INSTITUTE FOR SPACE APPLICATIONS AND REMOTE SENSING,
NATIONAL OBSERVATORY OF ATHENS

HYMARS – ALGORITHMS DEVELOPMENT AND APPLICATION OF NOVEL
HYPER-SPECTRAL IMAGING (HSI) TECHNIQUES TO OMEGA DATA
OBTAINED FROM THE MARS EXPRESS MISSION OF THE EUROPEAN
SPACE AGENCY (ESA)

Technical report

“Algorithms development and software implementation for supervised hyperspectral signal unmixing”

THEMELIS KONSTANTINOS

July, 2009
Table of Contents

I. ABSTRACT..........................................................................................................................3
II. INTRODUCTION..................................................................................................................3
   A. Hyperspectral Image Data Description .................................................................3
   B. Hyperspectral Unmixing .....................................................................................3
   C. The linear mixing model ..................................................................................4
   D. Physical constraints .........................................................................................4
III. LITERATURE REVIEW.......................................................................................................5
   A. Least Squares ........................................................................................................5
      i. Unconstrained ......................................................................................................5
      ii. Full-additivity ..................................................................................................5
      iii. Nonnegativity ................................................................................................6
   B. Bayesian Approach .............................................................................................6
      i. EM in linear regression ..................................................................................6
      ii. Gibbs Sampler .............................................................................................8
      iii. Rao-Blackwellization of the Gibbs Sampler ............................................10
IV. PROPOSED METHOD ......................................................................................................11
   A. MAP’s estimator ..................................................................................................11
      i. Imposing the constraints ...............................................................................14
V. RESULTS ............................................................................................................................15
   A. Constrained estimators ....................................................................................15
   B. Bayesian methods ............................................................................................17
   C. Proposed method ............................................................................................18
VI. MATLAB SOURCE CODES ..........................................................................................22
VII. REFERENCES ..................................................................................................................32
I. Abstract

This report studies the advances that have taken place in the area of supervised hyperspectral signal unmixing, [1]-[2], and describes a novel approach for spectral unmixing of hyperspectral images [3]. To this end, we present, analyze and compare the performance of algorithms used for supervised unmixing. The term “supervised” refers to the available prior knowledge of the mixing matrix of the linear mixing model. The availability of such knowledge is common place in hyperspectral data processing, partly because endmembers can be extracted from the hyperspectral data cube, and partly because spectral libraries have been established as a means to represent endmembers’ spectral signatures. The proposed method estimates the mixture coefficients (referred to as abundances) using a Bayesian Minimum Mean Square Error (MMSE) estimator. A suitable Gaussian prior is assigned to the abundances and the parameters of the prior distribution are properly adjusted in order to satisfy positivity and nonnegativity constraints.

II. Introduction

A. Hyperspectral Image Data Description

The evolution of passive remote sensing has witnessed the collection of measurements with significantly greater spectral breadth and resolution [1]-[2]. It has been motivated by a desire to extract increasingly detailed information about the material properties of pixels in a scene for both civilian and military applications. While multispectral sensing has largely succeeded at classifying whole pixels, further analysis of the constituent substances that comprise a pixel is limited by a relatively low number of spectral measurements. The recognition that pixels of interest are frequently a combination of numerous disparate components has introduced a need to quantitatively decompose, or “unmix,” these mixtures. In hyperspectral imagery, mixed pixels are a mixture of more than one distinct substance, and they exist for one of two reasons. First, if the spatial resolution of a sensor is low enough that disparate materials can jointly occupy a single pixel, the resulting spectral measurement will be some composite of the individual spectra. This is the case for remote sensing platforms flying at a high altitude or performing wide-area surveillance, where low spatial resolution is common. Second, mixed pixels can result when distinct materials are combined into a homogeneous mixture. This circumstance can occur independently of the spatial resolution of the sensor.

B. Hyperspectral Unmixing

Spectral unmixing [1] 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. Broadly speaking, unmixing is a special case of the generalized inverse problem that estimates parameters describing an object using an observation(s) of a signal that has interacted with the object before arriving at the sensor. In the case of hyperspectral sensing in the reflective regime, and ignoring atmospheric effects, the
incident signal is electromagnetic radiation that originates from the sun and is measured by a sensor after it has been reflected upwards by natural and man-made materials on the surface of the Earth.

C. 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. A complete model of the mixing process, however, is more complicated than a simple description of how surface mixtures interact. Mixing models can also incorporate the effects of the three-dimensional topology of objects in a scene, such as the height of trees, the size and density of their canopies, and the sensor observation angle. The basic premise of mixture modelling 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. In Fig. 1(a), the reflecting surface is portrayed as a checkerboard mixture, and any given package of incident radiation only interacts with one component (i.e., no multiple scattering between components). If the total surface area is considered to be divided proportionally according to the fractional abundances of the endmembers, then the reflected radiation will convey the characteristics of the associated media with the same proportions. In this sense, a linear relationship exists between the fractional abundance of the substances comprising the area being imaged and the spectra in the reflected radiation.

The model most frequently used is called the linear mixing model (LMM) [1] and it is expressed as

\[ r = Cx + v = \sum_{i=1}^{M} x_i c_i + v, \]  

where \( M \) is the number of endmembers, \( r \) is the Lx1 received pixel spectrum vector, \( C \) is the LxM matrix whose columns are the Lx1 endmembers, \( c_i \), \( i = 1, \ldots, M \), \( x \) is the Mx1 fractional abundance vector, whose entries are \( x_i \), \( i = 1, \ldots, M \), and \( v \) is the Lx1 additive observation noise vector.

D. Physical constraints

The LMM is subject to two constraints on the entries of the abundance vector \( x \), [1]. To be physically meaningful, the nonnegativity condition requires all abundances to be nonnegative such that \( x_i \geq 0 \), \( i = 1, \ldots, M \). Second, as a way of accounting for the entire composition of a mixed pixel, the full additivity condition requires

\[ \sum_{i=1}^{M} x_i = 1. \]  

It is easy to see that the physical constraints define the standard \( M-1 \) dimensional simplex in the \( M \) dimensional space. The geographical representation of the simplex of constraints in three-dimensional space is displayed in the next figure as the shaded triangle.
III. Literature Review

A. Least Squares

The class of inversion algorithms based on minimizing squared error start from the simplest form of least squares inversion and increase in complexity as further assumptions and parametric structure are imposed on the problem. Variations of the least squares concept have been adopted to reflect the unique circumstances associated with hyperspectral data.

i. Unconstrained

Starting with the LMM, and the assumption of no additive noise, the unconstrained least squares solution for $x$ is $x^U = (C^T C)^{-1} C^T r$ [4]. This unconstrained estimate for $x$ minimizes $\|r - Cx\|^2$. This form does not require the estimation of the additive noise and exists when there are more bands than endmembers (a reasonable assumption for hyperspectral sensing), and when $C$ has full column rank.

ii. Full-additivity

The unconstrained solution can be refined by constraining $x^{U}$ to fulfill the full additivity condition. This has the effect of restricting the least squares solution to lie on the hyperplane given by $\sum_{i=1}^{M} x_i = 1$. The solution for a general least squares estimate having linear constraints given by $Bx = d$, is obtained using Lagrange multipliers [5] and is given by
\[ x^F = x^U - (C^T C)^{-1} B^T \left[ B (C^T C)^{-1} B^T \right]^{-1} (B x^U - d) \]  

(III.1)

The full additivity constraint is enforced when \( B \) is a 1xM row vector having all ones and \( d = 1 \). Closer examination of \( x^F \) reveals that the solution enforcing full additivity consists of the unconstrained least squares solution, \( x^U \), with an additive correction term that depends on the matrix of endmembers, \( C \), and the error incurred by \( x^U \) in satisfying the full additivity constraint.

### iii. Nonnegativity

The complementary constraint, nonnegativity, is not as easy to address in closed form as full additivity. Minimizing \( ||r - Cx||^2 \), while maintaining \( x_i \geq 0, i = 1, \ldots, M \), falls in the domain of quadratic programming with linear inequalities as constraints. Exploiting the convexity of the constraints, interior point methods can be used to solve the minimization problem in polynomial time [6]. However, the recursive nature of interior point methods unveils the problem of computational complexity, bearing in mind that a different minimization problem has to be solved for every pixel of the hyperspectral image.

### B. Bayesian Approach

The Bayesian approach has proved to be useful in addressing the problem of parameter estimation. By design, Bayesian methods are often recommended as the proper way to make formal use of subjective information. An important advantage of Bayesian methods is that they provide the means to incorporate prior knowledge in data analysis. Bayesian analysis revolves around the posterior probability, which summarizes the degree of one’s certainty concerning a given situation. Bayes’s law states that the posterior probability is proportional to the product of the likelihood and the prior probability. The likelihood encompasses the information contained in the new data and the prior expresses the degree of certainty concerning the situation before the data are taken.

