of the proposed algorithm in a semi-supervised setting.[2] The image has 244 spectral bands and it covers the 420nm-2460nm spectral range [112]. We used the N-FINDR algorithm [20] to identify the spectral signatures of the endmembers in the image. It assumes the endmembers to be present in the image in the form of unmixed pixels [20]. The spectral channels of this image did not match with the available spectral libraries, hence we used an endmember extraction algorithm to identify material signatures from the data. The HySime algorithm (part of MUSIC library reduction algorithm) estimated the signal subspace in the sub-image to be 3. Therefore, we used the N-FINDR algorithm to find the 3 endmember signatures from the Alpine scene. The estimated fractional abundance maps of the Alpine scene can be seen in Fig. 2.12. The optimal parameter values were calculated using five-fold cross-validation, thus we have for

[2]Available online at: http://www.enmap.org/?q=node/21.

Table 2.4: Comparison of unmixing performance of IRLSTV with various SNR, N=3

Method	SNR=40				SNR=30			
	SRE	std	RMSE	std	SRE	std	RMSE	std
NCLS	32.62	0.28	0.52	0.02	25.97	0.65	1.29	0.13
IRLS	14.74	0.12	6.32	0.08	13.68	0.18	7.15	0.15
SUnSAL	36.09	0.34	0.48	0.02	28.85	0.78	1.17	0.12
SUnSAL-TV	35.65	0.33	0.50	0.02	28.80	0.75	1.17	0.11
MUSIC-CSR	28.21	0.12	1.25	0.02	26.29	0.33	1.54	0.06
RM-DANSER	13.74	0.07	5.87	0.12	13.71	0.08	5.90	0.12
IRLSTV	**37.85**	0.47	**0.42**	0.02	**29.47**	0.72	**1.13**	0.10
IRLSTV2	29.54	0.16	1.15	0.02	26.85	0.39	1.46	0.07
Method	SNR=20				SNR=10			
	SRE	std	RMSE	std	SRE	std	RMSE	std
NCLS	17.56	0.46	3.92	0.24	9.84	0.52	10.36	0.78
IRLS	9.34	0.36	11.90	0.53	8.95	0.45	11.98	0.71
SUnSAL	20.14	0.69	3.23	0.27	10.77	0.76	9.45	0.89
SUnSAL-TV	20.16	0.69	3.23	0.27	10.75	0.77	9.46	0.91
MUSIC-CSR	20.17	0.85	3.25	0.37	**11.16**	0.72	**9.31**	0.84
RM-DANSER	11.85	2.57	8.46	3.69	4.54	1.40	20.53	3.15
IRLSTV	**20.35**	0.81	**3.23**	0.30	10.86	0.68	9.41	0.76
IRLSTV2	20.11	0.37	3.30	0.15	10.37	0.96	10.29	1.14

Table 2.5: Comparison with various SNR, N=6

Method	SNR=40				SNR=30			
	SRE	std	RMSE	std	SRE	std	RMSE	std
NCLS	24.64	0.45	0.98	0.07	16.52	0.66	2.79	0.26
IRLS	7.93	0.11	7.39	0.16	6.04	0.30	9.77	0.39
SUnSAL	27.21	0.48	0.89	0.06	18.35	0.73	2.51	0.24
SUnSAL-TV	25.90	0.34	1.11	0.04	18.43	0.71	2.48	0.23
MUSIC-CSR	19.26	0.07	2.60	0.01	17.29	0.37	3.06	0.14
RM-DANSER	3.64	0.09	15.82	0.19	4.19	1.34	14.73	2.55
IRLSTV	**28.20**	0.46	**0.88**	0.05	**18.67**	0.48	**2.47**	0.19
IRLSTV2	26.66	0.47	0.98	0.06	18.62	0.54	2.49	0.14
Method	SNR=20				SNR=10			
	SRE	std	RMSE	std	SRE	std	RMSE	std
NCLS	8.59	0.44	7.24	0.47	2.83	0.25	13.10	0.56
IRLS	3.09	0.10	13.79	0.39	3.16	0.20	12.07	0.49
SUnSAL	9.56	0.49	6.51	0.42	3.16	0.28	11.78	0.51
SUnSAL-TV	9.69	0.50	6.37	0.42	3.12	0.29	11.77	0.54
MUSIC-CSR	10.08	0.36	6.04	0.33	3.96	0.32	11.51	0.62
RM-DANSER	5.05	0.76	11.12	1.43	0.76	0.57	15.60	1.55
IRLSTV	**10.27**	0.54	**5.98**	0.44	**4.10**	0.08	**8.39**	0.13
IRLSTV2	**10.31**	0.51	6.20	0.39	3.47	0.33	11.64	0.49

IRLSTV that $\lambda = 10^{-2}$, $\lambda_{TV} = 10^{-1}$ and $p = 0.4$. For the Alpine image we could not get

Table 2.6: Comparison with various SNR, N=9

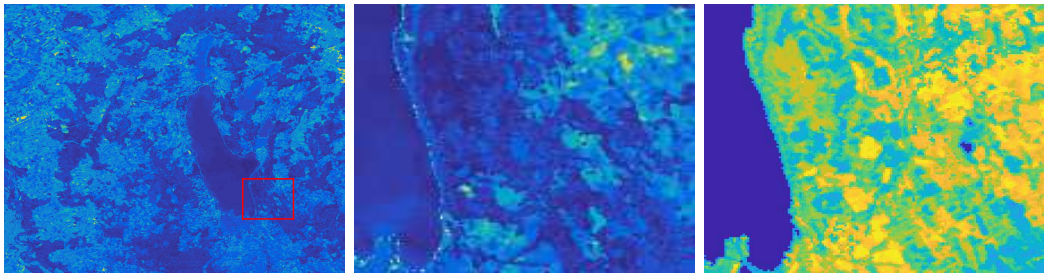
Method	SNR=40				SNR=30			
	SRE	std	RMSE	std	SRE	std	RMSE	std
NCLS	17.46	0.51	1.56	0.22	10.93	0.59	4.01	0.47
IRLS	2.81	0.07	8.76	0.05	1.72	0.12	9.58	0.19
SUnSAL	21.67	0.51	1.28	0.13	13.50	0.70	3.19	0.37
SUnSAL-TV	20.30	0.26	1.70	0.07	13.43	0.64	3.24	0.33
MUSIC-CSR	13.85	0.07	3.61	0.05	12.67	0.21	3.78	0.08
RM-DANSER	22.09	0.36	1.36	0.07	8.11	2.08	7.17	1.68
IRLSTV	**22.19**	0.52	**1.27**	0.10	**13.62**	0.41	**3.17**	0.20
IRLSTV2	**22.63**	0.74	**1.26**	0.13	13.19	0.55	3.70	0.28
Method	SNR=20				SNR=10			
	SRE	std	RMSE	std	SRE	std	RMSE	std
NCLS	4.91	0.15	7.88	0.35	0.43	0.26	12.30	0.78
IRLS	0.03	0.07	12.60	0.25	-0.31	0.13	10.76	0.37
SUnSAL	6.13	0.18	6.27	0.26	0.55	0.31	10.29	0.65
SUnSAL-TV	6.31	0.19	6.09	0.27	0.50	0.33	10.42	0.69
MUSIC-CSR	6.73	0.16	5.59	0.15	2.08	0.33	8.83	0.48
RM-DANSER	2.13	0.60	11.42	1.59	-2.41	0.64	18.10	2.48
IRLSTV	**6.80**	0.33	**5.48**	0.21	**4.05**	0.04	**5.91**	0.07
IRLSTV2	4.28	0.66	9.52	1.04	-1.00	2.71	15.30	6.93

Table 2.7: IRLSTV average running times in 100s of seconds

Method	N=3	N=6	N=9
NCLS	0.0003	0.0004	0.0015
IRLS	0.0175	0.0051	0.0090
SUnSAL	0.0002	0.0004	0.0011
SUnSAL-TV	0.0254	0.0310	0.0919
SUnSAL-CSR	0.0021	0.0025	0.0049
RMUSIC-DANSER	0.0092	0.0318	0.0443
IRLSTV	0.0246	0.0356	0.1614
IRLSTV2	30.0374	15.0031	50.0530

an unmixing result for IRLSTV2. Due to the larger image and increased computational complexity of IRLSTV2, MATLAB ran out of memory as it could not create large enough matrices.

Figures 2.11b and 2.11c show the sub-image of the Alpine scene used for the experiments. The band 40 and 100 are shown respectively in the images. Although the ground truth maps of the material abundances in this image are not available, these can be qualitatively assessed by looking at the image as there are not many features. From Fig. 2.11c we can clearly see the zones of the lake in the image, these are shown in dark blue color here. The lake covers most of the left side of the image ending at the bottom in a W shape. There can be seen an additional small lake or pond towards the middle right side of the image. The grassland and forest can be better determined looking at the Fig. 2.11b, with



(a) (b) (c)

Figure 2.11: German Alpine foothills image scene (1000 × 1000), band 40 in (a) and the selected sub-image (150 × 150), band 40 in (b), and band 100 in (c).

the light areas indicating the forest and darker areas the grass.

In Fig 2.12 it can be seen that SUnSAL-TV and RMUSIC-DANCER could not recover the water boundary. Furthermore, MUSIC-CSR and RMUSIC-DANCER were not able to clearly segment the grass from the forest. Where IRLSTV showed patches of grass, the fraction of forest was typically less; this was not apparent in NCLS. IRLSTV and MUSIC-CSR obtained the best results for fractional abundances of water, although MUSIC-CSR had a higher proportion of water estimates on the grasslands. The NCLS and SUnSAL-TV did not perform well in estimating the fractional abundances for the lake area, whereas IRLSTV showed good results. The forest and grass areas were also qualitatively better estimated by the IRLSTV. The NCLS and SUnSAL-TV also produced unrealistic fractional abundance values that are negative and/or larger than one.

Jasper Ridge image

Experiments were performed on the Jasper Ridge real hyperspectral image,[3] see Fig. 2.13. The image has 224 spectral bands in the $[380nm, 2500nm]$ range with a spectral resolution up to 9.46nm. Due to dense water and atmospheric effects the low SNR bands 1-3, 108-112, 154-166, and 220-224 were removed, hence 198 channels remained. The spectral channels of this image did not match with the available spectral libraries, hence we used an endmember extraction algorithm to identify material signatures from the data. The HySime algorithm estimated the signal subspace in the image to be 4. Hence we used the N-FINDR algorithm to find 4 endmember signatures from the hyperspectral image scene. Five-fold cross-validation was used to determine the optimal parameter values. These were $\lambda = \lambda_{TV} = 10^{-3}$, $p = 0.5$ for IRLSTV and $\lambda = 10^{-4}$, $\lambda_{TV} = 10^{-3}$ and $p = 0.5$ for IRLSTV2. The fractional abundance maps for tree, water, soil, and road are shown in Fig. 2.14.

The true fractional abundance maps for the Jasper Ridge hyperspectral image are unfortunately not available. However, as the image contains only 4 endmembers, they are

[3] Available online: http://facegis.nuarsa.info/?id=278.

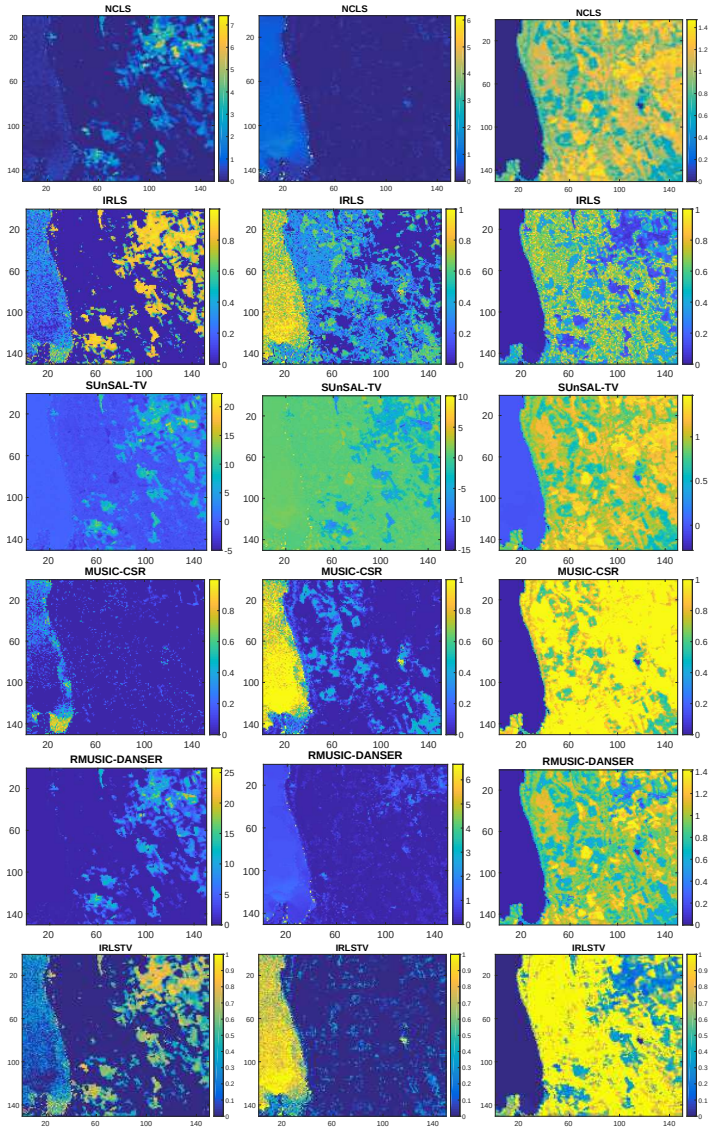
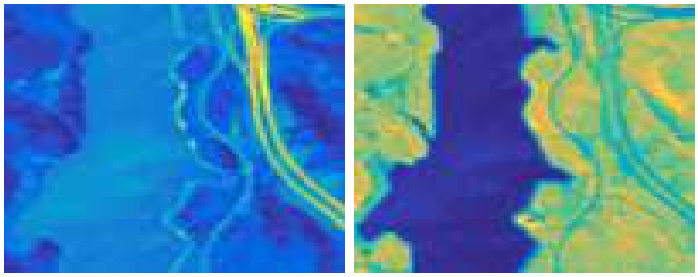


Figure 2.12: Fractional abundance maps of the German Alpine foothills data set estimated by NCLS, IRLS, SUnSAL-TV, MUSIC-CSR, RMUSIC-DANCER and IRLSTV. The abundance map of forest is shown in (a), water in (b), and grass in (c).

fairly easy to assess qualitatively. The image features can be seen in Fig. 2.13, with band 10 and 70 displayed for clearness. The road areas, starting from the top of the image and going towards the south-east side, can be seen in Fig. 2.13a in yellow, and the soil regions surrounding the roads and on the coast of the river in turquoise. Fig. 2.13b displays the large river flowing from the top of the image to the bottom in dark blue. The tree areas in the image can be seen in deep yellow in Fig. 2.13b.

Fig. 2.14, displays that RMUSIC-DANSER and SUnSAL-TV have fractional abundance values larger than 1 and SUnSAL-TV has negative fractional abundance values. This



(a) (b)

Figure 2.13: Jasper Ridge image (100 × 100), band 10 in (a), and band 70 in (b).

violates the sum-to-one and non-negativity constraints set on the fractional abundances and does not give realistic interpretable results. MUSIC-CSR, IRLSTV and IRLSTV2 were the only methods that can distinctly determine the water areas in the image and allocate high fractional abundances to these pixels. However, IRLSTV managed best in allocating zero fractional abundance values for areas outside of the water. Furthermore, IRLSTV seemed to offer more detail in the tree, soil and road maps in Fig. 2.14. While most methods struggled to identify the road areas in the image, IRLSTV was the only one that assigned high fractional abundances to these regions. IRLS and IRLSTV2 falsely determined the riverbank zone as road. From here we can see that although IRLSTV2 used an analytic sum-to-one constraint, it did not give as good of a performance as the IRLSTV with heuristic constraints. Also, due to increased computational complexity of IRLSTV2, the computation took significantly longer than IRLSTV.

The sparsity of the estimated $\hat{\boldsymbol{x}}$ matrix can be evaluated when looking at the density of the matrix (the proportion of non-zero abundance values in the matrix). There was a discernible difference in the sparsity of results even for the Jasper Ridge image, where the number of endmembers was very low. The density of $\hat{\boldsymbol{x}}$ for IRLSTV was 0.56, IRLSTV2 was 0.55, SUnSAL was 0.60 and SUnSAL-TV was 1. IRLS, MUSIC-CSR and RMUSIC-DANSER had the density 0.69, 0.64 and 0.61 respectively. IRLSTV did well in enforcing sparsity in the result, whereas SUnSAL-TV pushed some values towards zero, but did not equate them to zero exactly. Hence the fractional abundance values for the image were not easily interpretable for SUnSAL-TV.

2.7 Discussion

Sparse regression techniques have recently become popular for solving the statistical problem of spectral unmixing. Despite the success sparse unmixing has had in some applications, there are some limitations that need to be considered: using the classical ℓ_1 norm regularization on sparsity is often not sufficient; sparse unmixing methods do not take into account the spatial correlation in the images; the magnitude of the spectral libraries and the high mutual coherence limit the success of sparse unmixing algorithms.

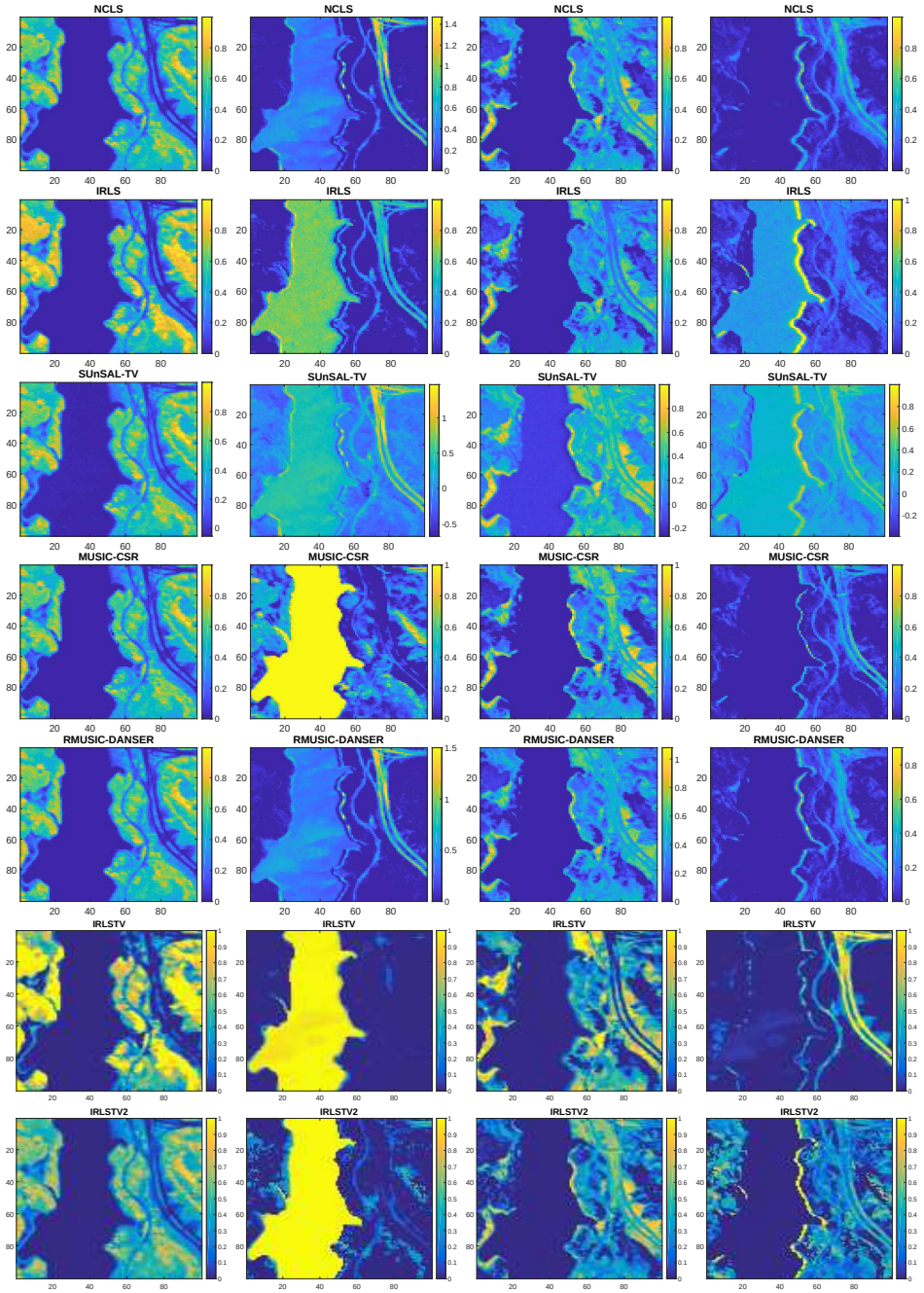


Figure 2.14: Fractional abundance maps for the Jasper Ridge data obtained by NCLS, IRLS, SUnSAL-TV, MUSIC-CSR, RMUSIC-DANSER, IRLSTV and IRLSTV2, for tree in (a), water in (b), soil in (c), and road in (d).

In this chapter we introduced a new algorithm called IRLSTV to overcome these limitations. We incorporated a library reduction step similar to MUSIC array processing algorithm, included a spatial total variation regularization term, and enforced sparsity on the estimates via ℓ_p norm. Our experimental results indicated that changing the ℓ_1 norm

on sparsity to ℓ_p norm with $0 < p < 1$ improved the unmixing performance. The benefits of including spatial context via the TV term was demonstrated. The library reduction step made the computation faster and improved the unmixing result significantly. The proposed algorithm was significantly better than other similar unmixing algorithms. The IRLSTV2 algorithm with analytical sum-to-one constraint showed similar or worse performance in the experiments compared to IRLSTV that has heuristic weights. This is possibly because the ASC and ANC constraint were not enforced jointly in IRLSTV2, which might create some instability in the algorithm. The heuristic weights kept the algorithm simple and fast that gave multiple benefits. Also, IRLSTV2 algorithm was significantly slower due to increased computational complexity.

3

Reparametrization for handling constraints

This chapter begins with an overview of the necessity of constraints for the hyperspectral unmixing problem and explains the reasons for reparametrization. It gives a review of the other methods that have used it beforehand. The chapter continues to develop a reparametrization method for the hyperspectral unmixing problem. The resulting algorithm is tested on both simulated and real hyperspectral data and compared with other state-of-the-art methods.

3.1 Overview

The unmixing problem comes with certain physical limitations that we need to adhere to in order to obtain valid estimates. As we aim at estimating the fractional abundances of materials in an image, we need to use the sum-to-one and non-negativity constraints. However, if one wishes to perform unconstrained optimization, then reparametrization with respect to the constraints allows that to be done.

Although reparametrization of coefficients is not uncommon in statistics, it has not been used in spectral unmixing until recently [79, 80]. [82] proposed a general form of reparametrization that fits under the above model, with the logarithmic function as one option for reparametrization. Logit functions were used for reparametrization in [113], whereas [84] used a Dirichlet distribution as a prior to enforce the constraints. In [85] the Dirichlet distribution was used in combination with reparametrization.

In this chapter, we propose to reparametrize the model's coefficients to account for sum-to-one and non-negativity constraints for a sparse unmixing technique with spatial total variation regularization that induces sparsity in the number of active pure spectral components via a hyper-Laplacian prior, the ℓ_p norm. We use the non-convex ℓ_p norm because it produces sparser results than the ℓ_1 norm [34]. Since this approach can shrink the fractional abundances of inactive endmembers to values very close to zero, greater

selectivity is offered as compared to the use of a more traditional ℓ_1 norm. In turn, this enables the method to be used on large, semi-supervised spectral libraries or in unsupervised scenarios where the number of pure signatures present in a typical pixel is much smaller than the number present in the image scene. Through the $|\nabla \boldsymbol{x}|$ term, we introduce a total variation regularizer into a sparse spectral unmixing framework. However, unlike [37] that uses the ADMM method and the ℓ_1 norm, we achieve greater sparsity by setting up a constrained $\ell_p - \ell_2$ optimization problem. We propose to estimate the fractional abundances $\boldsymbol{x}$ as a solution to the non-convex optimization problem

$$\arg\min_{\boldsymbol{x}} \|\boldsymbol{y} - \boldsymbol{M}\boldsymbol{x}\|_2^2 + \lambda \|\boldsymbol{x}\|_p^p + \lambda_{TV} \|\nabla \boldsymbol{x}\|_p^p \ , \tag{3.1}$$

where $\boldsymbol{y}, \boldsymbol{x}, \boldsymbol{M}$ with the respective dimensions $LP \times 1$, $NP \times 1$, and $LP \times NP$ describe the unmixing problem over all pixels jointly. They are defined as

$$\begin{aligned} \boldsymbol{y} &= \left[\boldsymbol{y}_1^\top, \boldsymbol{y}_2^\top, \ldots, \boldsymbol{y}_P^\top\right]^\top , \\ \boldsymbol{x} &= \left[\boldsymbol{x}_1^\top, \boldsymbol{x}_2^\top, \ldots, \boldsymbol{x}_P^\top\right]^\top , \\ \boldsymbol{M} &= I_P \otimes \boldsymbol{M}_0 \ . \end{aligned} \tag{3.2}$$

Here $\otimes$ denotes the Kronecker product, $\boldsymbol{y}_n \in \mathbb{R}^{L\times 1}$ is the hyperspectral reflectance of the nth pixel, $\boldsymbol{x}_n$ of dimension $N \times 1$ is the fractional abundance vector of the N endmembers in pixel n, and $\boldsymbol{M}_0 \in \mathbb{R}^{L\times N}$ is the spectral library that forms a block-diagonal matrix $\boldsymbol{M}$ with P blocks of $\boldsymbol{M}_0$ on the main diagonal and the rest of the matrix $\boldsymbol{M}$ consists of zeros.

As the fractional abundances in $\boldsymbol{x}$ refer to proportions, the optimization problem (3.1) should be subject to the non-negativity and sum-to-one constraints defined by $\boldsymbol{x} \geq 0, \boldsymbol{a}\boldsymbol{x}_n = 1$ for $n = 1, \ldots, P$ pixels, where $\boldsymbol{a} = [1, \ldots, 1]$ has dimension N. Unlike the renormalization heuristics used in [35, 103] and [78], we derive and hence incorporate a well-principled reparametrization for optimization problem (3.1), which was found to offer further performance enhancements in our experiments. In order to take the constraints into account, whilst performing unconstrained optimization, we reparametrize $\boldsymbol{x}$ in such a way that for each pixel we estimate the $N-1$ unconstrained parameters defined as [81]

$$t_i = \log\left(\frac{x_i}{1 - \sum_{j=2}^{N} x_j}\right) \in \mathbb{R} \ , \tag{3.3}$$

where $i = 2, \ldots, N$.

We propose a three-step algorithm to solve minimization problem (3.1). The first step performs library reduction using hyperspectral subspace identification by minimum error [100], and a binary test similar to MUSIC to identify the endmembers. In an environment with no noise the MUSIC step is able to identify the active endmembers correctly [76].

Additionally, we propose to use a robust version of the MUSIC library reduction as a comparison, as there can be some noise in the data. The next two steps solve optimization problem (3.1) using the reparametrization in (3.3) via a trust region algorithm based on first and second order analytical derivatives, that is stable and has strong convergence properties [52, 114–116].

The proposed algorithm (here called ROCSSUM) performs unconstrained $\ell_p - \ell_2$ optimization with spatial total variation regularization and MUSIC library reduction using the reparametrization of constraints. The method with the robust library reduction is called ROCSSURM accordingly. Experiments on both synthetic and real data show that the proposed method obtains better hyperspectral unmixing performance in noisy settings compared to other recent state-of-the-art statistical regularization methods such as IRLS [35], SUnSAL [102], SUnSAL-TV [37],SUnSAL-CSR [101], MUSIC-CSR [76], RMUSIC-DANSER [49]. The differences between our method and the competitors are summarized in Table 3.1.

Table 3.1: Method comparison

Method	Library pruning	Sparsity $\|.\|_p$	Spatial $\|\nabla.\|_p$	Sum-to-one	Non-negativity
NCLS	No	No	No	No	Yes
IRLS [35]	No	Yes	No	Normalize	Yes
SUnSAL [28]	No	p=1	No	No	Yes
SUnSAL-TV [37]	No	p=1	p=1	No	Yes
SUnSAL-CSR [101]	No	p=1	No	No	Yes
MUSIC-CSR [76]	Yes	p=1	No	No	Yes
RMUSIC-CSR	Robust	p=1	No	No	Yes
RMUSIC-DANSER [49]	Robust	Yes	No	No	Yes
ROCSSUM	Yes	Yes	Yes	Yes	Yes
ROCSSURM	Robust	Yes	Yes	Yes	Yes

3.2 Regularization

3.2.1 Reweighted ℓ_p Norm

The constrained, sparse $\ell_p - \ell_2$ hyperspectral unmixing task, with $0 < p < 1$, is defined as the optimization problem

$$\min_{\boldsymbol{x}} \|\boldsymbol{y} - \boldsymbol{M}\boldsymbol{x}\|_2^2 + \lambda \|\boldsymbol{x}\|_p^p, \tag{3.4}$$

where λ is the Lagrange multiplier that regulates the sparsity of the solution and $\boldsymbol{y}, \boldsymbol{x}, \boldsymbol{M}$ are defined as in (3.2). It is possible to rewrite the Lasso objective as a weighted ridge

regression (e.g., [104]). Then (3.4) becomes

$$\min_{\boldsymbol{x}} \|\boldsymbol{y} - \boldsymbol{M}\boldsymbol{x}\|_2^2 + \lambda \|\boldsymbol{D}\boldsymbol{x}\|_2^2 \ , \tag{3.5}$$
$$\text{with} \quad \boldsymbol{D} = \text{diag}\left(\boldsymbol{d}^{\frac{1}{2}}\right), \ \boldsymbol{d} = \left(\left(\boldsymbol{x}^{(\alpha-1)}\right)^2 + \boldsymbol{\epsilon}^2\right)^{\frac{p}{2}-1} \ ,$$

where the weights $\boldsymbol{d}$ for iteration α are calculated using the $\boldsymbol{x}^{(\alpha-1)}$ results from the previous iteration $\alpha - 1$. Hence the weighted ℓ_2 norm in (3.5) is a first-order approximation to the ℓ_p norm in optimization problem (3.4). Here, $\boldsymbol{D}$ is a positive diagonal matrix, $\boldsymbol{\epsilon}$ measures the error in the solution for the pixels in the whole image, and goes to zero as the IRLS algorithm converges. $\boldsymbol{\epsilon}_n$ for the nth pixel is a vector of size $N \times 1$, $\boldsymbol{\epsilon}$ covers the whole image s.t. $\left[\boldsymbol{\epsilon}_1^\top, \boldsymbol{\epsilon}_2^\top, \ldots, \boldsymbol{\epsilon}_P^\top\right]^\top$, size $NP \times 1$. When the algorithm converges then $\boldsymbol{\epsilon} \to 0$ and $\boldsymbol{x}^{(\alpha-1)} \cong \boldsymbol{x}^{(\alpha)}$. It follows that $\|\boldsymbol{D}\boldsymbol{x}\|_2^2 = \|\boldsymbol{x}\|_p^p$.

The algorithm proceeds by initializing the weights with ones and then iterating between: (i) solving the quadratic problem in (3.5) and (ii) updating the weight matrix $\boldsymbol{D}$. This is a fixed-point iteration for solving the optimization problem (3.4). The iterative update leads to an algorithm that guarantees convergence [45]. The algorithm enjoys two very attractive properties: simplicity and flexibility [105]. The next section shows that it is easy to accommodate additional p-norm penalty terms, such as a total-variation regularizer in the objective function.

3.2.2 Total Variation for Sparse Unmixing

We propose to solve the constrained, sparse, total variation (TV) regularization problem

$$\arg\min_{\boldsymbol{x}} \|\boldsymbol{y} - \boldsymbol{M}\boldsymbol{x}\|_2^2 + \lambda \|\boldsymbol{x}\|_p^p + \lambda_{TV} \|\nabla \boldsymbol{x}\|_p^p \ , \tag{3.6}$$

where $\boldsymbol{y}, \boldsymbol{x}, \boldsymbol{M}$ are defined as in (3.2). The TV term is equivalent to placing a hyper-Laplacian prior on the horizontal and vertical increments of the abundances. The regularization parameter λ_{TV} balances the influence of this prior against the sparsity and likelihood terms; larger values will result in smoother solutions. The differencing operator ∇ computes the spatial increments of the fractional abundances, i.e. differences of abundances over neighboring pixels. It can be decomposed into horizontal ∇_h and vertical ∇_v differences so that $\nabla \equiv [\nabla_h^\top, \nabla_v^\top]^\top$. The differences between horizontal neighbors are $\nabla_h \mathbf{x} = [\mathbf{a_1}, \mathbf{a_2}, ..., \mathbf{a_{T_h}}]^\top$, where $\boldsymbol{a}_n = \mathbf{x_n} - \mathbf{x_{n_h}}$, with n and n_h indicating a pixel and its horizontal neighbor, and T_h is the number of horizontal differences taken for the image. ∇_v performs similarly for vertical neighbors. The difference operator ∇ has one 1 and -1 in each row for the respective vertical or horizontal endmembers. The rest of the entries are zeros. The distance between the 1 and -1 in the rows of ∇_h is the number of endmembers N. The difference is taken in between $x_n[i]$ and $x_{n+1}[i]$ for endmember i and pixels n and $n+1$. For ∇_v the distance is cN, where c indicates the number of pixels in each row of

the image. In order to solve the optimization problem (3.6), the p-norm is rewritten in terms of a weighted 2-norm, to obtain

$$\arg\min_{\boldsymbol{x}} \quad \|\boldsymbol{y}-\boldsymbol{M}\boldsymbol{x}\|_2^2+\lambda\|\boldsymbol{D}\boldsymbol{x}\|_2^2+\lambda_{TV}\left\|\boldsymbol{\Phi}\nabla\boldsymbol{x}\right\|_2^2, \tag{3.7}$$

$$\text{with} \quad \boldsymbol{D}=\operatorname{diag}(\boldsymbol{d}^{\frac{1}{2}}), \quad \boldsymbol{d}=\left(\left(\boldsymbol{x}^{(\alpha-1)}\right)^2+\boldsymbol{\epsilon}^2\right)^{\frac{p}{2}-1},$$
$$\boldsymbol{\Phi}=\operatorname{diag}(\boldsymbol{\phi}^{\frac{1}{2}}), \quad \boldsymbol{\phi}=\left(\left(\nabla\boldsymbol{x}^{(\alpha-1)}\right)^2+\boldsymbol{\eta}^2\right)^{\frac{p}{2}-1}.$$

Here, $\boldsymbol{\Phi}$ is a positive diagonal matrix containing the weights, the constant $\boldsymbol{\eta}$ is a column vector containing the small integer value of 10^{-6} which is included in order to avoid division by zeros, and the weights $\boldsymbol{d}$ and $\boldsymbol{\phi}$ are calculated using the $\boldsymbol{x}^{(\alpha-1)}$ results from the previous iteration $\alpha-1$. Note that the weighted ℓ_2 norm is equivalent to the ℓ_p norm for small enough $\boldsymbol{\eta}$ and when $\boldsymbol{x}^{(\alpha-1)}=\boldsymbol{x}^{(\alpha)}$, that is $\left\|\boldsymbol{\Phi}\nabla\boldsymbol{x}\right\|_2^2=\|\nabla\boldsymbol{x}\|_p^p$.

3.3 Reparametrization of Constraints

The regularization problem in Eq. (3.6) requires the constraints $\boldsymbol{x}\geq 0$ and $\boldsymbol{a}\boldsymbol{x}_n=1$ for $n=1,\ldots,P$. An effective and theoretically founded way to address this is to reparametrize the model's coefficients such that we can use an unconstrained optimizer like the trust region algorithm. We propose to use the unconstrained parameters

$$t_i \;=\; \log\left(\frac{x_i}{1-\sum_{j=2}^N x_j}\right)\in\mathbb{R}\ (i=2,\ldots,N). \tag{3.8}$$

The original (constrained) parameters can be derived as [81]

$$\begin{aligned} e^{t_i} &= \frac{x_i}{1-\sum_{j=2}^N x_i} \\ e^{t_i} &= e^{t_i}\sum_{j=2}^N x_j + x_i \\ e^{t_i} &= x_i\left(\sum_{j=2}^N e^{t_j}+1\right) \\ x_i &= \frac{e^{t_i}}{1+\sum_{j=2}^N e^{t_j}}, \end{aligned} \tag{3.9}$$

where $i=2,\ldots,N$, and $x_1=1-\sum_{i=2}^N x_i=1-\sum_{i=2}^N\frac{e^{t_i}}{1+\sum_{j=2}^N e^{t_j}}$.

The next sections derive the quantities needed for the trust region optimization.

