2016

Accelerated Hyperspectral Unmixing with Endmember Variability via the Sum-Product Algorithm

Charan Puladas
Wright State University

Accelerated Hyperspectral Unmixing with Endmember Variability via the Sum-Product Algorithm

A thesis submitted in partial fulfillment
of the requirements for the degree of
Master of Science in Electrical Engineering

by

Charan Puladas
Bachelor of Technology, Jawaharlal Nehru Technological University, 2014

2016
Wright State University

Wright State University
GRADUATE SCHOOL

May 25, 2016

I HEREBY RECOMMEND THAT THE THESIS PREPARED UNDER MY SUPERVISION BY Charan Puladas ENTITLED Accelerated Hyperspectral Unmixing with - Endmember Variability via the Sum-Product Algorithm BE ACCEPTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF Master of Science in Electrical Engineering.

_____

Dr. Joshua N. Ash
Thesis Director

_____

Dr. Brian D. Rigling
Chair, Department of Electrical Engineering

Committee on
Final Examination

_____

Joshua N. Ash, Ph.D

_____

Arnab K. Shaw, Ph.D

_____

Steve Gorman, Ph.D

_____

Dr. Robert E.W. Fyffe, Ph.D
Vice President for Research and
Dean of the Graduate School

ABSTRACT

Puladas, Charan. M.S.E.E., Department of Electrical Engineering, Wright State University, 2016. *Accelerated Hyperspectral Unmixing with Endmember Variability via the Sum-Product Algorithm.*

The rich spectral information captured by hyperspectral sensors has given rise to a number of remote sensing applications, ranging from vegetative assessment and crop health monitoring, to military surveillance and combatant identification. However, due to limited spatial resolution, multiple ground materials generally contribute, i.e. mix, to form the spectrum recorded for a single pixel. The unmixing problem considers the inverse problem of determining the underlying material spectra, called endmembers, from sensor measurements. While classical unmixing approaches were deterministic in nature and did not attempt to identify in-scene materials, recent methods use labeled training data to generate statistical models of endmember variabilities and perform statistical unmixing for simultaneous material identification and abundance estimation.

However, the computational complexity of statistical unmixing is $\mathcal{O}(N^3)$, cubic in the number N of sensed spectral bands. This large computational demand is at odds with continuous technological improvements that are dramatically increasing the spectral resolution of remote spectroscopy methods. In particular, current sensor technology is transitioning from the hyperspectral realm (hundreds of spectral bands) to the ultraspectral realm (thousands of spectral bands) and eclipsing the ability to perform statistical unmixing.

In this thesis we develop a computationally tractable statistical unmixing method. The proposed method uses Markov chains to model endmember variability and the spectral correlation properties present within endmembers. We use a probabilistic graphical model over multiple Markov chains to capture the mixing effects of the spectral sensor and employ sum-product message passing to develop an accelerated statistical unmixing algorithm. The computational complexity, $\mathcal{O}(NM^3)$, of the proposed algorithm is only linear in the number of bands and depends on the number of endmembers M in a cubic fashion. As M

is generally small and fixed (in the 10s), the accelerated algorithm represents a dramatic speed-up over existing methods. Examples demonstrate comparable error rates with two orders of magnitude reduction in computation time compared to existing statistical unmixing methods.

Contents

List of Figures

Acknowledgment

I would like to take this opportunity to express my sincere gratitude to my thesis advisor, Dr. Josh Ash for trusting my capabilities, encouraging me to embark on research and supporting me throughout my thesis. I would also like to thank my thesis committee members, Dr. Arnab Shaw and Dr. Steve Gorman for taking time in reviewing my thesis. I am grateful to Wright State University and the Electrical Engineering department for the continued financial support throughout my thesis.

To Amma, Nanna, Lakshmi...

Chapter 1: Introduction

1.1 Overview of Hyperspectral Imaging

A hyperspectral sensor captures light that has been reflected back from a target material to a sensor [27]. This is called the reflectance of the material, and it is measured across different wavelengths of light which, collectively, constitute the spectral signature of a material. Depending on the application, hyperspectral sensors may be tuned to different bands in the electromagnetic spectrum and have different spectral sampling rates. Because reflectance is a key intrinsic characteristic of materials, hyperspectral imagery is often used to identify target materials in a scene.

The data from a hyperspectral image can be depicted as a cube, with two spatial dimensions —as in traditional color imagery—and the third dimension being the spectral dimension. Each pixel on this cube would enclose a certain portion of the land being surveyed and contains reflectance values of various materials that are being surveyed. Each material exhibits characteristic spectral signature that differs from other materials and hence is a key component in extracting a material from hyperspectral data.

High resolution is a key element in most imaging applications as this would enhance the quality of the image thereby throwing light on finer qualities present in the scene. Spatial resolution is key in monochromatic and color images but for applications where material

detection or its composition are needed then having better spectral resolution is required. Spectral resolution is directly proportional to the number of samples of spectral bands in a band of spectrum. In spatial domain, challenge is with the size of the pixel but in spectral domain the challenge is with spectral variability. Spectral variability is the change of spectral signature among different samples of the same material. Accounting for spectral variability or popularly called endmember variability can be difficult but for reliable detection of materials it has to be taken into account. More details about endmember variability are in Section 1.3.1.

An illustration of a hyperspectral scene under surveillance is shown below:

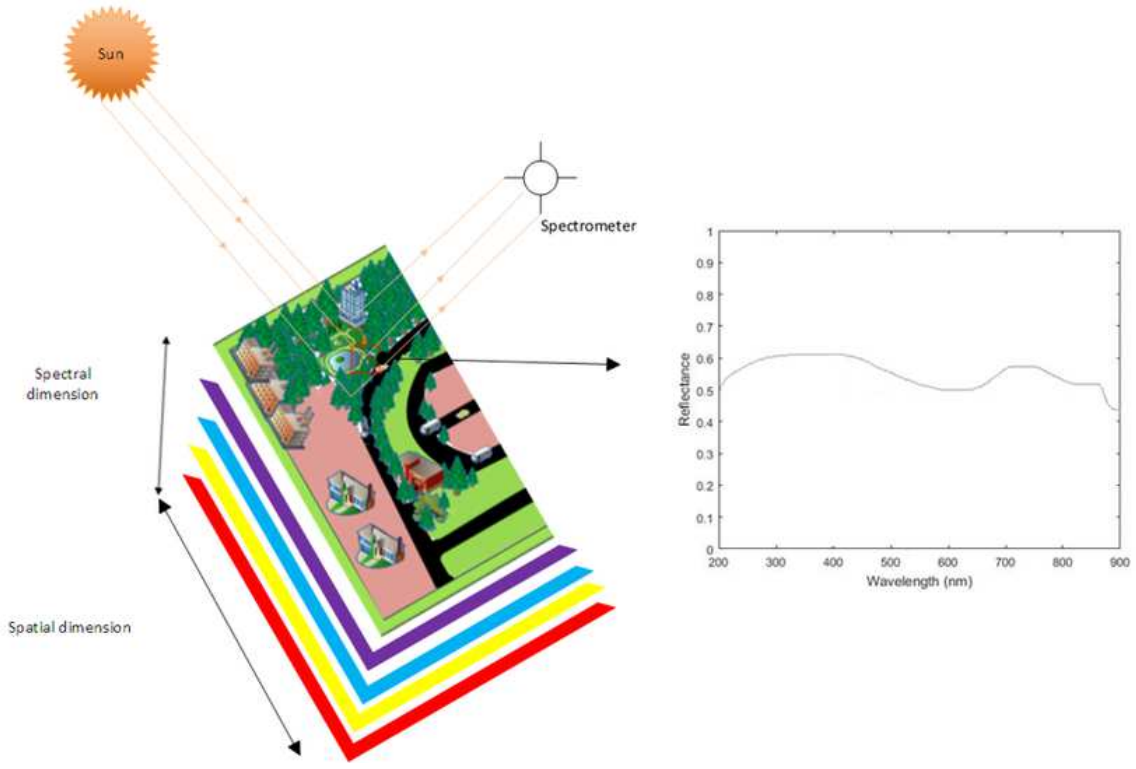

Figure 1.1: Scene under surveillance

In the above image the highlighted pixel in red block consists of different materials like a waterbody, vegetation and concrete. These materials in context of hyperspectral

imaging are referred as endmembers. Data in a pixel encloses spatial information at various wavelengths along the spectrum thus creating the 3-D structure.

1.1.1 Difference between Color, Multispectral, and Hyperspectral Cameras

A color camera captures light across 3 wavelengths of the electromagnetic spectrum: red, green and blue. Traditional color cameras have been designed this way to capture information similar to the human eye, including only the visible band from 380 nm to 700 nm. In contrast, an imaging spectrometer has the capability of surveying a scene in the spatial and spectral domains, but captures reflectance over many (> 3) spectral bands—potentially outside the visible region.

As illustrated in Figure 1.2, there are different classes of imaging spectrometers, each defined by the number of spectral bands captured by the sensor. In multispectral imaging, the number of spectral bands being sampled is approximately 10–20. However, continuous technological improvements have led to ever-increasing spectral resolution of remote spectroscopy methods, leading to hyperspectral cameras that capture 100's of spectral bands. Some recently developed sensors that capture 1000's of spectral bands are referred to as ultraspectral sensors. Landsat-8 is one of the popular multispectral sensor while Hyperion & AVIRIS are well-known hyperspectral sensors. The increasing spectral resolution of sensing technology is outpacing the processing capabilities of current sensor exploitation algorithms [7], and as described in Section 1.4, addressing this issue is the primary focus of this thesis. Additionally, and for convenience, we generally refer to N-band imagery as hyperspectral imagery (HSI), regardless of the particular number of bands N.

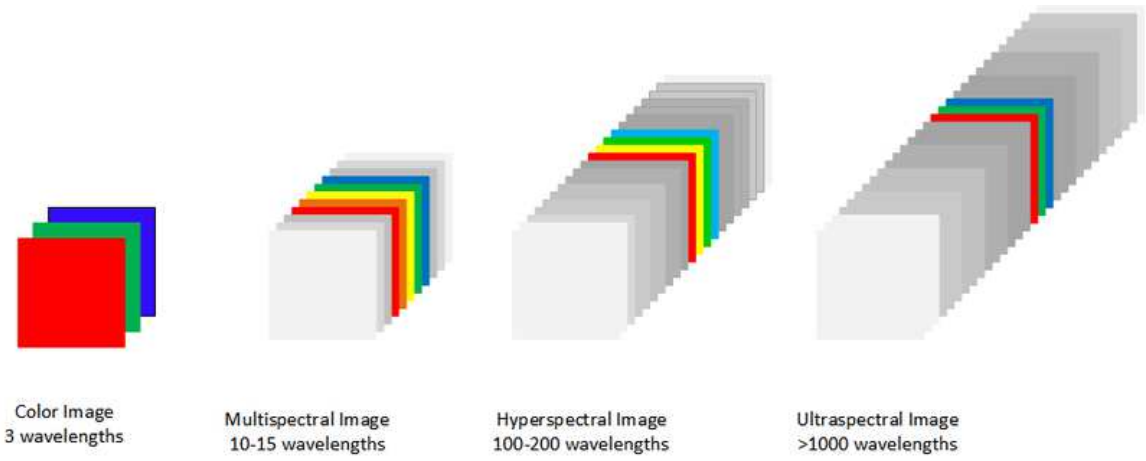

Figure 1.2: Spectral resolution in various sensors

1.2 Applications of Hyperspectral Imaging

Traditional uses of hyperspectral were primarily focused on remote sensing applications like mining for mineral ores, but this has changed in the recent past. Hyperspectral imaging is currently being used in a wide range of domains, and here we highlight a few example applications:

- **Mineralogy**:

  One of the first applications of hyperspectral imaging was in mineralogy because of its inherit traits. Different minerals in there crude form are identified through their unique spectral signatures. [11]

- **Surveillance**:

  It has been used to spy on enemy encampments and detect technological advancements being made in enemy countries using drones and satellite imagery.[8]

- **Agriculture**:

  This is a recent development where crop health is monitored and variations of crop

health in different geographical locations are recorded. HSI has also been used to control pests in agricultural fields.[20]

- **Ophthalmology**:

  Hyperspectral imaging has been used to detect the levels of oxygen in the retina to monitor eye vision[23].

- **Astronomy & space surveillance**:

  Similar to mineralogy, HSI characteristics fit well for astronomy where detection of materials and their composition is necessary to understand characteristics of distant objects. As spectral signatures are unique for every material, HSI is a powerful tool to detect them[12].

- **Food and pharmaceutical processing**:

  HSI is used to identify any faults, defects, and foreign bodies present in food materials which might not be detected by laser-based methods or high spatial-resolution cameras[17].

- **Environment**:

  Hyperspectral imaging is being used to monitor the composition of the atmosphere and detect changes in its composition. For example, it is used to monitor for harmful substances leaking in to the atmosphere from chemical industries[6].

While the list of HSI applications is diverse, one common theme is the use of measured spectra to identify particular materials and their abundances within imagery. This task is referred to as *unmixing* and is desribed in the following section.

1.3 The Unmixing Problem

Because hyperspectral sensors are often deployed far away from the area that they are imaging, the ground sample distance (GSD) is generally large. For example, the airborne

AVIRIS sensor has a ground spatial resolution of $20\text{m} \times 20\text{m}$ [24]. With low spatial resolution, i.e. "large pixels", there is a large chance that multiple ground materials contribute to the measured spectrum for each image pixel. The linear mixing model (further described in Sec. 2.1.1) describes how individual material spectra, referred to as *endmembers*, "mix" to form a measured spectrum

$$\boldsymbol{y} = \sum_{m=1}^{M} \beta_m \boldsymbol{x}^{(m)}. \tag{1.1}$$

Here $\boldsymbol{y}$ is the observed hyperspectral data for a given pixel. It is an $N \times 1$ vector containing the reflectance values across N different wavelengths. The variable $\boldsymbol{x}^{(m)}$ represents the spectral signature of the m^{th} endmember, and $0 \leq \beta_m \leq 1$ is the fractional composition of the m^{th} endmember in the pixel. The term coined for the fractional composition β_m of an endmember in spectral imaging is the *abundance ratio*, and $\sum_m \beta_m = 1$. Although the sensor measures mixed data $\boldsymbol{y}$, we are generally interested in the knowing the constituent materials in the scene. Consequently, *unmixing* is the process of determining the endmember spectra $\{\boldsymbol{x}^{(m)}\}$and estimating their corresponding abundances $\{\beta_m\}$ from measured data—thus unwinding the mixing process[21].

An illustration of mixing and unmixing is shown in Figure 1.3. The left plot illustrates 4 different endmember spectral signatures, and the right plot illustrates a mixed linear combination of those signatures. The unmixing process attempts to recreate the four endmembers on the left, and their abundances $\{\beta_m\}$, from the mixed measurements available in the image.

1.3.1 Endmember Variability

While classical unmixing approaches typically treat endmembers as constant quantities, more recent approaches accomodate natural variability within endmembers. For example, the spectrum of grass may vary depending on the species of grass considered and its health state, e.g., dry grass versus healthy green-colored grass. Another example, taken from the

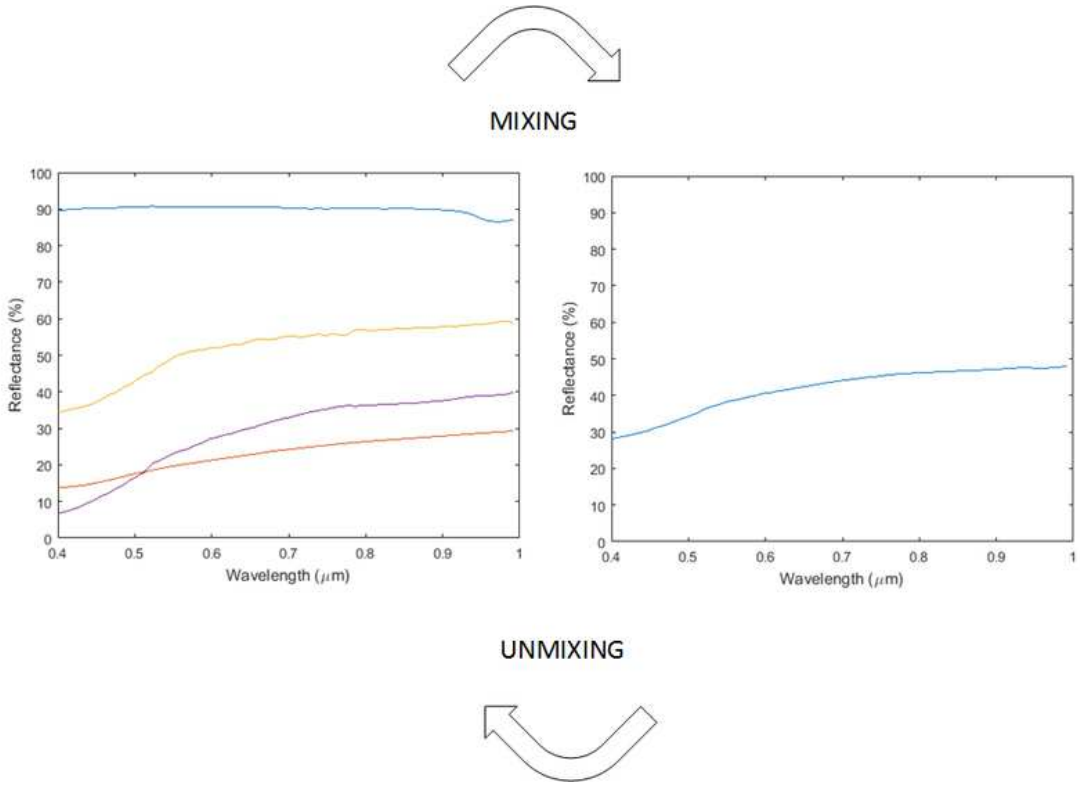

Figure 1.3: *Mixing*: multiple endmember spectra (left) are combined to form a single mixed spectrum (right). *Unmixing*: from a mixed spectrum, unmixing attempts to isolate the original endmember spectra and their abundances.