In the following sections, two different Bayesian models are described. The first model adopts a prior Gaussian distribution for the abundances, and the powerful Expectation Maximization (EM) algorithm [7] for the linear regression problem is used. The complexity of the second model is higher, since it adopts a Dirichlet prior distribution for the vector of abundances, and a Minimum Mean Square Error (MMSE) estimator is derived using a Gibbs Sampler [8].

#### i. EM in linear regression

We begin our discussion of the Bayesian treatment of linear regression by introducing a prior probability distribution over the model parameters \( x \). In our approach, we will assume that the noise precision parameter \( \beta = 1/\sigma^2 \) is a known constant. First note that the likelihood function

\[ p(r|x, C, \beta) = N(r|Cx, \beta^{-1}) = (2\pi)^{-L/2} \beta^{-L/2} \exp(-\beta/2 \ ||r - Cx||^2) \]  

(III.2)

is the exponential of a quadratic function of \( x \). The corresponding conjugate prior is therefore given by a Gaussian distribution of the form
$$p(x | \beta) = \prod_{m=1}^{M} N(x_m | m_0, \alpha^{-1}), \quad (\text{III.3})$$

having mean $m_0 = 1/M$ and covariance $\Sigma_0 = 1/\text{diag}(\alpha^{-1})$, where the mean value $m_0 = 1/M$ is selected due to the symmetry of the physical constraints of the problem. This prior distribution is stationary, meaning that the distribution of all the abundances is identical. It is clear that the abundances are hidden random variables and the model parameters are the parameter $\alpha$ of the prior for $x$ and the inverse variance $\beta$ of the additive noise.

Bayesian inference proceeds by computing the posterior distribution of the hidden variables:

$$p(x | r; \alpha, \beta) = \frac{p(r | x; \beta)p(x; \alpha)}{p(r; \alpha, \beta)} \quad (\text{III.4})$$

Notice that the marginal likelihood $p(r; \alpha, \beta)$ that appears on the denominator can be computed analytically:

$$p(r; \alpha, \beta) = \int p(r | w; \beta)p(w; \alpha)dw = N(r | 0, \beta^{-1}I + \alpha^{-1}C C^T) \quad (\text{III.5})$$

Then, the posterior of the hidden variable is

$$p(x | r; \alpha, \beta) = N(x | \mu, \Sigma), \quad (\text{III.6})$$

where

$$\mu = \beta \Sigma C^Tr + \alpha \Sigma m_0 \quad \text{and} \quad \Sigma = (\beta C^T C + \alpha I)^{-1}$$

The parameters of the model can be estimated by maximizing the logarithm of the marginal likelihood $p(r; \alpha, \beta)$

$$(\alpha_{ML}, \beta_{ML}) = \arg \min_{\alpha, \beta} \left\{ \log \beta^{-1}I + \alpha^{-1}C C^T + r^T(\beta^{-1}I + \alpha^{-1}C C^T)^{-1}r \right\}, \quad (\text{III.7})$$

Direct optimization of the marginal likelihood presents several computational difficulties, since its derivatives with respect to the parameters $(\alpha, \beta)$ are difficult to compute. Furthermore, the problem requires a constrained optimization algorithm since the estimates of $(\alpha, \beta)$ have to be positive since they represent inverse variances. Instead, the EM algorithm provides an efficient framework to simultaneously obtain estimates for $(\alpha, \beta)$ and infer values for $x$. Notice, that although the EM algorithm does not involve computations with the marginal likelihood, the algorithm converges to a local maximum of it. After initializing the parameters to some values $(\alpha^{(0)}, \beta^{(0)})$, the algorithm proceeds by iteratively performing the following steps:

- **E-step**
  Compute the expected value of the logarithm of the complete likelihood:

  $$Q^{(n)}(r, x; \alpha, \beta) = \frac{L}{2} \ln \beta + \frac{\beta}{2} (|| r - C \mu^{(n)} ||^2 + \text{tr}[C^T \Sigma^{(n)} C]) + \frac{M}{2} \ln \alpha + \frac{\alpha}{2} (|| \mu^{(n)} ||^2 + \text{tr}[\Sigma^{(n)}]) + \text{const} \quad (\text{III.8})$$

  where $\mu^{(n)}$ and $\Sigma^{(n)}$ are computed using the current estimates of the parameters $\alpha^{(n)}$ and $\beta^{(n)}$. 

\[ \mu^{(t)} = \beta^{(t)} \Sigma^{(t)} r \]  
\[ \Sigma^{(t)} = (\beta^{(t)} C^T C + \alpha^{(t)} I)^{-1} \] (III.9)  

- M-step

Maximize \( Q^{(t)}(r, x; \alpha, \beta) \) with respect to the parameters \( \alpha \) and \( \beta \):

\[ (\alpha^{(t+1)}, \beta^{(t+1)}) = \arg \max_{\alpha, \beta} Q^{(t)}(r, x; \alpha, \beta) \] (III.11)

The derivatives of \( Q^{(t)}(r, x; \alpha, \beta) \) with respect to the parameters are:

\[ \frac{\partial Q^{(t)}(r, x; \alpha, \beta)}{\partial \alpha} = \frac{M}{2 \alpha} \left( \frac{1}{2} \left( \mu^{(t)} \right)^2 + \text{tr} \left[ \Sigma^{(t)} \right] \right) \] (III.12)

\[ \frac{\partial Q^{(t)}(r, x; \alpha, \beta)}{\partial \beta} = \frac{L}{2 \beta} \left( \frac{1}{2} \left( \left\| r - C \mu^{(t)} \right\|^2 + \text{tr} \left[ C^T \Sigma^{(t)} C \right] \right) \right) \] (III.13)

Setting these to zero, we obtain the following formulas:

\[ \alpha^{(t+1)} = \frac{M}{\left\| \mu^{(t)} \right\|^2 + \text{tr} \left[ \Sigma^{(t)} \right]} \] (III.14)

\[ \beta^{(t+1)} = \frac{L}{\left\| r - C \mu^{(t)} \right\|^2 + \text{tr} \left[ C^T \Sigma^{(t)} C \right]} \] (III.15)

Notice that the maximization step can be analytically performed in contrast to direct maximization of the marginal likelihood \( p(r; \alpha, \beta) \), which would require numerical optimization. Furthermore, equations (III.14) and (III.15), guarantee that positive estimations for the parameters \( \alpha \) and \( \beta \) are produced, which is a requirement since these represent inverse variance parameters. However, the parameters should be initialized with care, since depending on the initialization a different local maximum may be attained. Inference on \( x \) is obtained directly since the sufficient statistics of the posterior \( p(x|r; \alpha, \beta) \) are computed in the E-step. The mean of this posterior can be used as a Bayesian linear minimum mean square error (LMMSE) inference for \( x \).

**ii. Gibbs Sampler**

This section introduces a hierarchical Bayesian model [8] to estimate the abundance vector \( x \), under the physical constraints of the problem. This model is based in the likelihood of the observations and on prior distributions for the unknown parameters. As before, the likelihood of the observations \( r \) can be expressed as:

\[ p(r | x, \sigma^2) = \left( \frac{1}{2\pi \sigma^2} \right)^{\frac{L}{2}} \exp \left[ - \frac{\left\| r - C x \right\|^2}{2\sigma^2} \right] \] (III.16)
The abundance vector can be written as \( \mathbf{x} = [\mathbf{x}_{M-1}^T, \mathbf{x}_M]^T \) with \( \mathbf{x}_{M-1} = [x_1, x_2, \ldots, x_{M-1}] \) and \( x_M = 1 - \sum_{m=1}^{M-1} x_m \). The LMM constraints impose that \( \mathbf{x} \) belongs to the simplex \( \mathcal{S} \)

\[
\mathcal{S} = \left\{ \mathbf{x} \mid x_m \geq 0, \forall m = 1, \ldots, M-1, \sum_{m=1}^{M-1} x_m \leq 1 \right\}.
\]

A uniform distribution on \( \mathcal{S} \) is chosen for \( \mathbf{x} \) in order to reflect the absence of prior knowledge regarding this unknown parameter vector. Note that choosing this prior distribution for \( \mathbf{x} \) is equivalent to choosing a prior Dirichlet \( D_M(1, \ldots, 1) \) for \( \mathbf{x} \).