3.3.1 Gradient

The linear mixing model in Equation (3.6) can be expressed as

$$\begin{aligned} f(\boldsymbol{x}) &= (\boldsymbol{y}-\boldsymbol{M}\boldsymbol{x})^\top(\boldsymbol{y}-\boldsymbol{M}\boldsymbol{x}) + \lambda(\boldsymbol{D}\boldsymbol{x})^\top(\boldsymbol{D}\boldsymbol{x}) + \lambda_{TV}(\boldsymbol{\Phi}\nabla\boldsymbol{x})^\top(\boldsymbol{\Phi}\nabla\boldsymbol{x}) \\ &= \boldsymbol{y}^\top\boldsymbol{y} + \boldsymbol{x}^\top\left(\boldsymbol{M}^\top\boldsymbol{M}\right)\boldsymbol{x} - 2\boldsymbol{y}^\top\boldsymbol{M}\boldsymbol{x} + \lambda\boldsymbol{x}^\top\boldsymbol{D}^\top\boldsymbol{D}\boldsymbol{x} \\ &\quad + \lambda_{TV}\boldsymbol{x}^\top\nabla^\top\boldsymbol{\Phi}^\top\boldsymbol{\Phi}\nabla\boldsymbol{x}\,, \end{aligned} \tag{3.10}$$

which has to be maximized with respect to the $N-1$ unconstrained parameters $\mathbf{t} = [t_2, \cdots, t_N]^\top$. The gradient is obtained as follows.

$$\frac{\partial f(\boldsymbol{x})}{\partial \mathbf{t}} = \frac{\partial f(\boldsymbol{x}(\mathbf{t}))}{\partial \boldsymbol{x}}\frac{\partial \boldsymbol{x}}{\partial \mathbf{t}} = \left(\frac{\partial \boldsymbol{x}}{\partial \mathbf{t}}\right)^\top \frac{\partial f(\boldsymbol{x})}{\partial \boldsymbol{x}} \tag{3.11}$$

First, we find the partial derivative with respect to $\boldsymbol{x}$. Here the weight matrices $\boldsymbol{D}$ and $\boldsymbol{\Phi}$ are kept fixed since they are computed using the previous iteration value $\boldsymbol{x}^{(\alpha-1)}$.

$$\begin{aligned} \frac{\partial f(\boldsymbol{x})}{\partial \boldsymbol{x}} &= \left(\boldsymbol{M}^\top\boldsymbol{M} + \left(\boldsymbol{M}^\top\boldsymbol{M}\right)^\top\right)\boldsymbol{x} - 2\left(\boldsymbol{y}^\top\boldsymbol{M}\right)^\top + \lambda\left(\boldsymbol{D}^\top\boldsymbol{D} + \left(\boldsymbol{D}^\top\boldsymbol{D}\right)^\top\right)\boldsymbol{x} \\ &\quad + \lambda_{TV}\left(\nabla^\top\boldsymbol{\Phi}^\top\boldsymbol{\Phi}\nabla + \left(\nabla^\top\boldsymbol{\Phi}^\top\boldsymbol{\Phi}\nabla\right)^\top\right)\boldsymbol{x} \\ &= 2\boldsymbol{M}^\top\boldsymbol{M}\boldsymbol{x} - 2\boldsymbol{M}^\top\boldsymbol{y} + 2\lambda\boldsymbol{D}^\top\boldsymbol{D}\boldsymbol{x} + 2\lambda_{TV}\nabla^\top\boldsymbol{\Phi}^\top\boldsymbol{\Phi}\nabla\boldsymbol{x}\,. \end{aligned} \tag{3.12}$$

The partial derivatives with respect to t_i for $i = 2, \ldots, N$, $k = 2, \ldots, N$, where i and k indicate the i'th and k'th endmember and $k \neq i$, are obtained using

$$\frac{\partial x_k}{\partial t_i} = \frac{\partial}{\partial t_i}\left(\frac{e^{t_k}}{1+\sum_{j=2}^N e^{t_j}}\right) = \frac{e^{t_k}\cdot\left(-e^{t_i}\right)}{\left(1+\sum_{j=2}^N e^{t_j}\right)^2} = -\frac{e^{t_k+t_i}}{\left(1+\sum_{j=2}^N e^{t_j}\right)^2}. \tag{3.13}$$

When $k = i$ the partial derivatives with respect to t_i become

$$\begin{aligned} \frac{\partial x_k}{\partial t_i} &= \frac{\partial}{\partial t_i}\left(\frac{e^{t_k}}{1+\sum_{j=2}^N e^{t_j}}\right) = \frac{e^{t_k}}{1+\sum_{j=2}^N e^{t_j}} + \frac{e^{t_k}\cdot\left(-e^{t_i}\right)}{\left(1+\sum_{j=2}^N e^{t_j}\right)^2} \\ &= \frac{e^{t_k}}{1+\sum_{j=2}^N e^{t_j}} - \frac{e^{t_k+t_i}}{\left(1+\sum_{j=2}^N e^{t_j}\right)^2}. \end{aligned} \tag{3.14}$$

For $k = 1$ the partial derivatives are

$$\begin{aligned}
\frac{\partial x_1}{\partial t_i} &= \frac{\partial}{\partial t_i}\left(1 - \frac{e^{t_2}}{1+\sum_{j=2}^{N} e^{t_j}} - \frac{e^{t_3}}{1+\sum_{j=2}^{N} e^{t_j}} - \frac{e^{t_4}}{1+\sum_{j=2}^{N} e^{t_j}} - \cdots - \frac{e^{t_N}}{1+\sum_{j=2}^{N} e^{t_j}}\right) \\
&= \frac{e^{t_2+t_i}}{\left(1+\sum_{j=2}^{N} e^{t_j}\right)^2} + \frac{e^{t_3+t_i}}{\left(1+\sum_{j=2}^{N} e^{t_j}\right)^2} + \frac{e^{t_4+t_i}}{\left(1+\sum_{j=2}^{N} e^{t_j}\right)^2} + \cdots \\
&\quad + \frac{e^{t_N+t_i}}{\left(1+\sum_{j=2}^{N} e^{t_j}\right)^2} - \frac{e^{t_i}}{1+\sum_{j=2}^{N} e^{t_j}} \\
&= -\frac{e^{t_i}}{1+\sum_{j=2}^{N} e^{t_j}} + \frac{e^{t_i}\cdot\sum_{j=2}^{N} e^{t_j}}{\left(1+\sum_{j=2}^{N} e^{t_j}\right)^2} = \frac{-e^{t_i} - e^{t_i}\cdot\sum_{j=2}^{N} e^{t_j} + e^{t_i}\cdot\sum_{j=2}^{N} e^{t_j}}{\left(1+\sum_{j=2}^{N} e^{t_j}\right)^2} \\
&= -\frac{e^{t_i}}{\left(1+\sum_{j=2}^{N} e^{t_j}\right)^2}. \qquad (3.15)
\end{aligned}$$

Thus the gradient is

$$\frac{\partial f(\boldsymbol{x})}{\partial \mathbf{t}} = \left(\frac{\partial x_k}{\partial t_i}\right)^{\top}\left(2\boldsymbol{M}^{\top}\boldsymbol{M}\boldsymbol{x} - 2\boldsymbol{M}^{\top}\boldsymbol{y} + 2\lambda\boldsymbol{D}^{\top}\boldsymbol{D}\boldsymbol{x} + 2\lambda_{TV}\nabla^{\top}\boldsymbol{\Phi}^{\top}\boldsymbol{\Phi}\nabla\boldsymbol{x}\right), \qquad (3.16)$$

where

$$\frac{\partial x_k}{\partial t_i} = \begin{cases} -\frac{e^{t_i}}{\left(1+\sum_{j=2}^{N} e^{t_j}\right)^2} & \text{, for } k = 1 \\ -\frac{e^{t_i+t_k}}{\left(1+\sum_{j=2}^{N} e^{t_j}\right)^2} & \text{, for } k = 2,\ldots,N \;\&\; k \neq i \\ -\frac{e^{t_i+t_k}}{\left(1+\sum_{j=2}^{N} e^{t_j}\right)^2} + \frac{e^{t_i}}{1+\sum_{j=2}^{N} e^{t_j}} & \text{, for } k = 2,\ldots,N \;\&\; k = i\ . \end{cases} \qquad (3.17)$$

This can be also expressed as:

$$\begin{aligned}
\frac{\partial f(\boldsymbol{x})}{\partial \mathbf{t}} &= \begin{bmatrix} -\frac{e^{t_i}}{\left(1+\sum_{j=2}^{N} e^{t_j}\right)^2} \\ -\frac{e^{t_i+t_k}}{\left(1+\sum_{j=2}^{N} e^{t_j}\right)^2} + \operatorname{diag}\left(\frac{e^{t_i}}{1+\sum_{j=2}^{N} e^{t_j}}\right) \end{bmatrix}^{\top} \left(2\boldsymbol{M}^{\top}\boldsymbol{M}\boldsymbol{x} - 2\boldsymbol{M}^{\top}\boldsymbol{y}\right. \\
&\quad \left. + 2\lambda\boldsymbol{D}^{\top}\boldsymbol{D}\boldsymbol{x} + 2\lambda_{TV}\nabla^{\top}\boldsymbol{\Phi}^{\top}\boldsymbol{\Phi}\nabla\boldsymbol{x}\right) \qquad (3.18)
\end{aligned}$$

The partial derivatives $\partial\boldsymbol{x}/\partial\mathbf{t}$ are of dimension $N\times(N-1)$, $\partial f(\boldsymbol{x})/\partial\mathbf{x}$ of dimension $N\times 1$, and $\partial f(\boldsymbol{x})/\partial\mathbf{t}$ of dimension $(N-1)\times 1$; they require the estimation of $N-1$ gradients.

3.3.2 Preliminary steps for the Hessian

The Hessian for $f(\boldsymbol{x})$ is obtained using the gradient in Equations (3.16)-(3.17) and the chain rule to find the second partial derivatives of $f(\boldsymbol{x})$ with respect to t_i, where $i =$

$2, \ldots, N$. Then the Chain rule for becomes:

$$\frac{\partial^2 f(\boldsymbol{x})}{\partial \mathbf{t}_i^2} = \left(\frac{\partial \boldsymbol{x}}{\partial \mathbf{t}_i}\right)^\top \frac{\partial^2 f(\boldsymbol{x})}{\partial \boldsymbol{x}^2} \frac{\partial \boldsymbol{x}}{\partial \mathbf{t}_i} + \left(\frac{\partial^2 \boldsymbol{x}}{\partial \mathbf{t}_i^2}\right)^\top \frac{\partial f(\boldsymbol{x})}{\partial \boldsymbol{x}} . \tag{3.19}$$

$$\frac{\partial^2 f(\boldsymbol{x})}{\partial \mathbf{t}_s \partial \mathbf{t}_i} = \left(\frac{\partial \boldsymbol{x}}{\partial \mathbf{t}_s}\right)^\top \frac{\partial^2 f(\boldsymbol{x})}{\partial \boldsymbol{x}^2} \frac{\partial \boldsymbol{x}}{\partial \mathbf{t}_i} + \left(\frac{\partial^2 \boldsymbol{x}}{\partial \mathbf{t}_s \partial \mathbf{t}_i}\right)^\top \frac{\partial f(\boldsymbol{x})}{\partial \boldsymbol{x}} . \tag{3.20}$$

Using Eq. (3.12) we get the second partial derivatives with respect to $\boldsymbol{x}$ with fixed weights $\boldsymbol{D}$ and $\boldsymbol{\Phi}$.

$$\frac{\partial^2 f(\boldsymbol{x})}{\partial \boldsymbol{x}^2} = 2\boldsymbol{M}^\top \boldsymbol{M} + \lambda \boldsymbol{D}^\top \boldsymbol{D} + \lambda_{TV} \nabla^\top \boldsymbol{\Phi}^\top \boldsymbol{\Phi} \nabla . \tag{3.21}$$

Next, we find the second order derivatives with respect to t_i for $i = 2, \ldots, N$. For $k = 1$,

$$\frac{\partial^2 x_1}{\partial t_i^2} = \frac{\partial}{\partial t_i}\left(-\frac{e^{t_i}}{\left(1+\sum_{j=2}^N e^{t_j}\right)^2}\right) = -\frac{e^{t_i}}{\left(1+\sum_{j=2}^N e^{t_j}\right)^2} - \frac{e^{t_i}\cdot\left(-2e^{t_i}\right)}{\left(1+\sum_{j=2}^N e^{t_j}\right)^3}$$

$$= -\frac{e^{t_i}}{\left(1+\sum_{j=2}^N e^{t_j}\right)^2} + \frac{2e^{2t_i}}{\left(1+\sum_{j=2}^N e^{t_j}\right)^3} . \tag{3.22}$$

For $k = 2, \ldots, N$ and $k \neq i$, we obtain

$$\begin{aligned}\frac{\partial^2 x_k}{\partial t_i^2} &= \frac{\partial}{\partial t_i}\left(-\frac{e^{t_k+t_i}}{\left(1+\sum_{j=2}^N e^{t_j}\right)^2}\right) = -\frac{e^{t_k+t_i}}{\left(1+\sum_{j=2}^N e^{t_j}\right)^2} - \frac{e^{t_k+t_i}\cdot\left(-2e^{t_i}\right)}{\left(1+\sum_{j=2}^N e^{t_j}\right)^3} \\ &= -\frac{e^{t_k+t_i}}{\left(1+\sum_{j=2}^N e^{t_j}\right)^2} + \frac{2e^{t_k+2t_i}}{\left(1+\sum_{j=2}^N e^{t_j}\right)^3} .\end{aligned} \tag{3.23}$$

When $k = i$,

$$\begin{aligned}\frac{\partial^2 x_k}{\partial t_i^2} &= \frac{\partial}{\partial t_i}\left(-\frac{e^{t_k+t_i}}{\left(1+\sum_{j=2}^N e^{t_j}\right)^2} + \frac{e^{t_i}}{1+\sum_{j=2}^N e^{t_j}}\right) \\ &= -\frac{e^{t_k+t_i}\cdot(1+1)}{\left(1+\sum_{j=2}^N e^{t_j}\right)^2} - \frac{e^{t_k+t_i}\cdot\left(-2e^{t_i}\right)}{\left(1+\sum_{j=2}^N e^{t_j}\right)^3} + \frac{e^{t_i}}{1+\sum_{j=2}^N e^{t_j}} + \frac{e^{t_i}\cdot\left(-e^{t_i}\right)}{\left(1+\sum_{j=2}^N e^{t_j}\right)^2} \\ &= \frac{2e^{t_k+2t_i}}{\left(1+\sum_{j=2}^N e^{t_j}\right)^3} - \frac{2e^{t_k+t_i}+2e^{t_i}}{\left(1+\sum_{j=2}^N e^{t_j}\right)^2} + \frac{e^{t_i}}{1+\sum_{j=2}^N e^{t_j}} .\end{aligned} \tag{3.24}$$

Hence the second order partial derivatives are

$$\frac{\partial^2 x_k}{\partial t_i^2} = \begin{cases} \frac{2e^{2t_i}}{\left(1+\sum_{j=2}^N e^{t_j}\right)^3} - \frac{e^{t_i}}{\left(1+\sum_{j=2}^N e^{t_j}\right)^2} & \text{, for } k=1 \\ \frac{2e^{2t_i+t_k}}{\left(1+\sum_{j=2}^N e^{t_j}\right)^3} - \frac{e^{t_i+t_k}}{\left(1+\sum_{j=2}^N e^{t_j}\right)^2} & \text{, for } k=2,\ldots,N \;\&\; k\neq i \\ \frac{2e^{2t_i+t_k}}{\left(1+\sum_{j=2}^N e^{t_j}\right)^3} - \frac{e^{t_i+t_k}+2e^{t_i}}{\left(1+\sum_{j=2}^N e^{t_j}\right)^2} + \frac{e^{t_i}}{1+\sum_{j=2}^N e^{t_j}} & \text{, for } k=2,\ldots,N \;\&\; k=i\,. \end{cases} \tag{3.25}$$

For $s \neq i$, $s = 2, \ldots, N$, the mixed second order derivatives for $k = 1$ are

$$\frac{\partial^2 x_1}{\partial t_i \partial t_s} = \frac{\partial}{\partial t_s}\left(-\frac{e^{t_i}}{\left(1+\sum_{j=2}^N e^{t_j}\right)^2}\right) = -\frac{e^{t_i}\cdot\left(-2e^{t_s}\right)}{\left(1+\sum_{j=2}^N e^{t_j}\right)^3} = \frac{2e^{t_i+t_s}}{\left(1+\sum_{j=2}^N e^{t_j}\right)^3}\,. \tag{3.26}$$

For $k = 2, \ldots, N$, such that $k \neq i$ & $k \neq s$,

$$\frac{\partial^2 x_k}{\partial t_i \partial t_s} = \frac{\partial}{\partial t_s}\left(-\frac{e^{t_k+t_i}}{\left(1+\sum_{j=2}^N e^{t_j}\right)^2}\right) = -\frac{e^{t_k+t_i}\cdot\left(-2e^{t_s}\right)}{\left(1+\sum_{j=2}^N e^{t_j}\right)^3} = \frac{2e^{t_k+t_i+t_s}}{\left(1+\sum_{j=2}^N e^{t_j}\right)^3}\,. \tag{3.27}$$

For $k = s$, $k \neq i$ and $k = 2, \ldots, N$,

$$\frac{\partial^2 x_k}{\partial t_i \partial t_s} = \frac{\partial}{\partial t_s}\left(-\frac{e^{t_k+t_i}}{\left(1+\sum_{j=2}^N e^{t_j}\right)^2}\right) = -\frac{e^{t_k+t_i}}{\left(1+\sum_{j=2}^N e^{t_j}\right)^2} + \frac{2e^{t_k+t_i+t_s}}{\left(1+\sum_{j=2}^N e^{t_j}\right)^3}\,. \tag{3.28}$$

For $k = i$, $k \neq s$, and $k = 2, \ldots, N$,

$$\begin{aligned} \frac{\partial^2 x_k}{\partial t_i \partial t_s} &= \frac{\partial}{\partial t_s}\left(-\frac{e^{t_k+t_i}}{\left(1+\sum_{j=2}^N e^{t_j}\right)^2} + \frac{e^{t_i}}{1+\sum_{j=2}^N e^{t_j}}\right) = \\ &= \frac{2e^{t_k+t_i+t_s}}{\left(1+\sum_{j=2}^N e^{t_j}\right)^3} - \frac{e^{t_i+t_s}}{\left(1+\sum_{j=2}^N e^{t_j}\right)^2}\,. \end{aligned} \tag{3.29}$$

In summary, we can write the equations for the mixed second derivative, where $i \neq s$, as

$$\frac{\partial^2 x_k}{\partial t_i \partial t_s} = \begin{cases} \frac{2e^{t_i+t_s}}{\left(1+\sum_{j=2}^N e^{t_j}\right)^3} & \text{, for } k=1 \\ \frac{2e^{t_k+t_i+t_s}}{\left(1+\sum_{j=2}^N e^{t_j}\right)^3} & \text{, for } k=2,\ldots,N \;\&\; k\neq i, k\neq s \\ \frac{2e^{t_k+t_i+t_s}}{\left(1+\sum_{j=2}^N e^{t_j}\right)^3} - \frac{e^{t_i+t_s}}{\left(1+\sum_{j=2}^N e^{t_j}\right)^2} & \text{, for } k=2,\ldots,N \;\&\; k=i\,. \end{cases} \tag{3.30}$$

3.3.3 Hessian

The derivatives that make up the Hessian matrix are described below.

$$\begin{aligned}\frac{\partial^2 f(\boldsymbol{x})}{\partial \mathbf{t}_i^2} &= \left(\frac{\partial \boldsymbol{x}}{\partial \mathbf{t}_i}\right)^\top \left(2\boldsymbol{M}^\top \boldsymbol{M} + \lambda \boldsymbol{D}^\top \boldsymbol{D} + \lambda_{TV} \nabla^\top \boldsymbol{\Phi}^\top \boldsymbol{\Phi} \nabla\right) \frac{\partial \boldsymbol{x}}{\partial \mathbf{t}_i} \\ &\quad + \left(\frac{\partial^2 \boldsymbol{x}}{\partial \mathbf{t}_i^2}\right)^\top \Big(2\boldsymbol{M}^\top \boldsymbol{M}\boldsymbol{x} - 2\boldsymbol{M}^\top \boldsymbol{y} + 2\lambda \boldsymbol{D}^\top \boldsymbol{D}\boldsymbol{x} \\ &\quad + \lambda_{TV} \nabla^\top \boldsymbol{\Phi}^\top \boldsymbol{\Phi} \nabla \boldsymbol{x}\Big), \end{aligned} \tag{3.31}$$

$$\begin{aligned}\frac{\partial^2 f(\boldsymbol{x})}{\partial \mathbf{t}_i \partial \mathbf{t}_s} &= \left(\frac{\partial \boldsymbol{x}}{\partial \mathbf{t}_i}\right)^\top \left(2\boldsymbol{M}^\top \boldsymbol{M} + \lambda \boldsymbol{D}^\top \boldsymbol{D} + \lambda_{TV} \nabla^\top \boldsymbol{\Phi}^\top \boldsymbol{\Phi} \nabla\right) \frac{\partial \boldsymbol{x}}{\partial \mathbf{t}_s} \\ &\quad + \left(\frac{\partial^2 \boldsymbol{x}}{\partial \mathbf{t}_i \partial \mathbf{t}_s}\right)^\top \Big(2\boldsymbol{M}^\top \boldsymbol{M}\boldsymbol{x} - 2\boldsymbol{M}^\top \boldsymbol{y} + 2\lambda \boldsymbol{D}^\top \boldsymbol{D}\boldsymbol{x} \\ &\quad + 2\lambda_{TV} \nabla^\top \boldsymbol{\Phi}^\top \boldsymbol{\Phi} \nabla \boldsymbol{x}\Big), \end{aligned} \tag{3.32}$$

where $s \neq i$ and

$$\frac{\partial x_k}{\partial t_i} = \begin{cases} -\frac{e^{t_i}}{\left(1+\sum_{j=2}^N e^{t_j}\right)^2} & \text{, for } k = 1 \\ -\frac{e^{t_i+t_k}}{\left(1+\sum_{j=2}^N e^{t_j}\right)^2} & \text{, for } k = 2, \ldots, N \;\&\; k \neq i \\ -\frac{e^{t_i+t_k}}{\left(1+\sum_{j=2}^N e^{t_j}\right)^2} + \frac{e^{t_i}}{1+\sum_{j=2}^N e^{t_j}} & \text{, for } k = 2, \ldots, N \;\&\; k = i \end{cases} \tag{3.33}$$

$$\frac{\partial^2 x_k}{\partial t_i^2} = \begin{cases} \frac{2e^{2t_i}}{\left(1+\sum_{j=2}^N e^{t_j}\right)^3} - \frac{e^{t_i}}{\left(1+\sum_{j=2}^N e^{t_j}\right)^2} & \text{, for } k = 1 \\ \frac{2e^{2t_i+t_k}}{\left(1+\sum_{j=2}^N e^{t_j}\right)^3} - \frac{e^{t_i+t_k}}{\left(1+\sum_{j=2}^N e^{t_j}\right)^2} & \text{, for } k = 2, \ldots, N \;\&\; k \neq i \\ \frac{2e^{2t_i+t_k}}{\left(1+\sum_{j=2}^N e^{t_j}\right)^3} - \frac{e^{t_i+t_k}+2e^{2t_i}}{\left(1+\sum_{j=2}^N e^{t_j}\right)^2} + \frac{e^{t_i}}{1+\sum_{j=2}^N e^{t_j}} & \text{, for } k = 2, \ldots, N \;\&\; k = i \end{cases} \tag{3.34}$$

$$\frac{\partial^2 x_k}{\partial t_i \partial t_s} = \begin{cases} \frac{2e^{t_i+t_s}}{\left(1+\sum_{j=2}^N e^{t_j}\right)^3} & \text{, for } k = 1 \\ \frac{2e^{t_k+t_i+t_s}}{\left(1+\sum_{j=2}^N e^{t_j}\right)^3} & \text{, for } k = 2, \ldots, N \;\&\; k \neq i, k \neq s \\ \frac{2e^{t_k+t_i+t_s}}{\left(1+\sum_{j=2}^N e^{t_j}\right)^3} - \frac{e^{t_i+t_s}}{\left(1+\sum_{j=2}^N e^{t_j}\right)^2} & \text{, for } k = 2, \ldots, N \;\&\; k = i \textit{ or } k = s\,, \end{cases} \tag{3.35}$$

where the partial derivatives $\partial \boldsymbol{x}/\partial \mathbf{t}$ are of dimension $N \times (N-1)$, $\partial f(\boldsymbol{x})/\partial \mathbf{x}$ of dimension $N \times 1$; $\partial f(\boldsymbol{x})/\partial \mathbf{t}$ of dimension $(N-1) \times 1$, $\partial^2 \boldsymbol{x}/\partial \mathbf{t}_i^2$ of dimension $N \times 1$, and $\partial^2 \boldsymbol{x}/\partial \mathbf{t}_i \partial \mathbf{t}_s$ of dimension $N \times 1$. The second order partial derivatives are $\partial^2 f(\boldsymbol{x})/\partial \mathbf{t}^2$ and $\partial^2 f(\boldsymbol{x})/\partial \mathbf{t}_i \partial \mathbf{t}_s$. These components form the Hessian matrix of size $(N-1) \times (N-1)$.

3.4 ROCSSUM unmixing algorithm

3.4.1 Trust region algorithm

The trust region algorithm works by approximating the function $f(\boldsymbol{x})$ one seeks to optimize by a simpler and more easily tractable function m, which reasonably reflects the behavior of f in a certain region Δ around a point $\boldsymbol{x}$. This region is called the trust region and Δ is the radius. The step size b in each iteration is obtained by minimizing the approximate function in the trust region [52], that is

$$\min_b m(b) \quad \text{s.t. } b \in \Delta. \tag{3.36}$$

If the trial step b results in a lower function value, then the current point will be updated to $\boldsymbol{x} + b$. Otherwise the trust region radius Δ will be for example shortened, and the process repeated.

In terms of approximating f, it is common practice to employ first and second order Taylor terms [114, 117]. The method can be made more sophisticated by extending the trial step search to the span of g and $B^{-1}g$ [51]. In summary, we have that

$$\begin{aligned} \min_b m(b) &= f + g^\top b + \frac{1}{2} b^\top B b \\ \text{s.t. } \|b\| &\leq \Delta,\ b \in \text{span}[g, B^{-1}g]. \end{aligned} \tag{3.37}$$

This approach is accurate and efficient [51]. Trust region algorithm has many advantages, that includes being reliable, robust and having very strong convergence properties [52]. The iterations are described in Algorithm 3.

Algorithm 3 Pseudocode of trust region optimization algorithm.

Input: f : function value w.r.t. $\boldsymbol{x}$, $\boldsymbol{g}$: gradient of size $(N-1) \times 1$, $\boldsymbol{B}$: Hessian of size $(N-1) \times (N-1)$.

Output: $\boldsymbol{x}$: estimated fractional abundance of size $(N-1) \times 1$.

Initialization:

1: $\Delta > 0$.

2: $b > 0$.

Main iteration:

3: Obtain b by approximately solving:
$\min_b m(b) = f + g^\top b + \frac{1}{2} b^\top B b$
s.t. $\|b\| \leq \Delta,\ b \in \text{span}[g, B^{-1}g]$.

4: **if** $f(\boldsymbol{x} + b) < f(\boldsymbol{x})$ **then**

5: $\quad \boldsymbol{x} = \boldsymbol{x} + b$.

6: **end if**

7: Update Δ

3.4.2 Numerical algorithm of ROCSSU

The sparsity and spatial regularization extensions for the unconstrained trust region algorithm with reparametrization of constraints are incorporated into the algorithm via a total variation regularizer, and a sparsity regularizer. The resulting ROCSSU pseudocode is presented in Algorithm 4. The algorithm starts with initialization in step 1. The constrained parameters are assigned equal proportional values according to the sum-to-one constraint, and then reparametrized to the unconstrained parameters. Within the algorithm's loop in step 3 the unconstrained parameters are converted to the constrained ones in order to find the weights for the regularization terms. The algorithm uses a weighted ℓ_2 norm to estimate the ℓ_p norm with $0 < p < 1$. The weights $\boldsymbol{D}$ and $\boldsymbol{\Phi}$ are defined as positive diagonal matrices with the values $\left(\left(\boldsymbol{x}^{(\alpha-1)}\right)^2 + \boldsymbol{\epsilon}^2\right)^{(p/2-1)/2}$ and $\left(\left(\nabla\boldsymbol{x}^{(\alpha-1)}\right)^2 + \boldsymbol{\eta}^2\right)^{(p/2-1)/2}$, respectively, on the diagonal. This is fixed-point iteration using the iterate $\boldsymbol{x}^{(\alpha-1)}$ to update the weights for iteration α. When $\boldsymbol{\epsilon} = \boldsymbol{\eta} = 0$ and $\boldsymbol{x}^{(\alpha-1)} = \boldsymbol{x}^{(\alpha)}$ then we have that $\|\boldsymbol{x}\|_p^p + \|\nabla\boldsymbol{x}\|_p^p = \|\boldsymbol{D}\boldsymbol{x}\|_2^2 + \left\|\boldsymbol{\Phi}\nabla\boldsymbol{x}\right\|_2^2$.

Algorithm 4 Pseudocode of the ROCSSU optimization algorithm with sparse spatial total variation regularization and reparametrization of constraints.

Task: Solve $\|\boldsymbol{y} - \boldsymbol{M}\boldsymbol{x}\|_2^2 + \lambda\|\boldsymbol{x}\|_p^p + \lambda_{TV}\|\nabla\boldsymbol{x}\|_p^p$, using reparametrization of constraints $\mathbf{t} = \log(\boldsymbol{x}/(1 - \sum_{j=2}^N x_j))$. Find $\boldsymbol{x}$.
Input: $\boldsymbol{M}_0$: $L \times N$ spectral library, $\boldsymbol{y}$: hyperspectral data of size $L \times P$, λ : sparsity regularization parameter, λ_{TV} : spatial regularization parameter, p : quasi-norm.
Output: $\boldsymbol{x}$: $N \times P$ fractional abundance matrix w.r.t. $\boldsymbol{M}_0$.

Initialization:

1: $\mathbf{x} \leftarrow [1, ..., 1]/N, \quad \mathbf{t} = \log\left(\frac{\boldsymbol{x}}{1-\sum_{j=2}^N x_j}\right)$.
2: $\boldsymbol{\epsilon} \leftarrow [1, ..., 1]$.

Iteration α:

3: $\boldsymbol{x} = \frac{e^{\mathbf{t}}}{1+\sum_{j=2}^N e^{t_j}}$.
4: $\boldsymbol{d} = \left(\left(\boldsymbol{x}^{(\alpha-1)}\right)^2 + \boldsymbol{\epsilon}^2\right)^{\frac{p}{2}-1}$.
5: $D = \mathrm{diag}(\boldsymbol{d}^{\frac{1}{2}})$.
6: $\boldsymbol{\phi} = \left(\left(\nabla\boldsymbol{x}^{(\alpha-1)}\right)^2 + \boldsymbol{\eta}^2\right)^{\frac{p}{2}-1}$.
7: $\boldsymbol{\Phi} = \mathrm{diag}(\boldsymbol{\phi}^{\frac{1}{2}})$.
8: $\boldsymbol{x}^{(\alpha)} \leftarrow \arg\min_{\boldsymbol{x}} \|\boldsymbol{y} - \boldsymbol{M}\boldsymbol{x}\|_2^2 + \lambda\|\boldsymbol{D}\boldsymbol{x}\|_2^2 + \lambda_{\mathrm{TV}}\left\|\boldsymbol{\Phi}\nabla\boldsymbol{x}\right\|_2^2$
s.t. $\mathbf{t} = \log\left(\frac{\boldsymbol{x}}{1-\sum_{j=2}^N x_j}\right)$.
9: $\boldsymbol{\epsilon}_n = \min\left(\boldsymbol{\epsilon}_n, \frac{\mathrm{sort}(\boldsymbol{x}_n)_{q+1}}{N}\right)$.

Post processing:

10: $\boldsymbol{x} = \frac{e^{\mathbf{t}}}{1+\sum_{j=2}^N e^{t_j}}$.

The function $\mathrm{sort}(\boldsymbol{x}_n)_{q+1}$ in step 9 of the algorithm rearranges the absolute values of

each pixel $\boldsymbol{x}_n$ into a decreasing sequence of numbers and selects the $(q+1)$th value for some fixed integer q. By definition, a vector $\boldsymbol{x}$ is *q-sparse* if and only if $r(\boldsymbol{x})_q = 0$ [34]. Thus the function $\text{sort}(\boldsymbol{x}_n)_{q+1}$ conveys information about how sparse each estimated solution vector $\boldsymbol{x}_n$ is into the value of $\boldsymbol{\epsilon}_n$ and to the updated weights w_n at each iteration.

The vector of residuals $\boldsymbol{\epsilon}$, together with the estimated solution, are used to update the weights for the sparsity inducing term $\|\boldsymbol{D}\boldsymbol{x}\|_2^2$. In addition, the estimated solution is used to calculate the differences between the fractional abundances of neighboring pixels and to give weights to the spatial regularization term in steps 6 and 7. Both of these new weights are used in the following iteration for the estimation of the solution (step 8).

Step 8 in Algorithm 4 is an optimization problem with a quadratic objective. We can adopt standard quadratic programming techniques, such as those described in [51], to solve it. We use $\|\boldsymbol{y} - \boldsymbol{M}\boldsymbol{x}\|_2^2 + \lambda \|\boldsymbol{D}\boldsymbol{x}\|_2^2 + \lambda_{TV} \left\|\boldsymbol{\Phi}\nabla\boldsymbol{x}\right\|_2^2$ together with the definition of reparametrization to express the function value, the gradient and the Hessian of the optimization problem. The derived gradient is given by Equations (3.16)-(3.17) and the Hessian by Equations (3.31)-(3.35). These are used in the trust region optimization of the objective function to update $\boldsymbol{x}$. The iterative process continues until the algorithm converges. As a final step (10), we convert the unconstrained working parameters $\mathbf{t}$ to the constrained fractional abundance parameters $\boldsymbol{x}$.

The purpose of the reparametrization is to take the constraints into account and to be able to carry out the estimation on real numbers. The sparsity constraint forces the abundances $\boldsymbol{x}$ to zero. If $x_i = 0$, then the reparametrized parameter $t_i = -\infty$ in Eq. (3.8), that can create instability in the optimization. This issue is avoided by the setup of the trust region algorithm that is bounded by definition [52], which is one of the advantages of the algorithm. As the optimization is only solved within the trust region, then t_i is bounded and would never reach $-\infty$. It follows that the trust region algorithm is pushing the x_i towards zero, but never exactly equating them to zero. However, the estimated coefficients that are very close to zero can be effectively treated as zero, thus we still achieve sparse results.

Concerning the computational complexity of the ROCSSU algorithm, the most costly step is the computation of the Hessian, which has the complexity of $\mathcal{O}(PN^2)$. Hence we can generalize that this is the computational complexity at each iteration. The complexity of the algorithm is comparable to other similar methods like SUnSAL-TV. Due to the number of endmembers having a big effect on the computational complexity of the algorithm, we included a library reduction step that is further detailed in the next section.

MUSIC library reduction

We include a library reduction step to the a sparse hyperspectral unmixing algorithm. We perform library pruning via MUSIC-Hysime algorithm seen in Alg. 2, where the HySime algorithm estimates the number of endmembers in the image and MUSIC determines the

endmembers from a spectral library. Here, we use the ROCSSU algorithm in Alg. 4 for solving the optimization problem in step 9 of the MUSIC in Alg. 2.

3.5 Experimental results

In this section, we employ the method of sparse total variation spatial regularization with library pruning and reparametrization of constraints using trust region optimization. We use both simulated and real world data to evaluate the performance of the proposed approach when compared to its competitors.

Performance Discriminators

The classical performance discriminator often used in the literature is the root mean square error (RMSE) [110]. The RMSE is defined as RMSE$= \sqrt{\frac{1}{NP}\sum_{n=1}^{P} \|\boldsymbol{x}_n - \hat{\boldsymbol{x}}_n\|^2}$. Here, $\boldsymbol{x}$ denotes the true vector of endmember fractional abundance values and $\hat{\boldsymbol{x}}$ the estimated one. A lower value indicates better performance for RMSE. We use the signal to reconstruction error (SRE) to assess the quality of reconstruction as it gives more information. SRE is measured in decibels: SRE(dB) $\equiv 10\log_{10}$(SRE), where SRE is defined as SRE $= \mathbb{E}\left[\|\boldsymbol{x}\|_2^2\right] / \mathbb{E}\left[\|\boldsymbol{x} - \hat{\boldsymbol{x}}\|_2^2\right]$. Higher values of the SRE(dB) indicate a superior unmixing performance. Note that in the rest of this paper we will use SRE as the short hand notation for SRE(dB). RMSE does not give as much information about the relationship between the power of the error and the power of the signal as SRE [28], but we report them both for comparison.

Simulated Data

To test how well the proposed algorithm works for material identification in hyperspectral images, we used the USGS digital spectral library released in September 2007. It can be retrieved from the Spectroscopy Lab web page under the name splib06.[1] It consists of the spectral reflectance of hundreds of materials measured in the lab. These spectral reflectance values have 224 bands that lie uniformly in the interval 0.4-2.5 μm. Many spectral signatures in the library represent the same material with minor differences between them. We therefore followed the common preprocessing steps of, e.g., [28], and pruned the library such that the minimum angle between any two signatures would not be smaller than 4.44°. The pruned library $\boldsymbol{M} \in \mathbb{R}^{224\times 240}$ included 240 different signatures.

We created a 75×75 simulated image with 224 spectral bands for each pixel with five randomly selected spectral signatures from the library $\boldsymbol{M}$ [28,37]. The data were generated using a linear mixing model, and the abundance's sum constraint was enforced on each pixel. The true abundances of the five endmembers in the simulated image are displayed in

[1] Available online: http://speclab.cr.usgs.gov/spectral.lib06/.

Fig. 2.2. The image has distinct square areas where there is only one endmember present (pure pixels) and areas that contain mixtures of two (fractional abundance of 1/2 each), three (1/3 each), four, and five endmembers. The background pixels surrounding the distinct square areas have abundance values of 0.1149, 0.0741, 0.2003, 0.2055 and 0.4052, respectively for the five endmembers. The simulated data were contaminated with white noise as well as spectrally correlated noise. The signal-to-noise ratio (SNR) was set to 40 dB (SNR $\equiv \mathbb{E}\left\|\boldsymbol{M}_0\boldsymbol{x}\right\|^2 / \mathbb{E}\left\|\boldsymbol{n}\right\|_2^2$) [28] . We set the maximum number of iterations to 400, and the number of endmembers after library pruning to 20 for these experiments.