ASTER spectral library [2], is depicted in Figure 1.4. Here, natural variations in a Calcite compound contribute to variability in the endmember spectra. Although both samples are of calcite compounds, we observe that they have different reflectance signatures. Consequently, understanding and modeling endmember variability is important for unmixing and material identification algorithms. As noted in [1], endmember variability may be modeled in 2 ways:

1. Endmembers as sets.

2. Endmembers as statistical distribution.

In the first method, a set of viable endmembers for each material class is considered. The second method considers endmembers as random vectors described by a probability density function $p(\boldsymbol{x}^{(m)})$ for each material class m. The statistical approach is more common and consequently has led to a number of recent statistical unmixing algorithms that account for endmember variability.

1.4 Contributions and Organization of the Thesis

Continuous improvement in sensor technology is contributing to ever-increasing spectral sampling rates for imaging spectrometers. Unfortunately, algorithmic advances have not kept pace with technology, and statistical unmixing has become computationally intractable for recent hyperspectral and ultraspectral sensors. In particular, for N-band data, current statistical unmixing algorithms with realistic inter-band correlation models have a computational complexity of $\mathcal{O}(N^3)$. In this thesis, we develop and evaluate a novel statistical unmixing algorithm with complexity $\mathcal{O}(N)$.

In Chapter 2, a brief overview of different types of mixing models is given along with a summary of different classical unmixing algorithms and statistical unmixing algorithms.

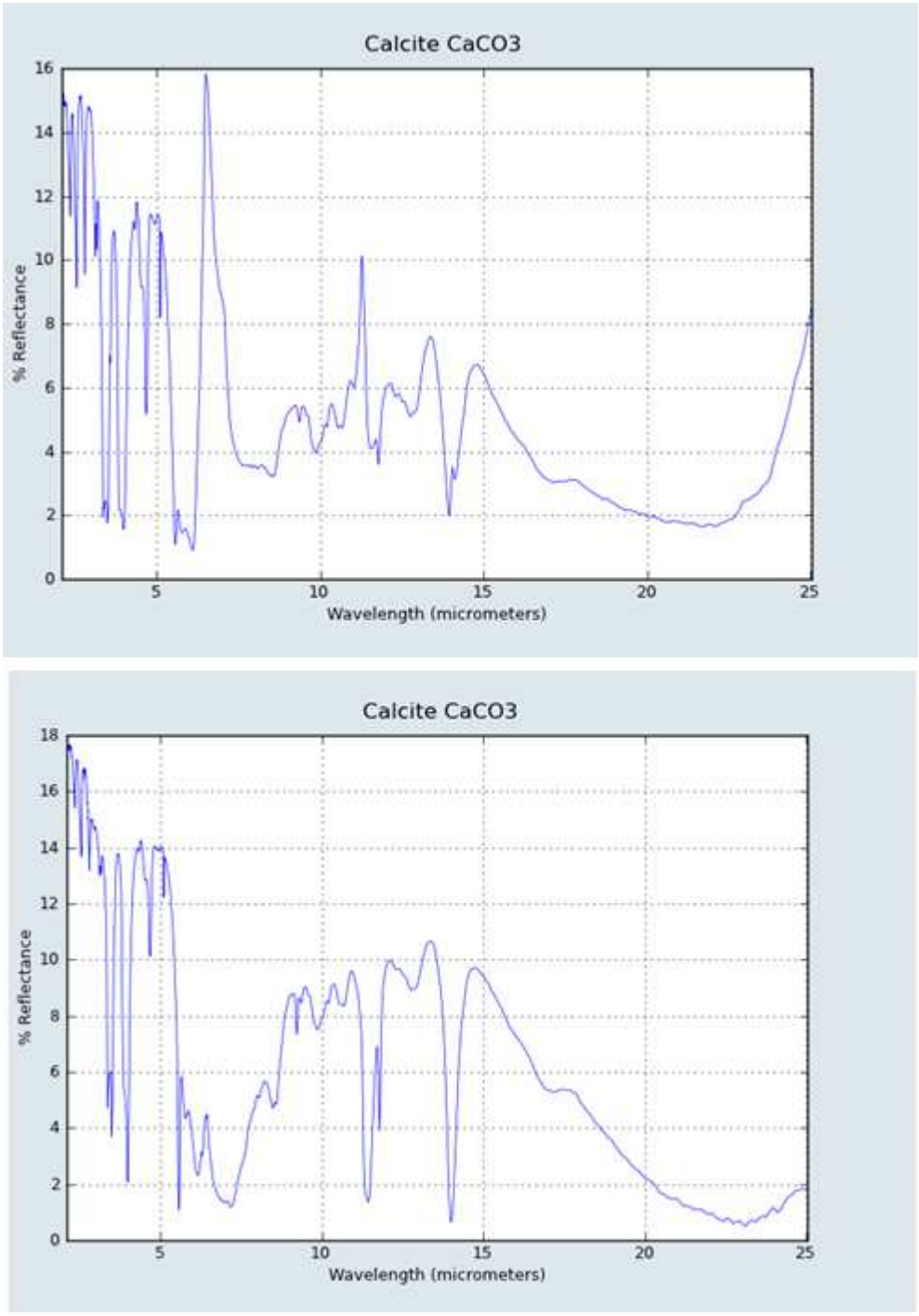

Figure 1.4: Endmember variability: two calcite samples exhibit different reflectance spectra. Source: ASTER spectral library [2].

In Chapter 3, we present our endmember variability model based on Markov chains and our accompanying unmixing algorithm that achieves linear complexity through the use of sum-product message passing. Chapter 4 compares results obtained by traditional unmixing and our proposed algorithm using data from the ASTER spectral library. We conclude in Chapter 5 with a summary of our algorithm's performance and suggestions for future research. Notation is summarized in the Appendix (6).

Chapter 2: Overview of Hyperspectral Unmixing algorithms

Over the past couple of decades there have been different types of spectral unmixing algorithms proposed, each based on different criteria. As discussed above, spectral unmixing involves estimating the spectral signatures of endmembers and their abundance values in a pixel. Endmember variability, measurement noise, modeling defects, environmental conditions and sensor inaccuracies make unmixing a challenging task. Unmixing, in plain terms, is the unwinding of the interaction of incident light with individual endmembers. In this Chapter, we review existing unmixing algorithms; however, first it is necessary to consider the forward model, i.e. spectral mixing.

2.1 Spectral Mixing

As light is reflected by M materials in a scene, the spectral measurement at each pixel can be modeled as mixture of the endmember spectra $\boldsymbol{X} = [\boldsymbol{x}^{(1)}, \ldots, \boldsymbol{x}^{(M)}]$ and their abundance ratios $\boldsymbol{\beta} = [\beta_1, \ldots, \beta_M]^T$. There are two types of mixing models: the Linear Mixing Model and non-linear mixing models[30].

2.1.1 The Linear Mixing Model

The linear mixing model (LMM) is an approximate model because of the primary assumption that, in a hyperspectral image the number of multiple scatterings of light among endmembers is negligible and may be neglected, i.e., on the ground there are not any conditions that cause light to be reflected from multiple endmembers. Under such conditions, the fractional abundances of the materials correspond to the composition of the measured spectra, which can be modeled as a linear combination of the endmembers [21]. This is referred as linear mixing model. This means that no endmember contains any impurity from surrounding endmembers which is unlikely in most of the data but LMM is perhaps the most widely used and accepted model for developing algorithms to solve the unmixing problem. The functional form of the LMM was given in (1.1) and repeated here for completeness:

$$\boldsymbol{y} = \sum_{m=1}^{M} \beta_m \boldsymbol{x}^{(m)} + \boldsymbol{w} = \boldsymbol{X}\boldsymbol{\beta} + \boldsymbol{w}. \tag{2.1}$$

Here, $\boldsymbol{y}$ is the $N \times 1$ observed pixel measurement resulting from M endmembers, $\boldsymbol{x}^{(m)}$ is the spectral signature of endmember m, β_m is the abundance ratio corresponding to endmember m, and $\boldsymbol{w}$ is background or instrument noise. The constraints on $\{\beta_m\}$ are

$$0 \leq \beta_m \leq 1, \quad \forall m \in \{1, \ldots, M\} \tag{2.2}$$

$$\sum_{m=1}^{M} \beta_m = 1 \tag{2.3}$$

These constraints signify that all the elements in the abundance vector should be non-negative, less than one, and should sum up to one—referred as sum-to-one constraint.

2.1.2 Nonlinear mixing model

The non-linear mixing model is based on the ideology that incident light reflects *multiple* times with in-scene materials, thus generating a spectral signature which has multiple scatterings involving different endmembers. The model is more complex than LMM and is potentially more accurate due to larger degrees of freedom. The general non-linear model may be expressed as

$$\boldsymbol{y} = f(\boldsymbol{X}, \boldsymbol{\beta}) + \boldsymbol{w}, \tag{2.4}$$

where f is a non-linear function, and $\boldsymbol{X}$ and $\boldsymbol{\beta}$ now potentially contain higher-order interaction elements. For example, a second order model was proposed by Nascimento and Bioucas-Dias [13] where heterogeneous endmembers are used to effect non-linear mixing and new abundance ratios are assigned for such heterogeneous endmembers. The model is similar to LMM, but has additional terms where each additional term represents the second degree interactions between two endmembers. A two-material example between soil and trees takes the form [13]

$$\boldsymbol{y} = [\boldsymbol{x}^{(soil)}, \boldsymbol{x}^{(tree)}, \boldsymbol{x}^{(soil,tree)}][\boldsymbol{\beta}_{soil}, \boldsymbol{\beta}_{tree}, \boldsymbol{\beta}_{soil,tree}]^T + \boldsymbol{w}. \tag{2.5}$$

Most unmixing literature focuses on the LMM, and we adopt the LMM for our work.

2.2 Classification of unmixing algorithms

There are different classifications for spectral unmixing algorithms. One of the well-known taxonomies of these algorithms is given by Bioucas-Dias [3] and includes the following unmixing algorithm categories: Geometrical unmixing, Statistical unmixing, Sparse regression unmixing, Signal subspace unmixing, and Spatial contextual unmixing. Here, we adopt a broader parsing of existing algorithms into two major categories:

1. Classical unmixing algorithms

2. Statistical unmixing algorithms

In the classical approaches, mixing of endmember spectra follows the LMM (2.1) and the endmembers are treated as deterministic unknown quantities. In the statistical approaches, the endmembers are treated as random quantities, typically following the Normal Composition Model (NCM) [8].

2.3 Classical Unmixing

The basic assumption in most of the classical unmixing algorithms is that, the spectral information can be modeled using LMM. Apart from LMM another assumption most of the classical unmixing algorithm make is the pure pixel assumption. In pure pixel based algorithms, the assumption is that, in the observed pixel measurements there would be at least one pixel measurement that would contain only a single endmember. The definition of a pure pixel is

$$\boldsymbol{y} = \boldsymbol{x}^{(m)}. \tag{2.6}$$

The definition of pure pixel defined above signifies that the observed pixel spectrum $\boldsymbol{y}$ is equal to a single endmember's spectral signature $\boldsymbol{x}^{(m)}$ and there would be M such distinct pure pixels, each containing a unique endmember spectra. Algorithms attempt to identify these pure pixels from the data to achieve unmixing. Some of the prominent algorithms in this category are the Successive Projection Algorithm (SPA), Vertex Component Analysis (VCA), N-FINDR and Successive Volume Maximization algorithms.

2.3.1 Geometrical analysis of Hyperspectral data

As most classical unmixing algorithms discussed in this Chapter use convex sets analysis on hyperspectral data, a brief introduction about some of the assumptions and terminology is discussed in this section. Application of convex geometry to hyperspectral unmixing was introduced by Craig [5]. The definition of a convex hull for endmember spectra $(\boldsymbol{x}^{(1)}, \ldots, \boldsymbol{x}^{(M)})$ is:

$$\text{conv}[\boldsymbol{x}^{(1)}, \ldots, \boldsymbol{x}^{(M)}] = \left[\boldsymbol{y} = \sum_{m=1}^{M} \theta_m \boldsymbol{x}^{(m)} \mid \boldsymbol{\theta} \geq 0, \sum_{m=1}^{M} \theta_m = 1 \right] \tag{2.7}$$

The above equation can be traced back to the noiseless case of the Linear Mixing Model, Eq.(2.1) [4]. The inference that can be drawn is that the observed data is a convex set of endmember spectra,i.e $\boldsymbol{y}[n] \in \text{conv}[\boldsymbol{x}^{(1)}, \ldots, \boldsymbol{x}^{(M)}], \forall n = 1, \ldots, L$ (L is the number of pixels in the data and "conv" is the convex hull). Unmixing based on convex geometry is based on estimating a set of vectors for endmember spectra, such that the estimated simplex of conv$[\hat{\boldsymbol{x}}^{(1)}, \ldots, \hat{\boldsymbol{x}}^{(M)}]$ gives the closest fit to the true endmembers simplex formed by the convex conv$[\boldsymbol{x}^{(1)}, \ldots, \boldsymbol{x}^{(M)}]$. An affine transformation on each endmember spectral vector $\boldsymbol{x}^{(m)}$ is chosen as the estimated convex conv$[\hat{\boldsymbol{x}}^{(1)}, \ldots, \hat{\boldsymbol{x}}^{(M)}]$. This would mean the presence of an affine transformation on the observed pixel spectra $\boldsymbol{y}[n]$ given as:

$$\boldsymbol{y}[n] = \boldsymbol{C}\boldsymbol{z}[n] + \boldsymbol{w} \tag{2.8}$$

Here $\boldsymbol{C} \in \mathbb{R}^{N \times (M-1)}$. The transformed measurement $\boldsymbol{z}[n]$ is formulated as:

$$\boldsymbol{z}[n] = \boldsymbol{C}^{\dagger}(\boldsymbol{y}[n] - \boldsymbol{w}) \tag{2.9}$$

where † denotes the Moore-Penrose pseudoinverse. Applying the above inference to Linear Mixing Model, $\boldsymbol{z}[n]$ can be written also as:

$$\boldsymbol{z} = \sum_{m=1}^{M} \boldsymbol{b}_m \beta_m \tag{2.10}$$

Here $\boldsymbol{b}_m = \boldsymbol{C}^\dagger(\boldsymbol{x}_m - \boldsymbol{w}) \in \mathbb{R}^{M-1}$ and the simplex $\boldsymbol{B}$ is defined on the convex hull of conv$[\boldsymbol{b}_1, \ldots, \boldsymbol{b}_M]$.

2.3.2 Successive Projection Algorithm

In SPA, the estimate $\hat{\boldsymbol{x}}^{(m)}$ of the spectral signature $\boldsymbol{x}^{(m)}$ is obtained using the orthogonal projection $P_m^\perp$corresponding to $\hat{\boldsymbol{x}}^{(m-1)}$. SPA assumes the pure pixel criteria and the objective of the algorithm is finding pure pixels for every endmember in the image. Here l_i ($i = 1, \ldots, M$) is used to index the pixels that are pure. The first endmember can be identified as

$$\hat{\boldsymbol{x}}^{(1)} = \boldsymbol{y}[\hat{l}_1], \tag{2.11}$$

where $\hat{l}_1$ is obtained using the equation

$$\hat{l}_1 = \arg \max_{n=1,\ldots,L} ||\boldsymbol{y}[n]||_2^2. \tag{2.12}$$

Here L is the number of pixels in the image. The estimated endmember spectra is assumed to be a perfect identification, i.e $\hat{\boldsymbol{x}}^{(m)} = \boldsymbol{x}^{(m)}$. Under this condition the other endmembers are identified using a geometric strategy termed as nulling. Nulling is the standard geometric operation of projecting an orthogonal component of the current vector. Using the estimated endmember vector $\hat{\boldsymbol{X}}^{(\boldsymbol{m-1})} = [\hat{\boldsymbol{x}}^{(1)}, \ldots, \hat{\boldsymbol{x}}^{(m-1)}]$, the orthogonal projection of $\hat{\boldsymbol{X}}^{(\boldsymbol{m-1})}$ is evaluated to estimate the next pure pixel index in the data. This process is repeatedly done to identify all the pure pixel indices from the spectral data that correspond

to individual endmember spectra. The algorithm is summarized below [30]:

$$
\begin{aligned}
&1: P^{\perp} = I \\
&2: \text{for } [m = 1, ...M] \\
&3: \quad \hat{l}_m = \arg \max_{n=1,...,L} ||P^{\perp}\boldsymbol{y}[n]||_2^2 \\
&4: \quad \hat{\boldsymbol{x}}^{(m)} = \boldsymbol{y}[\hat{l}_m] \\
&5: \quad P^{\perp} := \Big(\frac{(I - (P^{\perp}\hat{\boldsymbol{x}}^{(m)})(P^{\perp}\hat{\boldsymbol{x}}^{(m)})^T}{||P^{\perp}\hat{\boldsymbol{x}}^{(m)}||_2^2)}\Big)P^{\perp} \\
&6: \text{end for} \\
&7: \hat{\boldsymbol{X}} = [\hat{\boldsymbol{x}}^{(1)}, ..., \hat{\boldsymbol{x}}^{(M)}]
\end{aligned}
$$

2.3.3 Simplex Volume Maximization

In this section we discuss about two algorithms, N-FINDR & Successive Volume Maximization, that are based on the same principle: Simplex Volume Maximization . These algorithms assume LMM and pure pixel model to search for distinct pure pixels in the spectral data. A key point of these algorithms is that they make use of the fact that a pixel spectrum can be represented geometrically as a simplex and that the volume of such simplex is to be maximized by distinct pure pixels as vertices of the N-dimensional simplex. The objective function described in [30] is

$$
\max_{\boldsymbol{B}} \text{vol}(\boldsymbol{B}) \quad s.t. \quad \boldsymbol{b}_m \in \text{conv}\left[\boldsymbol{z}[1], \ldots, \boldsymbol{z}[L]\right], m = 1, ..., M \tag{2.13}
$$

In the above equation “vol” is the volume of the simplex $\boldsymbol{B}$. The values of $\boldsymbol{B}, \boldsymbol{b}_m$ & $\boldsymbol{z}[n]$ are obtained from the Section 2.3.1. Below are some algorithms that make use of simplex volume maximization.

