Faculty of Sciences
Department of Applied Mathematics, Computer Science and Statistics
Faculty of Engineering
Department of Telecommunications and Information Processing (TELIN)
Image Processing and Interpretation (IPI)

Volume-based geometric and Bayesian approaches in linear hyperspectral unmixing

Davor Josipovic

Promoter: Prof. Dr. Aleksandra Pizurica
Co-promoter: Prof. Dr. Hongyan Zhang
Co-promoter: Prof. Dr. Dries Benoit

Thesis submitted in fulfillment of the requirements for the degree of
Master of Science in Statistical Data Analysis

Year: 2016–2017

Compiled: June 21, 2017

Faculty of Sciences
Department of Applied Mathematics, Computer Science and Statistics
Faculty of Engineering
Department of Telecommunications and Information Processing (TELIN)
Image Processing and Interpretation (IPI)

Volume-based geometric and Bayesian approaches in linear hyperspectral unmixing

Davor Josipovic

Promoter: Prof. Dr. Aleksandra Pizurica
Co-promoter: Prof. Dr. Hongyan Zhang
Co-promoter: Prof. Dr. Dries Benoit

Thesis submitted in fulfillment of the requirements for the degree of
Master of Science in Statistical Data Analysis

Year: 2016–2017

Compiled: June 21, 2017

Abstract

Hyperspectral cameras make images in hundreds or thousands of spectral channels. Due to low spatial resolution, spectra measured by such cameras are mixtures of spectra of materials in the scene. The reverse process in which materials (called endmembers) are estimated is called hyperspectral unmixing. Unmixing is challenging due to model underspecification, observation noise, endmember variability and high dimensionality. The main focus of this thesis is understanding the statistical assumptions, derivations and implementation details of the geometrical and Bayesian approaches for unmixing. For the geometrical approach we mainly focus on Berman's ICE algorithm. For the Bayesian approach we focus on Arngren's BayesNMF-Vol algorithm. We start by defining the linear mixing model. Then we derive the maximum likelihood estimators for the endmembers and abundances. We show the equivalence between maximizing this log-likelihood function and minimizing the objective function of the ICE algorithm. We further analyze every detail of the minimization process of this function. The limitations drive us to more complex approaches. That is why we subsequently put the unmixing problem into a Bayesian framework. Solving for the posterior calls for custom made Gibbs samplers which we cover in great detail. Finally, the algorithm characteristics are illustrated using synthetic and real data.

The most important original contributions of this work are: (1) the in-depth elaborate analysis of Berman's ICE and Arngren's BayesNMF-Vol algorithm, with original derivations and insights that promote understanding of these methods and their limitations; (2) a novel unmixing algorithm, dubbed ICE-S, that extends the ICE algorithm with spatial information and (3) an original R-framework for the analysis of hyperspectral images, build from ground up and made available as open source code.

Preface

I started working on a topic of deep learning in image pixel classification mainly as a way to learn more about machine learning approaches that I found lacking in the excellent Master of Statistical Data Analysis programme at the University of Ghent. The purpose was to associate each pixel of a hyperspectral image with a class, for example water, brick, grass, etc. Only about 5 % of the image was labeled by a human. The rest of the pixels were for the algorithm to label. The deep learning approaches were mainly focused on prediction, and as such, 'anything goes'. I understood why a logistic regression classifier was a good fit, and even a 1-layer neural network – because it allowed for multiple different pixel-spectra to be associated with the same class – but the state-of-the art approaches did lots of very strange things that apparently worked better than these classic approaches. But why? Were they modeling human error? A pixel in the middle of an ocean could be composed of plastic too, like from a small buoy. If I was able to somehow decompose this pixel in plastic and water, wouldn't that make a better classifier? Little did I know that behind this decomposition there was more than 25 years of research called hyperspectral unmixing. And that is how this thesis was born.

I wish to thank my promoter Prof. Aleksandra Pizurica for the constant encouragements, suggestions, follow up and multiple readthroughs, Prof. Hongyan Zhang for introducing me to the problem of hyperspectral pixel classification and Prof. Dries Benoit for explaining me conceptually the reason behind the intrinsic regularization of the Bayesian approach.

Contents

Chapter 1

Introduction

In this thesis our main focus is *understanding* the methods used in hyperspectral unmixing. This will lead us to some new insights into the current methods, their limitations and some extensions and improvements aimed at overcoming these limitations. We start from the following questions: What is a hyperspectral image? How does it facilitate material detection? What is hyperspectral unmixing? And most importantly, how is it done?

In this Chapter we first learn about the hyperspectral image. Then we look at how materials (that compose the image) leave mixed "fingerprints" in the pixel spectra. Our goal is to extract them. But to be able to extract them, we first need a mathematical model of how they are mixed. Such a mathematical model is defined in Section 1.2. Once we have the mixing model, we explain the unmixing process in Section 1.3 together with some state-of-the-art in the field. Subsequently, in Section 1.4, we give some examples of unmixing applications. In Section 1.5 we cover the notational conventions used in this thesis. And finally in Section 1.6 we give an overview of the subsequent chapters and main contributions.

So what will one learn from these other chapters? Well, if you read the original papers you will note that they are quite succinct on various important topics. For example, when Berman [5] defines the objective function which is to be minimized, he just notes that it follows from the mixing model. We actually derive it and show which assumptions he makes underway. Also, Berman explains the minimization process in a few sentences. But the actual minimization of the constrained quadratic function is much more challenging because the positive-definiteness of the Hessian matrix can not always be guaranteed. Similarly, we find that in the Bayesian model the key underlying assumption is a directed acyclic graph (DAG) with the Markov property without which none of the derivations for the Gibbs sampler can be made. Furthermore, sampling from a degenerate distribution is a challenge in its own. But the original papers put hardly any emphasis on this. So in summary, you will find here all the derivations and nuances which were omitted from the original papers. This will definitely improve the understanding of the discussed unmixing approaches.

Apart from these elaborations and small improvement suggestions, one will also find our own contribution: the extension of the ICE algorithm with spatial information, which we dub ICE-S. But before we delve into the details, we review briefly the basic concepts of hyperspectral imaging.

1.1 Hyperspectral imaging

A hyperspectral image can be seen as an extension of an ordinary RGB image. Whereas an RGB image consists of three bands, red green and blue taken at 485 nm, 550 nm and 645 nm respectively, a hyperspectral image consists of many more (several tens to several hundreds) bands extended beyond the visible spectrum. It is usually represented as a 3D data cube such as the one depicted in Fig. 1.1.

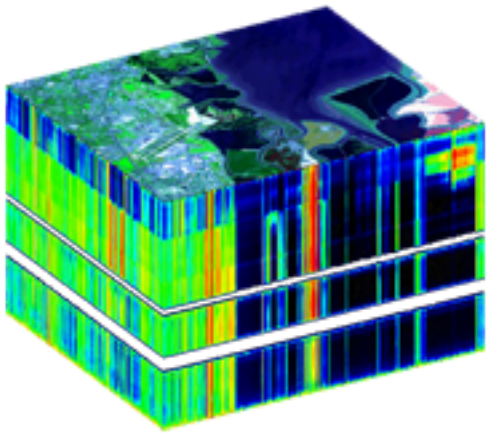


Figure 1.1: This sample data cube [14] was acquired with AVIRIS instrument on August 20, 1992 when it was flown on a NASA ER-2 plane at an altitude of 20,000 meters over Moffett Field, California, at the southern end of the San Francisco Bay. The top of the cube is a false-color image made to accentuate the structure in the water and evaporation ponds on the right. Also visible on the top of the cube is the Moffett Field airport. The sides of the cube are slices showing the edges of the top in all 224 of the AVIRIS spectral bands. The tops of the sides are in the visible part of the spectrum (400 nm), and the bottoms are in the infrared (2500 nm). The sides are pseudo-color, ranging from black and blue (low response) to red (high response).

A hyperspectral image is not to be confused with a multispectral image. For example, the Landsat Thematic Mapper attached to the Landsat 4 and 5 satellites detects only 6 bands (and extra one on a much higher spatial resolution). Researchers use the ratios of those bands to discriminate surface materials with limited accuracy. Hyperspectral images contain many more narrow and continuous bands. This allows better accuracy in analysis. In [11] one can find a more elaborate comparison.

1.2 Spectral mixing

Materials leave unique "fingerprints" in the electromagnetic spectrum, known as spectral signatures. These signatures enable identification of the materials that make up the image. Fig. 1.2 shows this. Notice that there are pure and mixed pixels. For example, the middle emphasized pixel consists entirely of water, while the top emphasized pixel is composed of soil and rock. The latter is due to the spatial resolution of the sensor.

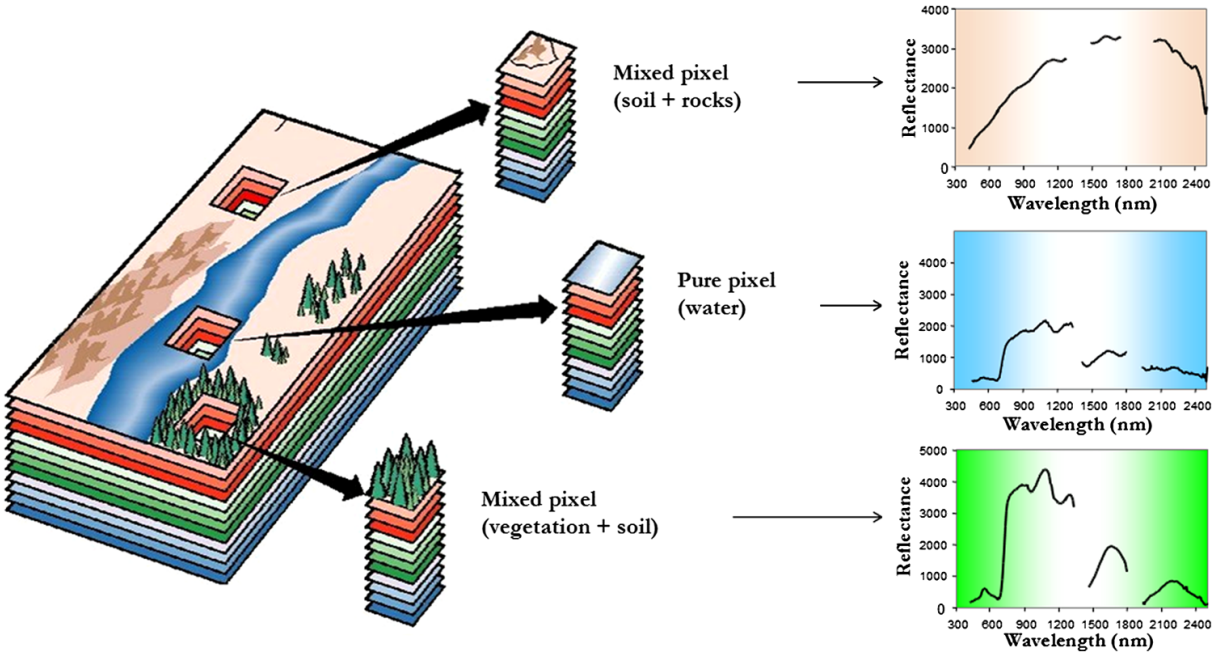


Figure 1.2: Hyperspectral image decomposition [8].

Hyperspectral unmixing is any process that separates the pixel spectra from a hyperspectral image into a collection of constituent spectral signatures, called *endmembers*, and a set of fractional *abundances*, one set per pixel. To relate to Fig. 1.2, the top emphasized pixel is composed of two endmembers (soil and rocks) in abundances of 60 and 40 percent respectively.

1.2.1 Linear mixing model

The linear mixing model (LMM) defines mathematically how the spectra of the materials in a scene are mixed together in a pixel. It assumes that the incident light-rays interact only with the material on which they scatter. In this case, the mixing occurs within the instrument itself due to the fact that the spatial resolution is not fine enough. This mechanism is shown in Fig. 1.3.

Under such a linear mixture, the relative area of the endmembers corresponds to their abundances. Thus the measured spectrum at a pixel $\boldsymbol{x}_n \in \mathbb{R}_+^B$ is a weighted average of the radiances of the materials (i.e. endmembers $\boldsymbol{e}_m$) present at the pixel. We write this as follows:

$$\begin{aligned} \boldsymbol{x}_n &= w_{n1}\boldsymbol{e}_1 + ... + w_{nM}\boldsymbol{e}_M + \boldsymbol{\epsilon} \\ &= \left(\sum_{m=1}^{M} w_{nm}\boldsymbol{e}_m\right) + \boldsymbol{\epsilon} \\ &= \boldsymbol{E}\boldsymbol{w}_n + \boldsymbol{\epsilon} \end{aligned} \tag{1.1}$$

where M denotes the number of endmembers in the image; $\boldsymbol{E} = [\boldsymbol{e}_1, \cdots, \boldsymbol{e}_M] \in \mathbb{R}_+^{B\times M}$ and $\boldsymbol{w}_n \in \mathbb{R}_+^{M\times 1}$ is a column vector of endmember proportions in $\boldsymbol{x}_n$. The noise $\boldsymbol{\epsilon}$ is usually taken as $\mathcal{N}(\boldsymbol{0}, \sigma^2\boldsymbol{I})$ distributed (i.e. the error terms are assumed independent).

Note also that

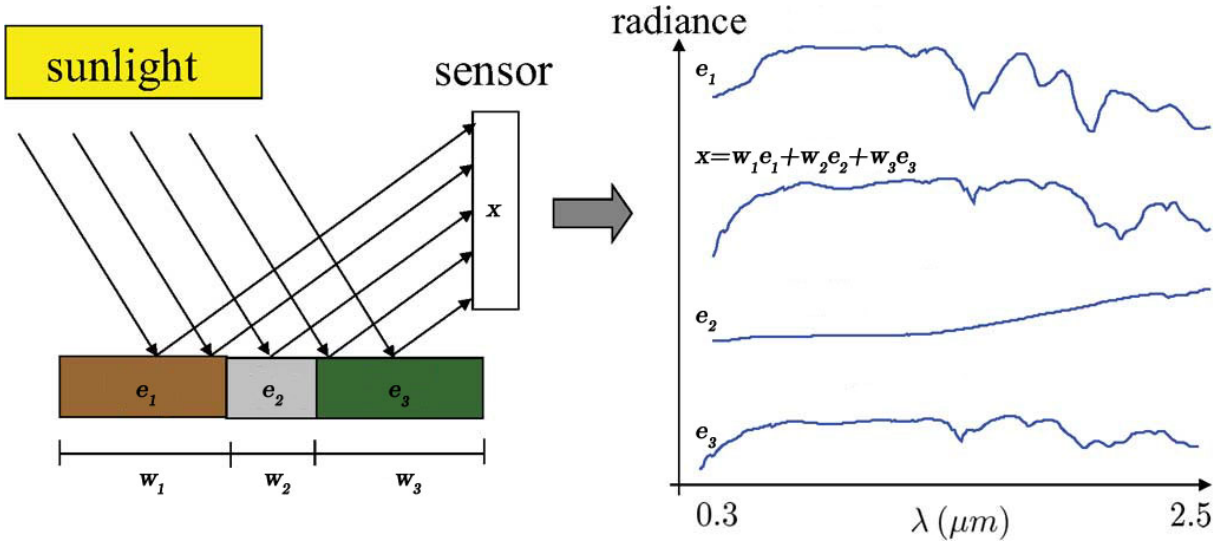


Figure 1.3: Linear mixing mechanics. Sunlight scattered by three materials denoted by $\{\boldsymbol{e}_1, \boldsymbol{e}_2, \boldsymbol{e}_3\}$ in a scene is incident on a sensor-pixel $\boldsymbol{x}$ that measures radiance in B bands.

$$\sum_{m=1}^{M} w_{nm} = |\boldsymbol{w}_n|_1 = 1 \tag{1.2}$$

together with non-negativity of w_{nm} is called the closure or convex geometry constraint [7]. This implies that $\boldsymbol{w}_n$ lies on a unit $(M-1)$-simplex.

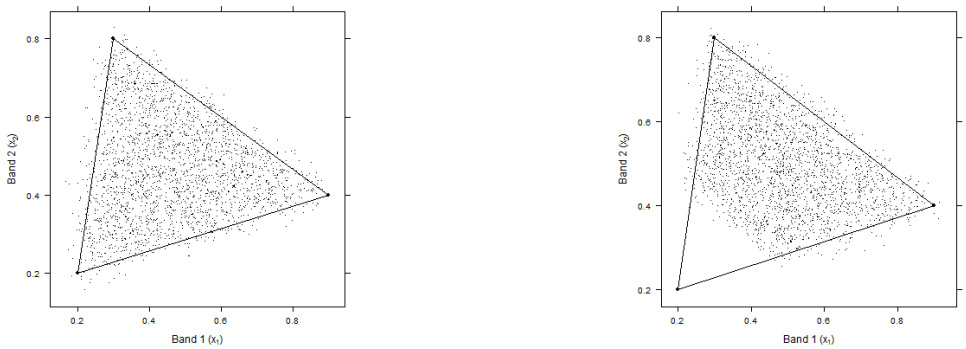
Eq. 1.1 can be further generalized to all pixels:

$$\boldsymbol{X} = \boldsymbol{W}\boldsymbol{E}^T + \boldsymbol{\mathcal{E}} \tag{1.3}$$

where matrix $\boldsymbol{X} = [\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N]^T \in \mathbb{R}_+^{N \times B}$ of the observed spectra and matrix $\boldsymbol{W} = [\boldsymbol{w}_1, \ldots, \boldsymbol{w}_N]^T \in \mathbb{R}_+^{N \times M}$ whose nth row represents the endmember concentration profile of nth sample $\boldsymbol{x}_i$.

Multidimensional simplex representation

The non-negativity and sum-to-unity constraints in Eq. 1.2 imply that $\boldsymbol{x}_n$ are contained within a simplex where the endmembers $\boldsymbol{e}_m$ form the vertices. This is shown in Fig 1.4. The fact that some $\boldsymbol{x}_n$ fall out of the simplex is due to noise. Furthermore, depending on the source, the image may not contain pure pixels (i.e. pixels constituted by only one endmember/material). Fig. 1.4b is such an example.



(a) Full mixing

(b) Partial mixing

Figure 1.4: 2500 pixels $\boldsymbol{x}_i \in \mathbb{R}^2_+$ represented in function of their band. The observed pixels may not span the entire simplex due to the mixing of the endmembers. This is shown in (b) where the bottom-left endmember is never available in pure form in the image. Such mixing is called partial.

1.2.2 Nonlinear mixing model

Nonlinear mixing is usually due to physical interactions (i.e. reflections and absorptions) between the light scattered by multiple materials in the scene. We do not consider nonlinear mixing in this thesis. This is partly justified by the fact that linear mixing is an acceptable approximation of the light scattering mechanisms in many real scenarios [8].

1.3 Hyperspectral unmixing process

The hyperspectral unmixing process is a step-by-step procedure of extracting the materials (i.e. endmember spectra) from hyperspectral images. This process usually consists of the following steps:

1. *Atmospheric correction* by which atmospheric effects are compensated by converting radiance to reflectance data. How this is done is shown in Section 5.1, using a real data example.

2. *Dimensionality reduction* is sometimes done for performance reasons due to the high number of bands. The most common way is principal component analysis (PCA). Sometimes other, more theoretically suitable techniques are used to reduce dimensionality. For example, the minimum noise fraction (MNF) transform which finds uncorrelated linear transformations that maximize the signal-to-noise ratio. We do not consider these techniques in this thesis and always use the complete dataset.

3. *Unmixing/Inversion* is the last step in which we identify the endmembers in the scene and the fractional abundances in each pixel. This step requires that we understand how endmember spectra are mixed in each pixel. In this thesis we assume the linear mixing model (LMM, cf. Section 1.2.1). Unmixing is thus the reverse process of Eq. 1.1 under non-negativity and sum-to-one constraints. If we consider the matrix formulation as in Eq. 1.3, then this process is a form of non-negative matrix factorization (NMF) with the additional sum-to-one constraint for the abundances.

The unmixing step is at the core of this thesis. It is thus appropriate to start with an overview of the present unmixing methods.

1.3.1 Unmixing methods

Bioucas-Dias e.a. [8] give a recent overview of unmixing algorithms. They divide the approaches in three general classes: geometric, statistical and sparse. The following gives a brief overview.

Geometrical methods

The algorithms in this class are based on the minimal volume spanned by the endmembers. Broadly speaking, they differ in whether pure pixels are assumed or not. A popular algorithm of the former kind is the N-FINDR algorithm [26]. It is based on the idea that the volume defined by a simplex formed by the purest pixels, is larger than any other volume defined by any other combination of pixels. This algorithm finds the set of pixels defining the largest volume by inflating a simplex inside the data. It performs well under full mixing, but not so much under partial mixing (cf. Fig. 2.1a), which brings us to the more general class of geometrical volume constrained approaches. These approaches seek an endmember matrix $\boldsymbol{E}$ that minimizes the volume of the simplex defined by its columns. Since the pure pixels are no longer assumed, this results in a more difficult nonconvex optimization problem. One such algorithm is the ICE algorithm [5], [6], [7] which is extensively discussed in Chapter 2.

Statistical methods

Statistical methods allow more precise modeling but come with a price of higher computational complexity. Especially the Bayesian approaches are popular. They have the ability to model statistical variability and to impose priors that can constrain solutions to physically meaningful ranges, and even regularize solutions. Common estimators are maximum a posteriori (MAP) estimators. Due to the complexity of resulting joint posterior, Markov chain Monte Carlo (MCMC) algorithms are used. These algorithms mainly differ by the choice of priors assigned to the unknown parameters.

In Chapter 3 we cover in great detail the BayesNMF-Vol [1], [3] algorithm which can be seen as a translation of the ICE algorithm into a Bayesian framework. Published in 2009, it is also one of the first Bayesian frameworks for hyperspectral unmixing. In the following years there was lot of research on this topic. For example this relatively new paper by Halimi e.a. [17] from 2015 is several layers more complex than the one described here. Halimi does not only incorporate the non-negativity and sum-to-one constraints into the model (which is an endeavor on its own), but goes further to model spatial information *and* spectral endmember variability (SEV) which is intrinsic [24] to remotely sensed spectral images.

Sparse regression methods

These methods are mostly semi-supervised in the sense that a large amount of spectra is fed into the model. The sparsity is related to the amount of endmembers in the solution (cf. [25]). There are also sparse unsupervised methods. We do not consider sparse methods in this thesis.

1.3.2 Spatial information

Most of de algorithms in the three classes ignore the spatial information that could improve the unmixing process. Bioucas-Dias e.a. [8] note though that incorporating this information into a model has recently motivated the development of a new class of algorithms. These algorithms exploit both the spatial and spectral features contained in an image. We do not cover these algorithms, but we do extend the ICE algorithm with spatial regulation. To our knowledge this has not been done before.

1.4 Applications

Higher spectral resolution of hyperspectral cameras enables material identification through spectroscopic analysis. This facilitates countless applications that require identifying materials in scenarios unsuitable for classical spectroscopic analysis. For example, hyperspectral cameras such as AVIRIS contribute significantly to earth observation and remote sensing. In Chapter 5 we base our analysis on one such real dataset.

The applications are not limited to remote sensing. Hyperspectral images can be found in any other domain where material identification through spectroscopic analysis is desired. Bioucas-Dias [8] refers to several domains such as food safety, pharmaceutical process monitoring and quality control, biomedical, industrial, biometric, and forensic applications.

1.5 Notation and conventions

Variable scalars are written in lowercase italics (a, b), column vectors in lowercase boldface italics ($\boldsymbol{a}$, $\boldsymbol{b}$) and matrices in uppercase boldface letters ($\boldsymbol{A}$, $\boldsymbol{B}$). Transposition is denoted by the superscript T ($\boldsymbol{a}^T$, $\boldsymbol{A}^T$).

For notational brevity, the columns or rows of a $N \times M$ matrix $\boldsymbol{A}$ are represented as a column vector $\boldsymbol{a}$ where the subscript denotes the origin: $\boldsymbol{a}_n := (\boldsymbol{A}_{n:})^T$ and $\boldsymbol{a}_m := \boldsymbol{A}_{:m}$. Similarly, $a_{nm} := \boldsymbol{A}_{n:m}$ is the (n, m)-th component of matrix $\boldsymbol{A}$. We use tilde ($\tilde{}$) to denote removed indices such that $\boldsymbol{A}_{\widetilde{a}:\widetilde{b}}$ denotes the $\boldsymbol{A}$ matrix with row a and column b removed. Similarly $\boldsymbol{a}_{n\widetilde{m}}$ denotes the vector $\boldsymbol{a}_n$ extracted from $\boldsymbol{A}$ with the m-th element removed.

The probability density function of a conventional distribution we denote in calligraphic style, like $\mathcal{N}$ for normal distribution. For other distributions we use the generic lowercase letter f which is determined by its argument or subscript, or both. For example, $f_{X|Y}(x|y) = f(x|y) = f_{X|Y}$. The latter form we use mainly in text to refer to a specific function, while the middle form is used in equations where we keep the capital letters such as $f(X|Y)$. The former and most correct notation is omitted for the sake of brevity.

Some special matrices and vectors include the identity matrix $\boldsymbol{I}_M$ with subscript denoting its order; a vector of ones $\mathbf{1}$, and the Moore-Penrose (MP) inverse of a matrix $\boldsymbol{A}$, which we write as $\boldsymbol{A}^\dagger$. The $\arg\min_{\boldsymbol{a}}$ of some function is denoted by $\boldsymbol{a}^*$.

Some important constants like the number of bands B, the number of endmembers M and the number of pixels N are written in uppercase italics. We use the same notation for important functions like the objective function L and the regularization functions V and S.

For matrix differential calculus we use the conventions layed out in [21]. Derivations are given only if they differ from the original papers.

