International Master ESECA
Signal, Image, Processing

Bayesian Nonparametric Methods

Abdelghafour HALIMI

Encadrant : Jean-Yves TOURNERET

Co-encadrant : Nicolas DOBIGEON

Septembre 2014

IRIT/INP-ENSEEIHT, 2 rue Camichel, BP 7122, 31071 Toulouse cedex 7, France

Remerciements

Ce travail a été effectué à l'Institut de Recherche en Informatique de Toulouse.

Tout d'abord, je tiens à remercier tout particulièrement et très chaleureusement mon encadrant M. Jean-Yves Tourneret pour ses conseils et orientations qui m'ont guidé jusqu'à l'aboutissement de ce travail. Qu'il trouve ici ma sincère gratitude.

Je tiens aussi à remercier mon co-encadrant M. Nicolas Dobigeon pour ses remarques pertinentes qui ont apporté une amélioration certaine à mon travail. Je le remercie également d'avoir examiné mon travail et proposé des modications constructives.

Que tous les professeurs qui ont contribué à ma formation trouvent ici ma plus profonde gratitude.

Enfin, mes remerciements vont à toute personne ayant contribuée, de près ou de loin, à la réalisation de ce travail.

Contents

List of Figures

Introduction

The hyperspectral imaging is a rapidly growing domain of the fact for which we try to extract more and more information of every pixel. The hyperspectral image is obtained by considering the same scene observed in various lengths of electromagnetic wave. By grouping all these images we obtain a cube of data which will be used for the analysis and the treatment to be made. Every pixel of this cube is represented by a spectrum for which the number of samples corresponds to the number of the considered wavelengths. One of the most treatments in hyperspectral imaging is the spectral unmixing which assumes that the vectors of data (i.e, spectra associated with the pixels of the image) are a linear combination of a number given by present pur spectres in the image to be treated. The spectral unmixing is the operation which consists in decomposing the spectre of a pixel into one collection of spectral components called pure spectra (endmembers) with coefficients named abundances which represent the proportions of every endmember in the analyzed pixel [1].

We find several Bayesian parametric methods of linear unmixing in the literature [2], [3]. On the other hand, the number of papers speaking about unmixing with Bayesian nonparametric methods is more limited. The work made in this stage consists in studying a method of linear unmixing based on a Bayesian non parametric method. The algorithm requires defining a priori distributions for the unknown parameters (the abundances and the endmembers) and estimating this latter from their posteriors. As the usual Bayesian estimator such as the estimator of maximum a posteriori (MAP) and the estimator of the minimum mean square error (MMSE) have no simple analytical expression, we suggest the use of Markov Chain Monte Carlo (MCMC) methods. The purpose of the MCMC methods is to simulate samples according to a distribution of interest (the posterior distribution of the unknown parameters of the model) and to use these simulated samples to estimate the unknown parameters of the studied model.

The present work consists of the study of the various models based on the Dirichlet process and then applied to the hyperspectral imaging for the linear unmixing according to the various formulations exposed in the literature. The report is organized as follows :

- **Chapter 1 :** How many classes should I use in my mixture model ? This question regularly exercises scientists as they explore their data. Most scientists address this question by first fitting several models, with different numbers of clusters or factors, and then selecting one using model comparison metrics. In the first chapter we describe Bayesian nonparametric (BNP) models. The BNP approach is to fit a single model that can adapt its complexity to the data. Furthermore, BNP models allow the complexity to grow as more data are observed.
- **Chapter 2 :** The second chapter deals with the Dirichlet distribution which forms our first step toward understanding the Dirichlet process model (DPM). The DPM model provides a distribution on distributions with many attractive

properties and is widely used in practice.

- **Chapter 3 :** Mixture models are often used to describe data which is distributed according to some set of underlying mechanisms where each data point is assumed to be independently generated by only one of these underlying distributions. The third chapter deals with the Dirichlet process mixture model which extends the basic mixture model by applying a Dirichlet process prior to the mixing proportions. After that, we describe three processes (Pólya's Urn, the Chinese Restaurant and the Indian Buffet) which allow us to generate samples from a Dirichlet process.
- **Chapter 4 :** The fourth chapter deals with mixing models in hyperspectral imagery. Mixed pixels are a mixture of more than one distinct substance.
- **Chapter 5 :** In the fifth chapter, we study two applications of the Dirichlet process mixture model (DPMM). The first application deals with the problem of clustering and the second application concerns the problem of unmixing in hyperspectral imagery.Spectral unmixing is the procedure by which the measured spectrum of a mixed pixel is decomposed into a collection of constituent spectra, or *endmembers*, and a set of corresponding fractions, or *abundances*, that indicate the proportion of each endmember present in the pixel. Endmembers normally correspond to familiar macroscopic objects in the scene, such as water, soil, metal, or any natural or man-made material.
- **Chapter 6 :** In the sixth chapter, we give some results when considering synthetic and real data. We begin by generating a mixture of (3,4 and 5) two-dimensional Gaussians and we represent the estimated mixture and the estimated number of gaussians in the mixture. In the second experiment, we test our algorithm on a data set generated from 3, 4, 5 and 6 endmembers. We represent respectively the data points, the true and the estimated endmembers, the true and estimated spectra and the estimation of the number of endmembers. We end up by applying our algorithm to the true image (Moffet). We estimate the number of endmembers and we show the estimated spectra. The performances of the algorithm has been compared via several criteria as Mean square error (MSE), Spectral angle distance (SAD), Root mean square error (RMSE), Global mean square error (GMSE) and the Reconstruction error (RE). The latter one is classically used to evaluate the quality of an unmixing method in the case of real hyperspectral images.

1 Nonparametric Bayesian Analysis

A statistical model tries to explain the data in terms of the properties of the system that generated it. Probabilistic models assume that the data have been generated from an unknown probability distribution which may or may not follow a general parametric form. The model structure M is defined in terms of random variables, referred collectively as the parameters, ψ. The model structure and the parameters are chosen such that the model can accurately represent the generative process that gave rise to the observed data. The Bayesian approach treats the parameters as being random quantities and therefore involves placing distributions over them, representing the prior belief. The prior is updated in light of the observations, giving the posterior. We denote the prior distribution of the parameters by $P(\psi|M)$ [4], [5].

The *likelihood function* is the probability density of the *observed* data X conditioned on the unknown model parameters ψ, therefore it is a function of ψ, $L(\psi|M) = P(X|\psi, M)$. The Bayes' rule yields the *posterior* density :

$$P(\psi|X, M) = \frac{P(\psi|M)\, P(X|\psi, M)}{P(X|M)}. \tag{1}$$

The denominator $P(X|M)$ obtained by integrating over the parameters,

$$P(X|M) = \int P(\psi|M)\, P(X|\psi, M)\, d\psi \tag{2}$$

is referred to as the *evidence* or the *marginal likelihood*. Note that this quantity does not depend on the parameters and it only appears as a normalizing constant for the posterior distribution of ψ. Therefore, we generally write

$$P(\psi|X, M) \propto P(\psi, M)\, P(X|\psi, M) \tag{3}$$

meaning the posterior for the parameters is proportional to the prior times the likelihood. Thus, the posterior distribution expresses the updated belief about the parameters ψ after observing data. A family of prior distributions F is said to be *conjugate* to the likelihood if the posterior is also distributed according to F. Using conjugate priors, the integral in the equation (2) can be analytically evaluated. However, the conjugate family is not rich enough to always match the prior belief. In this case, one would need to use priors from a larger family. Using a prior distribution from a more general family will typically result in the integral in the equation (2) being intractable, hence increased computational complexity in posterior calculations.

We referred to the set of all unknown variables in a model as the parameters, denoted by ψ. The term *parametric model* refers to the model that has a form that is expressed by a finite number of parameters. The likelihood $P(X|\psi)$ may be assumed to be of a known simple form such that obtaining the posterior distributions of interest

is straightforward. An alternative to parametric models is the *nonparametric models* which are models with (countably) infinitely parameters. Nonparametric models achieve high flexibility and robustness by defining the prior to be a nonparametric distribution from a space of all possible distributions.

2 Dirichlet Distribution and Dirichlet Process

2.1 Dirichlet Distribution

The Dirichlet distribution forms our first step toward understanding the Dirichlet process model (DPM) model. The Dirichlet distribution is a multi-parameter generalization of the Beta distribution and defines a distribution over distributions, i.e., the result of sampling a Dirichlet is a distribution on some discrete probability space.

The Dirichlet distribution with a base distribution $\mathbf{m} = \{m_1, m_2, ..., m_n\}$ and a concentration parameter α on $\boldsymbol{\pi} = \{\pi_1, \pi_2, ..., \pi_n\}$ is given by the formula [6]

$$P(\boldsymbol{\pi}; \alpha\mathbf{m}) = \frac{\Gamma(\sum_{i=1}^{n} \alpha_i)}{\prod_{i=1}^{n} \Gamma(\alpha_i)} \prod_{i=1}^{n} \pi_i^{\alpha_i - 1} = \frac{\Gamma(\alpha)}{\prod_{i=1}^{n} \Gamma(\alpha m_i)} \prod_{i=1}^{n} \pi_i^{\alpha\, m_i - 1} \tag{4}$$

where $\sum_{i=1}^{n} \pi_i = 1$, $\sum_{i=1}^{n} m_i = 1$, $\alpha = \sum_{i=1}^{n} \alpha_i$ and $m_i = \alpha_i / \alpha$

The mean and the covariance of the Dirichlet distribution are given by :

$$E[\pi_i] = \frac{\alpha\, m_i}{\sum_{j=1}^{n} \alpha\, m_i} = m_i \tag{5}$$

$$V[\pi_i] = \frac{m_i\,(1 - m_i)}{1 + \alpha} \tag{6}$$

$$C[\pi_i, \pi_j] = \frac{-m_i\, m_j}{1 + \alpha}. \tag{7}$$

By examining the equation (5), it can be seen that as α, the concentration parameter, is varied, the mean of the Dirichlet distribution does not change. In contrast, as α is increased, the covariance decreases. α is a precision parameter that controls how the distribution is concentrated around $\mathbf{m}$.

When $n = 2$, the Dirichlet distribution is equivalent to the beta distribution. Denoting the beta density's two hyperparameters by α and β, let $\pi \sim Beta(\alpha, \beta)$ indicates that

$$P(\pi|\alpha, \beta) = \frac{\Gamma(\alpha + \beta)}{\Gamma(\alpha)\,\Gamma(\beta)} \pi^{\alpha - 1}\ (1 - \pi)^{\beta - 1} \qquad \alpha, \beta\ > 0. \tag{8}$$

2.2 Dirichlet Process

It is often difficult to find simple parametric models which adequately describe complex, realistic datasets. Nonparametric statistical methods avoid assuming restricted functional forms, and thus allow the complexity and accuracy of the inferred model to grow as more data is observed. Strictly speaking, nonparametric models are rarely free of parameters, since they must have a concrete, computationally tractable representation. In Bayesian statistics, nonparametric methods typically learn distributions on function spaces, and thus effectively involve infinitely parameters. Complexity is controlled via appropriate prior distributions, so that small datasets produce simple predictions, while additional observations induce richer posteriors.

To motivate nonparametric statistical methods, consider De Finetti's representation of N infinitely exchangeable random variables [7] :

$$p\left(x_1, x_2, x_3, \ldots, x_N\right) = \int_{\Theta} p\left(\theta\right) \prod_{i=1}^{N} p\left(x_i|\theta\right) d\theta. \tag{9}$$

In general, this decomposition is only guaranteed when Θ is an infinite dimensional space of probability measures. Many Bayesian nonparametric methods thus involve families of computationally tractable distributions on probability measures [8].

In a fundamental paper on a Bayesian approach to nonparametric problems, Ferguson [9] defines a random process, called the Dirichlet process, whose samples functions are almost surely probability measures, and he derives many important properties of this process. The Dirichlet process provides a distribution on distributions with many attractive properties, and is widely used in practice.
Given the definition of the Dirichlet Distribution, the Dirichlet process can be defined as [10] :

> Let $\boldsymbol{\pi}$ be a set, and $\mathscr{B}$ a σ-field of subsets of $\boldsymbol{\pi}$. Let α be a finite, nonnull, nonnegative, finitely additive measure on $(\boldsymbol{\pi}, \mathscr{B})$. We say a random probability measure G on $(\boldsymbol{\pi}, \mathscr{B})$ is a Dirichlet process on $(\boldsymbol{\pi}, \mathscr{B})$ with parameter α and G_0, if for every $k = 1, 2, \ldots\ldots$ and measurable partition $(B_1, B_2, B_3, \ldots, B_k)$ on $\boldsymbol{\pi}$, the joint distribution of the random probabilities $(G(B_1), G(B_2), G(B_3), \ldots, G(B_k))$ is a Dirichlet distribution with parameters α and $(G_0(B_1), G_0(B_2), G_0(B_3), \ldots, G_0(B_k))$ that is
>
> $$(G(B_1), G(B_2), G(B_3), \ldots, G(B_k)) \sim \mathscr{D}(\alpha G_0(B_1), \alpha G_0(B_2), \ldots, \alpha G_0(B_k)) \tag{10}$$
>
> We denote the random probability measure G that has a Dirichlet Process distribution with *concentration parameter* α and *base distribution* G_0 by :
>
> $$G \sim \mathscr{D}(\alpha, G_0) \tag{11}$$
>
> Some authors define the Dirichlet Process using a single parameter by combining the two parameters to form the random measure $\boldsymbol{\alpha} = \alpha G_0$.

3 Dirichlet Process Mixture Model

Mixture models are often used to describe data which is distributed according to some set of underlying mechanisms where each data point is assumed to be independently generated by only one of these underlying distributions [11]. Finite mixture models can be expressed using the following equation (12) :

$$p(\mathbf{x}_i|\boldsymbol{\pi}, \boldsymbol{\theta}) = \sum_{k=1}^{M} \pi_k p(\mathbf{x}_i, \boldsymbol{\theta}_k) \tag{12}$$

where $\boldsymbol{\pi} = \{\pi_1, \pi_2, ..., \pi_M\}$ is the set of mixing proportions for component distributions such that $\sum_{k=1}^{M} \pi_k = 1$ and $\pi_k > 0$ and $\boldsymbol{\theta} = \{\theta_1, \theta_2, ..., \theta_M\}$ where $\boldsymbol{\theta}_k$ is a vector of parameters for the k^{th} component distribution for $k = 1, ..., M$.

The Dirichlet process mixture model extends the basic mixture model by applying a Dirichlet Process prior to the mixing proportions. This extension allows for a countably infinite number of mixture components [12], [13], [14].

Consider N data points, $\{x_1, x_2, \ldots, x_N\}$ each of which are assumed to have been independently generated by some distribution $F(\boldsymbol{\theta}_i)$ where $\boldsymbol{\theta}_i$ is the vector of parameters that defines the process generating observation $\mathbf{x}_i$. Under the Dirichlet process mixture model, $\boldsymbol{\theta}_i$ is generated by some unknown distribution G. Then, G is distributed according to the Dirichlet process, $\mathscr{D}(\alpha, G_0)$ where G_0 is the base distribution and α is the concentration parameter. Therefore, the complete model can be written as [14], [15] :

$$\begin{aligned} \mathbf{x}_i &\sim F(\boldsymbol{\theta}_i) \\ \boldsymbol{\theta}_i &\sim G \\ G &\sim \mathscr{D}(\alpha, G_0). \end{aligned} \tag{13}$$

The figure 1 represents a Dirichlet process mixture model.