A conjugate inverse gamma distribution with parameters \( \nu/2 \) and \( \gamma/2 \) is chosen as prior distribution for \( \sigma^2 \).

\[\sigma^2 \mid \nu, \gamma \sim IG(\nu/2, \gamma/2)\]

To simplify calculations, the hyperparameter \( \nu \) is fixed to \( \nu=2 \), whereas \( \gamma \) remains an adjustable hyperparameter. The hyperparameter associated to the parameter priors defined above is \( \gamma \). Of course, the quality of the unmixing procedure depends on the value of this hyperparameter. The hierarchical Bayesian approach uses a noninformative Jeffrey’s prior for the hyperparameter \( \gamma \)

\[p(\gamma) \propto \frac{1}{\gamma^2}, \quad (\text{III.19})\]

where \( 1_{\gamma^2}(\cdot) \) is the indicator function defined on \( \gamma^2 \). The posterior distribution of the unknown parameter vector \( \mathbf{\theta} = \{\mathbf{x}, \sigma^2\} \) can be computed from the following hierarchical structure:

\[p(\mathbf{\theta} \mid \mathbf{r}) \propto \int p(\mathbf{r} \mid \mathbf{\theta}) p(\mathbf{\theta} \mid \gamma) p(\gamma) \, d\gamma\]

By assuming the prior independence between \( \sigma^2 \) and \( \mathbf{x} \), i.e. \( p(\mathbf{x} \mid \gamma) = p(\mathbf{x}) p(\sigma^2 \mid \gamma) \), the hyperparameter \( \gamma \) can be integrated out from the joint distribution \( p(\mathbf{x}, \gamma \mid \mathbf{r}) \), yielding

\[p(\mathbf{x}, \sigma^2 \mid \mathbf{r}) \propto \frac{1}{\sigma^{L+2}} \exp \left[ - \frac{\| \mathbf{r} - C \mathbf{x} \|^2}{2\sigma^2} \right] 1_{\mathcal{S}}(\mathbf{x}),\]

where \( 1_{\mathcal{S}}(\cdot) \) is the indicator function defined on the simplex \( \mathcal{S} \). The objective of the Gibbs sampler described next is to allow one to generate samples distributed according to the joint distribution \( p(\mathbf{x}, \sigma^2 \mid \mathbf{r}) \). The Gibbs sampler consists of two steps:

- Generation of samples according to \( p(\mathbf{x} \mid \sigma^2, \mathbf{r}) \)
  By denoting \( C_{M-1} = [c_1, \ldots, c_{M-1}] \), straightforward computations yield

\[p(\mathbf{x} \mid \sigma^2, \mathbf{r}) \propto \exp \left[ - \frac{\| \mathbf{r} - C \mathbf{x} \|^2}{2\sigma^2} \right] 1_{\mathcal{S}}(\mathbf{x})\]

where

\[\Lambda = \left[ \frac{1}{\sigma^2}(C_{M-1} - c_M u^T)^T (C_{M-1} - c_M u^T)^{-1}\right]

and

\[\text{(III.23)}\]
\[ \mu = \Lambda \left[ \frac{1}{\sigma^2} \left( \mathbf{C}_{M \times M} \mathbf{u}^T \right) \left( \mathbf{r} - \mathbf{c}_M \right) \right], \]  
(III.24)

with \( \mathbf{u} = [1, \ldots, 1]^T \). As a consequence, \( \mathbf{x} | \sigma^2, \mathbf{r} \) is distributed according to a truncated Gaussian distribution

\[ \mathbf{x} | \sigma^2, \mathbf{r} \sim N_S(\mu, \Lambda). \]  
(III.25)

The generation of samples according to a truncated Gaussian distribution can be achieved using a standard accept-reject procedure, when the number of endmembers is relatively small.

- Generation of samples according to \( p(\sigma^2 | \mathbf{x}, \mathbf{r}) \)

Looking carefully at the joint distribution \( p(\mathbf{x}, \sigma^2 | \mathbf{r}) \), the conditional distribution of \( \sigma^2 | \mathbf{x}, \mathbf{r} \) is clearly the following inverse gamma distribution:

\[ \sigma^2 | \mathbf{x}, \mathbf{r} \sim IG \left[ \frac{L}{2}, \frac{\| \mathbf{r} - \mathbf{C} \mathbf{x} \|^2}{2} \right] \]  
(III.26)

The Gibbs sampler allows one to draw sample \( (\mathbf{x}^{(n)}, \sigma^2) \) asymptotically distributed according to \( p(\mathbf{x}, \sigma^2 | \mathbf{r}) \). The abundance vector can then easily be estimated by the empirical average following the minimum mean square error (MMSE) principle:

\[ \hat{\mathbf{x}}_{\text{MMSE}} = \frac{1}{N_b} \sum_{t=1}^{N_b} \mathbf{x}^{(n_b+t)}, \]  
(III.27)

where \( N_{\text{bi}} \) and \( N_b \) are the numbers of burn-in and computation iterations, respectively.

### iii. Rao-Blackwellization of the Gibbs Sampler

In statistics, the Rao–Blackwell theorem [9] characterizes the transformation of an arbitrarily crude estimator into an estimator that is optimal by the mean-squared-error criterion or any of a variety of similar criteria. The Rao-Blackwell theorem states that if \( g(\mathbf{X}) \) is any kind of estimator of a parameter \( \theta \), then the conditional expectation of \( g(\mathbf{X}) \) given \( T(\mathbf{X}) \), where \( T \) is a sufficient statistic, is typically a better estimator of \( \theta \), and is never worse. The process of transforming an estimator using the Rao-Blackwell theorem is sometimes called Rao-Blackwellization.

As mentioned earlier, the Gibbs sampler consists of two steps, (a) generation of samples according to \( p(\mathbf{x} | \sigma^2, \mathbf{r}) \), and (b) generation of samples according to \( p(\sigma^2 | \mathbf{x}, \mathbf{r}) \). In this version of the algorithm, instead of generating a random sample of these conditional distributions, the mean of the distributions is selected.

Analytically, the Rao-Blackwell Gibbs sampler consists of two modifications in the steps of the conventional Gibbs sampler:

- Compute the mean of \( p(\mathbf{x} | \sigma^2, \mathbf{r}) \)

As before, \( p(\mathbf{x} | \sigma^2, \mathbf{r}) \) is the truncated Gaussian distribution

\[ p(\mathbf{x} | \sigma^2, \mathbf{r}) \propto \exp \left[ -\frac{(\mathbf{x} - \mu)^T \Lambda^{-1} (\mathbf{x} - \mu)}{2\sigma^2} \right] 1_s(\mathbf{x}), \]  
(III.28)

where
\[ \Lambda = \left[ \frac{1}{\sigma^2} (C_{M-1} c M^u) \right]^{-1} \left( C_{M-1} c M^u \right) \]  \hspace{1cm} (III.29)

and \[ \mu = \Lambda \left[ \frac{1}{\sigma^2} (C_{M-1} c M^u) \right] T (r - c M) \],  \hspace{1cm} (III.30)

with \( u = [1, \ldots, 1]^T \in \mathbb{R}^{M-1} \).

- Compute the mean of \( p(\sigma^2 | x, r) \)

As before, \( p(\sigma^2 | x, r) \) is the conditional distribution of \( \sigma^2 | x, r \) and is an inverse gamma distribution:

\[ \sigma^2 | x, r \sim IG \left( \frac{L}{2}, \frac{\| r - C x \|^2}{2} \right) \]  \hspace{1cm} (III.31)

The conventional Gibbs Sampler and its Rao-Blackwellized version are compared in terms of MSE estimation performance in the results section.

**IV. Proposed method**

In this section, a novel approach is presented for spectral unmixing of hyperspectral images. By assuming knowledge of the number and spectral signatures of the materials present in an image, efficient estimation for their corresponding fractions in the pixels of the image is developed based on a recently proposed maximum a posteriori probability (MAP) method. By exploiting the constraints naturally imposed to the problem, closed form expressions are derived for the statistical parameters required by the MAP estimator.

**A. MAPs estimator**

A recently proposed soft constrained maximum a posteriori probability (MAP-s) method [10] is adopted and properly adjusted for abundance estimation in hyperspectral images. The MAP-s estimator belongs to the class of superefficient estimators, i.e. it always provides better performance than the LS estimator, in terms of the matrix mean square error (MSE). Let \( \Sigma_{LS} \equiv (C^T \Sigma^{-1} C)^{-1} \) denote the covariance matrix of the unconstrained LS estimator. An estimator \( \hat{x} \) is said superefficient or LS-dominating if its MSE does not exceed the MSE of the LS estimator, i.e.

\[ E[(x - \hat{x})^T (x - \hat{x})] \leq \Sigma_{LS} \]  \hspace{1cm} (IV.1)

To provide MSE improvement, superefficient estimators exploit some a priori knowledge about the parameters to be estimated. In the problem under consideration, the a priori knowledge is the set of constraints which are naturally imposed on \( x \). Under the assumption that \( x \) is a random vector with prior Gaussian distribution, the MAP-s estimate of \( x \) is:

\[ \hat{x} = \left( C^T \Sigma^{-1} C + \Sigma^{-1} \right)^{-1} \left( C^T \Sigma^{-1} r + \Sigma^{-1} \bar{x} \right), \]  \hspace{1cm} (IV.2)

where \( \bar{x} \in \mathbb{R}^M \) and \( \Sigma \in \mathbb{R}^{M \times M} \) are the mean and covariance matrix of \( x \), respectively. The main idea here, is to select the parameters \( \bar{x} \) and \( \Sigma \), based on the knowledge of
the polytope of constraints $\mathbf{S}$, so as to guarantee the LS-domination property (IV.1), $\forall \mathbf{x} \in \mathbf{S}$. After some calculations reported in [10], the LS-domination condition (IV.1) can be rewritten in terms of $\mathbf{\bar{S}}$ and $\mathbf{\bar{x}}$, as follows:

$$\mathbf{\bar{S}} \geq \frac{1}{2} \left[ (\mathbf{x} - \mathbf{\bar{x}})(\mathbf{x} - \mathbf{\bar{x}})^T - \Sigma_{\text{LS}} \right]$$  \hspace{1cm} (IV.3)

In order to compute the most appropriate parameters $\mathbf{\bar{x}}$ and $\mathbf{\bar{S}}$, according to the constraint set $\mathbf{S}$, Lemma 1 of [10] proved for positive definite matrices needs to be extended for positive semidefinite matrices. This is so, because in the problem under consideration the polytope of constraints $\mathbf{S}$ lies on a hyperplane of dimension $(M-1)$ in the $M$-dimensional space.