1.6 Thesis organization and contributions

The following chapters are organized as follows. In Chapter 2 we explain Berman's [5] ICE algorithm, which is perhaps the most typifying of the geometric class of algorithms. This results in a few small improvements: The closed form solution for $\boldsymbol{E}^*$ in Section 2.3.4 and stabilization of the objective function in Section 2.4.1. Subsequently, in Section 2.5 we propose a spatial information extension for the ICE algorithm, which we dub ICE-S. The extra parameter ν of ICE-S allows for smoother abundance maps, but does not seem to have much effect on the resulting endmembers. We also notice multiple local minima in the objective function and try to map them by running ICE-S over its whole parameter range in Section 4.3.2.

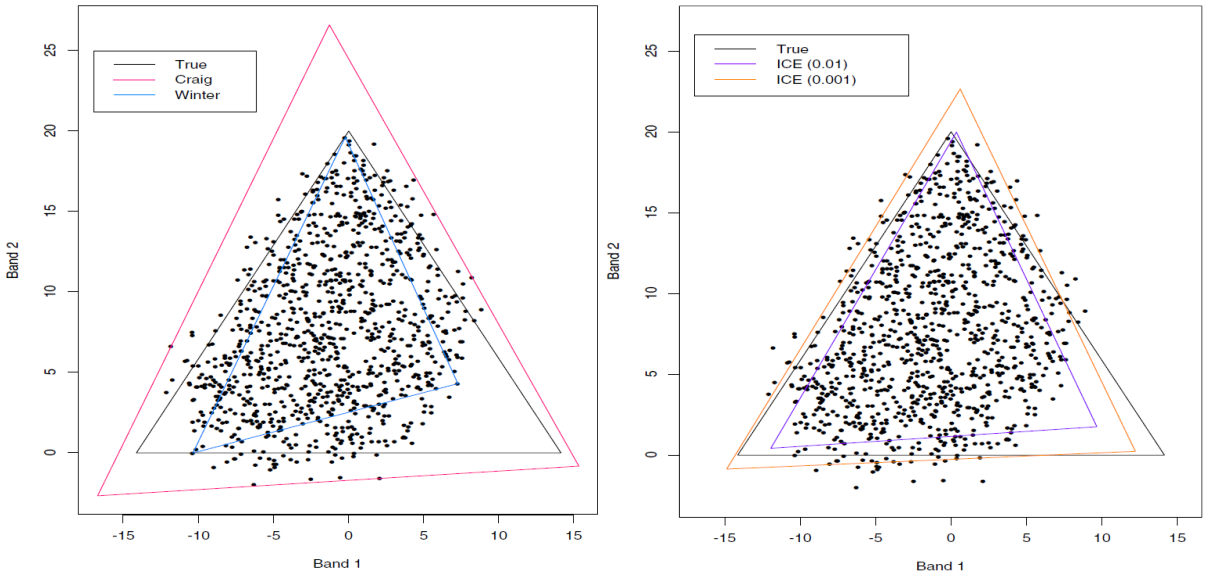
In Chapter 3 we explain the Bayesian framework and the MCMC solution for hyperspectral unmixing, which is mainly due to Arngren e.a. [1]. Their work, back in 2009, was one of the pioneering in the field and their insights are fundamental for all subsequent Bayesian unmixing models. That is why we put a considerable amount of effort into understand it.

Subsequently, we run these three algorithms on a synthetic (Chapter 4) and a real (Chapter 5) dataset and compare the results. We end with a small conclusion in Chapter 6. In the Appendices one will find all the derivations that are omitted from the corpus, and are not included in the original papers. There, one will also find the link and explanation of the source code that accompanies this thesis.

Chapter 2

ICE algorithm and extensions

Berman's iterated constrained endmembers (ICE) algorithm [5], [6], [7] is greatly inspired by Winters N-FINDR algorithm [26] which finds the M-simplex of *maximum* volume constrained to lie within the data cloud and Craig's algorithm [13] which finds the *minimum* volume M-simplex enclosing the data cloud. The ICE algorithm finds a M-simplex which has Craig's solution as its limiting case, and where the size of the simplex is controlled with a hyper-parameter μ. The comparison is shown in Fig. 2.1.



(a) (b)

Figure 2.1: Comparison of Winter's N-FINDR, Craig's algorithm and Berman's ICE algorithm for finding endmembers. Image taken from [7].

In Section 2.1 we explore the assumptions of the objective function central to the ICE algorithm. More precisely we derive it from LMM (Section 1.2.1) based on the Likelihood Principle. This is instructive as no such foundation has been found in the original papers. Once we learn that the objective function L is underspecified, we introduce in subsequent section a regularization term V which is proportional to the volume of the endmember simplex as a remedy. In Section 2.3 we explore in great detail every step in the so-called ALS algorithm which is used to minimize L. This Section also goes much deeper into implementation details than the original papers. Once done, we discuss two peculiarities of the ICE algorithm: omission of σ^2 and lurking imbalance due to large B – the number of

bands. And lastly in Section 2.5 we extend the ICE algorithm with spatial regularization and an extra hyper-parameter ν which controls it. To our knowledge this has not been tried before and is a creative contribution of this thesis.

2.1 Objective function

We start by rewriting $\boldsymbol{x}_n$ from Eq. 1.1 in each component x_{nb}:

$$\begin{aligned} x_{nb} &= \boldsymbol{w}_n^T \boldsymbol{e}_b + \epsilon \\ &\sim \mathcal{N}(\boldsymbol{w}_n^T \boldsymbol{e}_b, \sigma^2) \end{aligned} \tag{2.1}$$

The probability density function of x_{nb} is thus:

$$f(x_{nb} \mid \boldsymbol{w}_n, \boldsymbol{e}_b, \sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} \, e^{-\frac{1}{2\sigma^2}\left(x_{nb} - \boldsymbol{w}_n^T \boldsymbol{e}_b\right)^2} \tag{2.2}$$

Assuming that all pixel components x_{nb} are independent we can write the joint distribution as follows:

$$\begin{aligned} f(x_{11}, \ldots, x_{NB} \mid \boldsymbol{E}, \boldsymbol{W}, \sigma^2) &= f(x_{11} \mid \boldsymbol{w}_1, \boldsymbol{e}_1, \sigma^2) \times \cdots \times f(x_{NB} \mid \boldsymbol{w}_N, \boldsymbol{e}_B, \sigma^2) \\ &= \prod_{n=1}^{N} \prod_{b=1}^{B} f(x_{nb} \mid \boldsymbol{w}_n, \boldsymbol{e}_b, \sigma^2) \\ &= \ell(\boldsymbol{E}, \boldsymbol{W}, \sigma^2 \mid \boldsymbol{X}) \end{aligned} \tag{2.3}$$

Now, from the Likelihood Principle all we have to do is maximize $\ell(\boldsymbol{E}, \boldsymbol{W}, \sigma^2 \mid \boldsymbol{X})$. To facilitate the computation (i.e. to get rid of the product which is likely to cause numerical instability for large N and to get rid of the exponent), one usually uses the log-likelihood function $\mathcal{L} = \log(\ell)$. This is justified since the logarithm is a monotonically increasing function and thus preserves the maximum.

$$\begin{aligned} \mathcal{L}(\boldsymbol{E}, \boldsymbol{W}, \sigma^2 \mid \boldsymbol{X}) &= log\left[\ell(\boldsymbol{E}, \boldsymbol{W}, \sigma^2 \mid \boldsymbol{X})\right] \\ &= \sum_{n=1}^{N} \sum_{b=1}^{B} log\left[f(x_{nb} \mid \boldsymbol{w}_n, \boldsymbol{e}_b, \sigma^2)\right] \\ &= -NB \log\left(\sqrt{2\pi\sigma^2}\right) - \frac{1}{2\sigma^2} \sum_{n=1}^{N} \sum_{b=1}^{B} \left(x_{nb} - \boldsymbol{w}_n^T \boldsymbol{e}_b\right)^2 \end{aligned} \tag{2.4}$$

Now we are ready to define the objective loss function which we wish to minimize.

$$\begin{aligned} \underset{\boldsymbol{E}, \boldsymbol{W}, \sigma^2}{\arg\max} \, \mathcal{L}(\boldsymbol{E}, \boldsymbol{W}, \sigma^2 \mid \boldsymbol{X}) &= \underset{\boldsymbol{E}, \boldsymbol{W}, \sigma^2}{\arg\min} \; -\mathcal{L}(\boldsymbol{E}, \boldsymbol{W}, \sigma^2 \mid \boldsymbol{X}) \\ &= \underset{\boldsymbol{E}, \boldsymbol{W}, \sigma^2}{\arg\min} \; NB \log\left(\sqrt{2\pi\sigma^2}\right) + \frac{1}{2\sigma^2} \sum_{n=1}^{N} \sum_{b=1}^{B} \left(x_{nb} - \boldsymbol{w}_n^T \boldsymbol{e}_b\right)^2 \end{aligned} \tag{2.5}$$

Note that Berman [5] seems to disregard σ^2 – without explanation[1] – and only considers $\boldsymbol{E}$ and $\boldsymbol{W}$. In that case σ^2 is considered given and the additive constant $NB\log(\sqrt{2\pi\sigma^2})$ and positive factor $1/2\sigma^2$ can be removed since they have no impact on arg min.

$$\begin{aligned}\arg\max_{\boldsymbol{E},\boldsymbol{W}} \mathcal{L}(\boldsymbol{E},\boldsymbol{W} \mid \boldsymbol{X}) &= \arg\min_{\boldsymbol{E},\boldsymbol{W}} \sum_{n=1}^{N}\sum_{b=1}^{B}\left(x_{nb} - \boldsymbol{w}_n^T\boldsymbol{e}_b\right)^2 \\ &= \arg\min_{\boldsymbol{E},\boldsymbol{W}} L(\boldsymbol{W},\boldsymbol{E},\sigma^2 \mid \boldsymbol{X})\end{aligned} \tag{2.6}$$

The function $L(\boldsymbol{W},\boldsymbol{E})$ we call the loss function or the objective function. At this point it is instructive to show various ways in which the objective function can be rewritten, which will prove useful in later manipulations.

$$\begin{aligned}L(\boldsymbol{E},\boldsymbol{W} \mid \boldsymbol{X}) &= \sum_{n}^{N}\sum_{b}^{B}(x_{nb} - \boldsymbol{w}_n^T\boldsymbol{e}_b)^2 = \sum_{n}^{N}\sum_{b}^{B}(x_{nb} - \hat{x}_{nb})^2 \\ &= \|\boldsymbol{X} - \boldsymbol{W}\boldsymbol{E}^T\|_F^2 = tr\left[(\boldsymbol{X} - \boldsymbol{W}\boldsymbol{E}^T)^T(\boldsymbol{X} - \boldsymbol{W}\boldsymbol{E}^T)\right] \\ &= \sum_{n}^{N}(\boldsymbol{x}_n - \boldsymbol{E}^T\boldsymbol{w}_n)^T(\boldsymbol{x}_n - \boldsymbol{E}^T\boldsymbol{w}_n) \\ &= \sum_{b}^{B}(\boldsymbol{x}_b - \boldsymbol{W}\boldsymbol{e}_b)^T(\boldsymbol{x}_b - \boldsymbol{W}\boldsymbol{e}_b)\end{aligned} \tag{2.7}$$

2.2 Regularized objective function

Note that objective function L in Eq. 2.7 is ill posed. It will result in the same minimum value as long as all the pixels are within the simplex formed by the vertices of the endmembers. In that case $x_{nb} - \hat{x}_{nb} = 0\ \forall n, b$ and thus $L = 0$. That is why Berman [5] proposes a penalized version:

$$L_{reg}(\boldsymbol{E},\boldsymbol{W}) = \frac{(1-\mu)}{N}L + \mu V \tag{2.8}$$

Here V acts as a regularization term and is proportional to the volume of the endmember simplex. μ acts as a hyper-parameter of the model. Division by N is to make sure that L is independent of the amount of pixels used. To make V independent of the dimension of the simplex, the summed variance per band b of all the endmembers is used instead.

[1] Actually, that is a bit too quick. Berman uses the ICE algorithm mainly on data which is transformed by means of the minimum noise fraction (MNF) transform. This makes the errors uncorrelated between MNF bands and with variance 1. The data can still be spatially correlated. Furthermore, MNF transform tends to produce errors that look Gaussian for low dimensions. [6]

$$
\begin{aligned}
V &= \sum_{b=1}^{B} \widehat{Var}(\boldsymbol{e}_b) \\
&= \sum_{b=1}^{B} \frac{\sum_{m=1}^{M} \left[e_{bm} - \frac{\sum_{m=1}^{M} e_{bm}}{M} \right]^2}{M-1} \\
&= \sum_{b=1}^{B} \frac{\boldsymbol{e}_b^T (\boldsymbol{I}_M - \frac{\mathbf{1}\mathbf{1}^T}{M}) \boldsymbol{e}_b}{M-1} \\
&= \frac{SSD}{M(M-1)}
\end{aligned}
\tag{2.9}
$$

Note that Eq. 2.9 gives a direct relation between V and the sum of squared distances (SSD) of the endmember simplex. A worked out derivation of this relationship can be found in the Appendix Eq. A.2. Note further that the SSD is also proportional to the size of the simplex. In summary, the advantage of V to the volume is threefold: it is independent of M, proportional to the volume and cheaper to compute.

Since L_{reg} is approximately independent of the sample size N and the number of endmembers M, it allows for just one value of μ for all datasets. Berman [5], [7] reports that a value of 0.01 to 0.05 usually gives reasonable solutions after analyzing several dozen real-world datasets.

2.3 Minimization

Minimization of L_{reg} is not straightforward. The dimensionality is extremely large. Suppose for example one has an image of N=1000 pixels with B=100 bands and one seeks M=5 endmembers. Since our function L_{reg} has matrices $\boldsymbol{E} \in \mathbb{R}_+^{B \times M}$ and $\boldsymbol{W} \in \mathbb{R}_+^{N \times M}$ as arguments, we would have a total of $B \times M + N \times M = 100500$ parameters to find! And even if we had a closed-form solution, we still would need to make sure the non-negativity and sum-to-one constraints are met.

The approach taken by Berman is to split the function in smaller subproblems and solve them one by one. Berman only gives the closed form solution for $\boldsymbol{e}_b^*$ and mentions that $\boldsymbol{w}_n^*$ can be found using quadratic programming as explained in [22, Chapter 16]. In the following sections we delve into details of how this is actually done.

2.3.1 Coordinate descent

Coordinate descent [27] is based on the idea that the minimization of a multi-variable objective function $f(\boldsymbol{x})$ can be achieved by minimizing it along one component of $\boldsymbol{x}$ with respect to the remaining components at each iteration step. One thus reduces the difficult full minimization problem to a cycle of much simpler single-variable problems. One starts with a random initial value:

$$
\boldsymbol{x}^0 = (x_1^0, \ldots, x_n^0)
$$

Then one iteratively solves the single component optimization problems

$$
x_i^{k+1} = \underset{y \in \mathbb{R}}{\arg\min}\ f(x_1^{k+1}, \ldots, x_{i-1}^{k+1}, y, x_{i+1}^k, \ldots, x_n^k)
$$

where the superscript $k+1$ denotes the current iteration cycle. At each step one does a line search for which $f(\boldsymbol{x}^k) \geq f(\boldsymbol{x}^{k+1})$ always holds. Overall we get

$$f(\boldsymbol{x}^0) \geq f(\boldsymbol{x}^1) \geq f(\boldsymbol{x}^2) \geq \dots.$$

No significant improvement between two iteration cycles implies a stationary point is reached.

2.3.2 Block coordinate descent

An interesting feature of the objective function $L_{reg}(\boldsymbol{W}, \boldsymbol{E})$ in Eq. 2.8 is that if we only look at $L_{reg}(\boldsymbol{w}_n)$ and $L_{reg}(\boldsymbol{e}_b)$ with other parameters fixed, then the function in these parameters is convex. Thus we can minimize efficiently these blocks of parameters one at a time. Hence the name *block* coordinate descent. Now let us take a closer look at these functions and their form.

2.3.3 Minimize $L_{reg}(\boldsymbol{w}_n)$ with fixed $\boldsymbol{E}$ and $\boldsymbol{W}_{\tilde{n}:}$

Assuming fixed $\boldsymbol{E}$ we can rewrite L_{reg} from Eq. 2.8 in function of $\boldsymbol{w}_n$ as follows:

$$\begin{aligned} L_{reg}(\boldsymbol{W}) &= \sum_{n=1}^{N} \left[\frac{(1-\mu)}{N} ||\boldsymbol{x_n} - \boldsymbol{E}\boldsymbol{w}_n||_2^2 + \frac{\mu V}{N} \right] \\ &= \sum_{n=1}^{N} L_{n,reg}(\boldsymbol{w}_n) \end{aligned} \tag{2.10}$$

This shows that each $L_{n,reg}(\boldsymbol{w}_n)$ can be minimized separately since $\boldsymbol{w}_n$ are independent of each other. Now we rewrite $L_{n,reg}(\boldsymbol{w}_n)$ to quadratic standard form:

$$\begin{aligned} L_{n,reg}(\boldsymbol{w}_n) &= \frac{(1-\mu)}{N}(\boldsymbol{x_n} - \boldsymbol{E}\boldsymbol{w}_n)^T(\boldsymbol{x_n} - \boldsymbol{E}\boldsymbol{w}_n) + \frac{\mu V}{N} \\ &= \left(\frac{(1-\mu)\boldsymbol{x}_n^T\boldsymbol{x}_n}{N} + \frac{\mu V}{N} \right) - \boldsymbol{w}_n^T \frac{2(1-\mu)\boldsymbol{E}^T\boldsymbol{x}_n}{N} \\ &\quad + \frac{1}{2}\boldsymbol{w}_n^T \frac{2(1-\mu)\boldsymbol{E}^T\boldsymbol{E}}{N} \boldsymbol{w}_n \\ &= C^{te} + \boldsymbol{w}_n^T\boldsymbol{a} + \frac{1}{2}\boldsymbol{w}_n^T\boldsymbol{H}\boldsymbol{w}_n \end{aligned} \tag{2.11}$$

Note that the constant term plays no role in minimization. Thus we need to solve the following constrained quadratic problem:

$$\begin{aligned} &\arg\min_{\boldsymbol{w}_n} \ \boldsymbol{w}_n^T\boldsymbol{a} + \frac{1}{2}\boldsymbol{w}_n^T\boldsymbol{H}\boldsymbol{w}_n \\ &\text{subject to } \sum_{m=1}^{M} w_{nm} = 1 \text{ and } w_{nm} \geq 0, \ m = 1, \dots, M. \end{aligned} \tag{2.12}$$

For numerical stability (in case N is large) we can remove the factor $2(1-\mu)/N$ from both $\boldsymbol{a}$ and $\boldsymbol{H}$ without influencing the minimum.

Minimizing $L_{reg}(\boldsymbol{w}_n)$ under constraints

Nocedal [22, Ch. 16] describes two approaches for solving such a problem. The first is the *active set method* which is inspired by the simplex method from linear programming. In the simplex method, a solution is searched for along the vertices of the feasible polytope. Each vertex of such a polytope is a point where all equality, and a subset of inequality constraints, are active. An inequality constraint such as $w_{nm} \geq 0$ is considered active if it strictly equals zero at that point.

Quadratic problems are more complicated since the solution is not necessarily one of the vertices, nor does it necessarily lie on the boundary of the polytope. Nonetheless, searching along the boundary of the polytope is the essence of the active set algorithms. For this particular problem we have implemented the *primal* active set method for convex quadratic problem as described in [22, p. 472].

A second approach is the so called *interior-point method*. This method is polynomial time and approaches the solution through the interior of the feasible polytope rather than working its way around the boundary as the active set method does.

Convexity of $L_{reg}(\boldsymbol{w}_n)$

$L_{n,reg}(\boldsymbol{w}_n)$ is convex and our minimum is global if $\boldsymbol{H}$ is positive semidefinite. The minimum is unique if $\boldsymbol{H}$ is positive definite. We know that $\boldsymbol{x}^T\boldsymbol{E}^T\boldsymbol{E}\boldsymbol{x} = (\boldsymbol{E}\boldsymbol{x})^T\boldsymbol{E}\boldsymbol{x} = ||\boldsymbol{E}\boldsymbol{x}||_2^2 \geq 0$ and thus that $\boldsymbol{H}$ is positive semidefinite. If $\ker(\boldsymbol{E}) = \{\boldsymbol{x} \mid \boldsymbol{E}\boldsymbol{x} = \boldsymbol{0}\} = \{\boldsymbol{0}\}$, then the columns of $\boldsymbol{E}$ are linearly independent and strict inequality holds for $\boldsymbol{x} \neq \boldsymbol{0}$. Thus $\boldsymbol{H}$ is positive definite only if the endmembers (i.e. columns) of $\boldsymbol{E}$ are linearly independent! Since $\boldsymbol{E} \in \mathbb{R}_+^{B\times M}$, the latter can *only* be if $M \leq B$. So by having $M \ll B$ we increase the chance of having an unique solution for $\boldsymbol{w}_n$. The case where $M > B$ is common in synthetic images where we limit $B = 2$ and $M = 3$ for representational purposes as in Fig. 1.4. In such cases $\boldsymbol{H}$ is positive semidefinite. If this were an unconstrained problem, then we would be sure that multiple solutions for $\boldsymbol{w}_n$ exist. In the constrained case this is more complicated. The set of minima for semi-definite $\boldsymbol{H}$ might as well lie completely outside of the feasible set, so we might still have an unique solution under active constraints! In other words, in the constrained case, the requirements for a unique solution are more loose. Specifically [22, Lemma 16.1, p.452], an active set of constraints $\mathcal{A}$ with its corresponding matrix $\boldsymbol{A}$ has a unique solution if $\boldsymbol{Z}^T\boldsymbol{H}\boldsymbol{Z}$ is positive definite where $\ker(\boldsymbol{A}) = \boldsymbol{Z}$. So even if $\boldsymbol{H}$ is positive semi-definite, we still have a chance that the constrained solution is unique.

Implementation peculiarities

Note that we *could* run the active set algorithm on the full matrix $\boldsymbol{W}$ since $L_{reg}(\boldsymbol{W})$ is quadratic and convex, but this is dangerous. Nocedal [22, p. 388] notes that the complexity of an active set method is exponential, i.e. in the linear case there exist pathological cases for which the simplex method visits every single vertex before reaching the optimal point. Therefore it is safer to limit the dimensionality of the problem to just $L_{reg}(\boldsymbol{w}_n)$.

Another implementation peculiarity is the case when $\boldsymbol{Z}^T\boldsymbol{H}\boldsymbol{Z}$ is positive semi-definite. In that case the Karush-Kuhn-Tucker (KKT) matrix is singular and thus there are multiple solutions for $\boldsymbol{w}_n$. A general approach for solving such singular case is to use the Moore-Penrose (MP) pseudo-inverse which gives a solution $\boldsymbol{x}_{min}$ having the smallest norm

of all possible solutions $\boldsymbol{x}$. An important thing to note here is that $\boldsymbol{x}_{min} = (\boldsymbol{w}_n, \boldsymbol{\lambda})$, where $\boldsymbol{\lambda}$ is the vector of Lagrange multipliers. So although $\boldsymbol{x}_{min}$ has the minimal norm, that does not imply that $\boldsymbol{w}_n$ also has the minimal norm of all possible solutions.

Another issue with MP inverse is its calculation. It is usually computed using singular value decomposition (SVD). Thus $\boldsymbol{A}^{\dagger} = \boldsymbol{V}\boldsymbol{D}^{-1}\boldsymbol{U}^T$ where $\boldsymbol{D}^{-1}$ is the diagonal matrix of reciprocal eigenvalues. Very small eigenvalues cause MP inverse instability. On the other hand, removing small but non-zero eigenvalues causes inexact MP inverse which eventually results in solutions that might not exactly match the constraints. So although rare, these situations occur, especially when $M > B$ such as in synthetic images, and might cause errors of order 1e-8 or even higher.

During the process of implementation and testing we have found an inconsistency in the active set algorithm for convex QP as described by Nocedal [22, p.472]. Where it says "obtain W_{k+1} by adding one of the blocking constraints to W_k" it should be "obtain W_{k+1} by adding *the* blocking constraint which corresponds with the smallest α_k to W_k". This is essential as picking a random blocking constraint when there are multiple blocking constraints will, result in a wrong minimum for a well-chosen initial feasible point x_0.