Comparison with other algorithms

Table 3.2: Comparison of ROCSSUM at different noise levels

Method	SNR=40		SNR=30		SNR=20	
	SRE	RMSE	SRE	RMSE	SRE	RMSE
NCLS	5.50	16.79	0.91	24.74	-6.61	73.39
IRLS	8.59	11.48	5.60	15.68	1.53	24.60
SUnSAL	11.05	10.17	6.03	18.07	2.19	23.86
SUnSAL-TV	19.02	3.92	12.35	8.97	4.19	20.56
SUnSAL-CSR	12.85	8.57	6.45	16.96	2.68	23.32
MUSIC-CSR	**22.11**	**2.48**	12.35	7.27	4.74	16.85
RMUSIC-CSR	21.97	2.56	12.39	8.06	4.70	16.92
RMUSIC-DANSER	21.99	2.56	11.23	8.06	5.09	20.16
ROCSSUM	18.12	3.91	**15.23**	**5.16**	**5.47**	15.33
ROCSSURM	20.53	3.12	12.86	7.48	5.02	**15.32**

The simulated hyperspectral image in Fig. 2.2 was used to assess the performance of different sparse unmixing algorithms. Fig. 3.1 shows the abundance maps calculated using the optimal parameter values for λ, λ_{TV} and p by IRLS, SUnSAL-TV, MUSIC-CSR, RMUSIC-DANSER, ROCSSUM and ROCSSURM, for SNR= 40. We used five-fold cross-validation to find the optimal tuning values for all algorithms that were compared. Parameter values from -6 to 1 (with an interval of 1) were used for $\log_{10}(\lambda)$ and $\log_{10}(\lambda_{TV})$. Values from 0.1 to 1 (with an interval of 0.1) were used for p. The optimal parameter values for the proposed algorithms were found using five-fold cross-validation. These were $p = 0.5$ for all SNR, $\lambda = \lambda_{TV} = 10^{-4}$ for SNR= 40 and 10^{-3} for SNR= $30, 20$.

From Fig. 3.1 we observe that the use of the TV norm improved the results. SUnSAL-TV, ROCSSUM and ROCSSURM were visibly more accurate in finding the correct fractional abundances of the endmembers. The areas with high fractional abundance values were found more easily by the unmixing algorithms, whereas the regions with lower and very similar abundance fractions were more difficult to determine correctly. For endmember four, the difference of the true concentration of the background and the square boxes on the fifth line was only 0.0055 (they are 0.2 and 0.2055, respectively). Other unmixing algorithms struggled to deliver correct estimates in that region, but ROCSSUM and

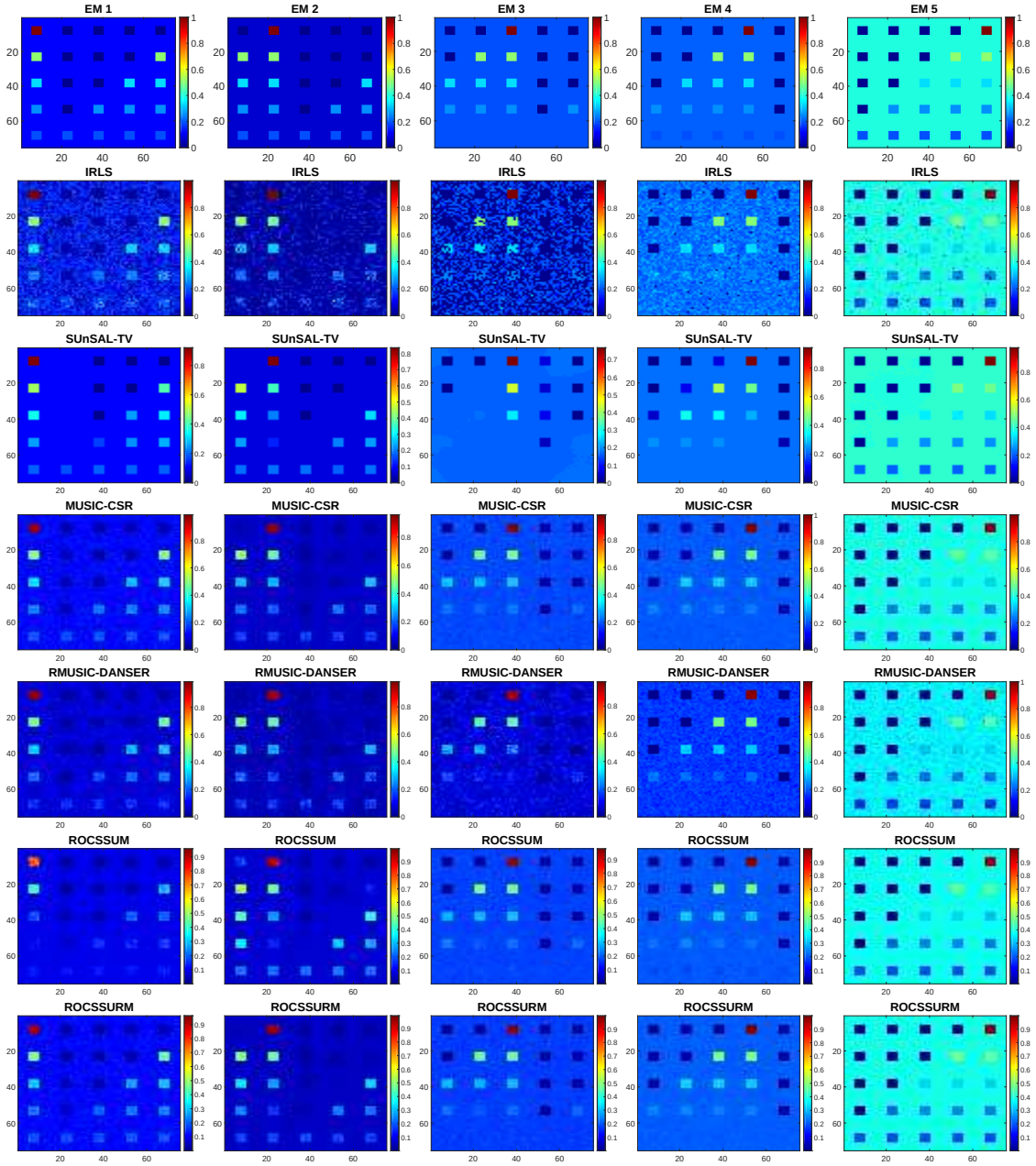


Figure 3.1: Fractional abundance maps obtained by IRLS, SUnSAL-TV, MUSIC-CSR, RMUSIC-DANSER, ROCSSUM, and ROCSSURM for endmember (EM) 1 in (a), EM 2 in (b), EM 3 in (c), and EM 4 in (d). SNR=40dB.

ROCSSURM managed it best.

The difference in the performance of algorithms with an increased noise can be seen in Fig. 3.2 and Fig. 3.3. The signal to noise ratio (SNR) was lowered to 30dB and 20dB in these images respectively. As the noise increased the performance of all algorithms deteriorated as expected. However, the largest change was seen in IRLS and SUnSAL-TV. The

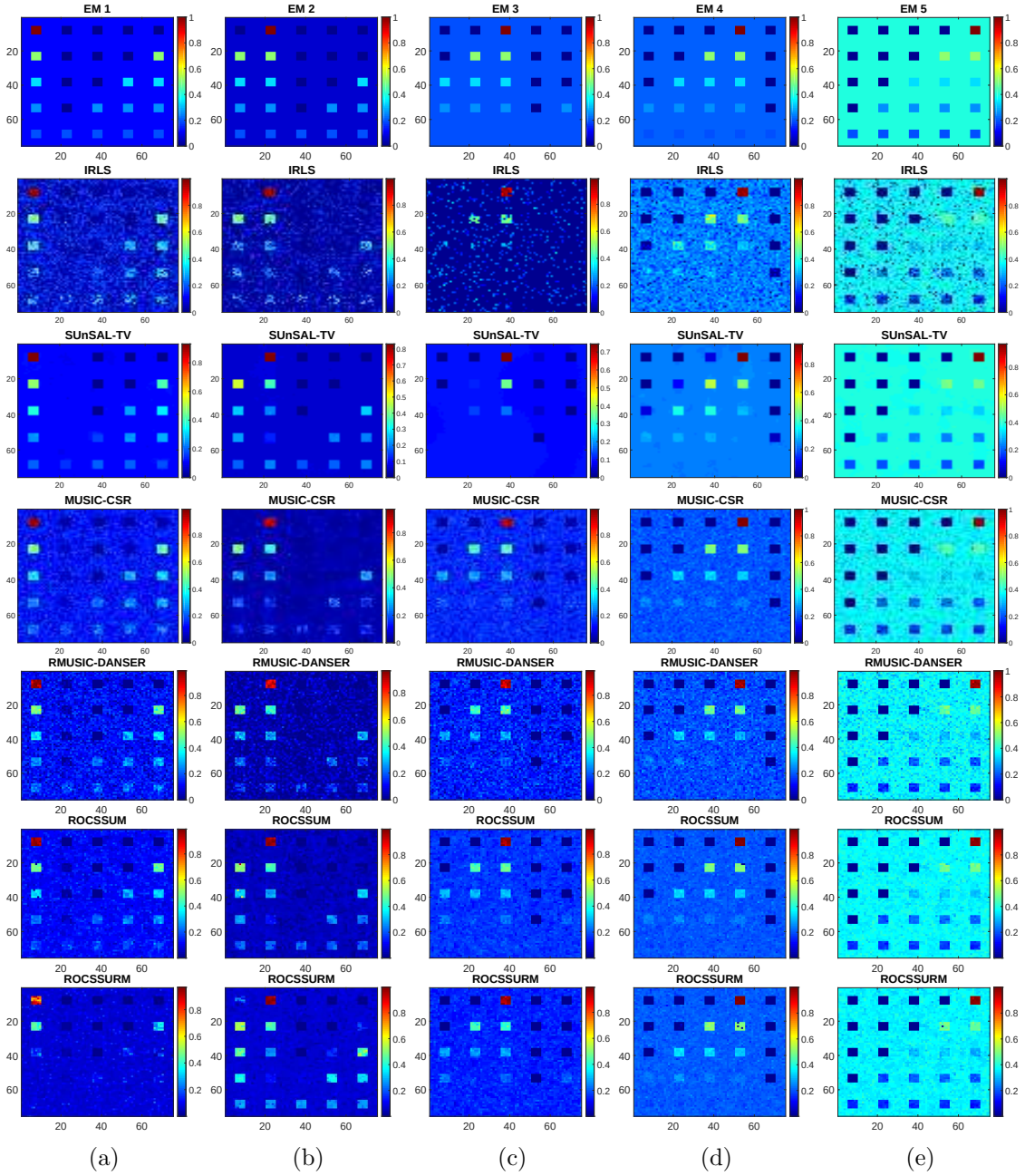


Figure 3.2: Fractional abundance maps obtained by IRLS, SUnSAL-TV, MUSIC-CSR, RMUSIC-DANSER, ROCSSUM, and ROCSSURM for endmember (EM) 1 in (a), EM 2 in (b), EM 3 in (c), and EM 4 in (d). SNR=30dB.

result of MUSIC-CSR and RMUSIC-DANSER became very noisy. However, ROCSSUM and ROCSSURM performed the best. This was also supported by the quantitative SRE values in Table 3.2.

Note that the SRE was calculated for each method of unmixing as an average. We used a 75×75 image for testing, which means that we estimated the fractional abundance of the endmembers for 5625 pixels. However, to see the variance of the performance from

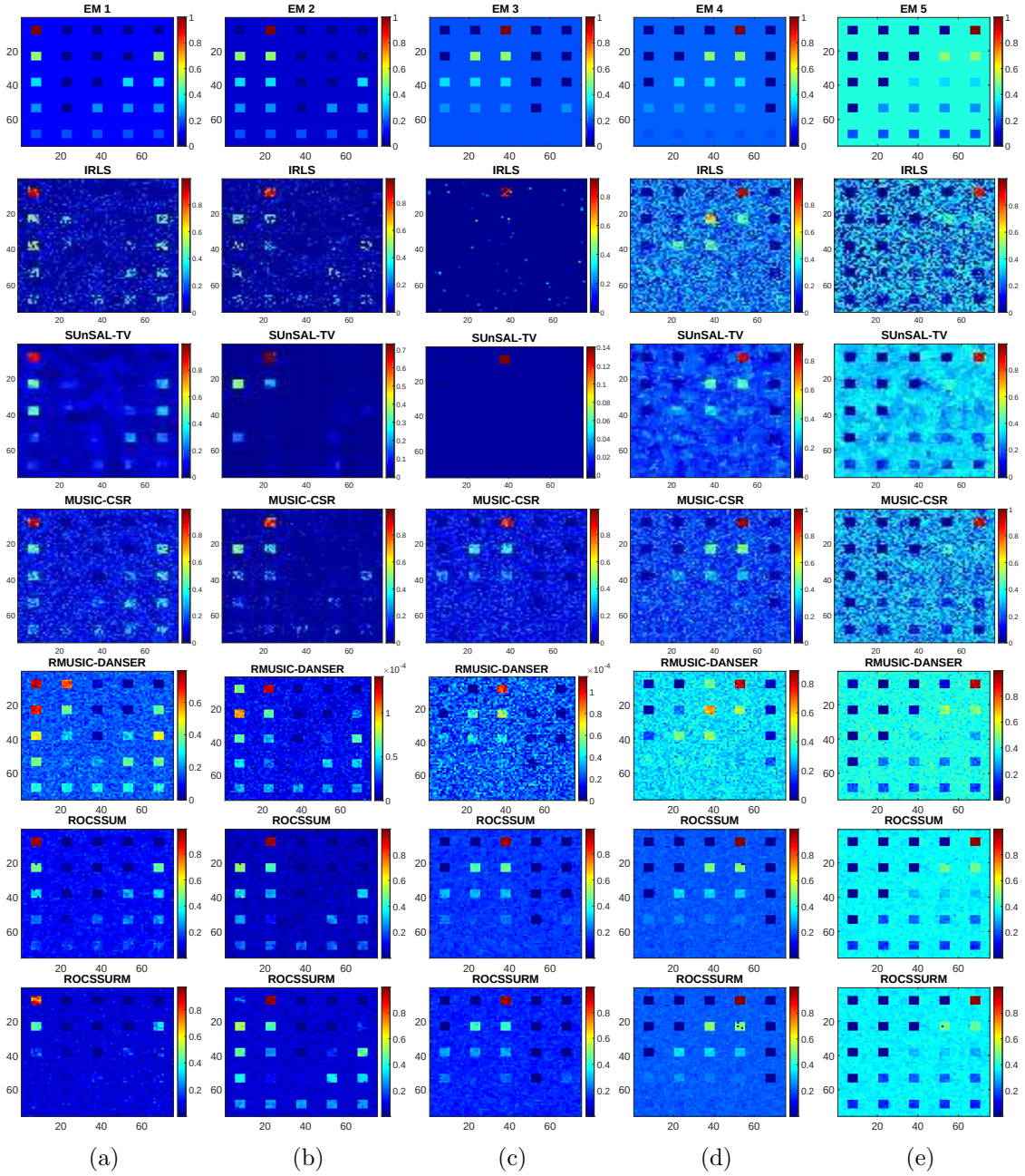


Figure 3.3: Fractional abundance maps obtained by IRLS, SUnSAL-TV, MUSIC-CSR, RMUSIC-DANSER, ROCSSUM, and ROCSSURM for endmember (EM) 1 in (a), EM 2 in (b), EM 3 in (c), and EM 4 in (d). SNR=20dB.

pixel to pixel we calculated the SRE for individual pixels and hence acquired the standard deviation for each method. These can be seen in Fig. 3.4 together with the best SRE results for each method for three different noise levels.

The traditional non-negatively constrained least squares (NCLS) method gave a poor solution compared to other algorithms. The results for the NCLS method were calculated using the SUnSAL algorithm with $\lambda = 0$. SUnSAL, SUnSAL-CSR and IRLS performed

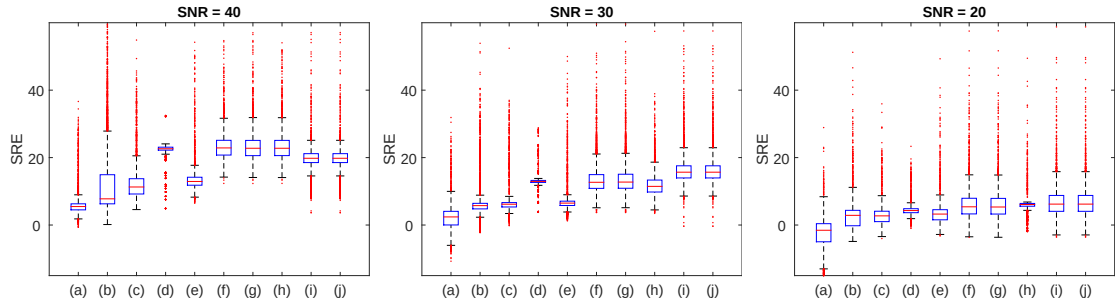


Figure 3.4: Individual SRE box-plots for three different noise levels for (a) NCLS, (b) IRLS, (c) SUnSAL, (d) SUnSAL-TV, (e) SUnSAL-CSR, (f) MUSIC-CSR, (g) RMUSIC-CSR, (h) RMUSIC-DANSER, (i) ROCSSUM, (j) ROCSSURM.

similarly with IRLS yielding a wider spread. SUnSAL-TV, MUSIC-CSR, RMUSIC-CSR, RMUSIC-DANSER, ROCSSUM and ROCSSURM had similar performance for SNR = 40, but ROCSSUM and ROCSSURM showed superior unmixing results at higher noise levels.

Monte Carlo simulation

Since the signal to noise ratio (SNR) in the images generally affects the unmixing results, we performed the experiments using four different noise levels SNR $\in \{10, 20, 30, 40\}$. The number of endmembers present in the spectrum of a pixel can also influence the outcome. Hence, we considered three different levels of endmember quantities mixed in the pixel such that N $\in \{3, 6, 9\}$. Here we performed 10 Monte Carlo simulations to account for the randomness in the noise.

For the simulated data experiments we used the endmember signatures from the USGS digital spectral library. It consists of the spectral reflectance of hundreds of materials measured in the lab. These spectral reflectance values have 224 bands that lie uniformly in the interval $0.4 - 2.5\mu$m. The chosen signatures can be seen in Fig. 2.10. We created a 50×50 image using the LMM in Eq. (1.1). Then we added Gaussian noise such that the noise level varies across the spectral bands. This kind of noise pattern is the most prevalent in real hyperspectral images [111]. The Dirichlet distribution was used for the uniform generation of the fractional abundance vectors $\boldsymbol{x}$. For this experiment we did not use the MUSIC library reduction step for the ROCSSU algorithm as the number of endmembers was already low. Hence we will call the algorithm ROCSSU from now on.

The SRE and RMSE performance measures for the quality of the reconstruction in the simulated image for different SNR levels and material densities are displayed in Table 3.3−3.5. Note that the RMUSIC-DANSER was shortened to RM-DANSER for these tables due to lack of space. The displayed performance measures are averages over 10 Monte-Carlo realizations for the simulated image. As before, the results indicated that as the number of endmembers in a pixel went up and the amount of noise in the image increased, the quality of image reconstruction of all algorithms decreased. SUnSAL and

SUnSAL-TV gave consistent results over all trialled noise and endmember levels. In high noise and low endmember density environment, SUnSAL produced the best results. IRLS and RMUSIC-DANSER produced mostly the lowest SRE and highest RMSE values. However, the RMUSIC-DANSER algorithm had good results for SNR= 40 and N= 9. The quality of the image reconstruction of the ROCSSU algorithm fell slightly behind the best performing method for high SNR when N= 3. However, in other noise and material density levels ROCSSU managed to produce the best results. The average running times over 10 Monte-Carlo realizations of the simulated image are presented in Table 3.6. It can be seen that including the reparametrization of constraints increased the computational complexity and hence resulted in a slightly longer running time than other methods.

Table 3.3: Comparison of unmixing performance of ROCSSU with various SNR, N=3

Method	SNR=40				SNR=30			
	SRE	std	RMSE	std	SRE	std	RMSE	std
NCLS	32.62	0.28	0.52	0.02	25.97	0.65	1.29	0.13
IRLS	14.74	0.12	6.32	0.08	13.68	0.18	7.15	0.15
SUnSAL	36.09	0.34	0.48	0.02	28.85	0.78	1.17	0.12
SUnSAL-TV	35.65	0.33	0.50	0.02	28.80	0.75	1.17	0.11
MUSIC-CSR	28.21	0.12	1.25	0.02	26.29	0.33	1.54	0.06
RM-DANSER	13.74	0.07	5.87	0.12	13.71	0.08	5.90	0.12
ROCSSU	30.82	0.46	1.08	0.03	28.53	0.42	1.30	0.05
Method	SNR=20				SNR=10			
	SRE	std	RMSE	std	SRE	std	RMSE	std
NCLS	17.56	0.46	3.92	0.24	9.84	0.52	10.36	0.78
IRLS	9.34	0.36	11.90	0.53	8.95	0.45	11.98	0.71
SUnSAL	20.14	0.69	3.23	0.27	10.77	0.76	9.45	0.89
SUnSAL-TV	20.16	0.69	3.23	0.27	10.75	0.77	9.46	0.91
MUSIC-CSR	20.17	0.85	3.25	0.37	**11.16**	0.72	**9.31**	0.84
RM-DANSER	11.85	2.57	8.46	3.69	4.54	1.40	20.53	3.15
ROCSSU	**20.40**	0.90	**3.21**	0.38	**11.63**	0.74	**8.92**	0.80

German Alpine foothills

A sub-image of the EnMAP data of German Alpine foothills image scene (Fig. 2.11) was used to test the performance of the proposed algorithm in a semi-supervised setting. The image has 244 spectral bands and it covers the 420nm-2460nm spectral range [112]. We used the N-FINDR algorithm to identify the spectral signatures of the endmembers in the image. It assumes the endmembers to be present in the image in the form of unmixed pixels [20]. The HySime algorithm estimated the signal subspace in the sub-image to be 3. Hence we used the N-FINDR algorithm to find the 3 endmember signatures from the Alpine scene. The estimated fractional abundance maps of the Alpine scene can be seen in Fig. 3.5. The optimal parameter values were calculated using cross-validation, hence we have $\lambda = \lambda_{TV} = 10$ and $p = 0.5$ for ROCSSU algorithm.

Table 3.4: Comparison with various SNR, N=6

Method	SNR=40				SNR=30			
	SRE	std	RMSE	std	SRE	std	RMSE	std
NCLS	24.64	0.45	0.98	0.07	16.52	0.66	2.79	0.26
IRLS	7.93	0.11	7.39	0.16	6.04	0.30	9.77	0.39
SUnSAL	27.21	0.48	0.89	0.06	18.35	0.73	2.51	0.24
SUnSAL-TV	25.90	0.34	1.11	0.04	18.43	0.71	2.48	0.23
MUSIC-CSR	19.26	0.07	2.60	0.01	17.29	0.37	3.06	0.14
RM-DANSER	3.64	0.09	15.82	0.19	4.19	1.34	14.73	2.55
ROCSSU	**27.25**	0.38	1.01	0.04	**18.99**	0.76	**2.40**	0.26
Method	SNR=20				SNR=10			
	SRE	std	RMSE	std	SRE	std	RMSE	std
NCLS	8.59	0.44	7.24	0.47	2.83	0.25	13.10	0.56
IRLS	3.09	0.10	13.79	0.39	3.16	0.20	12.07	0.49
SUnSAL	9.56	0.49	6.51	0.42	3.16	0.28	11.78	0.51
SUnSAL-TV	9.69	0.50	6.37	0.42	3.12	0.29	11.77	0.54
MUSIC-CSR	10.08	0.36	6.04	0.33	3.96	0.32	11.51	0.62
RM-DANSER	5.05	0.76	11.12	1.43	0.76	0.57	15.60	1.55
ROCSSU	**10.49**	0.44	**6.00**	0.38	**4.10**	0.36	**11.01**	0.61

Table 3.5: Comparison with various SNR, N=9

Method	SNR=40				SNR=30			
	SRE	std	RMSE	std	SRE	std	RMSE	std
NCLS	17.46	0.51	1.56	0.22	10.93	0.59	4.01	0.47
IRLS	2.81	0.07	8.76	0.05	1.72	0.12	9.58	0.19
SUnSAL	21.67	0.51	1.28	0.13	13.50	0.70	3.19	0.37
SUnSAL-TV	20.30	0.26	1.70	0.07	13.43	0.64	3.24	0.33
MUSIC-CSR	13.85	0.07	3.61	0.05	12.67	0.21	3.78	0.08
RM-DANSER	22.09	0.36	1.36	0.07	8.11	2.08	7.17	1.68
ROCSSU	**22.80**	0.61	**1.23**	0.15	**14.40**	0.75	**2.99**	0.36
Method	SNR=20				SNR=10			
	SRE	std	RMSE	std	SRE	std	RMSE	std
NCLS	4.91	0.15	7.88	0.35	0.43	0.26	12.30	0.78
IRLS	0.03	0.07	12.60	0.25	-0.31	0.13	10.76	0.37
SUnSAL	6.13	0.18	6.27	0.26	0.55	0.31	10.29	0.65
SUnSAL-TV	6.31	0.19	6.09	0.27	0.50	0.33	10.42	0.69
MUSIC-CSR	6.73	0.16	5.59	0.15	2.08	0.33	8.83	0.48
RM-DANSER	2.13	0.60	11.42	1.59	-2.41	0.64	18.10	2.48
ROCSSU	**7.48**	0.18	**5.41**	0.10	**2.10**	0.31	**8.77**	0.44

In Fig 3.5, we see that NCLS, SUnSAL-TV and RMUSIC-DANSER could not recover the water boundary. Furthermore, MUSIC-CSR and RMUSIC-DANSER were not able to clearly segment the grass from the forest (where ROCSSU showed patches of grass, the fraction of forest was typically less; this was not apparent in NCLS). ROCSSU obtained the best results for fractional abundances of water, and the lake boundary was

Table 3.6: ROCSSU average running times in 100s of seconds

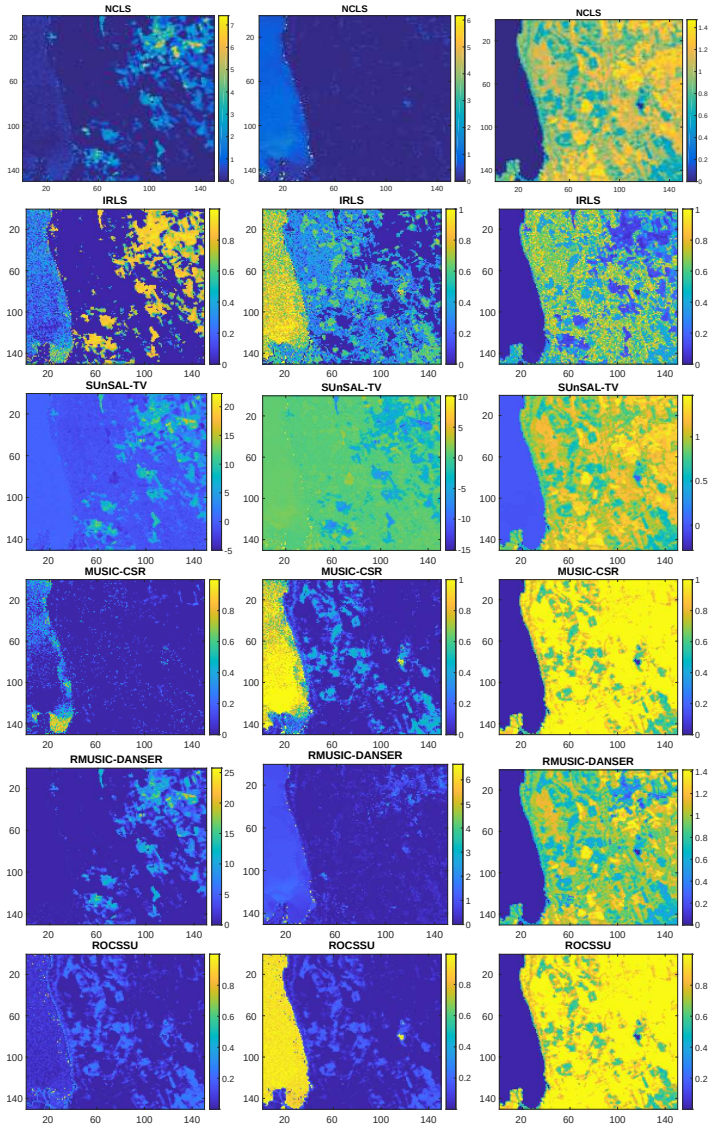
Method	N=3	N=6	N=9
NCLS	0.0003	0.0004	0.0015
IRLS	0.0175	0.0051	0.0090
SUnSAL	0.0002	0.0004	0.0011
SUnSAL-TV	0.0254	0.0310	0.0919
SUnSAL-CSR	0.0021	0.0025	0.0049
RMUSIC-DANSER	0.0092	0.0318	0.0443
ROCSSU	0.2286	1.0440	20.0059

very clear. MUSIC-CSR did a fairly good job at recovering the water boundary, but it had a higher proportion of water estimates on the grasslands. The forest and grass areas were qualitatively better estimated by the ROCSSU. Also, NCLS, RMUSIC-DANSER and SUnSAL-TV produced unrealistic fractional abundance values that were negative and/or larger than one.

Jasper Ridge

Experiments were performed on the Jasper Ridge real hyperspectral image, see Fig. 2.13. The image has 224 spectral bands in the $[380nm, 2500nm]$ range with a spectral resolution up to 9.46nm. Due to dense water and atmospheric effects, the low SNR bands 1-3, 108-112, 154-166, and 220-224 were removed, and 198 channels remained. We used a semi-supervised approach using N-FINDR in this experiment. The HySime algorithm estimated the signal subspace in the image to be 4. Hence we used the N-FINDR algorithm to find 4 endmember signatures from the hyperspectral image scene. The respective optimal parameter values were computed using cross-validation, thus $\lambda = 10^{-3}$, $\lambda_{TV} = 10^{-2}$ and $p = 0.3$ for ROCSSU algorithm. The fractional abundance maps for tree, water, soil, and road are shown in Fig. 3.6.

In Fig. 3.6, we see that both RMUSIC-DANSER and SUnSAL-TV had fractional abundance values larger than 1. Moreover, SUnSAL-TV had many negative fractional abundance values: this is not realistic. Besides these quantitative disadvantages of SUnSAL-TV, ROCSSU also seemed to offer more detail in, e.g., the water, tree and soil maps in Fig. 3.6. ROCSSU and MUSIC-CSR were the only methods that clearly identified the water areas. However MUSIC-CSR did not manage to suppress abundance values of water in tree and soil areas as well as ROCSSU. IRLS seemed to identify the areas of trees best, but failed in correctly determining the abundance of water and road. None of the methods performed particularly well in finding the road area in the image, but the ROCCSU results looked visually the clearest. SUnSAL-TV struggled to define the areas with these endmembers, whereas for ROCSSU, the water was well defined and the tree areas surrounding the roads were well mapped. Furthermore, in Fig. 3.6, ROCSSU appeared to attribute large abundance to water and tree pixels much more consistently.



(a) (b) (c)

Figure 3.5: Fractional abundance maps of the German Alpine foothills data set estimated by NCLS, IRLS, SUnSAL-TV, MUSIC-CSR, RMUSIC-DANSER and ROCSSU. The abundance map of forest is shown in (a), water in (b), and grass in (c).

3.6 Discussion

The statistical problem of spectral unmixing of hyperspectral data has several limitations. In this chapter, we focused on the constraints imposed on the fractional abundances that are being estimated. Enforcing the sum-to-one and non-negativity constraints on the model's parameters is crucial for obtaining realistic results.

A new algorithm called ROCSSU was introduced to overcome these limitations. We introduced a reparametrization of coefficients, to account for the sum-to-one and non-negativity constraints. We included a library reduction step similar to the MUSIC array

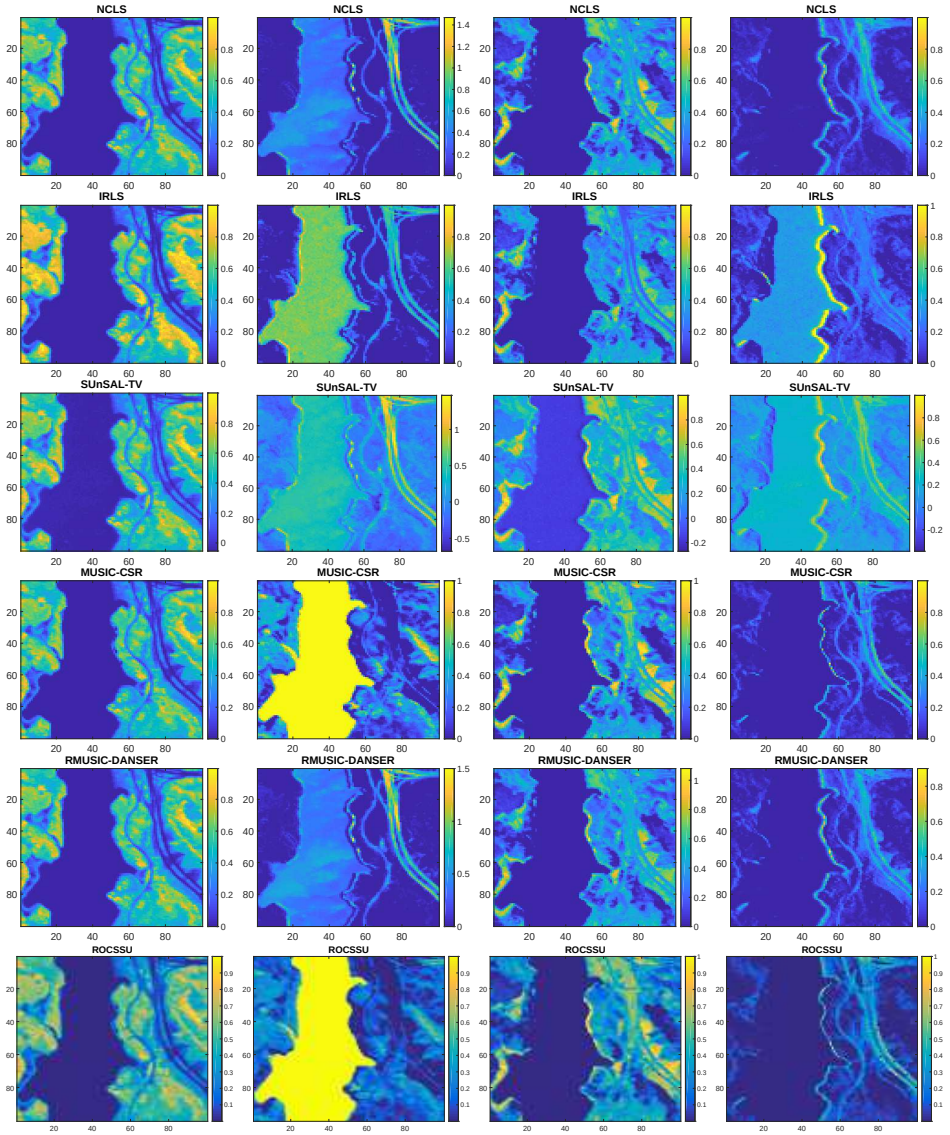


Figure 3.6: Fractional abundance maps for the Jasper Ridge data obtained by NCLS, IRLS, SUnSAL-TV, MUSIC-CSR, RMUSIC-DANSER, ROCSSU, for tree in (a), water in (b), soil in (c), and road in (d).

processing algorithm, employed a spatial total variation regularization term, and enforced sparsity on the estimates via ℓ_p norm. Our experimental results supported the benefits of the approach. Specifically, reparametrization of the model's coefficients was an effective and efficient way of taking the constraints into account. In addition to that, changing the ℓ_1 norm on sparsity to ℓ_p norm with $0 < p < 1$ improved the unmixing performance. Including spatial information via the TV term produced better results. The library reduction step made the computation faster and improved the unmixing result significantly where the initial library size was large. We have demonstrated that the proposed algorithm

showed improved performance in noisy situations as compared to other similar unmixing algorithms.

Comparison

The ROCSSU algorithm, that used reparametrization to account for the constraints on the fractional abundances, was slower than the IRLSTV algorithm introduced in the previous chapter. IRLSTV also enforced the constraints, but did so in a heuristic manner, using renormalization and projection onto the non-negative orthant. The heuristic constraints allowed for a few different benefits for the algorithm. It kept the computational complexity low, which made the algorithm significantly faster, and allowed the method to be used in an IRLS framework that has very strong convergence properties. The ROCSSU algorithm gave better results in some scenarios and IRSLTV in others. For the Monte Carlo simulation experiment, ROCSSU showed mostly superior unmixing performance to IRLSTV, especially when the signal-to-noise-ratio in the image was lower. This can be easily seen in the summary Tables 3.7-3.9. The largest difference in the performance of the two methods is for high SNR and small number of endmembers. In that case, IRLSTV obtains a much superior unmixing performance. Also, when we compare teh estimated fractional abundance maps in Fig. 2.14 and Fig. 3.6 for the real hyperspectral image Jasper Ridge, we can identify more desirable performance by IRLSTV. The algorithm is consistently allocating high fractional abundance values to the regions of same materials and it distinguishes between different endmembers in the image very well. Hence, although the reparametrization of coefficients has benefits, it did not work ideally in every setting.