**Lemma 1**: Let $\mathbf{P} = \mathbf{P}^T \geq 0$ and $\mathbf{x} \in \mathbb{R}(\mathbf{P})$, where $\mathbb{R}(\cdot)$ denotes column space. Then if

$$\mathbf{P} \geq \mathbf{xx}^T \Leftrightarrow \mathbf{x}^T \mathbf{P}^T \mathbf{x} \leq 1,$$  \hspace{1cm} (IV.4)

where $(\cdot)^{+}$ denotes the Moore-Penrose pseudoinverse.

Proof: It follows directly from the properties of the generalized Schur complement. According to the Albert nonnegative conditions [11], for matrices $\mathbf{P}$, $\mathbf{C}$ and $\mathbf{B}$ with appropriate dimensions, the following two statements are equivalent:

i) $\mathbf{P} \geq 0$ and $\mathbf{C} - \mathbf{B}^T \mathbf{P}^T \mathbf{B} \geq 0$ and $\mathbb{R}(\mathbf{B}) \subset \mathbb{R}(\mathbf{P})$.

ii) $\mathbf{C} \geq 0$ and $\mathbf{P} - \mathbf{B} \mathbf{C}^T \mathbf{B}^T \geq 0$ and $\mathbb{R}(\mathbf{B}^T) \subset \mathbb{R}(\mathbf{C})$.

Therefore, if $\mathbf{P} \geq 0$, $\mathbf{C} \geq 0$, and $\mathbb{R}(\mathbf{B}) \subset \mathbb{R}(\mathbf{P})$, and $\mathbb{R}(\mathbf{B}^T) \subset \mathbb{R}(\mathbf{C})$, then $\mathbf{P} \geq \mathbf{B} \mathbf{C}^T \mathbf{B}^T \Leftrightarrow \mathbf{C} \geq \mathbf{B}^T \mathbf{P}^T \mathbf{B}$. Setting $\mathbf{B} = \mathbf{x}$ and $\mathbf{C} = 1$ completes the proof.

Let $\varepsilon(\mathbf{c}, \mathbf{P}) = \left\{ \mathbf{x} : (\mathbf{x} - \mathbf{c})^T \mathbf{P}^T (\mathbf{x} - \mathbf{c}) \leq 1 \right\}$ denote an ellipsoid of centre $\mathbf{c}$ and $\mathbf{P}$ is a symmetric positive semidefinite matrix. Lemma 1 suggests that matrix $(\mathbf{P} - \mathbf{xx}^T)$ is positive semidefinite if and only if every point $\mathbf{x}$ belongs to the ellipsoid $\varepsilon(0, \mathbf{P})$. By setting

$$\mathbf{P} = 2 \mathbf{\bar{S}} + \Sigma_{\text{LS}},$$  \hspace{1cm} (IV.5)

Lemma 1 provides a geometrical interpretation of Eq.(IV.3): an MSE improvement over the LS estimator is achieved if and only if the ellipsoid $\varepsilon(\mathbf{x}, \mathbf{P})$ contains the polytope of constraints $\mathbf{S}$. Capitalizing on this result, it can be shown, as in [10], that $\mathbf{\bar{S}}$ and $\mathbf{\bar{x}}$ must satisfy the following set of linear matrix inequalities (LMI):

$$\begin{bmatrix} 2 \mathbf{\bar{S}} + \Sigma_{\text{LS}} & \mathbf{e}_i - \mathbf{\bar{x}} \\ (\mathbf{e}_i - \mathbf{\bar{x}})^T & 1 \end{bmatrix} \geq 0 \hspace{0.5cm} i = 1, 2, ..., M,$$  \hspace{1cm} (IV.6)

$$\mathbf{\bar{S}} \succeq 0,$$

where $\mathbf{e}_i$, $i = 1, 2, ..., M$ are the vertices of the $(M-1)$-standard simplex, which coincide with the columns of the identity matrix. The MAP-s estimator can then be summarized as follows: given $\mathbf{S}$ and $\Sigma_{\text{LS}}$, select the parameters $\mathbf{\bar{x}}$ and $\mathbf{\bar{S}}$ so that:

$$\min_{\mathbf{\bar{S}}, \mathbf{\bar{x}}} \det(\mathbf{\bar{S}}) \text{ subject to (6)},$$  \hspace{1cm} (IV.7)

where $\det(\cdot)$ denotes the determinant of a matrix.
In geometrical terms, minimization of the determinant of \( \Sigma \) corresponds to finding the minimum volume ellipsoid containing \( S \). The above minimization criterion does not guarantee the minimum achievable MSE, but gives rise to standard convex LMI problems that can be solved in polynomial time [12]. Although the minimization process is based on both the parameters \( \bar{x} \) and \( \bar{\Sigma} \), it has been shown that for symmetric constraints the optimal solution is obtained when \( \bar{x} \) is selected as the center of symmetry of the constraint set. In the problem of estimating the abundance vectors in a hyperspectral image, the polytope of constraints is explicitly defined as the standard \((M-1)\)-simplex \( S \). It can be shown that the minimum volume ellipsoid circumscribing \( S \) is the hypersphere \( e \left( \bar{x}, P \right) \), which is defined by

\[
P^t = \begin{bmatrix}
1 & -1 & \cdots & -1 \\
-1 & \frac{1}{M-1} & \cdots & \frac{1}{M-1} \\
\frac{1}{M-1} & \cdots & \frac{1}{M-1} & -1 \\
\frac{1}{M-1} & \cdots & \frac{1}{M-1} & -1 \\
\frac{1}{M-1} & \cdots & \frac{1}{M-1} & -1 \\
\end{bmatrix}
\] (IV.8)

while

\[
\bar{x} = \left[ \frac{1}{M}, \frac{1}{M}, \ldots, \frac{1}{M} \right]^T
\] (IV.9)

is selected as the center of the \((M-1)\)-dimensional simplex. For instance, for \( M = 3 \) the minimum volume ellipsoid \( e \left( \bar{x}, P \right) \) reduces to the disc shown in Fig. 1. Matrix \( P \), whose Moore-Penrose pseudoinverse is given by (IV.10), is a singular symmetric positive semidefinite matrix of rank \( M-1 \), expressed as follows:

\[
P = \begin{bmatrix}
\frac{(M-1)^2}{M^2} & \frac{M-1}{M^2} & \frac{M-1}{M^2} & \cdots & \frac{M-1}{M^2} \\
\frac{M-1}{M^2} & \frac{(M-1)^2}{M^2} & \cdots & \frac{M-1}{M^2} & \cdots \\
\frac{M-1}{M^2} & \cdots & \frac{(M-1)^2}{M^2} & \cdots & \frac{M-1}{M^2} \\
\frac{M-1}{M^2} & \cdots & \frac{M-1}{M^2} & \frac{(M-1)^2}{M^2} & \cdots \\
\frac{M-1}{M^2} & \cdots & \frac{M-1}{M^2} & \frac{(M-1)^2}{M^2} & \cdots \\
\end{bmatrix}
\] (IV.10)

It should be noted that since \( x \) satisfies the constraint \( \sum_{i=1}^{M} x_i = 1 \) and the elements of each row of \( P \) add to zero, \( \left( x - \bar{x} \right) \in \mathbb{R} \left( P \right) \), as required by the assumptions of Lemma 1. From (IV.10), (IV.5) and assuming knowledge of \( \Sigma_{LS} \), \( \Sigma \) can be written as follows:

\[
\bar{\Sigma} = \frac{1}{2} \left( P \cdot \Sigma_{LS} \right)
\] (IV.11)

By substituting \( \bar{x} \) and \( \bar{\Sigma} \) from (IV.9) and (IV.11) in (IV.2), an estimate of the abundance vector is directly obtained. These quantities need to be computed only
once and are then used for the estimation of the abundance vector of each pixel in the image. It should be noted that due to the form of matrix $P$, $\Sigma$ given by (IV.11) is near to singular. This may seriously affect the estimate in (IV.2), where the inverse of $\Sigma$ need to be computed. To tackle this problem, some kind of regularization can be applied, i.e., the inverse of $\Sigma$ is computed as $\left(\Sigma + \delta I\right)^{-1}$, where $\delta$ is a small positive constant and $I$ is the MxM identity matrix.