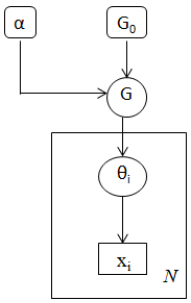

FIGURE 1 – Graphical representation of a Dirichlet process mixture model.

The data is assumed to be generated from a distribution parameterized by θ. The distribution of the parameter θ has a Dirichlet process prior with base distribution G_0 and concentration parameter α.

3.1 Generating Samples from a Dirichlet Process

How do we generate samples from a Dirichlet process ? In this section we describe the Pólya's Urn process, the Chinese restaurant process, and The Indian Buffet Process.

3.1.1 Pólya's Urn

A sequence $\{\theta_i\}_1^n$, $n \geq 1$ of random variables with values in $\mathscr{X}$ is a Pólya sequence with parameters α and G_0 if for every $\theta_i \in \mathscr{X}$

$$\theta_i \sim G$$

and

$$(\theta_{n+1}|\theta_1, \theta_2, \ldots, \theta_n) \sim G_n = \frac{\alpha G_0 + \sum_{i=1}^n \delta(\theta_i)}{\alpha + n} \tag{14}$$

where $\delta(\theta)$ denotes the unit measure concentrating at θ. Imagine $\mathscr{X}$ to be the set of colors of balls in an urn, with α being the initial number of balls, and G_0 the distribution of the colors of balls such that initially there are $\alpha\, G_0$ balls of color θ.

The sequence $\{\theta_i\}_1^n$described by the equation (14) represents the result of successive draws from the urn where after each draw, the ball drawn is replaced and another ball of the same color is added to the urn.

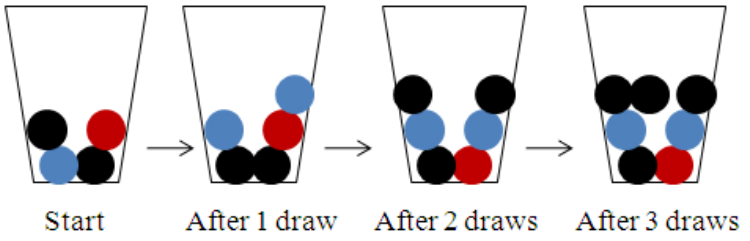

FIGURE 2 – Visualization of the urn-drawing scheme.

Blackwell and MacQueen [16] establish the connection between the Dirichlet process and Pólya sequences by extending the Pólya urn scheme to allow a continuum of colors. They show that for the extended scheme, the distribution of colors after n draws converges to a Dirichlet process as $n \to \infty$.

More formally, they state that if $\{\theta_i\}_1^n$ is a sequence of random variables constructed such that θ_1 has distribution G_0 and equation (14) holds, then

- G_n converges almost surely as $n \to \infty$ to a random discrete distribution G.
- G has a $\mathscr{D}(\alpha\, G_0)$ distribution
- The sequence $\{\theta_i\}_1^n$ is a sample from G.

For small values of α, G_n has only a few atoms whereas for large values, the atoms are numerous, concentrating on the G_0 distribution.

The figure 3 represents the DPM using the Pólya's urn process.
Note that G has been integrated out, and the parameters θ are drawn without

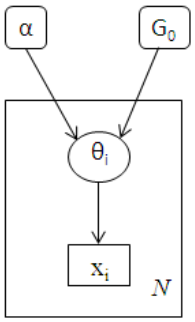

FIGURE 3 – Graphical representation of the DPM using the Pólya's urn process representation.

referring to G with the Pólya's urn scheme.

3.1.2 The Chinese Restaurant Process

The Pólya's urn scheme is closely related to the Chinese restaurant process (CRP) which is a distribution on partitions. The CRP is a sequential process that uses the metaphor of a Chinese restaurant with an infinite number of circular tables, each with infinite seating capacity. Customers arrive sequentially at the initially empty restaurant [4].
The first customer x_1 sits at the first table. For $n \geq 1$, suppose n customers have already entered the restaurant and are seated in some arrangement, occupying a total of K tables. Customer x_{n+1} chooses to sit next to customer x_l with equal probability $1/(n+\alpha)$ for each $1 \leq l \leq n$, and to sit alone at a new table with probability $\alpha/(n+\alpha)$. Denoting the table that customer i sits at as c_i,

$$P(c_{n+1} = k|c_1, c_2, \ldots, c_n) = \frac{\alpha}{\alpha+n}\delta(K+1) + \sum_{k=1}^{K} \frac{n_k}{\alpha+n}\delta(k) \tag{15}$$

or

$$P(c_{n+1} = k|c_1, c_2, \ldots, c_n) = \left\{ \begin{array}{c} \frac{n_k}{\alpha+n} \quad if\ k\ \leq\ K \\ \text{(i.e, k is a previously occupied table)} \\ \frac{\alpha}{\alpha+n} \quad otherwise \\ \text{(i.e, k is a next unoccupied table)} \end{array} \right\}$$

where n_k denotes the number of customers seated at table k.
After n customers have entered the restaurant, we have a partitioning of the customers $\{x_i\}_1^n$, the partitions being defined with the variables c_i. Ignoring the labeling of the tables and focusing on only the resulting partitioning, the customers are exchangeable. That is, the order in which they enter the restaurant does not play a

role in the resulting partitioning.
Suppose that independently of the sequence $\{x_i\}_1^n$ we paint each occupied table by picking colors ϕ_k from the distribution over the spectrum of possible colors, G_0. Letting θ_i denotes the color of the table occupied by the ith customer, the distribution of the colors would be given as

$$(\theta_{n+1}|\theta_1, \theta_2, \ldots, \theta_n) \sim \frac{\alpha}{\alpha + n} G_0 + \sum_{k=1}^{K} \frac{n_k}{\alpha + n} \delta(\phi_k) \tag{16}$$

Note that we have the same sequence of $\{\theta_i\}_1^n$ as defined by the Pólya's urn scheme given in the equation (14). Hence $\{\theta_i\}_1^n$ is a sample from $G \sim DP(\alpha, G_0)$.

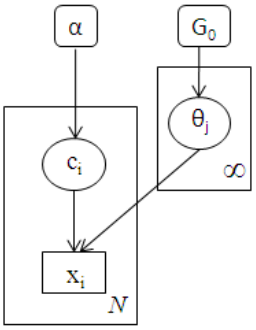

FIGURE 4 – Graphical representation of the DPM model using the Chinese Restaurant Process.

In the CRP framework, it becomes clear that the parameter assignments (coloring of the tables) is independent from the partitioning of the data (seating arrangement). This independence is shown in the graphical model in Figure 4 for the DPM using the CRP representation.

3.1.3 The Indian Buffet Process

In the Indian buffet process (IBP), N customers enter a restaurant one after another. Each customer encounters a buffet consisting of infinitely many dishes arranged in a line. The first customer starts at the left of the buffet and takes a serving from each dish, stopping after a Poisson (α) number of dishes as his plate becomes overburdened. The ith customer moves along the buffet, sampling dishes in proportion to their popularity, serving himself with probability $\frac{m_k}{i}$, where m_k is the number of previous customers who have sampled a dish. Having reached the end of all previous sampled dishes, the ith customer then tries a Poisson ($\frac{\alpha}{i}$)th number of new dishes. We can indicate which customers chose which dishes using a binary matrix $\mathbf{Z}$ with N rows and infinitely many columns, where $z_{ik} = 1$ if the ith customer sampled the kth dish.
The following figure 5 shows a matrix generated using the Indian Buffet Process [17]. The first customer tried 17 dishes. The second customer tried 7 of those dishes, and

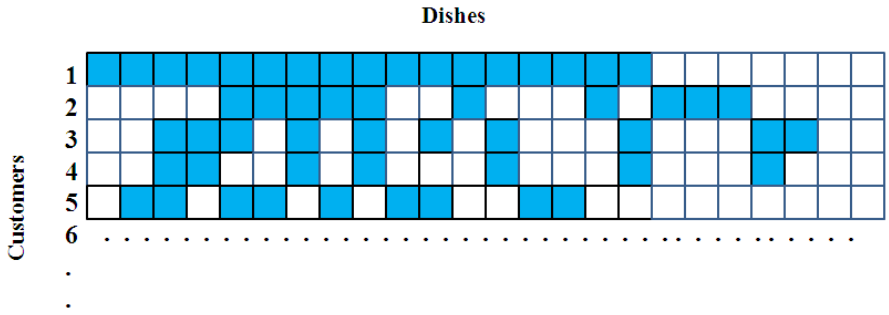

FIGURE 5 – A binary matrix generated by the Indian buffet process.

then tried 3 new dishes. The third customer tried 3 dishes tried by both previous customers, 5 dishes tried by only the first customer, and 2 new dishes. Vertically concatenating the choices of the customers produces the binary matrix shown in the figure 5.
Even though the buffet is infinite, it follows from the construction that each customer has a finite number of dishes with probability one, and thus, given a finite number of observations, we expect only a finite number of features to be present. The buffet analogy also highlights two important properties of the Indian Buffet Process. First, we expect the number of sampled dishes − or active features − to grow as the number of observations increases. Second, we expect there to exist a few popular features occurring in many observations and many rare features expressed in only a few observations.
Less obvious from the buffet construction is that the Indian Buffet Process is also infinitely exchangeable, that is, the order in which the customers attend the buffet as no impact on the distribution of $\mathbf{Z}$ up to permutations in the columns, and that columns are also independent [18].
Recall that in the buffet construction, the customers simply chose dishes based solely on their popularity. Once we note that permuting the columns should not affect the model, it is convenient to think of a canonical ordering for which all $\mathbf{Z}$ matrices that are the same up to column−permutations are equivalent. Griffiths [17] define a canonical representation called the *left-ordered form* of $\mathbf{Z}$, written as $[Z] = lof([Z])$. The left-ordered form first takes the binary sequence of $0's$ and $1's$ for each column referred to as a *history* h), treating the first customer as the most significant bit, and converts the binary sequence to a number. Thus, each column − or feature −receives a single value. We then order the columns by descending value.

4 Mixing models in hyperspectral imagery

In hyperspectral imagery, mixed pixels are a mixture of more than one distinct substance [1]. Spectral unmixing is the procedure by which the measured spectrum of a mixed pixel is decomposed into a collection of constituent spectra, or endmembers,

and a set of corresponding fractions, or abundances, that indicate the proportion of each endmember present in the pixel. Endmembers normally correspond to familiar macroscopic objects in the scene, such as water, soil, metal, vegetation, etc.

4.1 The linear mixing model

Analytical models for the mixing of disparate materials provide the foundation for developing techniques to recover estimates of the constituent substance spectra and their proportions from mixed pixels. The basic premise of mixture modeling is that within a given scene, the surface is dominated by a small number of distinct materials that have relatively constant spectral properties. These distinct substances (e.g., water, grass, mineral types) are called endmembers, and the fractions in which they appear in a mixed pixel are called fractional abundances. If most of the spectral variability within a scene is a consequence of endmembers appearing in varying proportions, it logically follows that some combination of their spectral properties can model the spectral variability observed by the remote sensing system.

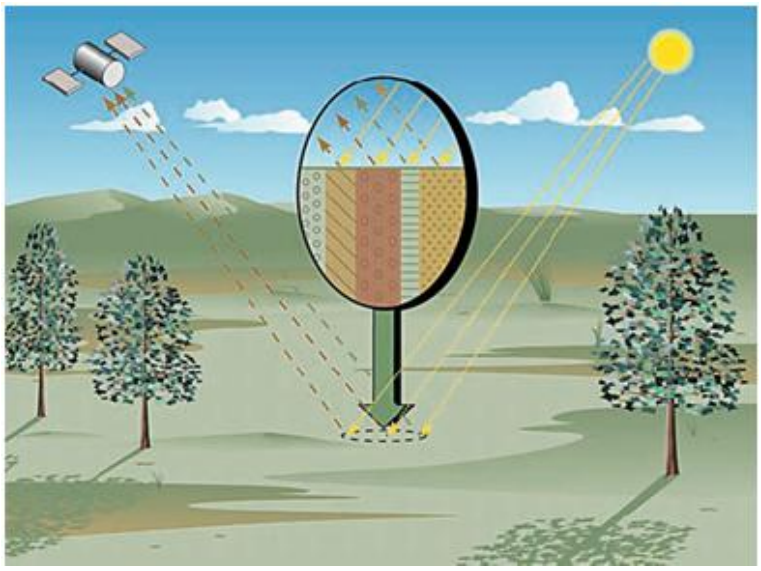

FIGURE 6 – Illustration of linear mixing.

The mathematical formulation for the linear mixing model can be expressed as [2] :

$$\boldsymbol{y} = \sum_{k=1}^{R} \alpha_k \boldsymbol{m}_k + \boldsymbol{n} = \boldsymbol{M}\boldsymbol{\alpha} + \boldsymbol{n} \tag{17}$$

where :

- $\boldsymbol{y} = (\boldsymbol{y}_1, ..., \boldsymbol{y}_L)^T$ is a vector of size L which represents the spectrum of a pixel of the image ;
- $\boldsymbol{m}_k = (m_{k1}, ..., m_{kL})^T$ is the k^{th} endmember ;
- L represents the number of spectral bands ;
- R represents the number of endmembers ;

- $\boldsymbol{n}$ is a white gaussian noise with zero mean and covariance matrix $\sigma^2 I_L$;
- $\boldsymbol{M} = (\boldsymbol{m}_1, ..., \boldsymbol{m}_R)$ is the matrix of size $L \times R$ of the endmembers ;
- $\boldsymbol{\alpha} = (\alpha_1, ..., \alpha_R)^T$ is the vector of size $R \times 1$ of the abundances.

The abundances contained in the vector $\boldsymbol{\alpha} = (\alpha_1, ..., \alpha_R)^T$ satisfy the following constraints :

$$\alpha_k \geq 0, \forall k \in 1, ..., R \quad \text{and} \quad \sum_{k=1}^{R} \alpha_k = 1. \tag{18}$$

4.2 The nonlinear mixing model

Nonlinear mixing models assume a randomly distributed, homogeneous mixture of materials, with multiple reflections of the illuminating radiation(see figure 7). These models represent the underlying physics at the foundation of hyperspectral phenomenology. Nonlinear models constitute a new interesting field of research for hyperspectral imagery and have shown interesting properties for abundance estimation, e.g., for scenes including mixtures of minerals, orchards, or vegetations.This model is generally written as follows [19], [20] :

$$\boldsymbol{y} = f(\boldsymbol{\alpha}, \boldsymbol{M}) + \boldsymbol{n}, \tag{19}$$

where f is an invertible nonlinear function.
In our study we considered only the linear mixing model.

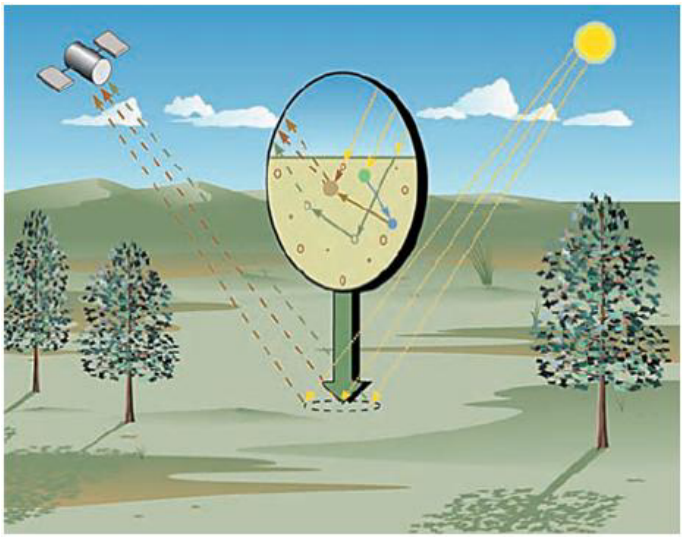

FIGURE 7 – Illustration of nonlinear mixing.