Table 3.7: Comparison of unmixing performance of IRLSTV and ROCSSU, N=3

Method	SNR=40				SNR=30			
	SRE	std	RMSE	std	SRE	std	RMSE	std
IRLSTV	**37.85**	0.47	**0.42**	0.02	**29.47**	0.72	**1.13**	0.10
ROCSSU	30.82	0.46	1.08	0.03	28.53	0.42	1.30	0.05
Method	SNR=20				SNR=10			
	SRE	std	RMSE	std	SRE	std	RMSE	std
IRLSTV	20.35	0.81	3.23	0.30	10.86	0.68	9.41	0.76
ROCSSU	**20.40**	0.90	**3.21**	0.38	**11.63**	0.74	**8.92**	0.80

Table 3.8: Comparison with various SNR, N=6

Method	SNR=40				SNR=30			
	SRE	std	RMSE	std	SRE	std	RMSE	std
IRLSTV	**28.20**	0.46	**0.88**	0.05	18.67	0.48	2.47	0.19
ROCSSU	27.25	0.38	1.01	0.04	**18.99**	0.76	**2.40**	0.26
Method	SNR=20				SNR=10			
	SRE	std	RMSE	std	SRE	std	RMSE	std
IRLSTV	10.27	0.54	**5.98**	0.44	**4.10**	0.08	**8.39**	0.13
ROCSSU	**10.49**	0.44	6.00	0.38	**4.10**	0.36	11.01	0.61

Table 3.9: Comparison with various SNR, N=9

Method	SNR=40				SNR=30			
	SRE	std	RMSE	std	SRE	std	RMSE	std
IRLSTV	22.19	0.52	1.27	0.10	13.62	0.41	3.17	0.20
ROCSSU	**22.80**	0.61	**1.23**	0.15	**14.40**	0.75	**2.99**	0.36
Method	SNR=20				SNR=10			
	SRE	std	RMSE	std	SRE	std	RMSE	std
IRLSTV	6.80	0.33	5.48	0.21	**4.05**	0.04	**5.91**	0.07
ROCSSU	**7.48**	0.18	**5.41**	0.10	2.10	0.31	8.77	0.44

4

Robust method

This chapter begins with reasons for the extension of the traditional least-squares model, and highlights the necessity for robust methods. It gives a brief overview of the currently available robust estimation methods for hyperspectral unmixing. The chapter then continues to introduce a new algorithm for robust unmixing of hyperspectral imagery. The performance of the algorithm is tested on synthetic and real hyperspectral images and compared with other state-of-the-art algorithms.

4.1 Introduction

The linear mixing model (LMM) that is traditionally used for spectral unmixing assumes that mixed spectra is a linear combination of endmembers, and that the errors follow a Gaussian distribution. Ergo the unmixing problem amounts to identifying the best subset of endmember signatures to synthesize the spectra in each pixel. The LMM assigns the randomness in the model to the observational errors, which are often used in a least-squares strategy to optimize the model. However, the assumption about random errors generally does not hold for real hyperspectral imagery data. Typically, some bands in the hyperspectral images are very noisy, with almost no signal at all. This can be often due to water absorption on some wavelengths or certain atmospheric effects. Hence, there are often a small proportion of observations, called *outliers*, divergent from the rest of the data. Even one large outlier can decidedly deform the results of the least-squares method [86].

A robust approach to statistical modeling can give protection from the influence of outliers [87] and produce reliable parameter estimates [88]. Hence the ability to diminish the effect of such outliers in noisy environments can offer greater robustness. When the data does not contain outliers, the robust methods produce similar results to classical methods, but when there are outliers present, the robust methods generate results similar

to that of the classical methods when applied to data without the outliers. Thus robust methods are great for identifying outliers and for modeling the majority of the data [88].

Outliers can influence data-based methods used to ascertain the tuning or penalty constants adopted in the smoothing steps [89], therefore [90, 91] used robust extensions to Generalized Additive Models. The non-Gaussian heavy-tailed relaxation, extending the standard LMM to robust method, motivated in hyperspectral image segmentation [94] and modeling papers [95] can be accommodated by the recently proposed regularization models and approaches for unmixing with a few modifications to the likelihood term and algorithmic framework. [96] replaced the least-squares based weights in their iteratively weighted least squares (IRLS) algorithm with others derived from quasi likelihood equations to reduce the influence of outliers. Recently, robust extensions have been used for the spectral unmixing problem [1, 97, 118].

In this chapter, we propose to extend the traditional least-squares method to the robust heavy-tailed case and use M-estimation, which is a formal approach to robust estimation. Hence we use a fixed M-estimate function $\rho(\boldsymbol{e})$ for the unmixing problem to restrain outliers. Hampel's three-part re-descending function behaves approximately like the least squares function when the errors are small. As the errors increase, the function decreases their influence or even assigns them zero weight [119]. The threshold parameters of Hampel's function are determined according to the variance of the estimation error, and applied to manage the size of the constraint on the outliers. This offers the proposed method greater robustness to noise and outliers, and enhanced performance results.

Here, we propose to enforce sparsity on the fractional abundance estimation via an ℓ_p norm with $0 < p < 1$. This encourages more sparsity than the ℓ_2 or ℓ_1 norm and improves the performance of the method. The problem is non-convex and non-differentiable, but we can approximate it by a weighted ℓ_2 norm. We also use the spectral-spatial information that is available in hyperspectral data using a spatial total variation regularization term $|\nabla \boldsymbol{x}|$. Rather than apply the ℓ_1 using the ADMM method [37], we propose to invoke more sparsity in the results via the ℓ_p norm with $0 < p < 1$. The resulting ℓ_p-ℓ_2 optimization can be carried out without difficulty in the framework of our new iteratively reweighted algorithm. Experiments on both synthetic data and real hyperspectral images demonstrate the improved performance of the proposed method compared to its least squares counterparts and other robust/bilinear methods.

4.2 Mixing Model

4.2.1 Robust M-Estimates

The objective function for M-estimation is denoted by $\rho(\boldsymbol{e})$, where $\boldsymbol{e} = \boldsymbol{y} - \boldsymbol{M}\boldsymbol{x}$ and ρ is the function that is used to minimize the residuals. M-estimate functions are flexible and they can be generalized to multiparameter problems with no difficulty [120]. They are

more robust to outliers and excessive noise, when the signal-to-noise ration is low, than the standard least squares function $\|\boldsymbol{e}\|^2$ [119]. However, it can be problematic to find a good estimate to $\rho(\boldsymbol{e})$. Generally an appropriate fixed function is chosen for M-estimation to reduce the impact of noise. The robust M-estimate function is defined as

$$\min_{\boldsymbol{x}} \quad \sum \rho\left(\boldsymbol{y} - \boldsymbol{M}\boldsymbol{x}\right) \ . \tag{4.1}$$

where $\boldsymbol{y}, \boldsymbol{x}, \boldsymbol{M}$, of the size $LP \times 1$, $NP \times 1$ and $LP \times NP$ respectively, describe the unmixing problem over all pixels jointly, and are defined as

$$\begin{aligned} \boldsymbol{y} &= \left[\boldsymbol{y}_1^\top, \boldsymbol{y}_2^\top, \ldots, \boldsymbol{y}_P^\top\right]^\top , \\ \boldsymbol{x} &= \left[\boldsymbol{x}_1^\top, \boldsymbol{x}_2^\top, \ldots, \boldsymbol{x}_P^\top\right]^\top , \\ \boldsymbol{M} &= I_P \otimes \boldsymbol{M}_0 \ , \end{aligned} \tag{4.2}$$

Here, $\otimes$ is the Kronecker product, $\boldsymbol{y}_n \in \mathbb{R}^{L\times 1}$ denotes the spectral reflectance of the nth pixel, the $N \times 1$ fractional abundance vector $\boldsymbol{x}_n$ holds the N endmember abundances of pixel n, and $\boldsymbol{M}_0 \in \mathbb{R}^{L\times N}$ is the spectral library that generates the block-diagonal matrix $\boldsymbol{M}$ with P entries of $\boldsymbol{M}_0$ on the main diagonal.

The robust function $\rho(\boldsymbol{e})$ is commonly approximated by Huber's function [88]. The Huber function curtails the effect of noise with large amplitudes to a fixed level, but it does not remove their influence. Hampel's method has a similar performance to Huber's for small errors, but it also has the ability to greatly reduce the significance of very noisy observations or even assign them zero weights. The difference between the outlier suppression of the Huber's and Hampel's method can be seen and compared to the least squares function in Fig.4.1.

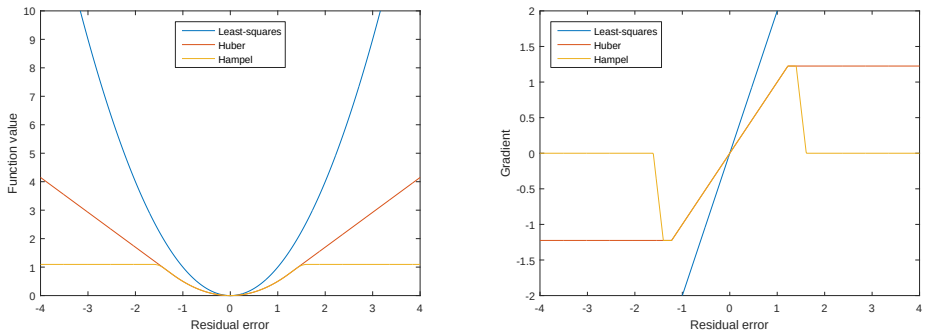


Figure 4.1: Illustration of the least squares objective function, robust Huber function and robust Hampel's function in terms of the residual error (left). Their respective gradients are on the right.

We use Hampel's function for increased outlier suppression. The Hampel's three part

re-descending function and its gradient are as follows:

$$\rho(\boldsymbol{e}) = \begin{cases} \frac{\boldsymbol{e}^2}{2} & ,0 \leq |\boldsymbol{e}| < \xi_a \\ \frac{\xi_a|\boldsymbol{e}|-\xi_a^2}{2} & ,\xi_a \leq |\boldsymbol{e}| < \xi_b \\ \frac{\xi_a\left[\xi_b+\xi_c-\xi_a+\frac{(|\boldsymbol{e}|-\xi_c)^2}{\xi_b-\xi_c}\right]}{2} & ,\xi_b \leq |\boldsymbol{e}| < \xi_c \\ \frac{\xi_a(\xi_b+\xi_c)}{2} - \frac{\xi_a^2}{2} & ,\xi_c \leq |\boldsymbol{e}| \end{cases} \tag{4.3}$$

$$\psi(\boldsymbol{e}) = \frac{\partial\rho(\boldsymbol{e})}{\partial\boldsymbol{e}} = \begin{cases} \boldsymbol{e} & ,0 \leq |\boldsymbol{e}| < \xi_a \\ sign\,(\boldsymbol{e})\cdot\xi_a & ,\xi_a \leq |\boldsymbol{e}| < \xi_b \\ \frac{sign(\boldsymbol{e})\cdot(|\boldsymbol{e}|-\xi_c)\xi_a}{\xi_b-\xi_c} & ,\xi_b \leq |\boldsymbol{e}| < \xi_c \\ 0 & ,\xi_c \leq |\boldsymbol{e}| \end{cases} \tag{4.4}$$

where $\psi(\boldsymbol{e})$ is the derivative of $\rho(\boldsymbol{e})$, and ξ_a, ξ_b, ξ_c are threshold parameters.

The threshold parameters control the magnitude of outlier suppression in M-estimation. They are estimated based on the variance of impulse-free estimation error σ^2. The threshold parameters are defined as $\xi_a = 1.96\cdot\hat{\sigma}$, $\xi_b = 2.24\cdot\hat{\sigma}$, $\xi_c = 2.58\cdot\hat{\sigma}$, where $\hat{\sigma}$ is the robust noise variance estimator $\hat{\sigma} = median\left(\left|\boldsymbol{y}_n - \boldsymbol{y}_{n-1}\right|\right) / \left(\sqrt{2}\cdot 0.6745\right)$, and $n = 2, ..., P$ [119, 121].

We adopt the chain rule to differentiate the robust Hampel's function with respect to $\boldsymbol{x}$. It becomes

$$\frac{\partial f}{\partial \boldsymbol{x}} = \left(\frac{\partial \boldsymbol{e}}{\partial \boldsymbol{x}}\right)^\top \frac{\partial f}{\partial \boldsymbol{e}} \,. \tag{4.5}$$

Thus

$$\begin{aligned} \frac{\partial \boldsymbol{e}}{\partial \boldsymbol{x}} &= \frac{\partial}{\partial \boldsymbol{x}}(\boldsymbol{y} - \boldsymbol{M}\boldsymbol{x}) = -\boldsymbol{M} \,, \\ \frac{\partial f}{\partial \boldsymbol{e}} &= \psi \,, \\ \frac{\partial f}{\partial \boldsymbol{x}} &= -\boldsymbol{M}^\top\psi \,. \end{aligned} \tag{4.6}$$

Setting the above equation in (4.6) to zero yields that the M-estimator $\boldsymbol{x}$ has to satisfy $-\boldsymbol{M}^\top\psi = 0$. However, it generally does not have an analytical solution. Let $\boldsymbol{W} = \mathrm{diag}\,(\boldsymbol{w}(\boldsymbol{e}))$, and

$$\boldsymbol{w}(\boldsymbol{e}) = \begin{cases} \psi\,(e)\,/e & ,e \neq 0 \\ \psi'\,(e) & ,e = 0 \end{cases} \tag{4.7}$$

where $w(e) = 1$ when $e = 0$ for Hampel's function. After substituting ψ from Eq.(4.7)

into (4.6) and setting it to zero, we get $-\boldsymbol{M}^\top \boldsymbol{W}(\boldsymbol{y} - \boldsymbol{M}\boldsymbol{x}) = 0$. Hence $\boldsymbol{x}$ becomes

$$\boldsymbol{x} = \left(\boldsymbol{M}^\top \boldsymbol{W}\boldsymbol{M}\right)^{-1} \boldsymbol{M}^\top \boldsymbol{W}\boldsymbol{y}. \tag{4.8}$$

The values $\boldsymbol{W}$ can be described as robustness weights and used for outlier identification. If the weight in $\boldsymbol{W}$ for an observation is very small or even zero, then it is considered to be an outlier by the M-estimator. The weights $\boldsymbol{W}$ depend on the errors $\boldsymbol{e}$ and consequently on the fractional abundance estimate $\boldsymbol{x}$. Hence an iterative procedure has to be used in order to compute $\boldsymbol{x}$. This fits neatly into the framework of our iterative algorithm.

4.2.2 Sparsity Regularization

We propose to solve the following sparse optimization problem

$$\min_{\boldsymbol{x}} \quad \rho\left(\boldsymbol{y} - \boldsymbol{M}\boldsymbol{x}\right) + \lambda \left\|\boldsymbol{x}\right\|_p^p, \tag{4.9}$$

where $0 < p < 1$, λ is the regularization parameter that controls the sparsity of the solution and $\boldsymbol{y}, \boldsymbol{x}, \boldsymbol{M}$ are defined as in (4.2). The above optimization problem in (4.9) is non-convex and non-differentiable, therefore challenging to estimate. It can be approximated by a differentiable weighted ℓ_2 norm:

$$\begin{aligned} \min_{\boldsymbol{x}} \quad & \rho\left(\boldsymbol{y} - \boldsymbol{M}\boldsymbol{x}\right) + \lambda \left\|\boldsymbol{D}\boldsymbol{x}\right\|_2^2 , \\ \text{with} \quad & \boldsymbol{D} = \operatorname{diag}\left(\boldsymbol{d}^{\frac{1}{2}}\right),\ \boldsymbol{d} = \left(\left(\boldsymbol{x}^{(\alpha-1)}\right)^2 + \boldsymbol{\epsilon}^2\right)^{\frac{p}{2}-1} , \end{aligned} \tag{4.10}$$

where $\boldsymbol{D}$ is a positive diagonal matrix containing the weights, $\boldsymbol{\epsilon}$ is the estimation error for the whole image, $\boldsymbol{\epsilon}_n$ for the nth pixel is a vector of size $N \times 1$, $\boldsymbol{\epsilon}$ covers the whole image s.t. $\left[\boldsymbol{\epsilon}_1^\top, \boldsymbol{\epsilon}_2^\top, \ldots, \boldsymbol{\epsilon}_P^\top\right]^\top$, size $NP \times 1$, and the weights $\boldsymbol{d}$ are calculated using the $\boldsymbol{x}^{(\alpha-1)}$ results from the previous iteration $\alpha - 1$. Hence the weighted ℓ_2 norm in (4.10) is a first-order approximation to the ℓ_p norm in (4.9). As the algorithm converges, $\boldsymbol{\epsilon}$ goes to zero and $\boldsymbol{x}^{(\alpha-1)} \cong \boldsymbol{x}^{(\alpha)}$. It can be shown that when $\boldsymbol{\epsilon} = 0$ and $\boldsymbol{x}^{(\alpha-1)} = \boldsymbol{x}^{(\alpha)}$then $\left\|\boldsymbol{D}\boldsymbol{x}\right\|_2^2 = \left\|\boldsymbol{x}\right\|_p^p$. Hence the weighted ℓ_2 norm is a good approximation of the ℓ_p norm for sparsity regularization.

The reformulation of the optimization problem (4.10), simplifies the method and allows the estimation to be performed in an iterative manner. The algorithm would start by initializing the weights and continue to iteratively update the estimate of $\boldsymbol{x}$ by solving the quadratic problem (4.10), and update the weights $\boldsymbol{D}$. This is a fixed-point iteration for solving the optimization problem (4.9).

4.2.3 Spatial Regularization

Here we extend the sparse optimization problem to include an additional term to exploit the spatial-contextual information in hyperspectral data. The spatial total variation (TV) regularization term is as follows:

$$\min_{\boldsymbol{x}} \quad \rho\left(\boldsymbol{y}-\boldsymbol{M}\boldsymbol{x}\right)+\lambda\left\|\boldsymbol{x}\right\|_p^p+\lambda_{TV}\left\|\nabla\boldsymbol{x}\right\|_p^p\ , \tag{4.11}$$

where $\boldsymbol{y},\boldsymbol{x},\boldsymbol{M}$ are defined as in (4.2). The regularization parameter λ_{TV} controls the impact of this term compared to the sparsity and robust terms. A greater value for λ_{TV} will increase the smoothness in the result, i.e. the neighbouring pixels will have more similar fractional abundance values for same endmembers.

The total variation term enforces piecewise constant shift in the fractional abundances of the same endmember among neighbouring pixels. The ∇ operator computes the difference in the estimated fractional abundance values for neighboring pixels in the image. It is composed of two parts $\nabla=[\nabla_h^\top,\nabla_v^\top]^\top$, where ∇_h is for taking horizontal differences and ∇_v for vertical differences. The horizontal differences are $\nabla_h\mathbf{x}=[\mathbf{a_1},\mathbf{a_2},...,\mathbf{a_m}]^\top$, where $\mathbf{a_n}=\mathbf{x_n}-\mathbf{x_{n_h}}$, with n and n_h denoting a pixel and its horizontal neighbor respectively. The operation of ∇_v is analogous for vertical neighbors. The difference operator ∇ is a matrix consisting mostly of zeros, it only has one 1 and -1 on each row for the neighboring horizontal or vertical pixels. As $\boldsymbol{x}$ is a column vector with the fractional abundances of all endmembers and all pixels, then the distance between the 1 and -1 in the rows of ∇_h is the number of endmembers N. Thus we take the difference between $x_n[i]$ and $x_{n_h}[i]$ for endmember i, pixels n and its horizontal neighbor n_h. For ∇_v the distance is cN, where c defines the number of pixels in each row.

We rewrite the ℓ_p optimization problem in (4.11) as a weighted ℓ_2 norm, to get

$$\begin{aligned}
\min_{\boldsymbol{x}} \quad & \rho\left(\boldsymbol{y}-\boldsymbol{M}\boldsymbol{x}\right)+\lambda\left\|\boldsymbol{D}\boldsymbol{x}\right\|_2^2+\lambda_{TV}\left\|\boldsymbol{\Phi}\nabla\boldsymbol{x}\right\|_2^2, \\
\text{with} \quad & \boldsymbol{D}=\operatorname{diag}(\boldsymbol{d}^{\frac{1}{2}}), \quad \boldsymbol{d}=\left(\left(\boldsymbol{x}^{(\alpha-1)}\right)^2+\boldsymbol{\epsilon}^2\right)^{\frac{p}{2}-1}, \\
& \boldsymbol{\Phi}=\operatorname{diag}(\boldsymbol{\phi}^{\frac{1}{2}}), \quad \boldsymbol{\phi}=\left(\left(\nabla\boldsymbol{x}^{(\alpha-1)}\right)^2+\boldsymbol{\eta}^2\right)^{\frac{p}{2}-1},
\end{aligned} \tag{4.12}$$

where $\boldsymbol{\eta}$ is a column vector containing small integer values of 10^{-6} that allows to avoid division by zero, $\boldsymbol{\Phi}$ is a positive diagonal matrix containing the weights of the spatial regularization term, and the weights $\boldsymbol{d}$ and $\boldsymbol{\phi}$ are calculated using the $\boldsymbol{x}^{(\alpha-1)}$ results from the previous iteration $\alpha-1$. For small enough $\boldsymbol{\eta}$ and for $\boldsymbol{x}^{(\alpha-1)}=\boldsymbol{x}^{(\alpha)}$, we have that $\left\|\boldsymbol{\Phi}\nabla\boldsymbol{x}\right\|_2^2=\left\|\nabla\boldsymbol{x}\right\|_p^p$.

4.3 Unmixing algorithm

In this section, we introduce a new algorithm based on iterative reweighing to optimize the robust function along with sparsity and total variation regularizers. The pseudo-code of the iteratively reweighted robust algorithm (IRRF) can be seen in Alg.5. The algorithm uses a weighted ℓ_2 norm to estimate the ℓ_p norm with $0 < p < 1$.

Algorithm 5 Pseudocode of IRRF.

Task: Solve $\rho\left(\boldsymbol{y}-\boldsymbol{M}\boldsymbol{x}\right)+\lambda\left\|\boldsymbol{x}\right\|_p^p+\lambda_{TV}\left\|\nabla\boldsymbol{x}\right\|_p^p$,
s.t. $\boldsymbol{x}\geq 0, \boldsymbol{a}\boldsymbol{x}_n=1$. Find $\boldsymbol{x}$.
Parameters: ϵ_{thr}: convergence threshold, j_{max}: maximum number of iterations.
Input: $\boldsymbol{M}_0$: $L\times N$ spectral library, $\boldsymbol{Y}$: $L\times P$ hyperspectral data matrix, λ : sparsity regularization parameter, λ_{TV} : spatial regularization parameter, p : quasi-norm.
Output: $\boldsymbol{X}$: $N\times P$ fractional abundance matrix w.r.t. $\boldsymbol{M}_0$.

Initialization:

$\boldsymbol{\epsilon}\leftarrow[1,...,1]$, $\boldsymbol{\epsilon}_{thr}\leftarrow[\epsilon_{thr},...,\epsilon_{thr}]$.
$\boldsymbol{d}\leftarrow[1,...,1]$, $\boldsymbol{D}=\mathrm{diag}\left(\boldsymbol{d}^{\frac{1}{2}}\right)$.
$\boldsymbol{x}\leftarrow\underset{\boldsymbol{x}}{\arg\min}\left\|\boldsymbol{y}-\boldsymbol{M}\boldsymbol{x}\right\|_2^2+\lambda\left\|\boldsymbol{D}\boldsymbol{x}\right\|_2^2$
s.t. $\boldsymbol{x}\geq 0$, $\boldsymbol{a}\boldsymbol{x}_n=1$ for $n=1,...,P$.
$j\leftarrow 1$.

Main iteration for iteration α:

while $\mathrm{sum}(\boldsymbol{\epsilon}>\boldsymbol{\epsilon}_{thr})>0$ **do**
  $\boldsymbol{d}\leftarrow\left(\left(\boldsymbol{x}^{(\alpha-1)}\right)^2+\boldsymbol{\epsilon}^2\right)^{\frac{p}{2}-1}$.
  $\boldsymbol{D}=\mathrm{diag}\left(\boldsymbol{d}^{\frac{1}{2}}\right)$.
  $\boldsymbol{\phi}=\left(\left(\nabla\boldsymbol{x}^{(\alpha-1)}\right)^2+\boldsymbol{\eta}^2\right)^{\frac{p}{2}-1}$.
  $\boldsymbol{\Phi}=\mathrm{diag}\left(\boldsymbol{\phi}^{\frac{1}{2}}\right)$.
  $\boldsymbol{x}^{(\alpha)}\leftarrow\underset{\boldsymbol{x}}{\arg\min}\sum\rho\left(\boldsymbol{y}-\boldsymbol{M}\boldsymbol{x}\right)+\lambda\left\|\boldsymbol{D}\boldsymbol{x}\right\|_2^2+\lambda_{\mathrm{TV}}\left\|\boldsymbol{\Phi}\nabla\boldsymbol{x}\right\|_2^2$
   s.t. $\boldsymbol{x}\geq 0$, $\boldsymbol{a}\boldsymbol{x}_n=1$ for $n=1,...,P$.
  $\boldsymbol{\epsilon}_n\leftarrow\min\left(\boldsymbol{\epsilon}_n,\frac{\mathrm{sort}(\boldsymbol{x}_n)_{q+1}}{N}\right)$.
  if $j>j_{max}$ **then**
    break;
  end if
  $j=j+1$.
end while

To begin with, the weights and the sparsity measure $\boldsymbol{\epsilon}$ are initialized. The weights for the sparsity and the spatial total variation term are positive diagonal matrices with the entries $\left(\left(\boldsymbol{x}^{(\alpha-1)}\right)^2+\boldsymbol{\epsilon}^2\right)^{(p/2-1)/2}$ and $\left(\left(\nabla\boldsymbol{x}^{(\alpha-1)}\right)^2+\boldsymbol{\eta}^2\right)^{(p/2-1)/2}$ respectively on the diagonal. This is fixed-point iteration using the iterate $\boldsymbol{x}^{(\alpha-1)}$ to update the weights for iteration α. It can be shown that when $\boldsymbol{\epsilon}=\boldsymbol{\eta}=0$ and $\boldsymbol{x}^{(\alpha-1)}=\boldsymbol{x}^{(\alpha)}$ then $\left\|\boldsymbol{x}\right\|_p^p+\left\|\nabla\boldsymbol{x}\right\|_p^p=\left\|\boldsymbol{D}\boldsymbol{x}\right\|_2^2+\left\|\boldsymbol{\Phi}\nabla\boldsymbol{x}\right\|_2^2$, hence as the algorithm converges the weighted ℓ_2 norm converges to

the required ℓ_p norm. The ∇ function computes the horizontal and vertical differences in the fractional abundances between neighbouring pixels in the image. $\boldsymbol{\eta}$ is a column vector containing fixed small integers of size $\approx 10^{-6}$ that is included to avoid infinite values in step 10. The function $\text{sort}(\boldsymbol{x}_n)_{q+1}$ in step 11 orders the absolute values of $\boldsymbol{x}_n$ in each pixel in a decreasing sequence and outputs the $(q+1)$th value for a fixed integer q. If $\text{sort}(\boldsymbol{x})_q = 0$, the vector $\boldsymbol{x}$ is *q-sparse* [34]. Accordingly, the information about the sparsity of $\boldsymbol{x}_n$ is transmitted through $\boldsymbol{\epsilon}_n$ to the updated weights $\boldsymbol{d}$ and $\boldsymbol{D}$ at each iteration. The iterative process continues until all $\boldsymbol{\epsilon}_n$ for $n = 1, \ldots, P$ are below the threshold level ϵ_{thr}, and the optimal solution of Eq. (4.11) has been found.

Step 10 in Algorithm 5 is an optimization problem with a quadratic objective and linear constraints. We can adapt standard quadratic programming techniques, such as those described in [51], to solve it. The function $\rho\left(\boldsymbol{y} - \boldsymbol{M}\boldsymbol{x}\right) + \lambda \left\|\boldsymbol{D}\boldsymbol{x}\right\|_2^2 + \lambda_{TV} \left\|\boldsymbol{\Phi}\nabla\boldsymbol{x}\right\|_2^2$ can be rewritten as

$$\begin{aligned} f(\boldsymbol{x}) &= \rho\left(\boldsymbol{y} - \boldsymbol{M}\boldsymbol{x}\right) + \lambda \left(\boldsymbol{D}\boldsymbol{x}\right)^\top \left(\boldsymbol{D}\boldsymbol{x}\right) + \lambda_{TV} \left(\boldsymbol{\Phi}\nabla\boldsymbol{x}\right)^\top \left(\boldsymbol{\Phi}\nabla\boldsymbol{x}\right) \\ &= \rho\left(\boldsymbol{y} - \boldsymbol{M}\boldsymbol{x}\right) + \lambda \boldsymbol{x}^\top \boldsymbol{D}^\top \boldsymbol{D}\boldsymbol{x} + \lambda_{TV} \boldsymbol{x}^\top \nabla^\top \boldsymbol{\Phi}^\top \boldsymbol{\Phi}\nabla\boldsymbol{x}\,, \end{aligned} \tag{4.13}$$

where $\boldsymbol{\Phi}$ is a diagonal matrix containing the weights of the TV term. Here the weight matrices $\boldsymbol{D}$ and $\boldsymbol{\Phi}$ are kept fixed since they are computed using the previous iteration value $\boldsymbol{x}^{(\alpha-1)}$. Thus the partial derivatives of $f(\boldsymbol{x})$ with respect to $\boldsymbol{x}$ are

$$\begin{aligned} \frac{\partial f(\boldsymbol{x})}{\partial \boldsymbol{x}} &= -\boldsymbol{M}^\top \psi + \lambda \left(\boldsymbol{D}^\top \boldsymbol{D} + \left(\boldsymbol{D}^\top \boldsymbol{D}\right)^\top\right) \boldsymbol{x} + \lambda_{TV} \left(\nabla^\top \boldsymbol{\Phi}^\top \boldsymbol{\Phi}\nabla + \left(\nabla^\top \boldsymbol{\Phi}^\top \boldsymbol{\Phi}\nabla\right)^\top\right) \boldsymbol{x} \\ &= -\boldsymbol{M}^\top \psi + 2\lambda \boldsymbol{D}^\top \boldsymbol{D}\boldsymbol{x} + 2\lambda_{TV} \nabla^\top \boldsymbol{\Phi}^\top \boldsymbol{\Phi}\nabla\boldsymbol{x}\,. \end{aligned} \tag{4.14}$$

After equating (4.14) to zero and following the steps to (4.8), we get $\left(\boldsymbol{M}^\top \boldsymbol{W}\boldsymbol{M} + \lambda \boldsymbol{D}^\top \boldsymbol{D} + \lambda_{TV} \nabla^\top \boldsymbol{\Phi}^\top \boldsymbol{\Phi}\nabla\right) \boldsymbol{x} = \boldsymbol{M}^\top \boldsymbol{W}\boldsymbol{y}$. Expressing $\boldsymbol{x}$ gives:

$$\boldsymbol{x} = \left(\boldsymbol{M}^\top \boldsymbol{W}\boldsymbol{M} + \lambda \boldsymbol{D}^\top \boldsymbol{D} + \lambda_{TV} \nabla^\top \boldsymbol{\Phi}^\top \boldsymbol{\Phi}\nabla\right)^+ \boldsymbol{M}^\top \boldsymbol{W}\boldsymbol{y} \tag{4.15}$$

where $\boldsymbol{A}^+$ defines the pseudo-inverse of the matrix $\boldsymbol{A}$ and $\boldsymbol{x}$ is the estimated solution to step 10 in the IRRF algorithm. The constraints on the fractional abundance estimate $\boldsymbol{x}$ are applied in an iterative manner. First the non-negativity constraint is enforced using a projection onto the positive orthant, then the sum-to-one constraint is enforced by renormalization of the estimated solution for each pixel. Although the constraints are implemented in a heuristic manner, it integrates well into the iteratively reweighted algorithm. It keeps the algorithm simple and fast, and produces good results.

The resulting IRRF algorithm is simple but also very flexible as the ℓ_p norm penalty terms can be easily integrated. The convergence of the algorithm is very fast compared to others, e.g. ADMM and gradient decent, it needs to do only a small number of iterations. The algorithm involves a considerable inversion procedure at each iteration as seen

in Eq.(4.15), which makes it computationally costly. The complexity of the problem is $\mathcal{O}(LN^2)$ because at each iteration the weighted ℓ_p-ℓ_2 minimization is required. Nonetheless, it generally takes less computational time than comparative methods due to the remarkably fast convergence.

4.4 Experimental results

This section details the use of both synthetic and real hyperspectral images for analyzing the unmixing performance of the iteratively reweighted robust function (IRRF). The simulated data experiments measure the results quantitatively and the real data analysis shows qualitative performance.

We use the signal to reconstruction error (SRE) and root mean square error (RMSE) to validate the unmixing performance for simulated data. The SRE is measured in decibels and is denoted as SRE(dB) SRE(dB) $= 10\log_{10}\left(\mathbb{E}\left[\|\boldsymbol{x}\|_2^2\right]/\mathbb{E}\left[\|\boldsymbol{x}-\hat{\boldsymbol{x}}\|_2^2\right]\right)$, where $\boldsymbol{x}$ is the true fractional abundances of endmembers and $\hat{\boldsymbol{x}}$ the estimated ones. The greater the SRE value, the more superior the reconstruction of the image. The RMSE is defined as RMSE$= \sqrt{\frac{1}{NP}\sum_{n=1}^{P}\|\boldsymbol{x}_n-\hat{\boldsymbol{x}}_n\|^2}$. Here, a lower value indicates a better reconstruction of the image.

Comparison with other algorithms

We chose two fully constrained least squares based algorithms for comparison: Fully Constrained Sparse Unmixing via variable Splitting and Augmented Lagrangian (SUnSAL-FC) [28], Iteratively Reweighted Least Squares (IRLS) [35]. We further examined the following recent robust/bilinear methods: Fully Constrained Correntropy-based Unmixing by Variable Splitting and Augmented Lagrangian (CUSAL-FC) [97], Coordinate Decent Algorithm with Mismodeling Effects (CDA-ME) [93], Robust Linear Mixing Model (rLMM) [92], Block Coordinate Descent Alternating Direction Method of Multipliers (BCD/ADMM) [99], Unsupervised Generalized Normal Compositional Model (UsGNCM) [98].

Simulated Data

We tested the performance of the method in material identification in hyperspectral images and used the USGS digital spectral library. We created a 75×75 simulated image with 224 spectral bands for each pixel with five randomly selected spectral signatures from the spectral library [28,37]. The data were generated using a linear mixing model, and the sum-to-one constraint was enforced on each pixel. The true abundances of the five endmembers in the simulated image are displayed in Fig. 2.2. The image has distinct square areas where there is only one endmember present (pure pixels) and areas that contain mixtures of two ,

three, four, and five endmembers. The background pixels surrounding the distinct square areas have abundance values of 0.1149, 0.0741, 0.2003, 0.2055 and 0.4052, respectively for the five endmembers. The simulated data were contaminated with white noise as well as spectrally correlated noise. The signal-to-noise ratio was set to 40dB, 30dB and 20dB. We set the maximum number of iterations to 50, and the number of endmembers after MUSIC library pruning step to 20 for these experiments.

The simulated hyperspectral image in Fig. 2.2 was used to assess the performance of different sparse unmixing algorithms. Table 4.1 displays the quantitative unmixing results, and Fig. 4.2 shows the abundance maps for SNR= 40. We used five-fold cross-validation to find the optimal tuning values for the unmixing. The optimal parameter values were $\lambda = \lambda_{TV} = 10^{-3}$, $p = 0.5$.. For this experiment we have not displayed the results of CUSAL-FC and UsGNCM due to the methods taking excessively long to compute and not getting presentable results respectively.

Table 4.1: Comparison of IRRF at different noise levels

Method	SNR=40		SNR=30		SNR=20	
	SRE	RMSE	SRE	RMSE	SRE	RMSE
IRLS	8.59	11.48	5.60	15.68	1.53	24.60
SUnSAL-FC	11.13	9.98	6.02	16.92	2.23	24.03
CDA-ME	-15.30	231.85	-13.44	185.72	-16.33	261.49
rLMM	0.05	37.53	0.05	37.53	0.04	37.53
BCD/ADMM	2.27	29.37	2.38	28.41	-3.13	37.26
IRRF	**32.00**	**0.71**	**22.77**	**2.44**	**11.21**	**7.86**

From Fig. 4.2 we see that the least-squares based methods, IRLS and SUnSAL-FC, reconstructed the simulated image better than some of the robust methods. The abundance maps of CDA-ME and rLMM were very noisy, and the algorithms failed to identify many of the square areas, where the materials had same fractional abundance values. The BCD/ADMM appeared to assign same proportions to neighbouring pixels more consistently as the abundance maps are very smooth. However, this often lead to larger areas in the image being misidentified. The IRRF method produced the best reconstruction of the images, with the squares being distinctly displayed from the background. These results were corroborated by the quantitative performance measures seen in Table 4.1.

The difference in the performance of algorithms with an increased noise can be seen in Fig. 4.3 and Fig. 4.4. The signal to noise ratio (SNR) was lowered to 30dB and 20dB in these images respectively. The increased noise in the image resulted in a lower unmixing performance for all algorithms. This can be seen in the displayed abundance maps as the noise in the images increases. For $SNR = 20$, noise with hardly any regularity can be seen for the abundance maps of CDA-ME and rLMM. The IRRF algorithm still retained the best unmixing performance with many of the regions in the image correctly identified.