### i. Imposing the constraints

As described above, the MAP-s estimator assumes that the vector of abundances has a prior Gaussian distribution, i.e. $x \sim N(\bar{x}, \bar{\Sigma})$, where $\bar{x}$, $\bar{\Sigma}$ are given by (IV.9) and (IV.11) respectively. Due to this statistical assumption, for some pixels in the image, the corresponding abundance vectors may lie outside the polytope $S$, violating the constraints. However, as it will become clear in the next section, in order to assess the performance of the proposed method for hyperspectral images, the constraints must be somehow imposed to the estimator. Let $x_i$ be an estimated vector that lies outside the polytope of constraints $S$. The main idea is to replace $x_i$ with a new estimation point, by means of projecting $x_i$ on the polytope $S$. In this way, the projection point will be the closest point to $x_i$ satisfying the constraints of the problem. Unfortunately, there is no known closed form expression for the projection of a point on the standard $(M-1)$-simplex in the M-dimensional space. In the following we propose an approximate solution, which is based on the Euclidean distances of $x_i$ from the vertices of $S$. Since the polytope $S$ is convex, any point $x_M$ on a $(M-1)$-dimensional hypersurface of $S$ can be expressed as a linear combination of the corresponding $M-1$ vertices of the polytope, i.e.

$$x_p = \theta_1 e_1 + \theta_2 e_2 + \ldots + \theta_{M-1} e_{M-1}, \quad \text{(IV.12)}$$

where $\theta_i$ are weight coefficients such that $\sum_{i=1}^{M-1} \theta_i = 1$ and $\theta_i \geq 0$, $i = 1, 2, \ldots, M-1$. Apparently, the vertex excluded in (IV.12) is the one with the largest Euclidean distance from $x_i$, which without loss of generality is assumed to be $e_M$. Let $d_1, d_2, \ldots, d_{M-1}$ denote the Euclidean distances between $x_M$ and the $M-1$ remaining vertices of the polytope. Then, the weight coefficients $\theta_i$ in (IV.12) can be computed according to the following relation:

$$\theta_i = \frac{\varphi(d_i)}{\sum_{i=1}^{M-1} \varphi(d_i)}$$

where $\varphi(\cdot)$ is a properly selected function. Two possible choices are $\varphi(x) = \sqrt{x}$ and $\varphi(x) = e^{\sqrt{x}}$. As can be easily verified, in both cases the weight parameter $\theta_i$ is reversely proportional to the Euclidean distance $d_i$, with the second choice weighting more heavily closely located vertices.
V. Results

In this section we present the simulation results for the comparison of the performance of the algorithms presented in the report. Simulations took place either in simulated or real hyperspectral data. The real data are an urban hyperspectral image scene, which is collected by the HYDICE hyperspectral sensor and is publicly available, http://www.agc.army.mil/Hypercube/. The image is composed of 210 spectral bands, in the region from 400 to 2500nm, with a spectral resolution of 10nm. Moreover, the preprocessing of the image involves the removal of the low SNR bands, thus 162 spectral bands remain available. The Matlab source codes used are available in the respective section of the report.

A. Constrained estimators

In Fig. 2, 3 and 4, six fraction abundance planes depict the fractional abundance values of six different endmembers, derived using the LS algorithm, the full additivity LS and the full additivity and nonnegativity LS, solved using a quadratic programming technique [13]. All algorithms were applied to the urban HYDICE hyperspectral image. The image contains six endmembers, namely, road, metal, dirt, grass, tree and roof. The spectral signatures of these endmembers have been identified using a supervised technique, which utilizes areas in the image that seem to contain pure materials, in order to extract the spectrum. The fraction planes exhibit large values (brighter pixels) where the corresponding endmember occupies large fractions of pixels and small values elsewhere (darker pixels). We can observe that the incorporation of any prior knowledge to the estimation problem results in significant improvement in estimation accuracy. It is clear that the discrimination of the endmembers is better achieved in figure 5, when both physical constraints are enforced.
Figure 2: Estimation of abundance fractions using LS algorithm.

Figure 3: Estimation of abundance fractions using full additivity LS algorithm.
B. Bayesian methods

In this section we compare the Bayesian techniques that were presented above for abundance estimation. We simulated a hyperspectral image scene to evaluate the estimation accuracy of the EM algorithm, the Gibbs sampler and the Rao-Blackwell Gibbs sampler. The simulation results are shown in Fig. 5, where the mean squared error (MSE) curves versus the SNR are displayed. In terms of performance, we observe that the Rao-Blackwell Gibbs sampler is equivalent to the conventional Gibbs sampler. It should be noted, however, that the Rao-Blackwell Gibbs sampler needs only a few iterations to converge, thus the equivalence in performance is achieved with much lower computational complexity. Moreover, we observe that both Gibbs samplers outperform the EM algorithm. This performance degradation can be justified by the fact that the EM model considers a stationary prior, as opposed to the hierarchical prior of the Gibbs sampler model.
Figure 5: Simulated data: MSE curves versus SNR, for the algorithms ML, EM, GG and RBGG

C. Proposed method

In this section, the performance of the proposed method is presented, both using simulated and real hyperspectral data. Fig. 6 depicts the simulation results obtained by simulating a hyperspectral image of $10^5$ pixels. The proposed method, which corresponds to the closed form MAP-s projected curve in the figure, is compared to the least squares algorithm, the MAP-s as described in [10] and projected on the polytope of constraints for comparison purposes, and a quadratic programming technique [13], denoted as MAP-h in the figure. The difference between the projected MAP-s and the closed form projected MAP-s is that the former is computed by solving a set of linear matrix inequalities, while the latter is computed using a closed form. We can easily observe that using the closed form solution for the MAP-s is roughly equivalent, in terms of performance, to solving the LMI problem. Note, however, that the same level of performance can be achieved with lower computation complexity in the former case. Moreover the proposed method clearly outperforms the least squares algorithm and its performance is closer to that of the quadratic programming technique. Again, we should point out that quadratic programming methods are computationally more expensive than the proposed method.
The following figures depict the unmixing results of the MAP-h algorithm, the projected MAP-s and the proposed method, respectively, when applied to the urban hyperspectral image scene. In the current experiment we considered four materials present in the image, namely asphalt, roof, grass, and tree. Each image represents the estimated abundances for a specific endmember. Pure white denotes that the estimated abundance of the respective endmember is one, while pure black corresponds to zero. In the last two figures, i.e. in the cases that the MAP-s is used, vectors violating constraints are projected on the polytope of constraints. We should note that a slight degradation in performance is observed as compared with the constrained convex optimization approach. However, the performance of the proposed estimator is similar to that of the LMI-based approach.
Figure 7: Real hyperspectral data. Abundance estimation using quadratic programming techniques.

Figure 8: Real hyperspectral data. Abundance estimation using the MAP-s estimator and solving the LMI optimization problem.
Figure 9: Real hyperspectral data. Abundance estimation using the closed form expressions for the MAP-s estimator.
VI. Matlab source codes

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% SCRIPT
% Urban_ls_adls_maph.m
% This script file compares the estimation performance of
% three algorithms, the unconstrained LS squares, the full additivity
% least squares and the full additivity and nonnegativity LS squares. The
% algorithms are applied to the detection of six endmembers of the HYDICE
% urban hyperspectral image, freely available. The six endmembers are,
% namely, road, metal, dirt, grass, tree and roof.
% (c) 2008 K. Themelis
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

close all; clear; clc
format long eng;
NofEn = 6;

% read the image
im = multibandread('D:\hspd\Urban\Urban', [307 307 210], ...
    'uint16=>double', 0, 'bil', 'ieee-be');
% discard low snr bands and scale
im = .001 * cat(3,im(:,:,5:75),im(:,:,77:86),im(:,:,88:100), ...
    im(:,:,112:135),im(:,:,154:197));
% display one band
figure;
% colormap(gray);
% imagesc(im(:,:,55),[0 1]);
% xlabel('x');
% ylabel('y');
% manually select six endmembers
c1 = squeeze(im(24,255,:)); % road
c2 = squeeze(im(145,65,:)); % metal
c3 = squeeze(im(173,217,:)); % dirt
c4 = squeeze(im(199,165,:)); % grass
c5 = squeeze(im(235,191,:)); % tree
c6 = squeeze(im(127,226,:)); % roof
% plot their spectral signatures
figure;plot(c1)figure;plot(c2);figure;plot(c3);figure;plot(c4);