N-FINDR

The N-FINDR algorithm is based on the geometric nature of spectral data. The objective function is a simplex containing spectra of observed pixels and the optimization is about maximizing the volume enclosed by this simplex such that the vertices of the simplex are distinct pure pixels which would represent individual endmember spectra. The algorithm starts with random selection of observed pixel data as vertices of the $M-1$ dimension simplex. The initial volume of the simplex formed using random pixels picked as pure pixels is calculated. On the next iteration, each pixel on the vertex is replaced by another pixel and the volume of the simplex is recalculated. If the volume increases then this change remains or else this is tested on other pixels. This iterative process is continued till the volume of the simplex converges to a maximum value. Once such maximum is reached then the vertices of the simplex are spectral signatures of individual endmembers. If there are no pure pixels for an endmember then the algorithm ends up estimating a mixed pixel spectra as endmember spectra for that endmember. Sometimes a local maxima could be encountered and the algorithm might stop prematurely before finding the closet match on endmember without pure pixel. To avoid such premature optimization, the algorithm is run several times with different initialization of the vertices that form the simplex [28].

The volume determination starts with augmenting the endmembers with a row of ones

$$\boldsymbol{E} = \begin{pmatrix} 1 & 1 & \ldots & 1 \\ \boldsymbol{x}^{(1)} & \boldsymbol{x}^{(2)} & \ldots & \boldsymbol{x}^{(M)} \end{pmatrix} \tag{2.14}$$

The volume of the simplex is determined using the formula

$$V(\boldsymbol{E}) = \frac{1}{(M-1)!} abs\big(|\boldsymbol{E}|\big), \tag{2.15}$$

where $M-1$ is the number of dimensions occupied by the data. After endmember determination & spectra estimation is done, the fractional abundances of the corresponding endmembers are evaluated. This procedure is done using least squares inversion or non-negative constrained inversion. The solution [14] follows as

$$\hat{\boldsymbol{\beta}} = (\boldsymbol{E}^T \boldsymbol{E})^{-1} \boldsymbol{E}^T \boldsymbol{y}. \tag{2.16}$$

This is done iteratively on each pixel to estimate endmember spectra and their abundance ratios.

Successive Volume Maximization

This algorithm is based on the same objective as discussed in N-FINDR, where endmember uncertainties are found by maximizing the volume of the simplex enclosed by observed spectral data. This algorithm [29] uses a similar idea as SPA, where the solution to estimating endmember spectra $\hat{\boldsymbol{x}}^{(m)}$ is given by Eq.(2.12). The algorithm starts with dimension reduction on the pixel spectra and then calculating a matrix $\boldsymbol{F}$ as:

$$\boldsymbol{F} = \begin{pmatrix} \boldsymbol{b}_1 & \boldsymbol{b}_2 & \dots & \boldsymbol{b}_M \end{pmatrix} \tag{2.17}$$

$$\boldsymbol{f}_m = \begin{pmatrix} \boldsymbol{b}_m \\ 1 \end{pmatrix}, \tag{2.18}$$

$$\overline{\boldsymbol{z}} = \begin{pmatrix} \boldsymbol{z} \\ 1 \end{pmatrix} \tag{2.19}$$

The algorithm projects orthogonal projections from estimated endmember vector successively every time to find each endmember spectra (similar to SPA) but instead of this operation being done directly on $\hat{\boldsymbol{X}}$, here it is done on a matrix with a vector of ones augmented

to $\boldsymbol{X}$ (Eq.(2.17)). From [29]

$$|det(\boldsymbol{F})|^2 = \prod_{m=1}^{M} \left\| \boldsymbol{P}^{\perp}_{F1:m-1} f_m \right\|_2^2 \tag{2.20}$$

Maximizing the volume of the simplex Eq.(2.13) is same as maximizing $\boldsymbol{F}$ Eq.(2.17) [29][30]. Successive volume maximization, (SVMAX) uses the successive structure of Eq.(2.20) to generate an estimation of endmember spectra recursively. An endmember $\hat{b}_m$ is generated by the knowledge of its previous estimates $\hat{b}_1, ..., \hat{b}_{m-1}$. The solution for the endmember spectral signature is

$$\hat{\boldsymbol{b}}_m = \boldsymbol{z}[\hat{l}_m], \tag{2.21}$$

$$\hat{l}_m = \arg \max_{n=1,...L} \left\| \mathbf{P}^{\perp}_{\boldsymbol{F}1:m-1} \overline{\boldsymbol{z}}[n] \right\|_2^2 \tag{2.22}$$

A detailed description and complexity assessment of the algorithm may be found in [29].

2.3.4 Simplex Volume Minimization

Simplex volume minimization is an algorithm that finds a simplex with the least volume enclosing all the pixel measurements. This is an improved version compared to the results obtained using SVMAX or N-FINDR as it performs better even if there are not any pure pixels in the measured data, thus giving more flexibility to the model to fit real data. The objective function varies from SVMAX as shown:

$$\min_{\boldsymbol{B}} \text{ vol}(\boldsymbol{B}) \; s.t. \boldsymbol{z}[n] \in \text{conv}[\boldsymbol{b}_1, \ldots, \boldsymbol{b}_M] \tag{2.23}$$

This objective function does not have a closed form scheme like successive volume maximization and has to be solved empirically, which makes the optimization more complex

to handle. Because of the constraints on the abundance values, this is a non-convex optimization. Additional details describing the optimization routine and of the algorithm may be found in [30][31].

2.3.5 Vertex Component Analysis

Vertex component analysis (VCA) was proposed to be an improvement over other existing algorithms in the classical unmixing category by Bioucas Dias and Nascimento [19]. The VCA sets out in the determination of endmember spectral signatures using the pixel spectrum obtained through a hyperspectral sensor. VCA assumes the pixel spectra can be modeled by LMM and assumes pure pixel model. VCA makes use of two facts, first is that the endmember space can be represented as a simplex whose vertices represent endmembers present in the observed data and second is that an affine transformation applied to a simplex results in a simplex. The algorithm starts from the inference that the abundance vector $\boldsymbol{\beta}$ can be represented as a simplex, then the geometry needed to represent the spectral signatures $\boldsymbol{x}$ over N bands of spectrum is also a simplex represented as S_x and the observed pixel spectrum is represented as a convex cone, C_p. The orthogonal projection of C_p on to a hyperplane results in a simplex that has vertices corresponding to S_x i.e S_p. Once S_p is determined, then the VCA algorithm functions similar to SPA where it iteratively projects data orthogonally to the subspace spanned by the endmembers already determined, and this iteration continues until all the endmembers are searched for.

The VCA algorithm has simpler computational complexity because of usage of dimension reduction techniques such as Principle Component Analysis (PCA), where the number of spectral bands would be reduced from N to P where $P \ll N$. For this reason, VCA has better processing and computational speed and is considered to be one of the best algorithms in unsupervised hyperspectral unmixing.

2.4 Statistical Unmixing

2.4.1 Normal Compositional Model

The Normal Composition model (NCM) is a popular mixing model because it addresses both endmember variability and sub-pixel mixing. The NCM applies a normal distribution on endmember reflectances to model endmember variability among different samples of an endmember along different wavelengths. Mathematically, the observation model follows the LMM

$$\boldsymbol{y} = \sum_{m=1}^{M} \beta_m \boldsymbol{x}^{(m)} + \boldsymbol{w}, \tag{2.24}$$

where the elements of $\boldsymbol{w}$ are additive white Gaussian noise with variance σ_w^2. However, now each endmember $\boldsymbol{x}^{(m)}$ is assumed to have an *a priori* Gaussian distribution

$$\boldsymbol{x}^{(m)} \sim \mathcal{N}(\boldsymbol{\mu}^{(m)}, \boldsymbol{C}^{(m)}) \tag{2.25}$$

The constraints on the abundances β_m are the same as the LMM: $\sum_{m=1}^{M} \beta_m = 1, 0 \leq \beta_m \leq 1$. The above set of equations signify that $\boldsymbol{x}^{(m)}$ is a random vector representing the normal distribution of an endmember m with a mean vector $\boldsymbol{\mu}^{(m)}$ & covariance matrix $\boldsymbol{C}^{(m)}$.

Most of the statistical unmixing algorithms use the NCM, and proceed by estimating the NCM parameters. These are estimated by using stochastic expectation maximization algorithm. For further explanation on the derivation of these parameters refer [26]. It can also be inferred from the NCM that the pixel measurement values $\boldsymbol{y}$ are also distributed normally

$$\boldsymbol{y} \sim \mathcal{N}(\boldsymbol{\mu}_y, \boldsymbol{C}_y) \tag{2.26}$$

with mean $\boldsymbol{\mu}_y$ and covariance matrix $\boldsymbol{C}_y$, given as

$$\boldsymbol{\mu}_y = \sum_{m=1}^{M} \beta_m \boldsymbol{\mu}^{(m)} \tag{2.27}$$

$$\boldsymbol{C}_y = \sum_{m=1}^{M} \beta_m^2 \boldsymbol{C}^{(m)} + \sigma_w^2 \boldsymbol{I}_M. \tag{2.28}$$

Prior knowledge of endmember reflectances are assumed to be available through a spectral library.

2.4.2 Brute Force Algorithm for Unmixing

One of the most direct approaches to statistical unmixing is to apply the maximum likelihood principle to the NCM. As mentioned in the previous section, by using a spectral database the mean $\boldsymbol{\mu}^{(m)}$ and covariance $\boldsymbol{C}^{(m)}$ of endmembers may be determined, and the resulting Gaussian parameters of the pixel measurements derived as in (2.27) and (2.28). The likelihood $p(\boldsymbol{y};\boldsymbol{\beta})$ of the abundance ratios $\boldsymbol{\beta}$ then has the form

$$p(\boldsymbol{y};\boldsymbol{\beta}) = (2\pi)^{-N/2}(det(\boldsymbol{C}_y(\boldsymbol{\beta}))^{-1/2}\, exp\Big[-\frac{1}{2}(\boldsymbol{y}-\boldsymbol{\mu}_y(\boldsymbol{\beta}))^T \boldsymbol{C}_y^{-1}(\boldsymbol{\beta})(\boldsymbol{y}-\boldsymbol{\mu}_y(\boldsymbol{\beta}))\Big], \tag{2.29}$$

where we now write $\boldsymbol{\mu}_y(\boldsymbol{\beta})$ and $\boldsymbol{C}_y(\boldsymbol{\beta})$ to explicitly indicate the dependence on $\boldsymbol{\beta}$. For a given pixel measurement $\boldsymbol{y}$, we may compute the maximum likelihood abundances as

$$\hat{\boldsymbol{\beta}} = \arg\max_{\boldsymbol{\beta}} p(\boldsymbol{y};\boldsymbol{\beta}). \tag{2.30}$$

However, maximizing the likelihood is same as minimizing the negative of log-likelihood, hence we minimize the negative log-likelihood to estimate the abundance vector $\boldsymbol{\beta}$,

$$\hat{\boldsymbol{\beta}} = \arg\min_{\boldsymbol{\beta}} \{-\ln p(\boldsymbol{y}; \boldsymbol{\beta})\}, \tag{2.31}$$

with

$$\ln p(\boldsymbol{y}; \boldsymbol{\beta}) = -\frac{N}{2}\ln(2\pi) - \frac{1}{2} det(\ln \boldsymbol{C}_y(\boldsymbol{\beta})) - \frac{1}{2}(\boldsymbol{y} - \boldsymbol{\mu}_y(\boldsymbol{\beta}))^T \boldsymbol{C}_y^{-1}(\boldsymbol{\beta})(\boldsymbol{y} - \boldsymbol{\mu}_y(\boldsymbol{\beta})). \tag{2.32}$$

This is a straightforward method for unmixing endmembers that supports arbitrary covariance structures in the NCM model, however when the spectral resolution is increased, the computational time required to unmix increases dramatically. Inspecting (2.32), we see that this is due to the need to recompute the determinant and inverse of $\boldsymbol{C}_y(\boldsymbol{\beta})$ for each candidate argument $\boldsymbol{\beta}$ of the objective function. If the measurement $\boldsymbol{y}$ contains N spectral bands, the complexity of each of these operations is $\mathcal{O}(N^3)$.

2.4.3 Bayesian Estimation of linear mixtures using NCM

The hierarchical Bayesian model proposed by Eches [22] uses the NCM model to treat pixel measurements as linear combination of endmembers with endmembers represented as normal distributions. This model doesn't use a spectral library to estimate endmember means and covariances, instead uses one of the classical unmixing algorithms like VCA or NFINDR to extract endmembers from observed spectral measurements and estimates corresponding mean vectors for each endmember. The covariance matrix for an endmember is assumed to be proportional to an identity matrix

$$\boldsymbol{C}^{(m)} = \sigma^2_{(m)} \boldsymbol{I}_N \tag{2.33}$$

The endmember variance $\sigma^2_{(m)}$ is assigned a conjugate inverse gamma distribution with suitable priors for the hyper-parameters of the conjugate inverse gamma distribution. The Bayesian approach is used to estimate abundance values of endmembers. The likelihood of the observed pixel $\boldsymbol{y}$ is

$$f(\boldsymbol{y}|\boldsymbol{\beta}, \sigma^2_{(m)}) = \frac{1}{[2\pi\sigma^2_{(m)}c(\boldsymbol{\beta})]^{N/2}} \cdot exp\Big[-\frac{\|\boldsymbol{y} - \boldsymbol{\mu}_y(\boldsymbol{\beta})\|^2}{2\sigma^2 c(\boldsymbol{\beta})}\Big]. \tag{2.34}$$

The value of $\boldsymbol{\mu}_y(\boldsymbol{\beta})$ is same as shown in (2.27) and $c(\boldsymbol{\beta}) = \sum_{m=1}^{M} \beta^2_{(m)}$. As the empirical evaluation of posterior is too complex to derive a MMSE or MAP estimate, the posterior is evaluated by EM algorithm but to limit the shortcomings of the EM algorithm Markov chain Monte Carlo (MCMC) simulations are done to iteratively generate samples of the joint posterior of the abundances and endmember variance. A uniform distribution is chosen as the prior for abundance values. The sampling algorithm used is the Metropolis-within-Gibbs sampler that would generate samples of the posterior function. An improvement is also made to this model where the endmember covariance is a diagonal matrix,i.e a different variance along the diagonal (This is discussed in Section 3.1).

The drawback of this model is that the endmember covariance is proportional to an identity matrix and does not account for spectral dependencies between wavelengths. Apart from this, the model uses Monte Carlo simulations to derive to optimum values which contributes to slow runtime.

2.4.4 Sampling piecewise convex unmixing and endmember extraction

S-PCUE was proposed by Zare et al. [32] and the algorithm searches for sets of endmembers in non-convex data sets and uses a distribution on endmembers to model endmember

variability. Through several sets of endmember distributions a piecewise convex model is built and the algorithm performs estimation of endmember uncertainties. After extraction of endmembers, the mean vector is evaluated for each endmember. In this method the covariance of the endmember is assumed to be proportional to an identity matrix.

The S-PCUE algorithm uses a Metropolis-within-Gibbs sampler to divide the observed non-convex data into convex sets, sample the convex sets, estimate the endmember distributions in each convex set, i.e its mean vector and estimates corresponding abundance values for each endmember. As this algorithm uses a sampling approach the algorithm is iterated approximately 50000 times to achieve convergence. The results from S-PCUE were found to be better than VCA on simulated data and AVIRIS Indian Pines data-set. The drawback of this approach is that the endmember covariance matrix is modeled to be fixed and proportional to an identity matrix. Another drawback of this algorithm is that it relies on Monte Carlo simulations for estimation which contribute to slow unmixing times.

Chapter 3: Accelerated Unmixing via the Sum-Product Algorithm

3.1 Probabilistic Graphical Model for Unmixing

The performance of statistical unmixing algorithms is governed by the assumptions made in their underlying models. However, to combat computational complexity, most existing algorithms assume that the *a priori* distribution of endmember reflectance values are independent at different wavelengths. For example, the S-PCUE algorithm [32] assumes the NCM for each endmember $\boldsymbol{x}^{(m)}$, $p(\boldsymbol{x}^{(m)}) \sim \mathcal{N}(\boldsymbol{\mu}^{(m)}, \boldsymbol{C}^{(m)})$; but, this model model constraints the covariance matrix to be proportional to an identity matrix

$$\boldsymbol{C}^{(m)} = \sigma^2_{(m)} \boldsymbol{I} \tag{3.1}$$

Such a covariance matrix does not model the correlations between wavelengths and assumes same variance $\sigma^2_{(m)}$ for all the samples. Similarly, the hierarchical Bayesian model

proposed by Eches [22] uses the NCM with a diagonal covariance matrix

$$\boldsymbol{C}^{(m)} = \begin{bmatrix} \sigma^2_{(m)}(1) & & & \\ & \sigma^2_{(m)}(2) & & \\ & & \ddots & \\ & & & \sigma^2_{(m)}(N) \end{bmatrix}. \tag{3.2}$$

Although this captures different reflectance variances among the bands, correlation properties are still not modeled although correlated values are very prevalent in actual endmember spectra. For example, the calcite spectra depicted in Fig. 1.4 show significant smoothness over many of the bands, suggesting strong correlation properties between the reflectances of adjacent wavelengths.

Although the brute force algorithm presented in Sec. 2.4.2 does support arbitrary covariance structures, the associated computational complexity is intractable for large spectral sampling rates. Therefore, we seek an alternative model that is both computationally tractable and that captures the correlation properties of endmembers. As described in the following section, we propose a Gaussian Markov chain for this purpose.