2.3.4 Minimize $L_{reg}(\boldsymbol{e}_b)$ with fixed $\boldsymbol{W}$ and $\boldsymbol{E}_{\tilde{b}:}$

Assuming fixed $\boldsymbol{W}$, using $||\boldsymbol{X} - \boldsymbol{W}\boldsymbol{E}^T||_F^2 = \sum_{b=1}^{B} ||\boldsymbol{x_b} - \boldsymbol{W}\boldsymbol{e}_b||_2^2$ and substituting the result for V from Eq. 2.9 into Eq. 2.8 we get:

$$\begin{aligned} L_{reg}(\boldsymbol{E}) &= \sum_{b=1}^{B} \left[\frac{(1-\mu)}{N} ||\boldsymbol{x_b} - \boldsymbol{W}\boldsymbol{e}_b||_2^2 + \mu \frac{\boldsymbol{e}_b^T (M\boldsymbol{I}_M - \boldsymbol{1}\boldsymbol{1}^T)\boldsymbol{e}_b}{M(M-1)} \right] \\ &= \sum_{b=1}^{B} L_{b,reg}(\boldsymbol{e}_b) \end{aligned} \tag{2.13}$$

This shows that each $L_{b,reg}(\boldsymbol{e}_b)$ can be minimized separately since $\boldsymbol{e}_b$ are independent of each other. Now we rewrite $L_{b,reg}(\boldsymbol{e}_b)$ in quadratic standard form:

$$\begin{aligned} L_{b,reg}(\boldsymbol{e}_b) = &\frac{(1-\mu)\boldsymbol{x}_b^T\boldsymbol{x}_b}{N} - \boldsymbol{e}_b^T \frac{2(1-\mu)\boldsymbol{W}^T\boldsymbol{x}_b}{N} \\ &+ \frac{1}{2}\boldsymbol{e}_b^T \left(\frac{2(1-\mu)\boldsymbol{W}^T\boldsymbol{W}}{N} + \frac{2\mu(\boldsymbol{I}_M - \boldsymbol{1}\boldsymbol{1}^T/M)}{(M-1)} \right) \boldsymbol{e}_b \\ = &\, C^{te} + \boldsymbol{e}_b^T\boldsymbol{c} + \frac{1}{2}\boldsymbol{e}_b^T\boldsymbol{G}\boldsymbol{e}_b \end{aligned} \tag{2.14}$$

Convexity of $L_{reg}(\boldsymbol{e}_b)$

The Hessian matrix $\boldsymbol{G}$ is symmetric since it is a sum of two symmetric matrices. $\boldsymbol{G}$ is also positive semi-definite. To see this we have to look at its matrix parts. $\boldsymbol{e}^T\boldsymbol{W}^T\boldsymbol{W}\boldsymbol{e} = ||\boldsymbol{W}\boldsymbol{e}||_2^2 \geq 0$ so $\boldsymbol{W}^T\boldsymbol{W}$ is positive semi-definite. That $(\boldsymbol{I}_M - \boldsymbol{1}\boldsymbol{1}^T/M)$ is also positive semi-definite follows from $\boldsymbol{x}^T(\boldsymbol{I}_M - \boldsymbol{1}\boldsymbol{1}^T/M)\boldsymbol{x} = \boldsymbol{x}^T(\boldsymbol{I}_M - \boldsymbol{v}\boldsymbol{v}^T)\boldsymbol{x} = \|\boldsymbol{x}\|_2^2 - \|\boldsymbol{v}^T\boldsymbol{x}\|_2^2 = \|\boldsymbol{x}\|_2^2\|\boldsymbol{v}\|_2^2 - \|\boldsymbol{v}^T\boldsymbol{x}\|_2^2 \geq 0$ where $\boldsymbol{v} = \boldsymbol{1}/\sqrt{M}$ and where the last inequality is due to the Cauchy-Schwarz inequality. The consequence is that $L_{b,reg}(\boldsymbol{e}_b)$ is convex. In practice, $\boldsymbol{G}$ is positive definite because to be semi-definite the columns of $\boldsymbol{W}$ have to be dependent and since $N \gg M$ this is highly unlikely. At the same time the $\ker(\boldsymbol{W}^T\boldsymbol{W})$ has to be the

same as $\ker(\boldsymbol{I}_M - \mathbf{1}\mathbf{1}^T/M) = \{\boldsymbol{x} \mid x_1 = x_2 = \cdots = x_M\}$ which is even more unlikely. So in the calculation we can safely use $\boldsymbol{G}^{-1}$ instead of $\boldsymbol{G}^\dagger$.

Closed solution for $\boldsymbol{e}_b$

The convexity of the problem thus assures us that solving for $\boldsymbol{e}_b$ in equation $\partial L_{b,reg}(\boldsymbol{e}_b)/\partial \boldsymbol{e}_b^T = \boldsymbol{c}^T + \boldsymbol{e}_b^T \boldsymbol{G} = 0$ will result in the global minimum:

$$\underset{\boldsymbol{e}_b}{\arg\min}\ L_{b,reg}(\boldsymbol{e}_b) = \left[\boldsymbol{W}^T\boldsymbol{W} + \lambda\left(\boldsymbol{I}_M - \frac{\mathbf{1}\mathbf{1}^T}{M}\right)\right]^{-1} \boldsymbol{W}^T\boldsymbol{x}_b \tag{2.15}$$

with $\lambda = N\mu/[(1-\mu)(M-1)]$.

But are we not a bit too fast? What about the constraint $e_{bm} \geq 0, \forall b, m$? It is a peculiarity in Berman's ICE algorithm that it is not enforced. But while it is not enforced, it is very likely to be satisfied since the endmembers will be within the pixel cloud. And since the pixel components are all greater or equal to zero, it is safe to assume that the endmember components will also be.

Closed solution for $\boldsymbol{E}$

Although Berman [5], [6], [7] does not mention it explicitly, $\arg\min_{\boldsymbol{E}}\ L_{reg}(\boldsymbol{E})$ can easily be solved in one go by replacing $\boldsymbol{x}_b$ with $\boldsymbol{X}$ and taking the transpose of the result in Eq. 2.15, resulting in:

$$\underset{\boldsymbol{E}}{\arg\min}\ L_{reg}(\boldsymbol{E}) = \boldsymbol{X}^T\boldsymbol{W}\left[\boldsymbol{W}^T\boldsymbol{W} + \lambda\left(\boldsymbol{I}_M - \frac{\mathbf{1}\mathbf{1}^T}{M}\right)\right]^{-1} \tag{2.16}$$

This result follows directly from Eq. 2.15 once we recognize that the $L_{b,reg}(\boldsymbol{e}_b)$ are independent. A more formal derivation based on matrix calculus laws from [21] is given in Appendix A.2.

2.3.5 The ICE algorithm

In the previous two sections we computed the parameters $\boldsymbol{E}^{k+1}$ and $\boldsymbol{W}^{k+1}$ for one whole cycle of a coordinate descent. We thus have that $L_{reg}(\boldsymbol{E}^k, \boldsymbol{W}^k) \geq L_{reg}(\boldsymbol{E}^{k+1}, \boldsymbol{W}^{k+1})$. Since at each step we are minimizing a quadratic problem, this approach is also called alternating least squares (ALS). We now describe the full algorithm.

The iteration process is stopped once successive L_{reg} values are close enough. Berman [6] reports that a ratio larger than 0.99999 gives sufficiently stable endmember estimates. Now the obvious question is: does this converge to a global minimum?

2.3.6 Non-Convexity of L_{reg}

The global minimum is achieved by coordinate descent only if the objective function is convex. Unfortunately, L_{reg} as defined in Eq. 2.8 is not convex [6]. There is no formal proof for this statement. But one can demonstrate this by using different starting points in the ICE algorithm. For example, if the starting endmember simplex is outside of the data cloud, the solution will sometimes tend to equal endmembers in the center of the data cloud. This seems to be a local minimum. An other local minimum seems to be when the endmembers lie on one straight line within the data cloud. From our conducted

Algorithm 1 (ICE algorithm)

$k \leftarrow 1$
$L^{(k)} \leftarrow$ maximum machine number
Generate a random matrix $\boldsymbol{W}^{(k)}$
repeat
  $k \leftarrow k + 1$
  $\boldsymbol{E}^{(k)} \leftarrow \arg\min_{\boldsymbol{E}} \; L_{reg}(\boldsymbol{E} \mid \boldsymbol{W}^{(k-1)})$ as in Eq. 2.16
  for $n = 1$ to N **do**
    $\boldsymbol{w} \leftarrow \arg\min_{\boldsymbol{w}_n} \; L_{n,reg}(\boldsymbol{w}_n \mid \boldsymbol{E}^{(k)})$ as in Eq. 2.12
    $\boldsymbol{W}_{n:}^{(k)} \leftarrow (\boldsymbol{w})^T$
  end for
  $L^{(k)} \leftarrow L_{reg}(\boldsymbol{W}^{(k)}, \boldsymbol{E}^{(k)})$ as in Eq. 2.8
until $L^{(k)}/L^{(k-1)} \geq tol$
return $\boldsymbol{E}^{(k)}$ and $\boldsymbol{W}^{(k)}$ as solution;

experiments on synthetic data, it also seems better to start the algorithm with a random $\boldsymbol{W}$ – instead of a random $\boldsymbol{E}$ – to end up at the global minimum.

Berman's experience is that endmembers which are pure or almost pure are consistently found, almost independently of the starting point, in the sense that there is little variation in the solutions. Less pure endmembers are found with greater variation.[2]

Sampling the solutions while initializing the algorithm with random $\boldsymbol{W}$ also shows that the chance to end up in a local minima increases with decreasing μ. We will return to this point in Chapter 4 (Fig. 4.2a), when we sample the solutions over the whole parameter space of μ.

2.4 Model peculiarities

2.4.1 L dependence on B

During our testing the $\arg\min_{\boldsymbol{w}_n} L_{n,reg}(\boldsymbol{w}_n)$ proved instable for bands $B \geq 100$. This was due to $\boldsymbol{E}^T\boldsymbol{E}$ in effect having very large coefficients in case $\boldsymbol{X}$ was not normalized. This in turn resulted in the KKT matrix having very large and very small eigenvalues (due to small coefficients of active constraint matrix $\boldsymbol{A}$), making the KKT matrix computationally singular. This results in numerical instability in the solution of $\boldsymbol{w}_n^*$ with the effect of not complying to the sum-to-one constraint. That is because the small eigenvalue, corresponding to the equality constraint, gets dropped in the MP inverse calculation because it is too small compared to the others.

The instability can be traced back to L and V being both independent of M and N, but not of B. So the larger the B, the larger L_{reg} will tend to be. This can be mitigated by dividing L_{reg} in Eq. 2.8 with B. Our new improved objective function thus becomes:

$$L_{imp,B}(\boldsymbol{W}, \boldsymbol{E}) = \frac{(1-\mu)}{NB} L + \frac{\mu V}{B} \tag{2.17}$$

The effect is that this divides the factors $\boldsymbol{a}$ and $\boldsymbol{H}$ in Eq. 2.12 by B mitigating the large eigenvalues without affecting the solution.

[2] cf. comments at: `https://www.researchgate.net/publication/229884791_ICE_A_new_method_for_the_multivariate_curve_resolution_of_hyperspectral_images`

$$\boldsymbol{a} = \frac{\boldsymbol{E}^T \boldsymbol{x}_n}{B}, \qquad \boldsymbol{H} = \frac{\boldsymbol{E}^T \boldsymbol{E}}{B} \tag{2.18}$$

Note that this adjustment has no effect on $\boldsymbol{E}^*$.

2.4.2 Disregard of σ^2

We already noted that Berman does not consider σ^2 in the likelihood Eq. 2.4. For the sake of example, let us add it into L_{reg} and see what happens.

$$\begin{aligned} L_{imp,\sigma^2}(\boldsymbol{E}, \boldsymbol{W}, \sigma^2 \mid \boldsymbol{X}) &= -\frac{(1-\mu)}{NB}\mathcal{L}(\boldsymbol{E}, \boldsymbol{W}, \sigma^2 \mid \boldsymbol{X}) + \frac{\mu V}{B} \\ &= \frac{(1-\mu)}{2\sigma^2 NB}\|\boldsymbol{X} - \boldsymbol{W}\boldsymbol{E}^T\|_F^2 + (1-\mu)\log\left(\sqrt{2\pi\sigma^2}\right) + \frac{\mu V}{B} \end{aligned} \tag{2.19}$$

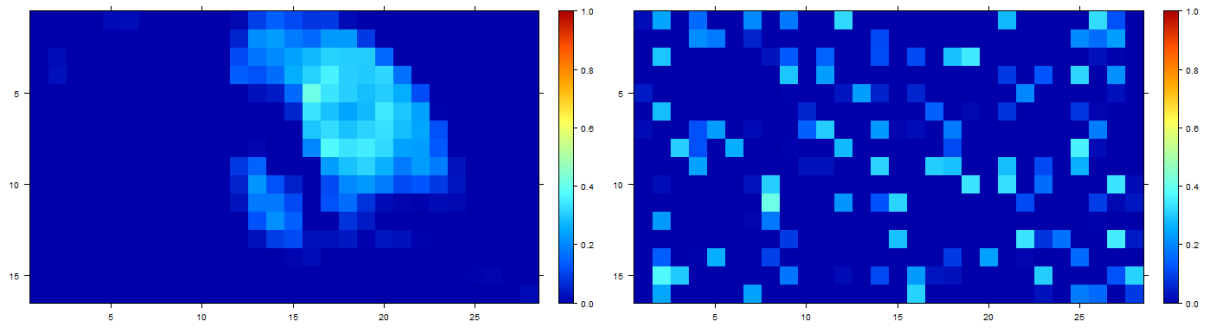
To minimize this we have to make slight adjustments to the ICE algorithm. $\boldsymbol{W}^*$ remains the same. For computing $\boldsymbol{E}^*$ we use the Eq. 2.16 with adjusted $\lambda = 2\sigma^2 N\mu/[(1-\mu)(M-1)]$. And finally, $\sigma^{2*} = \|\boldsymbol{X} - \boldsymbol{W}\boldsymbol{E}^T\|_F^2/(NB)$.

Although all the subproblems are convex, the function L_{reg} is definitely not. From practical experiments, it seems that there are many local minima.

2.5 Extension with spatial regulation (ICE-S)

2.5.1 Motivation

Suppose you were given two abundance maps of some mineral as in Fig. 2.2. Both abundance maps contain the same data, but the abundances are ordered differently. If these two images were both the output for the same material, one would tend to think that in the right image the algorithm was somehow malconfigured. The left image shows some configuration, order or a hidden manifold. In the right image the abundances are randomly spread throughout the image which makes it less likely.



(a) (b)

Figure 2.2: Spatial abundance of a mineral Alunite. The left image (a) is a copy of Fig. 5.7a – the original abundance. The right image (b) has the same abundance data randomly ordered.

We want a regularization that favors the left image, and disfavors the right image. There are various ways one could do this. The approach chosen here is to model each pixel with its four neighboring pixels as cliques and compute their variance.

2.5.2 New regularization function S

Thus the variance of an abundance w_{nm} and its neighbors on band m we define as:

$$S_{nm} = Var(\boldsymbol{U}_{:m}^{[n]}) \text{ where } \boldsymbol{U}^{[n]} = \begin{bmatrix} \boldsymbol{w}_{n-H}^T \\ \boldsymbol{w}_{n-1}^T \\ \boldsymbol{w}_{n}^T \\ \boldsymbol{w}_{n+1}^T \\ \boldsymbol{w}_{n+H}^T \end{bmatrix} \tag{2.20}$$

where H denotes the height of the image and is important for the calculation of adjacent pixels. Thus the S_{nm} measures the variance between abundances $\{w_{n-H,m},\ w_{n-1,m},\ w_{n,m},\ w_{n+1,m},\ w_{n+H,m}\}$ as shown in Fig. 2.3. The vector $\boldsymbol{U}_{:m}^{[n]}$ contains all the necessary information around abundance w_{nm} for this calculation. The total variance is simply the sum of all S_{nm}. Thus $S = \sum_{n=1}^{N} \sum_{m=1}^{M} S_{nm}$.

$w_{0,m}$				
$\vdots$		$w_{n-1,m}$		
	$w_{n-H,m}$	$w_{n,m}$	$w_{n+H,m}$	
		$w_{n+1,m}$		$\vdots$
				$w_{N,m}$

Figure 2.3: A 5×5 grid composed of endmember abundances for band m. The central 5 abundances are the components of $\boldsymbol{U}_{:m}^{[n]}$ defined in Eq. 2.20.

2.5.3 New objective function L_{VS}

The proposed new regularized objective function:

$$L_{VS}(\boldsymbol{W}, \boldsymbol{E} \mid \boldsymbol{X}) = \frac{(1-\mu)}{NB} ||\boldsymbol{X} - \boldsymbol{W}\boldsymbol{E}^T||_F^2 + \mu \left[\frac{\nu}{B} V + \frac{(1-\nu)}{NM} S \right] \tag{2.21}$$

where ν acts as a regularization parameter that distributes the regularization between volume V and spatial information S. Note that we also divide S by N and M, which is to make it of same magnitude as V.

2.5.4 Minimization

The fact that we use variance ensures the quadratic form of $S_n(\boldsymbol{W})$. This fits nicely into the existing ICE algorithm.

Derivation of $L_{VS}(\boldsymbol{W})$

We start from Eq. 2.21. Since we are using coordinate descent, we focus on solving $\boldsymbol{w}_n$ one at a time, just as we did previously. This time we have S which we need to write as a function of $\boldsymbol{w}_n$. The exact derivation is given in Appendix A.3.

$$S_n(\boldsymbol{w}_n) = \sum_{k \in \mathcal{K}_n} \frac{\boldsymbol{w}_n^T \boldsymbol{w}_n}{K_k} - 2\boldsymbol{w}_n^T \sum_{k \in \mathcal{K}_n} \frac{(\boldsymbol{U}^k_{\tilde{n}:})^T \mathbf{1}}{K_k(K_k - 1)} + C^{te} \tag{2.22}$$

The set $\mathcal{K}_n = \{n - H, n - 1, n, n + 1, n + H\}$ is defined here as the set of all adjacent numbers of abundance-pixel n, including n itself. Note that cardinality $K_k = |\mathcal{K}_k|$ is a variable and not always equal to 5. Using this result and Eq. 2.21 we can derive $L_{n,VS}$ for use in minimization:

$$\begin{aligned} L_{n,VS}(\boldsymbol{w}_n) &= \frac{(1-\mu)}{NB} ||\boldsymbol{x_n} - \boldsymbol{E}\boldsymbol{w}_n||_2^2 + \frac{\mu}{N}\left(\frac{\nu}{B}V + \frac{(1-\nu)}{M}S_n\right) \\ &= C^{te} - \boldsymbol{w}_n^T \left[2\frac{(1-\mu)\boldsymbol{E}^T\boldsymbol{x}_n}{NB} + 2\frac{\mu(1-\nu)}{NM}\sum_{k \in \mathcal{K}} \frac{(\boldsymbol{U}^k_{\tilde{n}:})^T \mathbf{1}}{K_k(K_k-1)}\right] \\ &\quad + \frac{1}{2}\boldsymbol{w}_n^T \left[2\frac{(1-\mu)\boldsymbol{E}^T\boldsymbol{E}}{NB} + 2\frac{\mu(1-\nu)}{NM}\sum_{k \in \mathcal{K}} \frac{\boldsymbol{I}_M}{K_k}\right] \boldsymbol{w}_n \\ &= C^{te} + \boldsymbol{w}_n^T \boldsymbol{a}_S + \frac{1}{2}\boldsymbol{w}_n^T \boldsymbol{H}_S \boldsymbol{w}_n \end{aligned} \tag{2.23}$$

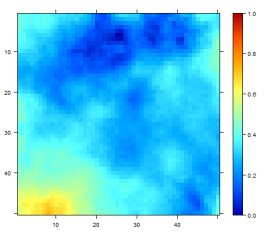
The factor $2/N$ can be removed since it does not affect the outcome. Now that we have $\boldsymbol{a}_S$ and $\boldsymbol{H}_S$ we can use the same minimization technique as for Eq. 2.12 to minimize it.

2.5.5 Derivation of $L_{VS}(\boldsymbol{E})$

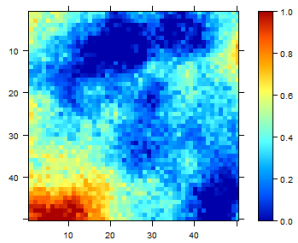
There are hardly any changes here compared to Eq. 2.15. S depends on $\boldsymbol{W}$ so it is a constant and has no effect on $\boldsymbol{E}^*$. The only thing that gets added to Eq. 2.15 is ν such that $\lambda = N\mu\nu/[(1-\mu)(M-1)]$.

2.6 Evaluation

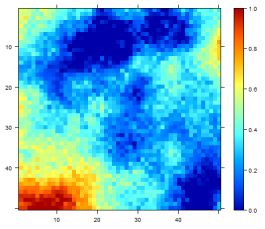
Fig. 2.4 shows how spatial regulation influences the abundance maps. The more spatial regulation we apply through the $(1-\nu)$ hyper-parameter – while holding $\mu\nu$ (i.e. volume regularization) constant – the more smooth the abundances. We do not see any improvement in endmember estimation, as Fig. 2.4d shows.



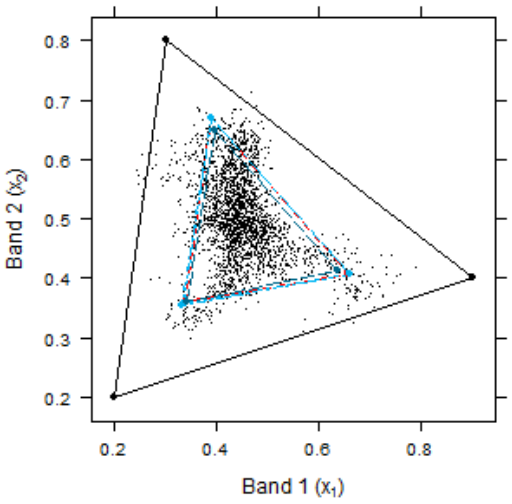
(a) Actual



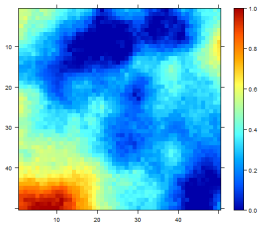
(b) $\text{ICE}_{0.01}$



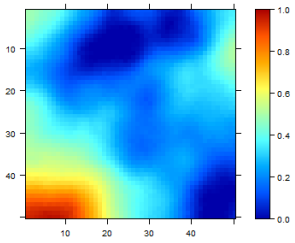
(c) $\text{ICE}_{0.02}$ $\text{S}_{50\%}$



(d) Endmembers



(e) $\text{ICE}_{0.1}$ $\text{S}_{90\%}$



(f) $\text{ICE}_{0.5}$ $\text{S}_{98\%}$

Figure 2.4: These images are taken from analysis in Chapter 4, Fig 4.3. Fig. (a) is the real abundance map. Fig. (b) is estimated with the ICE algorithm using $\mu = 0.01$. Note that 0.01 is suggested by Berman [7] as the optimal parameter based on analysis of various real datasets. Fig. (c) is estimated with the spatially extended ICE algorithm using $\mu = 0.02$ and $\nu = 0.5$ which results in weight 0.01 for V and 0.50 for S. Fig. (e-f) are similar, where weight for V is held constant with $\mu\nu = 0.01$. These images are merely to illustrate the effect of spatial regularization on the abundance maps. Fig. (d) compares the resulting endmembers (red for $\text{ICE}_{0.01}$, light to dark blue for the amount of regularization $\text{S}_{50\%}$, $\text{S}_{90\%}$, $\text{S}_{98\%}$ respectively while holding $\mu\nu = 0.01$) with the true endmembers (black). The endmember corresponding with the abundance maps is the bottom left one.

2.7 Conclusion

In this chapter we have seen how the objective function of the ICE algorithm can be derived from the LMM model using the Likelihood Principle. Subsequently we have thoroughly analyzed the minimization process of the objective function using coordinate descent. We made one contribution in this regard by dividing the objective function with B, thus improving its stability – especially for cases with unnormalized data. We also calculated the closed form solution for $\boldsymbol{E}^*$. Subsequently, driven by the idea that there is some order in abundances maps of materials, we have proposed and implemented a spatial regularization extension to the ICE algorithm, called ICE-S. This algorithm has one extra hyper-parameter $1 - \nu$ which controls the spatial regularization. The higher

we set this parameter, the smoother the abundance maps. The results also show that this parameter has no significant effect on endmember estimates – compared to the ICE algorithm under the same volume regularization weight.

Chapter 3

BayesNMF-Vol algorithm

A common approach to volume-constrained hyperspectral unmixing is to build a regularized objective function (like L_{reg} in Eq. 2.8) and solve for $\boldsymbol{E}^*$ by numerical optimization. Another approach is to build a probability model and treat the extraction of endmembers $\boldsymbol{E}$ as a Bayesian inference problem. This approach is based on [1], [3] and requires the definition of a likelihood, priors and most of all, a sound Belief (or Bayesian) Network.

Our primary focus here, like in the previous chapter, is understanding the principles and limitations of the approach. That means that we go further into details than the original papers. One essential point that is lacking in the original papers is the definition of a sound Belief Network. This step is essential because it defines the form of the joint distribution which in turn allows us to simplify calculations with Bayes rule. That is why we start this chapter with the definition of a Belief Network that underpins the BayesNMF-Vol model.

Then, in Section 3.2 we define the likelihood and priors in accordance with the original papers. We are mainly interested in the maximum a posteriori (MAP) estimators for this model. One reason is because there is a relationship between the MAP estimators and the maximum likelihood estimators (MLE) as used in the ICE Algorithm, which in turn allow direct comparison. We touch briefly on this relationship in Section 3.3.

The need for a posterior distribution calls for more complex MCMC techniques such as a Gibbs sampler on which we elaborate in Section 3.4. The original papers are accompanied with a technical paper [2] in which all the needed conditional distributions are derived. We refer to those when needed. In other cases where our own derivations differ (slightly), we put them in the Appendix. This is most notably the case for the $\boldsymbol{W}$-sampler in Section 3.4.4 where we are confronted with a degenerate distribution. In Section 3.2.4 we also look into the derivation and implementation of Jeffreys' prior which is a limiting case of the Inverse-Gamma distribution.

Lastly, in Section 3.5 we investigate some model peculiarities. The most notable and somewhat unexpected one is the intrinsic volume regularization of this Bayesian approach, due to the choice of the $\boldsymbol{W}$-prior. Several authors [16], [3] have a mathematical explanation for this phenomenon, while we give a more conceptual one.