The SRE image reconstruction measure calculated for each unmixing method is truly

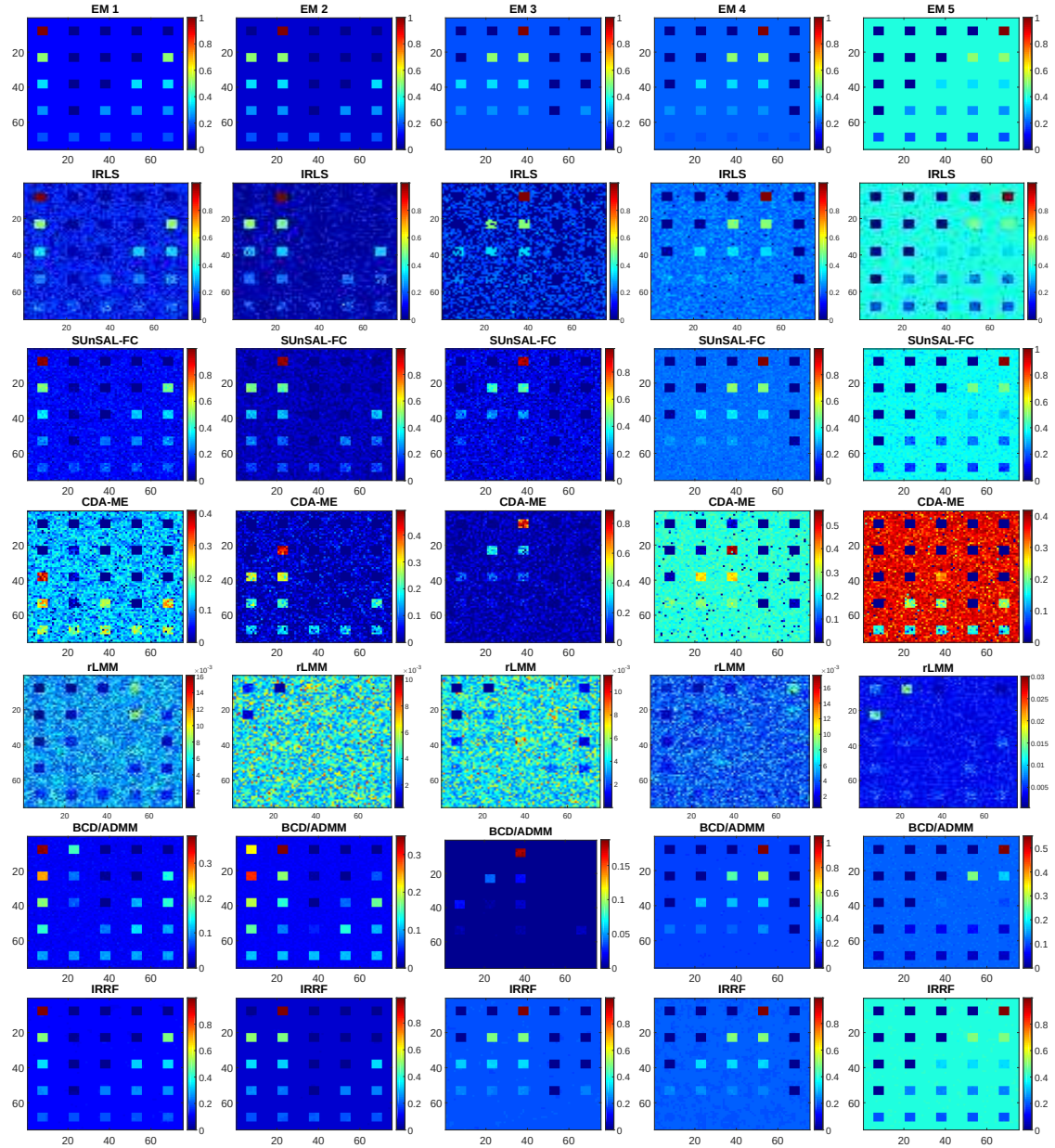


Figure 4.2: Fractional abundance maps obtained by IRLS, SUnSAL-FC, CDA-ME, rLMM, BCD/ADMM, and IRRF for endmember (EM) 1 in (a), EM 2 in (b), EM 3 in (c), and EM 4 in (d). SNR=40dB.

an average. In this experiment, we used a 75×75 pixel image for testing, hence we estimated the fractional abundance of the endmembers for 5625 pixels. Previously, we calculated the SRE for the whole image, which is an average SRE of the 5625 pixels. However, to see the variance of the performance from pixel to pixel, we calculated the SRE for individual pixels and acquired the standard deviation for each method. These can be seen in Fig. 4.5 together with the best SRE results for each method for three

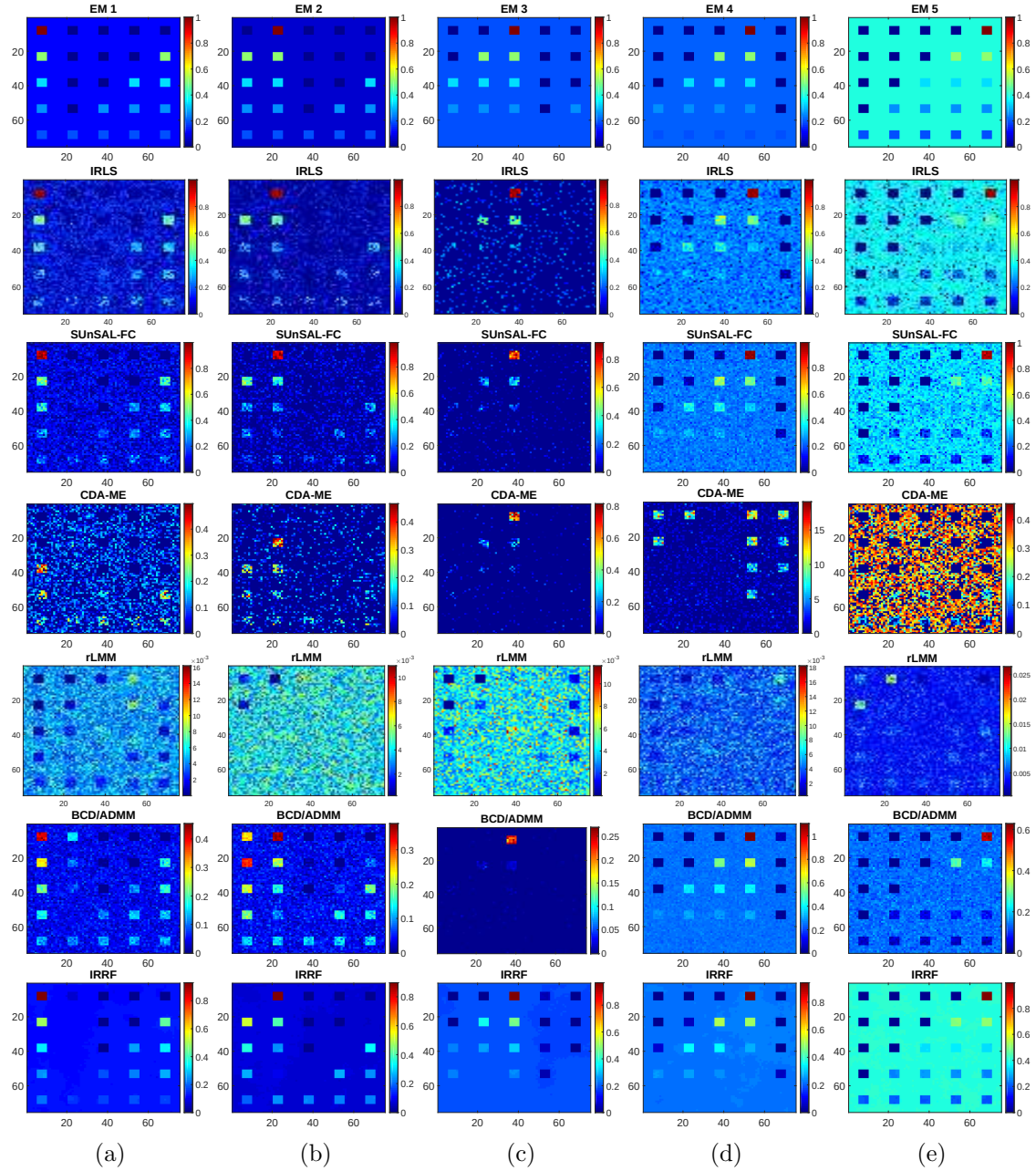


Figure 4.3: Fractional abundance maps obtained by IRLS, SUnSAL-FC, CDA-ME, rLMM, BCD/ADMM, and IRRF for endmember (EM) 1 in (a), EM 2 in (b), EM 3 in (c), and EM 4 in (d). SNR=30dB.

different noise levels.

For this experiment, the least-squares based methods IRLS and SUnSAL-FC outperformed CDA-ME, rLMM and BCD/ADMM. The spread of the individual SRE values for IRLS and SUnSAL-FC was smaller for noisier environments. The introduced IRRF method showed the best results overall, although it's quality of reconstruction also lowered with increased noise. The spread of the SRE values for IRRF increased for noisy images,

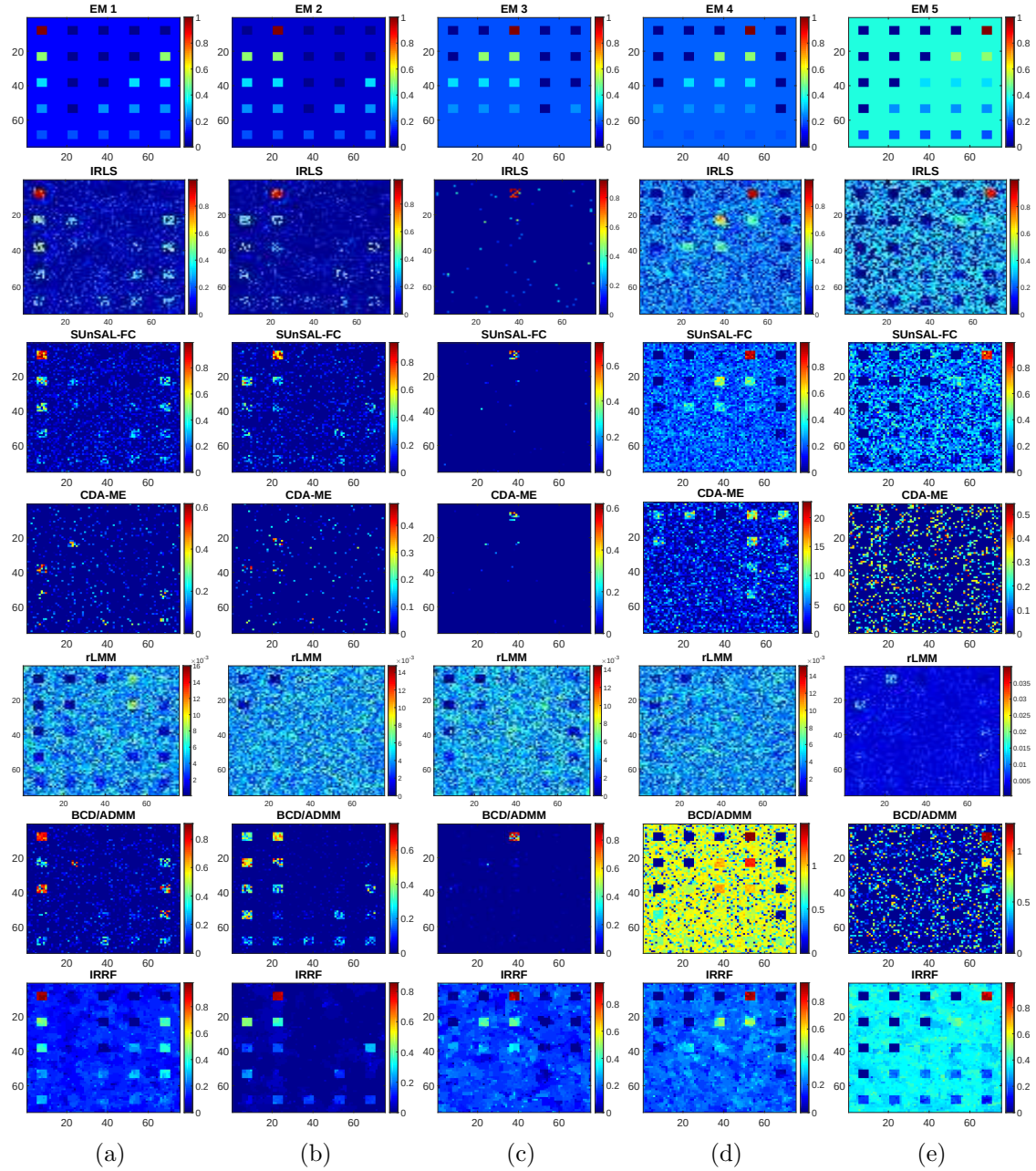


Figure 4.4: Fractional abundance maps obtained by IRLS, SUnSAL-FC, CDA-ME, rLMM, BCD/ADMM, and IRRF for endmember (EM) 1 in (a), EM 2 in (b), EM 3 in (c), and EM 4 in (d). SNR=20dB.

but still produced the highest SRE.

Monte Carlo simulation

Here we performed unmixing on synthetic hyperspectral data to evaluate the proposed model in parallel with other state-of-the-art methods. The signal to noise ratio (SNR) in the images and the number of endmembers present in each pixel can affect the un-

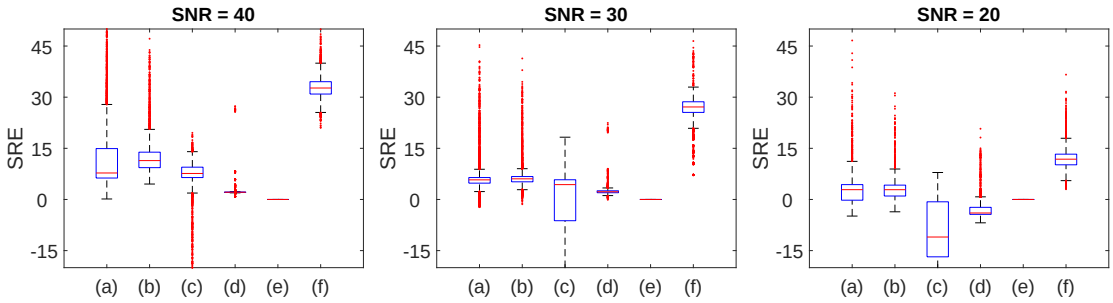


Figure 4.5: Individual SRE box-plots for three different noise levels for (a) IRLS, (b) SUnSAL-FC, (c) CDA-ME, (d) rLMM, (e) BCD/ADMM, (f) IRRF.

mixing results. Hence we implemented the analysis on four different noise levels SNR $\in \{10, 20, 30, 40\}$ and three endmember densities N $\in \{3, 6, 9\}$.

We used the endmembers from the USGS digital spectral library for the simulation. The library contains laboratory measurements of hundreds of different substances, covering the spectral interval of $0.4 - 2.5\mu$m over 224 contiguous bands. Fig.2.10 displays the materials used in the experiments.

The LMM in (1.1) was used to generate a hyperspectral image of size 50×50. To replicate the noise patterns typically found in real hyperspectral images [111] Gaussian noise was added such that the degree of noise fluctuates over the spectral range. Then the Dirichlet distribution was used for the uniform generation of the abundance vectors $\boldsymbol{x}$ [97, 98].

The parameters λ, λ_{TV} and p in the IRRF method needed to be fine-tuned to get the best predictive performance. Cross-validation (CV) is a popular model validation technique that uses the mean squared error (MSE) to assess the fit of the model. Notwithstanding, MSE uses squared errors and thus bestows inflated weight on large outliers. Therefore it is not appropriate for evaluating robust models that lower or omit the influence of outliers. Here we used a robust loss function (RLF) determined by: RLF $= 1 - e^{-|\boldsymbol{y}-\hat{\boldsymbol{y}}|}$. Here $\boldsymbol{y}$ denotes the hyperspectral image data and $\hat{\boldsymbol{y}}$ the reconstruction of the image calculated using the estimated fractional abundances $\boldsymbol{x}$ and the endmember library $\boldsymbol{M}$. We used five-fold RLF-CV, where $\log_{10}(\lambda)$ and $\log_{10}(\lambda_{TV})$ encompassed the values from -1 to -5. The parameter p covered the range 0.1 to 1 with an interval of 0.1. The optimal parameters were $\lambda = \lambda_{TV} = 10^{-3}$, $p = 0.5$.

The Tables 4.2−4.4 illustrate the SNR and RMSE performance measures of all the methods for different SNR levels and endmember densities. The fully constrained least squares based SUNSAL-FC generated mostly slightly worse results than the robust CUSAL-FC. Their difference increased for low SNR. CUSAL-FC produced a few of the best results in the table. The outcome of IRLS unmixing method was consistently worse than SUNSAL-FC and CUSAL-FC. The rLMM, UsGNCM and BCD/ADMM displayed a comparatively poor unmixing performance, especially when the SNR was low. The

Table 4.2: Comparison of unmixing performance of IRRF with various SNR, N=3

Method	SNR=40				SNR=30			
	SRE	std	RMSE	std	SRE	std	RMSE	std
IRLS	14.74	0.12	6.32	0.08	13.68	0.18	7.15	0.15
SUnSAL-FC	29.43	0.18	1.13	0.02	27.13	0.31	1.39	0.05
CUSAL-FC	29.44	0.18	1.13	0.02	27.26	0.29	1.38	0.04
CDA-ME	29.84	4.45	0.92	0.54	28.79	0.65	0.94	0.10
rLMM	4.80	1.24	19.31	2.66	4.85	1.19	19.13	2.57
BCD/ADMM	27.92	0.14	1.29	0.02	26.52	0.42	1.44	0.06
UsGNCM	12.83	0.14	7.73	0.13	13.11	0.15	7.48	0.14
IRRF	**37.99**	0.39	**0.41**	0.02	**30.72**	0.46	**0.92**	0.05
Method	SNR=20				SNR=10			
	SRE	std	RMSE	std	SRE	std	RMSE	std
IRLS	9.34	0.36	11.90	0.53	8.95	0.45	11.98	0.71
SUnSAL-FC	19.94	0.69	3.39	0.30	11.62	0.73	8.94	0.77
CUSAL-FC	19.65	0.26	3.11	0.12	11.99	0.40	7.95	0.42
CDA-ME	**20.93**	0.49	**2.53**	0.18	**12.95**	0.97	**7.29**	1.00
rLMM	2.63	4.98	27.88	36.04	-14.99	16.73	1324.8	3232.8
BCD/ADMM	6.07	0.71	16.34	1.12	1.62	0.07	23.44	0.24
UsGNCM	9.17	0.24	11.99	0.37	4.90	1.35	18.62	2.20
IRRF	20.91	0.50	3.00	0.19	9.52	0.15	11.01	0.23

Table 4.3: Comparison with various SNR, N=6

Method	SNR=40				SNR=30			
	SRE	std	RMSE	std	SRE	std	RMSE	std
IRLS	7.93	0.11	7.39	0.16	6.04	0.30	9.77	0.39
SUnSAL-FC	26.84	0.52	0.96	0.06	18.42	0.67	2.57	0.20
CUSAL-FC	27.07	0.54	0.91	0.06	18.59	0.64	2.52	0.19
CDA-ME	18.34	0.64	2.53	0.19	10.13	0.67	6.50	0.55
rLMM	2.89	0.22	11.80	0.35	2.75	0.19	11.91	0.36
BCD/ADMM	15.58	0.21	4.01	0.08	9.73	0.63	6.58	0.48
UsGNCM	11.63	0.95	4.48	0.60	8.71	1.00	7.08	0.76
IRRF	**28.12**	0.52	**0.80**	0.04	**19.27**	0.38	**2.26**	0.10
Method	SNR=20				SNR=10			
	SRE	std	RMSE	std	SRE	std	RMSE	std
IRLS	3.09	0.10	13.79	0.39	3.16	0.20	12.07	0.49
SUnSAL-FC	10.41	0.51	6.13	0.24	3.91	0.28	11.20	0.47
CUSAL-FC	**11.33**	0.21	**5.63**	0.16	4.83	0.25	10.66	0.31
CDA-ME	4.53	0.21	11.67	0.44	6.07	0.51	9.45	0.70
rLMM	-0.53	9.45	79.53	214.59	-23.79	18.80	4537.2	12418
BCD/ADMM	-1.92	0.18	21.98	0.63	-3.68	0.09	22.56	0.50
UsGNCM	4.43	0.52	11.57	1.03	-2.29	0.98	26.40	4.19
IRRF	8.27	0.17	6.75	0.17	**6.14**	0.09	**7.63**	0.13

BCD/ADMM produced good results only when the SNR was high and the endmember

Table 4.4: Comparison with various SNR, N=9

Method	SNR=40				SNR=30			
	SRE	std	RMSE	std	SRE	std	RMSE	std
IRLS	2.81	0.07	8.76	0.05	1.72	0.12	9.58	0.19
SUnSAL-FC	22.95	0.43	1.22	0.07	14.01	0.41	**3.15**	0.20
CUSAL-FC	22.68	0.40	1.28	0.07	**14.05**	0.45	3.18	0.23
CDA-ME	13.99	0.41	3.09	0.16	7.10	0.29	6.25	0.29
rLMM	2.43	0.10	8.77	0.25	2.18	0.09	8.96	0.19
BCD/ADMM	10.86	0.20	4.78	0.09	6.59	0.32	6.63	0.45
UsGNCM	11.07	0.24	3.50	0.19	6.27	1.16	6.98	1.70
IRRF	**23.08**	0.34	**1.19**	0.11	13.59	0.68	3.33	0.28
Method	SNR=20				SNR=10			
	SRE	std	RMSE	std	SRE	std	RMSE	std
IRLS	0.03	0.07	12.60	0.24	-0.31	0.13	10.76	0.37
SUnSAL-FC	7.04	0.28	5.95	0.32	1.75	0.27	8.85	0.17
CUSAL-FC	**7.24**	0.28	6.13	0.35	2.72	0.18	8.16	0.23
CDA-ME	2.69	0.16	9.41	0.31	2.76	0.38	8.63	0.56
rLMM	1.79	0.24	9.27	0.70	-24.01	17.14	1349.3	2625.9
BCD/ADMM	-3.06	0.20	18.48	0.46	-5.91	0.06	20.79	0.99
UsGNCM	2.52	0.22	11.54	0.89	-1.84	2.93	22.33	8.76
IRRF	7.12	0.17	**5.88**	0.14	**3.59**	0.21	**6.64**	0.22

density low. The CDA-ME algorithm produced great unmixing results when the number of endmembers in the pixel was small, particularly for low SNR. The execution deteriorated when the endmember density increased. The IRRF algorithm had the most consistent performance over the various SNR and endmember densities. It persistently produced the best or close to the best unmixing results. The average running times of the algorithms for different SNR are displayed in Table 4.5. It can be seen that the IRRF method was the fastest out of the robust methods.

Table 4.5: IRRF average running times in 1000s of seconds

Method	N=3	N=6	N=9
IRLS	0.0017	0.0005	0.0009
SUnSAL-FC	0.0007	0.0024	0.0019
CUSAL-FC	0.0913	0.1600	0.1912
CDA-ME	0.0363	0.0615	0.0652
rLMM	0.0999	0.1507	0.1560
BCD/ADMM	0.1263	0.2314	0.2156
UsGNCM	1.6689	2.9029	3.0337
IRRF	0.0140	0.0228	0.0314

German Alpine foothills image

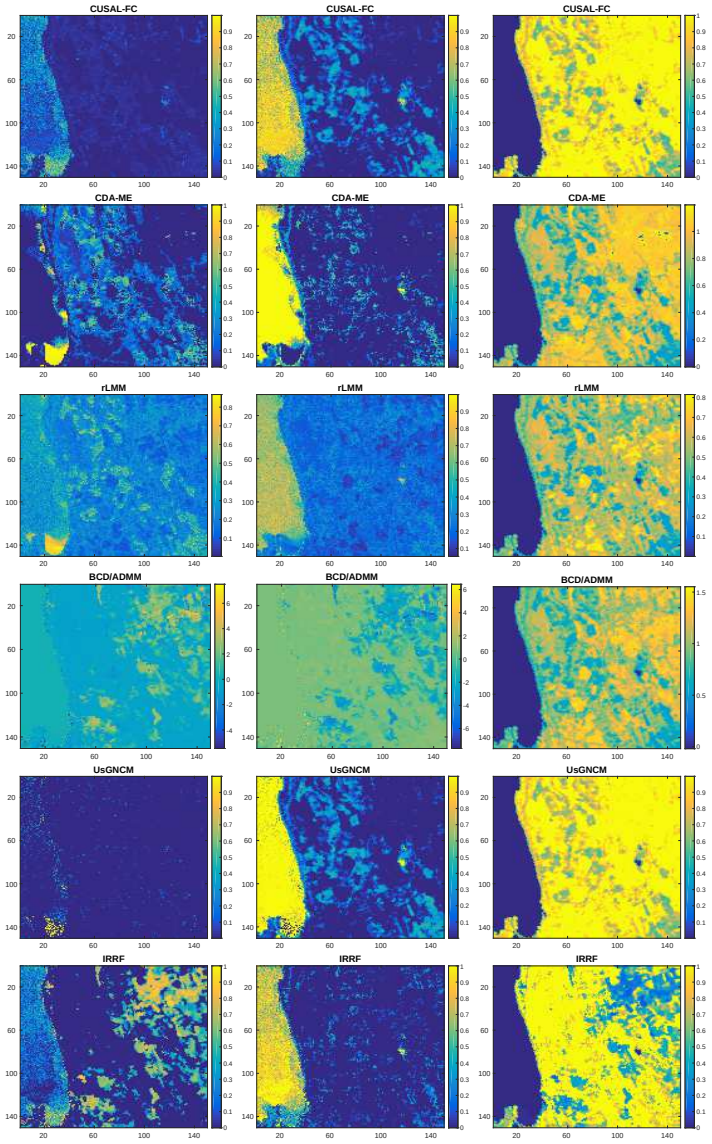
We used the Environmental Mapping and Analysis Program (EnMAP) German Alpine foothills image scene for hyperspectral unmixing. The image has 244 bands over the spectral range of 420nm−2460nm. We used a 150×150 sub-image for testing, this can be seen in Fig. 2.11. The N-FINDR algorithm was used to find the spectral signatures in the image scene. The RLF-CV was used to find the optimal parameter values; these were $\lambda = \lambda_{TV} = 10^{-1}$, $p = 0.2$ The resulting estimated fractional abundance maps of the Alpine scene for the proposed method and other robust/bilinear algorithms can be seen in Fig. 4.6.

Most methods failed to determine the regions of forest in the image. The proposed IRRF algorithm was the only one that assigned high fractional abundance values to these areas. The BCD/ADMM appeared to find the same zones as IRRF, but due to having fractional abundance values higher than 6 and below zero, it made the result difficult to interpret. The UsGNCM algorithm most precisely identified the regions of water. IRRF, CDA-ME and CUSAL-FC also determined most of the pixels that have water, but showed some inconsistency in the southern tip of the lake. The IRRF method managed to find the areas of grass effectively, whilst assigning them abundance values close to one.

Jasper Ridge

The Jasper Ridge hyperspectral image was used to test the performance of the proposed algorithm, see Fig. 2.13. The spectral range of the image is between 380nm−2500nm with a resolution of up to 9.46nm. After removing water absorption bands, the image has 198 bands. The N-FINDR algorithm was used to extract the endmember signatures from the image scene. The optimal parameter values from RLF-CV were $\lambda = 10^{-2}$, $\lambda_{TV} = 10^{-3}$, $p = 0.5$. The fractional abundance maps for tree, water, soil, and road for the proposed algorithm as well as other robust/bilinear methods are shown in Fig. 4.7.

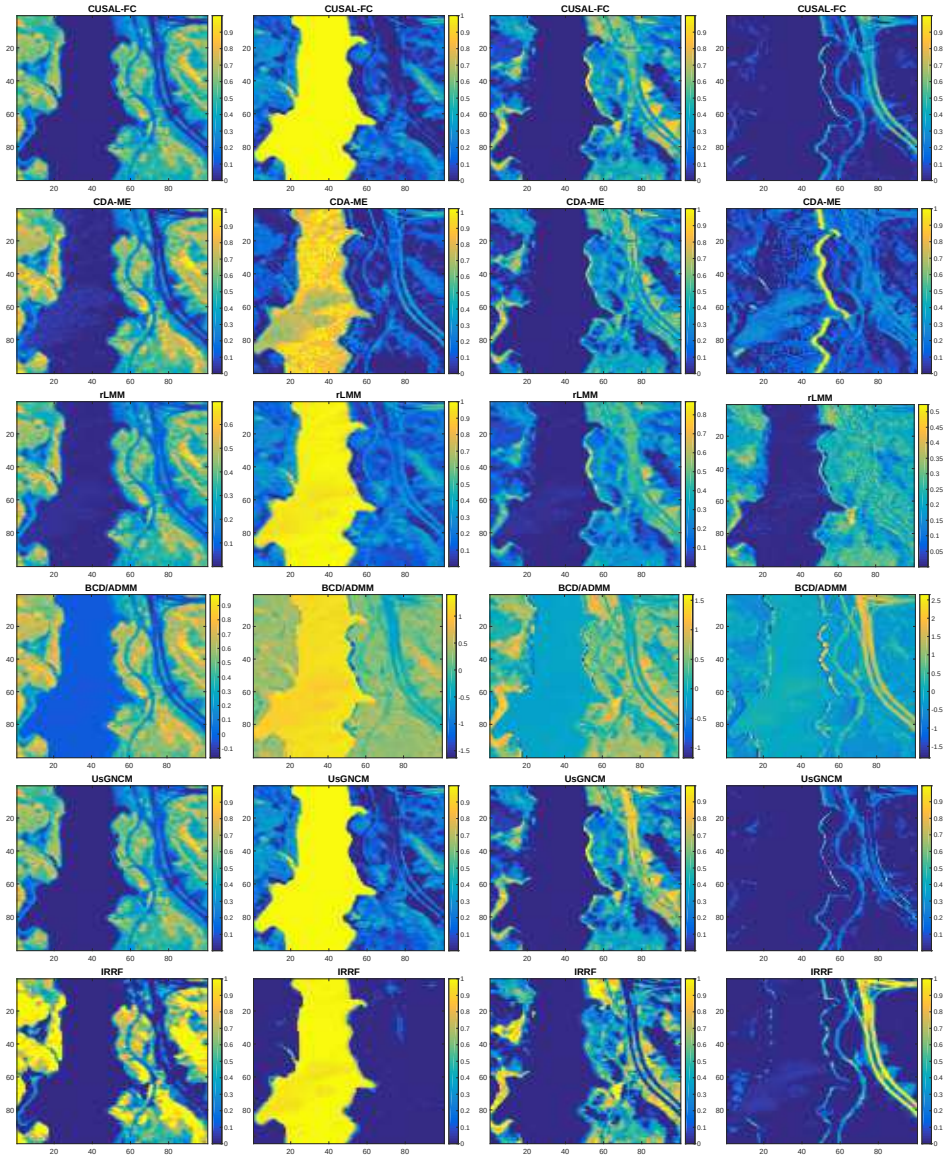
Most methods, other than CDA-ME and BCD/ADMM, found the areas of water in the Jasper Ridge image very well, and consistently assigned high fractional abundance values close or equal to one. Although, the BCD/ADMM algorithm seemed to find the correct areas for water and other endmembers, it also assigned fairly high abundance values for the rest of the pixels which made the result difficult to interpret. BCD/ADMM also assigned some fractional abundance values that are higher than 1 and below zero, which are not realistic. The CDA-ME algorithm falsely identified the shoreline as road, but appeared to find the pixels that have trees successfully. The UsGNCM method performed well in finding the water and tree zones, however was not as adequate for determining the road section. The proposed IRRF function demonstrated the best overall unmixing result compared to other methods. While the other algorithms struggled with identifying the regions of road in the image, IRRF assigned fractional abundances close to one to these areas. It also exhibited good performance in locating tree and water patches in the image.



(a) (b) (c)

Figure 4.6: Fractional abundance maps of the German Alpine foothills hyperspectral data obtained for forest in (a), water in (b), and grass in (c).

We measured the sparsity of the estimate $\boldsymbol{x}$ by the the proportion of non-zero values, i.e. density. The UsGNCM and rLMM had the density 1, which means that although they pushed some values towards zero, they did not equate them to zero exactly. Hence the fractional abundance values for the image were not easily interpretable. The density of CDA-ME was 0.18, so it is somewhat better suited for enforcing sparsity, whereas the density of IRRF was 0.08 and had the most sparse solution.



(a) (b) (c) (d)

Figure 4.7: Fractional abundance maps of the Jasper Ridge hyperspectral data obtained for tree in (a), water in (b), soil in (c), and road in (d).

4.5 Discussion

In this chapter, we extended the traditional least squares model for spectral unmixing to a robust method that can take the outliers and noise into account. We presented a new robust hyperspectral unmixing method with sparsity and spatial total variation regularization. We introduced a robust M-estimate function that can adaptively assign lower or even zero weights to noisy bands. This reduced the effect of outliers and made the model more insensitive to noise. In addition, we proposed to use the ℓ_p norm to enforce further sparsity on the fractional abundance estimates. This non-differentiable problem

was solved by an approximation of a weighted 2-norm that integrated easily into our iterative algorithmic framework. Also, we employed a spatial total variation regularizer to account for the spatial correlation in the images. Hence we presented a new improved algorithm for spectral unmixing that was simple and fast. Comparative experiments on both simulated and real-world data indicated that the proposed method gave an improved performance compared to the least squares and robust/bilinear methods.

5

Robust reparametrization

In the hyperspectral setting for the traditional least squares method, an option is to extend to the robust case and use a generalized M-lasso solution. M-estimation is a formal approach to robust estimation. It uses a fixed function $\rho(\boldsymbol{e})$ to restrain outliers; hence the traditional quadratic loss function is replaced by the robust loss function $\rho(\boldsymbol{e})$. Our experimental results on real hyperspectral image data show that noise with large amplitudes (outliers) often exists in the data, which supports the use of such methods.

In this chapter, we propose the M-estimate function, Hampel's three part e-descending function, for the unmixing problem, that either decreases the impact of or assigns zero weight to large errors and outliers [119]. The threshold parameters of the M-estimate function are calculated based on the variance of the estimation error, and used to control the magnitude of the restrictions on the outliers. Hence the ability to diminish the influence of such outliers in noisy environments can offer greater robustness.

We incorporate a sparsity inducing term via an ℓ_p norm with $0 < p < 1$ in the sparse regression problem to induce sparsity in the number of active pure spectral components. This makes the problem non-convex, but enables the method to be used on large, semi-supervised spectral libraries or in unsupervised scenarios where the number of pure spectral signatures present in a typical pixel is much smaller than the number present in the image scene.

We add a spatial total variation regularization term $|\nabla \boldsymbol{x}|$ into the sparse hyperspectral unmixing framework in order to take into account the spatial-spectral information inherent in spectral data. Instead of using the ℓ_1 norm with the ADMM method as in [37], we intend to promote more sparsity in the results via an ℓ_p norm with $0 < p < 1$. We propose the following non-convex optimization problem for the estimation of the fractional abundances $\boldsymbol{x}$

$$\arg\min_{\boldsymbol{x}} \rho\left(\boldsymbol{y} - \boldsymbol{M}\boldsymbol{x}\right) + \lambda \left\|\boldsymbol{x}\right\|_p^p + \lambda_{TV} \left\|\nabla \boldsymbol{x}\right\|_p^p \ , \tag{5.1}$$

where $\boldsymbol{y}, \boldsymbol{x}, \boldsymbol{M}$ have the corresponding dimensions $LP \times 1$, $NP \times 1$, $LP \times NP$ and define the unmixing problem over all pixels jointly. Then,

$$\begin{aligned} \boldsymbol{y} &= \left[\boldsymbol{y}_1^\top, \boldsymbol{y}_2^\top, \ldots, \boldsymbol{y}_P^\top\right]^\top , \\ \boldsymbol{x} &= \left[\boldsymbol{x}_1^\top, \boldsymbol{x}_2^\top, \ldots, \boldsymbol{x}_P^\top\right]^\top , \\ \boldsymbol{M} &= I_P \otimes \boldsymbol{M}_0 \, , \end{aligned} \tag{5.2}$$

where $\otimes$ is defined as the Kronecker product, $\boldsymbol{y}_n \in \mathbb{R}^{L \times 1}$ denotes the spectral reflectance of the nth pixel, $\boldsymbol{x}_n$ with size $N \times 1$ is the fractional abundance vector of the N endmembers in pixel n, and $\boldsymbol{M}_0 \in \mathbb{R}^{L \times N}$ is the endmember library that fashions a block-diagonal matrix $\boldsymbol{M}$ with P entries of $\boldsymbol{M}_0$ on the main diagonal and the remainder zeros.

To be physically realistic and interpretable, two constraints are imposed on the fractional abundances $\boldsymbol{x}$. The ANC and ASC are defined as $\boldsymbol{x} \geq 0, \boldsymbol{a}\boldsymbol{x}_n = 1$ for $n = 1, \ldots, P$ pixels, where $\boldsymbol{a} = [1, \ldots, 1]$ has dimension N. The ASC is sometimes executed using a renormalization heuristic as in [78] and [35, 103] or iteratively in the ADMM framework [28]. We derive and hence incorporate a well-principled reparametrization for optimization problem (5.1), which we have seen to increase the unmixing performance in our experiments. To both, perform unconstrained optimization and include the constraints [81], we reparametrize the fractional abundance vector $\boldsymbol{x}$ and estimate the $N - 1$ unconstrained parameters for each pixel, defined as

$$t_i = \log\left(\frac{x_i}{1 - \sum_{j=2}^{N} x_j}\right) \in \mathbb{R} \, , \tag{5.3}$$

where $i = 2, \ldots, N$.

We propose a new two-step algorithm to solve the robust optimization problem (5.1). First, we reparametrize the the coefficients as in Eq.(5.3). Next, we solve the optimization problem in Eq.(5.1) in an unconstrained setting via a trust region algorithm based on first and second order analytical derivatives. Trust region optimization is stable and has strong convergence properties [52,114,116], which is beneficial for our cause. The proposed algorithm ROBROC performs unconstrained $\ell_p - \ell_2$ optimization with spatial total variation regularization using the reparametrization of constraints. Qualitative hyperspectral unmixing results on simulated data and real hyperspectral image data corroborate the efficacy of the proposed method.