3.1.1 Markov Chain Model for Endmember Variability

In the past, probabilistic graphical models like the Gauss Markov random fields (GMRF) have been used on multispectral and hyperspectral images to model spatial and spectral dependencies for an endmember [25]. An adaptation of these concepts is used where a Markov chain is used in the spectral dimension to model the endmember variability. Endmember reflectances at each wavelength are modeled as random variables in a Markov chain and the conditional transition probabilities from one random variable to another induce the desired correlation properties. In our model we use a first order Markov chain such that the conditional probability of the reflectance x_{i+1} at wavelength λ_{i+1} only de-

pends only on the previous reflectance x_i

$$p(x_{i+1}|x_1, x_2, \ldots, x_i, x_{i+2}, \ldots, x_N) = p(x_{i+1}|x_i), \tag{3.3}$$

and with the probability of the entire endmember/chain given as

$$p(\boldsymbol{x}) = p(x_1) \prod_{i=1}^{N-1} p(x_{i+1}|x_i). \tag{3.4}$$

Adopting a first-order *Gaussian* model, we have

$$p(x_{i+1}|x_i) = \mathcal{N}[x_{i+1}; \alpha_i x_i + \mu_i, \sigma_i^2], \tag{3.5}$$

with the following designations

- x_i: random variable to representing the unknown reflectance of the ith element of an endmember, corresponding to a wavelength λ_i
- $p(x_{i+1}|x_i)$: the conditional (transitional) probability between x_i and x_{i+1}
- α_i: the correlation factor between x_i and x_{i+1}
- μ_i: mean of transition noise
- σ_i^2: variance of transition noise
- $p(x_1)$: Gaussian prior probability with mean μ_0 and variance σ_0^2

From above it is evident that other parameters like the correlation factor α_i and noise parameters μ_i, σ_i^2 and parameters of prior probability $\big(p(x_1)\big)$ μ_0, σ_0^2 are needed to form the probability distribution $p(\boldsymbol{x})$. The totality of these parameters, i.e $\{\mu_0, \sigma_0^2\}$, $\{\alpha_i, \mu_i, \sigma_i^2\}_{i=1}^{N-1}$ across the chain, define the variability of the associated endmember and are assumed to have been estimated offline using a spectral database, such as the ASTER

spectral library.

As illustrated in Fig. 3.1, the Markov chain may be graphically depicted using a *factor graph*, where circles represent random variables (in this case unknown reflectances of an endmember) and squares represent factors (conditional probabilities in this case) in a joint PDF over all of the variables considered. Edges in the graph connect factors to the variables that they depend on. For example, for factor i, we have $f_i = p(x_{i+1}|x_i)$ with edges connecting to x_i and x_{i+1}.

Figure 3.1: A Factor graph depicition of a Markov chain representing the random variability of a single endmember

3.1.2 Graphical model for unmixing

In this section we develop a graphical model representation for spectral measurements with endmember variability. This model will be used for unmixing and is formed as a combination of the linear mixing model and the Markov chains introduced above.

A typical hyperspectral pixel would generally contain more than a single endmember. As such, we employ one Markov chain for each endmember. If there are M endmembers, then we use M Markov chains to model all the endmembers in a pixel. We supplement known information like the correlation factor $\boldsymbol{\alpha}$ and pixel observations y_i into these chains. In Fig. 3.2, we shown an example of this model for the case $M = 3$. Here the variable node $x_i^{(m)}$ represents the reflectance at wavelength i for endmember m. $f_i^{(m)}$ is the factor node between spectral bands $\{i, i+1\}$ for endmember m, containing the factor function $p(x_{i+1}^{(m)}|x_i^{(m)})$ as shown in Figure 3.2. $\alpha_i^{(m)}$ is the correlation factor at factor node $f_i^{(m)}$, modeling the spectral correlation between spectral bands $\{i, i+1\}$ with $\mu_i^{(m)}$ and $\sigma_i^{(m)^2}$ being the mean and variance of transition noise at factor node $f_i^{(m)}$. The factor $\phi_i =$

$p(y_i|\ [x_i^{(1)}, \ldots, x_i^{(M)}], \boldsymbol{\beta})$ represents the distribution of the measurement at wavelength i. We follow the linear mixing model and, as in the NCM [26], assume that measurement noise is Gaussian, i.e.

$$y_i = \sum_{m=1}^{M} \beta_m x_i^{(m)} + \epsilon_i \tag{3.6}$$

where ϵ_i represents additive white Gaussian noise with precision p_ϵ. As such, we have

$$\phi_i = p\big(y_i|\ [x_i^{(1)}, \ldots, x_i^{(M)}], \boldsymbol{\beta}\big) = p(y_i|\boldsymbol{x}_i, \boldsymbol{\beta}) = \bar{\mathcal{N}}(y_i; \boldsymbol{\beta}^T \boldsymbol{x}_i, p_\epsilon) \tag{3.7}$$

for measurement factor ϕ_i, computed for each wavelength index $i = 1, \ldots, N$.

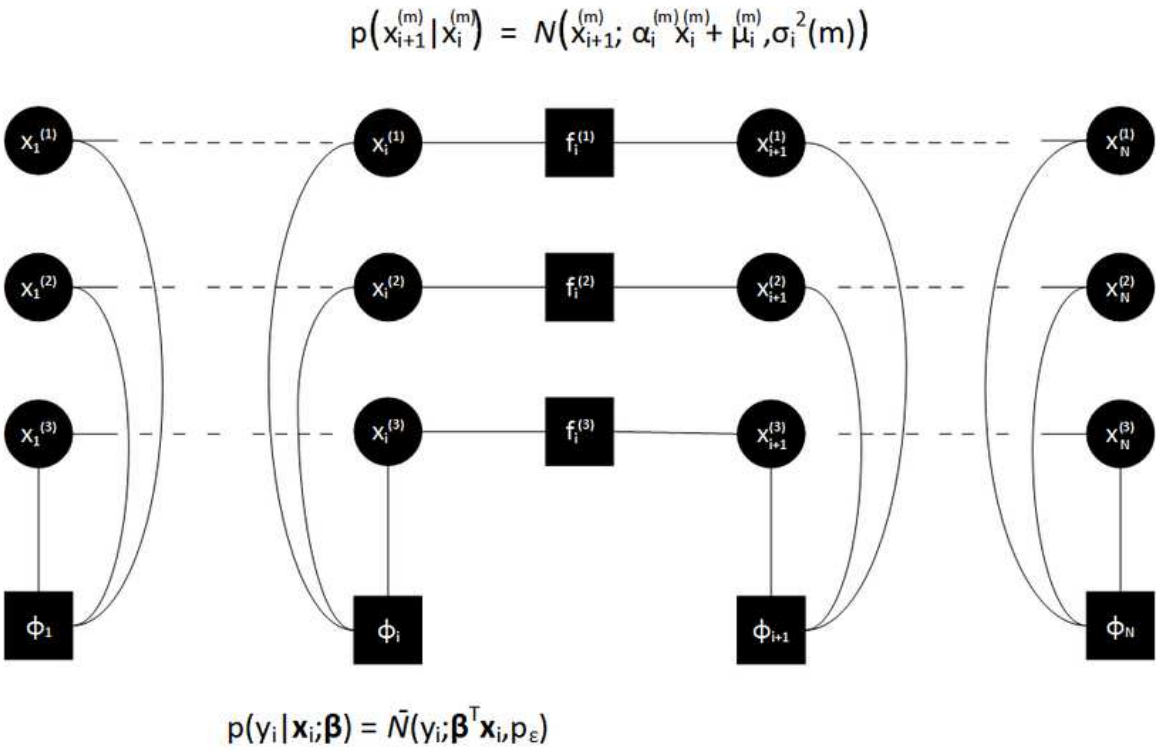

Figure 3.2: A factor graph depiction of the spectral measurement system consists of multiple interacting Markov chains, one for each endmember. $M = 3$ endmembers are shown in this example.

3.1.3 Likelihood of abundance ratios

The likelihood of the measurement model, depicted in Fig. 3.2, depends on the priors $p(x_1^{(m)})$, the conditional probabilities $p(x_{i+1}^{(m)}|x_i^{(m)})$, and $p(y_i|\boldsymbol{x_i};\boldsymbol{\beta})$, i.e the observed pixel measurements. We formulate the general likelihood for a pixel having M endmembers with N spectral bands in it. This is shown below:

$$p(\boldsymbol{y},\boldsymbol{X};\boldsymbol{\beta}) = \left(\prod_{m=1}^{M} p(x_1^{(m)}) \prod_{i=1}^{N-1} p(x_{i+1}^{(m)}|x_i^{(m)})\right) \cdot \left(\prod_{i=1}^{N} p(y_i|\boldsymbol{x_i};\boldsymbol{\beta})\right) \tag{3.8}$$

where $\boldsymbol{X} = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N\}$ includes all $N \times M$ unknown reflectances. The above equation is a joint likelihood on reflectance and abundance values. As our algorithm is a supervised learning algorithm we estimate only the abundance values in a pixel and there by detect if a particular endmember is present or not (i.e if $\beta_m > 0$, then m^{th} endmember is present in the pixel). Marginalization of the Eq.(3.8) is done to convert the joint likelihood equation to a likelihood equation on abundance values. This shown below

$$p(\boldsymbol{y};\boldsymbol{\beta}) = \int p(\boldsymbol{y};\boldsymbol{X},\boldsymbol{\beta})\, d\boldsymbol{X} \tag{3.9}$$

$$= \int \left(\prod_{m=1}^{M} p(x_1^{(m)}) \prod_{i=1}^{N-1} p(x_{i+1}^{(m)}|x_i^{(m)})\right) \cdot \left(\prod_{i=1}^{N} p(y_i|\boldsymbol{x_i};\boldsymbol{\beta})\right) d\boldsymbol{X}, \tag{3.10}$$

While the above integral appears intractable (of size $N \times M$), we show below that efficient integration strategies are available using the sum-product algorithm.

3.2 The Sum-Product Algorithm

Contemporary applications often require large models complex systems involving many hidden variables and uncertain parameters. One example is weather forecasting, which typically uses prior information of previous weather history, temporal dependencies and

other information to predict the weather conditions. These type of problems are frequently tackled using probabilistic graphical models and one of the well-known algorithms that fuses information within the graphical models is the sum-product algorithm [10] [16] [18]. The sum-product algorithm computes inferences on graphical models like Markov random fields and Bayesian networks by passing messages along edges in the graph. These inferences can be discreet or continuous in nature. Generally, the sum-product algorithm is used because of its efficiency and exactness in finding the marginal probability of random variables. Messages are passed both in forward and backward direction and the individual marginals are calculated by accumulating the forward and backward messages coming in to a variable node. In our algorithm, we make use of only forward going messages computed by sum-product algorithm. All the forward going messages are multiplied with local functions of variable and factor nodes. The final message at the end of the chain is the likelihood of the chain. The sum-product algorithm is generally known to be an exact algorithm on tree-structure graphs, but is only approximate for graphs with cycles.

Below we provide a general description of the sum-product algorithm, describing how messages are computed, updated, and fused. In Sec. 3.3 we specify these general rules to the unmixing problem.

3.2.1 Formulation of messages

Because there are two types of nodes in a factor graph, the sum-product algorithm consists of two types of messages:

1. Messages from variable nodes to factor nodes

2. Messages from factor nodes to variable nodes

The messages going from a factor node to a variable node are represented as m_{fx} and messages from variable node to factor node are represented as m_{xf}. Factor nodes are

represented as f and variable nodes are represented as x.

Message from variable node to factor node:

Variable nodes fuse the information of their local function with incoming messages from neighboring factor nodes. In this presentation, we assume no local function for the variable nodes. Using message passing update rules, the output message of a variable node is the product of the input messages from all neighboring factor nodes except the destination factor [9] [16]

$$m_{xf}(x) = \prod_{f' \in N(x) \setminus f} m_{f'x}(x), \tag{3.11}$$

where $N(x) \setminus f$ denotes the set of all neighboring factors of variable x, except f. For the example graph in Fig. 3.3, the outgoing message to factor f_g is

$$m_{x_t f_g}(x_t) = m_{f_j x_t}(x_t)\, m_{f_k x_t}(x_t) \tag{3.12}$$

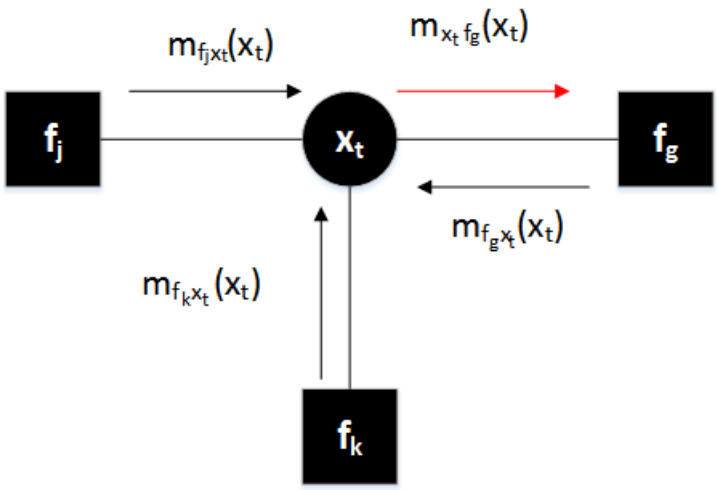

Figure 3.3: Message going from variable node to factor node

Message from factor node to variable node:

The second type of message originates from factor nodes and has variable nodes as its destination. Factor-to-variable messages transform incomming messages about neighboring variables into outgoing information describing the destination node. The outgoing message from factor f to variable x takes the form [16] [9]

$$m_{fx}(x) = \int_{X^-} f(X) \prod_{x' \in X^-} m_{x'f}(x') \, \mathrm{d}X^-, \tag{3.13}$$

where $X = N(f)$ denotes the set of neighboring variables of f in the factor graph, and $X^- = X \setminus x$ excludes the destination variable from this set.

A three-variable example is shown in Fig. 3.4. The factor node f_s here is a compatibility function governing the relation between $\{x_t, x_q, x_g\}$ and has its own local function (f_s) defined by some prior knowledge about $\{x_t, x_q, x_g\}$. The output message is the product of non-destination input messages from variable nodes neighboring factor f_s. As the product of all the messages is a multivariate function, integration is applied on all variables other than x_g. In this case integration is done on x_t and x_q, that is, the multivariate function is marginalized such that the output function only depends on x_g

$$m_{f_s x_g}(x_g) = \int_{x_t} \int_{x_q} m_{x_t f_s}(x_t) \, m_{x_q f_s}(x_q) \, f_s(x_t, x_q, x_g) \, \mathrm{d}x_t \, \mathrm{d}x_q \tag{3.14}$$

Marginalization:

Lastly, the factor-to-variable messages can be used to compute marginal distributions over the graph. For an a cyclic graph, if the outgoing messages have been computed for each node in the network in a round-robin fashion for at least d times, where d is the diameter

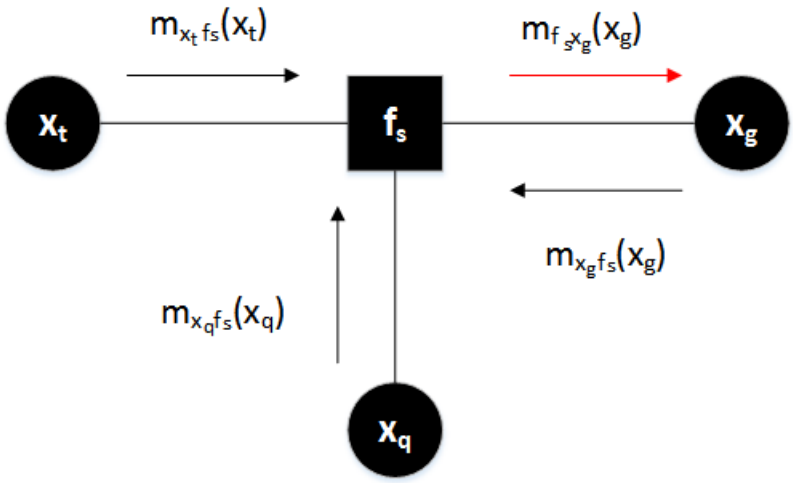

Figure 3.4: Message going from factor node to variable node

of the graph, then the marginal distribution of a variable x may be exactly computed as

$$p(x) = \prod_{f \in N(x)} m_{fx}(x). \tag{3.15}$$

In the unmixing problem, from (3.8), we have $p(\boldsymbol{y}, \boldsymbol{X}; \boldsymbol{\beta})$ and wish to marginalize all of the reflectances, as in (3.10). Applying the sum-product algorithm, we may use (3.15) to compute the marginal distribution over all but reflectances at the N^{th} wavelength, i.e $\boldsymbol{x}_N$. This yields $p(\boldsymbol{y}, \boldsymbol{x}_N; \boldsymbol{\beta})$ from which the remaining integral over $\boldsymbol{x}_N$ yields the desired marginalized likelihood $p(\boldsymbol{y}; \boldsymbol{\beta})$.

3.3 Sum-product unmixing

Figure 3.2 illustrates the factor graph governing Markov chain endmember variability and linear mixing for M endmembers and N spectral bands. The drawback of this model is that it is loopy, and as discussed earlier, the sum-product algorithm on graphs is only exact for graphs without cycles. To address this issue, we retune our model and combine the M scalar Markov chains into a single vector-valued Markov chain, as shown in Figure 3.5. This is

generally termed as clustering of nodes which is done essentially to eliminate any cycles present in the graph and this does not affect the computational complexity to calculate messages [16]. The sum-product algorithm is now applied to the reformulated graphical model where the messages are now multivariate functions. In the following section, we derive the messages required for efficient unmixing.