5 Applications of the Dirichlet Process Mixture Model (DPMM)

In this part we will see two applications of the Dirichlet process mixture model (DPMM). The first application deals with the problem of clustering and the second application concerns the problem of unmixing in hyperspectral imagery.

5.1 Clustering using DPMM

In this part we apply the Dirichlet Process to a Gaussian mixture model given by equation (20) for clustering :

$$p(x_i|\pi,\theta) = \sum_{i=1}^{M} \pi_i F(x|\mu_i,\Sigma_i) \tag{20}$$

where x is a D-dimensional continuous-valued data vector (i.e. measurement or features), $\pi_i = 1, ..., M$ are the mixture weights, and $F(x|\mu_i,\Sigma_i), i = 1, ..., M$ are the component Gaussian densities. Each component density is a D-variate Gaussian function of the form,

$$F(x|\mu_i,\Sigma_i) = \frac{1}{(2\pi)^{D/2}\,|\Sigma_i|^{\frac{1}{2}}} \exp\left\{-\frac{1}{2}\left(x-\mu_i\right)^{'}\Sigma_i^{-1}\left(x-\mu_i\right)\right\} \tag{21}$$

where $|.|$, μ_i, and Σ_i denote respectively the determinant operator, the mean vector and the covariance matrix. The mixture weights satisfy the constraint $\sum_1^M \pi_i = 1$, where M is the number of the Gaussians. The complete Gaussian mixture model is parameterized by the mean vectors, covariance matrices and mixture weights from all component densities. These parameters are collectively represented by the notation,

$$\theta = \left\{\pi_i, \mu_i, \Sigma_i\right\}, \; i = 1, \cdots, M$$

As shown by Neal [15], the likelihood of a data point given component parameters equation (21) can be combined with the probability of a class label given all other labels in equation (16). Then, the Gibbs sampler can be used to sample indicator variable values and component parameter values. The conditional probabilities for an indicator variable are defined in equation (22),

$$\begin{aligned} P(c_i = c_j \quad for\ some \quad j \neq i|c_{-i}, x_i, \theta) &= C\ \frac{n_{-i,j}}{\alpha+N-1}\ \times\ F\left(x_i|\theta_{c_j}\right) \\ P(c_i \neq c_j\ \forall\ j \neq i|c_{-i}, x_i) &= C\ \frac{\alpha}{\alpha+N-1}\ \int F(x_i|\theta)\ G_0(\theta)\ d\theta \end{aligned} \tag{22}$$

$$C = \frac{1}{\Sigma_j(\frac{n_{-i,j}}{\alpha+N-1}\ F(x_i|\theta_{c_j})) + \frac{\alpha}{\alpha+N-1}\ \int F(x_i|\theta)\ G_0(\theta)\ d\theta} \tag{23}$$

where α is the concentration parameter and C a normalizing constant computed by equation (23).

The Markov chain for the Gibbs sampler using the conditionals in equation (22) consists of all the indicator variables c and all component distribution parameters θ [15].
For the test, we assume there is a mixture of two-dimensional Gaussian variables, where the means and the covariances are unknown. But we know that the covariances are diagonal and isotropic. Therefore what we don't know are the mean μ, and the scalar factor σ . We use the Dirichlet process to model the problem and do clustering, the purpose of using Dirichlet process here is that we do not want to specify the number of components in the mixture, but instead give a prior over 1 to infinite. We assume it has a conjugate prior for μ and σ. Since the likelihood $F(x|\mu_i, \Sigma_i)$ is Gaussian, the conjugate prior which is called here the base distribution $G_0(\theta)$ should have a Normal-Gamma distribution.
Given hyperparameters λ,β,$\tau > 0$ and ν the conjugate prior $G_0(.,.)$ is

$$G_0(\mu, \sigma|\lambda, \beta, \tau, \nu) = \left\{ \begin{array}{cc} \frac{\sigma^{\lambda-1}\, exp(-\frac{\sigma}{\beta})}{\Gamma(\lambda)\, \beta^{\alpha}}\ (\frac{\sigma\tau}{2\pi})^{1/2}\ \exp(-\frac{\sigma\tau}{2}(\mu-\nu)^2) & where\ \ \sigma > 0 \\ 0 & otherwise \end{array} \right\}$$

More specifically, the marginal distribution of the precision parameter σ has Gamma distribution

$$f(\sigma|\lambda, \beta) = \left\{ \begin{array}{cc} \frac{\sigma^{\lambda-1}\, exp(-\frac{\sigma}{\beta})}{\Gamma(\lambda)\, \beta^{\alpha}} & where\ \ \sigma > 0 \\ 0 & otherwise \end{array} \right\}$$

and the marginal distribution of the mean parameter μ has a multivariate Student-t distribution with 2λ degrees of freedom, location ν and precision $\lambda\, \beta\, \tau$:

$$f(\mu|\lambda, \beta, \tau, \nu) = \sqrt{\frac{\beta\tau}{2\pi}}\ \frac{\Gamma(\frac{2\lambda+1}{2})}{\Gamma(\lambda)}\ (1 + \frac{\tau\beta}{2}(\mu-\nu)^2)^{-\frac{2\lambda+1}{2}}$$

which are our hyperparameters prior distributions. We can then define our model to be the following :

$$\begin{aligned} x_i|\mu_i, \sigma_i\ &\sim\ N(\mu_i, \sigma_i I_2) \\ \mu_i, \sigma_i|G\ &\sim\ G(\mu, \sigma) \\ G\ &\sim\ DP(\alpha\ G_0(\mu, \sigma)) \\ G_0\ &\sim\ NG(\mu, \sigma|\lambda, \beta, \tau, \nu) \end{aligned}$$

where $DP(\alpha\ G_0(\mu, \sigma))$ is the Dirichlet process with base measure G_0 and spread α, and G is a random distribution drawn from the DP. Our posteriors distributions which are based on the observations for these parameters are defined as follows : The posterior distribution for the base distribution $G_0(\theta)$ is defined like this :

$$G_0(\theta|x) = \frac{F(x|\theta)\ \times G_0(\theta)}{F(x)} \tag{24}$$

$$G_0(\mu, \sigma|x) = \frac{F(x|\mu, \sigma)\ \times G_0(\mu, \sigma|\lambda, \beta, \tau, \nu)}{F(x)} \tag{25}$$

where $F(x|\mu,\sigma) \propto N(\mu,\sigma)$ and $G_0 \propto NG(\mu,\sigma|\lambda,\beta,\tau,\nu)$
hence

$$G_0(\mu,\sigma|x) = NG(\mu,\sigma|\lambda^{'},\beta^{'},\tau^{'},\nu^{'})$$

and the posteriors marginal of the parameters μ, σ are :

$$f(\sigma|x) = \int G_0(\mu,\sigma|x)d\mu \tag{26}$$

$$f(\sigma|x) = Gamma(\sigma|\lambda^{'},\beta^{'})$$

$$f(\mu|x) = \int G_0(\mu,\sigma|x)d\sigma \tag{27}$$

$$f(\mu|x) = T_{2\lambda'}(\mu|\nu^{'},\frac{\beta^{'}}{\lambda^{'}\tau^{'}})$$

with

$$\lambda^{'} = \lambda + \frac{n}{2}$$

$$\beta^{'} = (\frac{1}{\beta} + \frac{1}{2\sum_{i=1}^{n}(x_i-\overline{x})^2 + \frac{\tau n(\overline{x}-\nu)^2}{2(\tau+n)}})^{-1}$$

$$\nu^{'} = \frac{\tau\nu + n\overline{x}}{\tau+n}$$

$$\tau^{'} = \tau + n$$

To determine the unknown parameters of the models of mixture, we use the Bayesian estimators like the MAP(maximum a posteriori) or the MMSE (Minimum mean square error).The calculation of the MMSE estimator raises problems of integration which we solve by using a MCMC method. We begin first by reminding the principle of the MMSE and MAP estimators before passing in the generation of samples following the a posteriori distributions mentioned previously.

5.1.1 Bayesian estimators

Estimator of the maximum a posteriori By definition the estimator of the MAP is calculated by maximizing the density a posteriori $f(\theta|y)$

$$\hat{\theta}_{MAP} = max_{\theta}\, f(\theta|y)$$

Estimator of the MMSE We are interested in this section in the estimator minimizing the mean square error $\left[\left(\theta - \widehat{\theta}_{MMSE}\right)^T \left(\theta - \widehat{\theta}_{MMSE}\right)\right]$. We show that :

$$\widehat{\theta}_{MMSE} = E\left[\theta|y\right] = \int \theta \, f(\theta|y) \, d\theta$$

So, the calculation of the estimator MMSE requires the determination of the average of the law a posteriori. Unfortunately, this integral still has no simple analytical expression. This problem is not appropriate to our model but constitute one of the major difficulties of the Bayesian inference. Diverse techniques were developed to approach such integral. The method used in this study is based on the MCMC methods. MCMC allows us to generate samples distributed according to the a posteriori distribution of interest. After a number of iterations, and whatever is the initial value of the chain $\theta^{(0)}$, The generated vectors $\theta^{(i)}$ are approximately distributed according to the a posteriori law $f(\theta|y)$. The estimator MMSE is then computed by the empirical average of the last elements of the chain.

$$\widehat{\theta}_{MMSE} = \frac{1}{N_r} \sum_{i=1}^{N_r} \theta^{(i+N_{bi})}$$

where N_{bi} represents the period of burning of the algorithm and N_r the number of samples to obtain a good digital approximation.

5.1.2 Gibbs Sampler

There are two essential techniques introduced into the literature to generate distributed samples following an a posteriori law : the algorithm of Metropolis-Hastings and the Gibbs sampler. We suggest here to use a Gibbs sampler allowing to generate a suite $\theta^{(i)}$ distributed according to the a posteriori law $f(\theta|y)$. For this, we have to sample the parameters θ_k according to their conditional laws by basing itself on the algorithm 1 [21].
The algorithm 1 describes the stages of the algorithm of Gibbs sampling for DPMM used here.

Algorithm 1 Gibbs sampling for Dirichlet process mixture model

Given $\alpha^{(t-1)}$, $\left\{\theta_k^{(t-1)}\right\}_{k=1}^{K}$ and $\left\{c_i^{(t-1)}\right\}_{i=1}^{n}$ from the previous iteration, sample a new set of $\left\{\theta_k^{(t)}\right\}_{k=1}^{K}$ and $\left\{c_i^{(t)}\right\}_{i=1}^{n}$ as follows :

1. Set $c = c^{(t-1)}, \alpha = \alpha^{(t-1)}$
2. For $i = 1, ..., n$
   a) Remove data item x_i from the cluster c_i since we are going to sample a new c_i for x_i.
   b) If x_i is the only data in its current cluster, this cluster becomes empty after step (2.a). This cluster is then removed, together with its parameter, and K is decreased by 1.
   c) Re-arrange cluster indices so that $1, ..., K$ are active (i.e ,non-empty)
   d) Draw a new sample for c_i from the following probabilities :

$$p(c_i = k, k \leq K) \propto \frac{n_{k,-i}}{n + \alpha - 1} F(x_i|\theta_k^{(t-1)}) \quad n_{k,-i} = \sum_{j \neq i} \delta(c_i - k)$$

$$p(c_i = K + 1) \propto \frac{\alpha}{n + \alpha - 1} \int F(x_i|\theta) G_0(\theta) d\theta$$

   e) If $c_i = K + 1$, we get a new cluster. Index this cluster as $K + 1$, sample a new cluster parameter ϕ_i from $H(\phi_i|x_i)$ defined as :

$$H(\phi|x_i) = \frac{F(x_i|\phi_i) \times G_0(\phi_i)}{\int F(x_i|\phi) \times G_0(\phi)\, d\phi}$$

   assign it to θ_{K+1} and increase K by 1.
3. For $k = 1, ..., K$
   Sample cluster parameter of each cluster θ_k from the following distribution :

$$\theta_k^{(t)} \quad \propto \quad G_0(\theta_k|x)\ \textit{Likelihood}\ (x_k^{(t)}|\theta_k^{(t-1)})$$

4. Set $c^{(t)} = c$

5.2 Unmixing using Dirichlet Process Mixture model

A hyperspectral image is a three-dimensional data-cube with one spectral and two spatial dimensions .The spectral dimension corresponds to wavelengths in which radiance is measured for each pixel [1]. Each pixel in a hyperspectral image is a spectral vector of radiance values. The purpose of unmixing in hyperspectral imagery is to determine the spectrally pure signatures (patterns) in a hyperspectral image that can be used to represent all the pixels in the image via a convex, linear model. Although similar to clustering, it differs in the sense that the endmembers represent vertices of a simplex that surround the pixels (at least partially) in D dimensional space, where D is the data dimensionality. Several algorithms exist for endmember detection, however, few simultaneously determine the number of endmembers [22], [23]. The proposed algorithm provides a method of determining the number of endmembers while simultaneously estimating endmember spectra and proportion maps for an image. The number of endmembers is determined by using the Dirichlet process. The algorithm provides a novel application of the Dirichlet process where, rather than determining means and variances of standard distributions like in the previous example of clustering , the algorithm determines vertices of a simplex which surround the pixels and the coefficients for a convex combination to describe each pixel in terms of the vertices (endmembers). This algorithm is initialized with a single endmember and more endmembers are incrementally added as needed. The proposed method determines the endmembers which surround the pixels and the proportion of each endmember in every pixel. The Dirichlet process is applied to determine the number of endmembers needed. This method differs from the DPMM since each pixel has an influence on every endmember whereas the DPMM partitions the pixels causing them to only influence the distribution parameters from which they are assumed to be generated. Furthermore, the proposed method does not only determine distribution parameters which instead, the algorithm determines a unique proportion vector for each pixel. The linear mixing model for hyperspectral imagery assumes that every pixel in a scene is a convex combination of the endmembers in the scene [1], [24].

$$x_i \;=\; \sum_{k=1}^{M} p_{ik}\, \mathbf{e}_k + \epsilon_i \quad i = 1, \cdots, N \tag{28}$$

where N is the number of pixels, M is the number of endmembers, ϵ_i is an error term, p_{ik} is the proportion of endmember k in pixel i , and e_k is the $k^{(th)}$ endmember [1]. The proportions satisfy the following constraints.

$$p_{ik} \;\geq 0 \qquad k = 1, \cdots, M \qquad \sum_{k=1}^{M} p_{ik} = 1 \tag{29}$$

Often, spectral unmixing is performed to determine the proportion values p_{ik} , in addition to determining the spectral signatures of the endmembers e_k [1]. Following this linear mixing model the likelihood for a given pixel can be defined in terms of the corresponding proportion vector and set of endmembers.