% organize the mixing matrix
tC = [c1 c2 c3 c4 c5 c6];
[M,N] = size(tC);
% initiate variables
x_ls = zeros(size(im,1),size(im,2),NofEn);
x_maph = zeros(size(im,1),size(im,2),NofEn);
x_adls = zeros(size(im,1),size(im,2),NofEn);
tC_pinv = pinv(tC);
tCZ = inv(tC.'*tC) * ones(N,1) * inv(ones(1,N) * inv(tC.'*tC) * ones(N,1));
for i = 1 : size(im,1)
    for j = 1 : size(im,2)
        y = squeeze(im(i,j,:));

        % Least Squares estimator
        x_ls(i,j,:) = tC_pinv * y;

        % Least Squares estimator
        x_adls(i,j,:) = tC_pinv * y - tCZ * (ones(1,N) * tC_pinv * y - 1);

        % MAPh estimator
        x_maph(i,j,:) = quadprog(2*tC'*tC,-2*tC'*y,eye(NofEn);1*eye ... (NofEn),[ones(NofEn,1); zeros(NofEn,1); ones(1,NofEn),1,[]);...
% plot the abundance fractions of LS estimator
figure;colormap(gray);
subplot(2,3,1);imagesc(x_ls(:,:,1),[0,1]);title('road');axis('image');
subplot(2,3,2);imagesc(x_ls(:,:,2),[0,1]);title('metal');axis('image');
subplot(2,3,3);imagesc(x_ls(:,:,3),[0,1]);title('dirt');axis('image');
subplot(2,3,4);imagesc(x_ls(:,:,4),[0,1]);title('grass');axis('image');
subplot(2,3,5);imagesc(x_ls(:,:,5),[0,1]);title('tree');axis('image');
subplot(2,3,6);imagesc(x_ls(:,:,6),[0,1]);title('roof');axis('image');
% plot the abundance fractions of full additivity LS estimator
figure;colormap(gray);
subplot(2,3,1);imagesc(x_maph(:,:,1),[0,1]);title('road');axis('image');
subplot(2,3,2);imagesc(x_maph(:,:,2),[0,1]);title('metal');axis('image');
subplot(2,3,3);imagesc(x_maph(:,:,3),[0,1]);title('dirt');axis('image');
subplot(2,3,4);imagesc(x_maph(:,:,4),[0,1]);title('grass');axis('image');
subplot(2,3,5);imagesc(x_maph(:,:,5),[0,1]);title('tree');axis('image');
subplot(2,3,6);imagesc(x_maph(:,:,6),[0,1]);title('roof');axis('image');
% plot the abundance fractions of full additivity and nonnegativity
% LS estimator
figure;colormap(gray);
subplot(2,3,1);imagesc(x_adls(:,:,1),[0,1]);title('road');axis('image');
subplot(2,3,2);imagesc(x_adls(:,:,2),[0,1]);title('metal');axis('image');
subplot(2,3,3);imagesc(x_adls(:,:,3),[0,1]);title('dirt');axis('image');
subplot(2,3,4);imagesc(x_adls(:,:,4),[0,1]);title('grass');axis('image');
subplot(2,3,5);imagesc(x_adls(:,:,5),[0,1]);title('tree');axis('image');
subplot(2,3,6);imagesc(x_adls(:,:,6),[0,1]);title('roof');axis('image');
err_ML = zeros(NofVec,1);
mse_ML = zeros(length(SNR_dB),1);
err_EM = zeros(NofVec,1);
mse_EM = zeros(length(SNR_dB),1);
err_GG = zeros(NofVec,1);
mse_GG = zeros(length(SNR_dB),1);
err_GGRB = zeros(NofVec,1);
mse_GGRB = zeros(length(SNR_dB),1);

[N,M] = size(Phi);
w_c = 1/M * ones(M,1);
vita = zeros(NofEMIter+1,length(SNR_dB));
alpha = zeros(NofEMIter+1,length(SNR_dB));
vita1 = zeros(NofGGIter+1,length(SNR_dB));
alpha1 = zeros(NofGGIter+1,length(SNR_dB));
vita11 = zeros(NofGGRBIter+1,length(SNR_dB));
alpha11 = zeros(NofGGRBIter+1,length(SNR_dB));

Phi_pinv = inv(Phi.' * Phi) * Phi.';

for j = 1 : length(SNR_dB)
    for i = 1 : NofVec
        % Model
        t(:,i) = Phi * w(:,i);
spow = t(:,i).' * t(:,i) / length(t(:,i));
npow = ( spow ) / (10 ^ ( SNR_dB(j) / 10 ));
epsilon = sqrt(npow) .* randn(length(t(:,i)),1);
t(:,i) = t(:,i) + epsilon;

        % Least Squares estimator
        w_ML = Phi_pinv * t(:,i);
        err_ML(i) = (w_ML - w(:,i)).' * (w_ML - w(:,i));

        % EM-based Bayesian approach
        epsilon ~ N(0,npow)
w ~ N(w_c/wpow)

        % parameters: wpow, npow
        vita(1,:) = ones(1,length(SNR_dB));
        alpha(1,:) = ones(1,length(SNR_dB));
        Sigma = zeros(NofEn,NofEn,NofEMIter);
        mi = zeros(NofEn,NofEMIter);
        for k = 1: NofEMIter,
            % E-step
            Sigma(:,k) = inv(vita(k,j) * Phi.' * Phi +
                        alpha(k,j) * eye(M));
            mi(:,k) = vita(k,j) * Sigma(:,k) * Phi.' * t(:,i) +
                alpha(k,j) * Sigma(:,k) * w_c;
        % M-step
            alpha(k+1,j) = M/(norm(mi(:,k))^2 + trace(Sigma(:,k,k)));
            vita(k+1,j) = N/(norm( t(:,i) - Phi*mi(:,k),2)^2 +
                trace(Sigma(:,k,k) * Phi.' * Phi));
        end
        err_EM(i) = (mi(:,k) - w(:,i)).' * (mi(:,k) - w(:,i));

        % Gibbs Sampler - Gaussian Distribution
        epsilon ~ N(0,npow)
w ~ N(0,wpow)

        % parameters: wpow-Gamma(1,2/alamda), npow-Gamma(1,2/vlamda)
        % hyperparameters: alamda=1/alamda stot R(*), vlamda=1/vlamda
        % sto R(*)
        vita1(1,:) = ones(1,length(SNR_dB));
alpha1(1,:) = ones(1,length(SNR_db));
w_g = zeros(M,NofGGIter);
mi_w = zeros(M,NofGGIter);
Sigma = zeros(NofEn,NofEn,NofGGIter);
for k = 1: NofGGIter
    % sample w|t,alpha,vita ~ N(mi_w, Sigma)
    temp = inv(vita1(k,j) * Phi.' * Phi + alphal(k,j) * eye(M));
    temp(4) = temp(2);
    temp(7) = temp(3);
    temp(8) = temp(6);
    Sigma(:,:,k) = temp;
    mi_w(:,k) = vital(k,j) * Sigma(:,:,k) * Phi.' * t(:,i) + alphal(k,j) * Sigma(:,:,k) * w_c;
    w_g(:,k) = mvnrnd(mi_w(:,k),Sigma(:,:,k));
    % sample alpha1 ~ Gamma(N/2, 2/( (w(:,i) - wc)*'*(w(:,i) - wc) ));
    kappa = M/2;
    theta = 2 / ( (w_g(:,k) - w_c).' * (w_g(:,k) - w_c) );
    alpha1(k+1,j) = gamrnd(kappa,theta);
    % sample vita1 ~ Gamma(M/2, 2/( (t(:,i) - Phi * w(:,i))*'*(t(:,i) - Phi * w(:,i)) ));
    kappa = N/2;
    theta = 2 / ( (t(:,i) - Phi * w_g(:,k))'* * (t(:,i) - Phi * w_g(:,k)) );
    vita1(k+1,j) = gamrnd(kappa,theta);
end
w_g = w_g(:,NofGGBIter:NofGGIter);
w_GG = sum(w_g,2)/size(w_g,2);
err_GG(i) = (w_GG - w(:,i)).' * (w_GG - w(:,i));

