Unmixing K-Gaussians with Application to Hyperspectral Imaging

Yonatan Woodbridge, Uri Okun, Gal Elidan and Ami Wiesel

***Abstract*— In this paper we consider parameter estimation of K Gaussians, given convex combinations of their realizations. In the remote sensing literature this setting is known as the normal compositional model (NCM) and has shown promising gains in modelling hyperspectral images. Current NCM parameter estimation techniques are based on Bayesian methodology and are computationally slow and sensitive to their prior assumptions. Here we introduce a deterministic variant of the NCM, named DNCM, which assumes that the unknown mixing coefficients are non-random. This leads to a standard Gaussian model, with a simple estimation procedure which we denote by K-Gaussians. Its iterations are provided in closed form, and do not require any sampling schemes or simplifying structural assumptions. We illustrate the performance advantages of K-Guassians using synthetic and real images, in terms of accuracy and computational costs in comparison to state of the art. We also demonstrate the use of our algorithm in hyperspectral target detection on a real image with known targets.**

***Index Terms*— Normal compositional model, Hyperspectral unmixing**

I. Introduction

Hyperspectral images (HSI) provide both spatial and spectral information of a scanned scene and are useful in material identification, target detection and other remote sensing tasks [1], [2]. Different objects typically leave unique signatures in the electromagnetic spectrum. Unmixing methods decompose the high dimensional image into these fundamental components which typically have a physical interpretation and may be useful in statistical inference. The Normal Compositional Model (NCM) and its modern extensions are among the leading approaches to this problem [3]. In this paper, we propose a non-Bayesian variation on these models that leads to a low complexity yet accurate unmixing algorithm.

An HSI is a 3-dimensional data cube, consisting of two spatial dimensions that determine pixel location and a third dimension that corresponds to spectral bands. The goal of unmixing is to retrieve the material composition in the image [2], [4]. Given spectral information contained in each pixel, this can be made possible owing to the fact that every material is characterized by its reflection and absorption patterns in the wide spectrum. Narrow spectral band measurements typically dictate low spatial resolution, in which multiple materials may be present in a single pixel. Therefore, most of the leading methods decompose the image into a few pure components, known as *endmembers*, which are linearly mixed across the pixels. The mixing coefficients are known as *abundance*.

Two classical unmixing methods are vertex component analysis (VCA) and non-negative matrix factorization (NMF) [5], [6]. VCA assumes presence of pure pixels in the image, and uses geometric methods to extract the endmembers. NMF methods assume a bilinear structure and seek an optimal decomposition into endmember and abundance matrices. Other variants of the NMF have been proposed, imposing additional sparsity or smoothness constraints [7].

A major concern with the classical decomposition methods is that they do not account for endmember variability. In practice, spectra of pure materials might vary across pixels, caused by intrinsic variabilty, atmospheric conditions, illumination and more [8]. Variability can be captured by additive perturbation terms as in [9]. Alternatively, it is natural to consider the endmembers as random vectors with unknown statistical distributions. The goal is then to estimate the parameters of these distributions given linear combinations of random vectors. There are two main approaches in this literature. The *stochastic mixing model* (SMM) assumes that the abundances are restricted to a discrete set [10], [11], while *compositional models* assume continuous abundances. The latter approach is known to be more accurate and flexible, and has been the focus of many recent papers. The basic model is known as *normal compositional model* (NCM), which assumes multivariate normal distribution of the endmembers. Variants of the NCM assume other endmember priors as the Beta distribution [12]. In parallel to the submission of our work, a similar approach based on a generalized Gaussian mixture model and maximum a-posteriori estimation (MAP) method was proposed in [13]. A related work is the piecewise convex unmixing [14], where a few groups of endmembers are assumed, characterizing different NCM distributions, so each pixel is a sample from one of these distributions.

There are of course approaches other than the NCM to model endmember variability. For example, the model proposed in [15], introduces an additive function to a linear mixture model that depends on the abundance. This approach is more general in a sense that it can also account for non-linearity or mismodelling effects.

The majority of methods described above are based on a Bayesian methodology and assume random abundances. The original NCM paper [3] uses a uniform prior distribution over the simplex and a stochastic expectation maximization (SEM) method is utilized in order to achieve parameter estimation. The article [16] proposes a particle swarm optimization method for estimation of the abundance, which is assumed to have a normal distribution. In [17], the authors propose a hybrid Gibbs sampler to obtain samples of the abundance, assuming Gaussian endmembers with spherical/diagonal covariances. An extension proposed in [18] also includes estimation of the number of endmembers. An unsupervised generalization denoted by UsGNCM exploits spatial properties and also includes estimation of endmember

means [19]. However, it still assumes spherical covariances.

In this paper, we introduce a straightforward yet effective version of NCM which we call the deterministic NCM (DNCM). Following the original NCM, we assume arbitrary Gaussian sources, with no simplifying structural constraints. However, we differ by not assuming any stochastic prior on the abundance vectors, considering them to be additional deterministic parameters. Thus, the set of parameters to be estimated are the mean and covariance of each endmember, as well as the abundance vector of each pixel. We also assume additive Gaussian noise of constant variance. We perform unmixing through maximum likelihood estimation (MLE), and for that purpose we turn to the expectation-maximization (EM) method [20]. Our latent variables are the unknown Gaussian realizations rather than their mixing coefficients. The resulting approach, which we call K-Gaussians, consists of closed form computations and standard optimization techniques. It allows a more realistic scenario of full endmember covariances. It is also more flexible in the sense that one can impose various constraints on the abundance based on prior knowledge, such as non-negativity, simplex (non-negative sum to one), and upper/lower bounds. Most importantly, the algorithm is significantly simpler and more scalable than its Bayesian competitors.

We examine K-Gaussians in synthetic HSI, comparing its estimation errors against other unmixing algorithms. Our main competitor is UsGNCM which typically provides accurate results when covariances are spherical, but is computationally intensive. We observe comparable or improved accuracy with a significant reduction of computational resources. For example, in the setting of [19] using their UsGNCM code, K-Gaussians requires only a few minutes compared to at least one hour of computation, with negligible loss of accuracy. Finally, we demonstrate the use of K-Gaussians in the application of hyperspectral target detection, using a real world HSI experiment with real known targets [21]. Remarkably, in addition to its efficiency, K-Gaussians also provides better receiver operating characteristics (ROC) than its competitors. This experiment involves a huge dataset, and we conjecture that UsGNCM would eventually outperform K-Gaussians given enough running time and memory resources (See Section IV for more details). Our conclusion is therefore that K-Gaussians is a promising scalable low-cost alternative to Bayesian NCM methods.

The rest of the paper is organized as follows. In section II, we introduce the basic DNCM model, compare it to existing models, and address its advantages and limitations. In section III, we introduce K-Gaussians algorithm. Finally, in section IV, we evaluate our model and EM algorithm using numerical experiments on synthetic and real hyperspectral data.

Notations: The field of real numbers is $\mathbb{R}$, and non-negative numbers is $\mathbb{R}_+$. The set of $p\times p$ positive definite matrices is denoted by $\mathbb{S}_{++}^{p\times p}$. Matrices are denoted by boldface upper-case letters (e.g, $\mathbf{A}, \mathbf{M}$), while vectors are denoted by boldface lower-case letters ($\mathbf{u}, \mathbf{v}$). The k-th column vector of some matrix (e.g, $\mathbf{M}$) is denoted by the matching lower-case, indexed by k (e.g, $\mathbf{m}_k$). The (i, j)-entry of a matrix, say $\mathbf{M}$, is denoted by $[\mathbf{M}]_{ij}$. The n-th entry of a vector $\mathbf{u}$ is denoted by u_n. If the vector itself is indexed, e.g. $\mathbf{m}_k$, then we denote its n-th entry by $[\mathbf{m}_k]_n$. As we shall see, the DNCM model consists of a set of covariances. We denote this set by $\mathcal{Q}$, and the k-th covariance in the set is denoted by $\mathbf{Q}^{(k)}$. Note that this should not be confused with a regular matrix, which we denote by bold upper case letter, other than $\mathbf{Q}$. We denote the ℓ_2 vector norm by $\|\cdot\|_2$.

II. The DNCM model

A. Problem formulation

The DNCM model consists of a fixed number K of independent sources. Each has a multivariate normal distribution with a different mean and covariance. Let $\mathbf{m}_k \in \mathbb{R}^p$ and $\mathbf{Q}^{(k)} \in \mathbb{S}_{++}^{p\times p}$ be the mean and covariance of the k-th component, respectively. A DNCM random vector is a linear combination of these Gaussian components, corrupted by an additional independent Gaussian noise. We refer to the coefficients vector as the abundance, and to the independent Gaussian components as endmembers. The abundance entries must therefore be non-negative. We can of course impose additional constraints, based on prior knowledge of the abundance. In our simulations, we shall focus on two types of constraints: non-negativity, as in VCA [22], and simplex. We denote the constraint set by $\mathcal{C}$, so that we have:

$$\mathcal{C} = \left\{\mathbf{a} \in \mathbb{R}_+^K\right\} \text{ or } \mathcal{C} = \left\{\mathbf{a} \in \mathbb{R}_+^K, \mathbf{1}^T\mathbf{a} = 1\right\}.$$

Formally, a DNCM random vector $\mathbf{y}$ is defined as:

$$\begin{aligned} \mathbf{y} &= \sum_{k=1}^{K} a_k \mathbf{x}_k + \boldsymbol{\epsilon} \\ \mathbf{x}_k &\sim \mathcal{N}\left(\mathbf{m}_k, \mathbf{Q}^{(k)}\right) \\ \boldsymbol{\epsilon} &\sim \mathcal{N}\left(\mathbf{0}, \sigma^2\mathbf{I}\right), \end{aligned}$$

where $[a_1, .., a_K]^T = \mathbf{a} \in \mathcal{C}$. In words, $\mathbf{x}_k$ is a realization of the k-th source, $\mathbf{m}_k$ and $\mathbf{Q}_k$ are its parameters, and $\boldsymbol{\epsilon}$ is an independent vector of Gaussian noise. We sometimes use a shorter matrix notation, for convenience, as follows:

$$\begin{aligned} \mathbf{M} &= [\mathbf{m}_1, ..., \mathbf{m}_K] \\ \mathbf{X} &= [\mathbf{x}_1, ..., \mathbf{x}_k] \\ \mathcal{Q} &= \{\mathbf{Q}^{(1)}, ..., \mathbf{Q}^{(K)}\} \\ \mathbf{A} &= [\mathbf{a}_1, ..., \mathbf{a}_N]. \end{aligned} \tag{1}$$