5.1 Mixing Model

5.1.1 Robust M-Estimates

M-estimate is defined by a minimum problem of the form $\sum \rho(\boldsymbol{e})$ =min, where ρ is an arbitrary function and $\boldsymbol{e} = \boldsymbol{y} - \boldsymbol{M}\boldsymbol{x}$. M-estimates are flexible and generalize easily to multiparameter problems [120]. They can handle outliers more effectively than the commonly used quadratic loss function $\|\boldsymbol{e}\|^2$ [119]. The M-estimate function for the likelihood term is the following function.

$$\min_{\boldsymbol{x}} \quad \rho\left(\boldsymbol{e}\right) \,, \tag{5.4}$$

where $\boldsymbol{e} = \boldsymbol{y} - \boldsymbol{M}\boldsymbol{x}$. Huber's function is frequently used for $\rho(\boldsymbol{e})$ [88]. It lessens the influence of the outliers to a certain level, but does not remove it completely. Hampel's method is better at reducing the effect of noise with large amplitudes as it can even assign them zero weight. The difference between the two robust functions and the least squares function can be seen in Fig.4.1.

The Hampel's three part re-descending function is defined as

$$\rho(\boldsymbol{e}) = \begin{cases} \frac{\boldsymbol{e}^2}{2} & 0 \leq |\boldsymbol{e}| < \xi_a, \\ \frac{\xi_a|\boldsymbol{e}| - \xi_a^2}{2} & \xi_a \leq |\boldsymbol{e}| < \xi_b, \\ \frac{\xi_a\left[\xi_b + \xi_c - \xi_a + \frac{(|\boldsymbol{e}| - \xi_c)^2}{\xi_b - \xi_c}\right]}{2} & \xi_b \leq |\boldsymbol{e}| < \xi_c, \\ \frac{\xi_a(\xi_b + \xi_c)}{2} - \frac{\xi_a^2}{2} & \xi_c \leq |\boldsymbol{e}|\,, \end{cases} \tag{5.5}$$

$$\psi(\boldsymbol{e}) = \frac{\partial \rho(\boldsymbol{e})}{\partial \boldsymbol{e}} = \begin{cases} \boldsymbol{e} & 0 \leq |\boldsymbol{e}| < \xi_a, \\ sign\left(\boldsymbol{e}\right) \cdot \xi_a & \xi_a \leq |\boldsymbol{e}| < \xi_b, \\ \frac{sign(\boldsymbol{e}) \cdot (|\boldsymbol{e}| - \xi_c)\xi_a}{\xi_b - \xi_c} & \xi_b \leq |\boldsymbol{e}| < \xi_c, \\ 0 & \xi_c \leq |\boldsymbol{e}|\,, \end{cases} \tag{5.6}$$

$$\omega(\boldsymbol{e}) = \frac{\partial \psi(\boldsymbol{e})}{\partial \boldsymbol{e}} = \frac{\partial^2 \rho(\boldsymbol{e})}{\partial \boldsymbol{e}^2} = \begin{cases} 1 & 0 \leq |\boldsymbol{e}| < \xi_a, \\ 0 & \xi_a \leq |\boldsymbol{e}| < \xi_b, \\ \frac{\xi_a}{\xi_b - \xi_c} & \xi_b \leq |\boldsymbol{e}| < \xi_c, \\ 0 & \xi_c \leq |\boldsymbol{e}|\,, \end{cases} \tag{5.7}$$

where $\psi(\boldsymbol{e}) = \partial\rho(\boldsymbol{e})/\partial\boldsymbol{e}$ is the derivative of the M-function $\rho(\boldsymbol{e})$, $\omega(\boldsymbol{e}) = \partial^2\rho(\boldsymbol{e})/\partial\boldsymbol{e}^2$ is the second derivative of $\rho(\boldsymbol{e})$, and ξ_a, ξ_b, ξ_c are threshold parameters.

In M-estimation the degree of outlier suppression is regulated by threshold parameters. The parameters are estimated based on the variance of impulse-free estimation error σ^2.

These can be estimated as $\xi_a = 1.96 \cdot \hat{\sigma}$, $\xi_b = 2.24 \cdot \hat{\sigma}$, $\xi_c = 2.58 \cdot \hat{\sigma}$, and $\hat{\sigma}$ can be estimated using the robust noise variance estimator $\hat{\sigma} = median\left(\left|\boldsymbol{y}_n - \boldsymbol{y}_{n-1}\right|\right) / \left(\sqrt{2} \cdot 0.6745\right)$, where $n = 2, ..., P$ [119, 121].

In order to find the gradient and the Hessian of the robust function $\rho(\boldsymbol{e})$, we need to use the chain rule to differentiate with respect to $\boldsymbol{x}$.

$$\frac{\partial f}{\partial \boldsymbol{x}} = \left(\frac{\partial \boldsymbol{e}}{\partial \boldsymbol{x}}\right)^{\top} \frac{\partial f}{\partial \boldsymbol{e}} \,. \tag{5.8}$$

Then we obtain

$$\begin{aligned} \frac{\partial \boldsymbol{e}}{\partial \boldsymbol{x}} &= \frac{\partial}{\partial \boldsymbol{x}} (\boldsymbol{y} - \boldsymbol{M}\boldsymbol{x}) = -\boldsymbol{M} \,, \\ \frac{\partial f}{\partial \boldsymbol{e}} &= \psi \,, \\ \frac{\partial f}{\partial \boldsymbol{x}} &= -\boldsymbol{M}^{\top}\psi \,. \end{aligned} \tag{5.9}$$

Similarly for the Hessian we need to find the second derivative with respect to $\boldsymbol{x}$.

$$\frac{\partial^2 f}{\partial \boldsymbol{x}^2} = \left(\frac{\partial \boldsymbol{e}}{\partial \boldsymbol{x}}\right)^{\top} \frac{\partial^2 f}{\partial \boldsymbol{e}^2} \frac{\partial \boldsymbol{e}}{\partial \boldsymbol{x}} + \left(\frac{\partial^2 \boldsymbol{e}}{\partial \boldsymbol{x}^2}\right)^{\top} \frac{\partial f}{\partial \boldsymbol{e}} \,, \tag{5.10}$$

$$\begin{aligned} \frac{\partial^2 f}{\partial \boldsymbol{e}^2} &= \frac{\partial \psi}{\partial \boldsymbol{e}} = \omega \,, \\ \frac{\partial^2 \boldsymbol{e}}{\partial \boldsymbol{x}^2} &= \frac{\partial}{\partial \boldsymbol{x}} (-\boldsymbol{M}) = 0 \,, \\ \frac{\partial^2 f}{\partial \boldsymbol{x}^2} &= \boldsymbol{M}^{\top}\omega\boldsymbol{M} \,. \end{aligned} \tag{5.11}$$

5.1.2 Sparsity Regularizer

The constrained, sparse $\ell_p - \ell_2$ hyperspectral unmixing task, with $0 < p < 1$, is defined as the optimization problem

$$\min_{\boldsymbol{x}} \quad \rho\left(\boldsymbol{y} - \boldsymbol{M}\boldsymbol{x}\right) + \lambda \left\|\boldsymbol{x}\right\|_p^p, \tag{5.12}$$

where λ is the Lagrange multiplier that regulates the sparsity of the solution and $\boldsymbol{y}, \boldsymbol{x}, \boldsymbol{M}$ are defined as in (5.2). It is possible to rewrite the Lasso objective as a weighted ridge regression, thus (5.12) becomes

$$\begin{aligned} \min_{\boldsymbol{x}} \quad & \rho\left(\boldsymbol{y} - \boldsymbol{M}\boldsymbol{x}\right) + \lambda \left\|\boldsymbol{D}\boldsymbol{x}\right\|_2^2 \,, \\ \text{with} \quad & \boldsymbol{D} = \operatorname{diag}\left(\boldsymbol{d}^{\frac{1}{2}}\right), \; \boldsymbol{d} = \left(\left(\boldsymbol{x}^{(\alpha-1)}\right)^2 + \boldsymbol{\epsilon}^2\right)^{\frac{p}{2}-1} \,, \end{aligned} \tag{5.13}$$

where the weights $\boldsymbol{d}$ are calculated using the $\boldsymbol{x}^{(\alpha-1)}$ results from the previous iteration $\alpha-1$. Hence the weighted ℓ_2 norm in (5.13) is a first-order approximation to the ℓ_p norm in (5.12). Here, $\boldsymbol{D}$ is a positive diagonal matrix, $\boldsymbol{\epsilon}$ measures the error in the solution for the pixels in the whole image, and goes to zero as the IRLS algorithm converges. $\boldsymbol{\epsilon}_n$ for the nth pixel is a vector of size $N \times 1$, $\boldsymbol{\epsilon}$ covers the whole image s.t. $\left[\boldsymbol{\epsilon}_1^\top, \boldsymbol{\epsilon}_2^\top, \ldots, \boldsymbol{\epsilon}_P^\top\right]^\top$, size $NP \times 1$. When the algorithm converges then $\boldsymbol{\epsilon} \to 0$ and $\boldsymbol{x}^{(\alpha-1)} \cong \boldsymbol{x}^{(\alpha)}$. It follows that $\|\boldsymbol{D}\boldsymbol{x}\|_2^2 = \|\boldsymbol{x}\|_p^p$.

The algorithm proceeds by initializing the weights with ones and then iterating between: (i) solving the quadratic problem in problem (5.13) and (ii) updating the weight matrix $\boldsymbol{D}$. This is a fixed-point iteration for solving the optimization problem (5.12).The algorithm enjoys two very attractive properties: simplicity and flexibility [116]. The next section shows that it is easy to accommodate additional p-norm penalty terms, such as a total-variation regularizer, in the objective function.

5.1.3 Spatial Total Variation Regularizer

We propose to solve the sparse, total variation (TV) regularization problem

$$\min_{\boldsymbol{x}} \quad \rho\left(\boldsymbol{y} - \boldsymbol{M}\boldsymbol{x}\right) + \lambda \left\|\boldsymbol{x}\right\|_p^p + \lambda_{TV} \left\|\nabla \boldsymbol{x}\right\|_p^p \ , \tag{5.14}$$

where $\boldsymbol{y}, \boldsymbol{x}, \boldsymbol{M}$ are defined as in (5.2). The TV term is equivalent to placing a hyper-Laplacian prior on the horizontal and vertical increments of the abundances. The parameter λ_{TV} balances the influence of this prior against the sparsity and likelihood terms; larger values will result in smoother solutions. The differencing operator ∇ computes the spatial increments of the fractional abundances, i.e. differences of abundances over neighboring pixels. It can be decomposed into horizontal ∇_h and vertical ∇_v differences so that $\nabla \equiv [\nabla_h^\top, \nabla_v^\top]^\top$. The differences between horizontal neighbors are $\nabla_h \mathbf{x} = [\mathbf{d_1}, \mathbf{d_2}, ..., \mathbf{d_m}]^\top$, where $\mathbf{d_n} = \mathbf{x_n} - \mathbf{x_{n_h}}$, with n and n_h indicating a pixel and its horizontal neighbor. ∇_v performs similarly for vertical neighbors. The difference operator ∇ has one 1 and -1 in each row for the respective vertical or horizontal endmembers. The rest of the entries are zeros. The distance between the 1 and -1 in the rows of ∇_h is the number of endmembers N. The difference is taken in between $x_n[i]$ and $x_{n+1}[i]$ for endmember i and pixels n and $n+1$. For ∇_v the distance is cN, where c indicates the number of pixels in each row of the image. In order to solve the optimization problem in (5.14) the ℓ_p norm is rewritten in terms of a weighted ℓ_2 norm, to obtain

$$\begin{aligned}
\min_{\boldsymbol{x}} \quad & \rho\left(\boldsymbol{y} - \boldsymbol{M}\boldsymbol{x}\right) + \lambda \left\|\boldsymbol{D}\boldsymbol{x}\right\|_2^2 + \lambda_{TV} \left\|\boldsymbol{\Phi}\nabla \boldsymbol{x}\right\|_2^2, \\
\text{with} \quad & \boldsymbol{D} = \operatorname{diag}(\boldsymbol{d}^{\frac{1}{2}}), \quad \boldsymbol{d} = \left(\left(\boldsymbol{x}^{(\alpha-1)}\right)^2 + \boldsymbol{\epsilon}^2\right)^{\frac{p}{2}-1}, \\
& \boldsymbol{\Phi} = \operatorname{diag}(\boldsymbol{\phi}^{\frac{1}{2}}), \quad \boldsymbol{\phi} = \left(\left(\nabla \boldsymbol{x}^{(\alpha-1)}\right)^2 + \boldsymbol{\eta}^2\right)^{\frac{p}{2}-1}.
\end{aligned} \tag{5.15}$$

Here, $\boldsymbol{\Phi}$ is a positive diagonal matrix containing the weights, the constant $\boldsymbol{\eta}$ is a column vector consisting of small integer values 10^{-6}, which is included in order to avoid division by zeros, and the weights $\boldsymbol{d}$ and $\boldsymbol{\phi}$ are calculated using the $\boldsymbol{x}^{(\alpha-1)}$ results from the previous iteration $\alpha - 1$. Note that the weighted ℓ_2 norm is equivalent to the ℓ_p norm for small enough $\boldsymbol{\eta}$ and when $\boldsymbol{x}^{(\alpha-1)} = \boldsymbol{x}^{(\alpha)}$, that is $\left\|\boldsymbol{\Phi}\nabla\boldsymbol{x}\right\|_2^2 = \|\nabla\boldsymbol{x}\|_p^p$.

5.2 Reparametrization of Constraints

In order to obtain realistic optimization results we need to constrain the regularization problem in Eq. (5.14) with sum-to-one and non-negativity constraints such that $\boldsymbol{x} \geq 0$ and $\boldsymbol{a}\boldsymbol{x}_n = 1$ for $n = 1, \ldots, P$ pixels. To address this in an effective and theoretically valid way we reparametrize the model's coefficients. Another advantage of reparametrization is that it allows us to use an unconstrained optimizer such as the trust region algorithm. We propose to use the unconstrained parameters

$$t_i \;=\; \log\left(\frac{x_i}{1-\sum_{j=2}^{N} x_j}\right) \in \mathbb{R}\ (i = 2, \ldots, N). \tag{5.16}$$

Then the original (constrained) parameters are

$$x_i = \frac{e^{t_i}}{1+\sum_{j=2}^{N} e^{t_j}}, \tag{5.17}$$

where $i = 2, \ldots, N$, and $x_1 = 1 - \sum_{i=2}^{N} x_i = 1 - \sum_{i=2}^{N} \frac{e^{t_i}}{1+\sum_{j=2}^{N} e^{t_j}}$ [81]. The next sections derive the quantities needed for the trust region optimization.

5.2.1 Gradient

The robust sparse spatially regularized mixing model in Equation (5.14) can be expressed as

$$\begin{aligned} f(\boldsymbol{x}) &= \rho\left(\boldsymbol{y} - \boldsymbol{M}\boldsymbol{x}\right) + \lambda\left(\boldsymbol{D}\boldsymbol{x}\right)^\top\left(\boldsymbol{D}\boldsymbol{x}\right) + \lambda_{TV}\left(\boldsymbol{\Phi}\nabla\boldsymbol{x}\right)^\top\left(\boldsymbol{\Phi}\nabla\boldsymbol{x}\right) \\ &= \rho\left(\boldsymbol{y} - \boldsymbol{M}\boldsymbol{x}\right) + \lambda\boldsymbol{x}^\top\boldsymbol{D}^\top\boldsymbol{D}\boldsymbol{x} + \lambda_{TV}\boldsymbol{x}^\top\nabla^\top\boldsymbol{\Phi}^\top\boldsymbol{\Phi}\nabla\boldsymbol{x}\,, \end{aligned} \tag{5.18}$$

which is maximized with respect to the $N-1$ unconstrained parameters $\mathbf{t} = [t_2, \cdots, t_N]^\top$. The derivation of the gradient using the chain rule similar to Eq.(5.8) is described below. First, the partial derivative with respect to $\boldsymbol{x}$ is found. Here the weight matrices $\boldsymbol{D}$ and $\boldsymbol{\Phi}$ are kept fixed since they are computed using the previous iteration value $\boldsymbol{x}^{(\alpha-1)}$. Using

Eq.(5.9) we have

$$\begin{aligned}\frac{\partial f(\boldsymbol{x})}{\partial \boldsymbol{x}} &= -\boldsymbol{M}^\top\psi + \lambda\left(\boldsymbol{D}^\top\boldsymbol{D} + \left(\boldsymbol{D}^\top\boldsymbol{D}\right)^\top\right)\boldsymbol{x} \\ &\quad +\lambda_{TV}\left(\nabla^\top\boldsymbol{\Phi}^\top\boldsymbol{\Phi}\nabla + \left(\nabla^\top\boldsymbol{\Phi}^\top\boldsymbol{\Phi}\nabla\right)^\top\right)\boldsymbol{x} \\ &= -\boldsymbol{M}^\top\psi + 2\lambda\boldsymbol{D}^\top\boldsymbol{D}\boldsymbol{x} + 2\lambda_{TV}\nabla^\top\boldsymbol{\Phi}^\top\boldsymbol{\Phi}\nabla\boldsymbol{x}\,. \end{aligned} \tag{5.19}$$

The partial derivatives with respect to t_i for $i = 2, \ldots, N$ derived as in (3.13)-(3.15). When $k = 2, \ldots, N$ and $k \neq i$ we have that

$$\frac{\partial x_k}{\partial t_i} = -\frac{e^{t_k+t_i}}{\left(1+\sum_{j=2}^N e^{t_j}\right)^2}. \tag{5.20}$$

When $k = i$ we have

$$\frac{\partial x_k}{\partial t_i} = \frac{e^{t_k}}{1+\sum_{j=2}^N e^{t_j}} - \frac{e^{t_k+t_i}}{\left(1+\sum_{j=2}^N e^{t_j}\right)^2}. \tag{5.21}$$

For $k = 1$ the partial derivatives are

$$\frac{\partial x_1}{\partial t_i} = -\frac{e^{t_i}}{\left(1+\sum_{j=2}^N e^{t_j}\right)^2}. \tag{5.22}$$

Hence we have the gradient of the function, which is the following

$$\frac{\partial f(\boldsymbol{x})}{\partial \mathbf{t}} = \left(\frac{\partial x_k}{\partial t_i}\right)^\top\left(-\boldsymbol{M}^\top\psi + 2\lambda\boldsymbol{D}^\top\boldsymbol{D}\boldsymbol{x} + 2\lambda_{TV}\nabla^\top\boldsymbol{\Phi}^\top\boldsymbol{\Phi}\nabla\boldsymbol{x}\right), \tag{5.23}$$

where

$$\frac{\partial x_k}{\partial t_i} = \begin{cases} -\frac{e^{t_i}}{\left(1+\sum_{j=2}^N e^{t_j}\right)^2} & \text{, for } k = 1 \\ -\frac{e^{t_i+t_k}}{\left(1+\sum_{j=2}^N e^{t_j}\right)^2} & \text{, for } k = 2, \ldots, N \;\&\; k \neq i \\ -\frac{e^{t_i+t_k}}{\left(1+\sum_{j=2}^N e^{t_j}\right)^2} + \frac{e^{t_i}}{1+\sum_{j=2}^N e^{t_j}} & \text{, for } k = 2, \ldots, N \;\&\; k = i\,. \end{cases} \tag{5.24}$$

This can be also expressed as:

$$\begin{aligned}\frac{\partial f(\boldsymbol{x})}{\partial \mathbf{t}} &= \begin{bmatrix} -\frac{e^{t_i}}{\left(1+\sum_{j=2}^N e^{t_j}\right)^2} \\ -\frac{e^{t_i+t_k}}{\left(1+\sum_{j=2}^N e^{t_j}\right)^2} + \operatorname{diag}\left(\frac{e^{t_i}}{1+\sum_{j=2}^N e^{t_j}}\right) \end{bmatrix}^\top \Big(-\boldsymbol{M}^\top\psi + 2\lambda\boldsymbol{D}^\top\boldsymbol{D}\boldsymbol{x} \\ &\quad + 2\lambda_{TV}\nabla^\top\boldsymbol{\Phi}^\top\boldsymbol{\Phi}\nabla\boldsymbol{x}\Big) \end{aligned} \tag{5.25}$$

The partial derivatives $\partial \boldsymbol{x}/\partial \mathbf{t}$ are of dimension $N \times (N-1)$,$\partial f(\boldsymbol{x})/\partial \mathbf{x}$ of dimension $N \times 1$, and $\partial f(\boldsymbol{x})/\partial \mathbf{t}$ of dimension $(N-1) \times 1$; they demand the estimation of $N-1$ gradients.

5.2.2 Initial steps for the Hessian

We used the gradient in Eq. (5.23)-(5.24) and the chain rule in Eq.(3.19)-(3.20) to derive the second partial derivatives of $f(\boldsymbol{x})$ with respect to t_i, where $i = 2, \ldots, N$. Using the results from Eq.(5.11) and Eq. (5.19) the second order partial derivatives with respect to $\boldsymbol{x}$ with fixed weights $\boldsymbol{D}$ and $\boldsymbol{\Phi}$ are

$$\frac{\partial^2 f(\boldsymbol{x})}{\partial \boldsymbol{x}^2} = \boldsymbol{M}^\top \omega \boldsymbol{M} + \lambda \boldsymbol{D}^\top \boldsymbol{D} + \lambda_{TV} \nabla^\top \boldsymbol{\Phi}^\top \boldsymbol{\Phi} \nabla \, . \tag{5.26}$$

The second order derivatives with respect to t_i follow the derivation shown in Eq. (3.22)-(3.24). For $k = 1$ the derivative is

$$\frac{\partial^2 x_1}{\partial t_i^2} = \frac{2e^{2t_i}}{\left(1 + \sum_{j=2}^{N} e^{t_j}\right)^3} - \frac{e^{t_i}}{\left(1 + \sum_{j=2}^{N} e^{t_j}\right)^2} \, . \tag{5.27}$$

When $k = 2, \ldots, N$ and $k \neq i$, we have

$$\frac{\partial^2 x_k}{\partial t_i^2} = \frac{2e^{t_k + 2t_i}}{\left(1 + \sum_{j=2}^{N} e^{t_j}\right)^3} - \frac{e^{t_k + t_i}}{\left(1 + \sum_{j=2}^{N} e^{t_j}\right)^2} \, . \tag{5.28}$$

For $k = i$, when $k = 2, \ldots, N$

$$\frac{\partial^2 x_k}{\partial t_i^2} = \frac{2e^{t_k + 2t_i}}{\left(1 + \sum_{j=2}^{N} e^{t_j}\right)^3} - \frac{2e^{t_k + t_i} + 2e^{t_i}}{\left(1 + \sum_{j=2}^{N} e^{t_j}\right)^2} + \frac{e^{t_i}}{1 + \sum_{j=2}^{N} e^{t_j}} \, . \tag{5.29}$$

Accordingly,

$$\frac{\partial^2 x_k}{\partial t_i^2} = \begin{cases} \frac{2e^{2t_i}}{\left(1+\sum_{j=2}^{N} e^{t_j}\right)^3} - \frac{e^{t_i}}{\left(1+\sum_{j=2}^{N} e^{t_j}\right)^2} \, , \text{for } k = 1 \\ \frac{2e^{2t_i + t_k}}{\left(1+\sum_{j=2}^{N} e^{t_j}\right)^3} - \frac{e^{t_i + t_k}}{\left(1+\sum_{j=2}^{N} e^{t_j}\right)^2} \, , \text{for } k = 2, \ldots, N \;\&\; k \neq i \\ \frac{2e^{2t_i + t_k}}{\left(1+\sum_{j=2}^{N} e^{t_j}\right)^3} - \frac{2e^{t_i + t_k} + 2e^{2t_i}}{\left(1+\sum_{j=2}^{N} e^{t_j}\right)^2} + \frac{e^{t_i}}{1+\sum_{j=2}^{N} e^{t_j}} \, , \text{for } k = 2, \ldots, N \;\&\; k = i \, . \end{cases} \tag{5.30}$$

The mixed second order derivatives with $s \neq i$ werre derived as in Eq.(3.26)-(3.28). When $k = 1$

$$\frac{\partial^2 x_1}{\partial t_i \partial t_s} = \frac{2e^{t_i + t_s}}{\left(1 + \sum_{j=2}^{N} e^{t_j}\right)^3} \, . \tag{5.31}$$

When $k = 2, \ldots, N$, $k \neq i$ & $k \neq s$,

$$\frac{\partial^2 x_k}{\partial t_i \partial t_s} = \frac{2e^{t_k+t_i+t_s}}{\left(1+\sum_{j=2}^{N} e^{t_j}\right)^3} \,. \tag{5.32}$$

When $k = i$ or $k = s$, such that $k = 2, \ldots, N$,

$$\frac{\partial^2 x_k}{\partial t_i \partial t_s} = \frac{2e^{t_k+t_i+t_s}}{\left(1+\sum_{j=2}^{N} e^{t_j}\right)^3} - \frac{e^{t_i+t_s}}{\left(1+\sum_{j=2}^{N} e^{t_j}\right)^2} \,. \tag{5.33}$$

Ergo, for $i \neq s$ the equations for the mixed second derivatives are

$$\frac{\partial^2 x_k}{\partial t_i \partial t_s} = \begin{cases} \frac{2e^{t_i+t_s}}{\left(1+\sum_{j=2}^{N} e^{t_j}\right)^3} & \text{, for } k = 1 \\ \frac{2e^{t_k+t_i+t_s}}{\left(1+\sum_{j=2}^{N} e^{t_j}\right)^3} & \text{, for } k = 2, \ldots, N \;\&\; k \neq i, k \neq s \\ \frac{2e^{t_k+t_i+t_s}}{\left(1+\sum_{j=2}^{N} e^{t_j}\right)^3} - \frac{e^{t_i+t_s}}{\left(1+\sum_{j=2}^{N} e^{t_j}\right)^2} & \text{, for } k = 2, \ldots, N \;\&\; k = i \textit{ or } k = s \,. \end{cases} \tag{5.34}$$

5.2.3 The Hessian

These components that form the Hessian matrix of size $(N-1) \times (N-1)$ are as follows.

$$\begin{aligned} \frac{\partial^2 f(\boldsymbol{x})}{\partial \mathbf{t}_i^2} &= \left(\frac{\partial \boldsymbol{x}}{\partial \mathbf{t}_i}\right)^\top \left(\boldsymbol{M}^\top \omega \boldsymbol{M} + \lambda \boldsymbol{D}^\top \boldsymbol{D} + \lambda_{TV} \nabla^\top \boldsymbol{\Phi}^\top \boldsymbol{\Phi} \nabla\right) \frac{\partial \boldsymbol{x}}{\partial \mathbf{t}_i} \\ &\quad + \left(\frac{\partial^2 \boldsymbol{x}}{\partial \mathbf{t}_i^2}\right)^\top \left(-\boldsymbol{M}^\top \psi + 2\lambda \boldsymbol{D}^\top \boldsymbol{D}\boldsymbol{x} + \lambda_{TV} \nabla^\top \boldsymbol{\Phi}^\top \boldsymbol{\Phi} \nabla \boldsymbol{x}\right) \,. \end{aligned} \tag{5.35}$$

$$\begin{aligned} \frac{\partial^2 f(\boldsymbol{x})}{\partial \mathbf{t}_i \partial \mathbf{t}_s} &= \left(\frac{\partial \boldsymbol{x}}{\partial \mathbf{t}_i}\right)^\top \left(\boldsymbol{M}^\top \omega \boldsymbol{M} + \lambda \boldsymbol{D}^\top \boldsymbol{D} + \lambda_{TV} \nabla^\top \boldsymbol{\Phi}^\top \boldsymbol{\Phi} \nabla\right) \frac{\partial \boldsymbol{x}}{\partial \mathbf{t}_s} \\ &\quad + \left(\frac{\partial^2 \boldsymbol{x}}{\partial \mathbf{t}_i \partial \mathbf{t}_s}\right)^\top \left(-\boldsymbol{M}^\top \psi + 2\lambda \boldsymbol{D}^\top \boldsymbol{D}\boldsymbol{x} + 2\lambda_{TV} \nabla^\top \boldsymbol{\Phi}^\top \boldsymbol{\Phi} \nabla \boldsymbol{x}\right) \,, \end{aligned} \tag{5.36}$$

where $s \neq i$, ψ and ω are defined as in Equation (5.6)-(5.7) and

$$\frac{\partial x_k}{\partial t_i} = \begin{cases} -\frac{e^{t_i}}{\left(1+\sum_{j=2}^{N} e^{t_j}\right)^2} & \text{, for } k = 1 \\ -\frac{e^{t_i+t_k}}{\left(1+\sum_{j=2}^{N} e^{t_j}\right)^2} & \text{, for } k = 2, \ldots, N \;\&\; k \neq i \\ -\frac{e^{t_i+t_k}}{\left(1+\sum_{j=2}^{N} e^{t_j}\right)^2} + \frac{e^{t_i}}{1+\sum_{j=2}^{N} e^{t_j}} & \text{, for } k = 2, \ldots, N \;\&\; k = i \end{cases} \tag{5.37}$$

$$\frac{\partial^2 x_k}{\partial t_i^2} = \begin{cases} \frac{2e^{2t_i}}{\left(1+\sum_{j=2}^{N} e^{t_j}\right)^3} - \frac{e^{t_i}}{\left(1+\sum_{j=2}^{N} e^{t_j}\right)^2} & , \text{for } k = 1 \\ \frac{2e^{2t_i+t_k}}{\left(1+\sum_{j=2}^{N} e^{t_j}\right)^3} - \frac{e^{t_i+t_k}}{\left(1+\sum_{j=2}^{N} e^{t_j}\right)^2} & , \text{for } k = 2,\ldots,N \;\&\; k \neq i \\ \frac{2e^{2t_i+t_k}}{\left(1+\sum_{j=2}^{N} e^{t_j}\right)^3} - \frac{e^{t_i+t_k}+2e^{2t_i}}{\left(1+\sum_{j=2}^{N} e^{t_j}\right)^2} + \frac{e^{t_i}}{1+\sum_{j=2}^{N} e^{t_j}} & , \text{for } k = 2,\ldots,N \;\&\; k = i \end{cases} \tag{5.38}$$

$$\frac{\partial^2 x_k}{\partial t_i \partial t_s} = \begin{cases} \frac{2e^{t_i+t_s}}{\left(1+\sum_{j=2}^{N} e^{t_j}\right)^3} & , \text{for } k = 1 \\ \frac{2e^{t_k+t_i+t_s}}{\left(1+\sum_{j=2}^{N} e^{t_j}\right)^3} & , \text{for } k = 2,\ldots,N \;\&\; k \neq i, k \neq s \\ \frac{2e^{t_k+t_i+t_s}}{\left(1+\sum_{j=2}^{N} e^{t_j}\right)^3} - \frac{e^{t_i+t_s}}{\left(1+\sum_{j=2}^{N} e^{t_j}\right)^2} & , \text{for } k = 2,\ldots,N \;\&\; k = i \textit{ or } k = s\,. \end{cases} \tag{5.39}$$

The partial derivatives $\partial \boldsymbol{x}/\partial \mathbf{t}$ are of dimension $N \times (N-1)$, $\partial f(\boldsymbol{x})/\partial \mathbf{x}$ of dimension $N \times 1$; $\partial f(\boldsymbol{x})/\partial \mathbf{t}$ of dimension $(N-1) \times 1$, $\partial^2 \boldsymbol{x}/\partial \mathbf{t}_i^2$ of dimension $N \times 1$, and $\partial^2 \boldsymbol{x}/\partial \mathbf{t}_i \partial \mathbf{t}_s$ of dimension $N \times 1$. The second order partial derivatives $\partial^2 f(\boldsymbol{x})/\partial \mathbf{t}^2$ and $\partial^2 f(\boldsymbol{x})/\partial \mathbf{t}_i \partial \mathbf{t}_s$ form the Hessian matrix of size $(N-1) \times (N-1)$.

5.3 Unmixing algorithm

5.3.1 Trust region optimization

The optimization of the full function $f(\boldsymbol{x})$ can be challenging and/or time consuming. Trust region optimization uses an approximation of $f(\boldsymbol{x})$ that closely mirrors the function f in a fixed region Δ (radius of the trust region) close to a point $\boldsymbol{x}$. The simpler function m makes the calculations more easily tractable. To find step size b for each iteration we minimize the approximate function m in the trust region [52],

$$\min_b m(b) \quad \text{s.t. } b \in \Delta. \tag{5.40}$$

The current estimate is updated to $\boldsymbol{x} + b$ if the trial step b has a lower function value. Alternatively the trust region radius Δ is changed, and the procedure is rerun.

The first and second order Taylor terms are frequently used for approximating the function f [114, 117]. An extension of the standard method is to broaden the trial step search to the span of g and $B^{-1}g$, which makes it more refined. Consequently we have an accurate and efficient method [51].

$$\begin{aligned} \min_b m(b) &= f + g^\top b + \frac{1}{2} b^\top B b \\ \text{s.t. } \|b\| &\leq \Delta,\ b \in \text{span}[g, B^{-1}g]. \end{aligned} \tag{5.41}$$

Trust region algorithm has many advantages, that includes being reliable, robust and having very strong convergence properties [52]. The iterative method of trust region optimization can be seen in Algorithm 3 on page 61.

5.3.2 Numerical algorithm of ROBROC

The robust unmixing algorithm including reparametrization of constraints along with sparsity and total variation regularizers is optimized by the unconstrained trust region algorithm. The pseudocode of the ROBROC algorithm can be seen in Alg.6. To begin with, the algorithm initializes the constrained parameters by assigning them equal fractional abundance values according to the sum-to-one constraint (step 1). These are then reparametrized to get the working parameters $\mathbf{t}$. In step 3, within the main loop, the unconstrained working parameters are converted back to the constrained ones for calculating the weights for the regularization terms. By definition the weights $\boldsymbol{D}$ and $\boldsymbol{\Phi}$ are positive diagonal matrices with the values $\left(\left(\boldsymbol{x}^{(\alpha-1)}\right)^2+\boldsymbol{\epsilon}^2\right)^{(p/2-1)/2}$ and $\left(\left(\nabla\boldsymbol{x}^{(\alpha-1)}\right)^2+\boldsymbol{\eta}^2\right)^{(p/2-1)/2}$, respectively, on the diagonal. This is fixed-point iteration using the iterate $\boldsymbol{x}^{(\alpha-1)}$ to update the weights for iteration α. When $\boldsymbol{\epsilon}=\boldsymbol{\eta}=0$ and $\boldsymbol{x}^{(\alpha-1)}=\boldsymbol{x}^{(\alpha)}$then we obtain $\|\boldsymbol{x}\|_p^p+\|\nabla\boldsymbol{x}\|_p^p=\|\boldsymbol{D}\boldsymbol{x}\|_2^2+\left\|\boldsymbol{\Phi}\nabla\boldsymbol{x}\right\|_2^2$. ∇ is an operator that enables us to calculate the horizontal and vertical differences in the fractional abundances between neighbouring pixels in the image. $\boldsymbol{\eta}$ is a column vector consisting of fixed small integers of size $\approx 10^{-6}$. It is added to absolute value of $\nabla\boldsymbol{x}$ with the purpose of avoiding infinite values in step 6.

Algorithm 6 Pseudocode of ROBROC.

Task: Solve $\rho\left(\boldsymbol{y}-\boldsymbol{M}\boldsymbol{x}\right)+\lambda\left\|\boldsymbol{x}\right\|_p^p+\lambda_{TV}\left\|\nabla\boldsymbol{x}\right\|_p^p$, using reparametrization of coefficients $\mathbf{t}=\log(\boldsymbol{x}/(1-\sum_{j=2}^{N}x_j))$. Find $\boldsymbol{x}$.

Input: $\boldsymbol{M}_0$: $L\times N$ spectral library, $\boldsymbol{Y}$: $L\times P$ hyperspectral data matrix, λ : sparsity regularization parameter, λ_{TV} : spatial regularization parameter, p : quasi-norm.

Output: $\boldsymbol{X}$: $N\times P$ fractional abundance matrix w.r.t. $\boldsymbol{M}_0$.

Initialization:

1: $\mathbf{x}\leftarrow[1,...,1]/N,\quad \mathbf{t}=\log\left(\frac{\boldsymbol{x}}{1-\sum_{j=2}^{N}x_j}\right)$.

2: $\boldsymbol{\epsilon}\leftarrow[1,...,1]$.

Iteration α:

3: $\boldsymbol{x}=\frac{e^{\mathbf{t}}}{1+\sum_{j=2}^{N}e^{t_j}}$.

4: $\boldsymbol{d}\leftarrow\left(\left(\boldsymbol{x}^{(\alpha-1)}\right)^2+\boldsymbol{\epsilon}^2\right)^{\frac{p}{2}-1}$.

5: $D=\mathrm{diag}(\boldsymbol{d}^{\frac{1}{2}})$.

6: $\boldsymbol{\phi}=\left(\left(\nabla\boldsymbol{x}^{(\alpha-1)}\right)^2+\boldsymbol{\eta}^2\right)^{\frac{p}{2}-1}$.

7: $\boldsymbol{\Phi}=\mathrm{diag}(\boldsymbol{\phi}^{\frac{1}{2}})$.

8: $\boldsymbol{x}^{(\alpha)}\leftarrow\arg\min_{\boldsymbol{x}}\sum\rho\left(\boldsymbol{y}-\boldsymbol{M}\boldsymbol{x}\right)+\lambda\left\|\boldsymbol{D}\boldsymbol{x}\right\|_2^2+\lambda_{\mathrm{TV}}\left\|\boldsymbol{\Phi}\nabla\boldsymbol{x}\right\|_2^2$

s.t. $\mathbf{t}=\log\left(\frac{\boldsymbol{x}}{1-\sum_{j=2}^{N}x_j}\right)$.

9: $\boldsymbol{\epsilon}_n\leftarrow\min\left(\boldsymbol{\epsilon}_n,\frac{\mathrm{sort}(\boldsymbol{x}_n)_{q+1}}{N}\right)$.