$$p(\mathbf{x}_i|\mathbf{p}_i, \mathbf{E}, \sigma_X) = \frac{1}{(2\pi\sigma_X^2)^{\frac{D}{2}}} \exp\left\{-\frac{1}{2\sigma_X^2}(\mathbf{x}_i - \mathbf{p}_i\mathbf{E})^T(\mathbf{x}_i - \mathbf{p}_i\mathbf{E})\right\} \tag{30}$$

where p_i is the proportion vector for the $i^{(th)}$ pixel, E is matrix containing all endmember values, D is the dimensionality of the pixel, and σ_X^2 is the variance. A conjugate Inverse-Gamma distribution with parameter $(\frac{\nu}{2})$ and $(\frac{\gamma}{2})$ is chosen as prior distribution for σ_X^2

$$\sigma_x^2|\nu, \gamma \ \sim \ IG(\frac{\nu}{2}, \frac{\gamma}{2}) \tag{31}$$

The hyperparameter ν will be fixed to $\nu = 2$ [25]. On the other hand, ν will be random and adjustable hyperparameter, whose prior distribution is defined below. The prior for ν is noninformative Jeffreys'prior [26], which reflects the lack of knowledge regarding this hyperparameter

$$f(\gamma) \ \propto \ \frac{1}{\gamma}\mathbf{1}_{R^+}(\gamma) \tag{32}$$

where $\mathbf{1}_{R^+}()$ is the indicator function defined on R^+.
Then as done in [24], encouraging the endmembers to have a tight fit around the data set can be done by using a sum of squared distances term.

$$p(\mathbf{E}|, \sigma_E) = \frac{1}{(2\pi\sigma_E^2)^{\frac{D}{2}}} \exp\left\{-\frac{1}{2\sigma_E^2}\sum_{k=1}^{M-1}\sum_{l=k+1}^{M}(\mathbf{e}_k - \mathbf{e}_l)^T(\mathbf{e}_k - \mathbf{e}_l)\right\} \tag{33}$$

where D is the dimensionality of the data, E is the matrix of endmembers, e_k is the $k^{(th)}$ endmember, and M is the number of endmembers.
The proposed algorithm uses the Dirichlet Process as a method to update proportion values for each endmember and incrementally add endmembers as needed. The proportion update for each endmember is done iteratively by incrementally increasing the weight of elements in the proportion vector. Given an initial proportion vector p_i for pixel x_i , we will draw δ_i (The incrementation step or the new abundance coefficient) following the Beta distribution $\delta_i \sim Beta(a, b)$ such that the mean of this distribution will be equal to the reconstructed error (the error between the pixel x_i and the reconstructed pixel p_iE) as follows :

$$mean_i = \frac{(\mathbf{x}_i - \mathbf{p}_i\mathbf{E})^T(\mathbf{x}_i - \mathbf{p}_i\mathbf{E})}{\beta} \tag{34}$$

where E is the set of endmembers, β is a step-size parameter used to normalize $mean_i$ between 0 and 1 and the variance of this distribution will be equal to a number chosen manually.
Using δ_i a set of potential proportion vectors, $\left\{p_i^1, p_i^2, \ldots, p_i^{M+1}\right\}$, updates is computed as follow :

$$\mathbf{p}_i^j = \begin{cases} \frac{1}{1+\delta_i}\left[p_{i1}, \ldots, p_{ij} + \delta_i, \ldots, p_{iM}\right] & j \leq M \\ \frac{1}{1+\delta_i}\left[p_{i1}, \ldots, \ldots, p_{iM}, \delta_i\right] & j = M+1 \end{cases} \tag{35}$$

The probability of selecting an update is computed using a Dirichlet Process prior and the likelihood of the data point given the updated proportion vector and endmember values. If the update increases the error, the probability is set to zero.

$$p(\mathbf{p}_i^j|\mathbf{x}_i,\mathbf{P},\mathbf{E},\sigma_X,\sigma_E) = \begin{cases} C\ \frac{m_j}{\alpha+N-1}\ F_{1:M}(\mathbf{p}_i^j) & j \leq M \\ C\ \frac{\alpha}{\alpha+N-1}\ F_{M+1}(\mathbf{p}_i^{M+1}) & j = M+1 \end{cases} \tag{36}$$

where

$$F_{1:M}(\mathbf{p}_i^j) = \begin{cases} p(\mathbf{x}_i|\mathbf{p}_i^j,\mathbf{E},\sigma_X)\ p(\mathbf{E}|\sigma_E) & if\ \left\|\mathbf{x}_i - \mathbf{p}_i^j\mathbf{E}\right\| \leq \|\mathbf{x}_i - \mathbf{p}_i\mathbf{E}\| \\ 0 & otherwise \end{cases} \tag{37}$$

$$F_{M+1}(\mathbf{p}_i^{M+1}) = \int p(\mathbf{x}_i|\mathbf{p}_i,\mathbf{E},\mathbf{e}_{M+1},\sigma_X)\ p(\mathbf{E}|\sigma_E)\ p(\mathbf{e}_{M+1}|\mathbf{E},\sigma_E)\ d\mathbf{e}_{M+1} \tag{38}$$

$m_j = \sum_{i=1}^{N} p_{ij}$, C is a normalizing constant, N is the number of pixels, σ_x and σ_E are the variances for the data likelihood and the prior on the endmembers, respectively, M is the number of endmembers with associated proportion values greater than zero, and P is the matrix of proportion values with p_{ij} the proportion value for the i^{th} pixel of the j^{th} endmember. If we take the prior on $p(\mathbf{e}_{M+1}|\mathbf{E},\sigma_E)$ equal to this :

$$p(\mathbf{e}_{M+1}|\mathbf{E},\sigma_E) = \frac{1}{(2\pi\sigma_E^2)^{\frac{D}{2}}}\ \exp\left\{-\frac{1}{2\sigma_E^2}\sum_{k=1}^{M-1}(\mathbf{e}_k - \mathbf{e}_{M+1})^T(\mathbf{e}_k - \mathbf{e}_{M+1})\right\} \tag{39}$$

The integral in (38) can be calculated as follows :

$$F_{M+1}(\mathbf{p}_i^{M+1}) = \int p(\mathbf{x}_i|\mathbf{p}_i,\mathbf{E},\mathbf{e}_{M+1},\sigma_X)\ p(\mathbf{E},\sigma_E)\ p(\mathbf{e}_{M+1}|\mathbf{E},\sigma_E)\ d\mathbf{e}_{M+1}$$

$$p(\mathbf{e}_{M+1}|\mathbf{E},\sigma_E) = \frac{1}{(2\pi\sigma_E^2)^{\frac{D}{2}}}\ \exp\left\{-\frac{1}{2\sigma_E^2}\sum_{k=1}^{M-1}(\mathbf{e}_k - \mathbf{e}_{M+1})^T(\mathbf{e}_k - \mathbf{e}_{M+1})\right\}$$

$$p(\mathbf{e}_{M+1}|\mathbf{E},\sigma_E) = \frac{1}{(2\pi\sigma_E^2)^{\frac{D}{2}}}\ \exp\left\{-\frac{1}{2\sigma_E^2}(\sum_{k=1}^{M-1}\mathbf{e}_k^T\mathbf{e}_k - 2*\mathbf{e}_{M+1}^T\sum_{k=1}^{M-1}\mathbf{e}_k + M\mathbf{e}_{M+1}^T\mathbf{e}_{M+1})\right\}$$

$$p(\mathbf{e}_{M+1}|\mathbf{E},\sigma_E) = \frac{1}{(2\pi\sigma_E^2)^{\frac{D}{2}}}\ \exp\left\{-\frac{1}{2\sigma_E^2}(M\mathbf{e}_{M+1}^T\mathbf{e}_{M+1} - 2*\mathbf{e}_{M+1}^T\sum_{k=1}^{M-1}\mathbf{e}_k + \sum_{k=1}^{M-1}\mathbf{e}_k^T\mathbf{e}_k)\right\}$$

$$p(\mathbf{e}_{M+1}|\mathbf{E},\sigma_E) = \frac{1}{(2\pi\sigma_E^2)^{\frac{D}{2}}}\ \exp\left\{-\frac{1}{\frac{2\sigma_E^2}{M}}(\mathbf{e}_{M+1}^T\mathbf{e}_{M+1} - 2*\mathbf{e}_{M+1}^T\frac{\sum_{k=1}^{M-1}\mathbf{e}_k}{M} + \frac{\sum_{k=1}^{M-1}\mathbf{e}_k^T\mathbf{e}_k}{M})\right\}$$

$$= \frac{1}{(2\pi\sigma_E^2)^{\frac{D}{2}}} \exp\left\{-\frac{1}{\frac{2\sigma_E^2}{M}}\left[(\mathbf{e}_{M+1} - \frac{\sum_{k=1}^{M-1}\mathbf{e}_k}{M})^2 - (\frac{\sum_{k=1}^{M-1}\mathbf{e}_k}{M})^T(\frac{\sum_{k=1}^{M-1}\mathbf{e}_k}{M}) + \frac{\sum_{k=1}^{M-1}\mathbf{e}_k^T\mathbf{e}_k}{M}\right]\right\} =$$

$$\frac{1}{(2\pi\sigma_E^2)^{\frac{D}{2}}} \exp\left\{-\frac{M}{2\sigma_E^2}\left[(\mathbf{e}_{M+1} - \frac{\sum_{k=1}^{M-1}\mathbf{e}_k}{M})^2\right]\right\}\ exp\left\{-\frac{M}{2\sigma_E^2}\left[\frac{\sum_{k=1}^{M-1}\mathbf{e}_k^T\mathbf{e}_k}{M} - (\frac{\sum_{k=1}^{M-1}\mathbf{e}_k}{M})^T(\frac{\sum_{k=1}^{M-1}\mathbf{e}_k}{M})\right]\right\}$$

$$= \frac{1}{M^{\frac{D}{2}}} \exp\left\{-\frac{M}{2\sigma_E^2}\left[\frac{\sum_{k=1}^{M-1}\mathbf{e}_k^T\mathbf{e}_k}{M} - (\frac{\sum_{k=1}^{M-1}\mathbf{e}_k}{M})^T(\frac{\sum_{k=1}^{M-1}\mathbf{e}_k}{M})\right]\right\} \quad \mathcal{N}\left(\frac{\sum_{k=1}^{M-1}\mathbf{e}_k}{M}, \frac{\sigma_E^2}{M}\right)$$

$$p(\mathbf{e}_{M+1}|\mathbf{E}, \sigma_E) \quad = \quad C \quad \mathcal{N}\left(\frac{\sum_{k=1}^{M-1}\mathbf{e}_k}{M}, \frac{\sigma_E^2}{M}\right)$$

$$p(\mathbf{x}_i|\mathbf{p}_i, \mathbf{E}, \mathbf{e}_{M+1}, \sigma_X) =$$

$$= \frac{1}{(2\pi\sigma_X^2)^{\frac{D}{2}}} \exp\left\{-\frac{1}{2\sigma_X^2}(\mathbf{x}_i - \sum_j^M \mathbf{p}_{ij}\mathbf{e}_j - \mathbf{p}_i(M+1)\mathbf{e}_{M+1})^T(\mathbf{x}_i - \sum_j^M \mathbf{p}_{ij}\mathbf{e}_j - \mathbf{p}_i(M+1)\mathbf{e}_{M+1})\right\}$$

$$p(\mathbf{x}_i|\mathbf{p}_i, \mathbf{E}, \mathbf{e}_{M+1}, \sigma_X) = \frac{1}{(2\pi\sigma_X^2)^{\frac{D}{2}}} \exp\left\{-\frac{1}{2\sigma_X^2}(h - \mathbf{p}_i(M+1)\mathbf{e}_{M+1})^T(h - \mathbf{p}_i(M+1)\mathbf{e}_{M+1})\right\}$$

$$= \frac{1}{(2\pi\sigma_X^2)^{\frac{D}{2}}} \exp\left\{-\frac{1}{2\sigma_X^2}\left[h^Th - 2(\mathbf{p}_i(M+1)\mathbf{e}_{M+1})^Th + (\mathbf{p}_i(M+1)\mathbf{e}_{M+1})^T(\mathbf{p}_i(M+1)\mathbf{e}_{M+1})\right]\right\}$$

$$= \frac{1}{(2\pi\sigma_X^2)^{\frac{D}{2}}} \exp\left\{-\frac{1}{2\sigma_X^2}\left[\ \mathbf{p}_i^2(M+1)\ \mathbf{e}_{M+1}^T\mathbf{e}_{M+1} - \ 2\mathbf{p}_i(M+1)\ \mathbf{e}_{M+1}^Th + h^Th\right]\right\}$$

$$= \frac{1}{(2\pi\sigma_X^2)^{\frac{D}{2}}} \exp\left\{-\frac{1}{\frac{2\sigma_X^2}{\mathbf{p}_i^2(M+1)}}\left[\ \mathbf{e}_{M+1}^T\mathbf{e}_{M+1} - \ \frac{2}{\mathbf{p}_i(M+1)}\ \mathbf{e}_{M+1}^Th + \frac{h^Th}{\mathbf{p}_i^2(M+1)}\right]\right\}$$

$$= \frac{1}{(2\pi\sigma_X^2)^{\frac{D}{2}}} \exp\left\{-\frac{1}{\frac{2\sigma_X^2}{\mathbf{p}_i^2(M+1)}}\left[\ (\mathbf{e}_{M+1} - \ \frac{h}{\mathbf{p}_i(M+1)})^2\ - \frac{h^Th}{\mathbf{p}_i^2(M+1)} + \frac{h^Th}{\mathbf{p}_i^2(M+1)}\right]\right\}$$

$$p(\mathbf{x}_i|\mathbf{p}_i, \mathbf{E}, \mathbf{e}_{M+1}, \sigma_X) = \frac{1}{(2\pi\sigma_X^2)^{\frac{D}{2}}} \exp\left\{-\frac{1}{\frac{2\sigma_X^2}{\mathbf{p}_i^2(M+1)}}\ (\mathbf{e}_{M+1} - \ \frac{h}{\mathbf{p}_i(M+1)})^2\right\}$$

$$p(\mathbf{x}_i|\mathbf{p}_i, \mathbf{E}, \mathbf{e}_{M+1}, \sigma_X) = \frac{1}{(\mathbf{p}_i(M+1))^{\frac{D}{2}}} \quad \mathcal{N}\left(\frac{h}{\mathbf{p}_i(M+1)}, \frac{\sigma_X^2}{\mathbf{p}_i^2(M+1)}\right)$$

$$p(\mathbf{x}_i|\mathbf{p}_i, \mathbf{E}, \mathbf{e}_{M+1}, \sigma_X) = A \quad \mathscr{N}\left(\frac{h}{\mathbf{p}_i(M+1)}, \frac{\sigma_X^2}{\mathbf{p}_i^2(M+1)}\right)$$

$$F_{M+1}(\mathbf{p}_i^{M+1}) = \int p(\mathbf{x}_i|\mathbf{p}_i, \mathbf{E}, \mathbf{e}_{M+1}, \sigma_X)\; p(\mathbf{E}, \sigma_E)\; p(\mathbf{e}_{M+1}|\mathbf{E}, \sigma_E)\; d\mathbf{e}_{M+1}$$

$$F_{M+1}(\mathbf{p}_i^{M+1}) = p(\mathbf{E}, \sigma_E) \int\; C\;\; \mathscr{N}\left(\frac{\sum_{k=1}^{M-1}\mathbf{e}_k}{M}, \frac{\sigma_E^2}{M}\right) A\;\; \mathscr{N}\left(\frac{h}{\mathbf{p}_i(M+1)}, \frac{\sigma_X^2}{\mathbf{p}_i^2(M+1)}\right) d\mathbf{e}_{M+1}$$

$$F_{M+1}(\mathbf{p}_i^{M+1}) = A.\; C.\; p(\mathbf{E}, \sigma_E) \int\;\; \mathscr{N}\left(\frac{\sum_{k=1}^{M-1}\mathbf{e}_k}{M}, \frac{\sigma_E^2}{M}\right)\;\; \mathscr{N}\left(\frac{h}{\mathbf{p}_i(M+1)}, \frac{\sigma_X^2}{\mathbf{p}_i^2(M+1)}\right) d\mathbf{e}_{M+1}$$

$$\mathscr{N}\left(\mu_1^T, \Sigma_1\right)\;.\;\mathscr{N}\left(\mu_2^T, \Sigma_2\right) = \;\mathscr{N}\left(\mu_T = (\Sigma_1^{-1}\mu_1 + \Sigma_2^{-1}\mu_2)\Sigma_T\;,\;\Sigma_T = (\Sigma_1^{-1} + \Sigma_2^{-1})^{-1}\right)$$

(see Appendices about the Product of n Multivariate Gaussian PDFs).