The matrices $\mathbf{M}$ and $\mathbf{X}$ are stacked vectors of means and endmember realizations, respectively, while $\mathcal{Q}$ is the set of endmember covariances. Using these matrix notations, $\mathbf{y}$ can be compactly expressed as:

$$\mathbf{y} = \mathbf{X}\mathbf{a} + \boldsymbol{\epsilon}.$$

It is easy to verify that $\mathbf{y}$ is normally distributed, with the following parameters:

$$\mathbf{y} \sim \mathcal{N}\left(\mathbf{M}\mathbf{a}, \mathbf{Q}\left(\mathbf{a}\right) + \sigma^2\mathbf{I}\right),$$

where

$$\mathbf{Q}\left(\mathbf{a}\right) = \sum_{k=1}^{K} a_k^2 \mathbf{Q}^{(k)}. \tag{2}$$

In practice, we observe N vectors:

$$\mathbf{Y} = [\mathbf{y}_1, ..., \mathbf{y}_N], \tag{3}$$

where each observation is a linear mixing of the K normally distributed endmembers. However, the abundance vectors $\mathbf{a} \in \mathcal{C}$ of different observation are not necessarily identical. We denote the abundance of the n-th observation $\mathbf{y}_n$ by $\mathbf{a}_n$. Then we have:

$$\mathbf{y}_n \sim \mathcal{N}(\mathbf{M}\mathbf{a}_n, \mathbf{Q}\left(\mathbf{a}_n\right) + \sigma^2\mathbf{I}) \tag{4}$$

Omitting constant terms, the negative log-likelihood (NL) of a single observation $\mathbf{y}$, corresponding to an abundance vector $\mathbf{a}$, is given by

$$\begin{aligned}\mathcal{L}(\mathbf{y};\mathbf{a},\mathbf{M},\mathcal{Q},\sigma^2) = \\ (\mathbf{y}-\mathbf{M}\mathbf{a})^T\left[\mathbf{Q}\left(\mathbf{a}\right)+\sigma^2\mathbf{I}\right]^{-1}(\mathbf{y}-\mathbf{M}\mathbf{a}) \\ +\log\left(\left|\mathbf{Q}\left(\mathbf{a}\right)+\sigma^2\mathbf{I}\right|\right).\end{aligned} \tag{5}$$

B. Relation to other models

DNCM is closely related to other classical statistical and factorization models, as detailed below:

NCM [3]: The motivation for DNCM are the stochastic NCMs. Both models assume linear combinations of hidden Gaussians. The only difference is that the vectors $\mathbf{a}_n$ are modelled as random variables in NCM, whereas in DNCM they are deterministic unknowns. Compared to GNCM of [19], DCNM is more general and does not constrain the covariances of the endmembers to have spherical/diagonal structure.

GMM [23]: This model can be regarded as a pure NCM. The GMM assumes 1-sparse abundance vectors, i.e., vectors with a single unit entry and zeros elsewhere. Similarly to NCM, it typically also models the abundance as stochastic and uses the marginalized mixture distribution

$$p(\mathbf{y}) = \sum_{k=1}^{K} \pi_k \phi(\mathbf{y}; \mathbf{m}_k, \mathbf{Q}^{(k)}), \tag{6}$$

where $(\pi_1, .., \pi_K) \in \mathcal{C}$ is the distribution of the pure pixels.

K-means [24]: DNCM is also related to the K-means algorithm which can be derived as a degenerate GMM with spherical covariances. Like GMM, K-means assumes 1-sparse abundances. On the other hand, the K-means is a deterministic algorithm which assigns each observation to a single cluster, and does not perform any soft marginalization. In this sense, it is closer to DNCM which also assumes deterministic abundances.

NMF [6]: DNCM can be interpreted as a generalized NMF model. If we set $\mathbf{Q}^{(k)} \to \mathbf{0}$, then DNCM reduces to the classical NMF formulation in which $\mathbf{Y} = \mathbf{M}\mathbf{A} + \boldsymbol{\epsilon}$. By allowing more flexible covariances, DNCM expresses the uncertainty and variability in the different endmembers. More precisely, NMF models also requires that the matrix $\mathbf{M}$ will have non-negative elements. This constraint can also be easily incorporated into DNCM.

C. Inherent limitations and identifiability

As explained above, DNCM is a highly expressive model which generalizes many existing models. As a consequence, it has many unknown parameters that must be estimated, and it is important to understand its inherent limitations and identifiability properties. Clearly, it inherits all the identifiability problems of GMM and NMF. These models suffer from trivial ambiguities such as cluster/vertex permutation.

A more critical limitation concerns the abundance parameters. Unlike the endmembers, the abundance varies with each observation. Consequently, as we increase the number of observations, we get more unknowns. Thus, we only expect accurate estimation of $\mathbf{M}$ and $\mathcal{Q}$, and may not succeed in reconstructing all the abundances in $\mathbf{A}$. In specific scenarios, such as partial knowledge of the model parameters, given certain conditions on the latter, identifiability can be guaranteed. However, as is typically the case for mixture models, this is mostly a theoretical concern. As we shall see in our experiments, our model still performs well.

III. Estimation

In this section, we introduce the K-Gaussians method for estimating $\mathbf{M}$, $\mathcal{Q}$, $\mathbf{A}$ and σ^2, given an observation matrix and number of endmembers. Similarly to NCM [3], we adopt an EM approach. Basically, the EM algorithm computes the MLE of a model that contains latent variables, by optimizing the conditional likelihood function. Every iteration of the EM algorithm increases the likelihood function, and under certain regularity conditions (that we do not prove here for the DNCM), is guaranteed to converge to a local maximum [20], [25]. Note that our objective is to minimize the NL, which is equivalent to computing the MLE. In our case the endmembers are latent variables, a property which allows for computational efficiency. Specifically, the DNCM log likelihood decouples the endmembers' mean and covariance from the abundance vectors. Minimization is then done separately for each group of parameters.

We first present the main computational steps of K-Gaussians. We then discuss covariance regularization, as well as choice of initialization points. Finally, we introduce a diagonal version of the K-Gaussians.

A. The K-Gaussians algorithm

Let us denote the current set of parameter estimates by:

$$\hat{\Theta} = \{\hat{\mathbf{A}}, \hat{\mathbf{M}}, \hat{\mathcal{Q}}, \hat{\sigma}^2\}, \tag{7}$$

where we use matrix notation:

$$\begin{aligned}\hat{\mathbf{M}} &= [\hat{\mathbf{m}}_1, ..., \hat{\mathbf{m}}_K] \\ \hat{\mathbf{A}} &= [\hat{\mathbf{a}}_1, ..., \hat{\mathbf{a}}_N] \\ \hat{\mathcal{Q}} &= \{\hat{\mathbf{Q}}^{(1)}, ..., \hat{\mathbf{Q}}^{(K)}\}.\end{aligned}$$

The t-th iteration estimate is denoted by $\hat{\Theta}^{(t)}$, but in our analysis we omit the index for clarity.

The E-step. In what follows, we temporarily omit the n index, and refer to a single observation $\mathbf{y}$ and a single abundance vector $\mathbf{a}$. Considering each column $\mathbf{x}_k$ of $\mathbf{X}$ as a latent variable, the E-step involves computing the expected *complete* log likelihood. The complete likelihood, which includes the latent variables, is given by:

$$p(\mathbf{y}, \mathbf{X}; \mathbf{a}, \mathbf{M}, \mathcal{Q}) = p(\mathbf{y}|\mathbf{X}; \mathbf{a}, \mathbf{M}, \mathcal{Q}) p(\mathbf{X}; \mathbf{a}, \mathbf{M}, \mathcal{Q}).$$

Since $\mathbf{y} - \mathbf{X}\mathbf{a} \sim \mathcal{N}(\mathbf{0}, \sigma^2\mathbf{I})$, it follows that $\mathbf{y}|\mathbf{X}; \mathbf{a}, \mathbf{M}, \mathcal{Q}, \sigma^2 \sim \mathcal{N}(\mathbf{X}\mathbf{a}, \sigma^2\mathbf{I})$. The complete log likelihood, after omitting constants, is then:

$$\mathcal{L}_c(\mathbf{y}, \mathbf{X}; \mathbf{a}, \mathbf{M}, \mathcal{Q}, \sigma^2) = p\log\sigma^2 + \frac{1}{\sigma^2}\underbrace{\|\mathbf{y} - \mathbf{X}\mathbf{a}\|_2^2}_{\mathcal{L}_q(\mathbf{y}, \mathbf{X}; \mathbf{a})} + \sum_{k=1}^{K}\underbrace{\left(\log|\mathbf{Q}^{(k)}| + (\mathbf{x}_k - \mathbf{m}_k)^T\left(\mathbf{Q}^{(k)}\right)^{-1}(\mathbf{x}_k - \mathbf{m}_k)\right)}_{\mathcal{L}_g\left(\mathbf{x}_k; \mathbf{m}_k, \mathbf{Q}^{(k)}\right)}, \tag{8}$$

where we denoted the quadratic term of $\mathcal{L}_c$ by $\mathcal{L}_q$, and the Gaussian likelihood within the sum by $\mathcal{L}_g$.

We defer the detailed derivation of the conditional expectation to the appendix, and present here the main results. Consider first the quadratic term:

$$\mathbb{E}\left(\mathcal{L}_q\left(\mathbf{y}, \mathbf{X}; \mathbf{a}\right)|\mathbf{y}, \hat{\Theta}\right), \tag{9}$$

using the notation in (7). In detail, we have:

$$\mathbb{E}\left(\mathcal{L}_q\left(\mathbf{y}, \mathbf{X}; \mathbf{a}\right)|\mathbf{y}, \hat{\Theta}\right) = \mathbf{y}^T\mathbf{y} + \mathbf{a}^T\mathbb{E}\left(\mathbf{X}^T\mathbf{X}|\mathbf{y}, \hat{\Theta}\right)\mathbf{a} - 2\mathbf{y}^T\mathbb{E}\left(\mathbf{X}|\mathbf{y}, \hat{\Theta}\right)\mathbf{a}. \tag{10}$$

Thus, we need to compute the conditional expectations $\mathbb{E}\left(\mathbf{X}|\mathbf{y}, \hat{\Theta}\right)$ and $\mathbb{E}\left(\mathbf{X}^T\mathbf{X}|\mathbf{y}, \hat{\Theta}\right)$. The first is given by:

$$\mathbb{E}\left(\mathbf{x}_k|\mathbf{y}, \hat{\Theta}\right) = \hat{\mathbf{m}}_k + \hat{a}_k\hat{\mathbf{Q}}^{(k)}\left[\hat{\mathbf{Q}}\left(\hat{\mathbf{a}}\right) + \hat{\sigma}^2\mathbf{I}\right]^{-1}\left(\mathbf{y} - \hat{\mathbf{M}}\hat{\mathbf{a}}\right), \tag{11}$$

where $\mathbf{x}_k$ is the k-th column of $\mathbf{X}$. The second expectation admits the form:

$$\left[\mathbb{E}\left(\mathbf{X}^T\mathbf{X}|\mathbf{y}, \hat{\Theta}\right)\right]_{ij} = \left[\mathbb{E}\left(\mathbf{X}|\mathbf{y}, \hat{\Theta}\right)^T\mathbb{E}\left(\mathbf{X}|\mathbf{y}, \hat{\Theta}\right)\right]_{ij} + \mathrm{Tr}\left\{\hat{\mathbf{Z}}^{ij}\right\}, \tag{12}$$

where:

$$\hat{\mathbf{Z}}^{ij} = \mathbb{1}_{i=j}\cdot\hat{\mathbf{Q}}^{(i)} - \left[\hat{\mathbf{a}}\right]_i\left[\hat{\mathbf{a}}\right]_j\hat{\mathbf{Q}}^{(i)}\left[\hat{\mathbf{Q}}\left(\hat{\mathbf{a}}\right) + \hat{\sigma}^2\mathbf{I}\right]^{-1}\hat{\mathbf{Q}}^{(j)}, \tag{13}$$

where $\mathbb{1}_{i=j}$ is the indicator function which equals to 1 if $i = j$, and 0 otherwise.