3.1 Belief Network for the Bayesian model

We start our model building from the LMM from Section 1.2.1, Eq. 1.3. Since we are interested in the model parameters $\boldsymbol{E}$, $\boldsymbol{W}$ and σ^2 we can write the posterior using the Bayes rule:

$$f(\boldsymbol{E}, \boldsymbol{W}, \sigma^2|\boldsymbol{X}) = \frac{f(\boldsymbol{X}|\boldsymbol{E}, \boldsymbol{W}, \sigma^2)f(\boldsymbol{E}, \boldsymbol{W}, \sigma^2)}{f(\boldsymbol{X})} \tag{3.1}$$

The immediate problem that we have here is how to define $f(\boldsymbol{E}, \boldsymbol{W}, \sigma^2)$. The original papers assume implicitly that $f(\boldsymbol{E}, \boldsymbol{W}, \sigma^2) = f(\boldsymbol{E})f(\boldsymbol{W})f(\sigma^2)$ which in turn allows us to write the joint probability as:

$$f(\boldsymbol{X}, \boldsymbol{E}, \boldsymbol{W}, \sigma^2) = f(\boldsymbol{X}|\boldsymbol{E}, \boldsymbol{W}, \sigma^2)f(\boldsymbol{E})f(\boldsymbol{W})f(\sigma^2) \tag{3.2}$$

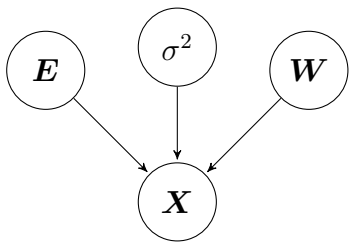
which in turn is equivalent with a directed acyclic graph (DAG) such as the one in Fig. 3.1a in which the Markov property is assumed [4]. Such graphs in the Bayesian framework are called Belief Networks. They are a convenient tool for describing direct influence and conditional independence assumptions between different variables. The BayesNMF-Vol model builds on these assumptions with extra hyper-parameters as depicted in Fig. 3.1b. According to graphical model theory [4], this DAG corresponds with the following joint distribution:

$$\begin{aligned} &f(\boldsymbol{X},\boldsymbol{E}, \boldsymbol{W}, \sigma^2, \alpha, \beta, \gamma) \\ &\quad = f(\boldsymbol{X}|\boldsymbol{E}, \boldsymbol{W}, \sigma^2)f(\boldsymbol{E}|\gamma)f(\boldsymbol{W})f(\sigma^2|\alpha, \beta)f(\alpha)f(\beta)f(\gamma) \end{aligned} \tag{3.3}$$

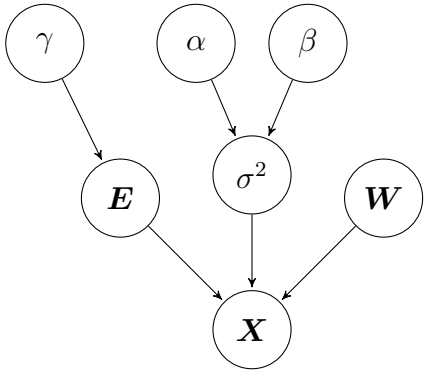
Specifically, in case of BayesNMF-Vol, the parameters α, β and γ are assumed given and thus the joint distribution can further be simplified to:

$$\begin{aligned} &f_{BayesNMF-Vol}(\boldsymbol{X},\boldsymbol{E}, \boldsymbol{W}, \sigma^2|\alpha, \beta, \gamma) \\ &\qquad = \frac{f(\boldsymbol{X}, \boldsymbol{E}, \boldsymbol{W}, \sigma^2, \alpha, \beta, \gamma)}{f(\alpha, \beta, \gamma)} \\ &\qquad = f(\boldsymbol{X}|\boldsymbol{E}, \boldsymbol{W}, \sigma^2)f(\boldsymbol{E}|\gamma)f(\boldsymbol{W})f(\sigma^2|\alpha, \beta) \end{aligned} \tag{3.4}$$

The last equality is due to $f(\alpha, \beta, \gamma) = f(\alpha)f(\beta)f(\gamma)$.



(a) LMM



(b) BayesNMF-Vol

Figure 3.1: (a) Belief Network for the linear mixing model (LMM) as defined in Eq. 1.3. (b) Belief Network for the BayesNMF-Vol algorithm. Note that $\boldsymbol{E}$, $\boldsymbol{W}$ and σ^2 are assumed directly independent of each other.

3.2 The partial probability density functions

We are now set to define each of the partial density functions in Eq. 3.4. The definitions are taken from Arngren [1].

3.2.1 The likelihood $f_{\boldsymbol{X}|\boldsymbol{E},\boldsymbol{W},\sigma^2}$

Following the LMM data model (cf. Eq. 1.3) we already derived the likelihood in Eq. 2.3. So without falling into repetition we write:

$$f(\boldsymbol{X}|\boldsymbol{E},\boldsymbol{W},\sigma^2) = \prod_{n=1}^{N}\prod_{b=1}^{B} \frac{\exp\left[-\frac{1}{2\sigma^2}\left(x_{nb}-\boldsymbol{w}_n^T\boldsymbol{e}_b\right)^2\right]}{\sqrt{2\pi\sigma^2}} \tag{3.5}$$

3.2.2 The endmember prior $f_{\boldsymbol{E}|\gamma}$

Our first concern with the endmember prior is that it should disallow negative endmember components. Thus

$$f(\boldsymbol{E}) \propto \prod_{b=1}^{B}\prod_{m=1}^{M} \mathbb{I}\left[e_{bm} \geq 0\right] \tag{3.6}$$

Our second concern is that the prior should encourage the simplex spanned by the estimated endmembers to be small. Arngren's reasoning for this inclusion is because of the functional equivalence between the MAP and ML estimators as described later in section 3.3. From this, it appears that the MAP estimator would suffer the same fate as objective function L from Eq. 2.7. And thus the volume regularization is added:

$$f(\boldsymbol{E}|\gamma) \propto e^{-\gamma V(\boldsymbol{E})} \prod_{b}^{B}\prod_{m}^{M} \mathbb{I}\left[e_{bm} \geq 0\right] \tag{3.7}$$

Note that the reason for putting the volume V in the exponent is partly due to sampling reasons: the Gibbs $\boldsymbol{E}$-sampler is Gaussian in that case (cf. section 3.4.5). Furthermore, this facilitates the correspondence between the MAP end ML estimators as show in section 3.3.

3.2.3 The abundances prior $f_{\boldsymbol{W}}$

The minimal requirements – as required by the LMM in Section 1.2.1 – for the abundances are the non-negativity and sum-to-one constraints. Thus:

$$f(\boldsymbol{W}) \propto \prod_{n=1}^{N}\left[\mathbb{I}\left[\|\boldsymbol{w}_n\|_1 = 1\right]\prod_{m=1}^{M}\mathbb{I}\left[w_{nm} \geq 0\right]\right] \tag{3.8}$$

Although this prior seems as least informative while incorporating the constraints, we will see later in Section 3.5.1 that it also prefers smaller endmember simplexes.

Other more complex priors could also be formulated. For example, if we acknowledge that relatively few endmembers are likely to be mixed in each pixel, then that would lead us to sparse or highly kurtotic priors for $\boldsymbol{W}$.

3.2.4 The variance prior $f_{\sigma^2|\alpha,\beta}$

The Inverse-Gamma distribution for variance σ^2 is an obvious choice since it is conjugate to the normal distribution.

$$
\begin{aligned}
f(\sigma^2|\alpha,\beta) &= \mathcal{IG}(\sigma^2|\alpha,\beta) \\
&= \frac{\beta^\alpha}{\Gamma(\alpha)}(\sigma^2)^{-\alpha-1}\exp\left(-\frac{\beta}{\sigma^2}\right)
\end{aligned} \tag{3.9}
$$

Arngren [1] notes that as $\alpha \to 0$ and $\beta \to 0$ the Inverse-Gamma approaches the Jeffreys prior. Now, why would one use that instead of the uniform distribution? The uniform distribution, sometimes called the Bayes-Laplace prior, has its roots in Thomas Bayes stating: "When the probability of a simple event is unknown, we may suppose all values as equally likely."

Jeffreys prior on the other hand has a more theoretical foundation [18]. Suppose person A and person A' are observing events from a normal distribution but use a different scale measure sp that the following equality holds: $\sigma' = c\sigma$ where $c \in \mathbb{R}_0^+$ a constant. Since their location measure is the same, they end up observing events x and x' related in the following way: $x' = c(x-\mu)+\mu$. Assuming they have the same prior information their prior probabilities are the same: $f'(\sigma,\mu) = f(\sigma,\mu)$. Assuming further independent parameters we have $f'(\sigma) = f(\sigma)$. From the change of variables theorem we can further deduce (cf. Appendix B.1) that $f'(\sigma') = f(\sigma)/c$. Combining these results gives the functional equation $cf'(c\sigma) = f(\sigma)$ which has $f(\sigma) = C^{te}/\sigma$ as the general solution. Note that this density is improper and is called the Jeffreys prior for *scale* parameters.

From a practical point of view the results do not differ much between these two priors. Nonetheless, giving more weight to smaller σ^2 values than larger ones seems more logical. And lastly, since $\mathcal{IG}$ is conjugate to $\mathcal{N}$ the σ^2-sampler is $\mathcal{IG}$ instead of a left truncated $\mathcal{N}$.

Now that we have defined all the constituting probability density functions of the posterior distribution, we can define the posterior.

3.2.5 The posterior $f_{\boldsymbol{E},\boldsymbol{W},\sigma^2|\boldsymbol{X},\alpha,\beta,\gamma}$

We derive the posterior:

$$
\begin{aligned}
&f(\boldsymbol{E},\boldsymbol{W},\sigma^2|\boldsymbol{X},\alpha,\beta,\gamma) \\
&\quad= \frac{f_{BayesNMF-Vol}(\boldsymbol{X},\boldsymbol{E},\boldsymbol{W},\sigma^2|\alpha,\beta,\gamma)}{f(\boldsymbol{X}|\alpha,\beta,\gamma)} \\
&\quad= \frac{f(\boldsymbol{X}|\boldsymbol{E},\boldsymbol{W},\sigma^2)f(\boldsymbol{E}|\gamma)f(\boldsymbol{W})f(\sigma^2|\alpha,\beta)}{\int_{\boldsymbol{E}}\int_{\boldsymbol{W}}\int_{\sigma^2} f(\boldsymbol{X}|\boldsymbol{E},\boldsymbol{W},\sigma^2)f(\boldsymbol{E}|\gamma)f(\boldsymbol{W})f(\sigma^2|\alpha,\beta)\,\mathrm{d}\sigma^2\,\mathrm{d}\boldsymbol{W}\,\mathrm{d}\boldsymbol{E}} \\
&\quad\propto f(\boldsymbol{X}|\boldsymbol{E},\boldsymbol{W},\sigma^2)f(\boldsymbol{E}|\gamma)f(\boldsymbol{W})f(\sigma^2|\alpha,\beta) \\
&\quad\propto \prod_{n=1}^{N}\prod_{b=1}^{B}\frac{\exp\left[-\frac{1}{2\sigma^2}\left(x_{nb}-\boldsymbol{w}_n^T\boldsymbol{e}_b\right)^2\right]}{\sqrt{2\pi\sigma^2}} \\
&\qquad\times e^{-\gamma V(\boldsymbol{E})}\prod_{b=1}^{B}\prod_{m=1}^{M}\mathbb{I}\left[e_{bm}\geq 0\right] \\
&\qquad\times \prod_{n=1}^{N}\left[\mathbb{I}\left[\|\boldsymbol{w}_n\|_1 = 1\right]\prod_{m=1}^{M}\mathbb{I}\left[w_{nm}\geq 0\right]\right] \\
&\qquad\times \frac{\beta^\alpha}{\Gamma(\alpha)}(\sigma^2)^{-\alpha-1}\exp\left(-\frac{\beta}{\sigma^2}\right)
\end{aligned} \tag{3.10}
$$

Note that the integral in the denominator is just a constant.

3.3 Relation between MAP end ICE estimators

Several authors [3], [8] have noted an interesting connection between the geometrical approaches such as the ICE algorithm and the Bayesian MAP estimator:

$$\begin{aligned}(\hat{\boldsymbol{E}}, \hat{\boldsymbol{W}})_{MAP} &= \arg\max_{\boldsymbol{E},\boldsymbol{W}} f_{\boldsymbol{E},\boldsymbol{W}}(\boldsymbol{E}, \boldsymbol{W}|\boldsymbol{X}, \sigma^2, \alpha, \beta, \gamma) \\ &= \arg\max_{\boldsymbol{E},\boldsymbol{W}} f_{\boldsymbol{X}}(\boldsymbol{X}|\boldsymbol{E}, \boldsymbol{W}, \sigma^2) f_{\boldsymbol{E}}(\boldsymbol{E}|\gamma) f_{\boldsymbol{W}}(\boldsymbol{W}) \\ &= \arg\min_{\boldsymbol{E},\boldsymbol{W}} -\log f_{\boldsymbol{X}}(\boldsymbol{X}|\boldsymbol{E}, \boldsymbol{W}, \sigma^2) - \log f_{\boldsymbol{E}}(\boldsymbol{E}|\gamma) - \log f_{\boldsymbol{W}}(\boldsymbol{W}) \\ &= \arg\min_{\boldsymbol{E},\boldsymbol{W}} ||\boldsymbol{X} - \boldsymbol{W}\boldsymbol{E}^T||_F^2 + \gamma V + C^{te}\end{aligned} \tag{3.11}$$

which is similar in form to L_{reg} in Eq. 2.7. In the ICE algorithm, the estimates are obtained by minimizing a two-term objective function L_{reg} where $-\log f_{\boldsymbol{X}}(\boldsymbol{X}|\boldsymbol{E}, \boldsymbol{W}, \sigma^2)$ plays the role of a data fitting criterion and $-\log f_{\boldsymbol{E}}(\boldsymbol{E}|\gamma) - \log f_{\boldsymbol{W}}(\boldsymbol{W})$ of a penalization. Conversely, from a Bayesian perspective, assigning prior distributions to $\boldsymbol{E}$ and $\boldsymbol{W}$ is a convenient way to ensure physical constraints inherent to the observation model. Thus such geometrical algorithms have statistical foundation as Bayesian MAP estimators.

3.4 The BayesNMF-Vol sampler algorithm

3.4.1 Choice of a sampler

Sampling from the posterior is not straightforward. The first difficulty is that we do not know the exact distribution since the denominator is not known. The Metropolis-Hastings (MH) algorithm does not require the denominator, and thus is a natural choice. Problem is the amount of parameters. The MH algorithm is highly dependent on a good proposal function, without which the acceptance rate is halved with each extra dimension. A viable solution is a Gibbs sampling approach which allows sampling at univariate level – at the expense of meticulous calculations.

3.4.2 The Gibbs sampler

The general Gibbs sampling procedure is described in [9]. We use it here to sample ($\boldsymbol{E}$, $\boldsymbol{W}$, σ^2) from the posterior $f_{\boldsymbol{E},\boldsymbol{W},\sigma^2|\boldsymbol{X},\alpha,\beta,\gamma}$. The specific procedure is defined in Algorithm 2. The algorithm generates a Gibbs sequence of random variables $((\boldsymbol{E}^{(0)}, \boldsymbol{W}^{(0)}, \sigma^{2(0)}), \ldots, (\boldsymbol{E}^{(k)}, \boldsymbol{W}^{(k)}, \sigma^{2(k)}))$ that constitute a Markov chain for which the stationary distribution is the posterior in which we are interested.

Algorithm 2 (BayesNMF-Vol sampling algorithm)

Initialize $\boldsymbol{E}^{(0)}$ and $\boldsymbol{W}^{(0)}$ in a random feasible way.
for $i = 1, 2, \ldots$ to k **do**
$\sigma_1^{2(i)} \sim f_{\sigma^2}(\sigma^2 | \boldsymbol{E}^{(i-1)}, \boldsymbol{W}^{(i-1)}, \boldsymbol{X}, \alpha, \beta, \gamma)$
$w_{11}^{(i)} \sim f_{\boldsymbol{W}}(w_{11} | \sigma_1^{2(i)}, \boldsymbol{E}^{(i-1)}, w_{12}^{(i-1)}, \ldots, w_{NM}^{(i-1)}, \boldsymbol{X}, \alpha, \beta, \gamma)$
$\vdots$
$w_{NM}^{(i)} \sim f_{\boldsymbol{W}}(w_{NM} | \sigma_1^{2(i)}, \boldsymbol{E}^{(i-1)}, w_{11}^{(i)}, \ldots, w_{NM-1}^{(i)}, \boldsymbol{X}, \alpha, \beta, \gamma)$
$e_{11}^{(i)} \sim f_{\boldsymbol{E}}(e_{11} | \sigma_1^{2(i)}, \boldsymbol{W}^{(i)}, e_{12}^{(i-1)}, \ldots, e_{BM}^{(i-1)}, \boldsymbol{X}, \alpha, \beta, \gamma)$
$\vdots$
$e_{BM}^{(i)} \sim f_{\boldsymbol{E}}(e_{BM} | \sigma_1^{2(i)}, \boldsymbol{W}^{(i)}, e_{11}^{(i)}, \ldots, e_{BM-1}^{(i)}, \boldsymbol{X}, \alpha, \beta, \gamma)$
end for
return $(\boldsymbol{E}, \boldsymbol{W}, \sigma^2)$ as Gibbs sample sequence.

Advantages to the MH algorithm are that we do not need to tune the proposal distribution and we have no inefficiency due to rejected proposals. The downside is that the Gibbs-progress can be stalled by highly correlated parameters.

In the following sections we derive each of the conditional distributions needed by the Gibbs sampler and define appropriate ways of sampling. These derivations are based on the technical paper [2]. In case our own derivations are different or include omitted steps, we put them into the Appendix.

3.4.3 Sampling σ^2

Because we have chosen a conjugate prior for the noise variance, its conditional distribution has the same functional form as the prior: An Inverse-Gamma.

$$\begin{aligned} & f(\sigma^2 | \boldsymbol{E}, \boldsymbol{W}, \boldsymbol{X}, \alpha, \beta, \gamma) \\ & \quad = \frac{f_{BayesNMF-Vol}(\boldsymbol{X}, \boldsymbol{E}, \boldsymbol{W}, \sigma^2 | \alpha, \beta, \gamma)}{\int_{\sigma^2} f_{BayesNMF-Vol}(\boldsymbol{X}, \boldsymbol{E}, \boldsymbol{W}, \sigma^2 | \alpha, \beta, \gamma) \, \mathrm{d}\sigma^2} \\ & \quad = \frac{f(\boldsymbol{X} | \boldsymbol{E}, \boldsymbol{W}, \sigma^2) f(\sigma^2 | \alpha, \beta)}{\int_{\sigma^2} f(\boldsymbol{X} | \boldsymbol{E}, \boldsymbol{W}, \sigma^2) f(\sigma^2 | \alpha, \beta) \, \mathrm{d}\sigma^2} \\ & \quad = f(\sigma^2 | \boldsymbol{E}, \boldsymbol{W}, \boldsymbol{X}, \alpha, \beta) \\ & \quad = \mathcal{IG}(\sigma^2 | \bar{\alpha}, \bar{\beta}) \end{aligned} \tag{3.12}$$

Note that in the second equation $f(\boldsymbol{E}|\gamma)$ and $f(\boldsymbol{W})$ can be removed from the integral since they are just constants. Thus $f_{\sigma^2 | \boldsymbol{E}, \boldsymbol{W}, \boldsymbol{X}, \alpha, \beta}$ is independent of γ given the other parameters. The Inverse-Gamma parameters, as calculated in [2], are given by

$$\bar{\alpha} = \alpha + \frac{1}{2} NB \tag{3.13}$$

$$\bar{\beta} = \beta + \frac{1}{2} \| \boldsymbol{X} - \boldsymbol{W} \boldsymbol{E}^T \|_F^2 \tag{3.14}$$

Parameter $\bar{\alpha}$ controls the shape and is *constant*. The larger it is, the more the density will shift to 0 and the less variable the outcome. $\bar{\beta}$ on the other hand controls the scale and is directly proportional to the error. The more pixels lie out of the simplex, the larger it is, and thus the larger the variance.

Note that $\bar{\alpha}$ and $\bar{\beta}$ most likely will never be equal to zero even if we set $\alpha = 0$ and $\beta = 0$ to simulate Jeffreys prior.

3.4.4 Sampling $\boldsymbol{W}$

$$\begin{aligned} f(\boldsymbol{W}|\boldsymbol{E}, \sigma^2, \boldsymbol{X}) \\ &= \prod_{n=1}^{N} f(\boldsymbol{w}_n|\boldsymbol{E}, \sigma^2, \boldsymbol{x}_n) \\ &\propto \prod_{n=1}^{N} \left[\mathcal{N}(\boldsymbol{w}_n|\boldsymbol{\mu}_n, \boldsymbol{\Sigma}_{\boldsymbol{w}}) \, \mathbb{I}\left[\|\boldsymbol{w}_n\|_1 = 1\right] \prod_{m=1}^{M} \mathbb{I}\left[w_{nm} \geq 0\right] \right] \end{aligned} \tag{3.15}$$

where the parameters are given by

$$\boldsymbol{\Sigma}_{\boldsymbol{w}}^{-1} = \frac{\boldsymbol{E}^T \boldsymbol{E}}{\sigma^2} \tag{3.16}$$

$$\boldsymbol{\mu}_n = (\boldsymbol{E}^T \boldsymbol{E})^{\dagger} \boldsymbol{E}^T \boldsymbol{x}_n \tag{3.17}$$

The derivation is worked out in Appendix B.3. First note that we can concentrate on $f_{\boldsymbol{w}_n|\boldsymbol{E},\sigma^2,\boldsymbol{x}_n}$ instead of the whole $f_{\boldsymbol{W}|\boldsymbol{E},\sigma^2,\boldsymbol{X}}$. Next, note that $f_{\boldsymbol{w}_n|\boldsymbol{E},\sigma^2,\boldsymbol{x}_n}$ is multivariate Gaussian, constrained on a unit-simplex. Fig. 3.2 shows how this probability density might look like for $\boldsymbol{w}_n \in \mathbb{R}^2$. Note that in this case, the feasible region is a straight line since the unit-simplex is 1-dimensional.

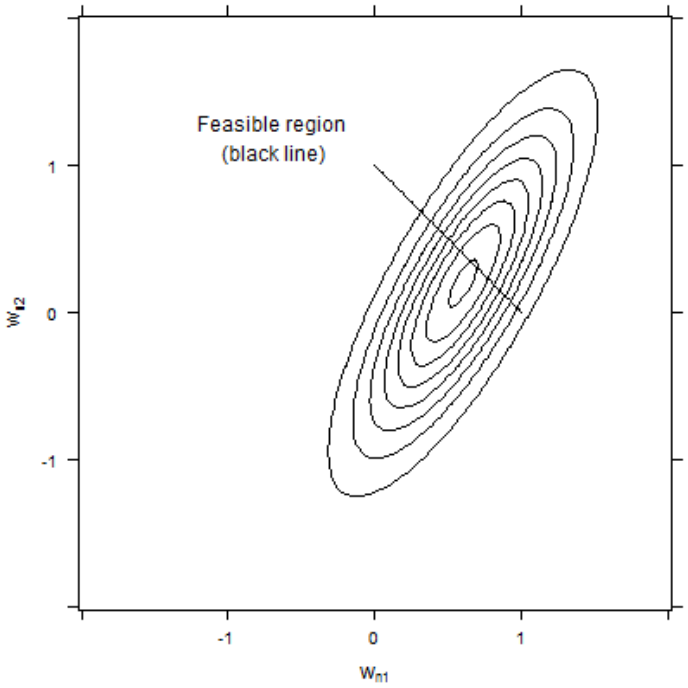


Figure 3.2: The contours are from $\mathcal{N}(\boldsymbol{w}_n|\boldsymbol{\mu}_n, \boldsymbol{\Sigma})$ with $\boldsymbol{\Sigma} = [0.2, 0.25; 0.25, 0.5]$, $\boldsymbol{\mu}_n = (0.8, 0.2)$ and $\boldsymbol{w}_n \in \mathbb{R}^2$. The support of the density $f_{\boldsymbol{w}_n|\boldsymbol{E},\sigma^2,\boldsymbol{x}_n}$ is only along the black line.

Sampling from $f_{\boldsymbol{w}_n|\boldsymbol{E},\sigma^2,\boldsymbol{x}_n}$ at this stage can be easily done with a MH sampler, using a uniform Dirichlet proposal. Dirichlet random variables meet the non-negativity as well

as the sum-to-one constraints and thus the only thing which remains to be checked is compliance to the Normal density. This way of sampling is very efficient and feasible for relatively low M.

Removing equality constraint from $f_{\boldsymbol{w}_n|\boldsymbol{E},\sigma^2,\boldsymbol{x}_n}$

A more complex approach is a univariate Gibbs sampler which works for any M. Note that we can not condition on $f_{\boldsymbol{w}_n|\boldsymbol{E},\sigma^2,\boldsymbol{x}_n}$ and go directly for w_{nm} since the sum-to-one constraint is deterministic, implying $\text{Var}(w_{nm}|w_{n\widetilde{m}}) = 0$. Our approach is inspired by Schmidt [23] who describes a general approach of how sampling can be done from a distribution constrained with any number of equality and inequality constraints. The idea is to first map the distribution to an affine subspace where the equality constraints hold. In that subspace the distribution is only truncated (because the inequality constraints still hold), but not degenerate.

The original papers refer to this novel idea, but do not translate it to the problem at hand. In this Section we derive the required components while relying on this general idea.[1] Thus we sample $\boldsymbol{y}$ from a reduced but equivalent space through the relationship $\boldsymbol{w}_n = \boldsymbol{A}\boldsymbol{y} + \boldsymbol{b}$ using

$$f(\boldsymbol{y}|\boldsymbol{E},\sigma^2,\boldsymbol{x}_n) = \mathcal{N}(\boldsymbol{y}|\boldsymbol{\mu}_{\boldsymbol{y}},\boldsymbol{\Sigma}_{\boldsymbol{y}})\,\mathbb{I}\left[\|\boldsymbol{y}\|_1 \leq 1\right]\prod_{m=1}^{M-1}\mathbb{I}\left[\boldsymbol{y} \geq 0\right] \tag{3.18}$$

where

$$\boldsymbol{\Sigma}_{\boldsymbol{y}}^{-1} = \frac{\boldsymbol{A}^T\boldsymbol{E}^T\boldsymbol{E}\boldsymbol{A}}{\sigma^2} \tag{3.19}$$

$$\boldsymbol{\mu}_{\boldsymbol{y}} = \left(\boldsymbol{A}^T\boldsymbol{E}^T\boldsymbol{E}\boldsymbol{A}\right)^{-1}\boldsymbol{A}^T\boldsymbol{E}^T\left[\boldsymbol{x}_n - \boldsymbol{E}\boldsymbol{b}\right] \tag{3.20}$$

The specification of $\boldsymbol{A}$ and $\boldsymbol{b}$ with full derivation of these results is given in Appendix B.4. In this new $\boldsymbol{y}$-space, the equality constraint does not hold and the density is a mere truncated Gaussian distribution in which the covariance matrix is much more stable. We can sample from this distribution using a Gibbs sampler for which the univariate density $f_{y_k|\boldsymbol{y}_{\widetilde{k}},\boldsymbol{E},\sigma^2,\boldsymbol{x}_n}$ is derived in Appendix B.5. The downside of this sampler compared to MH is slower sampling. The upside is that nothing ever gets rejected which results in a visible improvement in impure endmember detection.