Updates to a proportion vector are performed by sampling an updated proportion vector using the probabilities defined by (37). However, prior to considering a new endmember, the proportion vector given only the current endmembers should be estimated. This prevents adding unnecessary endmembers when the current endmembers can adequately describe the data point. Therefore, estimating a proportion vector for a data point occurs in two stages. The first stage estimates the endmembers without considering the addition of a new endmember. This step is performed by sampling updated proportion vectors from (37) with α set to zero. After many iteration, the proportion vector is updating once with a non-zero value for α.
After updating a single proportion vector, endmember values are updated. Endmember values are determined by minimizing the likelihood of the dataset given the endmembers and the proportions. This is done by minimizing the posterior $p(E|X) = p(E|\sigma_e) * \prod_{i=1}^{N} p(x_i|p_i, E, \sigma_x)$ with respect to E. The calculation is as follows :

$$p(\boldsymbol{e}_k|\boldsymbol{X}) = p(\boldsymbol{X}|\boldsymbol{e}_k)\; p(\boldsymbol{e}_k) = \left[\prod_{i=1}^{N} p(x_i|\boldsymbol{e}_k)\right]\; p(\boldsymbol{e}_k)$$

$$p(\mathbf{E}|, \sigma_E) = \frac{1}{(2\pi\sigma_E^2)^{\frac{D}{2}}}\; \exp\left\{-\frac{1}{2\sigma_E^2}\sum_{k=1}^{M-1}\sum_{l=k+1}^{M}(\mathbf{e}_k - \mathbf{e}_l)^T(\mathbf{e}_k - \mathbf{e}_l)\right\}$$

$$p(\mathbf{E}|, \sigma_E) = \frac{1}{(2\pi\sigma_E^2)^{\frac{D}{2}}} \exp\left\{-\frac{1}{2\sigma_E^2}\;\frac{1}{2}\;\sum_{k=1}^{M}\sum_{l\neq k}^{M}(\mathbf{e}_k - \mathbf{e}_l)^T(\mathbf{e}_k - \mathbf{e}_l)\right\} = \frac{1}{(2\pi\sigma_E^2)^{\frac{D}{2}}}\; exp\left\{-\frac{1}{2\sigma_E^2}\;\frac{1}{2}\;a\right\}$$

$$a = \sum_{k=1}^{M}\left[(M-1)\;\mathbf{e}_k^T\mathbf{e}_k + 2\;\mathbf{e}_k^T\left(\sum_{l\neq k}^{M}\mathbf{e}_l\right) + \sum_{l\neq k}^{M}\;\mathbf{e}_l^T\mathbf{e}_l\right]$$

$$a = (M-1)\sum_{k=1}^{M}\left[\mathbf{e}_k^T\mathbf{e}_k + 2\;\mathbf{e}_k^T\frac{\left(\sum_{l\neq k}^{M}\mathbf{e}_l\right)}{(M-1)} + \frac{\sum_{l\neq k}^{M}\mathbf{e}_l^T\mathbf{e}_l}{(M-1)}\right]$$

$$a = (M-1)\sum_{k=1}^{M}\left\{\left[\mathbf{e}_k - \frac{\left(\sum_{l\neq k}^{M}\mathbf{e}_l\right)}{(M-1)}\right]^T\left[\mathbf{e}_k - \frac{\left(\sum_{l\neq k}^{M}\mathbf{e}_l\right)}{(M-1)}\right] + \frac{\sum_{l\neq k}^{M}\mathbf{e}_l^T\mathbf{e}_l}{(M-1)} - \frac{(\sum_{l\neq k}^{M}\mathbf{e}_l)^T(\sum_{l\neq k}^{M}\mathbf{e}_l)}{(M-1)^2}\right\}$$

so

$$p(\mathbf{e}_k|,\sigma_E) \;\propto\; \mathscr{N}\left(\frac{\sum_{l\neq k}^{M}\mathbf{e}_l}{(M-1)},\; 2\;\frac{\sigma_E^2}{(M-1)}\;I\right)$$

$$p(x_i|\mathbf{E}) \;\propto\; \frac{1}{(2\pi\sigma_X^2)^{\frac{D}{2}}}\exp\left\{-\frac{1}{2\sigma_X^2}(\mathbf{x}_i - \sum_{j}^{M}\mathbf{p}_{ij}\mathbf{e}_j)^T(\mathbf{x}_i - \sum_{j}^{M}\mathbf{p}_{ij}\mathbf{e}_j)\right\} = \frac{1}{(2\pi\sigma_X^2)^{\frac{D}{2}}}\,exp\left\{-\frac{1}{2\sigma_X^2}\beta\right\}$$

$$\beta = (\mathbf{x}_i - \sum_{j\neq k}^{M}\mathbf{p}_{ij}\mathbf{e}_j - \mathbf{p}_{ik}\mathbf{e}_k)^T(\mathbf{x}_i - \sum_{j\neq k}^{M}\mathbf{p}_{ij}\mathbf{e}_j - \mathbf{p}_{ik}\mathbf{e}_k) = (h - \mathbf{p}_{ik}\mathbf{e}_k)^T(h - \mathbf{p}_{ik}\mathbf{e}_k)$$

$$\beta = (h - \mathbf{p}_{ik}\mathbf{e}_k)^T(h - \mathbf{p}_{ik}\mathbf{e}_k) = \mathbf{p}_{ik}^2(\frac{h}{\mathbf{p}_{ik}} - \mathbf{e}_k)^T(\frac{h}{\mathbf{p}_{ik}} - \mathbf{e}_k)$$

so

$$p(x_i|\mathbf{e}_k) \;\propto\; \mathscr{N}\left(\frac{h}{\mathbf{p}_{ik}},\; \frac{\sigma_X^2}{\mathbf{p}_{ik}^2}\;I\right)$$

$$p(\boldsymbol{X}|\boldsymbol{e}_k) = \prod_{i=1}^{N}p(x_i|\mathbf{e}_k) \propto \frac{1}{\sqrt{2\pi}^N\sqrt{\prod_{i=1}^{N}\frac{\sigma_X^2}{\mathbf{p}_{ik}^2}}}\exp\left[-\sum_{i=1}^{N}\frac{(\boldsymbol{e}_k - \frac{h}{\mathbf{p}_{ik}})^2}{2\frac{\sigma_X^2}{\mathbf{p}_{ik}^2}}\right]$$

$$p(\boldsymbol{X}|\boldsymbol{e}_k) = \prod_{i=1}^{N}p(x_i|\mathbf{e}_k) \propto \frac{1}{\sqrt{2\pi}^N\sqrt{\prod_{i=1}^{N}\frac{\sigma_X^2}{\mathbf{p}_{ik}^2}}}\exp\left[-\sum_{i=1}^{N}\frac{(\boldsymbol{e}_k - \frac{h}{\mathbf{p}_{ik}})^T(\boldsymbol{e}_k - \frac{h}{\mathbf{p}_{ik}})}{2\frac{\sigma_X^2}{\mathbf{p}_{ik}^2}}\right]$$

$$\sum_{i=1}^{N}\frac{(\boldsymbol{e}_k - \frac{h}{\mathbf{p}_{ik}})^T(\boldsymbol{e}_k - \frac{h}{\mathbf{p}_{ik}})}{2\frac{\sigma_X^2}{\mathbf{p}_{ik}^2}} = \left(\sum_{i=1}^{N}\frac{1}{2\frac{\sigma_X^2}{\mathbf{p}_{ik}^2}}\right)\boldsymbol{e}_k^T\boldsymbol{e}_k - \left(\sum_{i=1}^{N}\frac{(\frac{h}{\mathbf{p}_{ik}})^T}{\frac{\sigma_X^2}{\mathbf{p}_{ik}^2}}\right)\boldsymbol{e}_k + \left[\sum_{i=1}^{N}\frac{(\frac{h}{\mathbf{p}_{ik}})^T(\frac{h}{\mathbf{p}_{ik}})}{2\frac{\sigma_X^2}{\mathbf{p}_{ik}^2}}\right]$$

$$\sum_{i=1}^{N}\frac{(\boldsymbol{e}_k - \frac{h}{\mathbf{p}_{ik}})^T(\boldsymbol{e}_k - \frac{h}{\mathbf{p}_{ik}})}{2\frac{\sigma_X^2}{\mathbf{p}_{ik}^2}} = \mathrm{B}\left[\boldsymbol{e}_k^T\boldsymbol{e}_k - \frac{\left(\sum_{i=1}^{N}\frac{(\frac{h}{\mathbf{p}_{ik}})^T}{\frac{\sigma_X^2}{\mathbf{p}_{ik}^2}}\right)}{\mathrm{B}}\boldsymbol{e}_k + \frac{\left[\sum_{i=1}^{N}\frac{(\frac{h}{\mathbf{p}_{ik}})^T(\frac{h}{\mathbf{p}_{ik}})}{2\frac{\sigma_X^2}{\mathbf{p}_{ik}^2}}\right]}{\mathrm{B}}\right]$$

so

$$p(\boldsymbol{X}|\boldsymbol{e}_k) \;\propto\; \mathscr{N}\left(\mu^T = \frac{\left(\sum_{i=1}^{N}\frac{(\frac{h}{\mathbf{p}_{ik}})^T}{\frac{\sigma_X^2}{\mathbf{p}_{ik}^2}}\right)}{\mathrm{B}}, \Sigma = \mathrm{B}^{-1}I\right)$$

$$\Rightarrow \quad p(\boldsymbol{e}_k|\boldsymbol{X}) \quad \propto \quad p(\boldsymbol{X}|\boldsymbol{e}_k)\, p(\boldsymbol{e}_k) = \mathcal{N}\left(\mu_1^T, \Sigma_1\right) \ . \ \mathcal{N}\left(\mu_2^T, \Sigma_2\right)$$

so (see Appendices about the Product of n Multivariate Gaussian PDFs)

$$\Rightarrow \quad p(\boldsymbol{e}_k|\boldsymbol{X}) \quad \propto \quad \mathcal{N}\left(\mu_T = (\Sigma_1^{-1}\mu_1 + \Sigma_2^{-1}\mu_2)\Sigma_T \ , \ \Sigma_T = (\Sigma_1^{-1} + \Sigma_2^{-1})^{-1} \right)$$

We know that the MMSE and the MAP of a normal distribution are its mean and we have found that the posterior of $p(E|X)$ follows a normal distribution so we can generate our endmembers following their means μ_i [27].

This algorithm proceeds in an online manner, in which a single pixel's proportion vector is updated followed by an update of the endmember values. A simple initialization for E is to randomly select a single datapoint from the dataset. In this case, each data point will only have a single proportion value of 1.

6 Experiments and Results

6.1 Test for different mixtures of Gaussians

The algorithm 1 was tested on the data's points which were generated following this model :

$$p(x_i|\pi,\theta) = \sum_{i=1}^{M} \pi_i \; F(x|\mu_i,\Sigma_i)$$

which was a mixture of two-dimensional Gaussians with mean prior as a multivariate Student-t distribution and variance prior as a gamma distribution. With this hyperparameters $\lambda = 5$, $\beta = 0.9$, $\tau = 0.05$, $\nu = 0$ and the concentration parameter $\alpha = 0.000001$ the result was as follows :

Mixture of 3 two-dimensional Gaussians : number of points=200 ; Proba=[0.3 0.2 0.5]

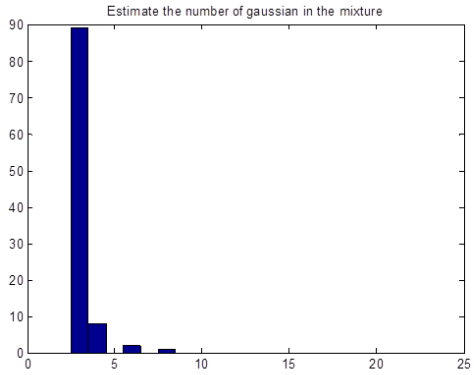

FIGURE 8 – Posterior distribution of the estimated number of gaussians of the mixture.

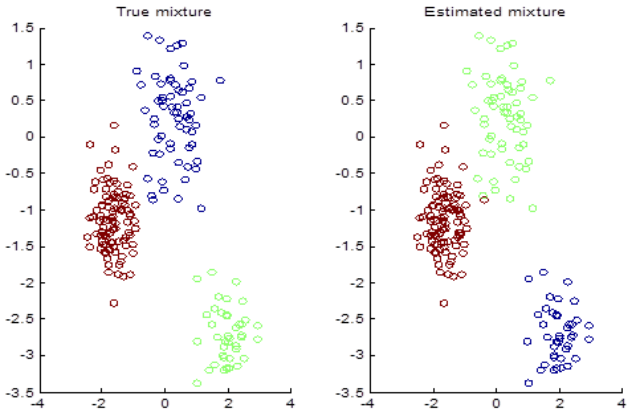

FIGURE 9 – True and estimated mixture.

Mixture of 4 two-dimensional Gaussians : number of points=200 ; Proba=[0.4 0.1 0.3 0.2]

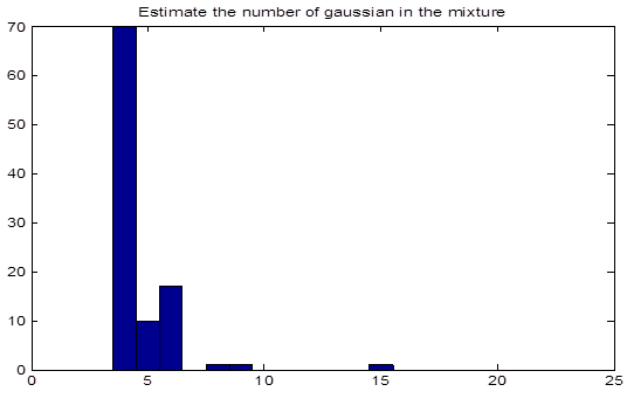

FIGURE 10 – Posterior distribution of the estimated number of gaussians of the mixture.