Regarding $\mathcal{L}_g$ in (8), this can be written as:

$$\log|\mathbf{Q}^{(k)}| + \mathrm{Tr}\left\{\left(\mathbf{Q}^{(k)}\right)^{-1}\mathbf{x}_k\mathbf{x}_k^T\right\} - 2\mathbf{m}_k^T\left(\mathbf{Q}^{(k)}\right)^{-1}\mathbf{x}_k + \mathbf{m}_k^T\left(\mathbf{Q}^{(k)}\right)^{-1}\mathbf{m}_k. \tag{14}$$

The conditional expectation depends on $\mathbb{E}\left(\mathbf{x}_k|\mathbf{y}, \hat{\Theta}\right)$ (see (11)) and $\mathbb{E}\left(\mathbf{x}_k\mathbf{x}_k^T|\mathbf{y}, \hat{\Theta}\right)$, which depends on previous terms (11)-(13):

$$\mathbb{E}\left(\mathbf{x}_k\mathbf{x}_k^T|\mathbf{y}, \hat{\Theta}\right) = \mathrm{Cov}\left(\mathbf{x}_k|\mathbf{y}, \hat{\Theta}\right) + \mathbb{E}\left(\mathbf{x}_k|\mathbf{y}, \hat{\Theta}\right)\mathbb{E}\left(\mathbf{x}_k|\mathbf{y}, \hat{\Theta}\right)^T, \tag{15}$$

where

$$\mathrm{Cov}\left(\mathbf{x}_k|\mathbf{y}, \hat{\Theta}\right) = \hat{\mathbf{Q}}^{(k)} - \hat{a}_k^2\hat{\mathbf{Q}}^{(k)}\left[\hat{\mathbf{Q}}\left(\hat{\mathbf{a}}\right) + \hat{\sigma}^2\mathbf{I}\right]^{-1}\hat{\mathbf{Q}}^{(k)}. \tag{16}$$

See appendix for detailed calculations.

The M-step. Having derived E-step expressions, we can now optimize the conditional log likelihood over $\hat{\Theta}$. This is done in a few steps. First, minimization over the n-th abundance vector, which is achieved by solving:

$$\tilde{\mathbf{a}}_n = \min_{\mathbf{a}_n \in \mathcal{C}}\left[\mathbf{a}_n^T\mathbb{E}\left(\mathbf{X}_n^T\mathbf{X}_n|\mathbf{y}_n, \hat{\Theta}\right)\mathbf{a}_n - 2\mathbf{y}_n^T\mathbb{E}\left(\mathbf{X}_n|\mathbf{y}_n, \hat{\Theta}\right)\mathbf{a}_n\right], \tag{17}$$

where $\mathbf{y}_n$ is the n-th observation, $\mathbf{X}_n$ is the corresponding latent variable matrix, and $\mathcal{C}$ is the relevant constraint. Recall that each column of $\mathbb{E}\left(\mathbf{X}_n|\mathbf{y}_n, \hat{\Theta}\right)$ is computed using (11). We use tilde $\tilde{(\cdot)}$ to denote a minimizer, as opposed to hat $\hat{(\cdot)}$, which denotes the current parameters estimate over which the latent variables are conditioned.

A few remarks are in need. First, under non-negativity or simplex constraints, the problem (17) is a quadratic program, which can be solved using log barrier method (see chapter 11, [26]), and whose solver is available in most scientific toolboxes. Second, the solution is independent of the unknown variance σ^2 which will be optimized later. Third, this problem is solved N times, once for each data point $\mathbf{y}_n$ using the corresponding abundance vector $\mathbf{a}_n$. These optimization procedures are decoupled and can be efficiently computed in parallel. Finally, both the quadratic objective and the simplex constraint are empirically shown to induce sparsity of the abundance $\mathbf{a}_n$. This usually brings advantage, especially in HSI, where pixels are typically composed of a subset of sources, and not the entire set.

Next, since

$$\tilde{\sigma}^2 = \min_{\sigma^2}\sum_{n=1}^{N}\mathbb{E}\Big(p\log\sigma^2 + \frac{1}{\sigma^2}\|\mathbf{y}_n - \mathbf{X}_n\hat{\mathbf{a}}_n\|_2^2\Big|\mathbf{y}_n, \hat{\Theta}\Big),$$

it is easy to verify that:

$$\tilde{\sigma}^2 = \frac{1}{Np}\sum_{n=1}^{N}\left(\mathbf{y}_n^T\mathbf{y}_n - 2\mathbf{y}_n^T\mathbb{E}\left(\mathbf{X}_n|\mathbf{y}_n, \hat{\Theta}\right)\hat{\mathbf{a}}_n + \hat{\mathbf{a}}_n^T\mathbb{E}\left(\mathbf{X}_n^T\mathbf{X}_n|\mathbf{y}_n, \hat{\Theta}\right)\hat{\mathbf{a}}_n\right). \tag{18}$$

Finally, minimization over $\mathbf{M}, \mathbf{Q}$ depends only on $\mathcal{L}_g$ in (8). This is achieved through sample mean and covariance of the expected term of (14). Adding the n-index of observation number, we get:

$$\left\{\tilde{\mathbf{m}}_k, \tilde{\mathbf{Q}}^{(k)}\right\} = \min_{\mathbf{M}\geq 0, \mathcal{Q}}\sum_{n=1}^{N}\mathbb{E}\left(\mathcal{L}_g\left(\left[\mathbf{X}_n\right]_k; \mathbf{m}_k, \mathbf{Q}^{(k)}\right)|\mathbf{y}_n, \hat{\Theta}\right), \tag{19}$$

where $\mathbf{M} \geq 0$ denotes the non-negativity constraint over each of $\mathbf{M}$'s entries. Ignoring that constraint, the minimizer is the standard sample mean and covariance:

$$\tilde{\mathbf{m}}_k = \frac{1}{N}\sum_{n=1}^{N}\mathbb{E}\left(\left[\mathbf{X}_n\right]_k|\mathbf{y}_n, \hat{\Theta}\right), \tag{20}$$

Algorithm 1 `Alt-Min`

Input: data matrix $\mathbf{Y} \in \mathbb{R}^{p\times N}$, $\tilde{\mathbf{Q}}^{(k)}$, $\mathbb{E}\left([\mathbf{X}_n]_k \,|\mathbf{y}_n, \hat{\Theta}^{(t)}\right)$, and $\mathbb{E}\left([\mathbf{X}_n]_k [\mathbf{X}_n]_k^T \,|\mathbf{y}_n, \hat{\Theta}^{(t)}\right)$ for $n = 1, .., N$.
Until convergence

- Compute $\tilde{\mathbf{m}}_k$ by solving (22).
- Compute $\tilde{\mathbf{Q}}^{(k)}$ through (21).

End until
Return $\tilde{\mathbf{m}}_k, \tilde{\mathbf{Q}}^{(k)}$.

$$\tilde{\mathbf{Q}}^{(k)} = \frac{1}{N}\sum_{n=1}^{N} \mathbb{E}\left([\mathbf{X}_n]_k [\mathbf{X}_n]_k^T \,|\mathbf{y}_n, \hat{\Theta}\right) - \tilde{\mathbf{m}}_k \tilde{\mathbf{m}}_k^T, \quad (21)$$

where $[\mathbf{X}_n]_k$ is the k-th column of the matrix $\mathbf{X}_n$. See appendix for a detailed derivation of (21).

Since (20) does not guarantee non-negativity, we propose an additional alternating-minimization procedure in case where that constraint is violated. In that procedure one iteratively minimizes the objective (19) separately over $\mathbf{M}$ and $\mathbf{Q}^{(k)}$ as follows:

$$\tilde{\mathbf{m}}_k \leftarrow \min_{\mathbf{m}\geq 0} \quad \Big\{\mathbf{m}^T \left(\tilde{\mathbf{Q}}^{(k)}\right)^{-1} \mathbf{m} - 2\mathbf{m}^T \left(\tilde{\mathbf{Q}}^{(k)}\right)^{-1} \sum_{n=1}^{N} \mathbb{E}\left([\mathbf{X}_n]_k \,|\mathbf{y}_n, \hat{\Theta}^{(t)}\right)\Big\}. \quad (22)$$

As before, this is a standard quadratic program with non-negative constraints. We then compute $\tilde{\mathbf{Q}}^{(k)}$ through (21). See pseudocode in Algorithm 1.

B. Covariance Regularization

The proposed DNCM model has many unknown parameters including positive definite covariance matrices, which must be inverted within its iterations. To avoid numerical and statistical deficiencies and ensure well conditioned iterations, we recommend to regularize the estimated covariances. We suggest a simple shrinkage of the sample covariance with diagonal loading. This is a standard procedure due to Ledoit and Wolf [27]. More specifically, after every calculation of the sample covariance $\tilde{\mathbf{Q}}^{(k)}$ in the M-step (21), we modify the estimate according to [28]:

$$\tilde{\mathbf{Q}}^{(k)} \leftarrow (1-\alpha_k)\tilde{\mathbf{Q}}^{(k)} + \alpha_k \frac{\mathrm{Tr}(\tilde{\mathbf{Q}}^{(k)})}{p}\mathbf{I} \quad (23)$$

given

$$\zeta\left(\tilde{\mathbf{Q}}^{(k)}\right) = \frac{p\mathrm{Tr}\Big\{\left(\tilde{\mathbf{Q}}^{(k)}\right)^2\Big\}}{\left(\mathrm{Tr}\big\{\tilde{\mathbf{Q}}^{(k)}\big\}\right)^2} - 1, \quad (24)$$

the value α_k is estimated then by:

$$\alpha_k \approx \frac{1}{N}\frac{\zeta(\tilde{\mathbf{Q}}^{(k)}) - \frac{p}{N} + 1 + p}{\zeta(\tilde{\mathbf{Q}}^{(k)})}. \quad (25)$$

A pseudocode that fully summarizes the K-Gaussians is given in algorithm 2.