Post processing:

10: $\boldsymbol{x}=\frac{e^{\mathbf{t}}}{1+\sum_{j=2}^{N}e^{t_j}}$.

In step 9, the function $\text{sort}(\boldsymbol{x}_n)_{q+1}$ orders the absolute values of the current abundance estimates $\boldsymbol{x}_n$ in each pixel in a decreasing sequence and chooses the $(q+1)$th number for a predetermined integer q. The abundance vector $\boldsymbol{x}$ is *q-sparse* if and only if $r(\boldsymbol{x})_q = 0$ [34]. Therefore the information about the sparsity of $\boldsymbol{x}_n$ at each iteration is transferred into the value of $\boldsymbol{\epsilon}_n$ and the updated weights.

The optimization problem in step 8 of Alg. 6 has a quadratic objective. We can use classic programming techniques, similar to the ones represented in [51], to solve it. We use the reparametrization along with the function $\sum \rho\left(\boldsymbol{y} - \boldsymbol{M}\boldsymbol{x}\right) + \lambda \left\|\boldsymbol{D}\boldsymbol{x}\right\|_2^2 + \lambda_{TV} \left\|\boldsymbol{\Phi}\nabla\boldsymbol{x}\right\|_2^2$ to find the function value, the gradient and the Hessian of the optimization problem. The derived gradient can be seen in Equations (5.23)-(5.24) and the Hessian in Equations (5.35)-(5.39). The function value, gradient and Hessian are employed within the trust region optimization of the objective function to estimate $\mathbf{t}$ at each iteration. The iterative process continues until the algorithm converges. Finally in step 10, the unconstrained working parameters $\mathbf{t}$ are converted back to the constrained fractional abundance parameters $\boldsymbol{x}$, which is the final solution of the estimation.

The purpose of the reparametrization is to take the constraints into account and to be able to carry out the estimation on real numbers. The sparsity constraint forces the abundances $\boldsymbol{x}$ to zero. If $x_i = 0$, then the reparametrized parameter $t_i = -\infty$ in Eq. (5.16), that can create instability in the optimization. This issue is avoided by the setup of the trust region algorithm that is bounded by definition [52], which is one of the advantages of the algorithm. As the optimization is only solved within the trust region, then t_i is bounded and would never reach $-\infty$. It follows that the trust region algorithm is pushing the x_i towards zero, but never exactly equating them to zero. However, the estimated coefficients that are very close to zero can be effectively treated as zero, thus we still achieve sparse results.

With regard to the computational complexity of the ROBROC algorithm, the most costly step is the computation of the Hessian, which has the complexity of $\mathcal{O}(PN^2)$. Hence we can generalize that this is the computational complexity at each iteration. However, due to the fast convergence of the trust region algorithm, the method is efficient.

5.4 Experimental results

In this section, we examine the performance of the robust sparse spatially regularized unmixing algorithm with reparametrization of constraints (ROBROC) on both simulated and real world data. We use the simulated data experiments to evaluate how the proposed algorithm performs quantitatively compared to other state-of-the-art algorithms, and the real satellite data to show qualitative performance results on real hyperspectral data.

For the simulated data experiments, the abundances are known, hence we used both the signal to reconstruction error (SRE) and the root mean square error (RMSE) to assess to quality of reconstruction of the image. SRE is defined in decibels: SRE(dB)

$= 10\log_{10}\left(\mathbb{E}\left[\|\boldsymbol{x}\|_2^2\right]/\mathbb{E}\left[\|\boldsymbol{x}-\hat{\boldsymbol{x}}\|_2^2\right]\right)$. Here $\boldsymbol{x}$ contains the true endmember fractional abundance values, and $\hat{\boldsymbol{x}}$ the estimated fractional abundance values. A higher value for SRE indicates a better reconstruction of the image. Average root mean square error can be calculated for the image in the following way: RMSE= $\sqrt{\frac{1}{NP}\sum_{n=1}^{P}\|\boldsymbol{x}_n-\hat{\boldsymbol{x}}_n\|^2}$. For RMSE, a lower value signifies a better unmixing performance.

Comparison with other algorithms

We compared the proposed ROBROC algorithm to two linear least squares based fully constrained algorithms to see how the robust method enhances performance. The methods are: Fully Constrained Sparse Unmixing via variable Splitting and Augmented Lagrangian (SUnSAL-FC) [28], Iteratively Reweighted Least Squares (IRLS) [35]. We also compared our approach with other recent robust/bilinear methods: Fully Constrained Correntropy-based Unmixing by Variable Splitting and Augmented Lagrangian (CUSAL-FC) [97], Coordinate Decent Algorithm with Mismodeling Effects (CDA-ME) [93], Robust Linear Mixing Model (rLMM) [92], Block Coordinate Descent Alternating Direction Method of Multipliers (BCD/ADMM) [99], Unsupervised Generalized Normal Compositional Model (UsGNCM) [98].

Simulated Data

In the first simulated data experiment we used the USGS digital spectral library. It consists of the spectral reflectance of hundreds of materials measured in the lab. These spectral reflectance values have 224 bands that lie uniformly in the interval 0.4-2.5 μm. We created a 75×75 simulated image with 224 spectral bands for each pixel with five randomly selected spectral signatures from the spectral library, see Fig. 2.1. The data were generated using a linear mixing model, and the abundance's sum constraint was enforced on each pixel. The true abundance of the five endmembers in the simulated image are displayed in Fig. 2.2. The simulated data were contaminated with white noise as well as spectrally correlated noise. The signal-to-noise ratio was set to 40dB, 30dB and 20dB. We set the maximum number of iterations to 400, and the number of endmembers after the MUSIC library pruning to 20 for these experiments. For this experiment we did not display the results of CUSAL-FC and UsGNCM due to the methods taking excessively long to compute and not getting presentable results respectively.

The simulated hyperspectral image in Fig. 2.2 was used to assess the performance of different sparse unmixing algorithms. The quantitative unmixing results for different methods are displayed in Table 5.1. The Fig. 5.1 shows the abundance maps calculated using the optimal parameter values for λ, λ_{TV} and p by IRLS, SUnSAL-FC, CDA-ME, rLMM, BCD/ADMM and ROBROC, for SNR= 40. We used five-fold cross-validation to find the optimal tuning parameters. The optimal parameter values were $p = 0.5$,

$\lambda = \lambda_{TV} = 10^{-4}$ for SNR= 40 and 10^{-4} for SNR= $30, 20$.

Table 5.1: Comparison of ROBROC at different noise levels

Method	SNR=40		SNR=30		SNR=20	
	SRE	RMSE	SRE	RMSE	SRE	RMSE
IRLS	8.59	11.48	5.60	15.68	1.53	24.60
SUnSAL-FC	11.13	9.98	6.02	16.92	2.23	24.03
CDA-ME	-15.30	231.85	-13.44	185.72	-16.33	261.49
rLMM	0.05	37.53	0.05	37.53	0.04	37.53
BCD/ADMM	2.27	29.37	2.38	28.41	-3.13	37.26
ROBROC	**14.48**	**6.94**	**12.51**	**7.54**	**5.88**	**14.86**

The fractional abundance maps in Fig. 5.1 show that the CDA-ME and rLMM methods did not perform well as there was too much noise in the images. Although the least-squares methods IRLS and SUnSAL-FC displayed significant noise in the abundance maps as well, they appeared to identify the square regions with common fractional abundances in the image better. These results were supported by the quantitative performance measures in Table 5.1 with IRLS and SUnSAL-FC having higher SRE and lower RMSE values. The abundance maps of BCD/ADMM algorithm showed much less noise and appeared to allocate similar fractional abundances to neighboring pixels more consistently. However, the algorithm attained a lower SRE than IRLS and SUnSAL-FC, which indicates that it assigned skewed proportions to large areas of pixels. The results of the ROBROC algorithm showed some noise as well, but it was much less severe than for other methods. Also, it managed to correctly identify most square zones of common fractional abundance in the simulated image. This was confirmed by the best quantitative performance measures seen in Table 5.1.

The performance of the methods with increased noise can be seen in Fig. 5.2 and Fig. 5.3. The SNR was lowered to 30dB and 20dB in these images respectively. As expected, all algorithms fared worse in noisier conditions. Most algorithms failed to identify any regularity for endmember 3 when SNR is 20dB, only rLMM and ROBROC showed some identification of the square areas. Also, rLMM attained higher SRE than BCD/ADMM in this case. The abundance maps of rLMM were much noisier than ROBROC though, and the algorithm did not locate as many square regions. The ROBROC method acquired the highest SRE and lowest RMSE results for the noisier scenarios.

The SRE calculated for each method of unmixing was an average for the image, i.e. for the 75×75 image it was an average of 5625 pixels. However, to see the variance of the performance from pixel to pixel we calculated the SRE for individual pixels and acquired the standard deviation for each method. These can be seen in Fig. 5.4 together with the best SRE results for three different noise levels. The variance of the quality of unmixing results for CDA-ME, rLMM, and ROBROC increased with noise. This was especially severe for CDA-ME, where the range of values became very wide. For IRLS and

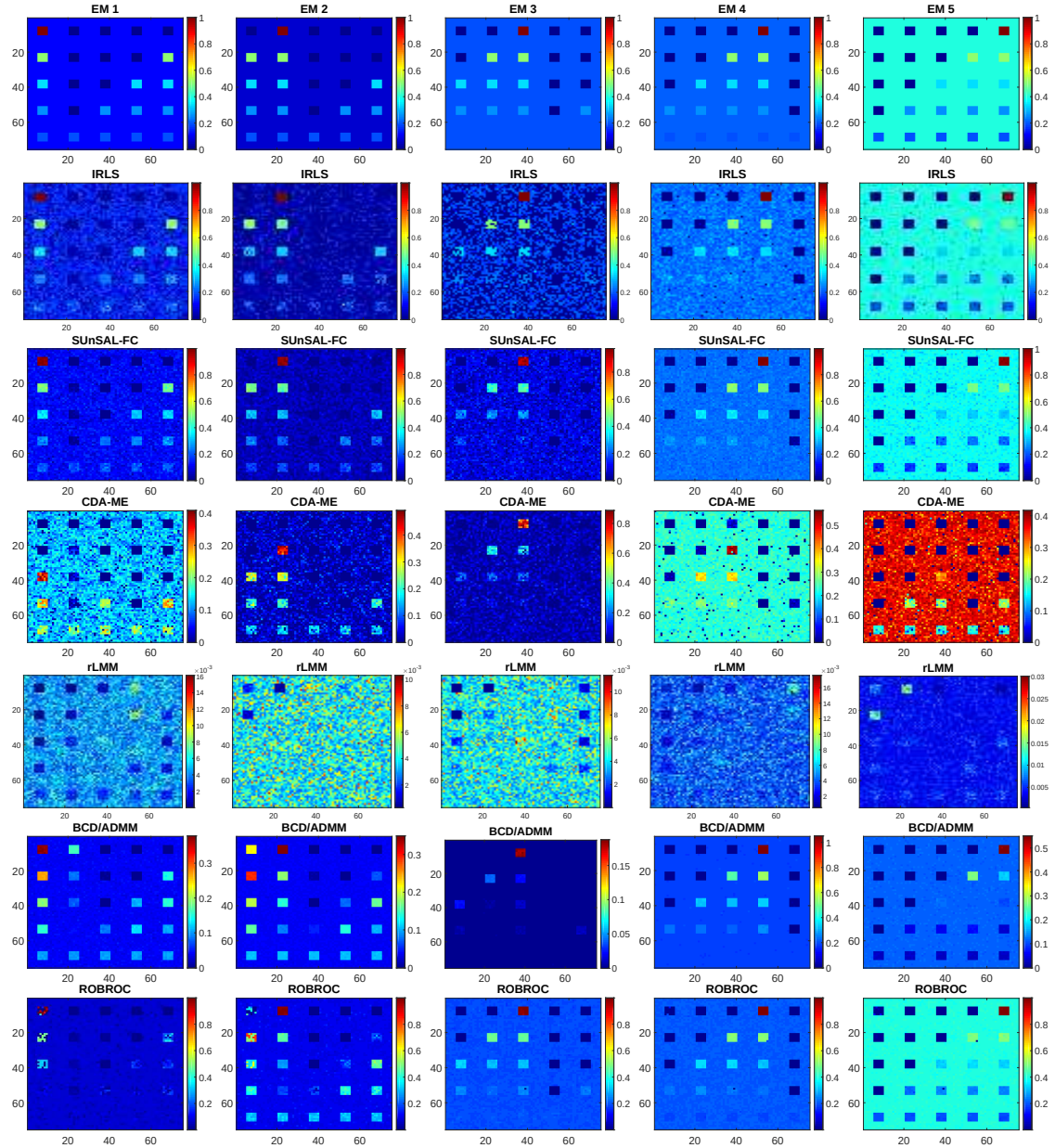


Figure 5.1: Fractional abundance maps obtained by IRLS, SUnSAL-FC, CDA-ME, rLMM, BCD/ADMM, and ROBROC for endmember (EM) 1 in (a), EM 2 in (b), EM 3 in (c), and EM 4 in (d). SNR=40dB.

SUNSAL-FC, the standard deviation for higher noise levels was lower than for SNR=40. In general, ROBROC showed the best results as the SRE values were the highest. Also, the standard deviation of values was fairly small, especially in scenarios with less noise.

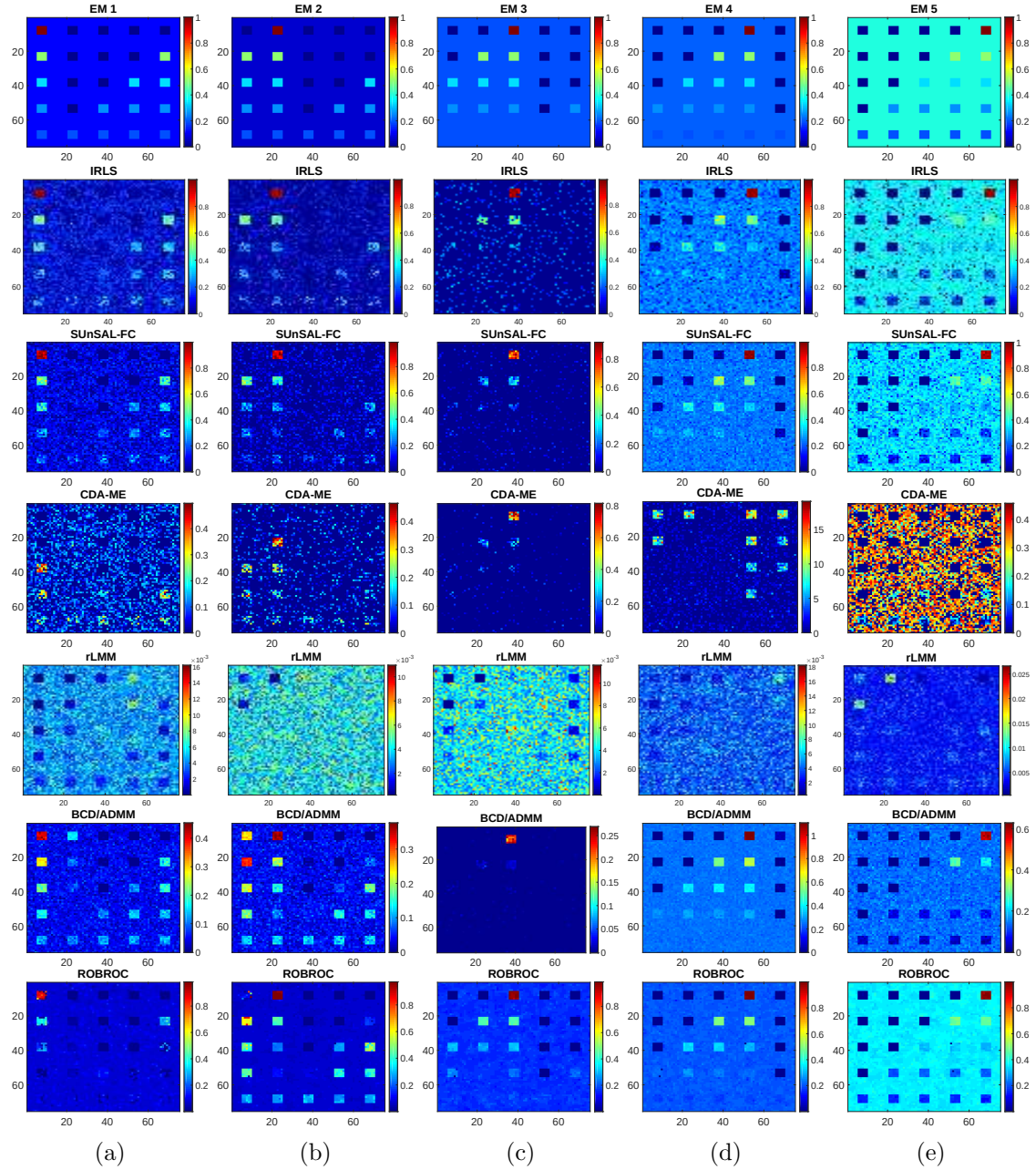


Figure 5.2: Fractional abundance maps obtained by IRLS, SUnSAL-FC, CDA-ME, rLMM, BCD/ADMM, and ROBROC for endmember (EM) 1 in (a), EM 2 in (b), EM 3 in (c), and EM 4 in (d). SNR=30dB.

Monte Carlo simulation

This section evaluates the performance of the proposed fully constrained robust unmixing method and compares it with other state-of-the-art methods. Since the signal to noise ratio (SNR) in the images generally affects the unmixing results, we performed the experiments using three different noise levels SNR $\in \{10, 20, 30\}$. The number of endmembers present in the spectrum of a pixel can also influence the outcome. Hence, we considered three

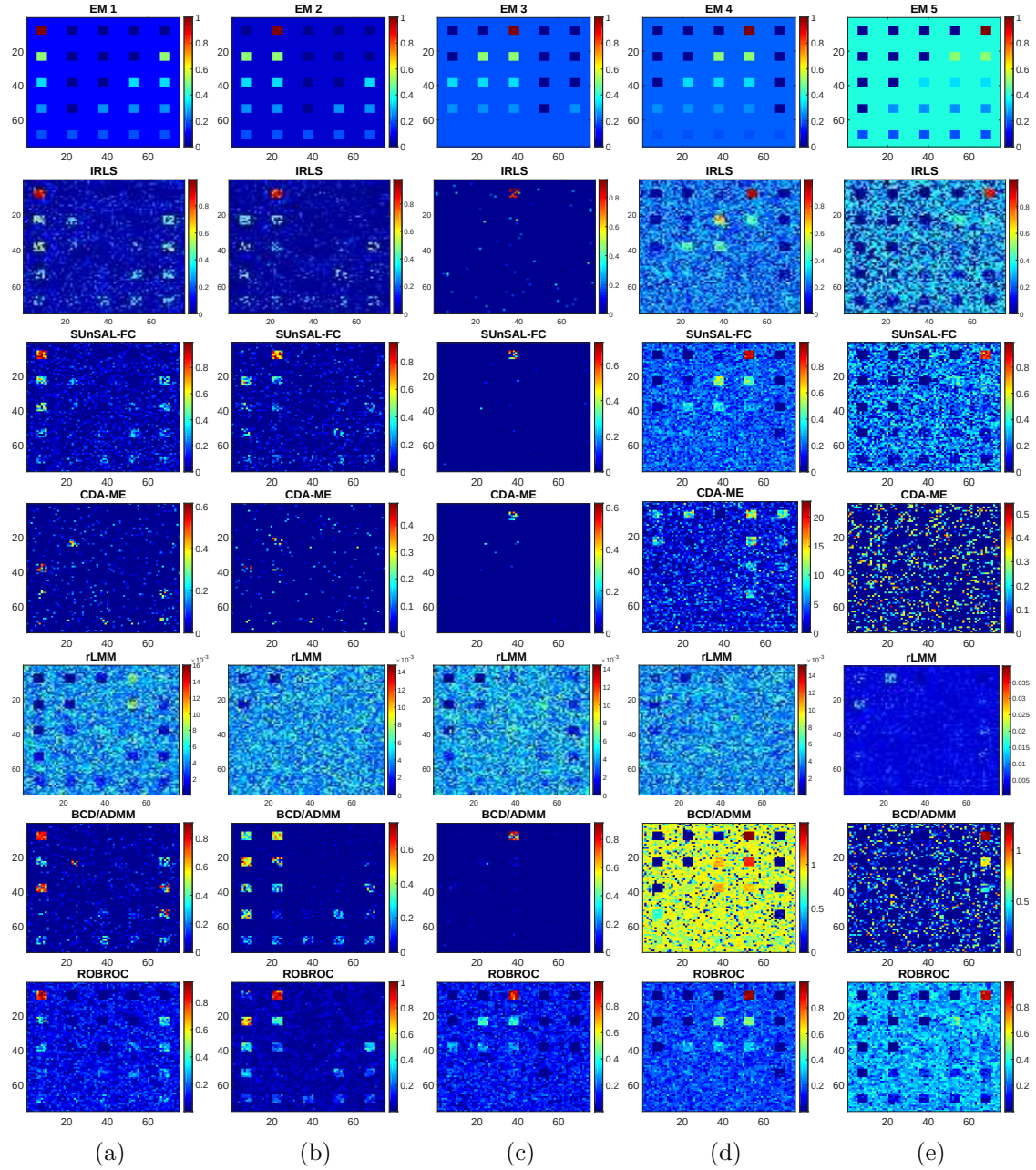


Figure 5.3: Fractional abundance maps obtained by IRLS, SUnSAL-FC, CDA-ME, rLMM, BCD/ADMM, and ROBROC for endmember (EM) 1 in (a), EM 2 in (b), EM 3 in (c), and EM 4 in (d). SNR=20dB.

different levels of endmembers mixed in the pixel such that $N \in \{3, 6, 9\}$.

For the simulated data experiments we used the endmember signatures from the USGS digital spectral library. It consists of the spectral reflectance of hundreds of materials measured in the lab. These spectral reflectance values have 224 bands that lie uniformly in the interval $0.4 - 2.5\mu$m. The chosen signatures can be seen in Fig.2.10. We created a 50×50 image using the LMM in Eq.(1.1). Then we added Gaussian noise such that the

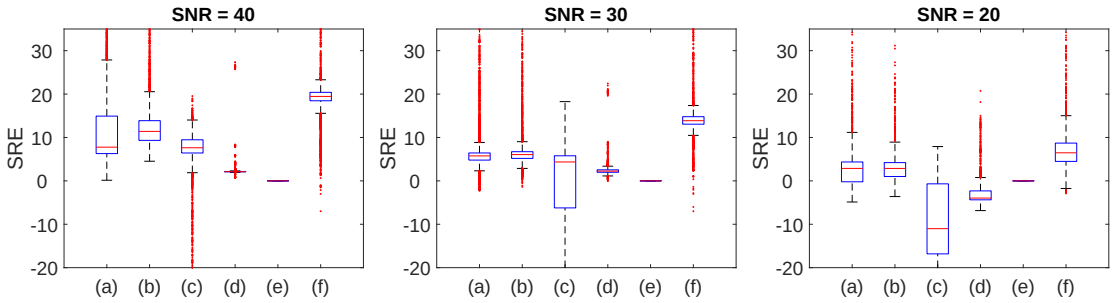


Figure 5.4: Individual SRE box-plots for three different noise levels for (a) IRLS, (b) SUnSAL-FC, (c) CDA-ME, (d) rLMM, (e) BCD/ADMM, (f) ROBROC.

noise level varies across the spectral bands. Then the Dirichlet distribution was used for the uniform generation of the fractional abundance vectors $\boldsymbol{x}$ as in [97, 98].

The proposed ROBROC algorithm has three adjustable parameters: λ, λ_{TV} and p. As the performance of the model depends on the choice of these values, we need a way to select them. Cross-validation (CV) is commonly used for this task, however standard CV uses the mean squared error (MSE) with a squared objective. This means that large errors are given significantly more weight, and the performance results are skewed. Therefore, standard CV is not suitable for robust estimators. Instead of the MSE we used a robust loss function (RLF) defined by: RLF $= 1 - e^{-|\boldsymbol{y}-\hat{\boldsymbol{y}}|}$. Here $\boldsymbol{y}$ is the image data, and $\hat{\boldsymbol{y}}$ is the estimated image data calculated using the estimated fractional abundances $\hat{\boldsymbol{x}}$ and the spectral library $\boldsymbol{M}$. We performed five-fold robust RLF-CV with parameter values ranging from -1 to -5 for $\log_{10}(\lambda)$ and $\log_{10}(\lambda_{TV})$; and values from 0.1 to 1 with an interval of 0.1 for p. The optimal parameter values for the simulated data were $\lambda = \lambda_{TV} = 10^{-3}$, $p = 0.5$.

The unmixing results for various SNR levels and different numbers of endmembers can be found in Table 5.2−5.4. The displayed performance measures were averages over 10 Monte-Carlo realizations for the simulated image. The CDA-ME algorithm had good image reconstruction when there was only a small number of endmembers and high signal-to-noise ratio, but it struggled when the number of endmembers in a pixel increased to 6 or higher. CUSAL-FC performed well for all endmember numbers and SNR levels, but it achieved the best unmixing result only in one instance. The least squares based method SUnSAL-FC mostly performed just slightly worse than CUSAL-FC, and IRLS, another least squares method, showed even worse unmixing. The results of the rLMM, BCD/ADMM and UsGNCM unmixing techniques were weak compared to other methods. The proposed robust algorithm ROBROC showed great performance across all lower SNR levels and all endmember numbers, and acquired the best SRE and RMSE values for most cases. In the few instances where other methods attained a better performance value, ROBROC was close second. The only time ROBROC fell behind other methods was when the SNR was the highest, suggesting that the method performs better in noisier

environments.

Table 5.2: Comparison of unmixing performance of ROBROC with various SNR, N=3

Method	SNR=40				SNR=30			
	SRE	std	RMSE	std	SRE	std	RMSE	std
IRLS	14.74	0.12	6.32	0.08	13.68	0.18	7.15	0.15
SUnSAL-FC	29.43	0.18	1.13	0.02	27.13	0.31	1.39	0.05
CUSAL-FC	29.44	0.18	1.13	0.02	27.26	0.29	1.38	0.04
CDA-ME	29.84	4.45	0.92	0.54	28.79	0.65	0.94	0.10
rLMM	4.80	1.24	19.31	2.66	4.85	1.19	19.13	2.57
BCD/ADMM	27.92	0.14	1.29	0.02	26.52	0.42	1.44	0.06
UsGNCM	12.83	0.14	7.73	0.13	13.11	0.15	7.48	0.14
ROBROC	20.47	0.89	3.28	0.32	27.34	1.39	1.36	0.15
Method	SNR=20				SNR=10			
	SRE	std	RMSE	std	SRE	std	RMSE	std
IRLS	9.34	0.36	11.90	0.53	8.95	0.45	11.98	0.71
SUnSAL-FC	19.94	0.69	3.39	0.30	11.62	0.73	8.94	0.77
CUSAL-FC	19.65	0.26	3.11	0.12	11.99	0.40	7.95	0.42
CDA-ME	**20.93**	0.49	**2.53**	0.18	**12.95**	0.97	**7.29**	1.00
rLMM	2.63	4.98	27.88	36.04	-14.99	16.73	1324.8	3232.8
BCD/ADMM	6.07	0.71	16.34	1.12	1.62	0.07	23.44	0.24
UsGNCM	9.17	0.24	11.99	0.37	4.90	1.35	18.62	2.20
ROBROC	**21.04**	0.49	3.02	0.31	**13.52**	0.38	**7.20**	0.31

Table 5.3: Comparison with various SNR, N=6

Method	SNR=40				SNR=30			
	SRE	std	RMSE	std	SRE	std	RMSE	std
IRLS	7.93	0.11	7.39	0.16	6.04	0.30	9.77	0.39
SUnSAL-FC	26.84	0.52	0.96	0.06	18.42	0.67	2.57	0.20
CUSAL-FC	27.07	0.54	0.91	0.06	18.59	0.64	2.52	0.19
CDA-ME	18.34	0.64	2.53	0.19	10.13	0.67	6.50	0.55
rLMM	2.89	0.22	11.80	0.35	2.75	0.19	11.91	0.36
BCD/ADMM	15.58	0.21	4.01	0.08	9.73	0.63	6.58	0.48
UsGNCM	11.63	0.95	4.48	0.60	8.71	1.00	7.08	0.76
ROBROC	18.43	0.98	2.72	0.31	**19.76**	0.54	**2.21**	0.17
Method	SNR=20				SNR=10			
	SRE	std	RMSE	std	SRE	std	RMSE	std
IRLS	3.09	0.10	13.79	0.39	3.16	0.20	12.07	0.49
SUnSAL-FC	10.41	0.51	6.13	0.24	3.91	0.28	11.20	0.47
CUSAL-FC	**11.33**	0.21	**5.63**	0.16	4.83	0.25	10.66	0.31
CDA-ME	4.53	0.21	11.67	0.44	6.07	0.51	9.45	0.70
rLMM	-0.53	9.45	79.53	214.59	-23.79	18.80	4537.2	12418
BCD/ADMM	-1.92	0.18	21.98	0.63	-3.68	0.09	22.56	0.50
UsGNCM	4.43	0.52	11.57	1.03	-2.29	0.98	26.40	4.19
ROBROC	**11.80**	0.26	**5.20**	0.17	5.33	0.22	9.52	0.22

Table 5.4: Comparison with various SNR, N=9

Method	SNR=40				SNR=30			
	SRE	std	RMSE	std	SRE	std	RMSE	std
IRLS	2.81	0.07	8.76	0.05	1.72	0.12	9.58	0.19
SUnSAL-FC	22.95	0.43	1.22	0.07	14.01	0.41	**3.15**	0.20
CUSAL-FC	22.68	0.40	1.28	0.07	**14.05**	0.45	3.18	0.23
CDA-ME	13.99	0.41	3.09	0.16	7.10	0.29	6.25	0.29
rLMM	2.43	0.10	8.77	0.25	2.18	0.09	8.96	0.19
BCD/ADMM	10.86	0.20	4.78	0.09	6.59	0.32	6.63	0.45
UsGNCM	11.07	0.24	3.50	0.19	6.27	1.16	6.98	1.70
ROBROC	17.85	0.56	3.00	0.08	**15.00**	0.72	**2.83**	0.35
Method	SNR=20				SNR=10			
	SRE	std	RMSE	std	SRE	std	RMSE	std
IRLS	0.03	0.07	12.60	0.24	-0.31	0.13	10.76	0.37
SUnSAL-FC	7.04	0.28	5.95	0.32	1.75	0.27	8.85	0.17
CUSAL-FC	**7.24**	0.28	6.13	0.35	2.72	0.18	8.16	0.23
CDA-ME	2.69	0.16	9.41	0.31	2.76	0.38	8.63	0.56
rLMM	1.79	0.24	9.27	0.70	-24.01	17.14	1349.3	2625.9
BCD/ADMM	-3.06	0.20	18.48	0.46	-5.91	0.06	20.79	0.99
UsGNCM	2.52	0.22	11.54	0.89	-1.84	2.93	22.33	8.76
ROBROC	**8.22**	0.22	**5.22**	0.14	**2.81**	0.20	**8.04**	0.14

The average running times over 10 Monte-Carlo realizations of the simulated image can be seen in Table 5.5. The table gives an indication of the speed of the algorithms, but does not show everything. According to the table the proposed method ROBROC was one of the slowest, similar to the UsGNCM algorithm. However, as mentioned beforehand, we could not get results for the CUSAL-FC and UsGNCM algorithms for the first simulated image, whereas ROBROC showed good performance. The unmixing by the UsGNCM could not be performed due to too large matrices having to be created by Matlab. This indicates that the computational time of the algorithm would have been much slower if it would have been possible to execute. In addition, we let the unmixing of the CUSAL-FC run for two weeks before giving up due to time constraints. Hence, although CUSAL-FC showed a fast computational time in Table 5.5, it is not a feasible option when the image and library size is large.

German Alpine foothills

The Environmental Mapping and Analysis Program (EnMAP) German Alpine foothills image scene was used to test the performance of the proposed algorithm. The hyperspectral image covers the spectral range of 420nm−2460nm that divides into 244 spectral bands. We use a 150 × 150 pixel sub-image from the scene, see Fig. 2.11. We used the N-FINDR algorithm to identify the spectral signatures of the pure endmembers in the image. The optimal parameter values from RLF-CV for the data were $\lambda = \lambda_{TV} = 10$,

Table 5.5: ROBROC average running times in 1000s of seconds

Method	N=3	N=6	N=9
IRLS	0.0017	0.0005	0.0009
SUnSAL-FC	0.0007	0.0024	0.0019
CUSAL-FC	0.0913	0.1600	0.1912
CDA-ME	0.0363	0.0615	0.0652
rLMM	0.0999	0.1507	0.1560
BCD/ADMM	0.1263	0.2314	0.2156
UsGNCM	1.6689	2.9029	3.0337
ROBROC	1.4371	3.2533	7.9527

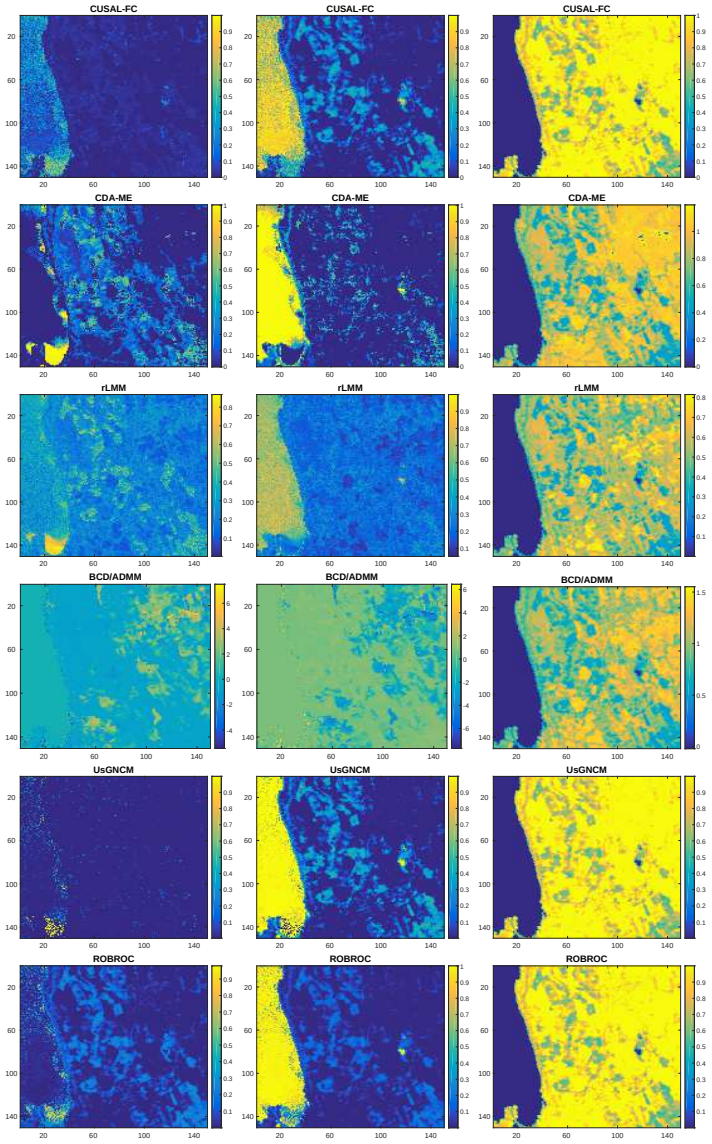
$p = 0.3$. The resulting estimated fractional abundance maps of the Alpine scene for the proposed method and other robust/bilinear algorithms can be found in Fig. 5.5.

The CUSAL-FC, UsGNCM and ROBROC algorithms managed to separate the areas of grass in the image well, and assigned high fractional abundances to these pixels. The proposed ROBROC method identified the water zones in the Alpine image the best. The CUSAL-FC, CDA-ME and UsGNCM appeared to determine water fairly well, but there were certain areas, specifically in the south-east side of the image, where they appeared to have some difficulty. The rLMM and BCD/ADMM did not display a very good performance for the Alpine image in general. Also, the BCD/ADMM algorithm assigned values larger than 1 to some abundances, which is not realistic. None of the methods, except ROBROC, performed well in finding the abundance of the forest.

Jasper Ridge

We used the Jasper Ridge hyperspectral image for unmixing, see Fig. 2.13. The image has 224 spectral bands in the 380nm−2500nm range with a spectral resolution up to 9.46nm. The low SNR and water absorption bands were removed, thus unmixing was performed on 198 remaining bands. The N-FINDR algorithm was used to extract the endmember signatures from the image scene. The optimal parameter values found by RLF-CV were $\lambda = 10^{-1}$,$\lambda_{TV} = 10^{-2}$, $p = 0.4$. The fractional abundance maps for tree, water, soil, and road for the proposed algorithm as well as other robust/bilinear methods are shown in Fig. 5.6.

The CUSAL-FC, RLMM, UsGNCM and ROBROC methods determined the areas of water in the Jasper image very distinctly, consistently assigning them high fractional abundance values. The BCD/ADMM algorithm seemed to find the water areas as well, but there was a lot of noise in the rest of the image. The estimation of tree and soil abundance was fairly similar for all methods. The road areas appeared to be more difficult to determine as most methods were not able to model this. The CDA-ME algorithm falsely identified the shoreline of water as road. ROBROC and CUSAL-FC managed to find road areas the best.



(a) (b) (c)

Figure 5.5: Fractional abundance maps of the German Alpine foothills hyperspectral data obtained for forest in (a), water in (b), and grass in (c).