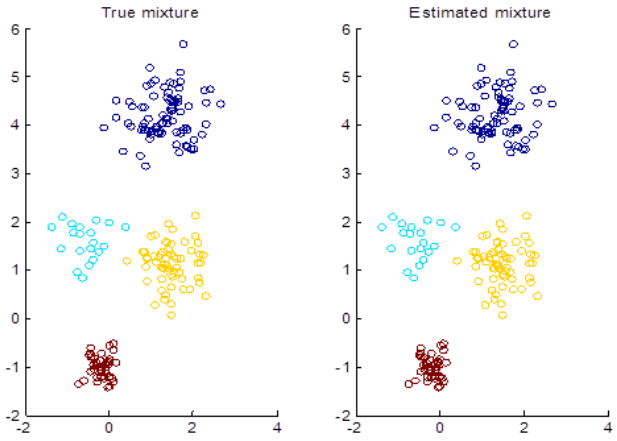

FIGURE 11 – True and estimated mixture.

Mixture of 5 two-dimensional Gaussians : number of points=200 ; Proba=[0.4 0.1 0.3 0.05 0.15]

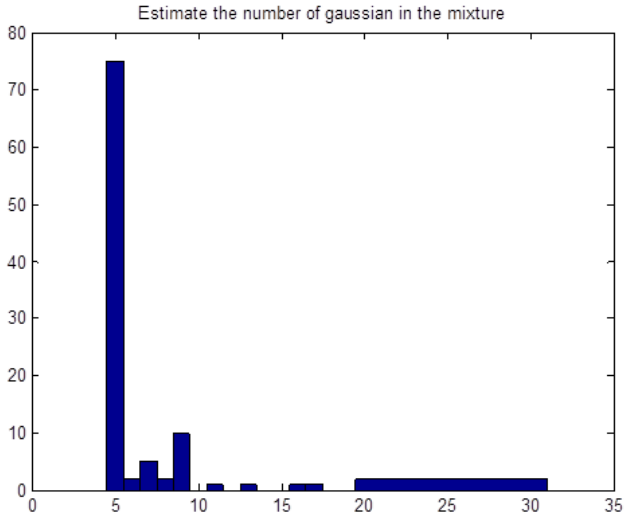

FIGURE 12 – Posterior distribution of the estimated number of gaussian of the mixture.

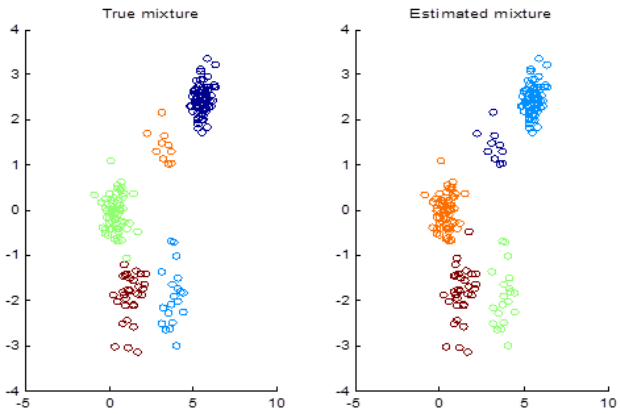

FIGURE 13 – True and estimated mixture.

		Means(μ_i)		precision(σ)		Mixture weights(π_i)	
		$MSE^2(\mu)$	$SAD(\mu)$	$MSE^2(\sigma)$	$SAD(\sigma)$	$GMSE^2$	RMSE
Mixture of 3 two-dimensional gaussians	1	0.0018	0.1095	0.0024	0	1.0e-04*0.25	0.0041
	2	0.0155	0.01	0.0025	0	0	
	3	0.0025	0.0212	0.0013	0	1.0e-04*0.25	
Mixture of 4 two-dimensional gaussians	1	0.0001	0.0008	0.0006	0	0	0
	2	0.0689	0.0605	0.0001	0	0	
	3	0.0021	0.0086	0.0015	0	0	
	4	0.0007	0.0245	0.0003	0	0	
Mixture of 5 two-dimensional gaussians	1	0.00001	0.0001	0.0006	0	0.0576	0.1364
	2	0.0464	0.0462	0.0001	0	0.0036	
	3	0.0027	0.0635	0.0001	0	0.0196	
	4	0.0182	0.0103	0.0008	0	0.0121	
	5	0.0074	0.0070	0.0016	0	0.0001	

TABLE 1 – Means μ, precision σ and Mixture weights(π_i) for different numbers of mixture of Gaussians

The Table 1 represents the means μ, the precision σ and Mixture weights(π_i) for a number M of gaussians

$MSE^2(\mu) = \|\hat{\mu_i} - \mu_i\|^2 \quad i = 1, \ldots, M$

$MSE^2(\sigma) = \|\hat{\sigma_i} - \sigma_i\|^2 \quad i = 1, \ldots, M$

$SAD(\mu) = \arccos\left(\frac{\langle \hat{\mu_i}\ ,\ \mu_i \rangle}{\|\mu_i\|\ \|\hat{\mu_i}\|}\right)$

$SAD(\sigma) = \arccos\left(\frac{\langle \hat{\sigma_i}\ ,\ \sigma_i \rangle}{\|\sigma_i\|\ \|\hat{\sigma_i}\|}\right)$

$GMSE^2 = (\hat{\pi_i} - \pi_i)^2$

$RMSE = \sqrt{(1/M) * \|\hat{\pi} - \pi\|^2}$, π is the vector of probabilities of the mixture.

MSE, GMSE and RMSE stand respectively for the mean square error, the global mean square error and the root mean square error.

The spectral angle distance (SAD) measures the angle between the actual and the corresponding estimated parameter.

6.2 Unmixing the endmembers

The algorithm is used for unmixing a spectra obtained by a mixture of 3, 4, 5 and 6 pure spectra (endmembers) which follows the linear mixing model (17) and uses normalized versions of the selected endmembers .The spectra is corrupted by a white noise with variance $\sigma^2 = 0.00012$ which corresponds to a $SNR = 30dB$. The following figures show the pure spectra used as well as the spectra of the mixture stemming from the linear mixing model.

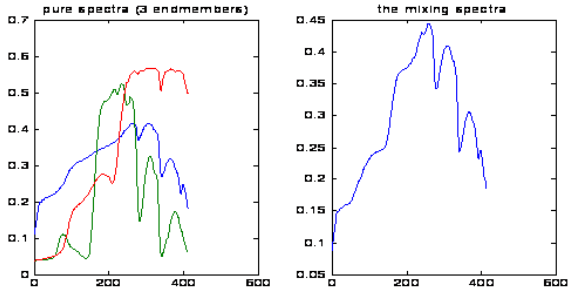

FIGURE 14 – Pure(3 endmembers) and mixing spectra.

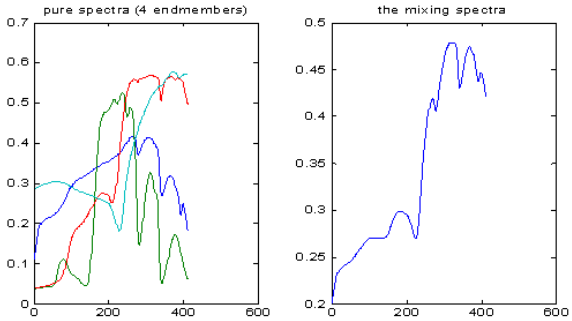

FIGURE 15 – Pure(4 endmembers)and mixing spectra.

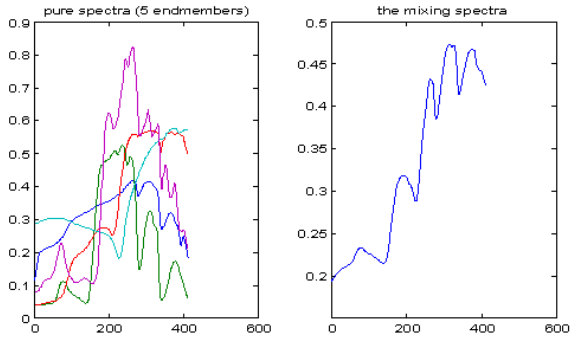

FIGURE 16 – Pure(5 endmembers) and mixing spectra.

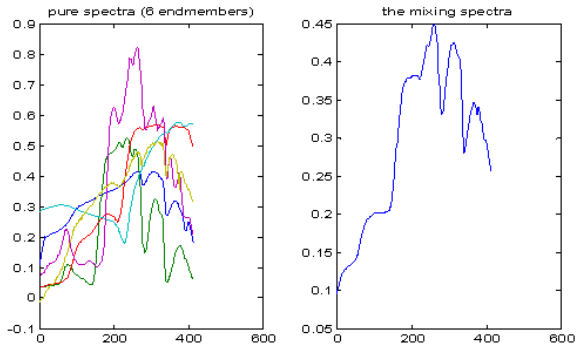

FIGURE 17 – Pure(6 endmembers) and mixing spectra.

the algorithm was run first 150 iterations on a five-dimensional data set to estimate the number of endmembers and then 300 iterations on R-1 dimensional data set (where R is the estimated number) but this time we will fix the number of endmembers by using the estimated number to have a better estimation of the endmembers generated from 3 , 4, 5, and 6 spectra, with 100 iterations per proportion vector update (for each iteration we do 100 iteration with α temporarily set to 0 to have a better estimation of the abundances). The data set used in the test was chosen using the function *convhulln* which allows us to select only the edge of our simplex from different dimensions (in the test we have used the fifth dimension) as shown in the following figures.

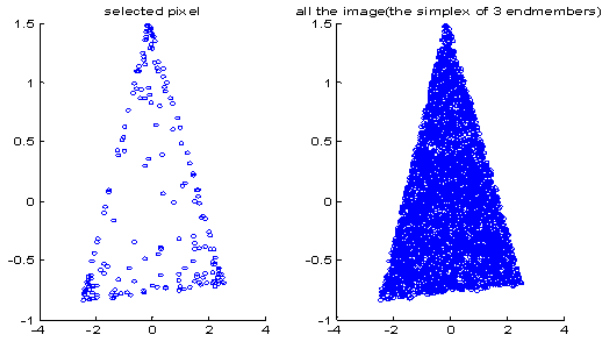

FIGURE 18 – Simplex of 3 endmembers.

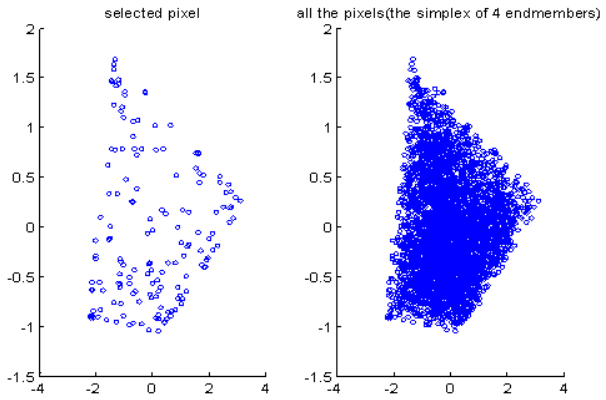

FIGURE 19 – Simplex of 4 endmembers.

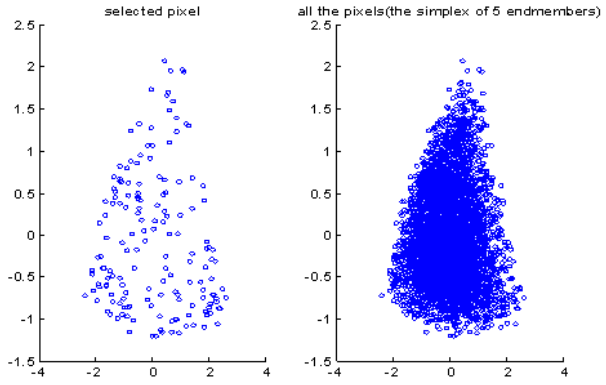

FIGURE 20 – Simplex of 5 endmembers.

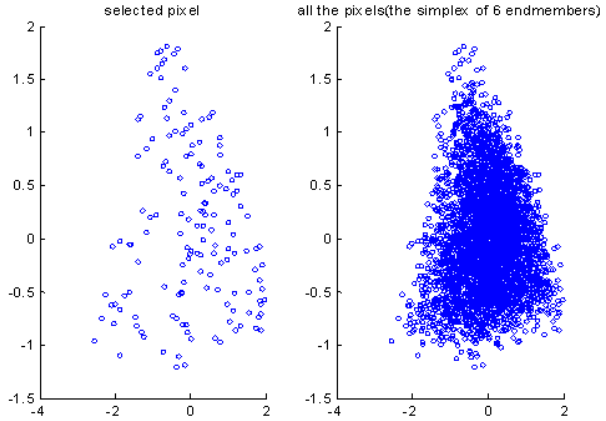

FIGURE 21 – Simplex of 6 endmembers.

The results in the following figures show the endmembers found by using these values of the parameters $\beta = 5$, $\alpha = 13000$, $\sigma_X = 1$ and $\sigma_E = 3$ for 6 endmembers ;then $\alpha = 300$ for 5 endmembers , $\alpha = 100$ for 4 endmembers and $\alpha = 10$ for 3 endmembers and the true image (Moffet).

Three endmembers :

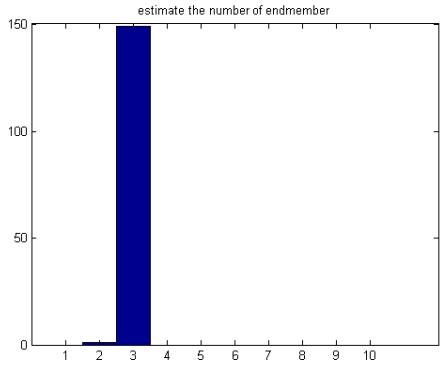

FIGURE 22 – Posterior distribution of the estimated number of endmembers.

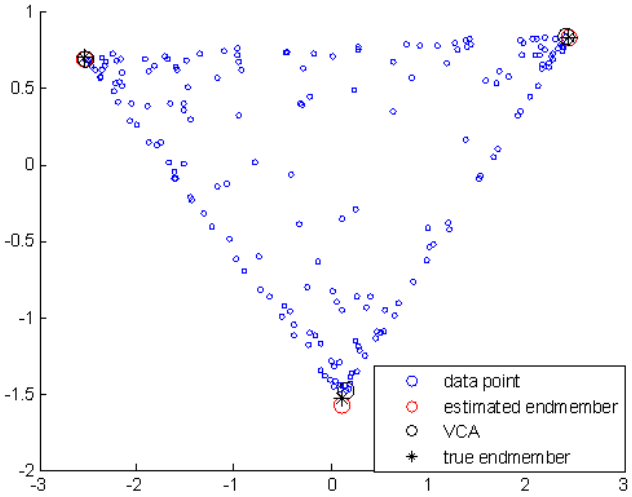

FIGURE 23 – Representation of the three endmembers(true,estimated by our algorithm and by VCA).

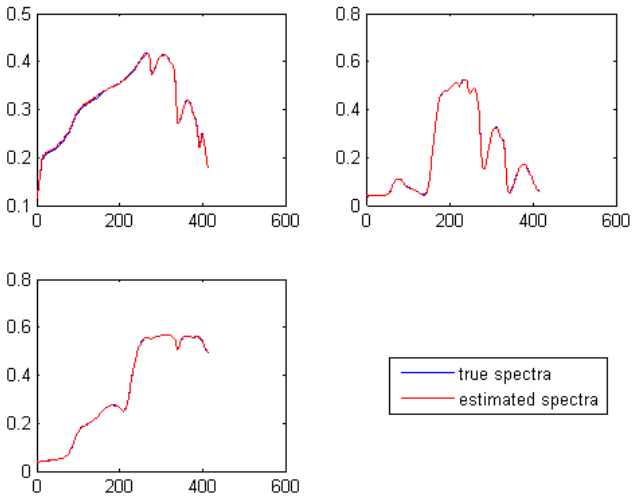

FIGURE 24 – Representation of the three spectra.