Note that the regularization procedure might hinder the convergence of K-Gaussians, causing an increase in the negative log likelihood at some iterations. We observe through numerical experiments that for sufficiently large samples, the values of α_k are small enough not to affect the negative log likelihood tendency to decrease.

Algorithm 2 K-`Gaussians`

Input: data matrix $\mathbf{Y} \in \mathbb{R}^{p\times N}$.
Initialize: $t \leftarrow 0$
$\hat{\Theta}^{(t)} = \{\hat{\mathbf{M}}^{(t)}, \hat{\mathcal{Q}}^{(t)}, \hat{\mathbf{A}}^{(t)}, (\hat{\sigma}^2)^{(t)}\}$,
($\hat{\Theta}^{(t)}$ is the t-th iteration estimate.)
K components (number of columns in $\hat{\mathbf{M}}^{(0)}$)
Until convergence

- **For each** $\mathbf{y}_n$, $n = 1, .., N$:
  - Compute $\mathbb{E}\left(\mathbf{X}_n|\mathbf{y}_n, \hat{\Theta}^{(t)}\right), \mathbb{E}\left(\mathbf{X}_n^T\mathbf{X}_n|\mathbf{y}_n, \hat{\Theta}^{(t)}\right)$ and $\mathbb{E}\left([\mathbf{X}_n]_k [\mathbf{X}_n]_k^T \,|\mathbf{y}_n, \hat{\Theta}^{(t)}\right)$ (11)-(13),(15)-(16).
  - $\tilde{\mathbf{a}}_n \leftarrow \min_{\mathbf{a}\in\mathcal{C}} \mathbb{E}\left(\mathcal{L}_q\left(\mathbf{a}, \mathbf{y}_n, \mathbf{X}_n\right)|\mathbf{y}_n, \hat{\Theta}^{(t)}\right)$
- **end for**
- $\tilde{\sigma}^2 \leftarrow \frac{1}{Np}\sum_{n=1}^{N} \mathbb{E}\Big(\mathcal{L}_q\left(\mathbf{y}_n, \mathbf{X}_n; \mathbf{a}_n\right)|\mathbf{y}_n, \hat{\Theta}^{(t)}\Big)$. (18)
- $\tilde{\mathbf{M}} \leftarrow \frac{1}{N}\sum_{n=1}^{N} \mathbb{E}\left(\mathbf{X}_n|\mathbf{y}_n, \hat{\Theta}^{(t)}\right)$.
- $$\left(\tilde{\mathbf{Q}}^{(k)}\right) \leftarrow \frac{1}{N}\sum_{n=1}^{N} \mathbb{E}\left([\mathbf{X}_n]_k [\mathbf{X}_n]_k^T \,|\mathbf{y}_n, \hat{\Theta}^{(t)}\right) - \hat{\mathbf{m}}_k^{(t+1)}\left(\hat{\mathbf{m}}_k^{(t+1)}\right)^T, \quad k = 1, .., K.$$
- $\tilde{\mathbf{Q}}^{(k)} \leftarrow (1-\alpha_k)\tilde{\mathbf{Q}}^{(k)} + \alpha_k \frac{\mathrm{Tr}(\tilde{\mathbf{Q}}^{(k)})}{p}\mathbf{I}, \quad k = 1, ...K$
  α_k given in (24)-(25).
- For $k = 1, .., K$, if $\tilde{\mathbf{m}}_k$ has negative entries, do Alt-Min (algorithm 1).
- $t \leftarrow t + 1$.
- $\hat{\Theta}^{(t)} = \{\tilde{\mathbf{M}}, \tilde{\mathcal{Q}}, \tilde{\mathbf{A}}, \tilde{\sigma}^2\}$.

End until
Return $\hat{\Theta}^{(t)} = \{\hat{\mathbf{M}}^{(t)}, \hat{\mathcal{Q}}^{(t)}, \hat{\mathbf{A}}^{(t)}, (\hat{\sigma}^2)^{(t)}\}$.

C. Initialization points

Due to the complexity of K-Gaussians and the non-convexity of the DNCM model, we require good initial matrices of means and covariance, initial abundance for each pixel and independent noise variance σ^2. Following [19], we propose using the VCA estimate as an initial point. The VCA algorithm returns a matrix of endmember means, and the abundance is obtained either by multiplying the pseudoinverse of the latter with the data matrix (see [22]), or applying the FCLS algorithm [29], imposing simplex constraints. We propose using identity matrices as initial covariances. Another option is the GMM estimate, which also returns the covariance of every component. However, in this paper we shall focus on VCA initialization.

D. Diagonal DNCM

A special case of the DNCM is that of diagonal covariances, in which the K-Gaussians can be made more

efficient, and that is for two reasons. First, matrix inversion is accomplished through inverting each of the diagonal entries. Second, the term $\mathbb{E}\left(\mathbf{X}^T\mathbf{X}|\mathbf{y},\hat{\mathbf{\Theta}}\right)$ admits a closed form, which is of course much faster than computing traces of K^2 matrices, as in (12). In this subsection, we will shortly derive this expression.

We can write the term $\mathrm{Tr}\Big\{ [\hat{\mathbf{a}}]_i [\hat{\mathbf{a}}]_j \hat{\mathbf{Q}}^{(i)} \left[\hat{\mathbf{Q}}(\hat{\mathbf{a}}) + \hat{\sigma}^2\mathbf{I}\right]^{-1} \hat{\mathbf{Q}}^{(j)}\Big\}$ from (13) as:

$$[\hat{\mathbf{a}}]_i [\hat{\mathbf{a}}]_j \begin{bmatrix}\hat{q}_1^{(i)} \cdots \hat{q}_p^{(i)}\end{bmatrix} \left[\hat{\mathbf{Q}}(\hat{\mathbf{a}}) + \hat{\sigma}^2\mathbf{I}\right]^{-1} \begin{bmatrix}\hat{q}_1^{(j)} \\ \vdots \\ \hat{q}_p^{(j)}\end{bmatrix},$$

where $\hat{q}_\ell^{(i)}$ is the ℓ-th diagonal entry of $\hat{\mathbf{Q}}^{(i)}$. Since each covariance $\hat{\mathbf{Q}}^{(i)}$ of past iteration is diagonal, the matrix of traces $\mathrm{Tr}\{\hat{\mathbf{Z}}^{ij}\}$ can be written as:

$$\mathbf{D}_1 - \mathbf{D}_2\mathbf{D}_3 \left[\hat{\mathbf{Q}}(\hat{\mathbf{a}}) + \hat{\sigma}^2\mathbf{I}\right]^{-1} \mathbf{D}_3^T\mathbf{D}_2$$

where:

$$\mathbf{D}_1 = \begin{bmatrix}\mathrm{Tr}\{\hat{\mathbf{Q}}^{(1)}\} & & \\ & \ddots & \\ & & \mathrm{Tr}\{\hat{\mathbf{Q}}^{(K)}\}\end{bmatrix}$$

$$\mathbf{D}_2 = \begin{bmatrix}[\hat{\mathbf{a}}]_1 & & \\ & \ddots & \\ & & [\hat{\mathbf{a}}]_K\end{bmatrix}$$

$$\mathbf{D}_3 = \begin{bmatrix}\hat{q}_1^{(1)} & \cdots & \hat{q}_p^{(1)} \\ \vdots & \ddots & \vdots \\ \hat{q}_1^{(K)} & \cdots & \hat{q}_p^{(K)}\end{bmatrix},$$

i.e., the i-th row of $\mathbf{D}_3$ is the diagonal of $\hat{\mathbf{Q}}^{(i)}$. Altogether, (12) becomes:

$$\left[\mathbb{E}\left(\mathbf{X}^T\mathbf{X}|\mathbf{y},\hat{\Theta}\right)\right] = \left[\mathbb{E}\left(\mathbf{X}|\mathbf{y},\hat{\Theta}\right)^T \mathbb{E}\left(\mathbf{X}|\mathbf{y},\hat{\Theta}\right)\right] + \mathbf{D}_1 - \mathbf{D}_2\mathbf{D}_3 \left[\hat{\mathbf{Q}}(\hat{\mathbf{a}}) + \hat{\sigma}^2\mathbf{I}\right]^{-1} \mathbf{D}_3^T\mathbf{D}_2. \quad (26)$$

IV. Numerical Experiments

In this section, we use numerical experiments to validate the DNCM, and test the performance of the proposed K-Gaussians algorithm, as well as its complexity, in comparison to other methods. We provide results of four experiments, ranging from synthetic simulations to a completely real world setting.

In the first synthetic experiment we use $K = 3$ components, and consider the normalized mean square error (NMSE) based on the Frobenius norm as our means of evaluation. Specifically, for a given true endmembers matrix $\mathbf{M}$ and its estimate $\hat{\mathbf{M}}$, we define:

$$\mathrm{NMSE}(\hat{\mathbf{M}}) = \frac{\|\hat{\mathbf{M}} - \mathbf{M}\|_{\mathrm{F}}^2}{\|\mathbf{M}\|_{\mathrm{F}}^2}.$$

Similar NMSEs are defined for the endmember covariances and the abundances. We note that the order of $\hat{\mathbf{M}}$'s columns might not match that of $\mathbf{M}$. For that reason we compute the permutation of $\hat{\mathbf{M}}$'s columns that yields the minimal $\mathrm{MSE}(\hat{\mathbf{M}})$. This permutation is then also applied to the covariances and abundances.

We compare five algorithms and modify them to output estimates of the unknown means, covariances and abundances. To that end we examine the following methods:

- K-Gaussians: as defined previously. Following [19], the algorithm is initialized by VCA estimates. The initial covariances are set as $0.01 \times \mathbf{I}$, and $\sigma_0^2 = 10^{-4}$. The algorithm has four variants, with covariances that can be either full or diagonal, and abundances that can be non-negative or simplex. We run the algorithm using 4-workers parallel pool.
- GMM: returns estimates of the cluster means and covariances. We set the abundance estimates as the posterior probability of each pixel, which are mostly 1-sparse vectors. The model assumes no independent noise ($\sigma_0^2 = 0$).
- VCA [22]: returns only estimates of the endmember means. We compute the abundances using the FCLS algorithm [29], so that they satisfy non-negativity and sum to one constraints.
- NMF [5]: similar to VCA. Because the solution is invariant to scaling, i.e. $\hat{\mathbf{M}}\hat{\mathbf{A}} = \frac{1}{\alpha}\hat{\mathbf{M}}\alpha\hat{\mathbf{A}}$, for $\alpha > 0$, we select α that results in the best NMSE for $\hat{\mathbf{M}}$.
- UsGNCM [19]: endmembers estimate initialized by VCA. Following the original code provided by the authors, unless mentioned otherwise we assume that the number of spatial regions is equal to the number of components, and set the initial variance as 0.001. The number of iterations is $12,000$, out of which $11,000$ are burn-in iterations.