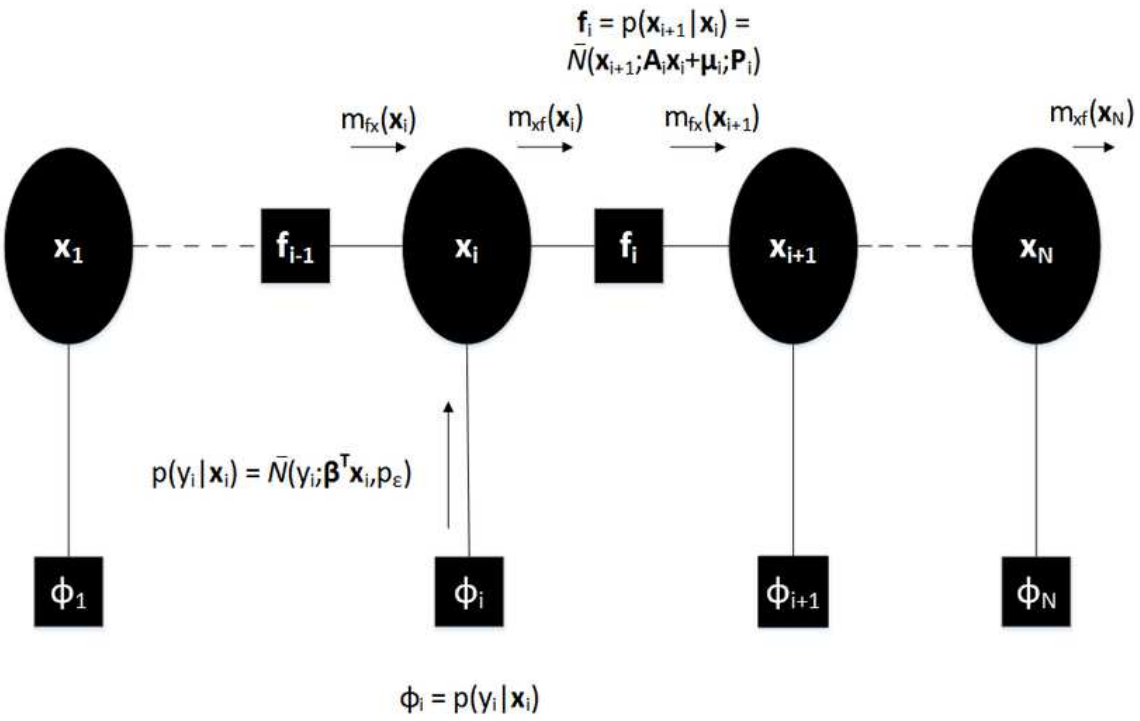

Figure 3.5: Clustered Markov chain for multiple endmembers with N spectral bands

3.3.1 Derivation of Unmixing Messages

In this section, we derive the necessary unmixing messages: $m_{xf}(\boldsymbol{x}_i)$ (message going from a variable node to factor node) and $m_{fx}(\boldsymbol{x}_{i+1})$ (message going from factor node to variable node) and show that these are each scaled Gaussian functions, parameterized by a scale factor, mean vector, and covariance matrix.

We use the following notation in the message derivations below:

$\boldsymbol{x}_i$: vector representation of the reflectance at wavelength i across all M endmembers; $\boldsymbol{x}_i = [x_i^{(1)}, \ldots, x_i^{(M)}]^T$

y_i : observed pixel measurement value at the i^{th} wavelength

$\boldsymbol{\beta}$: vector containing the abundance ratios of all endmembers; $\boldsymbol{\beta} = [\beta_1, \ldots, \beta_M]^T$

$\alpha_i^{(m)}$: correlation factor between $x_i^{(m)}$ and $x_{i+1}^{(m)}$ of the m^{th} endmember

$\boldsymbol{A}_i$: matrix containing correlation factors between $\boldsymbol{x}_i$ and $\boldsymbol{x}_{i+1}$ $\boldsymbol{A}_i = \text{diag}([\alpha_i^{(1)}, \ldots, \alpha_i^{(m)}, \ldots, \alpha_i^{(M)}])$

Message going from variable node to factor node

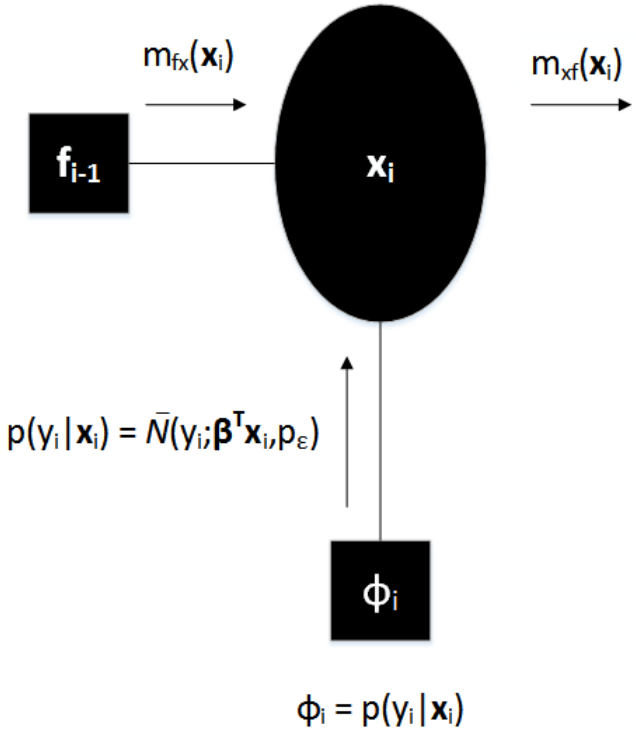

Figure 3.6: Evaluation of variable-to-factor vector message for unmixing

Figure 3.6 depicts a portion of the graph needed to evalute the message from a variable node $\boldsymbol{x}_i$ to factor f_i. The message $m_{fx}(\boldsymbol{x}_i)$ entering node $\boldsymbol{x}_i$ in Figure 3.6 is the message

from node $\boldsymbol{f}_{i-1}$ which is a probability distribution on $\boldsymbol{x}_i$

$$m_{fx}(\boldsymbol{x}_i) \propto \bar{\mathcal{N}}(\boldsymbol{x}_i; \boldsymbol{m}_{\mu}^{(i)}, \boldsymbol{m}_{P}^{(i)}), \tag{3.16}$$

where $\boldsymbol{m}_{\mu}^{(i)}, \boldsymbol{m}_{P}^{(i)}$ are the mean and inverse covariance (precision matrix) of $m_{fx}(\boldsymbol{x}_i)$. The second message is from the factor node ϕ_i and the factor function of this node is $p(y_i|\boldsymbol{x}_i)$ which is the observed pixel measurement. From Figure 3.6, we see that 2 messages act as input to node $\boldsymbol{x}_i$, the outgoing message of variable node $\boldsymbol{x}_i$ is computed using the sum-product rule given by Eq.(3.12):

$$m_{xf}(\boldsymbol{x}_i) = m_{fx}(\boldsymbol{x}_i) p(y_i|\boldsymbol{x}_i) \tag{3.17}$$

$$= \bar{\mathcal{N}}(\boldsymbol{x}_i; \boldsymbol{m}_{\mu}^{(i)}, \boldsymbol{m}_{P}^{(i)}) \bar{\mathcal{N}}(y_i; \boldsymbol{\beta}^T \boldsymbol{x}_i, p_\epsilon) \tag{3.18}$$

Expanding the Gaussians, we have

$$m_{xf}(\boldsymbol{x}_i) = \underbrace{(2\pi)^{(-M/2)} det(\boldsymbol{m}_{P}^{(i)})^{1/2}}_{\tilde{k}_i^0} exp\Big[-\frac{1}{2}(\boldsymbol{x}_i - \boldsymbol{m}_{\mu}^{(i)})^T \boldsymbol{m}_{P}^{(i)} (\boldsymbol{x}_i - \boldsymbol{m}_{\mu}^{(i)}) \Big] \tag{3.19}$$

$$\underbrace{(2\pi)^{-1/2} (p_\epsilon)^{1/2}}_{\tilde{k}_i^1} \; exp\Big[-\frac{1}{2}(y_i - \boldsymbol{\beta}^T \boldsymbol{x}_i)^T p_\epsilon (y_i - \boldsymbol{\beta}^T \boldsymbol{x}_i) \Big]$$

and

$$m_{xf}(\boldsymbol{x}_i) = \tilde{k}_i^0 \; \tilde{k}_i^1 exp\Big[-\frac{1}{2} \boldsymbol{x}_i^T (\boldsymbol{m}_{P}^{(i)} + \boldsymbol{\beta} p_\epsilon \boldsymbol{\beta}^T) \boldsymbol{x}_i - 2(\boldsymbol{m}_{\mu}^{(i)^T} \boldsymbol{m}_{P}^{(i)} + y_i^T p_\epsilon \boldsymbol{\beta}^T) \boldsymbol{x}_i \Big] \tag{3.20}$$

$$exp\Big[-\frac{1}{2}(\boldsymbol{m}_{\mu}^{(i)^T} \boldsymbol{m}_{P}^{(i)} \boldsymbol{m}_{\mu}^{(i)} + y_i^T p_\epsilon y_i) \Big].$$

Using the following identity ([15];Sec.8.1.6)

$$exp\Big[-\frac{1}{2}\boldsymbol{x}^T\boldsymbol{A}\boldsymbol{x}+\boldsymbol{c}^T\boldsymbol{x}\Big]=exp\Big[-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{A}^{-1}\boldsymbol{c})^T\boldsymbol{A}(\boldsymbol{x}-\boldsymbol{A}^{-1}\boldsymbol{c})+\frac{1}{2}\boldsymbol{c}^T\boldsymbol{A}^{-1}\boldsymbol{c}\Big], \tag{3.21}$$

we may rearranging Eq.(3.20) as

$$m_{xf}(\boldsymbol{x}_i)=\tilde{k}_i^0\,\tilde{k}_i^1\,\underbrace{exp\Big[-\frac{1}{2}(\boldsymbol{m}_\mu^{(i)-T}\boldsymbol{m}_P^{(i)}\boldsymbol{m}_\mu^{(i)}+y_i^Tp_\epsilon y_i-b^T(\tilde{\boldsymbol{m}}_P^{(i)})^{-1}b)\Big]}_{\tilde{k}_i^2} \tag{3.22}$$
$$exp\Big[-\frac{1}{2}(\boldsymbol{x}_i-(\tilde{\boldsymbol{m}}_P^{(i)})^{-1}b)^T\tilde{\boldsymbol{m}}_P^{(i)}(\boldsymbol{x}_i-(\tilde{\boldsymbol{m}}_P^{(i)})^{-1}b)\Big].$$

In the above equation, $\tilde{\boldsymbol{m}}_P^{(i)}$ and $\boldsymbol{b}^T$ are

$$\tilde{\boldsymbol{m}}_P^{(i)}=\boldsymbol{m}_\mu^{(i)}+\boldsymbol{\beta}p_\epsilon\boldsymbol{\beta}^T \tag{3.23}$$
$$\boldsymbol{b}^T=\boldsymbol{m}_\mu^{(i)^T}\boldsymbol{m}_P^{(i)}+y_i^Tp_\epsilon\boldsymbol{\beta}^T. \tag{3.24}$$

To represent the equation as a Gaussian distribution we multiply and divide by the term $(2\pi)^{M/2}det(\tilde{\boldsymbol{m}}_P^{(i)})^{-1/2}$, yielding

$$m_{xf}(\boldsymbol{x}_i)=\tilde{k}_i^0\,\tilde{k}_i^1\,\tilde{k}_i^2\,\underbrace{(2\pi)^{M/2}det(\tilde{\boldsymbol{m}}_P^{(i)})^{-1/2}}_{\tilde{k}_i^3}\,\mathcal{N}\big[\boldsymbol{x}_i;\tilde{\boldsymbol{m}}_\mu^{(i)},\tilde{\boldsymbol{m}}_P^{(i)}\big]. \tag{3.25}$$

The mean vector $\tilde{\boldsymbol{m}}_\mu^{(i)}$ is

$$\tilde{\boldsymbol{m}}_\mu^{(i)}=(\tilde{\boldsymbol{m}}_P^{(i)})^{-1}\boldsymbol{b}, \tag{3.26}$$

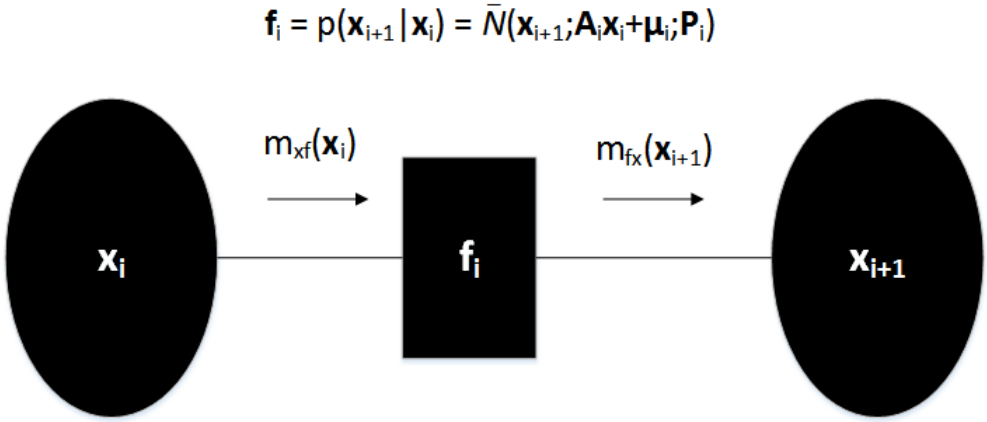

Figure 3.7: Evaluation of factor-to-variable vector message for unmixing

and the final message is the following scaled multivariate Gaussian

$$m_{xf}(\boldsymbol{x}_i) = \tilde{K}_i \, \bar{\mathcal{N}}\big(\boldsymbol{x}_i; \tilde{\boldsymbol{m}}_{\mu}^{(i)}, \tilde{\boldsymbol{m}}_{P}^{(i)}\big), \tag{3.27}$$

with mean $\tilde{\boldsymbol{m}}_{\mu}^{(i)}$, precision $\tilde{\boldsymbol{m}}_{P}^{(i)}$, and scaling constant $\tilde{K}_i = \tilde{k}_i^0 \; \tilde{k}_i^1 \; \tilde{k}_i^2 \; \tilde{k}_i^3$.

Message going from factor node to variable node

The second stage of the derivation is for messages of the form $m_{fx}(\boldsymbol{x}_{i+1})$: from a factor node f_i to a variable node $\boldsymbol{x}_i$. Figure 3.7 depicts the portion of the graph needed to evaluate these messages. The factor node f_i has a factor function that models the spectral correlation of the reflectances between spectral bands $\{i, i+1\}$ for all the M endmembers in the chain and the mathematical representation of this function is the conditional PDF of $\boldsymbol{x}_{i+1}$ given $\boldsymbol{x}_i$

$$f_i = p(\boldsymbol{x}_{i+1}|\boldsymbol{x}_i) = \bar{\mathcal{N}}(\boldsymbol{x}_{i+1}; \boldsymbol{A}_i\boldsymbol{x}_i + \boldsymbol{\mu}_i, \boldsymbol{P}_i), \tag{3.28}$$

where $\boldsymbol{\mu}_i$ and $\boldsymbol{P}_i$ are mean vector precision matrix of the transition noise at the factor node f_i.

The incoming message to node f_i is $m_{xf}(\boldsymbol{x}_i)$ from $\boldsymbol{x}_i$. The outgoing message at node

f_i, $m_{fx}(\boldsymbol{x}_{i+1})$, is derived below using (3.28) and (3.27), and the message passing rules described in Section 3.2.1. From (3.13), we have

$$m_{fx}(\boldsymbol{x}_{i+1}) = \int m_{fx}(\boldsymbol{x}_i)\; p(\boldsymbol{x}_{i+1}|\boldsymbol{x}_i)\; d\boldsymbol{x}_i \tag{3.29}$$

$$= \tilde{K}_i \int \bar{\mathcal{N}}(\boldsymbol{x}_i; \tilde{\boldsymbol{m}}_\mu^{(i)}, \tilde{\boldsymbol{m}}_P^{(i)})\; \bar{\mathcal{N}}(\boldsymbol{x}_{i+1}; \boldsymbol{A}_i\boldsymbol{x}_i + \boldsymbol{\mu}_i, \boldsymbol{P}_i)\; d\boldsymbol{x}_i. \tag{3.30}$$

Expanding the Gaussians, we obtain

$$m_{fx}(\boldsymbol{x}_{i+1}) = \tilde{K}_i \underbrace{2\pi^{-N/2}\; det(\tilde{\boldsymbol{m}}_P^{(i)})^{1/2}}_{k_{i+1}^0} \int exp\Big[-\frac{1}{2}(\boldsymbol{x}_i - \tilde{\boldsymbol{m}}_\mu^{(i)})^T \tilde{\boldsymbol{m}}_P^{(i)}\; (\boldsymbol{x}_i - \tilde{\boldsymbol{m}}_\mu^{(i)})\Big]$$
$$\underbrace{2\pi^{-M/2}\; det(\boldsymbol{P}_i)^{1/2}}_{k_{i+1}^1}\; exp\Big[-\frac{1}{2}(\boldsymbol{x}_{i+1} - \boldsymbol{A}_i\boldsymbol{x}_i - \boldsymbol{\mu}_i)^T \boldsymbol{P}_i (\boldsymbol{x}_{i+1} - \boldsymbol{A}_i\boldsymbol{x}_i - \boldsymbol{\mu}_i)\Big] d\boldsymbol{x}_i \tag{3.31}$$

$$= \tilde{K}_i\; k_{i+1}^0\; k_{i+1}^1 \int exp\Big[-\frac{1}{2}\Big((\boldsymbol{x}_i)^T \tilde{\boldsymbol{m}}_P^{(i)} (\boldsymbol{x}_i) - 2(\tilde{\boldsymbol{m}}_\mu^{(i)})^T \tilde{\boldsymbol{m}}_P^{(i)} (\boldsymbol{x}_i)$$
$$+ (\tilde{\boldsymbol{m}}_\mu^{(i)})^T \tilde{\boldsymbol{m}}_P^{(i)} (\tilde{\boldsymbol{m}}_\mu^{(i)}) + (\boldsymbol{x}_{i+1})^T \boldsymbol{P}_i (\boldsymbol{x}_{i+1}) - 2(\boldsymbol{x}_{i+1})^T \boldsymbol{P}_i \boldsymbol{A}_i (\boldsymbol{x}_i) - 2\boldsymbol{\mu}_i^T \boldsymbol{P}_i \boldsymbol{x}_{i+1} +$$
$$2\boldsymbol{\mu}_i^T \boldsymbol{P}_i \boldsymbol{A}_i \boldsymbol{x}_i + \boldsymbol{x}_i^T \boldsymbol{A}_i^T \boldsymbol{P}_i \boldsymbol{A}_i \boldsymbol{x}_i + \boldsymbol{\mu}_i^T \boldsymbol{P}_i \boldsymbol{\mu}_i \Big)\Big] d\boldsymbol{x}_i \tag{3.32}$$

$$= \tilde{K}_i\; k_{i+1}^0\; k_{i+1}^1\; exp\Big[\frac{-1}{2}\big[(\boldsymbol{x}_{i+1})^T \boldsymbol{P}_i \boldsymbol{x}_{i+1} - 2\boldsymbol{\mu}_i^T \boldsymbol{P}_i \boldsymbol{x}_{i+1} \big] \Big]$$
$$exp\Big[\frac{-1}{2} [(\tilde{\boldsymbol{m}}_\mu^{(i)})^T \tilde{\boldsymbol{m}}_P^{(i)} \tilde{\boldsymbol{m}}_\mu^{(i)} + \boldsymbol{\mu}_i^T \boldsymbol{P}_i \boldsymbol{\mu}_i] \Big] \int exp\Big[\frac{-1}{2} \Big((\boldsymbol{x}_i)^T \underbrace{(\tilde{\boldsymbol{m}}_P^{(i)} + \boldsymbol{A}_i^T \boldsymbol{P}_i \boldsymbol{A}_i)}_{\boldsymbol{R}} \boldsymbol{x}_i -$$
$$2 \Big(\underbrace{(\tilde{\boldsymbol{m}}_\mu^{(i)})^T \tilde{\boldsymbol{m}}_P^{(i)} + (\boldsymbol{x}_{i+1})^T \boldsymbol{P}_i \boldsymbol{A}_i - \boldsymbol{\mu}_i^T \boldsymbol{P}_i \boldsymbol{A}_i}_{\boldsymbol{s}^T} \Big) \boldsymbol{x}_i \Big) \Big]\; d\boldsymbol{x}_i. \tag{3.33}$$