Four endmembers :

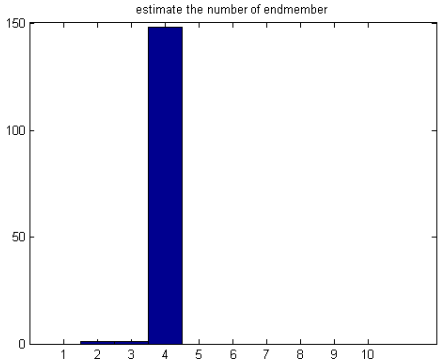

FIGURE 25 – Posterior distribution of the estimated number of endmembers.

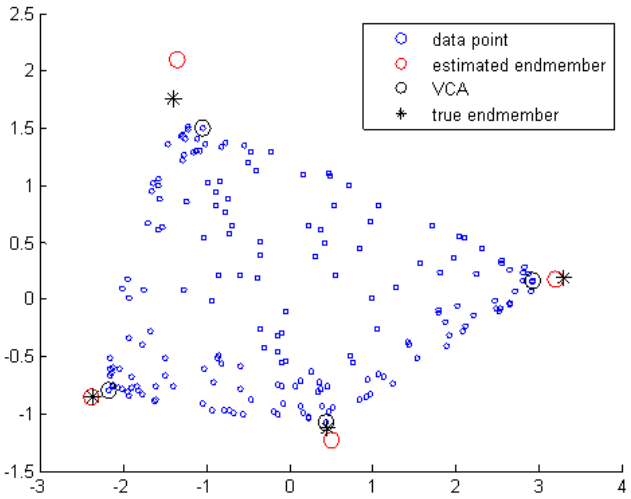

FIGURE 26 – Representation of the four endmembers(true,estimated by our algorithm and by VCA).

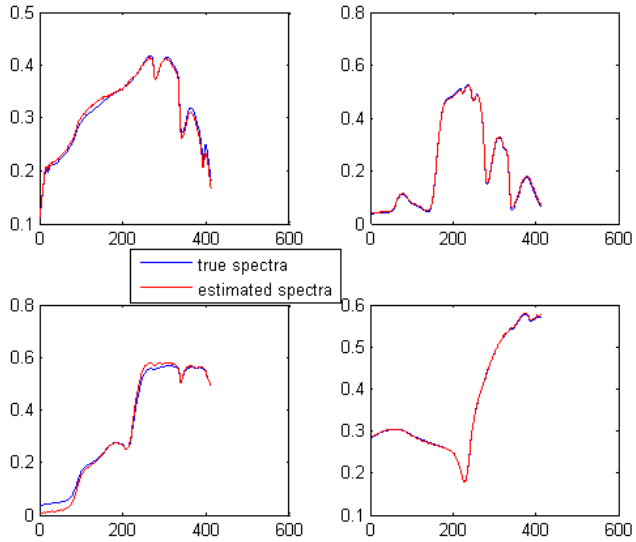

FIGURE 27 – Representation of the four spectra.

Five endmembers :

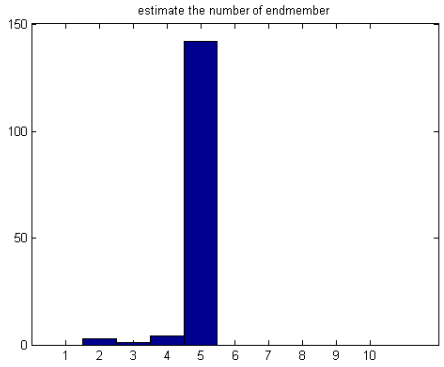

FIGURE 28 – Posterior distribution of the estimated number of endmembers.

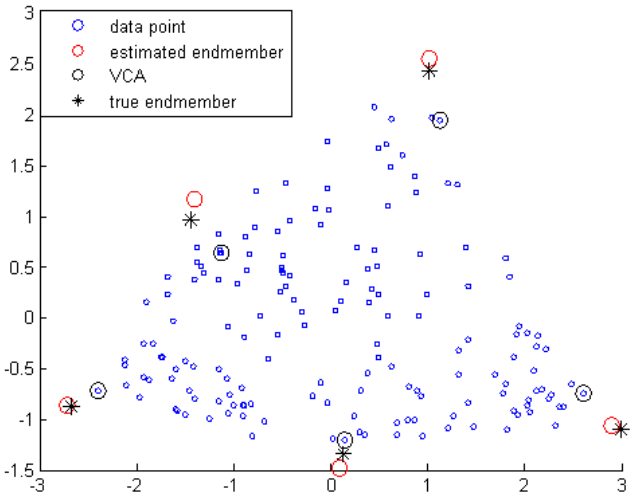

FIGURE 29 – Representation of the five endmembers(true,estimated by our algorithm and by VCA).

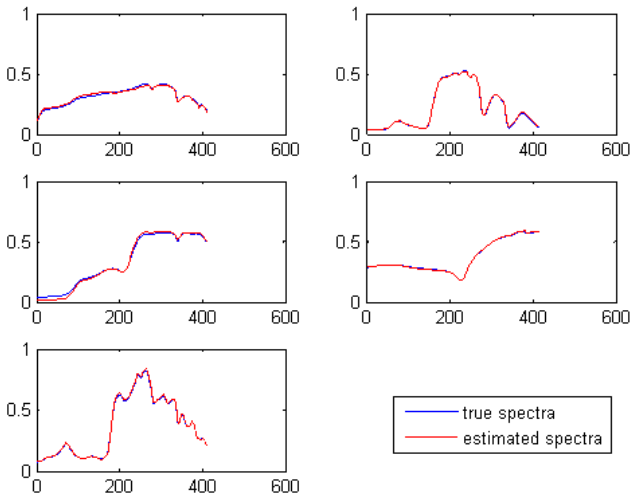

FIGURE 30 – Representation of the five spectra.

Six endmembers :

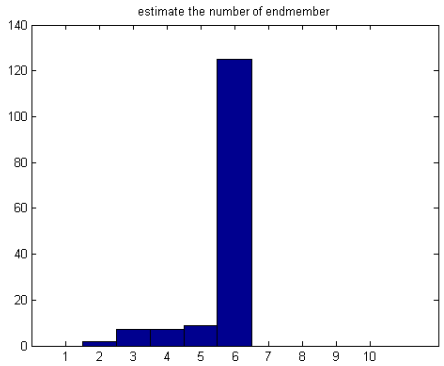

FIGURE 31 – Posterior distribution of the estimated number of endmembers.

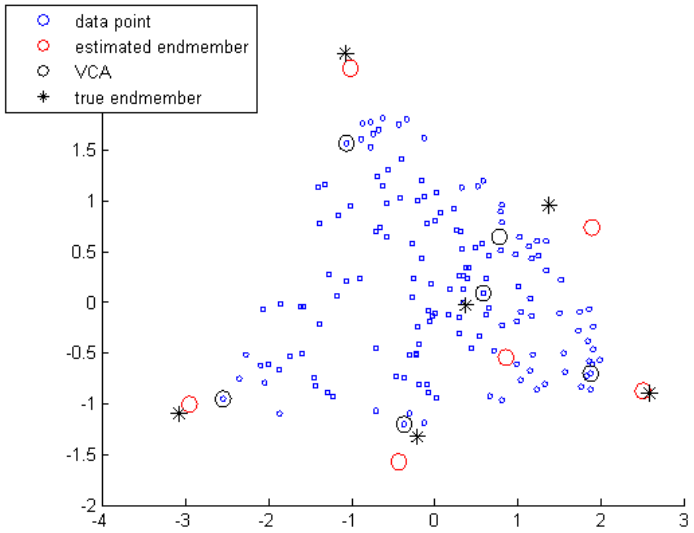

FIGURE 32 – Representation of the six endmembers(true,estimated by our algorithm and by VCA).

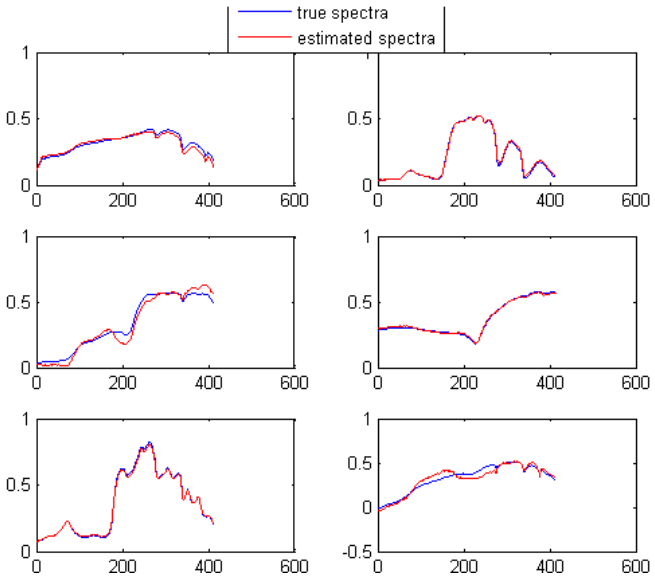

FIGURE 33 – Representation of the six spectra.

As it can be seen,the correct number of endmembers were found on all test runs. Also, the endmembers generated provide a tight fit around the test set.

True image (Moffet)

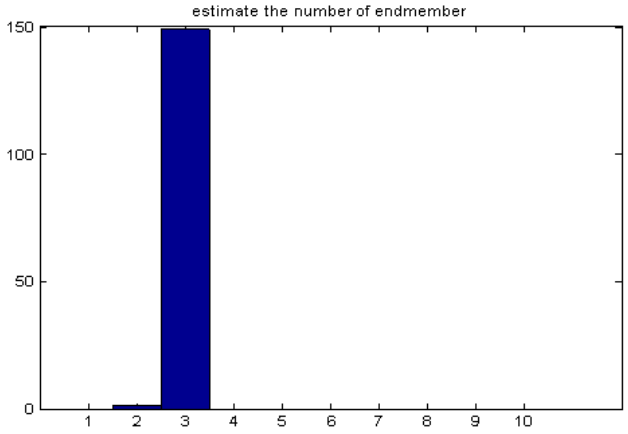

FIGURE 34 – Posterior distribution of the estimated number of endmembers for the true image(Moffet).

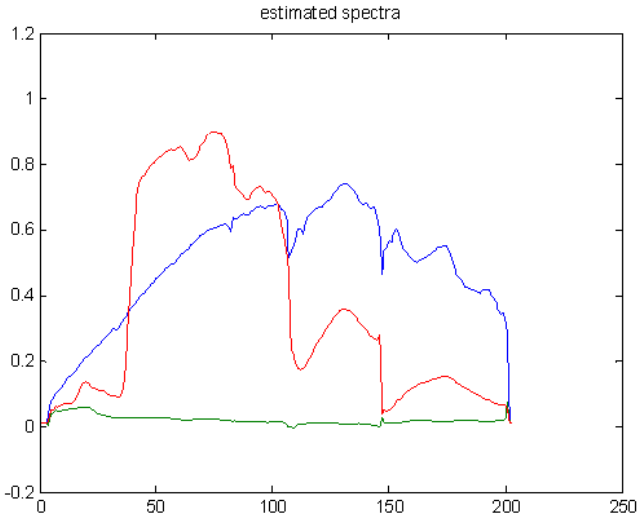

FIGURE 35 – Estimated spectra.

6.3 Unmixing performance

Criteria estimation for endmembers : the performances of the algorithm has been compared via two criteria. First, the mean square errors (MSEs)

$$MSE_r^2 = \|\widehat{\mathbf{m}}_r - \mathbf{m}_r\|\|^2 \quad r = 1, \ldots, R \tag{40}$$

are good quality indicators for the estimates. In addition, another metric frequently encountered in hyperspectral imagery literature, known as the spectral angle dis-

tance (SAD), has been considered. The SAD measures the angle between the actual and the corresponding estimated spectrum [28].

$$SAD_r = \arccos\left(\frac{\langle \widehat{\mathbf{m}}_r, \mathbf{m}_r \rangle}{\|\widehat{\mathbf{m}}_r\| \ \|\mathbf{m}_r\|}\right) \tag{41}$$

where $\langle ., . \rangle$ stands for the scalar product.

Criteria estimation for abundances : The quality of the unmixing strategy for synthetic images can be measured by comparing the estimated and actual abundances by using the root mean square error (RMSE) [29], [28].

$$RMSE = \sqrt{\frac{1}{nR}\sum_{p=1}^{n} \|\boldsymbol{\alpha}(p) - \widehat{\boldsymbol{\alpha}}(p)\|^2} \tag{42}$$

where $\boldsymbol{\alpha}(p)$ and $\widehat{\boldsymbol{\alpha}}(p)$ are the actual and estimated abundance vectors of the p^{th} pixel of the image and n is the number of pixels. The global abundance MSEs have been computed as

$$GMSE_r^2 = \sum_{p=1}^{P} (\widehat{a}_{p,r} - a_{p,r})^2 \tag{43}$$

where $\widehat{a}_{p,r}$ is the estimated abundance coefficient of the material $\#r$ in the pixel $\#p$ In the case of real hyperspectral images, the reconstruction error (RE) is classically used to evaluate the quality of an unmixing method.

$$RE = \sqrt{\frac{1}{nL}\sum_{p=1}^{n} \|\boldsymbol{y}(p) - \widehat{\boldsymbol{y}}(p)\|^2} \tag{44}$$

where L is the number of spectral bands and $y(p)$ and $\widehat{y}(p)$ are the measured and estimated spectra for the pixel p.

The experiments were carried out by using three methods :Our algorithm, the Vertex Component Analysis (VCA) & the VCA/Fully constrained least squares(VCA/FCLS) [30] and the method given by Zare.