% Gibbs Sampler - Gaussian Distribution - Rao-Blackwell
% epsilon ~ N(0,npow)
% w ~ N(0,wpow)
% parameters: wpow-Gamma(1,2/alambda), npow-Gamma(1,2/blambda)
% hyperparameters: alambda-1/alambda sto R(+), blambda-1/blambda sto R(+)

vital1(1,:) = ones(1,length(SNR_db));
alphall(1,:) = ones(1,length(SNR_db));
w_g = zeros(M,NofGGBIter);
mi_w = zeros(M,NofGGBIter);
Sigma = zeros(NofEn,NofEn,NofGGBIter);
for k = 1: NofGGBIter
    % sample the mean of w|t,alpha,vita ~ N(mi_w, Sigma)
    temp = inv(vital1(k,j) * Phi.' * Phi + alphall(k,j) * eye(M));
    temp(4) = temp(2);
    temp(7) = temp(3);
    temp(8) = temp(6);
    Sigma(:,:,k) = temp;
    mi_w(:,k) = vital1(k,j) * Sigma(:,:,k) * Phi.' * t(:,i) + alphall(k,j) * Sigma(:,:,k) * w_c;
    w_g(:,k) = mvnrnd(mi_w(:,k),Sigma(:,:,k));
    % sample the mean of alphal ~ Gamma(N/2, 2/( (w(:,i) - wc)' * (w(:,i) - wc) )
    kappa = M/2;
    theta = 2 / ( (w_g(:,k) - w_c)' * (w_g(:,k) - w_c) );
    alphall(k+1,j) = kappa * theta;
    % sample the mean of vital1 ~ Gamma(M/2, 2/( (t(:,i) - Phi * w(:,i))'* * (t(:,i) - Phi * w(:,i)) );
    kappa = N/2;
    theta = 2 / ( (t(:,i) - Phi * w_g(:,k))'* * (t(:,i) - Phi * w_g(:,k)) );
    vital1(k+1,j) = kappa * theta;
end
w_g = w_g(:,NofGGBIter:NofGGBIter);
w_GGB = sum(w_g,2)/size(w_g,2);
err_GGB(i) = (w_GGB - w(:,i)).' * (w_GGB - w(:,i));
mse_ML(j) = sum(err_EM,1)/NofVec;
mse_EM(j) = sum(err_EM,1)/NofVec;
mse_GG(j) = sum(err_GG,1)/NofVec;
mse_GGRB(j) = sum(err_GGRB,1)/NofVec;

figure;
hold on;
semilogy(SNR_db,mse_ML,SNR_db,mse_EM,SNR_db,mse_GG,SNR_db,mse_GGRB);
hold off;
legend('ML','EM','GG','GGRB');
xlabel('SNR (dB)');
title('Mean Square Error');

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% SCRIPT
% cmbr_all.m
% This script file compares the estimation performance of
% four algorithms, the unconstrained LS squares, the projected MAP-s, the
% closed form projected MAP-s and the MAP-h, as they are described in the
% technical report . The
% algorithms are applied to simulated hyperspectral data and the
% performance estimation is in terms of mean square error.
% (c) 2008 K. Themelis
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
clc;
clear all;
format long eng;

NofVec = 3000;
NofEn = 4; % sort(ceil(5 + .7 * randn(NofVec,1))); 
NofEst = 4;
NofBnd = 5;

% GENERATE vctr of abundances
% l' * x = 1
% p0 = 0;
pK = 1;
pi = sort( rand(NofEn-1,NofVec));
x = zeros(NofEn,NofVec);
for i = 1 : NofVec
    x(:,i) = [pK - pi(3,i) pi(3,i) -pi(2,i) pi(2,i) -pi(1,i) pi(1,i) -p0 ]';
end
% plot3(x(1,:),x(2,:),x(3,:),'.'); xlabel('x'); ylabel('y');
zlabel('z');

% tC = randn(NofBnd,NofEn);
tC = .5 + .2 * randn(NofBnd,NofEn);
y = zeros(NofBnd,1);
x_ls = zeros(NofEn,1);
x_maps_lmi = zeros(NofEn,1);
x_maps_prov = zeros(NofEn,1);
x_maph_full = zeros(NofEn,1);
% x_maph_add = zeros(NofEn,1);
\( xc = \frac{1}{NofEn} \cdot \text{ones}(NofEn,1); \)

\[
\begin{align*}
SNR_{dB} & = \{0:40\}; \\
\text{mse}_x1 & = \text{zeros}(\text{length}(SNR_{dB}),\text{NofEst}); \\
\text{mse}_x2 & = \text{zeros}(\text{length}(SNR_{dB}),\text{NofEst}); \\
\text{mse}_x3 & = \text{zeros}(\text{length}(SNR_{dB}),\text{NofEst}); \\
\text{mse}_x4 & = \text{zeros}(\text{length}(SNR_{dB}),\text{NofEst}); \\
\text{mse}_{tr} & = \text{zeros}(\text{NofEn},\text{NofEst}+1,\text{NofVec},\text{length}(SNR_{dB})); \\
\text{ERR} & = \text{zeros}(\text{NofEn},\text{NofEst},\text{NofVec}); \\
P_{ellpsd} & = (\text{zeros}(\text{NofEn},\text{NofEn}) - (1/(\text{NofEn}-1)) ) + (\text{NofEn}/(\text{NofEn}-1)) \cdot \text{eye}(\text{NofEn});
\end{align*}
\]

\[
\text{for } i = 1 : \text{length}(SNR_{dB}) \\
\quad \text{for } j = 1 : \text{NofVec} \\
\hspace{1cm} \% \text{some variables} \\
\hspace{1.5cm} \% \text{tx = x(:,j);} \\
\hspace{1.5cm} \text{tCC = tC.' * tC;} \\
\hspace{1.5cm} \text{tCC_inv = inv(tCC);} \\
\hspace{1cm} \% \text{additive noise} \\
\hspace{1.5cm} \% \text{y = tC * tx;} \\
\hspace{1.5cm} \text{vrnc = ( y.' * y ) / (10 .^ (SNR_{dB}(i) ./ 10)) ;} \\
\hspace{1.5cm} \text{noise = sqrt(vrnc) .* randn(length(y),1);} \\
\hspace{1.5cm} \text{Sv = diag(vrnc);} \\
\hspace{1.5cm} \text{Sv_inv = diag(1./vrnc);} \\
\hspace{1.5cm} \text{Sls = inv(tC.' * Sv_inv * tC);} \\
\hspace{1cm} \% \text{model} \\
\hspace{1.5cm} \% \text{y = y + noise;} \\
\hspace{1cm} \% \text{Least Squares estimator - Crammer Rao bound} \\
\hspace{1.5cm} \% \text{x_ls = Sls * tC' * Sv_inv * y;} \\
\hspace{1cm} \% \text{MAPs estimator} \\
\hspace{1.5cm} \% \text{setlmis([]);} \\
\hspace{1.5cm} \text{Sm = lmivar(1,[4 1]);} \\
\hspace{1.5cm} \text{lmitem([-1 1 1 Sm],1,1,'s');} \\
\hspace{1.5cm} \text{lmitem([-1 1 1 0],Sls);} \\
\hspace{1.5cm} \text{lmitem([-1 2 1 0],[1 0 0 0]-xc');} \\
\hspace{1.5cm} \text{lmitem([-1 2 2 0],1);} \\
\hspace{1.5cm} \text{lmitem([-2 1 1 Sm],1,1,'s');} \\
\hspace{1.5cm} \text{lmitem([-2 1 1 0],Sls);} \\
\hspace{1.5cm} \text{lmitem([-2 2 1 0],[0 1 0 0]-xc');} \\
\hspace{1.5cm} \text{lmitem([-2 2 2 0],1);} \\
\hspace{1.5cm} \text{lmitem([-3 1 1 Sm],1,1,'s');} \\
\hspace{1.5cm} \text{lmitem([-3 1 1 0],Sls);} \\
\hspace{1.5cm} \text{lmitem([-3 2 1 0],[0 0 1 0]-xc');} \\
\hspace{1.5cm} \text{lmitem([-3 2 2 0],1);} \\
\hspace{1.5cm} \text{lmitem([-4 1 1 Sm],1,1,'s');} \\
\hspace{1.5cm} \text{lmitem([-4 1 1 0],Sls);} \\
\hspace{1.5cm} \text{lmitem([-4 2 1 0],[0 0 0 1]-xc');} \\
\hspace{1.5cm} \text{lmitem([-4 2 2 0],1);} \\
\hspace{1.5cm} \text{lmitem([-5 1 1 Sm],1,1)};
\]
LMIs = getlmis;
[copt,xopt] = mincx(LMIs,mat2dec(LMIs,eye(NofEn)),[1e-5, 0 ,
0 , 0 , 1]);
Sm_opt = dec2mat(LMIs,xopt,Sm);
x_maps_lmi = inv(tC.' * Sv_inv * tC + pinv(Sm_opt) ) * (tC.'* Sv_inv * y + pinv(Sm_opt) * xc);

% MAPs estimator me provolh
if t1 < 0
    p4 = x_maps_lmi;
    pp = eye(NofEn);
    pp(:,t2) = [];
    if exp(1/norm(pp(:,1)-p4)) + exp(1/norm(pp(:,2)-p4)) +
    exp(1/norm(pp(:,3)-p4)) > 15
        x_maps_prov = x_maps_lmi;
    else
        x_maps_prov = ( exp(1/norm(pp(:,1)-p4))* pp(:,1) +
        exp(1/norm(pp(:,2)-p4))* pp(:,2) + exp(1/norm(pp(:,3)-p4))* pp(:,3) )
    end
else
    x_maps_prov = x_maps_lmi;
end