A. Synthetic data

The first experiments are synthetic and their goal is to analyze the performance of the K-Gaussians algorithm in an exact DNCM setting. To ensure fairness, we follow the guidelines proposed in [19].

The first synthetic experiment simulates the exact model used in [19] for testing their UsGNCM algorithm. The image consists of 50×50 pixels with 200 bands. A 3-label map was constructed using the Potts model. The abundance map was generated by using a different truncated Dirichlet distribution for each label (with parameters $(15, 15, 1)$, $(1, 8, 8)$ and $(3, 1, 3)$). The exact parameters are not available for reproduction, so we approximate the label map, endmembers means and diagonal covariances, as shown in Fig. 1. We generate a random vector of length 200 for each pixel, by calculating the convex combination of three multivariate normal random vectors, corresponding to the three endmembers (coefficients given by the abundance map), and adding independent noise of variance 10^{-5}.

The errors of this experiment are presented in table I. This synthetic model was specifically designed by the authors of [19] to demonstrate the advantages of UsGNCM. Since these settings meet the GNCM assumptions, such as diagonal covariances and Potts model for label map, the UsGNCM leads to the smallest errors. Using 30 iterations, our proposed K-Gaussians methods results in an improvement of the VCA initialization and comparable errors, with significantly

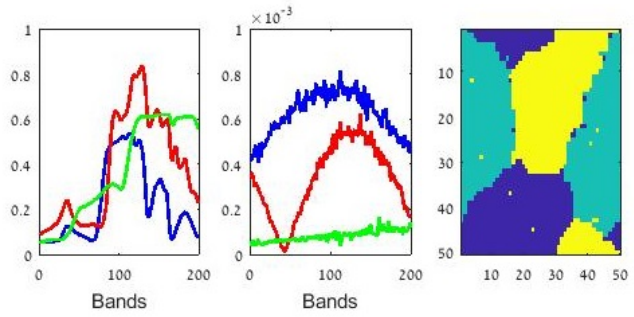

Fig. 1: Settings of the first synthetic dataset, following [19]. Left: Endmember means. Center: diagonal entries of the three corresponding covariance matrices. Right: label map.

smaller computational complexity (5-6 minutes compared to an hour).

Algorithm	MSE($\hat{\mathbf{M}}$)	MSE($\hat{\mathbf{Q}}$)	MSE($\hat{\mathbf{A}}$)	time (sec)
GMM	0.0755	18.1685	1.6846	-
K-Gaussians	0.0018	0.0643	1.0001	250.5
VCA	0.0036	—	1.0831	-
NMF	0.1728	—	0.9298	-
UsGNCM	**0.00001**	**0.0033**	**0.8997**	11,018

TABLE I: MSE results for the first synthetic experiment, with running time of K-Gaussians and UsGNCM (right column).

In the second experiment, we continue to follow the same guidelines but change some of the parameters to test the robustness of UsGNCM to its modelling assumptions. Specifically, we use the same endmember means but switch to full covariances. Since estimation requires now a larger dataset, we use a 133×133 image. We use an abundance map that has been extracted from a real multispectral image [1], using simplex constraint least squares on estimated endmember means (see Fig. 2). Endmember variability is illustrated in Fig. 3, plotting $5,000$ samples drawn from the multivariate normal distribution, given true mean and covariance of each endmember. Performing PCA on the data, we note that the first three principal components account for 99.78% of data variability, suggesting that it lies in a lower dimensional space, a common property of HSI's [30].

The errors are reported in table II. It is easy to observe the advantages of K-Gaussians over UsGNCM (that assumes diagonal covariances). K-Gaussians is slightly more accurate and significantly faster. Abundance and covariance estimates of the K-Gaussians are shown in Fig. 4. We note the difference between the estimated covariances and the true ones (bottom of Fig. 2), especially the middle and left components, where the estimated left covariance has larger diagonal values. Interestingly, the best algorithm in terms of abundance estimation is UsGNCM. The estimated abundance maps for the other algorithms: VCA, UsGNCM and GMM are shown in Fig. 5.

[1]A Sentinel 2 multispectral image of the Copernicus earth observation program. Downloaded from : `https://scihub.copernicus.eu/dhus/#/home`. File name is: *S2A_MSIL2A_20171023T105111_N0206_R051_T30TXK_20171023T131219.*

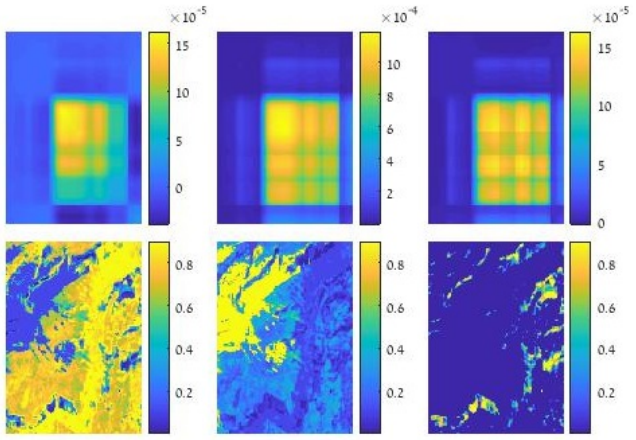

Fig. 2: Settings of the second synthetic experiment. Top row: sample covariance matrices of three endmembers. Bottom row: abundance map of the corresponding components.

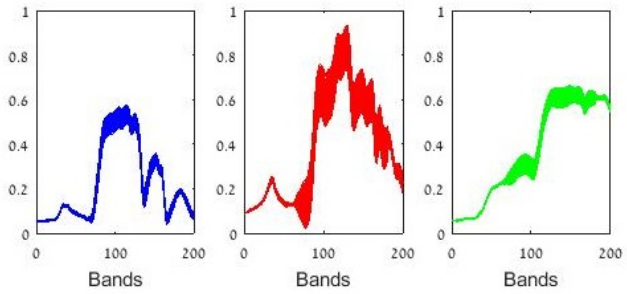

Fig. 3: Plots of $5,000$ samples for each endmember, given true mean and covariance from the second synthetic experiment.

Algorithm	MSE($\hat{\mathbf{M}}$)	MSE($\hat{\mathbf{Q}}$)	MSE($\hat{\mathbf{A}}$)	time (sec)
GMM	0.0543	71.9780	0.8507	-
K-Gaussians	**0.0019**	**0.1795**	0.3661	3.92×10^3
VCA	0.0182	—	0.5483	-
NMF	0.2958	—	0.9748	-
UsGNCM	0.0223	1.0052	**0.2621**	84.6×10^3

TABLE II: MSE results for the second synthetic experiment, with running time of K-Gaussians and UsGNCM (right column).

In an additional experiment we use 10 components, extracting 34-dimensional endmember means from AVIRIS image over Cuprite [31] (see Fig. 6), and using random diagonal covariances. The abundance map consists of 200×200 pixels, each of Dirichlet distribution with unit parameters. As previously, the independent noise variance is 10^{-5}. When running UsGNCM, we assume one spatial class. We run 200 iterations of K-Gaussians. In table III we provide the error results. As expected, we find a clear advantage to using UsGNCM, since the settings follow the GNCM assumptions. However, K-Gaussians improve upon VCA initialization, and is comparable to UsGNCM, obtaining these results

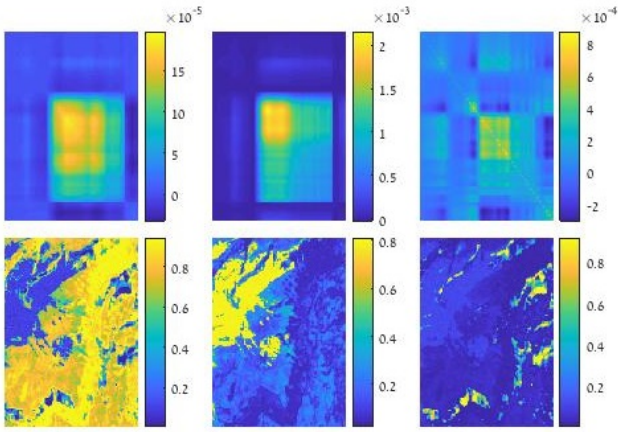

Fig. 4: Covariance and abundance estimates of each component using the K-Gaussians on the second synthetic experiments.

within shorter running time. We note that a larger number of components imposes more challenge to K-Gaussians regarding choice of initialization point. We leave that issue for future research.

Algorithm	MSE($\hat{\mathbf{M}}$)	MSE($\hat{\mathbf{Q}}$)	MSE($\hat{\mathbf{A}}$)	time (sec)
GMM	0.0949	3.0203	5.7957	-
K-Gaussians	0.0263	0.3859	0.8133	8,773
VCA	0.0322	—	2.6718	-
NMF	0.3200	—	0.9530	-
UsGNCM	**0.0022**	**0.0137**	**0.7954**	60,818

TABLE III: MSE results for the third synthetic experiment, with running time of K-Gaussians and UsGNCM (right column).

B. Code complexity

The main motivation to DNCM was to reduce the complexity of Bayesian NCM-based algorithms. To analyze the computational complexity, we run a synthetic experiment, examining the running time of the K-Gaussians and UsGNCM, on a 3-component data, with increasing number of dimensions (bands) and sample size (N). We run the UsGNCM using $12,000$ iterations. We use a desktop computer with a Intel Core i5-3470 CPU and 7.88 GB of total physical memory. We run K-Gaussians without parallelization. Results are shown in tables IV-V, for K-Gaussians and UsGNCM, respectively. Note that we do not include the GMM, VCA and NMF, as these algorithms are fast and run up to a few seconds. The MSE of the K-Gaussians estimates in this experiment are comparable, and even smaller than the UsGNCM. Running time of both algorithms seem to linearly depend on N and the number of bands, with a larger coefficient for the UsGNCM.