Making the substitutions $\boldsymbol{R}$ and $\boldsymbol{s}^T$ indicated in (3.33), we have

$$\begin{aligned} m_{fx}(\boldsymbol{x}_{i+1}) = \tilde{K}_i\, k_{i+1}^0\, k_{i+1}^1\, exp\Big[\frac{-1}{2}\big[(\boldsymbol{x}_{i+1})^T\boldsymbol{P}_i\boldsymbol{x}_{i+1} - 2\boldsymbol{\mu}_i^T\boldsymbol{P}_i\boldsymbol{x}_{i+1}\big]\Big] \\ exp\Big[\frac{-1}{2}\big[(\tilde{\boldsymbol{m}}_\mu^{(i)})^T\tilde{\boldsymbol{m}}_P^{(i)}\tilde{\boldsymbol{m}}_\mu^{(i)} + \boldsymbol{\mu}_i^T\boldsymbol{P}_i\boldsymbol{\mu}_i\big]\Big] \\ \int exp\Big[\frac{-1}{2}\big((\boldsymbol{x}_i)^T\boldsymbol{R}\boldsymbol{x}_i - 2\boldsymbol{s}^T\boldsymbol{x}_i\big)\Big]\, d\boldsymbol{x}_i \end{aligned} \tag{3.34}$$

Using the following identity ([15];Sec. 8.1.1)

$$\int exp\Big[-\frac{1}{2}\boldsymbol{x}^T\boldsymbol{C}\boldsymbol{x} + \boldsymbol{q}^T\boldsymbol{x}\Big]d\boldsymbol{x} = \sqrt{(det(2\pi(\boldsymbol{C})^{-1})}\, exp\Big[\frac{1}{2}\boldsymbol{q}^T\boldsymbol{C}^{-1}\boldsymbol{q}\Big], \tag{3.35}$$

We integrate the quadratic form in (3.34) to obtain

$$\begin{aligned} m_{fx}(\boldsymbol{x}_{i+1}) = \tilde{K}_i\, k_{i+1}^0 \cdot k_{i+1}^1\, \sqrt{det(2\pi\boldsymbol{R}^{-1})}\, exp\Big[\frac{\boldsymbol{s}^T\boldsymbol{R}^{-1}\boldsymbol{s}}{2}\Big] exp\Big[\frac{-1}{2}[(\boldsymbol{x}_{i+1})^T\boldsymbol{P}_i\boldsymbol{x}_{i+1} \\ - 2\boldsymbol{\mu}_i^T\boldsymbol{P}_i\boldsymbol{x}_{i+1}]\Big] exp\Big[\frac{-1}{2}[(\tilde{\boldsymbol{m}}_\mu^{(i)})^T\tilde{\boldsymbol{m}}_P^{(i)}\tilde{\boldsymbol{m}}_\mu^{(i)} + \boldsymbol{\mu}_i^T\boldsymbol{P}_i\boldsymbol{\mu}_i]\Big] \end{aligned} \tag{3.36}$$

$$\begin{aligned} = \tilde{K}_i\, k_{i+1}^0\, k_{i+1}^1\, \sqrt{det(2\pi\boldsymbol{R}^{-1})}\, exp\Big[\frac{1}{2}\big[(\tilde{\boldsymbol{m}}_\mu^{(i)})^T\tilde{\boldsymbol{m}}_P^{(i)} + (\boldsymbol{x}_{i+1})^T\boldsymbol{P}_i\boldsymbol{A}_i - (\boldsymbol{\mu}_i)^T\boldsymbol{P}_i\boldsymbol{A}_i\big]\boldsymbol{R}^{-1}\cdot \\ \big[(\tilde{\boldsymbol{m}}_P^{(i)})^T\tilde{\boldsymbol{m}}_\mu^{(i)} + \boldsymbol{A}_i^T\boldsymbol{P}_i^T(\boldsymbol{x}_{i+1}) - \boldsymbol{A}_i^T\boldsymbol{P}_i^T\boldsymbol{\mu}_i\big]\Big]\, exp\Big[\frac{-1}{2}\big[(\boldsymbol{x}_{i+1})^T\boldsymbol{P}_i\boldsymbol{x}_{i+1} - 2\boldsymbol{\mu}_i^T\boldsymbol{P}_i\boldsymbol{x}_{i+1}\big]\Big] \\ exp\Big[\frac{-1}{2}\big[(\tilde{\boldsymbol{m}}_\mu^{(i)})^T\tilde{\boldsymbol{m}}_P^{(i)}\tilde{\boldsymbol{m}}_\mu^{(i)} + \boldsymbol{\mu}_i^T\boldsymbol{P}_i\boldsymbol{\mu}_i\big]\Big] \end{aligned} \tag{3.37}$$

$$= \tilde{K}_i k_{i+1}^0 k_{i+1}^1 \sqrt{det(2\pi \boldsymbol{R}^{-1})}\, exp\Bigg[\frac{1}{2}\Bigg((\tilde{\boldsymbol{m}}_{\mu}^{(i)})^T \tilde{\boldsymbol{m}}_P^{(i)} \boldsymbol{R}^{-1} (\tilde{\boldsymbol{m}}_P^{(i)})^T \tilde{\boldsymbol{m}}_{\mu}^{(i)} + (\tilde{\boldsymbol{m}}_{\mu}^{(i)})^T \tilde{\boldsymbol{m}}_P^{(i)} \boldsymbol{R}^{-1} \boldsymbol{A}_i^T \boldsymbol{P}_i^T \boldsymbol{x}_{i+1} - (\tilde{\boldsymbol{m}}_{\mu}^{(i)})^T \tilde{\boldsymbol{m}}_P^{(i)} \boldsymbol{R}^{-1} \boldsymbol{A}_i^T \boldsymbol{P}_i^T \boldsymbol{\mu}_i + (\boldsymbol{x}_{i+1})^T \boldsymbol{P}_i \boldsymbol{A}_i \boldsymbol{R}^{-1} (\tilde{\boldsymbol{m}}_P^{(i)})^T \tilde{\boldsymbol{m}}_{\mu}^{(i)} + (\boldsymbol{x}_{i+1})^T \boldsymbol{P}_i \boldsymbol{A}_i \boldsymbol{R}^{-1} \boldsymbol{A}_i^T \boldsymbol{P}_i^T \boldsymbol{x}_{i+1} - (\boldsymbol{x}_{i+1})^T \boldsymbol{P}_i \boldsymbol{A}_i \boldsymbol{R}^{-1} \boldsymbol{A}_i^T \boldsymbol{P}_i^T \boldsymbol{\mu}_i - \boldsymbol{\mu}_i^T \boldsymbol{P}_i \boldsymbol{A}_i \boldsymbol{R}^{-1} (\tilde{\boldsymbol{m}}_P^{(i)})^T (\tilde{\boldsymbol{m}}_{\mu}^{(i)}) - \boldsymbol{\mu}_i^T \boldsymbol{P}_t \boldsymbol{A}_i \boldsymbol{R}^{-1} \boldsymbol{A}_i^T \boldsymbol{P}_i^T \boldsymbol{x}_{i+1} + \boldsymbol{\mu}_i^T \boldsymbol{P}_i \boldsymbol{A}_i \boldsymbol{R}^{-1} \boldsymbol{A}_i^T \boldsymbol{P}_i^T \boldsymbol{\mu}_i \Bigg)\Bigg]. \tag{3.38}$$

From here on we focus on rearranging the form of (3.38) into a Gaussian equation, as the integration of a Gaussian with respect to $\boldsymbol{x}_i$ marginalizes the multivariate function and the left over terms represent a Gaussian distribution on $\boldsymbol{x}_{i+1}$. From (3.38),

$$m_{fx}(\boldsymbol{x}_{i+1}) = \tilde{K}_i\, k_{i+1}^0\, k_{i+1}^1\, \sqrt{det(2\pi \boldsymbol{R}^{-1})}\, exp\Bigg[-\frac{1}{2} \boldsymbol{x}_{i+1}^T (\boldsymbol{P}_i - \boldsymbol{P}_i \boldsymbol{A}_i \boldsymbol{R}^{-1} \boldsymbol{A}_i^T (\boldsymbol{P}_i)^T) \boldsymbol{x}_{i+1} + \frac{1}{2}\Big(2(\tilde{\boldsymbol{m}}_{\mu}^{(i)})^T \tilde{\boldsymbol{m}}_P^{(i)} \boldsymbol{R}^{-1} \boldsymbol{A}_i^T \boldsymbol{P}_i^T - 2\boldsymbol{\mu}_i^T \boldsymbol{P}_i \boldsymbol{A}_i \boldsymbol{R}^{-1} \boldsymbol{A}_i^T \boldsymbol{P}_i^T + 2\boldsymbol{\mu}_i^T \boldsymbol{P}_i \Big) \boldsymbol{x}_{i+1} \Bigg]$$
$$exp\Bigg[-\frac{1}{2}\Bigg((\tilde{\boldsymbol{m}}_{\mu}^{(i)})^T \Big(\tilde{\boldsymbol{m}}_P^{(i)} - \tilde{\boldsymbol{m}}_P^{(i)} \boldsymbol{R}^{-1} (\tilde{\boldsymbol{m}}_P^{(i)})^T \Big) \tilde{\boldsymbol{m}}_{\mu}^{(i)} + \boldsymbol{\mu}_i^T \Big(\boldsymbol{P}_i - \boldsymbol{P}_i \boldsymbol{A}_i \boldsymbol{R}^{-1} \boldsymbol{A}_i^T \boldsymbol{P}_i^T \Big) \boldsymbol{\mu}_i \Bigg) \Bigg]$$
$$exp\Big[\frac{1}{2} (-2\boldsymbol{\mu}_i^T \boldsymbol{P}_i \boldsymbol{A}_i \boldsymbol{R}^{-1} (\tilde{\boldsymbol{m}}_P^{(i)})^T \tilde{\boldsymbol{m}}_{\mu}^{(i)}) \Big]. \tag{3.39}$$

Next step is to rearrange the above equation into a squared form and apply the identity shown in Eq.(3.21) [15], so that Eq.(3.39) looks like a Gaussian in terms of $\boldsymbol{x}_{i+1}$:

$$m_{fx}(\boldsymbol{x}_{i+1}) = \tilde{K}_i\, k_{i+1}^0\, k_{i+1}^1\, \sqrt{det(2\pi \boldsymbol{R}^{-1})}\, exp\Bigg[-\frac{1}{2} \boldsymbol{x}_{i+1}^T \underbrace{(\boldsymbol{P}_i - \boldsymbol{P}_i \boldsymbol{A}_i \boldsymbol{R}^{-1} \boldsymbol{A}_i^T (\boldsymbol{P}_i)^T)}_{\boldsymbol{m}_P^{(i+1)}} \boldsymbol{x}_{i+1} + \Big(\underbrace{(\tilde{\boldsymbol{m}}_{\mu}^{(i)})^T \tilde{\boldsymbol{m}}_P^{(i)} \boldsymbol{R}^{-1} \boldsymbol{A}_i^T \boldsymbol{P}_i^T + \boldsymbol{\mu}_i^T \boldsymbol{P}_i (\mathbf{1} - \boldsymbol{A}_i \boldsymbol{R}^{-1} \boldsymbol{A}_i^T \boldsymbol{P}_i^T)}_{\boldsymbol{h}^T} \Big) \boldsymbol{x}_{i+1} \Bigg]$$
$$exp\Bigg[-\frac{1}{2}\Bigg((\tilde{\boldsymbol{m}}_{\mu}^{(i)})^T \Big(\tilde{\boldsymbol{m}}_P^{(i)} - \tilde{\boldsymbol{m}}_P^{(i)} \boldsymbol{R}^{-1} (\tilde{\boldsymbol{m}}_P^{(i)})^T \Big) \tilde{\boldsymbol{m}}_{\mu}^{(i)} + \boldsymbol{\mu}_i^T \Big(\boldsymbol{P}_i - \boldsymbol{P}_i \boldsymbol{A}_i \boldsymbol{R}^{-1} \boldsymbol{A}_i^T \boldsymbol{P}_i^T \Big) \boldsymbol{\mu}_i$$
$$\boldsymbol{\mu}_i^T \boldsymbol{P}_i \boldsymbol{A}_i \boldsymbol{R}^{-1} (\tilde{\boldsymbol{m}}_P^{(i)})^T \tilde{\boldsymbol{m}}_{\mu}^{(i)} \Bigg) \Bigg] \tag{3.40}$$

We Assign the constant k_{i+1}^2 to the latter part of above equation

$$k_{i+1}^2 = exp\Big[-\frac{1}{2}\Big((\tilde{\boldsymbol{m}}_{\mu}^{(i)})^T \Big(\tilde{\boldsymbol{m}}_P^{(i)} - \tilde{\boldsymbol{m}}_P^{(i)} \boldsymbol{R}^{-1} (\tilde{\boldsymbol{m}}_P^{(i)})^T \Big) \tilde{\boldsymbol{m}}_{\mu}^{(i)} + \boldsymbol{\mu}_i^T \Big(\boldsymbol{P}_i - \boldsymbol{P}_i \boldsymbol{A}_i \boldsymbol{R}^{-1} \boldsymbol{A}_i^T \boldsymbol{P}_i^T \Big) \boldsymbol{\mu}_i - \boldsymbol{\mu}_i^T \boldsymbol{P}_i \boldsymbol{A}_i \boldsymbol{R}^{-1} (\tilde{\boldsymbol{m}}_P^{(i)})^T \tilde{\boldsymbol{m}}_{\mu}^{(i)} \Big) \Big] \tag{3.41}$$

and rewrite (3.40) in terms of $\boldsymbol{m}_P^{(i+1)}$, i.e Precision of $m_{fx}(\boldsymbol{x}_{i+1})$, and the pseudo constant $\boldsymbol{h}^T$ to give

$$m_{fx}(\boldsymbol{x}_{i+1}) = \tilde{K}\ k_{i+1}^0\ k_{i+1}^1\ k_{i+1}^2\ \sqrt{det(2\pi \boldsymbol{R}^{-1})}\ exp\Big[-\frac{1}{2} \boldsymbol{x}_{i+1}^T \boldsymbol{m}_P^{(i+1)} \boldsymbol{x}_{i+1} + \boldsymbol{h}^T \boldsymbol{x}_{i+1} \Big] \tag{3.42}$$

Multiplying and dividing the above equation by $\Big(2\pi^{M/2} det(\boldsymbol{m}_P^{(i+1)})^{-1/2} \Big)$, we obtain

$$\begin{aligned} m_{fx}(\boldsymbol{x}_{i+1}) =& \tilde{K}\ k_{i+1}^0\ k_{i+1}^1\ k_{i+1}^2\ \sqrt{det(2\pi \boldsymbol{R}^{-1})}\ exp\Big[-\frac{1}{2} \boldsymbol{x}_{i+1}^T \boldsymbol{m}_P^{(i+1)} \boldsymbol{x}_{i+1} + \boldsymbol{h}^T \boldsymbol{x}_{i+1} \Big] \\ & \Big(2\pi^{M/2} det(\boldsymbol{m}_P^{(i+1)})^{-1/2} \Big)\ \Big(2\pi^{M/2} det(\boldsymbol{m}_P^{(i+1)})^{-1/2} \Big)^{-1} \end{aligned} \tag{3.43}$$

As $\boldsymbol{R}$ is an $M \times M$ matrix, $\sqrt{det(2\pi \boldsymbol{R}^{-1})}$ is written as $2\pi^{M/2} \sqrt{det(\boldsymbol{R}^{-1})}$, and

$$\begin{aligned} m_{fx}(\boldsymbol{x}_{i+1}) =& \tilde{K}\ k_{i+1}^0\ k_{i+1}^1\ k_{i+1}^2\ \underbrace{2\pi^{M/2} \sqrt{det(\boldsymbol{R}^{-1})}\ 2\pi^{M/2} det(\boldsymbol{m}_P^{(i+1)})^{-1/2}}_{k_{i+1}^3} \\ & 2\pi^{-M/2} det(\boldsymbol{m}_P^{(i+1)})^{1/2} exp\Big[-\frac{1}{2} \boldsymbol{x}_{i+1}^T \boldsymbol{m}_P^{(i+1)} \boldsymbol{x}_{i+1} + \boldsymbol{h}^T \boldsymbol{x}_{i+1} \Big] \end{aligned} \tag{3.44}$$

Rearranging (3.44) using the identity (3.21), we have

$$m_{fx}(\boldsymbol{x}_{i+1}) = \tilde{K}\ k_{i+1}^0\ k_{i+1}^1\ k_{i+1}^2\ k_{i+1}^3\ \underbrace{exp\Big[\frac{1}{2}\boldsymbol{h}^T(\boldsymbol{m}_P^{(i+1)})^{-1}\boldsymbol{h}\Big]}_{k_{i+1}^4}\ 2\pi^{-M/2}\ det(\boldsymbol{m}_P^{(i+1)})^{-1/2}$$
$$exp\Big[-\frac{1}{2}\Big(\boldsymbol{x}_{i+1} - (\boldsymbol{m}_P^{(i+1)})^{-1}\boldsymbol{h}\Big)^T \boldsymbol{m}_P^{(i+1)}\ \Big(\boldsymbol{x}_{i+1} - (\boldsymbol{m}_P^{(i+1)})^{-1}\boldsymbol{h}\Big)\Big] \tag{3.45}$$

The above equation represents a Gaussian distribution in vector form with mean $\boldsymbol{m}_\mu^{(i+1)}$ and precision $\boldsymbol{m}_P^{(i+1)}$. This is concisely represented as

$$m_{fx}(\boldsymbol{x}_{i+1}) = \underbrace{\tilde{K}\ k_{i+1}^0\ k_{i+1}^1\ k_{i+1}^2\ k_{i+1}^3\ k_{i+1}^4}_{K_{i+1}} \bar{\mathcal{N}}\Big(\boldsymbol{x}_{i+1}; \boldsymbol{m}_\mu^{(i+1)}, \boldsymbol{m}_P^{(i+1)}\Big), \tag{3.46}$$

where precision matrix $\boldsymbol{m}_P^{(i+1)}$ and mean vector $\boldsymbol{m}_\mu^{(i+1)}$ are

$$\boldsymbol{m}_P^{(i+1)} = \boldsymbol{P}_i - \boldsymbol{P}_i\boldsymbol{A}\boldsymbol{R}^{-1}\boldsymbol{A}^T\boldsymbol{P}_i^T \tag{3.47}$$
$$\boldsymbol{m}_\mu^{(i+1)} = (\boldsymbol{m}_P^{(i+1)})^{-1}\ \boldsymbol{h} \tag{3.48}$$
$$\boldsymbol{h}^T = (\tilde{\boldsymbol{m}}_\mu^{(i)})^T\tilde{\boldsymbol{m}}_P^{(i)}\boldsymbol{R}^{-1}\boldsymbol{A}^T\boldsymbol{P}_t^T + \boldsymbol{\mu}_i^T\boldsymbol{P}_i(\mathbf{1} - \boldsymbol{A}\boldsymbol{R}^{-1}\boldsymbol{A}^T\boldsymbol{P}_i^T) \tag{3.49}$$

Finally, concatinating the constants as in (3.46), the outgoing message is given as a scaled Gaussian

$$m_{fx}(\boldsymbol{x}_{i+1}) = K_{i+1}\ \bar{\mathcal{N}}\Big(\boldsymbol{x}_{i+1}; \boldsymbol{m}_\mu^{(i+1)}, \boldsymbol{m}_P^{(i+1)}\Big) \tag{3.50}$$

This derivation of the message between a factor node and variable node was generic and may be applied to any factor node i to compute messages on Markov chains of any length N.