5.5 Discussion

Sparse regression techniques based on various least squares methods have gained popularity for spectral unmixing. However, the impulsive noise environment of hyperspectral images makes it difficult to handle outliers. Also, in order to obtain realistic estimates for the fractional abundances of endmembers, it is essential to account for sum-to-one and non-negativity constraints, which can be difficult as it makes the optimization problem non-convex. Due to the characteristics of hyperspectral data, it is advantageous to enforce sparsity on the estimates and take the spatial correlation into account.

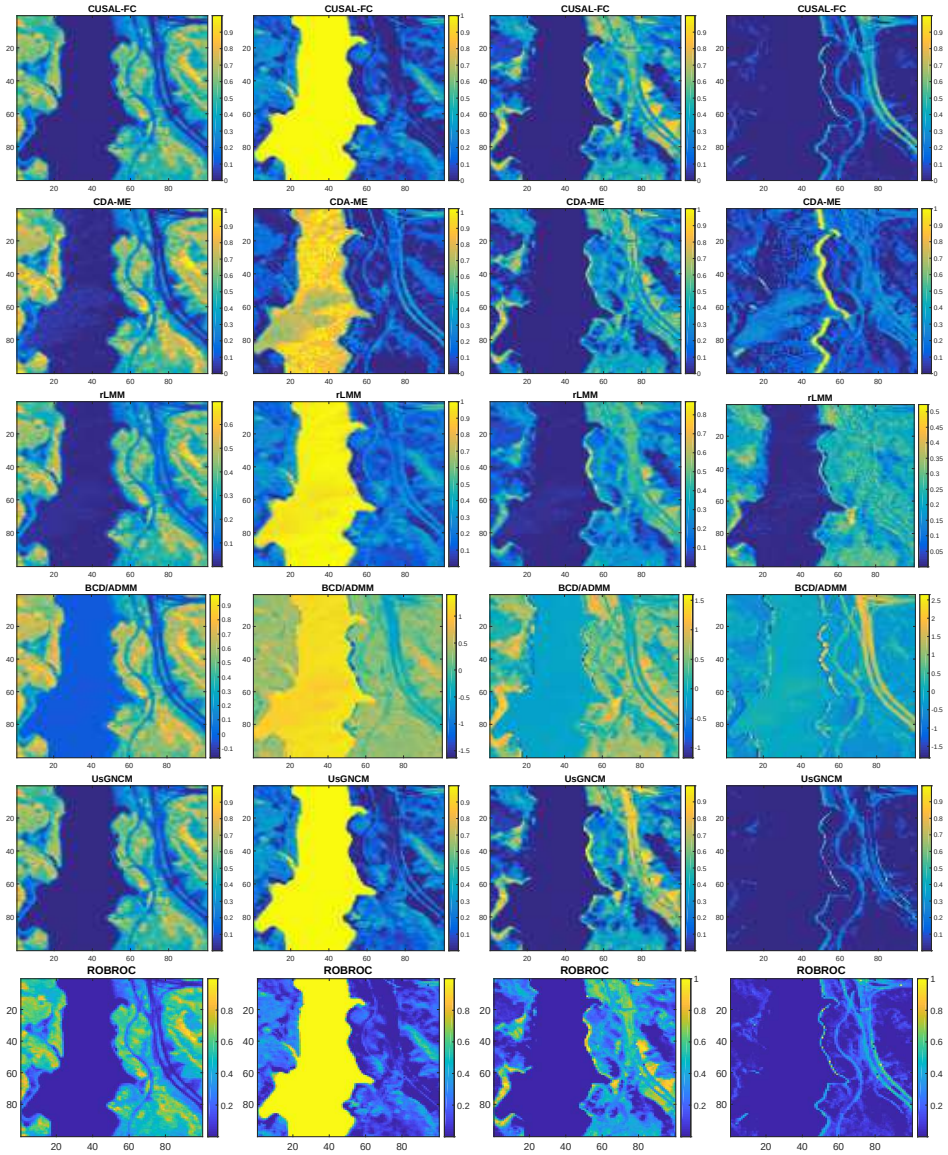


Figure 5.6: Fractional abundance maps of the Jasper Ridge hyperspectral data obtained for tree in (a), water in (b), soil in (c), and road in (d).

In this chapter, we introduced a new algorithm called ROBROC to overcome these restraints and employ a robust model. We introduced a reparametrization of coefficients, to account for the sum-to-one and non-negativity constraints. We also employed a spatial total variation regularization term, and enforced sparsity on the estimates via ℓ_p norm. Even though the optimization problem is non-convex, the trust region optimization based algorithm ROBROC handled the situation well; it solved a sequence of smoothed robust sub-problems instead of directly solving the non-convex ℓ_p norm. Experimental results on simulated and real hyperspectral data showed that the robust M-estimate method with reparametrization of coefficients gave an improved performance, compared to its least

squares counterparts, and other robust and bilinear methods.

Comparison

The IRRF algorithm presented in the previous chapter differs from ROBROC from the way the constraints were handled and by the optimization algorithm used. IRRF enforced constraints in a heuristic manner using renormalization and projection onto the non-negative orthant, which kept the method more simple and flexible. The reparametrization used for ROBROC increased the computational complexity of the algorithm which resulted in slower processing times. The unmixing performance of ROBROC and IRRF varied. Their performance results for the Monte Carlo simulation can be seen in the summary Tables 5.6-5.8, that consolidate the results for the two methods. The unmixing results of ROBROC and IRRF are generally fairly similar. The largest difference in their performance is for high SNR ratio and low endmember number. IRRF is much superior in this case. When looking at the obtained fractional abundance maps of the endmembers in Fig. 4.6 and 5.5 of the Alpine image, ROBROC appeared to identify the water areas better. However, the abundance maps for the Jasper Ridge hyperspectral image (Fig. 4.7 and 5.6) and the simulated hyperspectral image (Fig. 4.2-4.4 and Fig. 5.1-5.3) demonstrated mostly better image reconstruction by the IRRF method. Both algorithms have their benefits and preferred scenarios.

Table 5.6: Comparison of unmixing performance of IRRF and ROBROC, N=3

Method	SNR=40				SNR=30			
	SRE	std	RMSE	std	SRE	std	RMSE	std
IRRF	**37.99**	0.39	**0.41**	0.02	**30.72**	0.46	**0.92**	0.05
ROBROC	20.47	0.89	3.28	0.32	27.34	1.39	1.36	0.15
Method	SNR=20				SNR=10			
	SRE	std	RMSE	std	SRE	std	RMSE	std
IRRF	20.91	0.50	**3.00**	0.19	9.52	0.15	11.01	0.23
ROBROC	**21.04**	0.49	3.02	0.31	**13.52**	0.38	**7.20**	0.31

Table 5.7: Comparison with various SNR, N=6

Method	SNR=40				SNR=30			
	SRE	std	RMSE	std	SRE	std	RMSE	std
IRRF	**28.12**	0.52	**0.80**	0.04	19.27	0.38	2.26	0.10
ROBROC	18.43	0.98	2.72	0.31	**19.76**	0.54	**2.21**	0.17
Method	SNR=20				SNR=10			
	SRE	std	RMSE	std	SRE	std	RMSE	std
IRRF	8.27	0.17	6.75	0.17	**6.14**	0.09	**7.63**	0.13
ROBROC	**11.80**	0.26	**5.20**	0.17	5.33	0.22	9.52	0.22

Table 5.8: Comparison with various SNR, N=9

Method	SNR=40				SNR=30			
	SRE	std	RMSE	std	SRE	std	RMSE	std
IRRF	**23.08**	0.34	**1.19**	0.11	13.59	0.68	3.33	0.28
ROBROC	17.85	0.56	3.00	0.08	**15.00**	0.72	**2.83**	0.35
Method	SNR=20				SNR=10			
	SRE	std	RMSE	std	SRE	std	RMSE	std
IRRF	7.12	0.17	5.88	0.14	**3.59**	0.21	**6.64**	0.22
ROBROC	**8.22**	0.22	**5.22**	0.14	2.81	0.20	8.04	0.14

6

Summary and concluding remarks

Sparse regression techniques based on various least squares methods have gained popularity for spectral unmixing. Despite the success that sparse unmixing has had in certain applications, some of the following limitations need to be considered: enforcing the sum-to-one and non-negativity constraints on the model's parameters is crucial for obtaining realistic results; due to the substantial size of the spectral libraries, using the classical ℓ_1 norm regularization to enforce sparsity is often not sufficient; sparse unmixing methods do not account for spatial correlation in the images; the magnitude of the spectral libraries and the high mutual coherence limit the success of sparse unmixing algorithms; and the impulsive noise environment of hyperspectral images makes it difficult to handle outliers.

We introduced four new algorithms that tackled the above-mentioned limitations and overcame the problems. The main contributions are as follows:

- We proposed to use the ℓ_p norm to enforce further sparsity on the fractional abundance estimates. This non-differentiable problem was solved by an approximation of a weighted 2-norm that integrated easily into our iterative algorithmic framework.

- We employed a spatial total variation regularizer to account for the spatial correlation in the images.

- We included a library reduction step similar to the MUSIC array processing algorithm, that could be used when the number of endmembers was large.

- We introduced a reparametrization of coefficients, to account for the sum-to-one and non-negativity constraints.

- We proposed a robust M-estimate function that adaptively assigned reduced or even zero weights to noisy bands. This reduced the effect of outliers and made the model more insensitive to noise.

Our experimental results support the benefits of the proposals. Specifically, reparametrization of the model's coefficients was an effective and efficient way of taking the constraints into account. Changing the ℓ_1 norm on sparsity to ℓ_p norm with $0 < p < 1$ improved the unmixing performance, and including spatial information via the TV term enhanced the results. In situations when the endmember number was low, the usage of the sparsity promoting ℓ_p norm was still beneficial. The library reduction step made the computation faster and increased the quality of the reconstruction of the images significantly when the initial library size was huge. The unmixing algorithms performed well, even when there was no spectral library available, and the endmembers were extracted from the data by an endmember extraction algorithm. Even though the optimization problem was non-convex, the IRLS and trust region optimization based algorithms handled the situation well. They solved a sequence of smoothed sub-problems instead of directly minimizing the non-convex ℓ_p optimization problem. Experimental results on simulated and real hyperspectral data showed that the introduced sparse total variation regularized hyperspectral unmixing methods gave an improved performance, compared to its least squares counterparts, and other robust and bilinear methods. Finally, the proposed algorithms showed improved performance in noisy situations as compared to other similar unmixing algorithms.

We recommend the use of the different introduced methods according to their strengths. The MUSIC-IRLSTV and ROCSSUM algorithm are more appropriate for images that do not contain many outliers since they are based on the least squares mixing model. ROCSSUM performs better in noisy situations when the spectral library and image size is smaller, or when the speed of computation is not important. The MUSIC-IRSLTV is less complex and provides much faster computational results, and it can be used for larger images as well as in circumstances when speed matters. The IRRF and ROBROC methods can be adopted when there are outliers present in the data. The robust methodology included in the algorithms can handle noisy bands and outliers more competently. For better unmixing performance, it would make sense to determine the presence of outliers by examining the image data before choosing to use robust methods. The ROBROC algorithm generally provides a better unmixing result for lower signal-to-ratio scenarios. However, as the computational complexity of the method is higher, it is more suitable for use on smaller images. The IRRF algorithm on the other hand, is great for unmixing larger images as well as providing computational results in a timely manner.

Bibliography

[1] M. Toomik, S. Lu, and J. D. Nelson, "M-estimation for robust sparse unmixing of hyperspectral images," in *Image and Signal Processing for Remote Sensing XXII*, vol. 10004. International Society for Optics and Photonics, 2016, p. 100040V.

[2] C. Crevoisier, A. Chédin, H. Matsueda, T. Machida, R. Armante, and N. A. Scott, "First year of upper tropospheric integrated content of CO2 from IASI hyperspectral infrared observations," *Atmospheric Chemistry and Physics*, vol. 9, no. 14, pp. 4797–4810, 2009.

[3] T. Matsunaga, S. Yamamoto, and T. Tachikawa, "Detection of large point sources of carbon dioxide by a satellite hyperspectral camera," in *7th Workshop on Hyperspectral Image and Signal Processing: Evolution in Remote Sensing (WHISPERS)*. IEEE, 2015, pp. 1–4.

[4] F. Salem, M. Kafatos, T. El-Ghazawi, R. Gomez, and R. Yang, "Hyperspectral image assessment of oil-contaminated wetland," *International Journal of Remote Sensing*, vol. 26, no. 4, pp. 811–821, 2005.

[5] G. Lassalle, A. Credoz, R. Hedacq, S. Fabre, D. Dubucq, and A. Elger, "Assessing soil contamination due to oil and gas production using vegetation hyperspectral reflectance," *Environmental science & technology*, vol. 52, no. 4, pp. 1756–1764, 2018.

[6] A. W. Nolin and J. Dozier, "A hyperspectral method for remotely sensing the grain size of snow," *Remote sensing of Environment*, vol. 74, no. 2, pp. 207–216, 2000.

[7] A. Jalali and D. P. Shukla, "Snow grain size estimation of a site in the indian himalayan region using hyperspectral remote sensing: Aviris-NG data," in *IGARSS 2019 IEEE International Geoscience and Remote Sensing Symposium*. IEEE, 2019, pp. 4080–4083.

[8] B. Koch, "Status and future of laser scanning, synthetic aperture radar and hyperspectral remote sensing data for forest biomass assessment," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 65, no. 6, pp. 581–590, 2010.

[9] J. Yue, G. Yang, C. Li, Z. Li, Y. Wang, H. Feng, and B. Xu, "Estimation of winter wheat above-ground biomass using unmanned aerial vehicle-based snapshot hyperspectral sensor and crop height improved models," *Remote Sensing*, vol. 9, no. 7, p. 708, 2017.

[10] W. Zhu, Q. Yu, Y. Q. Tian, R. F. Chen, and G. B. Gardner, "Estimation of chromophoric dissolved organic matter in the Mississippi and Atchafalaya river plume regions using above-surface hyperspectral remote sensing," *Journal of Geophysical Research: Oceans (1978–2012)*, vol. 116, no. C2, 2011.

[11] A. Sahay, A. Gupta, G. Motwani, M. Raman, S. M. Ali, M. Shah, S. Chander, P. R. Muduli, and R. Samal, "Distribution of coloured dissolved and detrital organic matter in optically complex waters of Chilika lagoon, Odisha, India, using hyperspectral data of AVIRIS-NG," *Current Science*, vol. 116, no. 7, pp. 1166–1171, 2019.

[12] P.-F. Coheur, L. Clarisse, D. Hurtmans, C. Clerbaux *et al.*, "Atmospheric composition measurements with the IASI hyperspectral sounder: local to global," in *The 22nd Colloquium on High Resolution Molecular Spectroscopy*, 2011.

[13] Q. Pian, R. Yao, N. Sinsuebphon, and X. Intes, "Compressive hyperspectral time-resolved wide-field fluorescence lifetime imaging," *Nature photonics*, vol. 11, no. 7, p. 411, 2017.

[14] B. W. Pengra, C. A. Johnston, and T. R. Loveland, "Mapping an invasive plant, Phragmites australis, in coastal wetlands using the EO-1 Hyperion hyperspectral sensor," *Remote Sensing of Environment*, vol. 108, no. 1, pp. 74–81, 2007.

[15] R. Pu, "Broadleaf species recognition with in situ hyperspectral data," *International Journal of Remote Sensing*, vol. 30, no. 11, pp. 2759–2779, 2009.

[16] B. Hörig, F. Kühn, F. Oschütz, and F. Lehmann, "Hymap hyperspectral remote sensing to detect hydrocarbons," *International Journal of Remote Sensing*, vol. 22, no. 8, pp. 1413–1422, 2001.

[17] A. Xie, D.-W. Sun, Z. Zhu, and H. Pu, "Nondestructive measurements of freezing parameters of frozen porcine meat by NIR hyperspectral imaging," *Food and Bioprocess Technology*, vol. 9, no. 9, pp. 1444–1454, 2016.

[18] N. Pettorelli, W. F. Laurance, T. G. O'Brien, M. Wegmann, H. Nagendra, and W. Turner, "Satellite remote sensing for applied ecologists: opportunities and challenges," *Journal of Applied Ecology*, vol. 51, no. 4, pp. 839–848, 2014.

[19] T. Lillesand, R. W. Kiefer, and J. Chipman, *Remote sensing and image interpretation*. John Wiley & Sons, 2014.

[20] M. E. Winter, "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*. International Society for Optics and Photonics, 1999, pp. 266–275.

[21] J. W. Boardman, "Automating spectral unmixing of AVIRIS data using convex geometry concepts," in *Summaries of the 4th Annual JPL Airborne Geoscience Workshop. AVIRIS Workshop*, vol. 1. JPL, 1993, pp. 11–14.

[22] J. Wang and C.-I. Chang, "Applications of independent component analysis in endmember extraction and abundance quantification for hyperspectral imagery," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 44, no. 9, pp. 2601–2616, 2006.

[23] J. M. Nascimento and J. M. Dias, "Vertex component analysis: A fast algorithm to unmix hyperspectral data," *IEEE transactions on Geoscience and Remote Sensing*, vol. 43, no. 4, pp. 898–910, 2005.

[24] L. Miao and H. Qi, "Endmember extraction from highly mixed data using minimum volume constrained nonnegative matrix factorization," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 45, no. 3, pp. 765–777, 2007.

[25] 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 Transactions on Signal Processing*, vol. 57, no. 11, pp. 4418–4432, 2009.

[26] A. Ambikapathi, T.-H. Chan, W.-K. Ma, and C.-Y. Chi, "A robust alternating volume maximization algorithm for endmember extraction in hyperspectral images," in *2nd Workshop on Hyperspectral Image and Signal Processing: Evolution in Remote Sensing*. IEEE, 2010, pp. 1–4.

[27] J. Li, A. Agathos, D. Zaharie, J. M. Bioucas-Dias, A. Plaza, and X. Li, "Minimum volume simplex analysis: A fast algorithm for linear hyperspectral unmixing," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 53, no. 9, pp. 5067–5082, 2015.

[28] M.-D. Iordache, J. M. Bioucas-Dias, and A. Plaza, "Sparse unmixing of hyperspectral data," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 49, no. 6, pp. 2014–2039, June 2011.

[29] P. J. Gemperline and E. Cash, "Advantages of soft versus hard constraints in self-modeling curve resolution problems. alternating least squares with penalty functions," *Analytical Chemistry*, vol. 75, no. 16, pp. 4236–4243, 2003.

[30] Q. S. Ul Haq, L. Tao, F. Sun, and S. Yang, "A fast and robust sparse approach for hyperspectral data classification using a few labeled samples," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 50, no. 6, pp. 2287–2302, 2012.

[31] R. Tibshirani, "Regression shrinkage and selection via the lasso," *Journal of the Royal Statistical Society. Series B (Methodological)*, pp. 267–288, 1996.

[32] Z. Guo, T. Wittman, and S. Osher, "ℓ_1 unmixing and its application to hyperspectral image enhancement," in *SPIE Defense, Security, and Sensing*. International Society for Optics and Photonics, 2009, pp. 73 341M–73 341M.

[33] N. Akhtar, F. Shafait, and A. Mian, "Futuristic greedy approach to sparse unmixing of hyperspectral data," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 53, no. 4, pp. 2157–2174, 2015.

[34] I. Daubechies, R. DeVore, M. Fornasier, and C. S. Güntürk, "Iteratively reweighted least squares minimization for sparse recovery," *Communications on Pure and Applied Mathematics*, vol. 63, no. 1, pp. 1–38, 2010.

[35] F. Chen and Y. Zhang, "Sparse hyperspectral unmixing based on constrained $\ell_p - \ell_2$ optimization," *IEEE Geoscience and Remote Sensing Letters*, vol. 10, no. 5, pp. 1142–1146, 2013.

[36] 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*, vol. 5, no. 2, pp. 354–379, 2012.

[37] M.-D. Iordache, J. M. 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.

[38] P. Rodríguez and B. Wohlberg, "Efficient minimization method for a generalized total variation functional," *IEEE Transactions on Image Processing*, vol. 18, no. 2, pp. 322–332, 2009.

[39] A. Chambolle, "An algorithm for total variation minimization and applications," *Journal of Mathematical imaging and vision*, vol. 20, no. 1-2, pp. 89–97, 2004.

[40] L. I. Rudin, S. Osher, and E. Fatemi, "Nonlinear total variation based noise removal algorithms," *Physica D: Nonlinear Phenomena*, vol. 60, no. 1, pp. 259–268, 1992.

[41] R. Tibshirani, M. Saunders, S. Rosset, J. Zhu, and K. Knight, "Sparsity and smoothness via the fused lasso," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, vol. 67, no. 1, pp. 91–108, 2005.

[42] R. J. Tibshirani, *The solution path of the generalized lasso*. Stanford University, 2011.

[43] 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.

[44] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," *Cognitive modeling*, vol. 5, pp. 213–220, 1988.

[45] C. Lu, Z. Lin, and S. Yan, "Smoothed low rank and sparse matrix recovery by iteratively reweighted least squares minimization," *IEEE Transactions on Image Processing*, vol. 24, no. 2, pp. 646–654, 2015.

[46] M.-J. Lai, Y. Xu, and W. Yin, "Improved iteratively reweighted least squares for unconstrained smoothed ℓ_q minimization," *SIAM Journal on Numerical Analysis*, vol. 51, no. 2, pp. 927–957, 2013.

[47] C. J. Miosso, R. Von Borries, M. Argàez, L. Velázquez, C. Quintero, and C. M. Potes, "Compressive sensing reconstruction with prior information by iteratively reweighted least-squares," *IEEE Transactions on Signal Processing*, vol. 57, no. 6, pp. 2424–2431, 2009.

[48] M. Fornasier, H. Rauhut, and R. Ward, "Low-rank matrix recovery via iteratively reweighted least squares minimization," *SIAM Journal on Optimization*, vol. 21, no. 4, pp. 1614–1640, 2011.

[49] X. Fu, W.-K. Ma, J. M. Bioucas-Dias, and T.-H. Chan, "Semiblind hyperspectral unmixing in the presence of spectral library mismatches," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 54, no. 9, pp. 5171–5184, 2016.

[50] A. Beck, "On the convergence of alternating minimization for convex programming with applications to iteratively reweighted least squares and decomposition schemes," *SIAM Journal on Optimization*, vol. 25, no. 1, pp. 185–209, 2015.

[51] J. Nocedal and S. Wright, *Numerical optimization*. Springer Science & Business Media, 2006.

[52] Y.-X. Yuan, "A review of trust region algorithms for optimization," in *Iciam*, vol. 99, 2000, pp. 271–282.

[53] Y. Wu, E. Mansimov, R. B. Grosse, S. Liao, and J. Ba, "Scalable trust-region method for deep reinforcement learning using Kronecker-factored approximation," in *Advances in neural information processing systems*, 2017, pp. 5279–5288.

[54] N. I. Gould, D. P. Robinson, and H. S. Thorne, "On solving trust-region and other regularised subproblems in optimization," *Mathematical Programming Computation*, vol. 2, no. 1, pp. 21–57, 2010.

[55] Y. Tarabalka, J. A. Benediktsson, and J. Chanussot, "Spectral–spatial classification of hyperspectral imagery based on partitional clustering techniques," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 47, no. 8, pp. 2973–2987, 2009.

[56] X. Kang, S. Li, and J. A. Benediktsson, "Spectral–spatial hyperspectral image classification with edge-preserving filtering," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 52, no. 5, pp. 2666–2677, 2014.

[57] J. Li, J. M. Bioucas-Dias, and A. Plaza, "Spectral–spatial classification of hyperspectral data using loopy belief propagation and active learning," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 51, no. 2, pp. 844–856, 2013.

[58] M. Fauvel, J. A. Benediktsson, J. Chanussot, and J. R. Sveinsson, "Spectral and spatial classification of hyperspectral data using SVMs and morphological profiles," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 46, no. 11, pp. 3804–3814, 2008.

[59] A. Plaza, J. A. Benediktsson, J. W. Boardman, J. Brazile, L. Bruzzone, G. Camps-Valls, J. Chanussot, M. Fauvel, P. Gamba, A. Gualtieri *et al.*, "Recent advances in techniques for hyperspectral image processing," *Remote sensing of environment*, vol. 113, pp. S110–S122, 2009.

[60] J. Li, J. M. Bioucas-Dias, and A. Plaza, "Spectral–spatial hyperspectral image segmentation using subspace multinomial logistic regression and markov random fields," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 50, no. 3, pp. 809–823, 2012.

[61] G. Martín and A. Plaza, "Spatial-spectral preprocessing prior to endmember identification and unmixing of remotely sensed hyperspectral data," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 5, no. 2, pp. 380–395, 2012.

[62] Y. Chen, N. M. Nasrabadi, and T. D. Tran, "Hyperspectral image classification using dictionary-based sparse representation," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 49, no. 10, pp. 3973–3985, 2011.

[63] W. Tang, Z. Shi, and Y. Wu, "Regularized simultaneous forward–backward greedy algorithm for sparse unmixing of hyperspectral data," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 52, no. 9, pp. 5271–5288, 2014.

[64] A. Zare, "Spatial-spectral unmixing using fuzzy local information," in *IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*. IEEE, 2011, pp. 1139–1142.

[65] S. Jia and Y. Qian, "Spectral and spatial complexity-based hyperspectral unmixing," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 45, no. 12, pp. 3867–3879, 2007.

[66] A. Zare, O. Bchir, H. Frigui, and P. Gader, "Spatially-smooth piece-wise convex endmember detection," in *2nd Workshop on Hyperspectral Image and Signal Processing: Evolution in Remote Sensing (WHISPERS)*. IEEE, 2010, pp. 1–4.

[67] A. Huck and M. Guillaume, "Robust hyperspectral data unmixing with spatial and spectral regularized NMF," in *2nd Workshop on Hyperspectral Image and Signal Processing: Evolution in Remote Sensing (WHISPERS)*. IEEE, 2010, pp. 1–4.

[68] M. V. Afonso, J. M. Bioucas-Dias, and M. A. T. Figueiredo, "An augmented lagrangian approach to the constrained optimization formulation of imaging inverse problems," *IEEE Transactions on Image Processing*, vol. 20, no. 3, pp. 681–695, 2011.

[69] A. Zymnis, S. J. Kim, J. Skaf, M. Parente, and S. Boyd, "Hyperspectral image unmixing via alternating projected subgradients," in *Conference Record of the Forty-First Asilomar Conference on Signals, Systems and Computers (ACSSC)*. IEEE, 2007, pp. 1164–1168.

[70] J. Sigurdsson, M. O. Ulfarsson, and J. R. Sveinsson, "Blind hyperspectral unmixing using total variation and ℓ_q sparse regularization," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 54, no. 11, pp. 6371–6384, 2016.

[71] W. He, H. Zhang, and L. Zhang, "Total variation regularized reweighted sparse nonnegative matrix factorization for hyperspectral unmixing," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 55, no. 7, pp. 3909–3921, 2017.

[72] F. D. Van der Meer and X. Jia, "Collinearity and orthogonality of endmembers in linear spectral unmixing," *International Journal of Applied Earth Observation and Geoinformation*, vol. 18, pp. 491–503, 2012.

[73] A. M. Bruckstein, D. L. Donoho, and M. Elad, "From sparse solutions of systems of equations to sparse modeling of signals and images," *SIAM review*, vol. 51, no. 1, pp. 34–81, 2009.

[74] G. Bienvenu and L. Kopp, "Adaptivity to background noise spatial coherence for high resolution passive methods," in *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, vol. 5. IEEE, 1980, pp. 307–310.

[75] R. O. Schmidt, "Multiple emitter location and signal parameter estimation," *IEEE Transactions on Antennas and Propagation*, vol. 34, no. 3, pp. 276–280, 1986.

[76] M.-D. Iordache, J. M. Bioucas-Dias, A. Plaza, and B. Somers, "MUSIC-CSR: Hyperspectral unmixing via multiple signal classification and collaborative sparse regression," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 52, no. 7, pp. 4364–4382, 2014.

[77] X. Fu, W.-K. Ma, J. M. Bioucas-Dias, and T.-H. Chan, "A robust subspace method for semiblind dictionary-aided hyperspectral unmixing," in *Proc. IEEE WHISPERS*, 2014, pp. 1–4.

[78] D. C. Heinz and C.-I. Chang, "Fully constrained least squares linear spectral mixture analysis method for material quantification in hyperspectral imagery," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 39, no. 3, pp. 529–545, 2001.

[79] O. Eches, N. Dobigeon, and J.-Y. Tourneret, "Enhancing hyperspectral image unmixing with spatial correlations," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 49, no. 11, p. 4239, 2011.

[80] O. Eches and J.-Y. Tourneret, "Markov random fields for joint unmixing and segmentation of hyperspectral images," in *2nd Workshop on Hyperspectral Image and Signal Processing: Evolution in Remote Sensing (WHISPERS)*. IEEE, 2010, pp. 1–4.

[81] W. Zucchini, I. L. MacDonald, and R. Langrock, *Hidden Markov models for time series: an introduction using R*. Chapman and Hall/CRC, 2016, pp. 9–11.

[82] K. Themelis and A. A. Rontogiannis, "A soft constrained map estimator for supervised hyperspectral signal unmixing," in *16th European Signal Processing Conference*. IEEE, 2008, pp. 1–5.

[83] F. Bartolucci and A. Farcomeni, "A multivariate extension of the dynamic logit model for longitudinal data based on a latent markov heterogeneity structure," *Journal of the American Statistical Association*, vol. 104, no. 486, pp. 816–831, 2009.

[84] O. Eches, J. A. Benediktsson, N. Dobigeon, and J.-Y. Tourneret, "Adaptive markov random fields for joint unmixing and segmentation of hyperspectral images," *IEEE Transactions on Image Processing*, vol. 22, no. 1, pp. 5–16, 2012.

[85] N. Dobigeon, S. Moussaoui, J.-Y. Tourneret, and C. Carteret, "Bayesian separation of spectral sources under non-negativity and full additivity constraints," *Signal Processing*, vol. 89, no. 12, pp. 2657–2669, 2009.

[86] S.-C. Chan and Y.-X. Zou, "A recursive least M-estimate algorithm for robust adaptive filtering in impulsive noise: fast algorithm and convergence performance analysis," *IEEE Transactions on Signal Processing*, vol. 52, no. 4, pp. 975–991, 2004.

[87] H. Xu, C. Caramanis, and S. Mannor, "Robust regression and lasso," in *Advances in Neural Information Processing Systems*, 2009, pp. 1801–1808.

[88] R. A. Maronna, R. D. Martin, V. J. Yohai, and M. Salibián-Barrera, *Robust statistics: theory and methods (with R)*. John Wiley & Sons, 2019.

[89] E. Cantoni and E. Ronchetti, "Resistant selection of the smoothing parameter for smoothing splines," *Statistics and Computing*, vol. 11, no. 2, pp. 141–146, 2001.

[90] R. K. Wong, F. Yao, and T. C. Lee, "Robust estimation for generalized additive models," *Journal of Computational and Graphical Statistics*, vol. 23, no. 1, pp. 270–289, 2014.

[91] C. Croux, I. Gijbels, and I. Prosdocimi, "Robust estimation of mean and dispersion functions in extended generalized additive models," *Biometrics*, vol. 68, no. 1, pp. 31–44, 2012.

[92] C. Févotte and N. Dobigeon, "Nonlinear hyperspectral unmixing with robust nonnegative matrix factorization," *IEEE Transactions on Image Processing*, vol. 24, no. 12, pp. 4810–4819, 2015.

[93] A. Halimi, P. Honeine, and J. M. Bioucas-Dias, "Hyperspectral unmixing in presence of endmember variability, nonlinearity, or mismodeling effects," *IEEE Transactions on Image Processing*, vol. 25, no. 10, pp. 4565–4579, 2016.

[94] T. Veracini, S. Matteoli, M. Diani, and G. Corsini, "Robust hyperspectral image segmentation based on a non-gaussian model," in *2nd International Workshop on Cognitive Information Processing*. IEEE, 2010, pp. 192–197.

[95] D. B. Marden and D. G. Manolakis, "Modeling hyperspectral imaging data," in *AeroSense*. International Society for Optics and Photonics, 2003, pp. 253–262.

[96] A. Alimadad and M. Salibian-Barrera, "An outlier-robust fit for generalized additive models with applications to disease outbreak detection," *Journal of the American Statistical Association*, vol. 106, no. 494, pp. 719–731, 2011.

[97] F. Zhu, A. Halimi, P. Honeine, B. Chen, and N. Zheng, "Correntropy maximization via ADMM: application to robust hyperspectral unmixing," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 55, no. 9, pp. 4944–4955, 2017.

[98] A. Halimi, N. Dobigeon, and J.-Y. Tourneret, "Unsupervised unmixing of hyperspectral images accounting for endmember variability," *IEEE Transactions on Image Processing*, vol. 24, no. 12, pp. 4904–4917, 2015.

[99] P.-A. Thouvenin, N. Dobigeon, and J.-Y. Tourneret, "Hyperspectral unmixing with spectral variability using a perturbed linear mixing model," *IEEE Transactions on Signal Processing*, vol. 64, no. 2, pp. 525–538, 2016.

[100] J. M. Bioucas-Dias and J. M. P. Nascimento, "Hyperspectral subspace identification," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 46, no. 8, pp. 2435–2445, 2008.

[101] M.-D. Iordache, J. M. Bioucas-Dias, and A. Plaza, "Collaborative sparse regression for hyperspectral unmixing," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 52, no. 1, pp. 341–354, 2014.

[102] J. M. Bioucas-Dias and M. A. Figueiredo, "Alternating direction algorithms for constrained sparse regression: Application to hyperspectral unmixing," in *2nd Workshop on Hyperspectral Image and Signal Processing: Evolution in Remote Sensing*. IEEE, 2010, pp. 1–4.

[103] H. Pu, Z. Chen, B. Wang, and W. Xia, "Constrained least squares algorithms for nonlinear unmixing of hyperspectral imagery," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 53, no. 3, pp. 1287–1303, 2015.

[104] C. Chen, L. He, H. Li, and J. Huang, "Fast iteratively reweighted least squares algorithms for analysis-based sparse reconstruction," *Medical image analysis*, vol. 49, pp. 141–152, 2018.

[105] N. Bissantz, L. Dümbgen, A. Munk, and B. Stratmann, "Convergence analysis of generalized iteratively reweighted least squares algorithms on convex function spaces," *SIAM Journal on Optimization*, vol. 19, no. 4, pp. 1828–1845, 2009.

[106] X. Zongben, C. Xiangyu, X. Fengmin, and Z. Hai, "L1/2 regularization: a thresholding representation theory and a fast solver." *IEEE Transactions on neural networks and learning systems*, vol. 23, no. 7, pp. 1013–1027, 2012.

[107] S. K. Sengijpta, "Fundamentals of statistical signal processing: Estimation theory," *Technometrics*, vol. 37, no. 4, pp. 465–466, 1995.

[108] N. Keshava and J. F. Mustard, "Spectral unmixing," *IEEE Signal Processing Magazine*, vol. 19, no. 1, pp. 44–57, 2002.

[109] J. J. Settle and N. A. Drake, "Linear mixing and the estimation of ground cover proportions," *International Journal of Remote Sensing*, vol. 14, no. 6, pp. 1159–1177, 1993. [Online]. Available: http://dx.doi.org/10.1080/01431169308904402

[110] M. Zortea and A. Plaza, "Spatial preprocessing for endmember extraction," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 47, no. 8, pp. 2679–2693, 2009.

[111] Y. Wang, C. Pan, S. Xiang, and F. Zhu, "Robust hyperspectral unmixing with correntropy-based metric," *IEEE Transactions on Image Processing*, vol. 24, no. 11, pp. 4027–4040, 2015.

[112] S. Migdall, H. Bach, and H. Kaufmann, "Simulation of EnMAP-like hyperspectral images based on textural information and radiative transfer approach," in *Proceedings of the Hyperspectral Workshop, Frascati, Italy*, 2010, pp. 17–19.

[113] F. Bartolucci, A. Farcomeni, and F. Pennoni, "A note on the application of the oakes' identity to obtain the observed information matrix of hidden markov models," *arXiv preprint arXiv:1201.5990*, 2012.

[114] R. H. Byrd, R. B. Schnabel, and G. A. Shultz, "Approximate solution of the trust region problem by minimization over two-dimensional subspaces," *Mathematical programming*, vol. 40, no. 1-3, pp. 247–263, 1988.

[115] M. A. Branch, T. F. Coleman, and Y. Li, "A subspace, interior, and conjugate gradient method for large-scale bound-constrained minimization problems," *SIAM Journal on Scientific Computing*, vol. 21, no. 1, pp. 1–23, 1999.

[116] T. Steihaug, "The conjugate gradient method and trust regions in large scale optimization," *SIAM Journal on Numerical Analysis*, vol. 20, no. 3, pp. 626–637, 1983.

[117] J. J. Moré and D. C. Sorensen, "Computing a trust region step," *SIAM Journal on Scientific and Statistical Computing*, vol. 4, no. 3, pp. 553–572, 1983.

[118] Y. Altmann, A. Maccarone, A. McCarthy, G. Newstadt, G. S. Buller, S. McLaughlin, and A. Hero, "Robust spectral unmixing of sparse multispectral Lidar waveforms using gamma Markov random fields," *IEEE Transactions on Computational Imaging*, vol. 3, no. 4, pp. 658–670, 2017.

[119] Z. G. Zhang, S. C. Chan, Y. Zhou, and Y. Hu, "Robust linear estimation using m-estimation and weighted l1 regularization: Model selection and recursive implementation," in *IEEE International Symposium on Circuits and Systems (ISCAS)*. IEEE, 2009, pp. 1193–1196.

[120] P. J. Huber, *Robust statistics*. Springer, 2011.

[121] Z. Zhang, S.-C. Chan, K.-L. Ho, and K. Ho, "On bandwidth selection in local polynomial regression analysis and its application to multi-resolution analysis of non-uniform data," *Journal of Signal Processing Systems*, vol. 52, no. 3, pp. 263–280, 2008.