% MAPh estimator
x_maph_full = quadprog(2*tC'*tC,-2*tC'*y,[eye(NofEn);...
1*eye(NofEn)],[ones(NofEn,1); zeros(NofEn,1)],...
[1 1 1 1],[1,1,[],[]],[0;0;0;0],optimset('LargeScale',...
'off','Display','off'));

% Sygkentrwsh apotelesmatwn se pinaka ekthimsewn
est(:,:,j,i) = [tx  x_ls   x_maps_lmi   x_maps_prov ...
x_maph_full ]; %x_maph_add
end

ERR(:,1,:) = ( est(:,1,:,i) - est(:,2,:,i) ).^2;
ERR(:,2,:) = ( est(:,1,:,i) - est(:,3,:,i) ).^2;
ERR(:,3,:) = ( est(:,1,:,i) - est(:,4,:,i) ).^2;
ERR(:,4,:) = ( est(:,1,:,i) - est(:,5,:,i) ).^2;
% ERR(:,5,:) = ( est(:,1,:,i) - est(:,6,:,i) ).^2;

mse_x1(i,:) = sum(ERR(:,1,:),3)/NofVec;
 mse_x2(i,:) = sum(ERR(:,2,:),3)/NofVec;
 mse_x3(i,:) = sum(ERR(:,3,:),3)/NofVec;
 mse_x4(i,:) = sum(ERR(:,4,:),3)/NofVec;
 mse_tr(i,:) = sum(sum(ERR,1),3)/NofVec;
end

figure;
semilogy(SNR_db,mse_tr(:,1),'r-'
,SNR_db,mse_tr(:,2),'g:',SNR_db,mse_tr(:,3),'b-'
,SNR_db,mse_tr(:,4),'k--'); %SNR_db,mse_tr(:,5),'y'
title('Simulated data');
xlabel('SNR (dB)');
ylabel('MSE');
legend('Least Squares','Map-s','closed form Map-s','Map-h')
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% SCRIPT
% envi_data.m
% This script file compares the estimation performance of
% three algorithms, the unconstrained LS squares, the
% closed form projected MAP-s and the MAP-h, as they are described in the
% technical report. The algorithms are applied to a real hyperspectral
% image data.
% (c) 2008 K. Themelis
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
close all;clear;clc;
format long eng;
NofEn = 4;
mstdev = 1e-4; % (2.234181e3 + 1.82800e3 + 4.08288e3)/3;
% sqrt(500e6); % 88218; % 1e4;
% gia 1e-5 idia me LS
im = multibandread('D:\hspd\Urban\Urban', [307 307 210], 'uint16=>double', 0, 'bil', 'ieee-be');
% discard low snr bands and scale
im = .001 * cat(3,im(:,:,5:75),im(:,:,77:86),im(:,:,88:100),im(:,:,112:135),im(:,:,154:197));

% figure;
% colormap(gray);
% imagesc(im(:,:,1));
% xlabel('x');
% ylabel('y');

c1 = squeeze(im(185,39,:)); % figure;plot(c1) % asphalt
c2 = squeeze(im(115,202,:)); % roof
c3 = squeeze(im(173,83,:)); % grass
c4 = squeeze(im(235,191,:)); % tree
% figure;plot(c1)figure;plot(c2);figure;plot(c3);figure;plot(c4);
tC = [c1 c2 c3 c4];

% Ypolologismo Sls
% mstdev = 1e-4; % (2.234181e3 + 1.82800e3 + 4.08288e3)/3;
% sqrt(500e6); % 88218; % 1e4;
% stdev = mstdev * ones(size(im,3),1);
Sv_inv = diag(1./stdev.^2);
Sls = inv(tC.' * Sv_inv * tC);

x_ls = zeros(size(im,1),size(im,2),NofEn);
x_maph = zeros(size(im,1),size(im,2),NofEn);
x_maps = zeros(size(im,1),size(im,2),NofEn);
x_maps1 = zeros(size(im,1),size(im,2),NofEn);
x_maps2 = zeros(size(im,1),size(im,2),NofEn);
x_maps3 = zeros(size(im,1),size(im,2),NofEn);

xc = 1/NofEn * ones(NofEn,1);
tCC = tC.' * tC;

setlmis([]);
Sm = lmivar(1,[4 1]);
lmiterm([-1 1 1 Sm],[1 1,'s']);
lmiterm([-1 1 0],[Sls]);
lmiterm([-1 2 1 0],[1 0 0 0]-xc');
lmiterm([-1 2 2 0],1);
lmiterm([-2 1 1 Sm],[1 1,'s']);
lmiterm([-2 1 0],[Sls]);
lmiterm([-2 2 1 0],[0 1 0 0]-xc');
lmiterm([-2 2 2 0],1);
lmiterm([-3 1 1 Sm],[1 1,'s']);
lmiterm([-3 1 0],[Sls]);
lmiterm([-3 2 1 0],[0 0 1 0]-xc');
lmiterm([-3 2 2 0],1);
lmiterm([-4 1 1 Sm],[1 1,'s']);
lmiterm([-4 1 0],[Sls]);
lmiterm([-4 2 1 0],[0 0 0 1]-xc');
lmiterm([-4 2 2 0],1);
lmiterm([-5 1 1 Sm],[1 1,1]);
LMIs = getlmis;
[copt,xopt] = mincx(LMIs,mat2dec(LMIs,eye(NofEn)),[1e-5, 0 , 0 , 0 , 1]);
Smx = dec2mat(LMIs,xopt,Sm);

% [V,D] = svd( tC.' * S_v_inv * tC);
% polytope = eye(NofEn);
% z = V' * polytope;
% K = diag(max(z,[],1).^2);

for i = 1 : size(im,1)
    for j = 1 : size(im,2)
        y = squeeze(im(i,j,:));

% Least Squares estimator
% x ls(i,j,:) = pinv( tC ) * y;

% MAPh estimator
% x maph(i,j,:) = quadprog(2*tC'*tC,-2*tC'*y,[eye(NofEn);-1*eye(NofEn)],ones(NofEn,1); zeros(NofEn,1)),[1 1 1 1],i,[],[0;0;0;0],optimset('LargeScale', 'off','Display','off'));

% MAPs estimator
% x_maps(i,j,:) = inv(tC.' * S_v_inv * tC + pinv(Smx) ) * ( tC.'* S_v_inv * y + pinv(Smx) * xc );

% MAPs estimator me provolh

[t1, t2] = min(x_image(i, j,:));
[t4, t3] = max(x_image(i, j,:));

p4 = squeeze(x_image(i, j,:));
pp = eye(NofEn);
pp(:, t2) = [];
if isinf(exp(1/norm(pp(:, 1) - p4)) + exp(1/norm(pp(:, 2) - p4)) +
exp(1/norm(pp(:, 3) - p4))
  x_image1(i, j,:) = x_image(i, j,:);
else
  x_image1(i, j,:) = (exp(1/norm(pp(:, 1) - p4)) * pp(:, 1) +
exp(1/norm(pp(:, 2) - p4)) * pp(:, 2) + exp(1/norm(pp(:, 3) - p4)) *
pp(:, 3)) ... /
  (exp(1/norm(pp(:, 1) - p4)) + exp(1/norm(pp(:, 2) - p4)) +
exp(1/norm(pp(:, 3) - p4)));
end
end

figure; colormap(gray);
subplot(2, 2, 1); imagesc(x_ls(:,:,1)); title('asphalt'); axis('image');
subplot(2, 2, 2); imagesc(x_ls(:,:,2)); title('roof'); axis('image');
subplot(2, 2, 3); imagesc(x_ls(:,:,3)); title('grass'); axis('image');
subplot(2, 2, 4); imagesc(x_ls(:,:,4)); title('tree'); axis('image');
figure; colormap(gray);
subplot(2, 2, 1); imagesc(x_maph(:,:,1)); title('asphalt'); axis('image');
subplot(2, 2, 2); imagesc(x_maph(:,:,2)); title('roof'); axis('image');
subplot(2, 2, 3); imagesc(x_maph(:,:,3)); title('grass'); axis('image');
subplot(2, 2, 4); imagesc(x_maph(:,:,4)); title('tree'); axis('image');
figure; colormap(gray);
subplot(2, 2, 1); imagesc(x_image(:,:,1)); title('asphalt'); axis('image');
subplot(2, 2, 2); imagesc(x_image(:,:,2)); title('roof'); axis('image');
subplot(2, 2, 3); imagesc(x_image(:,:,3)); title('grass'); axis('image');
subplot(2, 2, 4); imagesc(x_image(:,:,4)); title('tree'); axis('image');
VII. References