The likelihood of the chain shown in Figure 3.5 is found by determining the last mes-

sage coming out of the chain, which would be a Gaussian distribution in terms of $\boldsymbol{x}_N$. As proposed in Eq.((3.10)), we marginalize out the reflectance variables, so that the likelihood does not contain any reflectance variables. The likelihood equation for the chain becomes

$$p(\boldsymbol{y};\boldsymbol{\beta}) = \int m_{xf}(\boldsymbol{x}_N) d\boldsymbol{x}_N \tag{3.51}$$

In the above equation, the integration of the message $m_{xf}(\boldsymbol{x}_N)$ leads to a residual constant of integration which is the desired likelihood of the chain.

3.4 Maximum likelihood estimator for abundance ratios

From the previous section, we are able to apply the sum-product algorithm in order to efficiently evaluate the likelihood $p(\mathbf{y};\boldsymbol{\beta})$ of the abundance ratios. From here, we may optimize the likelihood expression in order to obtain a maximum likelihood estimate $\hat{\boldsymbol{\beta}}$ of the abundance ratios

$$\hat{\boldsymbol{\beta}} = \arg \max_{\boldsymbol{\beta}} p(\mathbf{y};\boldsymbol{\beta}) \quad s.t. \quad 0 \leq \beta_m \leq 1, \sum_m \beta_m = 1 \tag{3.52}$$

This non-linear optimization may be performed using any favored constrained optimization routine. In our work, we used an interior point algorithm implemented by Matlab's `fmincon` function. The sum-to-one constraint was achieved via a linear equality constraint on the parameter vector as $\mathbf{1}_M^T \boldsymbol{\beta} = 1$, where $\mathbf{1}_M$ is vector of M ones.

3.5 Computational complexity

Computational complexity of evaluating the likelihood plays a key role in judging the efficiency of unmixing algorithms. Because the brute force algorithm in Section 2.4.2 is

the only other algorithm that supports realistic correlation effects between endmember elements, we compare the complexity of the proposed sum-product unmixing algorithm against brute force.

The brute force unmxing algorithm has a computational complexity of $\mathcal{O}(N^3)$ because the likelihood equation shown in (2.32) requires the computation of an inverse and determinant of the covariance matrix of the pixel spectra, which is of size $N \times N$. The computational complexity of both operations for an $N \times N$ matrix is $\mathcal{O}(N^3)$, yielding an overall complexity that is also $\mathcal{O}(N^3)$.

Our proposed algorithm, sum-product unmixing (SPUX) has a computational complexity of $\mathcal{O}(M^3N)$. As the likelihood evaluation involves computing messages at each variable node and factor node along the chain, the total computational cost can be broken down into 2 parts, the computational cost of messages m_{fx} and m_{xf}. Computation of the message m_{fx} is of $\mathcal{O}(M^3)$, this is because while calculating the mean of m_{fx} computing the inverse precision matrix of m_{fx} is needed, this is shown in Eq.(3.26). As the precision matrix is of size $M \times M$, the complexity for inverting it is $\mathcal{O}(M^3)$. This message is computed N times as there are N such m_{fx} messages in a chain of length N, hence the computational complexity is $\mathcal{O}(M^3N)$. Similarly the complexity involved in calculating the message m_{xf} is $\mathcal{O}(M^3)$, this is because to calculate the mean of m_{xf} the inverse precision matrix of transition noise and the inverse precision matrix of m_{xf} are needed. Both of these matrices are of size $M \times M$ and thus the computational complexity to invert these precision matrices is $\mathcal{O}(2M^3)$. The message m_{xf} is calculated N times along the chain so the computational complexity for calculating N m_{xf} message is $\mathcal{O}(2M^3N)$. Neglecting the constant 2, the overall computational complexity is $\mathcal{O}(M^3N)$. As both the messages have $\mathcal{O}(M^3N)$ complexity, the complexity of finding the likelihood of the chain is $\mathcal{O}(M^3N)$.

Compared to the brute force unmixing algorithm, we have reduced the computational impact from the number of spectral bands from N^3 to N, i.e the spectral resolution of a hyperspectral sensor only imposes a linear impact the proposed algorithm's computational complexity—but this is at the cost of increasing the computational cost associated with the number of endmembers captured in an pixel. Since generally $N \gg M$, the sum-product unmixing algorithm represents a highly accelerated method to perform unmixing compared to other unmixing algorithms.

Chapter 4: Results and Analysis

In this chapter we demonstrate the unmixing performance of our proposed algorithm and empirically evaluate the computational time required to estimate abundance ratios. Since the brute force algorithm presented in Section 2.4.2 is the only other algorithm supporting realistic correlation effects between endmember elements, we benchmark our performance against this method.

4.1 Dataset

We use endmember reflectances from ASTER spectral library provided by NASA's Jet Propulsion Lab [2] which is a collaboration of three other spectral libraries (The John Hopkins University (JHU) Spectral Library, the Jet Propulsion Laboratory (JPL) Spectral Library, and the United States Geological Survey (USGS)). From this library we selected a subset of materials that had at least $N = 128$ spectral samples in the range $0.4\mu m - 1\mu m$ and at least 10 sample reflectances from which we could estimate the Markov chain parameters $\{\alpha_i, \mu_i, \sigma_i^2\}$ and the Gaussian parameters of mean $\boldsymbol{\mu}^{(m)}$ and covariance $\boldsymbol{C}^{(m)}$. This yielded three materials: soil, borax, and concrete. To perform experiments on mixed data, endmembers were resampled to $N = 256$ spectral samples and mixed pixels were synthesized using Equations (2.26), (2.27), and (2.28).

4.2 Unmixing Performance

4.2.1 Abundance Histograms

Here we compare the empirical distribution of abundance estimates produced by sum-product unmixing and brute force unmixing. Fixing the true abundance values as $\beta(soil) = 0.5$, $\beta(concrete) = 0.1$ and $\beta(borax) = 0.4$, we then simulated 1000 mixed pixel measurements as described above. Applying both algorithms to this dataset, we generate 1000 abundance estimates for each material—for each algorithm. Histograms of these results are shown below.

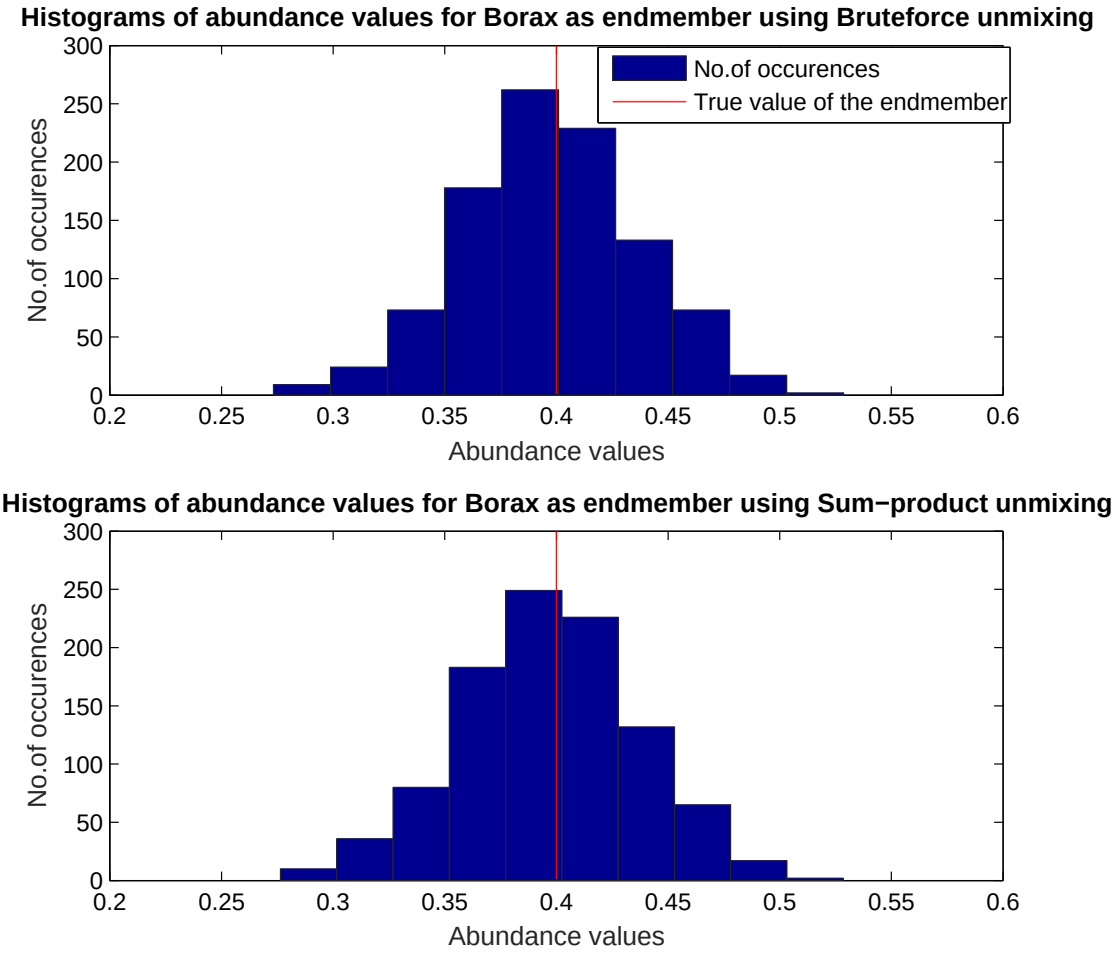

Figure 4.1: Histogram of abundance estimates for Borax as an endmember

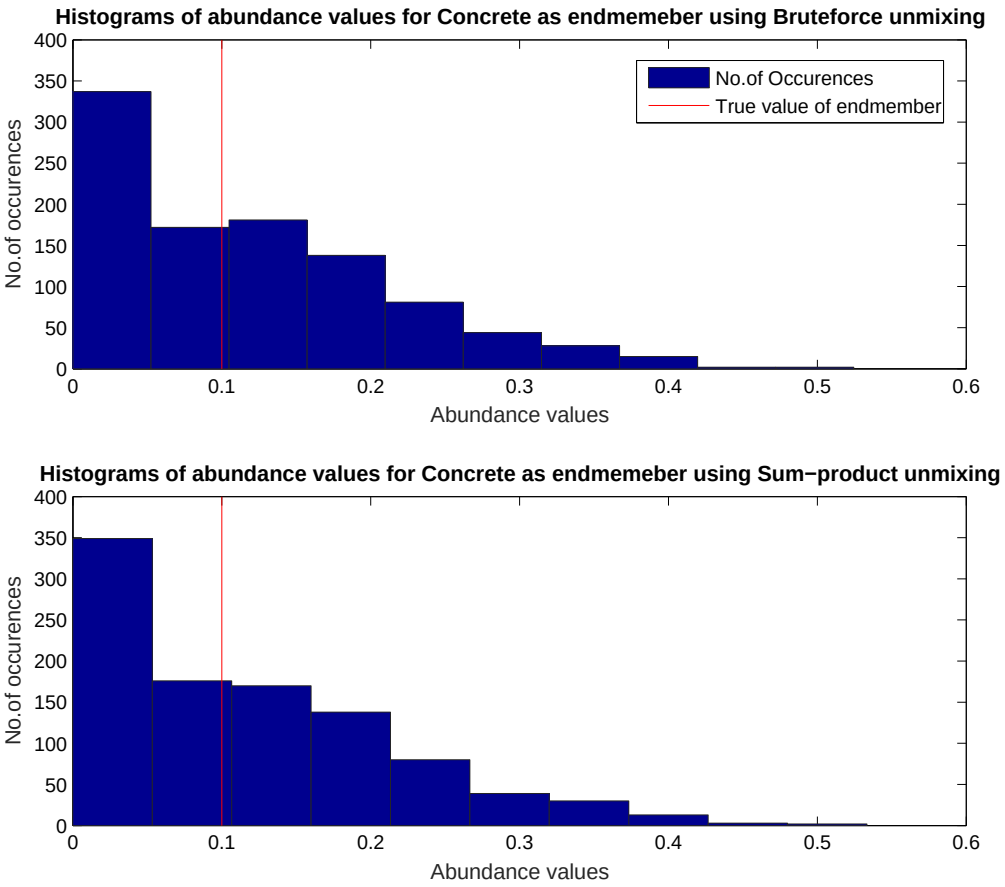

Figure 4.2: Histogram of abundance estimates for Concrete as an endmember

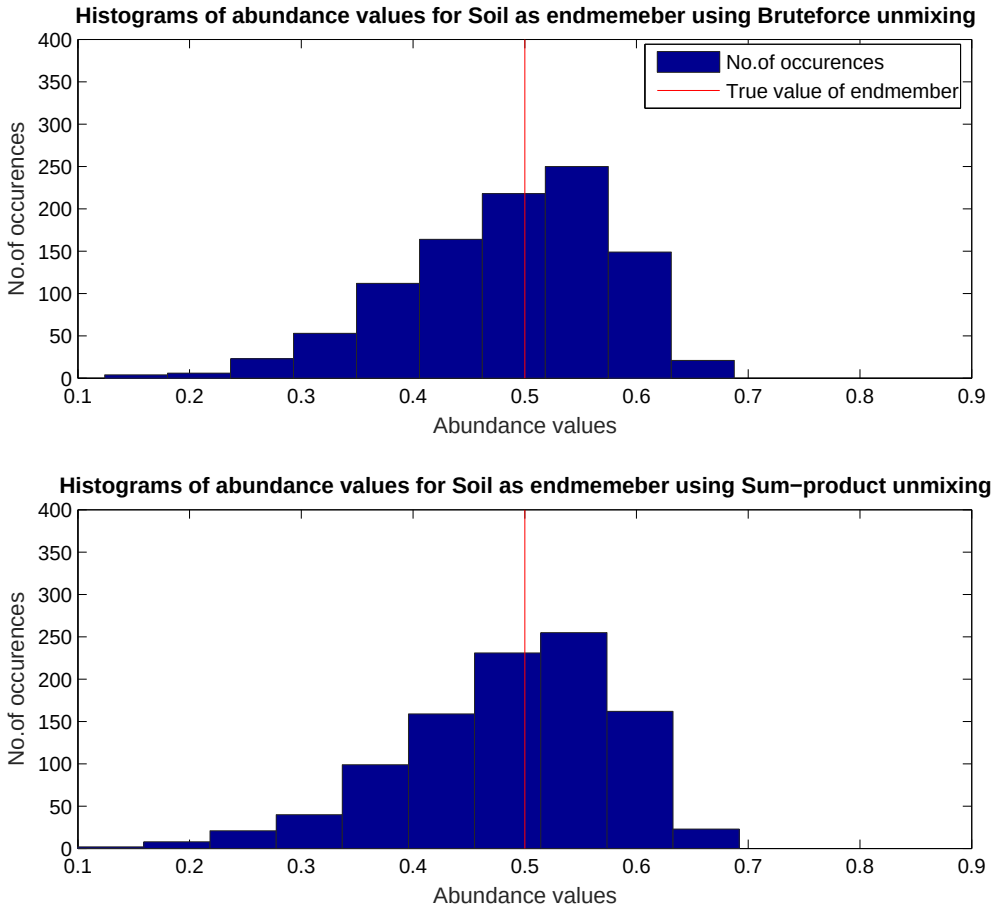

Figure 4.3: Histogram of abundance estimates for Soil as an endmember

We make two observations from the histogram results. First, the estimates for both algorithms are roughly evenly distributed on both sides of the true abundance value (shown by the red vertical line), indicating approximate unbiasedness. Second, and more importantly, the sum-product results appear nearly identical to the brute force results, which are known to be MLE-optimal.