3.4.5 Sampling $\boldsymbol{E}$

The conditional distribution of the endmembers $\boldsymbol{E}$ arises from the product of the Gaussian likelihood and the volume penalizing prior.

[1] We do not base ourselves on the results of Schmidt [23] – only his general idea – because we are not sure that his resulting inequality constraints match with those that follow from our own derivations. The general procedure does not seem to account completely for additional inequality constraints that arise once the mapping is done. This requires further research.

$$
\begin{aligned}
&f(\boldsymbol{E}|\boldsymbol{W},\sigma^2,\boldsymbol{X},\gamma) \\
&\quad= \prod_{b=1}^{B} f(\boldsymbol{e}_b|\boldsymbol{W},\sigma^2,\boldsymbol{x}_b,\gamma) \\
&\quad\propto \prod_{b=1}^{B}\left[\mathcal{N}(\boldsymbol{e}_b|\boldsymbol{\mu}_b,\boldsymbol{\Sigma}_{\boldsymbol{E}})\prod_{m=1}^{M}\mathbb{I}\left[e_{bm}\geq 0\right]\right]
\end{aligned}
\tag{3.21}
$$

where the parameters are given by

$$
\boldsymbol{\Sigma}_{\boldsymbol{E}}^{-1} = \frac{\boldsymbol{W}^T\boldsymbol{W}}{\sigma^2} + 2\gamma\left(\frac{\boldsymbol{I}_M - \boldsymbol{1}\boldsymbol{1}^T/M}{M-1}\right) \tag{3.22}
$$

$$
\boldsymbol{\mu}_b = \frac{\boldsymbol{\Sigma}_{\boldsymbol{E}}\boldsymbol{W}^T\boldsymbol{x}_b}{\sigma^2} \tag{3.23}
$$

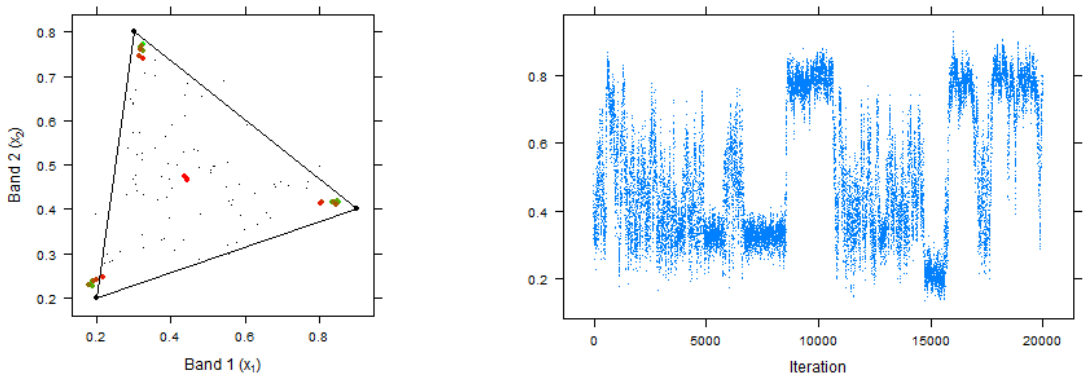
The derivation of the parameters is given in Appendix B.6. We see that $f_{\boldsymbol{e}_b|\boldsymbol{W},\sigma^2,\boldsymbol{x}_b,\gamma}$ is a multivariate Normal distribution constrained to the positive domain. The density $f_{e_{bm}|\boldsymbol{e}_{b\widetilde{m}},\boldsymbol{W},\sigma^2,\boldsymbol{x}_b,\gamma}$ for use within a Gibbs sampler is derived in Appendix B.7. The results are different and computationally more efficient, but equivalent to the results in [2].

Concerning the inverse, we have already touched upon a similar issue in Section 2.3.4. We conclude that it is highly unlikely that $\Sigma_{\boldsymbol{E}}$ is singular.

3.5 Model peculiarities

3.5.1 Intrinsic regularization of the Bayesian approach

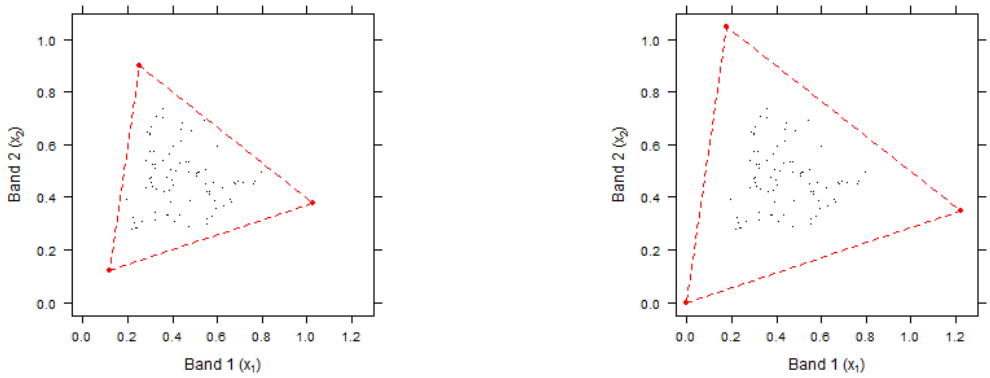
An interesting feat is that the Bayesian approach is inherently regularized. Fig. 3.3 shows that even if $\gamma = 0$, the endmember simplex will be stable and never much larger than the data points. Arngren already noticed this in [1] but had no explanation. Two years later, in [3], he turns up with a mathematical explanation for one dimension. The main insight is that we are not so much interested in $(\hat{\boldsymbol{E}},\hat{\boldsymbol{W}})_{MAP}$ from Eq. 3.11 but in $(\hat{e}_{bm})_{MAP}$ and $(\hat{w}_{nm})_{MAP}$ $\forall n,m,b$. Thus we are interested in marginal distributions. While the MAP estimator of the joint distribution is ill-defined, the MAP estimators of the components are not! And since we are dealing with indicator functions in priors the Riemann integration cannot be used, and one must resort to Lebesgue integration to come up with a closed-form solution. Arngren [3] does this for one dimension.



(a) (b)

Figure 3.3: Data for (a) is simulated as a 2-band 9×9 image with $\text{SNR}_{\text{dB}} = 15$ dB. The MAP endmember estimates are shown from green to red for $\gamma = \{0, 0.0001, 0.001, 0.01, 0.1, 1, 10, 100, 140, 1000\}$. The estimates are based on 3000 samples with 500 burn-in. The estimates are stable until we reach $\gamma = 140$. For $\gamma = 150$ and above the endmember approximations become multi-modal, as the trace in (b) shows for e_{11}. Once the simplex reaches minimal size, the MAP estimators become stable again.

We can understand this intrinsic regularization conceptually by looking at the prior for $\boldsymbol{W}$ in Eq. 3.8. The key for understanding is seeing that this prior conveys more information than just the non-negativity and sum-to-one constraints. It also says that each w_{nm} that satisfies the constraints is equally likely. In Fig. 3.4 we have two endmember simplexes for the same dataset. Given $\boldsymbol{E}$, for the left figure the abundances w_{nm} are uniformly distributed. But for the right figure, all the abundances w_{nm} are around 0.35. Our prior on $\boldsymbol{W}$ will thus favor the left simplex.



(a) (b)

Figure 3.4: Simplex in (a) is the MAP estimate for $\boldsymbol{E}$ with $\gamma = 0$ such that all w_{nm} are equally likely. Simplex in (b) on the other hand is such that most w_{nm} fall within the 0.3-0.4 range.

3.6 Conclusion

In this chapter we defined the Bayesian framework for the LMM model using a DAG with the Markov property. Under these assumptions we defined the priors for $\boldsymbol{E}$, $\boldsymbol{W}$ and σ^2 as uninformative as possible, while fulfilling the requirements of the LMM. For $\boldsymbol{E}$ and $\boldsymbol{W}$ this resulted in improper priors only asserting non-negativity and sum-to-zero constraints through indicator functions. A volume regularization on $\boldsymbol{E}$ was also made possible through the $\boldsymbol{E}$-prior, controlled by the hyper-parameter γ. For σ^2 we used the Jeffreys prior for scale parameters. We derived it and showed how it can be implemented as the limiting case of the Inverse-Gamma distribution. Furthermore, we showed how the MAP estimators of this model are related to the ICE estimators. We further went into greater detail than the original papers in deriving all the necessary conditional distributions for the Gibbs sampling procedure. Notable here is the sampling procedure from a degenerate distribution described in Section 3.4.4. We first mapped this distribution to a subspace in which the equality constraints hold, and sampled from that subspace. Then we back-transformed the samples to the original space.

The simulation results from Section 3.5.1 indicate that the Bayesian model is intrinsically volume-regularized and thus that γ can safely be set to zero. We explained the reason for this conceptually and referred to some mathematical proofs in other papers. We also noted that although the MAP estimator is ill-defined for the whole distribution, it is not once we consider the marginals $(\hat{e}_{bm})_{MAP}$ and $(\hat{w}_{nm})_{MAP}$. It is not always clear from these papers if, and how, $(\hat{\boldsymbol{E}}, \hat{\boldsymbol{W}})_{MAP}$ are computed. In subsequent chapters we will use the marginal MAP estimators. Let us now turn to some real data examples.

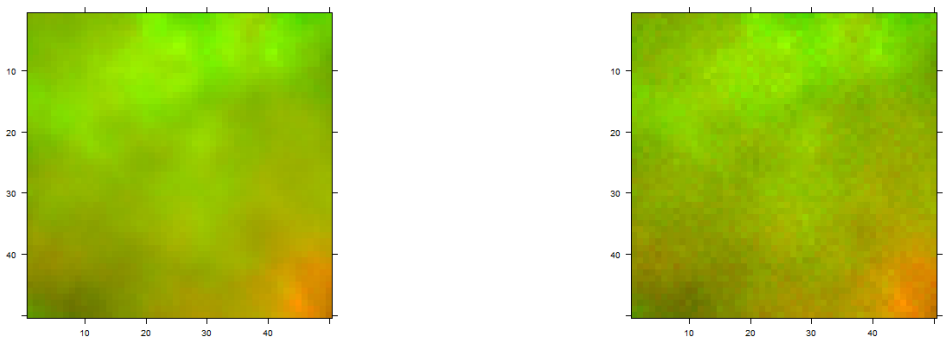
Chapter 4

Synthetic data

4.1 Image generation procedure

For simulating hyperspectral images we have three parameters to play with: the number of pixels N, the number of endmembers M and the number of bands B. Also, the image content (i.e. the ordening of the pixels) can be adjusted.

We confine ourselves to the 50×50-pixel image ($N = 2500$) in Fig. 4.1. For representational clarity we use only $B = 2$ bands and $M = 3$ synthetic endmembers as shown in Fig. 4.3d. The 3 abundance maps shown in Fig. 4.3a-c were generated using Markov Random Fields (MRF) with Matern covariance function. Once mixed with the endmembers, they result in Fig. 4.1a. Then we add $\mathrm{SNR}_{\mathrm{dB}} = 15$ dB Gaussian noise which results in Fig. 4.1b.[1]



(a) No noise (b) $\mathrm{SNR}_{\mathrm{dB}} = 15$ dB

Figure 4.1: False color synthetic image with (a) and without (b) noise. The two bands are represented as red and green.

In the following section we run the ICE, ICE-S and BayesNMF-Vol algorithms on the synthetic image and compare the estimated endmembers and abundance maps with the originals. Then we select an appropriate error metric for the endmembers. And in the subsequent section we explore the whole parameter space of the three models to find the optimal parameters.

[1] Clark [10] reports AVIRIS data exceeding SNR of 500 at most wavelengths. This corresponds with $\mathrm{SNR}_{\mathrm{dB}} = 10 \log_{10} 500 \sim 27$ dB for real data.

4.2 Analysis

ICE, ICE-S and BayesNMF-Vol algorithms were run to extract the endmembers and abundance maps. For ICE and ICE-S we use the threshold 0.9999 of the ratio between the current and the previous outcome of objective function as the stopping criterion. For BayesNMF-Vol algorithm we use 3000 samples with 500 samples as burn-in. To make the comparison between BayesNMF-Vol and ICE algorithms meaningful to some degree, we use the MAP point estimators in the Bayesian setting. We extract them from the MC samples using the kernel density estimation technique with Gaussian kernel as implemented in the R *density()* function.

The resulting endmembers and abundance maps are shown in Fig. 4.3 for comparison. The image can be split in three groups. The first group is the first row which consists of original abundance maps and endmembers. The three subsequent rows comprise the second group consisting of ICE, ICE-S and BayesNMF-Vol estimates with suggested parameters. For the ICE algorithm we use $\mu = 0.01$ as suggested by Berman [7]. For ICE-S we chose 98 % spatial regularization with same volume regularization as ICE, mainly to contrast the smoothing effect with ICE. For BayesNMF-Vol we set the parameter $\gamma = 0$ to show the intrinsic regularization. The last group consists of two last rows of ICE and ICE-S estimates with optimal parameters which are found in the next section.

Overall we see similar endmember estimates for ICE and ICE-S algorithms. As long as the volume regularization is kept the same, spatial regularization can be used to smoothen the abundance maps without changing the endmember estimates much.

4.3 Evaluation

Now that we have the endmembers and abundance maps, we want an objective means to compare them with the real ones as a way of evaluating the algorithms. For that we first need to select an error measure.

4.3.1 Error metric selection

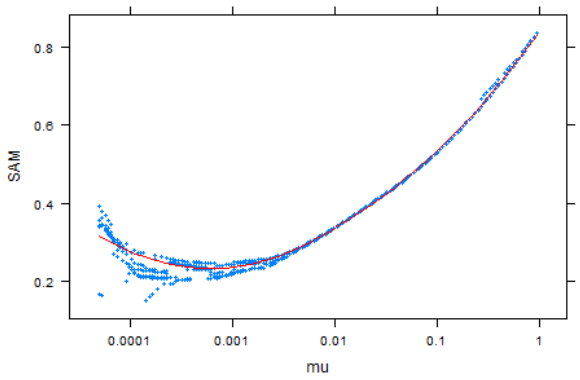
We are mainly interested in endmember identification here. So an obvious measure candidate is the mean squared error (MSE). The problem with this measure is that it is dependent on variations in albedo (i.e. the brightness or gain factor) of the spectra [15]. Variations in albedo can occur due to different topographic illumination effects, different device calibration, etc. A more appropriate measure would be one that is only dependent on the shape of the spectra, but not on albedo. Such a measure is the spectral angle mapper (SAM) [19]. This method quantifies the similarity between spectra by calculating the "angle" between them using the following formula:

$$\alpha_{SAM} = \cos^{-1}\left[\frac{\boldsymbol{e}_{ref}\boldsymbol{e}_{test}}{\|\boldsymbol{e}_{ref}\|\|\boldsymbol{e}_{test}\|}\right] \tag{4.1}$$

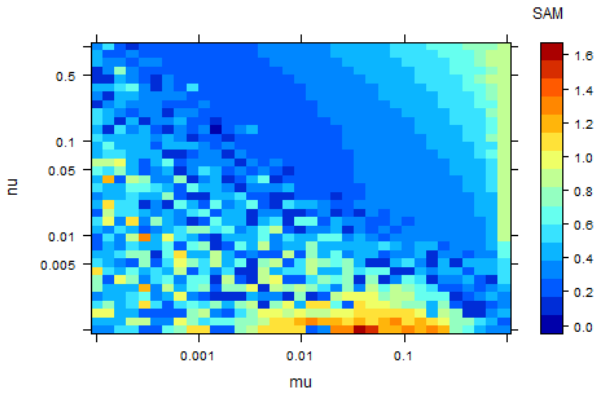
This measure will not change for $\boldsymbol{e}_{ref}$ and $\boldsymbol{e}_{test}$ when $\boldsymbol{e}_{test}$ is multiplied with some $c \in \mathbb{R}_0^+$ and thus the measure is insensitive to albedo. Further more, when $\boldsymbol{e}_{ref} = \boldsymbol{e}_{test}$ – and for any $c\boldsymbol{e}_{test}$ – this measure will result in 0. For a more in-depth discussion of this measure, and a comparison with a variant of a MSE-based measure, one can consult Dennison [15].

4.3.2 Results

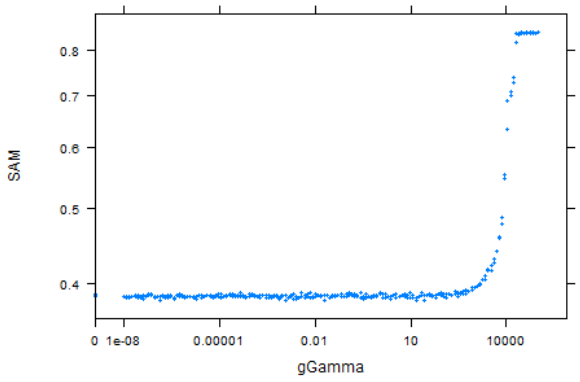
Fig. 4.2 gives an overview of SAM values in function of the parameters μ, ν and γ for ICE, ICE-S and BayesNMF-Vol algorithms respectively on the above mentioned synthetic image. A few noteworthy things are visible in these graphs. The ICE algorithm seems to have only one global minimum for μ down to $\mu = 0.005$. Thereafter seem to be lots of local minima in which the algorithm gets stuck. The optimal and least variable μ seems to be around 0.005. For ICE-S, the more noise we see in the Fig. 4.2b heat map, the more local minima there are in that region. Best parameters are the ones with least local minima, and the smallest SAM value. We estimate that subjectively to be around $\mu = 0.0015$ and $\nu = 0.55$. For the BayesNMF-Vol, the parameter γ does not play a mayor role in interval $]-\inf, 100]$.



(a) ICE



(b) ICE-S



(c) BayesNMF-Vol

	Parameters			Error metric
	μ	ν	γ	SAM
ICE	0.005	-	-	**0.22**
ICE-S	0.0015	0.55	-	0.25
BayesNMF-Vol	-	-	0	0.38

(d) Best parameters subjectively selected from Fig. (a-c) and the corresponding SAM values.

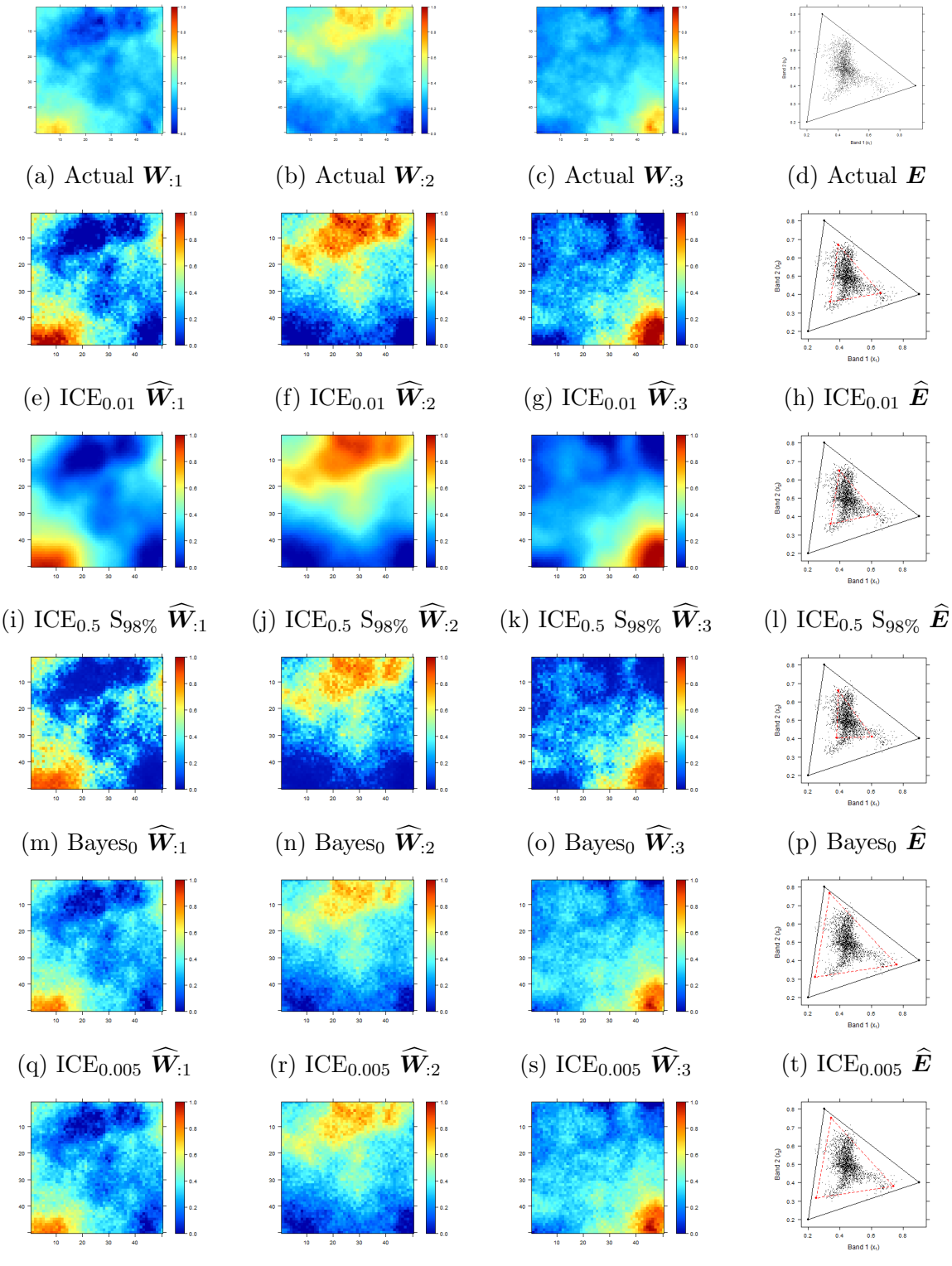
Figure 4.2: Figures (a-c) show SAM values in function of the model hyper-parameters. The ICE algorithm was run 5×150 times for different values of μ. The ICE-S algorithm was run 35×35 times for different values of μ and ν. The BayesNMF-vol algorithm was run 5×150 times for different values of γ. At each run the initialization was different so that in case of the ICE algorithm, the minima differ. The red curve in (a) is a quadratic LOESS curve. The runs are based on the aforementioned synthetic image.

Based on these SAM values we conclude that ICE and ICE-S perform similarly. For the used parametrization, ICE gives us the best SAM value of 0.22. Note though that SAM values can go down as far as 0.03 for ICE-S, but these minima all seem local. There are also seemingly stable regions where SAM of 0.20 is attained. But due to large parameter

space and the slowness of the algorithm it is difficult to pinpoint them. The generation of Fig. 4.2b which consists of 35×35 points took 10 days on two physical processor cores.

4.4 Conclusion

We have constructed a synthetic image based on 2 bands and 3 endmembers. We have further run ICE, ICE-S and BayesNMF-Vol on the synthetic image using a wide range of hyper-parameter configurations. We have compared the estimated endmembers with the original ones using the SAM metric. We conclude that although ICE-S was able to produce the least SAM errors, they all seemed to be local minima. In the SAM range around 0.20-0.30 both ICE and ICE-S were stable and performed similarly. Spatial regularization of ICE-S can be used to smoothen the abundance maps while keeping similar endmember estimates as of ICE, as long as the same volume regularization is used for both algorithms.



(a) Actual $\boldsymbol{W}_{:1}$ (b) Actual $\boldsymbol{W}_{:2}$ (c) Actual $\boldsymbol{W}_{:3}$ (d) Actual $\boldsymbol{E}$

(e) $\text{ICE}_{0.01}$ $\widehat{\boldsymbol{W}}_{:1}$ (f) $\text{ICE}_{0.01}$ $\widehat{\boldsymbol{W}}_{:2}$ (g) $\text{ICE}_{0.01}$ $\widehat{\boldsymbol{W}}_{:3}$ (h) $\text{ICE}_{0.01}$ $\widehat{\boldsymbol{E}}$

(i) $\text{ICE}_{0.5}$ $\text{S}_{98\%}$ $\widehat{\boldsymbol{W}}_{:1}$ (j) $\text{ICE}_{0.5}$ $\text{S}_{98\%}$ $\widehat{\boldsymbol{W}}_{:2}$ (k) $\text{ICE}_{0.5}$ $\text{S}_{98\%}$ $\widehat{\boldsymbol{W}}_{:3}$ (l) $\text{ICE}_{0.5}$ $\text{S}_{98\%}$ $\widehat{\boldsymbol{E}}$

(m) Bayes_{0} $\widehat{\boldsymbol{W}}_{:1}$ (n) Bayes_{0} $\widehat{\boldsymbol{W}}_{:2}$ (o) Bayes_{0} $\widehat{\boldsymbol{W}}_{:3}$ (p) Bayes_{0} $\widehat{\boldsymbol{E}}$

(q) $\text{ICE}_{0.005}$ $\widehat{\boldsymbol{W}}_{:1}$ (r) $\text{ICE}_{0.005}$ $\widehat{\boldsymbol{W}}_{:2}$ (s) $\text{ICE}_{0.005}$ $\widehat{\boldsymbol{W}}_{:3}$ (t) $\text{ICE}_{0.005}$ $\widehat{\boldsymbol{E}}$

(u) $\text{ICE}_{0.0015}$ $\text{S}_{45\%}$ $\widehat{\boldsymbol{W}}_{:1}$ (v) $\text{ICE}_{0.0015}$ $\text{S}_{45\%}$ $\widehat{\boldsymbol{W}}_{:2}$ (w) $\text{ICE}_{0.0015}$ $\text{S}_{45\%}$ $\widehat{\boldsymbol{W}}_{:3}$ (x) $\text{ICE}_{0.0015}$ $\text{S}_{45\%}$ $\widehat{\boldsymbol{E}}$

Figure 4.3: Endmembers and abundance maps of the synthetic image for ICE, ICE-S and BayesNMF-Vol. The subscript for ICE is the μ hyper-parameter and the percentage for S is related to the $1-\nu$ hyper-parameter. The Bayes subscript is the γ hyper-parameter.