C. Hyperspectral Target detection

Finally, in the third experiment we demonstrate the use of DNCM and K-Gaussians using a real hyperspectral

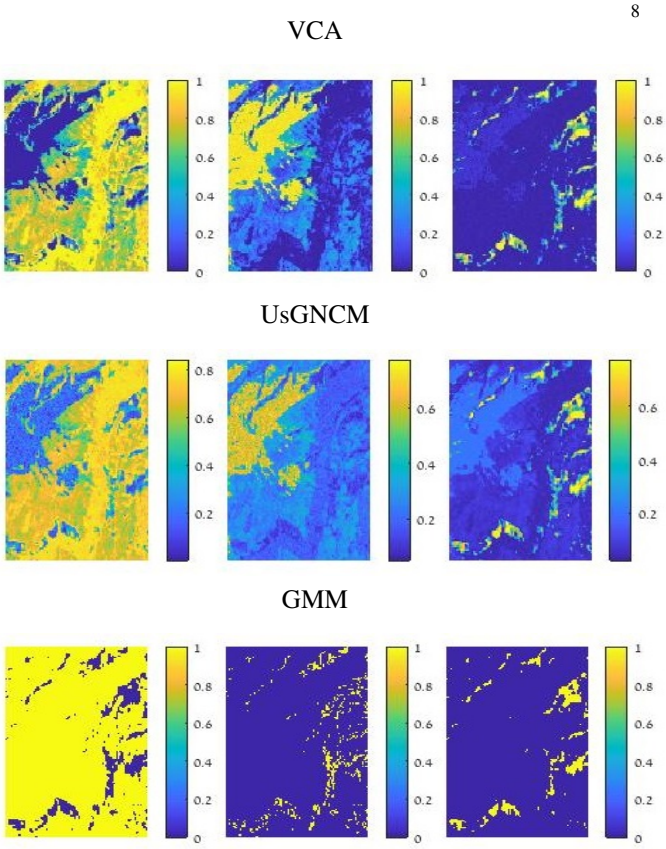

Fig. 5: Abundance estimates of each component of VCA, UsGNCM and GMM on the second synthetic experiments.

	K-Gaussians (time in seconds)		
	$N = 529$	$N = 1024$	$N = 1521$
bands=10	88.30	171.05	250.51
bands=20	98.52	182.72	260.80
bands=30	97.49	187.82	275.73

TABLE IV: Running time of K-Gaussians, for three components.

	UsGNCM (time in seconds)		
	$N = 529$	$N = 1024$	$N = 1521$
bands=10	373.81	590.57	802.55
bands=20	434.00	668.58	898.22
bands=30	478.98	746.69	1004.79

TABLE V: Running time of GNCM, for three components.

image. The dataset is from SHARE2012, an imaging data collection campaign taken near Rochester, NY [21][2]. The image contains known targets of red and blue felt materials, which were placed in the actual field, and their locations are also known. The HSI image (its RGB bands), as well as target location and spectra, are shown in Fig. 7. Before we turn to the detection task, we begin by demonstrating the modelling capabilities of our approach.

Technical details on SHARE2012 and algorithm param-

[2]Data can be downloaded from `https://www.rit.edu/cos/share2012/`

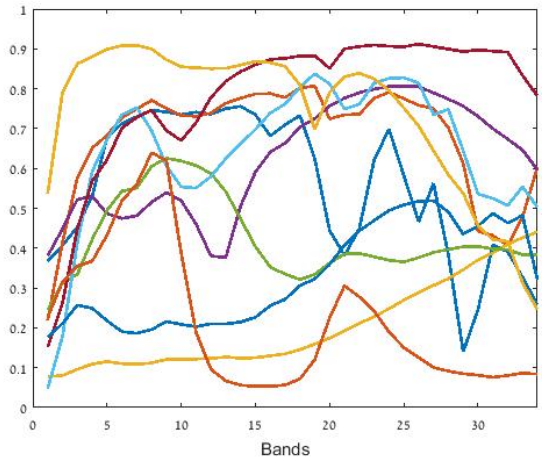

Fig. 6: Means of ten endmembers for the third synthetic dataset.

eters: Following similar size to that of [32], we use a 301×301 sub-image which is sufficiently large to include all targets. Each pixel is a 320×1 vector of spectral data, of wavelengths ranging from 0.4 to 2.45 micrometers. Our data is therefore a tensor of size $301 \times 301 \times 320$. As in [32], all algorithms assume $K = 6$ endmembers. Note that SHARE2012 is a very large dataset and therefore we had to constrain some of the algorithms. GMM and VCA are highly scalable and they run until convergence. Same is the NMF but it performs badly, so these results are not reported. K-Gaussians is limited to 15 iterations which take roughly 30 hours. Due to time and memory constraints, we limit the number of UsGNCM iterations to $2,700$, using the last 225 as burn-in iterations. This led to 45 hours of computation. The estimated abundance maps of each algorithm among VCA, K-Gaussians, UsGNCM and GMM are shown in Fig. 9, along with the corresponding endmembers means and variances. Note that the K-Gaussians endmembers look similar to those of VCA, up to scaling factor, while UsGCNM differs significantly. On the other hand, the UsGNCM and K-Gaussians abundance maps differ, with VCA allowing negative values.

The main goal of all the algorithms is to characterize the statistics of the background as accurately as possible. All algorithms model the spectral content within the n'th pixel as a multivariate Gaussian with unknown mean $\mathbf{m}_n$ and covariance $\mathbf{Q}_n$. The goal is therefore to estimate these parameters. GMM estimates them directly. VCA estimates only the means and we modify it to output covariance matrices as well. The NCM-based algorithms estimate the parameters of the global endmembers and the local abundances. These are then combined to output the local parameters at each pixel:

$$\begin{aligned} \mathbf{m}_n &\leftarrow \hat{\mathbf{M}}\hat{\mathbf{a}}_n, \\ \mathbf{Q}_n &\leftarrow \sum_{k=1}^{K} [\hat{\mathbf{a}}_n]_k^2 \, \hat{\mathbf{Q}}^{(k)} + \hat{\sigma}^2 \mathbf{I}. \end{aligned} \tag{27}$$

We begin comparing the modelling capabilities of the different algorithms in terms of their likelihood. Unfortunately, we have no ground truth of the true statistics, and therefore we perform this comparison only in a few manually cropped patches which we identify as pure regions, namely trees, shadow and pavement (see Fig. 7). Thus, one can assume Gaussian distribution on every patch, with the likelihood value given estimated parameters reflecting the model fit.

Within these patches, we compute the negative log likelihood of the multivariate normal associated with the different estimated parameters. More specifically, we use the K-Gaussians estimates to compute the mean and covariance of each pixel via (27). We then compute their average to obtain mean and covariance for the whole patch. We then compute the likelihood of all pixels in a patch using the latter parameters. We repeat this procedure on GMM and VCA estimates. Since VCA does not return endmember covariance, we use a spherical matrix whose entries are the mean error of each pixel. The results are presented in Table VI below. Showing the negative log likelihood, a lower value indicates a better fit. The selected patches are pure and as expected, GMM (which also assumes pure regions) provides the highest likelihoods. The negative likelihood obtained from VCA estimate is usually large, while K-Gaussians succeeds to get close to GMM. Nevertheless, from Fig. 9 we note the similarities between the estimates of VCA and K-Gaussians. The likelihood of UsGNCM is lower than that of the VCA, but is quite high relative to K-Gaussians. One reason might be its limited number of iterations.

	negative log likelihood ($\times 10^4$)			
	grass	trees	pavement	shadow
GMM	-0.2034	-0.2023	-0.2012	-0.2072
K-Gaussians	-0.1076	-0.0815	-0.0657	-0.1234
VCA	0.3335	5.5716	0.3057	0.7141
UsGNCM	-0.0608	0.6212	0.0738	0.6294

TABLE VI: Likelihood value of different patches from Fig. 7, given estimates of the various algorithms.

Motivated by this initial exploration analysis, we turn to the target detection task for which this campaign was designed for [32], [33]. Each of the algorithms estimates the multivariate normal parameters differently and these lead to different detectors which we compare via their ROC curves. It is important to emphasize that although the presence of a very few targets in a large image is a typical setting in real situations, these experiment results should be taken with some caution, as we deal with one image and estimating the ROC is somewhat problematic. Yet this experiment serves as a qualitative assessment of the validity of the background estimation model as part of a bigger application (target detection). This demonstrates the expressive power of the suggested model, and its advantage in applying it to real-world problems. Having stressed that, we now turn to the technical details.

All our detectors are based on the classical matched filter (MF) detector [34], [35]. It assumes

$$\mathbf{y} = a\mathbf{s} + \mathbf{w}, \quad \mathbf{w} \sim \mathcal{N}(\mathbf{m}_n, \mathbf{Q}_n), \tag{28}$$

and is designed to test the hypothesis:

$$\begin{aligned} \mathcal{H}_0 &: a = 0 \\ \mathcal{H}_1 &: a > 0, \end{aligned}$$

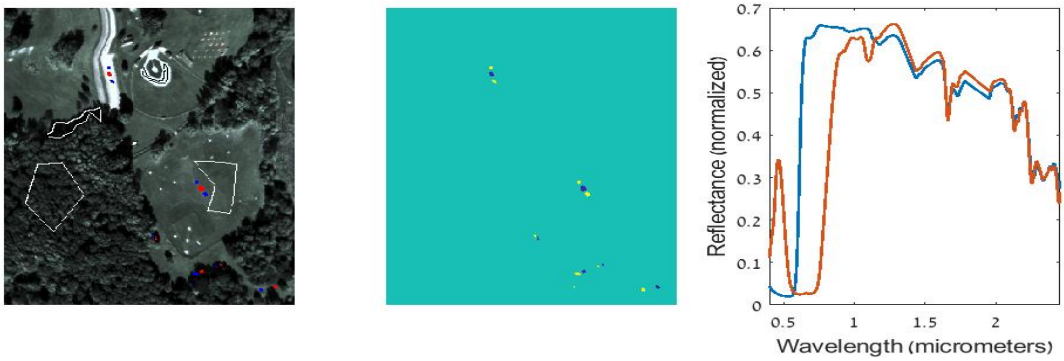

Fig. 7: Left: RGB field image of SHARE 2012 data, with patches corresponding to results from table VI. Middle: Location of the targets in the image; the targets are of two types (”blue” and ”red”), each type is marked by a different color. Right: spectral signature of the two target types.

This leads to the following detector:

$$MF = \frac{\mathbf{s}^T \mathbf{Q}_n^{-1}(\mathbf{y}_n - \mathbf{m}_n)}{\mathbf{s}^T \mathbf{Q}_n^{-1}\mathbf{s}} \lessgtr \tau. \tag{29}$$

Since we assume a spherical covariance matrix for VCA, $\mathbf{Q}_n^{-1}$ is canceled out in the above formula, and therefore it depends only on $\mathbf{y}_n$ and $\mathbf{M}_n$. We add as a baseline the random detector, which reject the null hypothesis at random with probability of $\frac{1}{2}$. We also consider the global estimator, which assumes that the full image is normally distributed. Therefore, sample mean and covariances are computed, and these are plugged in the MF (29).