4.2.2 Root Mean Square Error vs Noise level

In this section we evaluate the root mean square error (RMSE) of abundance estimates $\hat{\boldsymbol{\beta}}$ as a function of the standard deviation of additive white Gaussian noise (AWGN). As reflectance values are between $[0, 1]$, we vary standard deviation σ of noise on the same range of that of reflectances. The true abundance values used in creating the synthetic mixed pixels were β(Borax) = 0.4, β(Concrete) = 0.1 and β(Soil) = 0.5. For each value of σ we generated 1000 mixed pixels and empirically evaluated the RMSE from the associated estimates. These results are shown in Figure 4.4. It can be seen from the plot that the errors between the algorithms are similar for small values of σ, but as noise increases, the gap between the RMSE curves increases—indicating better performance for our sum-product unmixing algorithm compared to the brute force algorithm.

4.3 Computation Time

The primary motivation of this work was to develop an accelerated unmixing algorithm, accommodating endmember variability and correlation, that remains computationally tractable for large spectral sampling rates. Here we quantify the run-time of our sum-product algorithm and compare it to the brute force approach. To this end, we artificially increased the spectral sample rate of the ASTER data by interpolating the given samples—creating sample rates ranging from $N = 128$ to $N = 8192$. Using this data, we empirically measured

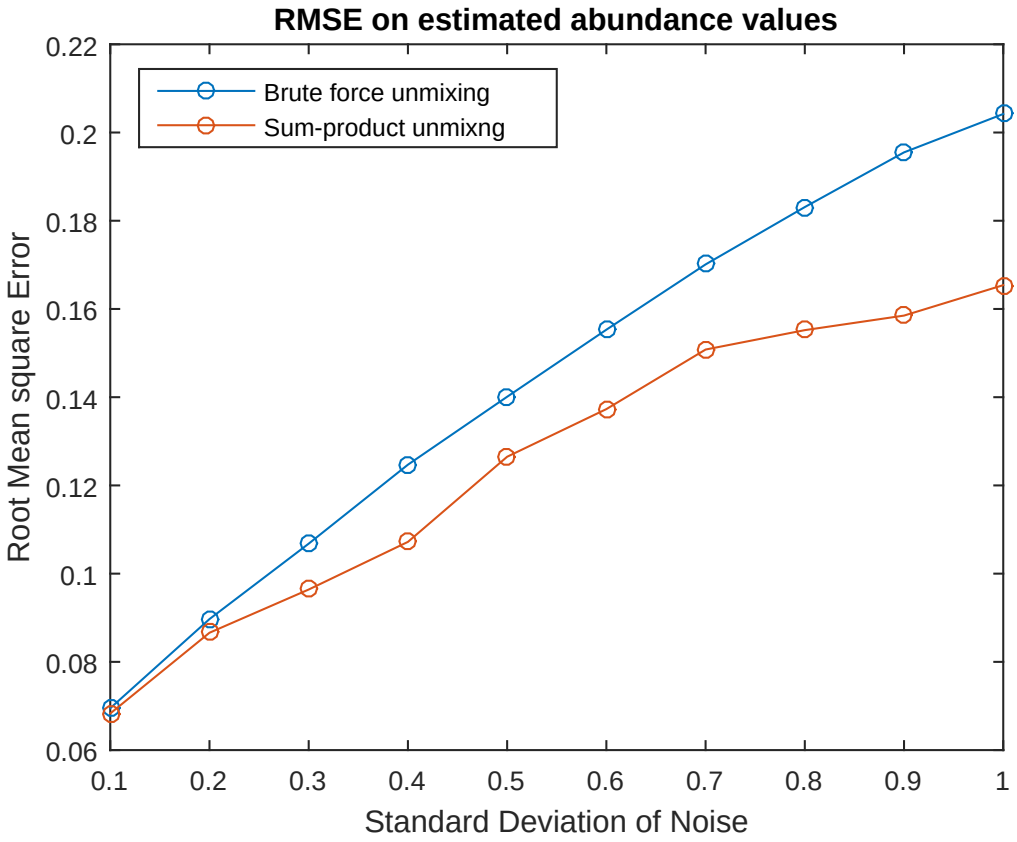

Figure 4.4: Root mean square error of abundance value estimates for the proposed sum-product algorithm and the brute force algorithm

the average time required to evaluate the likelihood function for each method. The results are show in Figure 4.5.

From the figure, we observe that when the number of spectral samples exceeds $N \approx 500$, the sum-produce unmixing algorithm is faster. The theoretical timing predictions in Section 3.5 are corroborated by the figure. The performance gap increases dramatically with N, and at $N = 8192$ the sum-product algorithm is nearly three orders of magnitude faster than the brute force approach.

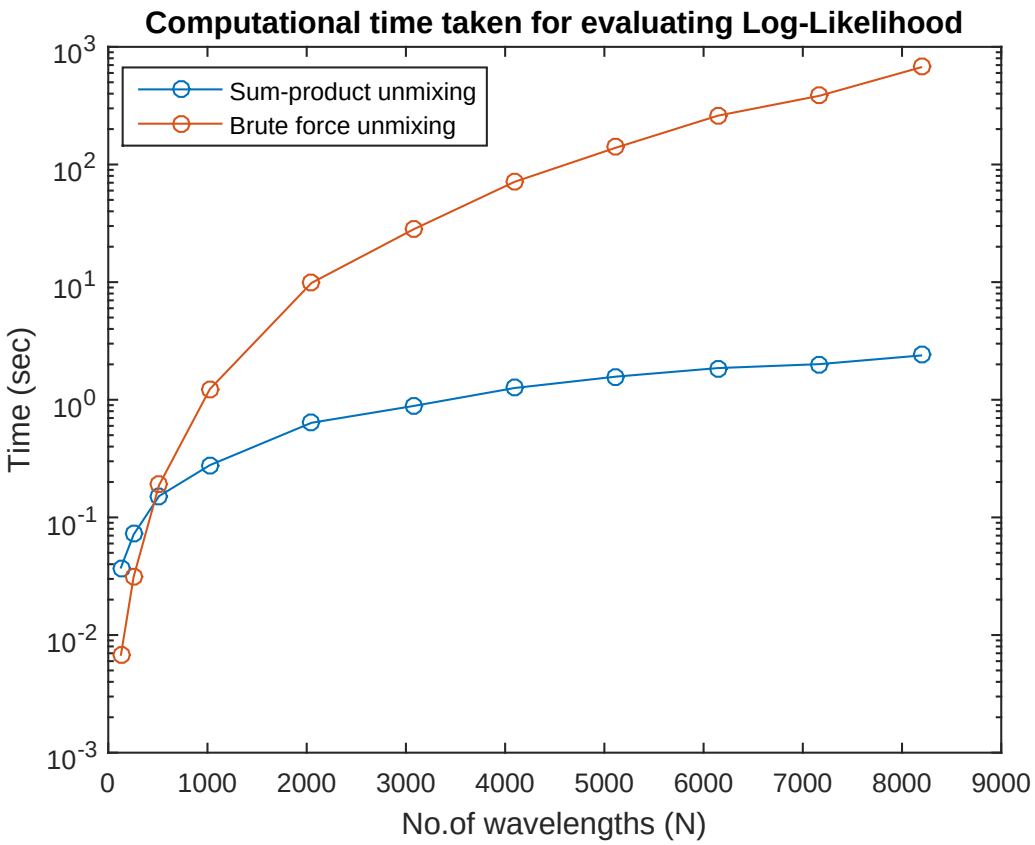

Figure 4.5: Computation time for evaluating abundance likelihoods. The proposed sum-product unmixing algorithm is orders of magnitude faster for large N (ultraspectral data).

Chapter 5: Conclusions

Advances in sensor technology are pushing remote spectrographic methods to ever-higher spectral resolutions. With current technology transitioning from the hyperspectral realm (hundreds of wavelengths) to the ultraspectra realm (thousands of wavelengths), the volume of data generated by these sensors is eclipsing our ability perform statistical unmixing with models that support endmember variability and realistic inter-wavelength correlation properties. In this thesis, we developed a new statistical unmixing method to address this computational bottleneck. Our approach utilized Markov chain models to capture endmember variability, and by formulating the unmixing problem as an inference task on a probabilistic factor graph, we where able to employ the sum-product algorithm for efficient inference. For N-band data, traditional methods have computational complexity $\mathcal{O}(N^3)$, whereas the message-passing method developed only has complexity $\mathcal{O}(NM^3)$ that is linear in the number of bands. Since the number M of endmembers considered is generally much less than N, this algorithm represents a significant computational savings. Examples demonstrated that the new algorithm had estimation error rates consistent with optimal brute force methods, but that it executed three orders of magnitude faster for large spectral sample rates N.

In future work we will extend the method to non-linear mixing models by generalizing the observation factors $\{\phi_i\}$ in Figure 3.2. This approach will maintain most of the message-passing infrastructure developed in this thesis and enable the first consideration of endmem-

ber variability in mixing environments that are non-linear. Following [26], future work will also consider richer modeling parametrizations, including, e.g., background noise parameters and abundance priors, that can be jointly estimated in an expectation maximization (EM) context. While current methods restrict this approach to unrealistic endmember models with inter-band independence assumptions, the message-passing methods developed in this thesis will enable EM methods to be applied to more realistic endmember correlation structures captured by the Markov chain models proposed in this thesis.

Appendix: Notation

In this appendix, we summarize some of the notation used throughout the thesis.

$\boldsymbol{v}$	bold lowercase variables denote column vectors
$\boldsymbol{A}$	bold uppercase variables denote matricies
$\boldsymbol{I}$	identity matrix
$det(\boldsymbol{A})$	matrix determinant
$\boldsymbol{A}^T$	matrix transpose
$p(x)$	probability density function on x
$x_i^{(m)}$	reflectance of endmember m at i^{th} wavelength
$\boldsymbol{x}^{(m)}$	reflectance vector of the m^{th} endmember across N wavelengths
$\boldsymbol{x}_i$	reflectance vector at i^{th} wavelength for M endmembers
$\boldsymbol{\mu}^{(m)}$	mean vector of endmember m
$\boldsymbol{C}^{(m)}$	covariance matrix of endmember m
$\mathcal{N}(\boldsymbol{x}; \boldsymbol{\mu}, \boldsymbol{C})$	Gaussian distribution with mean $\boldsymbol{\mu}$ and covariance matrix $\boldsymbol{C}$ over variable $\boldsymbol{x}$
$\bar{\mathcal{N}}(\boldsymbol{x}; \boldsymbol{\mu}, \boldsymbol{P})$	Gaussian distribution with mean $\boldsymbol{\mu}$ and precision matrix $\boldsymbol{P}$ over variable $\boldsymbol{x}$

$\boldsymbol{\mu}_i$	mean vector of transition noise
$\boldsymbol{P}_i$	precision matrix of transition noise
$\boldsymbol{m}_{\mu}^{(i)}$	mean vector of the incoming message at wavelength i
$\tilde{\boldsymbol{m}}_{\mu}^{(i)}$	mean vector of the outgoing message at wavelength i
$\boldsymbol{m}_{P}^{(i)}$	precision matrix of the incoming message at wavelength i
$\tilde{\boldsymbol{m}}_{P}^{(i)}$	precision matrix of the outgoing message at wavelength i

Bibliography

[1] A.Zare and K.C.Ho. Endmember variabilty in hyperspectral analysis: Addressing spectral variabilty during spectral unmixing. In *IEEE Signal processing Magazine.*, volume 31.1, pages 95–104, Jan. 2014.

[2] A. M. Baldridge, S.J. Hook, C.I. Grove, and G. Rivera. The aster spectral library version 2.0. In *Remote Sensing of Environment*, volume 113, pages 711–715, 2009.

[3] Jose M. Bioucas-Dias and Antonio Plaza. An overview on hyperspectral unmixing: geometrical, statistical and sparse regression based approaches. In *IEEE International Geoscience and remote Sensing Symposium (IGARSS'11)*, pages 1135–1138, 24-29 July. 2011.

[4] J. W. Boardman. Automating spectral unmixing of aviris data using convex geometry concepts. In *JPL, Summaries of the 4th Annual JPL Airborne Geoscience Workshop*, volume 1, pages 11–14, October. 1993.

[5] M.D. Craig. Unsupervised unmixing of remotely sensed images. In *Australasian Remote Sensing Conf.,Perth Australia*, pages 324–330, 1990.

[6] David.J.W, Barry.L.F, Tim.J.W, Drew.P, Paul.G.L, and L.D.Worthy. Detection and identification of toxic air pollutants using airborne LWIR hyperspectral imaigng. In

SPIE Conference on Multispectral and Hyperspectral Remotre Sensing instruments and Applications II, volume 5655, pages 134–141, January. 2005.

[7] D.Bannon. Hyperspectral imaging sensors come into their own for aerospace and defense applications. http://www.militaryaerospace.com/articles/print/volume-24/issue-1/product-intelligence/hyperspectral-imaging-sensors-come-into-their-own-for-aerospace-.html. Accessed: 2015-05-06.

[8] D.Stein, Schoonmaker.J, and Coolbaugh.E. Hyperspectral imaging for intelligence , surveillance and reconnaissance. In *Space and Naval Systems Warfare Center (SSC) San Diego Biennial Review*, pages 108–116, 2001.

[9] B.S Erik and T.F William. Signal and image processing with belief propagation (DSP applications). In *IEEE Signal Processing Magazine*, volume 25.2, pages 114–141, March. 2008.

[10] B.J. Frey and N. Jojic. A comparison of algorithms for inference and learning in probabilistic graphical models. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*, volume 27.9, September. 2005.

[11] Vane G. High spectral resolution remote sensing of the earth. In *Sensors*, volume 2, pages 11–19, December. 1985.

[12] Vane G and A.F.H.Goetz. Terrestrial imaging spectrometry. In *Remote Sensing of Environment*, volume 24, pages 1–29, February. 1988.

[13] J.M.P.Nascimento and J.M.Bioucas-Dias. Nonlinear mixture model for hyperspectral unmixing. In *SPIE Proceedings on Image and Signal processing for Remote Sensing*, volume 7477, September. 2009.

[14] J.O.Rawlings, S.G.Pantula, and D.A.Dickey. *Applied Regression Analysis: A Research tool*. Wadsworth, 1989.

[15] K.B.Petersen and M.S.Pedersen. *The Matrix Cookbook*, volume 20121115. Technical University of Denmark, Denmark, 2012.

[16] F.R Kschischang, B.J Frey, and F.A Loeliger. Factor graphs and the sum-product algorithm. In *IEEE Transactions on Information Theory*, pages 498–519, February. 2001.

[17] Renfu Lu and Yud.R.Chen. Hyperspectral imaging for safety inspection of food and agricultural products. In *SPIE Conference on Pathogen Detection and Remediation for Safe Eating*, volume 3544, pages 121–133, January. 1999.

[18] D.J.C. MacKay. Information theory, inference & learning algorithms. In *Cambridge University Press*, 2002.

[19] Jose' M.P.Nascimento and Jose M. Bioucas-Dias. Vertex component analysis: a fast algorithm to unmix hyperspectral data. In *IEEE Transcation On Geoscience and Remote Sensing*, volume 43.4, pages 898–910, 4 April. 2005.

[20] R.B Myneni and F.G Hall. The interpretation of spectral vegetation indexes. In *IEEE Transcations On Geoscience and Remote Sensing*, volume 33.2, pages 481–486, March. 1995.

[21] N.Keshava and J.F.Mustard. Spectral unmixing. In *IEEE Signal processing Magazine.*, volume 19.1, pages 44–57, Jan. 2002.

[22] O.Eches, Paul Gader, and George Casella. Bayesian estmation of linear mixtures using the normal compositional model.application to hyperspectral imagery. In *IEEE Transcations On Image Processing*, volume 19.6, pages 1403–1413, June. 2010.

[23] A.M. Shahidi, S.R.Patel, J.G.Flanagan, and C.Hudson. Regional variation in human retinal vessel oxygen saturation. In *Experimental Eye Research*, volume 113, pages 143–147, August. 2013.

[24] G. Shaw and H. Burke. Spectral imaging for remote sensing. *Lincoln Laboratory Journal*, 14(1):3–28, 2003.

[25] S.M.Schweizer and J.M.F.Moura. Efficient detection in hyperspectral imagery. In *IEEE Transactions on Image Processing*, volume 10.4, pages 584–597, April. 2001.

[26] David Stein. Application of the normal compositional model to the analysis of hyperspectral imagery. In *IEEE Workshop on Advances in Techniques for Analysis of Remotely Sensed Data*, pages 44–51, 27-28 October. 2003.

[27] V.Gregg, A.F.H.Goetz, and J.B.G.Wellman. Airborne imaging spectrometer: A new tool for remote sensing. In *IEEE Transcations On Geoscience and Remote Sensing*, volume GE-22.6, pages 546–549, November. 1984.

[28] Micheal E. Winter. N-findr: an algorithm for fast autonomous spectral endmember determination in hyperspectral data. In *Proc. SPIE Imaging Spectroscopy V*, volume 5735, pages 266–275, 27 October. 1999.

[29] W.K.Ma, T.H Chan, A.M.Ambikapathi, and C.Y.Chi. A simplex volume maximization framework for hyperspectral endmember extraction. In *IEEE Transaction on Geoscience and Remote Sensing Soceity*, volume 49.11, pages 4177–4193, 27 October. 2011.

[30] W.K.Ma, T.H Chan, N.Gillis, P.Gader, A.M.Ambikapathi, M.B.Dias C.Yung, C.Jose, and A.J.Plaza. A signal processing perspective on hyperspectral unmixing. In *IEEE Signal processing Magazine.*, volume 31.1, pages 67–81, Jan. 2014.

[31] W.K.Ma, T.H Chan, Y.M.Huang, and C.Y.Chi. A convex analysis-based minimum-volume enclosing simplex algorithm for hyperspectral unmixing. In *IEEE Transactions on Signal Processing*, volume 57.11, pages 4418–4432, November. 2009.

[32] Alina Zare, Paul Gader, and George Casella. Sampling piecewise convex unmixing and endmember extraction. In *IEEE Transcations On Geoscience and Remote Sensing*, volume 51.3, pages 1655–1665, March. 2013.