The results are contained in the Tables 2,3 and 4.

	endmembers	Bayesian		VCA		Zare	
		MSE^2	SAD	MSE^2	SAD	MSE^2	SAD
$R=3$	#1	0.0020	0.0067	0.0043	0.0087	0.0107	0.0155
	#2	0.0002	0.0027	0.0006	0.0044	0.0188	0.0243
	#3	0.0001	0.0015	0.0003	0.0018	0.0273	0.0150
$R=4$	#1	0.0195	0.0214	1.5614	0.1799	0.0120	0.0150
	#2	0.0103	0.0182	0.0354	0.0337	0.1858	0.0767
	#3	0.1286	0.0419	0.0111	0.0092	0.0325	0.0171
	#4	0.0010	0.0038	0.0176	0.0137	0.0742	0.0274
$R=5$	#1	0.0392	0.0303	0.0209	0.0206	0.0256	0.0239
	#2	0.0109	0.0186	0.3003	0.0841	0.3338	0.0995
	#3	0.0950	0.0362	0.5229	0.0835	0.0953	0.0306
	#4	0.0042	0.0071	0.1304	0.0431	0.2273	0.0537
	#5	0.0188	0.0076	0.2719	0.0269	0.2005	0.0356
$R=6$	#1	0.1578	0.0591	0.0624	0.0376	0.2766	0.0743
	#2	0.0238	0.0264	0.3065	0.0976	0.7405	0.1464
	#3	0.6027	0.0972	0.5012	0.0870	0.2597	0.0641
	#4	0.0249	0.0200	0.5912	0.0863	0.4110	0.0746
	#5	0.0284	0.0137	0.8182	0.0552	0.4421	0.0548
	#6	0.8026	0.1228	0.1034	0.0441	Not detected	Not detected
	Times(s)	5000		2		1800	

TABLE 2 – Estimation of the endmembers by using three methods :Bayesian, VCA and Zare

	abundances	Bayesian		VCA/FCLS		Zare	
		$GMSE^2$	$RMSE$	$GMSE^2$	$RMSE$	$GMSE^2$	$RMSE$
$R=3$	#1	0.0158	0.0067	0.0143	0.0073	0.3360	0.0325
	#2	0.0048		0.0133		0.1273	
	#3	0.0041		0.0017		0.1192	
$R=4$	#1	0.1201	0.0339	1.6491	0.0713	0.2237	0.0339
	#2	0.0372		0.6916		0.2549	
	#3	0.3551		1.1915		0.1476	
	#4	0.3216		0.1532		0.2047	
$R=5$	#1	0.0917	0.0229	0.2384	0.0475	0.2384	0.0402
	#2	0.0372		0.4577		0.4597	
	#3	0.1880		0.4120		0.1451	
	#4	0.0873		0.3926		0.2919	
	#5	0.0251		0.3374		0.1816	
$R=6$	#1	0.2205	0.0561	0.4550	0.0683	3.4271	0.1045
	#2	0.0688		0.3619		0.5429	
	#3	0.4823		0.8953		2.8174	
	#4	0.0331		1.0606		0.9097	
	#5	1.0508		0.8502		0.5400	
	#6	0.9982		0.5987		Not detected	

TABLE 3 – Estimation of the abundances by using three methods :Bayesian, VCA/FCLS and Zare

RE	Bayes	VCA/FCLS	ZARE
$R=3$	0.0101	0.0101	0.0241
$R=4$	0.0104	0.0206	0.0175
$R=5$	0.0109	0.0134	0.0121
$R=6$	0.0110	0.0144	0.0137
$Moffet$	0.0291	0.0340	0.0368

TABLE 4 – Reconstruction error for the three methods : Bayesian, VCA/FCLS and Zare

Conclusions

The present work consisted of the study of the various models based on the Dirichlet process and then applied to the imaging hyperspectral for the linear unmixing according to the various formulations exposed in the literature. The report was organized as follows :
In the first chapter we describe Bayesian nonparametric (BNP) models. The BNP approach is to fit a single model that can adapt its complexity to the data. Furthermore, BNP models allow the complexity to grow as more data are observed.
The second chapter deals with the Dirichlet distribution which forms our first step toward understanding the Dirichlet process model (DPM). The DPM model provides a distribution on distributions with many attractive properties and is widely used in practice.
The third chapter deals with the Dirichlet process mixture model which extends the basic mixture model by applying a Dirichlet process prior to the mixing proportions. After that, we describe three processes (Pólya's Urn, the Chinese Restaurant and the Indian Buffet) which allow us to generate samples from a Dirichlet process.
The fourth chapter deals with mixing models in hyperspectral imagery. Mixed pixels are a mixture of more than one distinct substance.
In the fifth chapter, we see two applications of the Dirichlet process mixture model (DPMM). The first application deals with the problem of clustering and the second application concerns the problem of unmixing in hyperspectral imagery.
In the sixth chapter, we give some results concerning the simulations. We begin by generating a mixture of (3,4 and 5) two-dimensional Gaussians and we represent the estimated mixture and the estimated number of gaussians in the mixture. In the second experiment, we test our algorithm on a data set generated from 3, 4, 5 and 6 endmembers. We represent respectively in different figures the data points, the true and the estimated endmembers, the true and estimated spectra and the estimation of the number of endmembers. As it can be seen,the correct number of endmembers were found on all test runs. Also, the endmembers generated provide a tight fit around the test set. We end up by applying our algorithm to the true image (Moffet). We then estimate the number of endmembers and we show the estimated spectra.
The performances of the algorithm has been compared via several criteria as Mean square error (MSE), Spectral angle distance (SAD), Root mean square error (RMSE), Global mean square error (GMSE) and the Reconstruction error (RE). The latter one is classically used to evaluate the quality of an unmixing method in the case of real hyperspectral images. Three method were used in order to do the experiments : :Our algorithm, the Vertex Component Analysis (VCA) & the VCA/Fully constrained least squares(VCA/FCLS) and the method given by Zare.

Appendices

A The Product of n Multivariate Gaussian PDFs

The multivariate Gaussian PDF can be written as

$$p(\boldsymbol{x}) = \frac{1}{(2\pi)^{d/2}\sqrt{|\mathbf{V}|}} \exp\left[(\boldsymbol{x}-\boldsymbol{\mu})^T \mathbf{V}^{-1} (\boldsymbol{x}-\boldsymbol{\mu})\right]$$

where d is the dimensionality of $\boldsymbol{x}$, $\boldsymbol{\mu}$ is the d-dimensional mean vector, and $\mathbf{V}$ is the d-by-d dimensional covariance matrix ; this document adopts the standard notation of using bold face symbols to represent vectors and matrices. The Gaussian PDF can also be written in canonical notation as

$$p(\boldsymbol{x}) = \exp\left[\boldsymbol{\zeta} + \boldsymbol{\eta}^T \boldsymbol{x} - \frac{1}{2}\, \boldsymbol{x}^T \boldsymbol{\Lambda} \boldsymbol{x}\right] \tag{45}$$

where $\boldsymbol{\Lambda} = \mathbf{V}^{-1}$, $\boldsymbol{\eta} = \mathbf{V}^{-1}\boldsymbol{\mu}$ and $\boldsymbol{\zeta} = -\frac{1}{2}(d\log(2\pi) - \log|\boldsymbol{\Lambda}| + \boldsymbol{\eta}^T\boldsymbol{\Lambda}^{-1}\boldsymbol{\eta})$ So the product of n Gaussian PDFs $i = 1...n$ is

$$\prod_{i=1}^{n} p_i(\boldsymbol{x}) = \exp\left[\boldsymbol{\zeta}_{i=1,\ldots,n} + \left(\sum_{i=1}^{n} \boldsymbol{\eta_i}\right)^T \boldsymbol{x} - \frac{1}{2}\, \boldsymbol{x}^T \left(\sum_{i=1}^{n} \boldsymbol{\Lambda}_i\right) \boldsymbol{x}\right]$$

where

$$\boldsymbol{\zeta}_{i=1,\ldots,n} = \sum_{i=1}^{n} \boldsymbol{\zeta}_i = -\frac{1}{2}(nd\log(2\pi) - \sum_{i=1}^{n}\log|\boldsymbol{\Lambda}_i| + \sum_{i=1}^{n} \boldsymbol{\eta}_i^T \boldsymbol{\Lambda}_i^{-1}\boldsymbol{\eta}_i)$$

So

$$\begin{aligned}\prod_{i=1}^{n} p_i(\boldsymbol{x}) &= \exp\left[\boldsymbol{\zeta}_{i=1,\ldots,n} + \boldsymbol{\zeta}_n - \boldsymbol{\zeta}_n + \left(\sum_{i=1}^{n} \boldsymbol{\eta_i}\right)^T \boldsymbol{x} - \frac{1}{2}\, \boldsymbol{x}^T \left(\sum_{i=1}^{n} \boldsymbol{\Lambda}_i\right) \boldsymbol{x}\right] \\ &= \exp\left(\boldsymbol{\zeta}_{i=1,\ldots,n} - \boldsymbol{\zeta}_n\right)\ exp\left[\boldsymbol{\zeta}_n + \boldsymbol{\eta}_n^T \boldsymbol{x} - \frac{1}{2}\, \boldsymbol{x}^T \boldsymbol{\Lambda}_n \boldsymbol{x}\right]\end{aligned} \tag{46}$$

where

$$\boldsymbol{\Lambda}_n = \sum_{i=1}^{n} \boldsymbol{\Lambda}_i, \quad \boldsymbol{\eta}_n = \sum_{i=1}^{n} \boldsymbol{\eta}_i$$

and

$$\boldsymbol{\zeta}_n = -\frac{1}{2}(d\log(2\pi) - \log|\boldsymbol{\Lambda}_n| + \boldsymbol{\eta}_n^T \boldsymbol{\Lambda}_n^{-1}\boldsymbol{\eta}_n) \tag{47}$$

Comparing Equations (45),(46), and (47) shows that the result is a scaled Gaussian PDF over $\boldsymbol{x}$ with a mean vector and covariance matrix given by

$$\mathbf{V}_n^{-1} = \sum_{i=1}^{n} \mathbf{V}_i^{-1} \quad \text{and} \quad \mathbf{V}_n^{-1}\mu_n = \sum_{i=1}^{n} \mathbf{V}_i^{-1}\mu_i$$

The scaling factor is again a Gaussian function.

References

[1] N. Keshava and J. F. Mustard, "Spectral unmixing," *IEEE Signal Processing Magazine*, pp. 44–57, Jan. 2002.

[2] N. Dobigeon, J.-Y. Tourneret, and C.-I Chang, "Semi-supervised linear spectral unmixing using a hierarchical Bayesian model for hyperspectral imagery," *IEEE Trans. Signal Processing*, vol. 56, no. 7, pp. 2684–2695, July 2008.

[3] D. C. Heinz and C. -I Chang, "Fully constrained least-squares linear spectral mixture analysis method for material quantification in hyperspectral imagery," *IEEE Trans. Geosci. and Remote Sensing*, vol. 29, no. 3, pp. 529–545, March 2001.

[4] S. J. Gershman and D. M. Blei, "A tutorial on Bayesian nonparametric models," *Journal of mathematical psychology*, vol. 56, no. 1, pp. 1–12, 2012.

[5] D. Görür, "Nonparametric Bayesian discrete latent variable models for unsupervised learning," Ph.D. dissertation, Von der Fakultät IV - Elektrotechnik und Informatik der Technischen Universität Berlin, 2007.

[6] A. Zare, "Hyperspectral endmember detection and band selection using Bayesian methods," Ph.D. dissertation, University of Florida, Florida, 2008.

[7] M. I. Jordan, "Dirichlet processes, chinese restaurant processes and all that," *Tutorial at Neural Information Processing Systems*, December 2005.

[8] T. S. Ferguson, "Prior distributions on spaces of probability measures," *The Annals of Statistics*, vol. 2, no. 4, pp. 615–629, July 1974.

[9] ——, "A bayesian analysis of some nonparametric problems," *The Annals of Statistics*, vol. 1, no. 2, pp. 209–230, 1973.

[10] C. E. Antoniak, "Mixtures of Dirichlet processes with applications to Bayesian nonparametric problems," *The Annals of Statistics*, vol. 2, no. 6, pp. 1152–1174, 1974.

[11] R. M. Neal, "Bayesian mixture modeling by Monte Carlo simulation," Department of Computer Science, University of Toronto, Tech. Rep., CRG-TR-91-2, June 1991.

[12] C. E. Rasmussen, "The infinite gaussian mixture model," *in Advances in Neural Information Processing Sytems 12*, pp. 554–560, 2000.

[13] A. Ranganathan, "The Dirichlet process mixture (DPM) model," 2006. [Online]. Available : http ://www.cc.gatech.edu/ $\sim$ ananth/docs/dirichlet.pdf

[14] S. Jain and R. M. Neal, "A split-merge Markov Chain Monte Carlo procedure for the Dirichlet process mixture model," Department of Statistics, University of Toronto, Tech. Rep., No. 2003, July 2000.

[15] R. M. Neal, "Markov chain sampling methods for Dirichlet process mixture models," Department of Statistics, University of Toronto, Tech. Rep., No.9815, September 1998.

[16] D. Blackwell and J. MacQueen, "Ferguson distributions via Pólya urn schemes." *The Annals of Statistics*, vol. 1, no. 2, pp. 353–355, March 1973.

[17] T. L. Griffiths and Z. Ghahramani, "Infinite latent feature models and the indian buffet process," Gatsby Computational Neuroscience Unit, University College London, Tech. Rep., GCNU TR 2005−001, May 2005.

[18] F. Doshi-Velez, "The indian buffet process : Scalable inference and extensions," Master's thesis, University of Cambridge, Cambridge, 2009.

[19] W. Fan, B. Hu, J. Miller, and M. Li, "Comparative study between a new nonlinear model and common linear model for analysing laboratory simulated-forest hyperspectral data," *International Journal of Remote Sensing*, vol. 30, no. 11, pp. 2951–2962, June 2009.

[20] J. M. P. Nascimento and J. M. Bioucas-Dias, "Nonlinear mixture model for hyperspectral unmixing," L. Bruzzone, C. Notarnicola, and F. Posa, Eds., vol. 7477, no. 1. SPIE, 2009, p. 74770I.

[21] X. Yu, "Gibbs sampling methods for dirichlet process mixture model : Technical details," 2009. [Online]. Available : http ://yuxiaodong.files.wordpress.com/2009/09/technical-details-in-gibbs-sampling-for-dp-mixture-model.pdf

[22] A. Zare and P. D. Gader, "Sparsity promoting iterated constrained endmember detection for hyperspectral imagery," *IEEE Geoscience and Remote Sensing Letters*, vol. 4, no. 3, pp. 446–450, july 2007.

[23] O. Eches, N. Dobigeon, and J.-Y. Tourneret, "Estimating the number of endmembers in hyperspectral images using the normal compositional model and a hierarchicalbayesian algorithm," *IEEE Journal of Selected Topics in Signal Processing*, vol. 4, no. 3, pp. 582–591, 2010.

[24] M. Berman, H. Kiiveri, R. Lagerstrom, A. Ernst, R. Dunne, and J. F. Huntington, "ICE : A statistical approach to identifying endmembers in hyperspectral images," *IEEE Trans. Geosci. and Remote Sensing*, vol. 42, no. 10, pp. 2085–2095, Oct. 2004.

[25] E. Punskaya, C. Andrieu, A. Doucet, and W. J. Fitzgerald, "Bayesian curve fitting using mcmc with applications to signal segmentation," *IEEE Trans. Signal Processing*, vol. 50, no. 3, pp. 747–758, 2002.

[26] H. Jeffreys, *Theory of Probability*, 3rd ed. London : Oxford University Press, 1961.

[27] K. E. Themelis, A. A. Rontogiannis, and K. D. Koutroumbas, "A novel hierarchical bayesian approach for sparse semisupervised hyperspectral unmixing," *IEEE Trans. Signal Processing*, vol. 60, no. 2, pp. 585–599, Feb. 2012.

[28] N. Dobigeon, S. Moussaoui, M. Coulon, J.-Y. Tourneret, and A. O. Hero, "Joint bayesian endmember extraction and linear unmixing for hyperspectral imagery," *IEEE Trans. Signal Processing*, vol. 57, no. 11, pp. 4355–4368, Nov. 2009.

[29] A. Halimi, Y. Altmann, N. Dobigeon, and J.-Y. Tourneret, "Nonlinear unmixing of hyperspectral images using a generalized bilinear model," *IEEE Trans. Geosci. and Remote Sensing*, vol. 49, no. 11, pp. 4153–4162, 2011.

[30] J. M. Nascimento and J. M. Bioucas-Dias, "Vertex component analysis : a fast algorithm to unmix hyperspectral data," *IEEE Trans. Geosci. and Remote Sensing*, vol. 43, no. 4, pp. 898–910, April 2005.