Chapter 5

Real AVIRIS data - Cuprite

This chapter illustrates the discussed algorithms on a real hyperspectral dataset. This real dataset was acquired over Cuprite Hills (Navada, USA) in 1995 by the NASA/JPL "Airborne Visual and Infra-Red Imaging Spectrometer" (AVIRIS). The instrument was flown in an ER-2 aircraft (a modified U-2 spy plane) at 20 km. The spacing between pixels is 15 m, and the size of each pixel is about 18 m. With 224 bands, a spectral sampling and bandpass of ~ 10 nm and signal-to-noise ratios exceeding 500 at most wavelengths, AVIRIS data are widely used for terrestrial remote sensing. This dataset is freely available [14].

In the first section we explain hyperspectral data preprocessing and apply it to the AVIRIS data. The second section is devoted to the selection of the appropriate "alunite hill" scene for data analysis. In subsequent Section 5.3 we run the ICE, ICE-S and BayesNMF-Vol algorithms on the selected scene. We summarize our results in Section 5.4.

5.1 Preprocessing AVIRIS data

This Section is mainly based on the *Preprocessing AVIRIS Data Tutorial*[1]. The preprocessing of the data can be seen as a 3 step process.

5.1.1 Raw data

First we have the raw quantized pixel data as acquired by the sensor. NASA/JPL always processes the AVIRIS data to remove geometric and radiometric errors associated with the motion of the aircraft used during data collection.

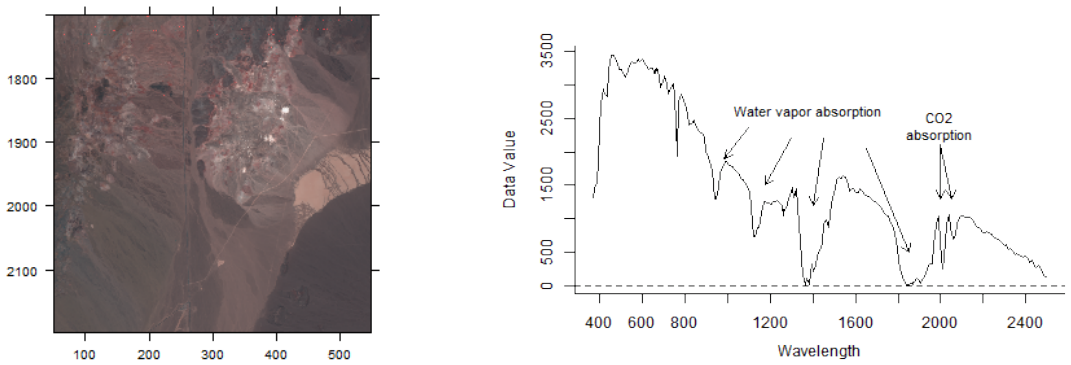
5.1.2 Radiance data

Radiance is the amount of radiation coming from an area. Due to the variance in band-sensitivity of the sensor, the raw data has to be calibrated to compensate this variability. This is done by multiplying the raw data by a series of gain values, one for each band resulting in $W/(cm^2 * sr * nm)$.

Radiance image includes the radiation effects of the sun. In fact, the spectrum of a radiance pixel closely matches that of the solar irradiance curve, i.e. the solar spectrum.

[1] http://www.harrisgeospatial.com/portals/0/pdfs/envi/PreprocessAVIRIS.pdf

Besides, atmospheric gas absorptions also cause specific irregularities in the spectrum, as shown in Fig. 5.1.



(a) Radiance map

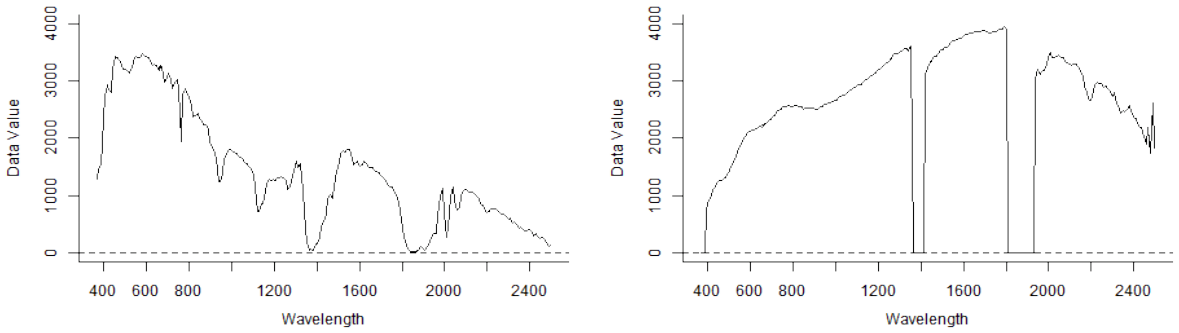
(b) Top left pixel spectrum

Figure 5.1: Left (a) is the false color Cuprite radiance image as obtained from [14]. The x and y axis denote the coordinates of each pixel from the original ENVI data which is much larger. For the RGB colors, the wavelengths 753.1287 nm, 530.8180 nm and 482.1898 nm were taken with intensities of each band normalized with the maximal band value. The right image (b) is the radiance spectrum of the top left pixel. The shape of the solar irradiance is clearly visible, and so are the water vapor and CO_2 absorptions.

For hyperspectral data analysis, one removes the effects of solar irradiance and atmosphere by calibrating the data to reflectance.

5.1.3 Reflectance data

Reflectance is the proportion of the radiation reflected off a surface to the radiation striking it. In hyperspectral data analysis, materials are identified by their reflectance spectra. So calibrating the data to reflectance is an important step towards identifying materials from an image. An atmospheric correction tool can remove the effects of atmospheric scattering and gas absorptions, to produce reflectance data as shown in Fig. 5.2.



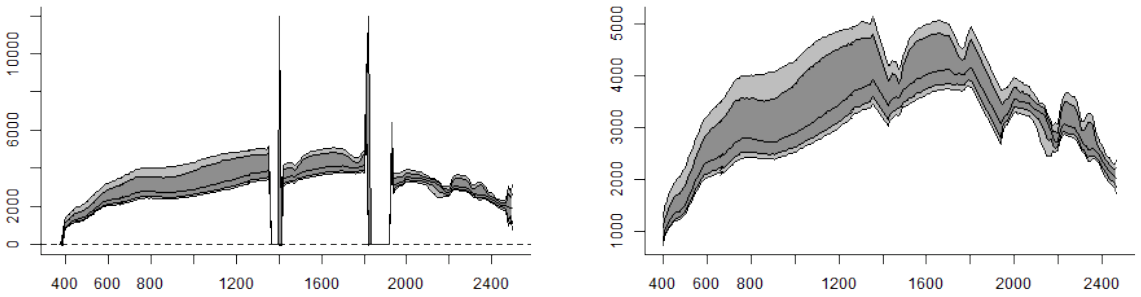
(a) Radiance (b) Reflectance

Figure 5.2: The left image (a) is the radiance spectrum of the top-left pixel (cf. Fig. 5.1b). The right image (b) shows the same spectrum processed by a atmospheric correction tool to produce the reflectance spectrum.

Note that the imagery may still have variations in illumination due to topography after this step.

5.1.4 Data cleanup

There can be some irregularity in the reflectance data caused by the atmospheric correction tool, mostly around the absorption bands. Fig. 5.3 depicts the intensity distribution in function of the band which clearly shows outlying values around the absorption wavelengths. 27 bands in total have been removed due to negative, outlying or noisy values, resulting in 197 remaining bands.



(a) Unclean data (b) Clean data

Figure 5.3: Plot (a) shows the 5th, 16th, 50th, 84th and 95th percentile per band of the uncleaned data. Plot (b) shows the same data, but with bands 366 −385 nm, 1363 −1413 nm and 1821 −1918 nm removed due to negative or zero values and bands 1811 nm and 1928 nm removed due to extreme outlying values. The last tree bands 2477 −2497 nm have also been removed due to seemingly high noise (cf. [20]).

5.2 Scene selection

The three algorithms have been evaluated on the "alunite hill" that appears in the Cuprite scene as depicted in Fig. 5.4. The geological properties of this area have been extensively investigated by geologists in [10], and [11]. The results of these studies are summarized in Fig. 5.5. The endmembers are dominated by three materials: muscovite, alunite, and kaolinite. They allow us to assess, at least subjectively, the accuracy of the three models.

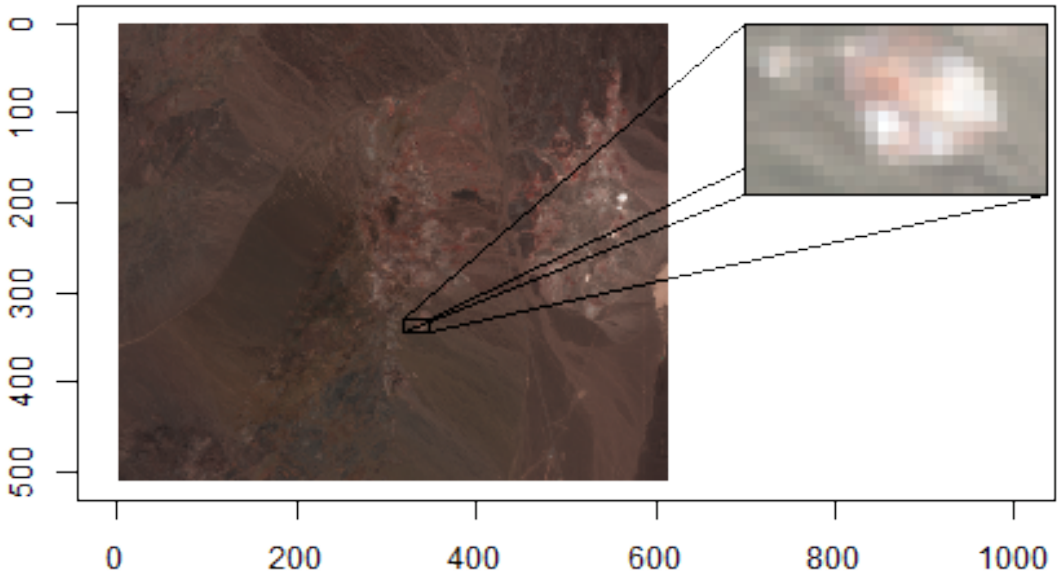


Figure 5.4: Left is the false color Cuprite reflectance image as obtained form [14]. Right is the zoomed in area of the "alunite hill" (16 x 28 pixels) used for material analysis. False colors are generated as described in Fig. 5.2.

5.3 Analysis

The 16 x 28 subimage with 197 bands, depicted in Fig. 5.4, has been unmixed using the ICE, ICE-S and BayesNMF-Vol algorithms using the same parameters as in the previous chapter. To assess the performance of the algorithms we need reference endmembers and abundance maps.

We compare the predicted endmembers with the reference spectra from the USGS library [12]. The problem that arises immediately is the correct choice of reference material. The original papers [10] and [11] from which Fig. 5.5 is taken are not very clear on which exact reference sample is used for matching. For example the are 17 different Kaolinite mineral samples and 23 Kaolinite mixtures in the library, and choosing the corresponding "Kaolinite + smectite or muscovite" reference sample is not evident. The reference samples we have chosen as reference endmembers are all collected at the Cuprite scene, have

B or higher purity level (i.e. have slight impurities) and are analyzed with the 2151-band ASD Fieldspec spectrometer. These are CU-98-5C (Alunite), CU91-252D (Muscovite) and CU91-200A (Kaolinite). These reference endmember spectra are depicted in Fig. 5.7d.

Once we have selected the reference endmembers for performance tests, we also need the reference abundance maps. For this we use the modified ICE algorithm with $\mu = 0.01$ and $\boldsymbol{E}$ fixed to the aforementioned USGS endmembers. We introduce a new parameter $\boldsymbol{a} \in \mathbb{R}^M$ which we interpret as the gain factor for each endmember. This is needed since spectra of materials measured remotely are almost always much weaker than those of pure reference materials. Minimizing the objective function of the ICE algorithm results in $\widehat{\boldsymbol{W}}$ and $\widehat{\boldsymbol{a}}$. The reference abundance maps are depicted in Fig. 5.7a-c.

5.4 Results

The resulting endmembers and abundance maps are shown in Fig. 5.7. A summary of the SAM errors is given in Table 5.6. Do note that these error measures are here for the sake of completeness and do not reflect performance correctly.[2]

Both ICE and ICE-S perform similarly here given the same volume regularization. For high volume regularization, the algorithm assumes more pure endmembers, which results in higher contrast abundance maps. The unregularized BayesNMF-Vol seems to perform badly. Volume regularization seems to have a large effect here – in contrast to our tests with the synthetic image.

	Parameters			Error metric
	μ	ν	γ	SAM
ICE	0.01	-	-	0.41
ICE-S	0.5	0.02	-	0.43
BayesNMF-Vol	-	-	0	0.92
ICE	0.005	-	-	**0.36**
ICE-S	0.0015	0.55	-	0.37
BayesNMF-Vol	-	-	0.001	0.43

Figure 5.6: SAM comparison for different parameter configurations.

[2]First, there is the uncertainty about the real endmembers in the image. As already noted, the spectra of reference materials is taken from small samples in labs within a controlled environment, while we are using spectra of some 18 m^2 of rock from an altitude of 20 km.

Second, the identification of materials based on (endmember) spectra is a topic on its own. Clark [11] notes that trusting in the least-squares matching algorithms as a form of similarity between reference and unknown is far from sufficient. Materials can be spectrally similar but chemically very different. Although not completely identical, they are similar enough "that noise and natural variations in field spectra make the assignment of the proper threshold at which to define identification or misidentification problematic." So simple measures like MSE (and even SAM) might be deceiving.

In short, assessing the accuracy of the estimated endmembers would require an experienced geologist/spectroscopist with a specific methodology. USGS Tetracorder [11] is an expert system that tries to mimic that methodology. Unfortunately, explaining and assessing the estimated endmembers with such a methodology would require a thesis on its own.

5.5 Conclusion

In this chapter we analyzed the "alunite hill" that appears in the AVIRIS Cuprite scene. For reference endmembers we based ourselves on geological work in the area and the USGS library. The reference abundance maps we extracted using a modified ICE algorithm based on the selected reference spectra. The SAM error results rate the ICE algorithm with $\mu = 0.005$ as the best performer.

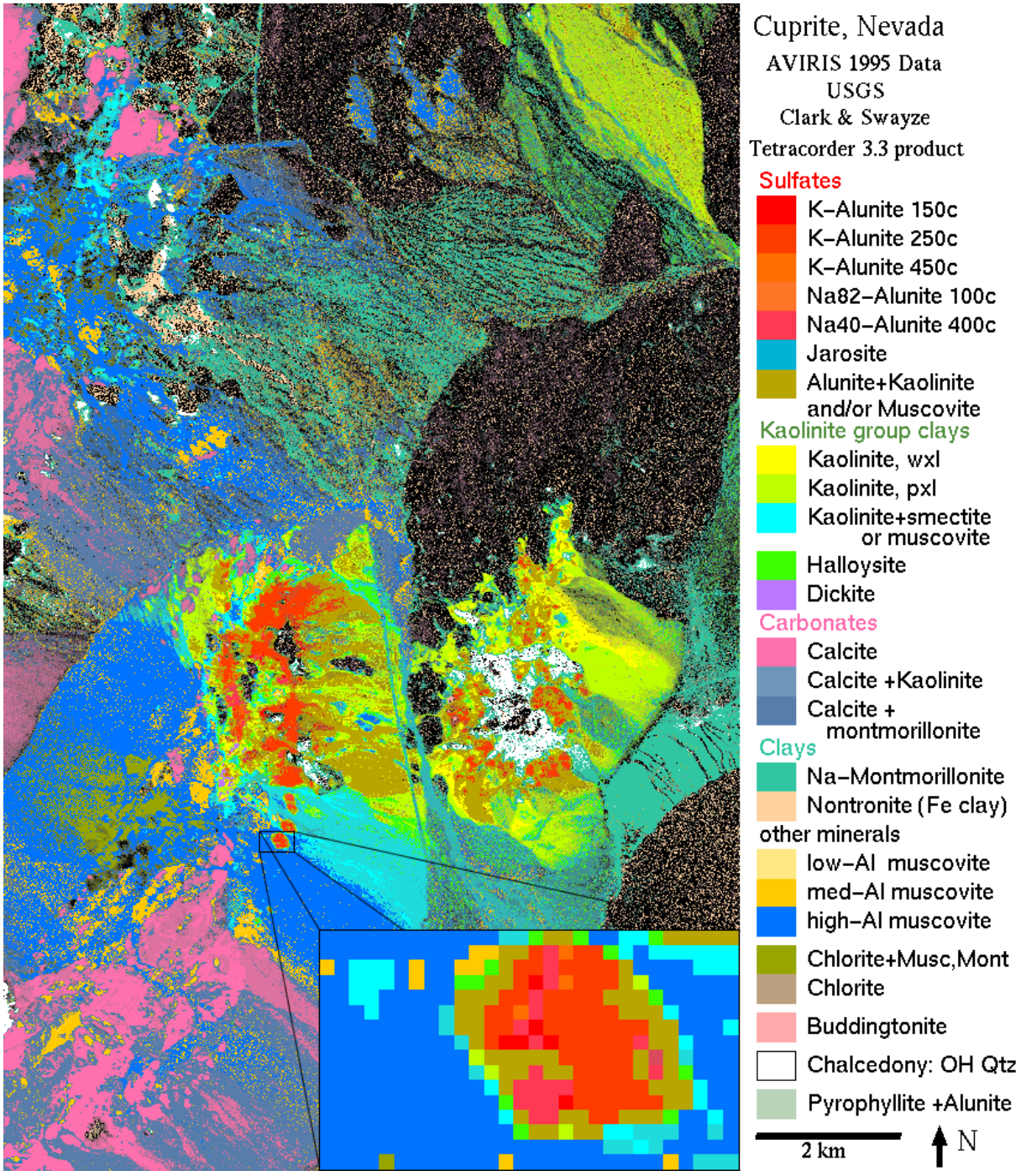
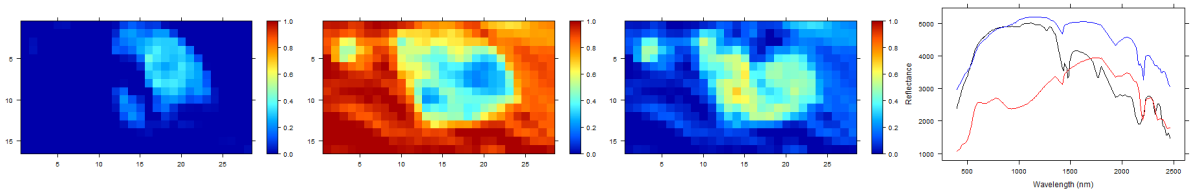
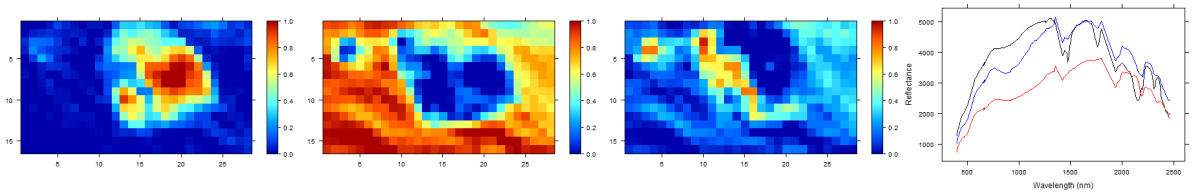


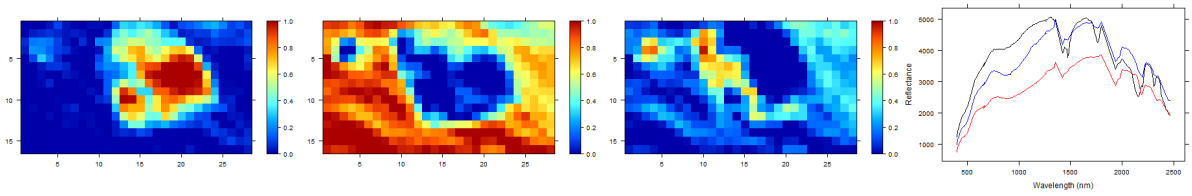
Figure 5.5: The image is taken from [11] with the "alunite hill" (16 × 28 pixels) zoomed in at the bottom right for clarity. The materials are identified using the USGS spectral library and Tetracoder's least squares shape-matching algorithm. According to [11] "mineral maps such as these have been extensively field checked to confirm the accuracy of the algorithm". Note that the surroundings (blue) are mostly high-Al muscovite. The hill edges are mainly kaolinite mixed with muscovite (cyan) and/or alunite (olive green), to mainly K-Alunite (250C - medium temperature) towards the center (orange).



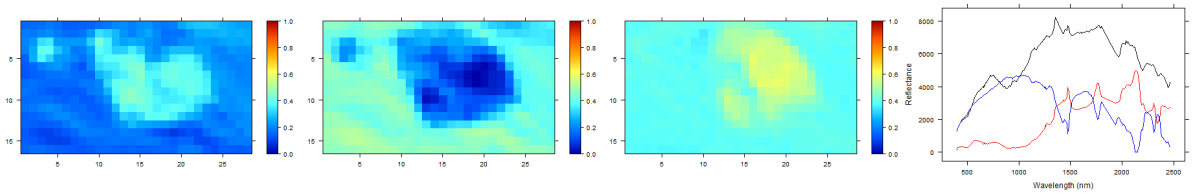
(a) Actual Alunite (b) Actual Muscovite (c) Actual Kaolinite (d) USGS Spectra



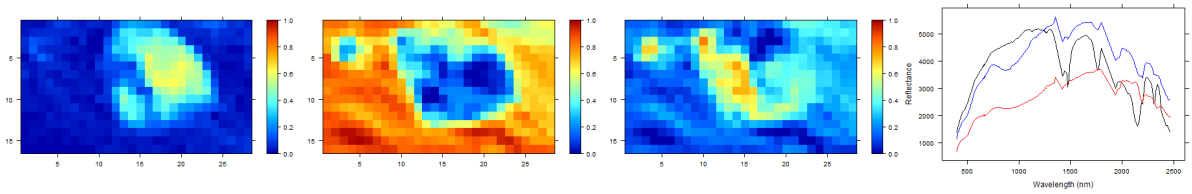
(e) ICE$_{0.01}$ $\widehat{\boldsymbol{W}}_{:1}$ (f) ICE$_{0.01}$ $\widehat{\boldsymbol{W}}_{:2}$ (g) ICE$_{0.01}$ $\widehat{\boldsymbol{W}}_{:3}$ (h) ICE$_{0.01}$ $\widehat{\boldsymbol{E}}$



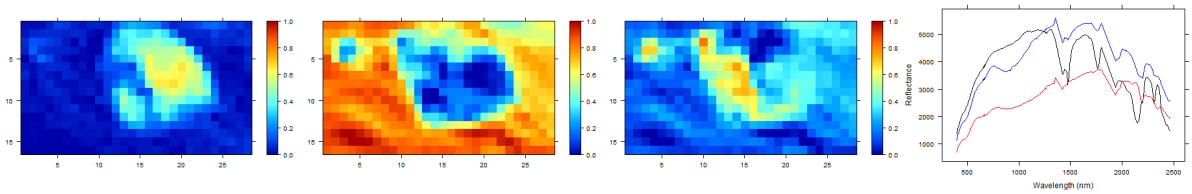
(i) ICE$_{0.5}$ S$_{98\%}$ $\widehat{\boldsymbol{W}}_{:1}$ (j) ICE$_{0.5}$ S$_{98\%}$ $\widehat{\boldsymbol{W}}_{:2}$ (k) ICE$_{0.5}$ S$_{98\%}$ $\widehat{\boldsymbol{W}}_{:3}$ (l) ICE$_{0.5}$ S$_{98\%}$ $\widehat{\boldsymbol{E}}$



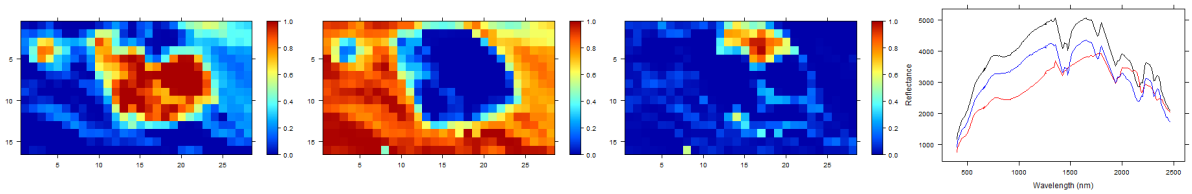
(m) Bayes$_0$ $\widehat{\boldsymbol{W}}_{:1}$ (n) Bayes$_0$ $\widehat{\boldsymbol{W}}_{:2}$ (o) Bayes$_0$ $\widehat{\boldsymbol{W}}_{:3}$ (p) Bayes$_0$ $\widehat{\boldsymbol{E}}$



(q) ICE$_{0.005}$ $\widehat{\boldsymbol{W}}_{:1}$ (r) ICE$_{0.005}$ $\widehat{\boldsymbol{W}}_{:2}$ (s) ICE$_{0.005}$ $\widehat{\boldsymbol{W}}_{:3}$ (t) ICE$_{0.005}$ $\widehat{\boldsymbol{E}}$



(u) ICE$_{0.0015}$ S$_{45\%}$ $\widehat{\boldsymbol{W}}_{:1}$ (v) ICE$_{0.0015}$ S$_{45\%}$ $\widehat{\boldsymbol{W}}_{:2}$ (w) ICE$_{0.0015}$ S$_{45\%}$ $\widehat{\boldsymbol{W}}_{:3}$ (x) ICE$_{0.0015}$ S$_{45\%}$ $\widehat{\boldsymbol{E}}$



(y) Bayes$_{0.001}$ $\widehat{\boldsymbol{W}}_{:1}$ (z) Bayes$_{0.001}$ $\widehat{\boldsymbol{W}}_{:2}$ (aa) Bayes$_{0.001}$ $\widehat{\boldsymbol{W}}_{:3}$ (ab) Bayes$_{0.001}$ $\widehat{\boldsymbol{E}}$

Figure 5.7: The algorithms and parameters used here are the same as in Fig. 4.2. The endmembers are Alunite (black), Muscovite (red) and Kaolinite (blue).