In Fig. 8 we present two sets of ROC curves, each for a different target as illustrated in Fig. 7, by plotting the log false positive rate against detection rate for every threshold value of τ. It is easy to observe the advantages of K-Gaussians over the rest of the detectors. Comparing it to a similar experiment which examines a manifold-learning based algorithm and as well uses blue/red felt targets [33], most algorithms in Fig. 8 outperform the latter. For instance, with a log false positive rate of 10^{-1}, detection rate of 0.3-0.4 is achieved by their algorithm on both targets. The main reason for the gap is that smaller sub-images were used in their experiment.

As can be seen in the figures, UsGNCM is quite comparable to the VCA, and even improves upon the latter. Moreover, UsGNCM is significantly better than both GMM and the global detector. Recall however that the number of UsGNCM iterations is limited to $2,700$ due to time and computational constraints. Note that all detectors perform better in detecting the first target than the second (left and right plots of Fig. 8, respectively). This is due to the nature of the target. However, rating the various algorithms by their performance, the order is roughly the same. As to the UsGNCM, we believe that if given a sufficient number of iterations, the ROC curve of UsGNCM will turn competitive to that of K-Gaussians. To conclude, with limited computational resources, target detection results demonstrate the advantage of using K-Gaussians over the rest of the algorithms, and whose results are achieved by roughly $2/3$ the running time of that of UsGNCM.

V. Summary

Parameter estimation of signals generated from a mixture of sources is a fundamental problem, with various applications such as HSI. In this paper, we introduced the DNCM, which allows for measurements that are convex combinations of Gaussians, and is a variant of the classical NCM model. This leads to a simple and intuitive model formulation, which can be easily seen as an extension of the GMM and NMF. Importantly, our deterministic method which we call K-Gaussians, is composed of relatively simple steps and results in a scalable estimation algorithm. Numerical experiments demonstrate that K-Gaussians estimates improve upon VCA initialization, allowing for full covariances, and is comparable to competitive methods such as UsGNCM, achieved at a relatively shorter running time. However, the DNCM suffers from a few potential weaknesses, namely high dimensionality, non-linearity and non-convexity.

On the practical side, one open issue is formulating a model where endmembers are not statistically independent, as it is the case in most real life data. Another issue is the case of unknown component number, and whether we can incorporate that in the model.

Another drawback which should be addressed is that our experience suggests that K-Gaussians is sensitive to its initialization and may be attracted to a nearby local minimum. Along a different axis, the K-Gaussians can be further generalized by assuming colored independent noise. We leave all these questions for investigation in future research.

Acknowledgement

Many thanks to the anonymous reviewers and the editor, whose comments have significantly helped to improve this manuscript. The authors would also like to thank Nicolas Dobigeon for helpful discussions and for providing some datasets. Many thanks to Michael Braverman and Barak Nakash for preparing the data and providing the detection code. Part of this research was supported by ISF grant 1339/15.

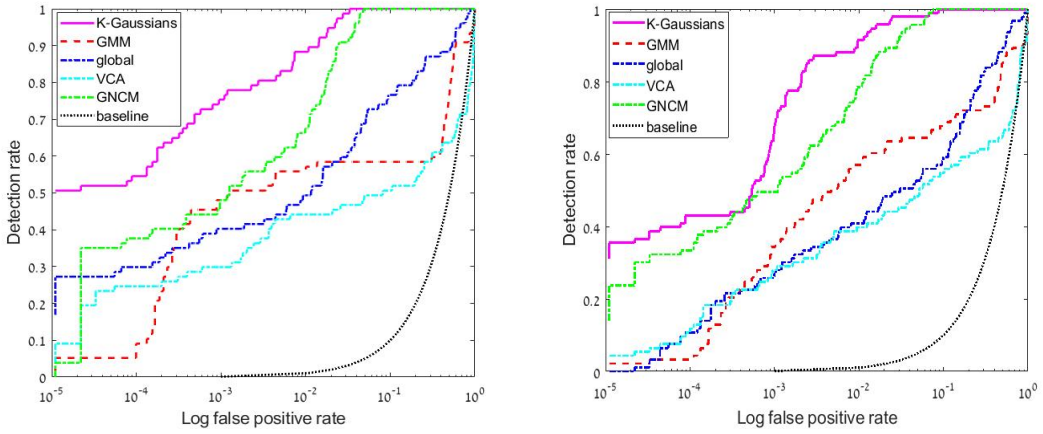

Fig. 8: ROC curves of the first target (right) and second target (left) from right panel of Fig. 7, on SHARE 2012 data, using MF detector.

APPENDIX: DERIVATION OF THE EM STEPS

Recall that all columns of $\mathbf{X}$ are instances of K endmember distributions, so that the prior probability of $\mathbf{x}_k$ is

$$\mathbf{x}_k \sim \mathcal{N}\left(\mathbf{m}_k, \hat{\mathbf{Q}}^{(k)}\right). \tag{30}$$

Because both $\mathbf{x}_k$ and $\mathbf{y}$ are Gaussian, which are linearly related through

$$\mathbf{y} = \mathbf{X}\hat{\mathbf{a}} + \hat{\boldsymbol{\epsilon}}, \quad \hat{\boldsymbol{\epsilon}} \sim \mathcal{N}(\mathbf{0}, \hat{\sigma}^2\mathbf{I}). \tag{31}$$

The stacked vector $\begin{bmatrix}\mathbf{x}_k \\ \mathbf{X}\hat{\mathbf{a}} + \hat{\boldsymbol{\epsilon}}\end{bmatrix}$ is normally distributed, with mean $\begin{bmatrix}\hat{\mathbf{m}}_k \\ \hat{\mathbf{M}}\hat{\mathbf{A}}\end{bmatrix}$ and covariance $\begin{bmatrix}\mathbf{Q}^{(k)} & \hat{a}_k\mathbf{Q}^{(k)} \\ \hat{a}_k\mathbf{Q}^{(k)} & \hat{\mathbf{Q}}(\hat{\mathbf{a}}) + \hat{\sigma}^2\mathbf{I}\end{bmatrix}$ (note that different columns of $\mathbf{X}$ are independent). Using that, and the rule for Gaussian conditional expectation/covariance, we arrive at equation (11), and the conditional covariance given in (16).

It remains to derive $\mathbb{E}\left(\mathbf{X}^T\mathbf{X}|\mathbf{y}, \hat{\Theta}\right)$. Notice that

$$\left[\mathbf{X}^T\mathbf{X}\right]_{ij} = \sum_{\ell=1}^{p} \left[\mathbf{x}_i\right]_\ell \left[\mathbf{x}_j\right]_\ell. \tag{32}$$

We can write the expectation of each term in the sum as:

$$\begin{aligned}\mathbb{E}\left(\left[\mathbf{x}_i\right]_\ell \left[\mathbf{x}_j\right]_\ell \middle| \mathbf{y}, \hat{\Theta}\right) = \\ \underbrace{\mathbb{E}\left(\left[\mathbf{x}_i\right]_\ell \middle| \mathbf{y}, \hat{\Theta}\right)\mathbb{E}\left(\left[\mathbf{x}_j\right]_\ell \middle| \mathbf{y}, \hat{\Theta}\right)}_{(I)} + \\ \underbrace{\mathrm{Cov}\left(\left[\mathbf{x}_i\right]_\ell, \left[\mathbf{x}_j\right]_\ell \middle| \mathbf{y}, \hat{\Theta}\right)}_{(II)}.\end{aligned} \tag{33}$$

We use (11) to compute the term (I) in (33), as $\mathbb{E}\left(\left[\mathbf{x}_i\right]_\ell | \mathbf{y}, \hat{\Theta}\right)$ is the ℓ-th element of $\mathbb{E}\left(\mathbf{x}_i | \mathbf{y}, \hat{\Theta}\right)$ from (11). Turning now to (II), we suppose that $i \neq j$, where $\mathbf{x}_i$ and $\mathbf{x}_j$ are statistically independent (from this case we will be able derive an expression also when $i = j$). Given $\hat{\Theta}$, the stacked vector $\left[\left[\mathbf{x}_i\right]_\ell, \left[\mathbf{x}_j\right]_\ell, \mathbf{y}^T\right]^T$ has the following distribution:

$$\begin{bmatrix}\left[\mathbf{x}_i\right]_\ell \\ \left[\mathbf{x}_j\right]_\ell \\ \mathbf{y}\end{bmatrix} \middle| \hat{\Theta} \sim \mathcal{N}\left(\begin{bmatrix}\left[\hat{\mathbf{m}}_i\right]_\ell \\ \left[\hat{\mathbf{m}}_j\right]_\ell \\ \hat{\mathbf{M}}\hat{\mathbf{a}}\end{bmatrix}, \mathbf{S}\right),$$

where

$$\mathbf{S} = \left(\begin{array}{cc|c} \hat{Q}^{(i)}_{\ell,\ell} & 0 & \hat{a}_i\hat{Q}^{(i)}_{\ell,1} \cdots \hat{a}_i\hat{Q}^{(i)}_{\ell,p} \\ 0 & \hat{Q}^{(j)}_{\ell,\ell} & \hat{a}_j\hat{Q}^{(j)}_{\ell,1} \cdots \hat{a}_j\hat{Q}^{(j)}_{\ell,p} \\ \hline \hat{a}_i\hat{Q}^{(i)}_{\ell,1} & \hat{a}_j\hat{Q}^{(j)}_{\ell,1} & \\ \vdots & \vdots & \hat{\mathbf{Q}}(\hat{\mathbf{a}}) + \hat{\sigma}^2\mathbf{I} \\ \hat{a}_i\hat{Q}^{(i)}_{\ell,p} & \hat{a}_j\hat{Q}^{(j)}_{\ell,p} & \end{array}\right). \tag{34}$$