Chapter 6

Conclusion

In this thesis we explained how hyperspectral images are made and their main characteristic, the large amount of spectral bands. We defined a linear mixing model (LMM) which was driven by the idea that due to low spatial resolution of a hyperspectral sensor, spectra of many materials are mixed within each pixel. The LMM allows us to do the inverse step in a mathematically sound way: extract the material spectra from each pixel spectrum. For a whole hyperspectral image, this results in a few constituting materials – of which the spectra are called endmembers – and abundance maps of these materials. The process is called hyperspectral unmixing.

We briefly discussed three kinds of approaches to hyperspectral unmixing: geometrical, statistical and sparse. In this thesis we focused on Berman's ICE algorithm which is typifying for the geometrical approach. We also focused on one of the first Bayesian approaches, called Arngren's BayesNMF-Vol algorithm. Since the Bayesian approaches are currently a hot topic, we also sketched the complexity of some state-of-the-art approaches compared to the one described in this thesis.

Our first focus was the ICE algorithm. We first discussed the objective function derivation and implementation details since the original papers are quite succinct on that part. This resulted in small improvements such as the closed form solution for $\boldsymbol{E}^*$ and a stability improvement of the objective function by dividing it with the number of bands B. Subsequently we added spatial regularization to the ICE algorithm. We dubbed this the ICE-S algorithm. Spatial regularization favors structure in abundance maps, while disfavoring randomness. The synthetic and real data experiments showed clearly the smoothing effect of ICE-S, but were not able to show any difference in endmember estimates – compared to the ICE algorithm under the same volume regularization.

Our second focus was the BayesNMF-Vol algorithm. In a similar way as for the ICE algorithm, we analyzed the assumptions and implementation details in much greater detail than the original papers. We showed how the assumption of a DAG is essential for deriving the conditional distributions used by the Gibbs sampler. We also showed in full detail how sampling from a degenerate distribution can be done – something that proved quite challenging but was only mentioned as a side note in the original papers. We also explained the relationship between the Bayesian MAP estimators and ICE estimators. This led us to believe that the two algorithms are similar, and that the Bayesian approach should also be volume-regularized. The latter was proven wrong for at least two reasons. For one, due to high dimensionality one does not focus on the joint MAP estimator, but on the marginal MAP estimators. Second, because the Bayesian approach is intrinsically volume-regularized due to the "uninformative" prior for the abundances.

The ICE, ICE-S and BayesNMF-Vol algorithms were tested on both a synthetic and a real image. ICE-S was able to produce the best endmembers on the synthetic data once the hyper-parameters were optimized. This was expected because of the extra parameter (and thus an extra degree of freedom). But the values were very dependent on the initialization values. So much in fact that the outcomes were rather unreliable. We visualized this uncertainty by sampling SAM values over the whole hyper-parameter space. Overall we feel that the ICE and ICE-S perform similarly under the same volume regularization. The spatial regularization can be used in such cases to smoothen the abundance maps. On real data with Berman's suggested parametrization the ICE algorithm proved best. What we noticed almost consistently during our tests is the bad performance of the BayesNMF-Vol algorithm when not using volume regularization. A small volume regularization often helps the algorithm to achieve (much) better results, especially in higher dimensions.

This thesis is accompanied by a framework in R for hyperspectral analysis based on the *hyperSpec* package. Together with the implementation of the three algorithms, various functions were made for random image generation, false-color-, abundance- and endmember-plotting, parallel hyper-parameter optimization and USGS library spectra extraction – among others – to streamline the analysis process. Lots remains to be done. The research of hyperspectral unmixing is a never ending endeavor, mainly because of underspecification. But the framework can be used as a good starting point for anyone who wishes to start implementing new unmixing algorithms in R.

Appendices

Appendix A

ICE appendix

A.1 Relation between V and SSD

This is used in Eq. 2.9. Berman [5] notes these equivalent relations, but never derives them. We start from the sum of variance of $\boldsymbol{e}_b$ and show that it is proportional to the sum of squared distances (SSD) or rewrite it in matrix form which can be used in Eq. 2.13 to solve for $\boldsymbol{E}^*$.

$$
\begin{aligned}
V &= \sum_{b=1}^{B} Var(\boldsymbol{e}_b) \\
&= \sum_{b=1}^{B} \frac{\sum_{m=1}^{M} \left[e_{bm} - \frac{\sum_{m=1}^{M} e_{bm}}{M} \right]^2}{M-1} \\
&= \frac{\sum_{b=1}^{B}\sum_{m=1}^{M} e_{bm}^2}{M-1} - \frac{2\sum_{b=1}^{B}\sum_{m=1}^{M}\left(e_{bm}\sum_{m=1}^{M} e_{bm}\right)}{M(M-1)} + \frac{\sum_{b=1}^{B}\sum_{m=1}^{M}\left(\sum_{m=1}^{M} e_{bm}\right)^2}{M^2(M-1)} \\
&= \frac{\sum_{b=1}^{B}\sum_{m=1}^{M} e_{bm}^2}{M-1} - \frac{2\sum_{b=1}^{B}\left(\sum_{m=1}^{M} e_{bm}\right)^2}{M(M-1)} + \frac{\sum_{b=1}^{B}\left(\sum_{m=1}^{M} e_{bm}\right)^2}{M(M-1)} \\
&= \frac{\sum_{b=1}^{B}\sum_{m=1}^{M} e_{bm}^2}{M-1} - \frac{\sum_{b=1}^{B}\left(\sum_{m=1}^{M} e_{bm}\right)^2}{M(M-1)}
\end{aligned}
\tag{A.1}
$$

From here we can either go towards SSD:

$$
\begin{aligned}
V &= \frac{M\sum_{m=1}^{M}\boldsymbol{e}_m^T\boldsymbol{e}_m - \left(\sum_{m=1}^{M}\boldsymbol{e}_m\right)^T\left(\sum_{m=1}^{M}\boldsymbol{e}_m\right)}{M(M-1)} \\
&= \frac{\sum_{m=1}^{M}\sum_{k=1}^{M}\left(\boldsymbol{e}_k^T\boldsymbol{e}_k + \boldsymbol{e}_m\boldsymbol{e}_m - 2\boldsymbol{e}_k\boldsymbol{e}_m\right)/2}{M(M-1)} \\
&= \frac{\sum_{m=1}^{M}\sum_{k=1}^{M}\left(\boldsymbol{e}_k - \boldsymbol{e}_m\right)^T\left(\boldsymbol{e}_k - \boldsymbol{e}_m\right)/2}{M(M-1)} \\
&= \frac{\sum_{m=1}^{M-1}\sum_{k=m+1}^{M}\left(\boldsymbol{e}_k - \boldsymbol{e}_m\right)^T\left(\boldsymbol{e}_k - \boldsymbol{e}_m\right)}{M(M-1)} \\
&= \frac{SSD}{M(M-1)}
\end{aligned}
\tag{A.2}
$$

Or we can rewrite it in function of $\boldsymbol{e}_b$:

$$
\begin{aligned}
V &= \frac{\sum_{b=1}^{B}\left[M\boldsymbol{e}_b^T\boldsymbol{e}_b - \boldsymbol{e}_b^T\mathbf{1}\mathbf{1}^T\boldsymbol{e}_b\right]}{M(M-1)} \\
&= \sum_{b=1}^{B}\frac{\boldsymbol{e}_b^T(\boldsymbol{I}_M - \frac{\mathbf{1}\mathbf{1}^T}{M})\boldsymbol{e}_b}{M-1}
\end{aligned}
\tag{A.3}
$$

A.2 Minimum of $L_{reg}(\boldsymbol{E})$ given $\boldsymbol{W}$

This is the solution for the minimization of Eq. 2.3.4 using differential calculus [21]. We start by calculating the differential. Then we massage it into canonical form $\mathrm{d}L_{reg}(\boldsymbol{E}) = tr(\boldsymbol{A}^T\mathrm{d}\boldsymbol{E})$ from which we deduce that $\mathrm{D}L_{reg}(\boldsymbol{E}) = \boldsymbol{A}^T$. Once we know the derivative we can set it equal to 0 and solve for $\boldsymbol{E}$.

$$
\begin{aligned}
\mathrm{d}L_{reg}(\boldsymbol{E}) &= \mathrm{d}\frac{(1-\mu)}{N}tr\left[(\boldsymbol{X} - \boldsymbol{W}\boldsymbol{E}^T)^T(\boldsymbol{X} - \boldsymbol{W}\boldsymbol{E}^T)\right] \\
&\quad + \mathrm{d}\frac{\mu}{M-1}tr\left[\boldsymbol{E}(\boldsymbol{I}_M - \mathbf{1}\mathbf{1}^T/M)\boldsymbol{E}^T\right] \\
&= \frac{(1-\mu)}{N}tr\left[-2\boldsymbol{X}^T\boldsymbol{W}\mathrm{d}\boldsymbol{E}^T + 2\boldsymbol{E}\boldsymbol{W}^T\boldsymbol{W}\mathrm{d}\boldsymbol{E}^T\right] \\
&\quad + \frac{\mu}{M-1}tr\left[2\boldsymbol{E}\mathrm{d}\boldsymbol{E}^T - 2\boldsymbol{E}\mathbf{1}\mathbf{1}^T/M\mathrm{d}\boldsymbol{E}^T\right] \\
&= tr\left[\frac{2(1-\mu)}{N}\left(-\boldsymbol{X}^T\boldsymbol{W} + \boldsymbol{E}\boldsymbol{W}^T\boldsymbol{W} + \lambda(\boldsymbol{E} - \boldsymbol{E}\mathbf{1}\mathbf{1}^T/M)\right)\mathrm{d}\boldsymbol{E}^T\right] \\
&= tr\left[\frac{2(1-\mu)}{N}\left(-\boldsymbol{W}^T\boldsymbol{X} + \boldsymbol{W}^T\boldsymbol{W}\boldsymbol{E}^T + \lambda(\boldsymbol{I}_M - \mathbf{1}\mathbf{1}^T/M)\boldsymbol{E}^T\right)\mathrm{d}\boldsymbol{E}\right] \\
&= tr\left[\boldsymbol{A}^T\mathrm{d}\boldsymbol{E}\right]
\end{aligned}
\tag{A.4}
$$

where $\lambda = N\mu/[(1-\mu)(M-1)]$. We reduced $\mathrm{d}L_{reg}(\boldsymbol{E})$ to the canonical form from which we deduce that $\mathrm{D}L_{reg}(\boldsymbol{E}) = \boldsymbol{A}^T$. Now we can solve $\boldsymbol{A}^T = 0$ for $\boldsymbol{E}^*$:

$$
\arg\min_{\boldsymbol{E}}\ L_{reg}(\boldsymbol{E}) = \boldsymbol{X}^T\boldsymbol{W}\left[\boldsymbol{W}^T\boldsymbol{W} + \lambda(\boldsymbol{I}_M - \mathbf{1}\mathbf{1}^T/M)\right]^{-1}
\tag{A.5}
$$

A.3 Derivation of $S_n(\boldsymbol{w}_n)$

We use here the definitions from Section 2.5. We further define l as the rows of $\boldsymbol{U}^{[n]}$ corresponding to the elements in $\mathcal{K}_n$.

$$
\begin{aligned}
S_n(\boldsymbol{w}_n) &= \sum_{k\in\mathcal{K}} S_k(\boldsymbol{w}_n) + C_1^{te} \\
&= \sum_k \sum_m \mathrm{Var}(\boldsymbol{U}_{:m}^{[k]}) + C_1^{te} \\
&= \sum_k \sum_m \sum_l \frac{(w_{lm} - \sum_l w_{lm}/K_k)^2}{K_k - 1} + C_1^{te} \\
&= \sum_k \sum_m \left[\frac{\sum_l w_{lm^2}}{K_k - 1} - \frac{(\sum_l w_{lm})^2}{K_k(K_k-1)} \right] + C_1^{te} \\
&= \sum_k \sum_m \left[\frac{w_{nm}^2 + \sum_{l\neq n} w_{lm^2}}{K_k - 1} - \frac{w_{nm}^2 + 2w_{nm}\sum_{l\neq n} w_{lm} + (\sum_{l\neq n} w_{lm})^2}{K_k(K_k-1)} \right] + C_1^{te} \\
&= \sum_k \sum_m \left[\frac{w_{nm}^2}{K_k - 1} - \frac{w_{nm}^2 + 2w_{nm}\sum_{l\neq n} w_{lm}}{K_k(K_k-1)} \right] + C_2^{te} \\
&= \sum_{k\in\mathcal{K}} \frac{\boldsymbol{w}_n^T \boldsymbol{w}_n}{K_k} - 2\boldsymbol{w}_n^T \sum_{k\in\mathcal{K}} \frac{(\boldsymbol{U}_{\widetilde{n}:}^k)^T \mathbf{1}}{K_k(K_k-1)} + C_2^{te}
\end{aligned}
\tag{A.6}
$$

Appendix B

BayesNMF-Vol appendix

B.1 Density of linearly transformed variable

This is the deduction of the not so trivial step in Section 3.2.4. We have the linear transformation of $\sigma' = \phi(\sigma) = c\sigma$. We further assume that $\sigma \sim f(\sigma)$. We are looking for $f'(\sigma')$. We start by writing the probability

$$\begin{aligned} P(\sigma' < a) &= \int_0^a f'(\sigma')\,\mathrm{d}\sigma' \\ &= \int_0^{\phi^{-1}(a)} f(\sigma)\,\mathrm{d}\sigma \\ &= \int_0^a f(\phi^{-1}(\sigma'))\frac{\mathrm{d}\phi^{-1}(\sigma')}{\mathrm{d}\sigma'}\,\mathrm{d}\sigma' \\ &= \int_0^a f(\phi^{-1}(\sigma'))\frac{1}{c}\,\mathrm{d}\sigma' \end{aligned} \tag{B.1}$$

where we use integration by substitution in the second last equation. Thus we have that $f'(\sigma') = f(\phi^{-1}(\sigma'))/c = f(\sigma)/c$.

B.2 Derivation of $\mathcal{N}(x_k|\boldsymbol{x}_{\widetilde{k}}, \hat{\mu}_k, \hat{\sigma}_k^2)$ from $\mathcal{N}(\boldsymbol{x}|\boldsymbol{\mu}, \boldsymbol{\Sigma})$

This form is mainly useful for the Gibbs sampler where we wish to sample x_k from $\boldsymbol{x}$, one at a time. Suppose that $\boldsymbol{x} \sim \mathcal{N}(\boldsymbol{x}|\boldsymbol{\mu}, \boldsymbol{\Sigma})$ and that $\boldsymbol{\Sigma}^{-1} = (\sigma_{ij})$.

$$
\begin{aligned}
f(x_k|\boldsymbol{x}_{\widetilde{k}}) &= \frac{\mathcal{N}(\boldsymbol{x}|\boldsymbol{\mu},\boldsymbol{\Sigma})}{\int_{x_k}\mathcal{N}(\boldsymbol{x}|\boldsymbol{\mu},\boldsymbol{\Sigma})\,\mathrm{d}x_k} \\
&= C_1^{te}\exp\left[-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu})^T\boldsymbol{\Sigma}^{-1}(\boldsymbol{x}-\boldsymbol{\mu})\right] \\
&= C_2^{te}\exp\left[-\frac{1}{2}\left(\boldsymbol{x}^T\boldsymbol{\Sigma}^{-1}\boldsymbol{x}-2\boldsymbol{x}^T\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}\right)\right] \\
&= C_2^{te}\exp\left[-\frac{1}{2}\left(\sum_i x_i\sum_j\sigma_{ij}y_j-2\sum_i x_i\sum_j\sigma_{ij}\mu_j\right)\right] \\
&= C_3^{te}\exp\left[-\frac{1}{2}\left(\sigma_{kk}x_k^2+x_k\sum_{j\neq k}\sigma_{kj}y_j+x_k\sum_{i\neq k}\sigma_{ik}y_i-2x_k\sum_j\sigma_{kj}\mu_j\right)\right] \\
&= C_3^{te}\exp\left[-\frac{1}{2}\left(\sigma_{kk}x_k^2+x_k\boldsymbol{\Sigma}_{k:\widetilde{k}}^{-1}\boldsymbol{y}_{\widetilde{k}}+x_k(\boldsymbol{\Sigma}_{\widetilde{k}:k}^{-1})^T\boldsymbol{y}_{\widetilde{k}}-2x_k\boldsymbol{\Sigma}_{k:}^{-1}\boldsymbol{\mu}\right)\right] \\
&= C_3^{te}\exp\left[-\frac{1}{2}\frac{\left(x_k^2-2\{\boldsymbol{\Sigma}_{k:}^{-1}\boldsymbol{\mu}-\boldsymbol{\Sigma}_{k:\widetilde{k}}^{-1}\boldsymbol{y}_{\widetilde{k}}/2-(\boldsymbol{\Sigma}_{\widetilde{k}:k}^{-1})^T\boldsymbol{y}_{\widetilde{k}}/2\}\sigma_{kk}^{-1}x_k\right)}{\sigma_{kk}^{-1}}\right] \\
&= C_4^{te}\exp\left[-\frac{1}{2}\frac{\left(x_k-\left\{\boldsymbol{\Sigma}_{k:}^{-1}\boldsymbol{\mu}-\frac{1}{2}\left[\boldsymbol{\Sigma}_{k:\widetilde{k}}^{-1}+(\boldsymbol{\Sigma}_{\widetilde{k}:k}^{-1})^T\right]\boldsymbol{y}_{\widetilde{k}}\right\}\sigma_{kk}^{-1}\right)^2}{\sigma_{kk}^{-1}}\right] \\
&= C_4^{te}\exp\left[-\frac{1}{2}\frac{\left(x_k-\left\{\boldsymbol{\Sigma}_{k:}^{-1}\boldsymbol{\mu}-\boldsymbol{\Sigma}_{k:\widetilde{k}}^{-1}\boldsymbol{y}_{\widetilde{k}}\right\}\sigma_{kk}^{-1}\right)^2}{\sigma_{kk}^{-1}}\right] \\
&= \mathcal{N}(x_k|\hat{\mu}_k,\hat{\sigma}_k^2)
\end{aligned}
\tag{B.2}
$$

The second to last equality follows from the fact that $\boldsymbol{\Sigma}$ is symmetric. From the above we conclude that

$$
\hat{\sigma}_k^2 = \left(\boldsymbol{\Sigma}_{kk}^{-1}\right)^{-1} \tag{B.3}
$$

$$
\hat{\mu}_k = \frac{\boldsymbol{\Sigma}_{k:}^{-1}\boldsymbol{\mu}-\boldsymbol{\Sigma}_{k:\widetilde{k}}^{-1}\boldsymbol{y}_{\widetilde{k}}}{\boldsymbol{\Sigma}_{kk}^{-1}} \tag{B.4}
$$

B.3 Derivation of $f_{\boldsymbol{W}|\boldsymbol{E},\sigma^2,\boldsymbol{X}}$

These results are used in Section 3.4.4.

$$
\begin{aligned}
&f(\boldsymbol{W}|\boldsymbol{E},\sigma^2,\boldsymbol{X},\alpha,\beta,\gamma) \\
&\quad= \frac{f_{BayesNMF-Vol}(\boldsymbol{X},\boldsymbol{E},\boldsymbol{W},\sigma^2|\alpha,\beta,\gamma)}{\int_{\boldsymbol{W}} f_{BayesNMF-Vol}(\boldsymbol{X},\boldsymbol{E},\boldsymbol{W},\sigma^2|\alpha,\beta,\gamma)\,\mathrm{d}\boldsymbol{W}} \\
&\quad= \frac{f(\boldsymbol{X}|\boldsymbol{E},\boldsymbol{W},\sigma^2)f(\boldsymbol{W})}{\int_{\boldsymbol{W}} f(\boldsymbol{X}|\boldsymbol{E},\boldsymbol{W},\sigma^2)f(\boldsymbol{W})\,\mathrm{d}\boldsymbol{W}} \\
&\quad= f(\boldsymbol{W}|\boldsymbol{E},\sigma^2,\boldsymbol{X}) \\
&\quad\propto \prod_{n=1}^{N}\left[\mathcal{N}(\boldsymbol{x}_n|\boldsymbol{E}\boldsymbol{w}_n,\sigma^2)\,\mathbb{I}\left[\|\boldsymbol{w}_n\|_1=1\right]\prod_{m=1}^{M}\mathbb{I}\left[w_{nm}\geq 0\right]\right] \\
&\quad\propto \prod_{n=1}^{N}\left[\mathcal{N}(\boldsymbol{w}_n|\boldsymbol{\mu}_n,\boldsymbol{\Sigma}_{\boldsymbol{w}})\,\mathbb{I}\left[\|\boldsymbol{w}_n\|_1=1\right]\prod_{m=1}^{M}\mathbb{I}\left[w_{nm}\geq 0\right]\right] \\
&\quad\propto \prod_{n=1}^{N} f(\boldsymbol{w}_n|\boldsymbol{E},\sigma^2,\boldsymbol{x}_n)
\end{aligned}
\tag{B.5}
$$

Normal density function $f_{\boldsymbol{w}_n|\boldsymbol{E},\sigma^2,\boldsymbol{x}_n}$ from Eq. B.5 must be rewritten in function of $\boldsymbol{w}_n$. Thus we equate

$$
\begin{aligned}
\mathcal{N}(\boldsymbol{x}_n|\boldsymbol{E}\boldsymbol{w}_n,\sigma^2) &= \frac{\exp\left[-\frac{1}{2}(\boldsymbol{x}_n-\boldsymbol{E}\boldsymbol{w}_n)^T(\sigma^2\boldsymbol{I})^{-1}(\boldsymbol{x}_n-\boldsymbol{E}\boldsymbol{w}_n)\right]}{\sqrt{|2\pi\sigma^2\boldsymbol{I}|}} \\
&\propto \frac{\exp\left[-\frac{1}{2}(\boldsymbol{w}_n-\boldsymbol{\mu}_n)^T\boldsymbol{\Sigma}_{\boldsymbol{w}}^{-1}(\boldsymbol{w}_n-\boldsymbol{\mu}_n)\right]}{\sqrt{|2\pi\boldsymbol{\Sigma}_{\boldsymbol{w}}|}} \\
&= \mathcal{N}(\boldsymbol{w}_n|\boldsymbol{\mu}_n,\boldsymbol{\Sigma}_{\boldsymbol{w}})
\end{aligned}
\tag{B.6}
$$

from which we get parameters $\boldsymbol{\mu}_n$ and $\boldsymbol{\Sigma}$.

$$
\boldsymbol{\Sigma}_{\boldsymbol{w}}^{-1} = \frac{\boldsymbol{E}^T\boldsymbol{E}}{\sigma^2} \tag{B.7}
$$

$$
\boldsymbol{\mu}_n = (\boldsymbol{E}^T\boldsymbol{E})^{\dagger}\boldsymbol{E}^T\boldsymbol{x}_n \tag{B.8}
$$

The same derivation approach is used in [2].

B.4 Mapping $f_{\boldsymbol{w}_n|\boldsymbol{E},\sigma^2,\boldsymbol{x}_n}$ to $f_{\boldsymbol{y}|\boldsymbol{E},\sigma^2,\boldsymbol{x}_n}$

This is the mapping from the de $\boldsymbol{w}_n$-space to the $\boldsymbol{y}$-subspace in which the equality constraint $|\boldsymbol{w}_n| = 1$ holds, as initiated in Section. 3.4.4. We first introduce a transformed variable $\boldsymbol{y}$ which relates to $\boldsymbol{w}_n$ in the following way:

$$
\begin{bmatrix} 1 & 0 & 0 & \cdots & 0 \\ 0 & 1 & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & 1 \\ -1 & -1 & -1 & \cdots & -1 \end{bmatrix}
\begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_{M-1} \end{bmatrix}
+
\begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix}
=
\begin{bmatrix} w_{n1} \\ w_{n1} \\ \vdots \\ w_{nM-1} \\ w_{nM} \end{bmatrix}
\tag{B.9}
$$

We write this in a more concise way as:

$$\boldsymbol{A}\boldsymbol{y} + \boldsymbol{b} = \boldsymbol{w}_n \tag{B.10}$$

Now we can use this equality to derive a distribution of $\boldsymbol{y}$ based on $f_{\boldsymbol{w}_n|\boldsymbol{E},\sigma^2,\boldsymbol{x}_n}$:

$$\begin{aligned}
&f(\boldsymbol{y}|\boldsymbol{E},\sigma^2,\boldsymbol{x}_n) \\
&\quad= f(\boldsymbol{A}\boldsymbol{y}+\boldsymbol{b}|\boldsymbol{E},\sigma^2,\boldsymbol{x}_n) \\
&\quad= \mathcal{N}(\boldsymbol{A}\boldsymbol{y}+\boldsymbol{b}|\boldsymbol{\mu}_n,\boldsymbol{\Sigma}_{\boldsymbol{w}})\,\mathbb{I}\left[\|\boldsymbol{A}\boldsymbol{y}+\boldsymbol{b}\|_1 = 1\right]\prod_{m=1}^{M}\mathbb{I}\left[(\boldsymbol{A}\boldsymbol{y}+\boldsymbol{b})_m \geq 0\right] \\
&\quad= \mathcal{N}(\boldsymbol{A}\boldsymbol{y}+\boldsymbol{b}|\boldsymbol{\mu}_n,\boldsymbol{\Sigma}_{\boldsymbol{w}})\prod_{m=1}^{M}\mathbb{I}\left[(\boldsymbol{A}\boldsymbol{y}+\boldsymbol{b})_m \geq 0\right] \\
&\quad= \mathcal{N}(\boldsymbol{A}\boldsymbol{y}+\boldsymbol{b}|\boldsymbol{\mu}_n,\boldsymbol{\Sigma}_{\boldsymbol{w}})\,\mathbb{I}\left[\|\boldsymbol{y}\|_1 \leq 1\right]\prod_{m=1}^{M-1}\mathbb{I}\left[y_m \geq 0\right] \\
&\quad= \mathcal{N}(\boldsymbol{y}|\boldsymbol{\mu}_{\boldsymbol{y}},\boldsymbol{\Sigma}_{\boldsymbol{y}})\,\mathbb{I}\left[\|\boldsymbol{y}\|_1 \leq 1\right]\prod_{m=1}^{M-1}\mathbb{I}\left[y_m \geq 0\right]
\end{aligned} \tag{B.11}$$

where

$$\boldsymbol{\Sigma}_{\boldsymbol{y}}^{-1} = \boldsymbol{A}^T\boldsymbol{\Sigma}_{\boldsymbol{w}}^{-1}\boldsymbol{A} = \frac{\boldsymbol{A}^T\boldsymbol{E}^T\boldsymbol{E}\boldsymbol{A}}{\sigma^2} \tag{B.12}$$

$$\begin{aligned}
\boldsymbol{\mu}_{\boldsymbol{y}} &= \boldsymbol{\Sigma}_{\boldsymbol{y}}\boldsymbol{A}^T\boldsymbol{\Sigma}_{\boldsymbol{w}}^{-1}(\boldsymbol{\mu}_n - \boldsymbol{b}) \\
&= \left(\boldsymbol{A}^T\boldsymbol{E}^T\boldsymbol{E}\boldsymbol{A}\right)^{-1}\boldsymbol{A}^T\boldsymbol{E}^T\boldsymbol{E}\left[(\boldsymbol{E}^T\boldsymbol{E})^+\boldsymbol{E}^T\boldsymbol{x}_n - \boldsymbol{b}\right] \\
&= \left(\boldsymbol{A}^T\boldsymbol{E}^T\boldsymbol{E}\boldsymbol{A}\right)^{-1}\boldsymbol{A}^T\boldsymbol{E}^T\left[\boldsymbol{x}_n - \boldsymbol{E}\boldsymbol{b}\right]
\end{aligned} \tag{B.13}$$

To get rid of the MP inverse we use the equality $(\boldsymbol{A}^T\boldsymbol{A})^\dagger\boldsymbol{A}^T = \boldsymbol{A}^\dagger$ and $\boldsymbol{A}^T\boldsymbol{A}\boldsymbol{A}^\dagger = \boldsymbol{A}^T$ which holds for any matrix $\boldsymbol{A}$ (cf. [21, p.38]).

The $(M-1, M-1)$ matrix $\boldsymbol{A}^T\boldsymbol{E}^T\boldsymbol{E}\boldsymbol{A}$ is nonsingular $\iff$ columns of $\boldsymbol{E}\boldsymbol{A}$ are independent $\iff$ $\operatorname{rank}(\boldsymbol{E}\boldsymbol{A}) = M-1$ $\iff$ $\ker(\boldsymbol{E}\boldsymbol{A}) = \{\boldsymbol{x} : \boldsymbol{A}\boldsymbol{x} \in \ker\boldsymbol{E}\} = \{\boldsymbol{0}\}$. Knowing that $\operatorname{rank}(\boldsymbol{E}\boldsymbol{A}) \leq \min(\operatorname{rank}(\boldsymbol{E}), M-1) \leq min(B, M-1)$ the matrix will certainly be singular if $B < M-1$. More generally, it will be singular if $\operatorname{range}(\boldsymbol{A})$ intersects with $\ker(\boldsymbol{E})$ in more than $\{\boldsymbol{0}\}$. From our testing, the latter seems to be very unlikely, thus the matrix can be assumed nonsingular.

B.5 Derivation of $f_{y_k|\boldsymbol{y}_{\tilde{k}},\boldsymbol{E},\sigma^2,\boldsymbol{x}_n}$ for Gibbs sampling

We start from results given in Eq. 3.18.

$$
\begin{aligned}
&f(y_k|\boldsymbol{y}_{\widetilde{k}}, \boldsymbol{E}, \sigma^2, \boldsymbol{x}_n) \\
&\quad = \frac{f(\boldsymbol{y}|\boldsymbol{E}, \sigma^2, \boldsymbol{x}_n)}{\int_{y_k} f(\boldsymbol{y}|\boldsymbol{E}, \sigma^2, \boldsymbol{x}_n)\,\mathrm{d}y_k} \\
&\quad = \frac{\mathcal{N}(\boldsymbol{y}|\boldsymbol{\mu}_{\boldsymbol{y}}, \boldsymbol{\Sigma}_{\boldsymbol{y}})\,\mathbb{I}\left[\|\boldsymbol{y}\|_1 \leq 1\right]\prod_{m=1}^{M-1}\mathbb{I}\left[y_m \geq 0\right]}{\prod_{m\neq k}^{M-1}\mathbb{I}\left[y_m \geq 0\right]\int_{y_k}\mathcal{N}(\boldsymbol{y}|\boldsymbol{\mu}_{\boldsymbol{y}}, \boldsymbol{\Sigma}_{\boldsymbol{y}})\,\mathbb{I}\left[\|\boldsymbol{y}\|_1 \leq 1\right]\mathbb{I}\left[y_k \geq 0\right]\mathrm{d}y_k} \\
&\quad = \frac{\mathcal{N}(\boldsymbol{y}|\boldsymbol{\mu}_{\boldsymbol{y}}, \boldsymbol{\Sigma}_{\boldsymbol{y}})\,\mathbb{I}\left[y_k \leq 1 - \sum_{m\neq k}^{M-1} y_m\right]\mathbb{I}\left[y_k \geq 0\right]}{\int_{y_k}\mathcal{N}(\boldsymbol{y}|\boldsymbol{\mu}_{\boldsymbol{y}}, \boldsymbol{\Sigma}_{\boldsymbol{y}})\,\mathbb{I}\left[\|\boldsymbol{y}\|_1 \leq 1\right]\mathbb{I}\left[y_k \geq 0\right]\mathrm{d}y_k} \\
&\quad = \mathcal{N}(y_k|\mu_{yk}, \sigma_{yk}^2)\,\mathbb{I}\left[y_k \leq 1 - \sum_{m\neq k}^{M-1} y_m\right]\mathbb{I}\left[y_k \geq 0\right]
\end{aligned}
\tag{B.14}
$$

where the parameters μ_k and σ_k^2 we compute from the result in Appendix B.2:

$$
\sigma_{yk}^2 = \left[\left(\boldsymbol{\Sigma}_{\boldsymbol{y}}^{-1}\right)_{kk}\right]^{-1} \tag{B.15}
$$

$$
\mu_{yk} = \frac{\left(\boldsymbol{\Sigma}_{\boldsymbol{y}}^{-1}\right)_{k:}\boldsymbol{\mu}_{\boldsymbol{y}} - \left(\boldsymbol{\Sigma}_{\boldsymbol{y}}^{-1}\right)_{k:\widetilde{k}}\boldsymbol{y}_{\widetilde{k}}}{\sigma_{yk}^{-2}} \tag{B.16}
$$

Note that we are generating a Gibbs sequence for $\boldsymbol{y}$, so we need previous components of $\boldsymbol{y}$. We have $\boldsymbol{w}_n$ though, and we can get $\boldsymbol{y}$ by the inverse transformation of Eq. B.10. Matrix $\boldsymbol{A}$ is a $(M \times M-1)$ matrix. It has no inverse since it is non-square. But it does have a left inverse $\boldsymbol{A}_{Left}^{-1}$ since its columns are independent and equivalently that the Gram matrix $\boldsymbol{A}^T\boldsymbol{A}$ is non-singular. We use this property to write $\boldsymbol{y}$ in function of $\boldsymbol{w}_n$:

$$
\boldsymbol{y} = (\boldsymbol{A}^T\boldsymbol{A})^{-1}\boldsymbol{A}^T\boldsymbol{A}\boldsymbol{y} = \boldsymbol{A}_{Left}^{-1}\boldsymbol{A}\boldsymbol{y} = \boldsymbol{A}_{Left}^{-1}(\boldsymbol{w}_n - \boldsymbol{b}) = \boldsymbol{w}_{n\widetilde{M}} \tag{B.17}
$$

Note the last equality. $\boldsymbol{y}$ is just $\boldsymbol{w}_n$ with the last component removed!

B.6 Derivation of $f_{\boldsymbol{E}|\boldsymbol{W},\sigma^2,\boldsymbol{X},\gamma}$

We start from the results given in Eq. 3.21.

$$
\begin{aligned}
&f(\boldsymbol{E}|\boldsymbol{W},\sigma^2,\boldsymbol{X},\alpha,\beta,\gamma)\\
&\quad= \frac{f_{BayesNMF-Vol}(\boldsymbol{X},\boldsymbol{E},\boldsymbol{W},\sigma^2|\alpha,\beta,\gamma)}{\int_{\boldsymbol{E}} f_{BayesNMF-Vol}(\boldsymbol{X},\boldsymbol{E},\boldsymbol{W},\sigma^2|\alpha,\beta,\gamma)\,\mathrm{d}\boldsymbol{E}}\\
&\quad= \frac{f(\boldsymbol{X}|\boldsymbol{E},\boldsymbol{W},\sigma^2)f(\boldsymbol{E}|\gamma)}{\int_{\boldsymbol{E}} f(\boldsymbol{X}|\boldsymbol{E},\boldsymbol{W},\sigma^2)f(\boldsymbol{E}|\gamma)\,\mathrm{d}\boldsymbol{E}}\\
&\quad= f(\boldsymbol{E}|\boldsymbol{W},\sigma^2,\boldsymbol{X},\gamma)\\
&\quad\propto \prod_{n=1}^{N}\prod_{b=1}^{B} \frac{\exp\left[-\frac{1}{2\sigma^2}\left(x_{nb}-\boldsymbol{w}_n^T\boldsymbol{e}_b\right)^2\right]}{\sqrt{2\pi\sigma^2}} e^{-\gamma V(\boldsymbol{E})} \prod_{b=1}^{B}\prod_{m=1}^{M} \mathbb{I}\left[e_{bm}\geq 0\right]\\
&\quad= \prod_{n=1}^{N}\prod_{b=1}^{B} \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{1}{2\sigma^2}\left(x_{nb}-\boldsymbol{w}_n^T\boldsymbol{e}_b\right)^2\right)\\
&\qquad\times \exp\left(-\gamma\sum_{b=1}^{B}\frac{\boldsymbol{e}_b^T(\boldsymbol{I}_M-\frac{\mathbf{1}\mathbf{1}^T}{M})\boldsymbol{e}_b}{M-1}\right)\prod_{b=1}^{B}\prod_{m=1}^{M}\mathbb{I}\left[e_{bm}\geq 0\right]\\
&\quad= \prod_{b=1}^{B}\left(\frac{1}{\sqrt{2\pi\sigma^2}}\right)^N \exp\left(-\frac{1}{2}\left[\frac{(\boldsymbol{x}_b-\boldsymbol{W}\boldsymbol{e}_b)^T(\boldsymbol{x}_b-\boldsymbol{W}\boldsymbol{e}_b)}{\sigma^2}+2\gamma\frac{\boldsymbol{e}_b^T(\boldsymbol{I}_M-\mathbf{1}\mathbf{1}^T/M)\boldsymbol{e}_b}{M-1}\right]\right)\\
&\qquad\times\prod_{b=1}^{B}\prod_{m=1}^{M}\mathbb{I}\left[e_{bm}\geq 0\right]\\
&\quad\propto \prod_{b=1}^{B}\left[\mathcal{N}(\boldsymbol{e}_b|\boldsymbol{\mu}_b,\boldsymbol{\Sigma}_{\boldsymbol{E}})\prod_{m=1}^{M}\mathbb{I}\left[e_{bm}\geq 0\right]\right]\\
&\quad\propto \prod_{b=1}^{B} f(\boldsymbol{e}_b|\boldsymbol{W},\sigma^2,\boldsymbol{x}_b,\gamma)
\end{aligned}
\tag{B.18}
$$

So we need to equate

$$
\begin{aligned}
&(\boldsymbol{e}_b-\boldsymbol{\mu}_b)^T\boldsymbol{\Sigma}_{\boldsymbol{E}}^{-1}(\boldsymbol{e}_b-\boldsymbol{\mu}_b)+C^{te}\\
&\qquad= \frac{(\boldsymbol{x}_b-\boldsymbol{W}\boldsymbol{e}_b)^T(\boldsymbol{x}_b-\boldsymbol{W}\boldsymbol{e}_b)}{\sigma^2}+2\gamma\frac{\boldsymbol{e}_b^T(\boldsymbol{I}_M-\mathbf{1}\mathbf{1}^T/M)\boldsymbol{e}_b}{M-1}
\end{aligned}
\tag{B.19}
$$

and solve for $\boldsymbol{e}_b$ and $\boldsymbol{\Sigma}_{\boldsymbol{E}}^{-1}$ where we get:

$$
\boldsymbol{\Sigma}_{\boldsymbol{E}}^{-1} = \frac{\boldsymbol{W}^T\boldsymbol{W}}{\sigma^2}+2\gamma\left(\frac{\boldsymbol{I}_M-\mathbf{1}\mathbf{1}^T/M}{M-1}\right) \tag{B.20}
$$

$$
\boldsymbol{\mu}_b = \frac{\boldsymbol{\Sigma}_{\boldsymbol{E}}\boldsymbol{W}^T\boldsymbol{x}_b}{\sigma^2} \tag{B.21}
$$

B.7 Derivation of $f_{e_{bm}|\boldsymbol{e}_{b\widetilde{m}},\boldsymbol{W},\sigma^2,\boldsymbol{x}_b,\gamma}$ for Gibbs sampling

The density functions $f_{\boldsymbol{e}_b|\boldsymbol{W},\sigma^2,\boldsymbol{x}_b,\gamma}$ from Eq. B.18 is relatively difficult to sample from. So we derive $f_{e_{bm}|\boldsymbol{e}_{b\widetilde{m}},\boldsymbol{W},\sigma^2,\boldsymbol{x}_b,\gamma}$ which can be used in a Gibbs sampling procedure.

$$
\begin{aligned}
&f(e_{bm}|\boldsymbol{e}_{b\widetilde{m}}, \boldsymbol{W}, \sigma^2, \boldsymbol{x}_b, \gamma) \\
&\quad = \frac{f(\boldsymbol{e}_b|\boldsymbol{W}, \sigma^2, \boldsymbol{x}_b, \gamma)}{\int_{e_{bm}} f(\boldsymbol{e}_b|\boldsymbol{W}, \sigma^2, \boldsymbol{x}_b, \gamma)\,\mathrm{d}e_{bm}} \\
&\quad = \frac{\mathcal{N}(\boldsymbol{e}_b|\boldsymbol{\mu}_b, \boldsymbol{\Sigma}_{\boldsymbol{E}}) \prod_{i=1}^{M} \mathbb{I}\left[e_{bi} \geq 0\right]}{\prod_{i\neq m}^{M} \mathbb{I}\left[e_{bi} \geq 0\right] \int_{e_{bm}} \mathcal{N}(\boldsymbol{e}_b|\boldsymbol{\mu}_b, \boldsymbol{\Sigma}_{\boldsymbol{E}}) \mathbb{I}\left[e_{bm} \geq 0\right] \mathrm{d}e_{bm}} \\
&\quad = \frac{\mathcal{N}(\boldsymbol{e}_b|\boldsymbol{\mu}_b, \boldsymbol{\Sigma}_{\boldsymbol{E}}) \mathbb{I}\left[e_{bm} \geq 0\right]}{\int_{e_{bm}} \mathcal{N}(\boldsymbol{e}_b|\boldsymbol{\mu}_b, \boldsymbol{\Sigma}_{\boldsymbol{E}}) \mathbb{I}\left[e_{bm} \geq 0\right] \mathrm{d}e_{bm}} \\
&\quad = \mathcal{N}(e_{bm}|\bar{\mu}_{bm}, \bar{\sigma}_m^2) \mathbb{I}\left[e_{bm} \geq 0\right]
\end{aligned}
\tag{B.22}
$$

Using the results from Appendix B.2 we have

$$
\bar{\mu}_{bm} = \frac{(\boldsymbol{\Sigma}_{\boldsymbol{E}}^{-1})_{m:}\boldsymbol{\mu}_b - (\boldsymbol{\Sigma}_{\boldsymbol{E}}^{-1})_{m:\widetilde{m}}\boldsymbol{e}_{b\widetilde{m}}}{(\boldsymbol{\Sigma}_{\boldsymbol{E}}^{-1})_{mm}} \tag{B.23}
$$

$$
\bar{\sigma}_m^2 = 1/(\boldsymbol{\Sigma}_{\boldsymbol{E}}^{-1})_{mm} \tag{B.24}
$$

Appendix C

Source code

The R source code is in RMD files. The compiled output is available at `http://josipovic.be/statistics/dissertation/`.

All the source code, data, R and its libraries, and output used in this thesis are compiled and made available at: `https://hyperspec-unmix.sourceforge.io`. A small wiki for running the code is available. The source code is composed of RMD files where each covers a topic on its own. For example, the *HyperImage.rmd* file has the definition of the *HyperSpecExt* object, and functions for false color plotting and plotting of the abundance maps. The *Alg-ICE.rmd* has the implementation of the ICE algorithm and depends on, a.o., the definitions in *HyperImage.rmd*. This dependence is declared in the first chunk. This dependence declaration allows *Alg-ICE.rmd* to be compiled on its own. The files with "TEST" suffix, such as *Alg-ICE-TEST.rmd*, contain some kind of processing with the aforementioned algorithm. All RMD files are included in the *Main.rmd* file which, when compiled with *knitr*, gives an overview of all the source code and output.

The R library and used packages – most notably the *HyperSpec* package – are available in the */r* directory. The used Cuprite and USGS data can be found in the */data* directory. These files fall under public license and are made available for convenience and compatibility.

Bibliography

[1] Morten Arngren, Mikkel N. Schmidt, and Jan Larsen. Bayesian nonnegative matrix factorization with volume prior for unmixing of hyperspectral images. In *2009 IEEE International Workshop on Machine Learning for Signal Processing*, pages 1–6. IEEE, 2009.

[2] Morten Arngren, Mikkel N. Schmidt, and Jan Larsen. Supplementary material for unmixing of hyperspectral images using Bayesian nonnegative matrix factorization with volume prior. 2010.

[3] Morten Arngren, Mikkel N. Schmidt, and Jan Larsen. Unmixing of hyperspectral images using Bayesian non-negative matrix factorization with volume prior. *Journal of Signal Processing Systems*, 65(3):479–496, 2011.

[4] David Barber. *Bayesian reasoning and machine learning*. Cambridge University Press, 2012.

[5] Mark Berman, Harri Kiiveri, Ryan Lagerstrom, Andreas Ernst, Rob Dunne, and Jonathan F. Huntington. Ice: an automated statistical approach to identifying endmembers in hyperspectral images. In *Geoscience and Remote Sensing Symposium, 2003. IGARSS'03. Proceedings. 2003 IEEE International*, volume 1, pages 279–283. IEEE, 2003.

[6] Mark Berman, Harri Kiiveri, Ryan Lagerstrom, Andreas Ernst, Rob Dunne, and Jonathan F. Huntington. Ice: A statistical approach to identifying endmembers in hyperspectral images. *IEEE transactions on Geoscience and Remote Sensing*, 42(10):2085–2095, 2004.

[7] Mark Berman, Aloke Phatak, Ryan Lagerstrom, and Bayden R. Wood. Ice: a new method for the multivariate curve resolution of hyperspectral images. *Journal of Chemometrics*, 23(2):101–116, 2009.

[8] José M. Bioucas-Dias, Antonio Plaza, Nicolas Dobigeon, Mario Parente, Qian Du, Paul Gader, and Jocelyn Chanussot. Hyperspectral unmixing overview: Geometrical, statistical, and sparse regression-based approaches. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 5(2):354–379, 2012.

[9] George Casella and Edward I. George. Explaining the Gibbs sampler. *The American Statistician*, 46(3):167–174, 1992.

[10] Roger N. Clark, Gregg A. Swayze, and Andrea Gallagher. Mapping minerals with imaging spectroscopy. *US Geological Survey, Office of Mineral Resources Bulletin*, 2039:141–150, 1993.

[11] Roger N. Clark, Gregg A. Swayze, K. Eric Livo, Raymond F. Kokaly, Steve J. Sutley, J. Brad Dalton, Robert R. McDougal, and Carol A. Gent. Imaging spectroscopy: Earth and planetary remote sensing with the usgs tetracorder and expert systems. *Journal of Geophysical Research: Planets*, 108(E12), 2003.

[12] Roger N. Clark, Gregg A. Swayze, Richard Wise, K. Eric Livo, T. Hoefen, Raymond F. Kokaly, and Steve J. Sutley. Usgs digital spectral library splib06a. *US Geological Survey, Digital Data Series*, 231, 2007.

[13] Maurice D. Craig. Minimum-volume transforms for remotely sensed data. *IEEE Transactions on Geoscience and Remote Sensing*, 32(3):542–552, 1994.

[14] AVIRIS Free Data. Jet propulsion lab. *California Inst. Technol., Pasadena.[Online]*, 1997.

[15] Philip E. Dennison, Kerry Q. Halligan, and Dar A. Roberts. A comparison of error metrics and constraints for multiple endmember spectral mixture analysis and spectral angle mapper. *Remote Sensing of Environment*, 93(3):359–367, 2004.

[16] Nicolas Dobigeon, Saïd Moussaoui, Martial Coulon, Jean-Yves Tourneret, and Alfred O Hero. Joint Bayesian endmember extraction and linear unmixing for hyperspectral imagery. *IEEE Transactions on Signal Processing*, 57(11):4355–4368, 2009.

[17] Abderrahim Halimi, Nicolas Dobigeon, Jean-Yves Tourneret, and Paul Honeine. A new Bayesian unmixing algorithm for hyperspectral images mitigating endmember variability. In *Acoustics, Speech and Signal Processing (ICASSP), 2015 IEEE International Conference on*, pages 2469–2473. IEEE, 2015.

[18] Edwin T. Jaynes. Prior probabilities. *IEEE Transactions on systems science and cybernetics*, 4(3):227–241, 1968.

[19] Fred A. Kruse, A. B. Lefkoff, J. W. Boardman, K. B. Heidebrecht, A. T. Shapiro, P. J. Barloon, and A. F. H. Goetz. The spectral image processing system (SIPS)-interactive visualization and analysis of imaging spectrometer data. *Remote sensing of environment*, 44(2-3):145–163, 1993.

[20] Rong Liu, Bo Du, and Liangpei Zhang. Hyperspectral unmixing via double abundance characteristics constraints based nmf. *Remote Sensing*, 8(6):464, 2016.

[21] Jan R. Magnus and Heinz Neudecker. *Matrix differential calculus with applications in statistics and econometrics*. John Wiley & Sons, 3rd edition, 2007.

[22] Jorge Nocedal and Stephen J. Wright. *Numerical Optimization*. Springer, New York, 2nd edition, 2006.

[23] Mikkel Schmidt. Linearly constrained Bayesian matrix factorization for blind source separation. In *Advances in neural information processing systems*, pages 1624–1632, 2009.

[24] Ben Somers, Gregory P. Asner, Laurent Tits, and Pol Coppin. Endmember variability in spectral mixture analysis: A review. *Remote Sensing of Environment*, 115(7):1603–1616, 2011.

[25] Wei Tang, Zhenwei Shi, Ying Wu, and Changshui Zhang. Sparse unmixing of hyperspectral data using spectral a priori information. *IEEE Transactions on Geoscience and Remote Sensing*, 53(2):770–783, 2015.

[26] Michael E. Winter. N-findr: An algorithm for fast autonomous spectral end-member determination in hyperspectral data. In *SPIE's International Symposium on Optical Science, Engineering, and Instrumentation*, pages 266–275. International Society for Optics and Photonics, 1999.

[27] Stephen J. Wright. Coordinate descent algorithms. *Mathematical Programming*, 151(1):3–34, 2015.

Faculty of Sciences
Department of Applied Mathematics, Computer Science and Statistics
Faculty of Engineering
Department of Telecommunications and Information Processing (TELIN)
Image Processing and Interpretation (IPI)

Volume-based geometric and Bayesian approaches in linear hyperspectral unmixing

Davor Josipovic

Promoter: Prof. Dr. Aleksandra Pizurica
Co-promoter: Prof. Dr. Hongyan Zhang
Co-promoter: Prof. Dr. Dries Benoit

Thesis submitted in fulfillment of the requirements for the degree of
Master of Science in Statistical Data Analysis

Year: 2016–2017

Compiled: June 21, 2017