Recall that we assume the linear model (31), where the k-th column of $\mathbf{X}$ is distributed according to (30), which explains the above expectation and the covariance diagonal blocks. As to the off diagonal blocks, we simply obtain that from $\mathrm{Cov}\left(\left[\mathbf{x}_i\right]_\ell, \mathbf{X}\hat{\mathbf{a}} + \hat{\boldsymbol{\epsilon}}\right)$ and the fact that $\mathbf{x}_i$, $\mathbf{x}_j$ are independent. To be clear, in (34) we denote the (m,n)-th entry of $\hat{Q}^{(i)}$ by $\hat{Q}^{(i)}_{m,n}$. Using again the conditional covariance rule of Gaussian vectors, we get:

$$\mathrm{Cov}\left(\begin{bmatrix}\left[\mathbf{x}_i\right]_\ell \\ \left[\mathbf{x}_j\right]_\ell\end{bmatrix} \middle| \mathbf{y}, \hat{\Theta}\right) = \mathbf{S}_{1,1} - \mathbf{S}_{1,2}\left(\hat{\mathbf{Q}}(\hat{\mathbf{a}}) + \hat{\sigma}^2\mathbf{I}\right)^{-1}\mathbf{S}_{2,1},$$

with $\mathbf{S}_{1,1}, \mathbf{S}_{1,2}, \mathbf{S}_{2,1}$ as the corresponding blocks in (34). Plugging in the blocks in (34), the $(2,1)$ and $(1,1)$ entries of the above covariance, corresponding to the cases $i \neq j$ and $i = j$ respectively, are:

$$\begin{aligned}&\mathrm{Cov}\left(\left[\mathbf{x}_i\right]_\ell, \left[\mathbf{x}_j\right]_\ell | \mathbf{y}, \hat{\Theta}\right) = \\ &-\hat{a}_i\hat{a}_j\left[\hat{\mathbf{q}}^{(i)}_\ell\right]^T\left(\hat{\mathbf{Q}}(\hat{\mathbf{a}}) + \hat{\sigma}^2\mathbf{I}\right)^{-1}\hat{\mathbf{q}}^{(j)}_\ell, \quad i \neq j, \\ &\hat{Q}^{(i)}_{\ell,\ell} - \hat{a}_i^2\left[\hat{\mathbf{q}}^{(i)}_\ell\right]^T\left(\hat{\mathbf{Q}}(\hat{\mathbf{a}}) + \hat{\sigma}^2\mathbf{I}\right)^{-1}\hat{\mathbf{q}}^{(i)}_\ell, \quad i = j\end{aligned} \tag{35}$$

where $\hat{\mathbf{q}}^{(i)}_\ell$ is the ℓ-th column of $\hat{\mathbf{Q}}^{(i)}$. We therefore have derived an expression for (II) in (33).

In order to obtain an expression for (32), we need to sum the covariances in (35) over ℓ. From these equations, one can verify that $\sum_{\ell=1}^{p}\mathrm{Cov}\left(\left[\mathbf{x}_i\right]_\ell, \left[\mathbf{x}_j\right]_\ell | \mathbf{y}, \hat{\Theta}\right)$ is equivalent to the trace of $\hat{\mathbf{Z}}^{ij}$ defined in (13).

Altogether, from (32),(33) and the last results, we arrive

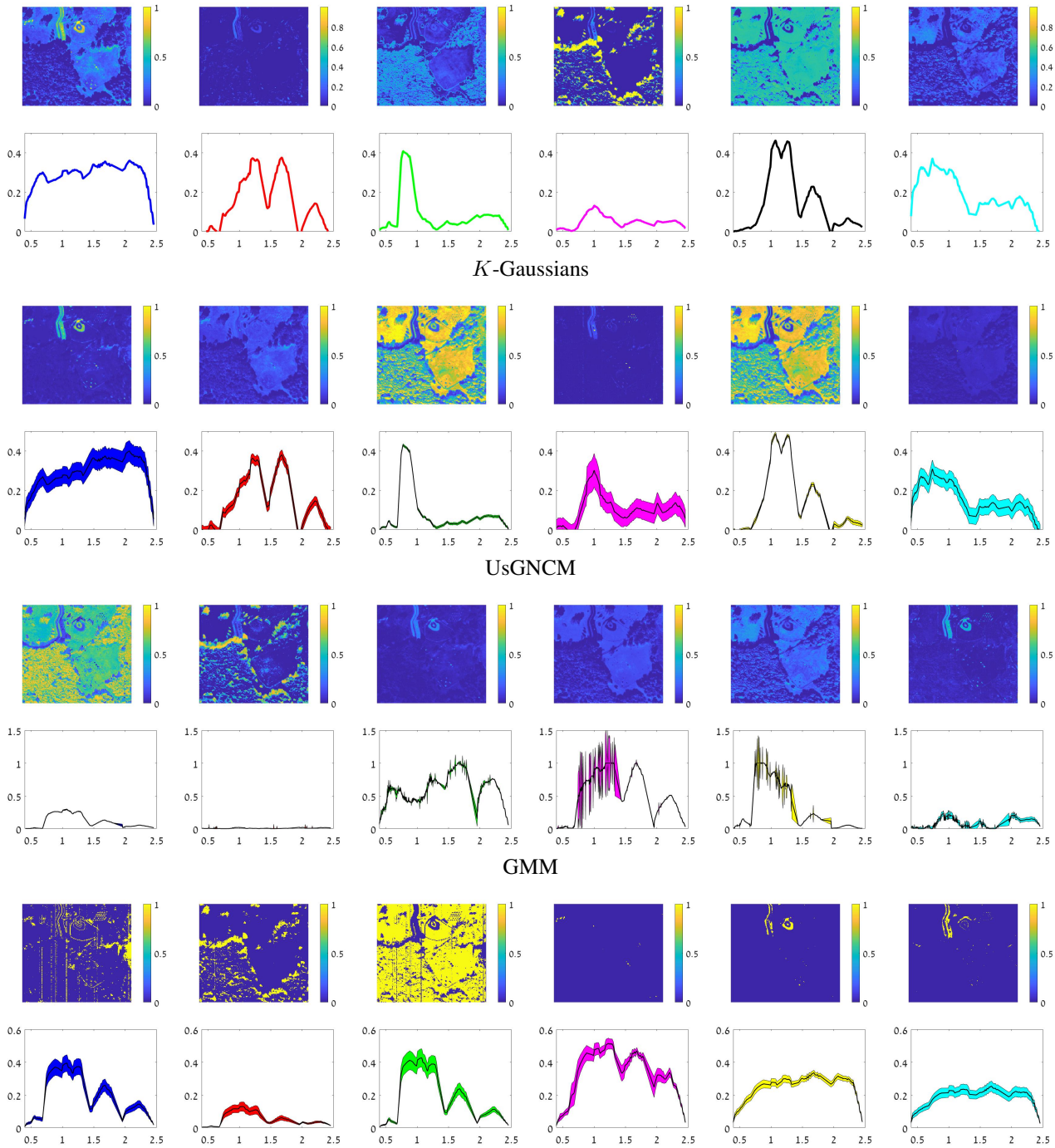

Fig. 9: SHARE 2012 experiment: estimated abundances (first row) and their corresponding endmember means $\pm 0.5 \times$ std. (second row, colored area between graphs) of 6 components, for each algorithm: VCA, K-Gaussians, UsGNCM and GMM.

at (12):

$$\left[\mathbb{E}\left(\mathbf{X}^T\mathbf{X}|\mathbf{y},\hat{\Theta}\right)\right]_{ij} = \left[\mathbb{E}\left(\mathbf{X}|\mathbf{y},\hat{\Theta}\right)^T \mathbb{E}\left(\mathbf{X}|\mathbf{y},\hat{\Theta}\right)\right]_{ij} + \mathrm{Tr}\left\{\hat{\mathbf{Z}}^{ij}\right\},$$

Derivation of (20)-(21). Using the definition of $\mathcal{L}_g$ in (8), the objective of (19) can be written as:

$$N\log|\mathbf{Q}^{(k)}| + \mathbb{E}\left(\sum_{n=1}^{N}\left([\mathbf{X}_n]_k - \mathbf{m}_k\right)^T\left(\mathbf{Q}^{(k)}\right)^{-1}\left([\mathbf{X}_n]_k - \mathbf{m}_k\right)\middle|\mathbf{y},\hat{\boldsymbol{\Theta}}\right). \quad (36)$$

Considering $\mathbf{m}_k$, it can be easily confirmed that the gradient of (36) vanishes at $\frac{1}{N}\sum_{n=1}^{N}\mathbb{E}\left([\mathbf{X}_n]_k|\mathbf{y}_n,\hat{\boldsymbol{\Theta}}\right)$, and therefore $\tilde{\mathbf{m}}_k$ in (20) is the minimizer. Now, plugging in $\tilde{\mathbf{m}}_k$, (36) can

be written as:

$$N\log|\mathbf{Q}^{(k)}| + \mathrm{Tr}\Big\{ \left(\mathbf{Q}^{(k)}\right)^{-1} \mathbf{B} \Big\},$$

where

$$\mathbf{B} = \mathbb{E}\left(\sum_{n=1}^{N} \left([\mathbf{X}_n]_k - \tilde{\mathbf{m}}_k\right) \left([\mathbf{X}_n]_k - \tilde{\mathbf{m}}_k\right)^T \middle| \mathbf{y}, \hat{\mathbf{\Theta}} \right).$$

Taking derivative over $\mathbf{Q}^{(k)}$, we arrive at the term:

$$N \left(\mathbf{Q}^{(k)}\right)^{-1} - \left(\mathbf{Q}^{(k)}\right)^{-1} \mathbf{B} \left(\mathbf{Q}^{(k)}\right)^{-1},$$

which vanishes at $\mathbf{Q}^{(k)} = \frac{1}{N}\mathbf{B}$. One can then use the result on $\tilde{\mathbf{m}}_k$ from (20) to simplify this term and get (21).

REFERENCES

[1] G. A. Shaw and H. K. Burke, "Spectral imaging for remote sensing," *Lincoln laboratory journal*, vol. 14, no. 1, pp. 3–28, 2003.

[2] J. M. Bioucas-Dias, A. Plaza, N. Dobigeon, M. Parente, Q. Du, P. Gader, and J. Chanussot, "Hyperspectral unmixing overview: Geometrical, statistical, and sparse regression-based approaches," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 5, no. 2, pp. 354–379, 2012.

[3] D. Stein, "Application of the normal compositional model to the analysis of hyperspectral imagery," in *Advances in techniques for analysis of remotely sensed data, 2003 IEEE Workshop on*, pp. 44–51, IEEE, 2003.

[4] N. Keshava and J. F. Mustard, "Spectral unmixing," *IEEE signal processing magazine*, vol. 19, no. 1, pp. 44–57, 2002.

[5] D. D. Lee and H. S. Seung, "Algorithms for non-negative matrix factorization," in *Advances in neural information processing systems*, pp. 556–562, 2001.

[6] N. Gillis, "The why and how of nonnegative matrix factorization," *Regularization, Optimization, Kernels, and Support Vector Machines*, vol. 12, no. 257, 2014.

[7] Z. Yang, G. Zhou, S. Xie, S. Ding, J.-M. Yang, and J. Zhang, "Blind spectral unmixing based on sparse nonnegative matrix factorization," *IEEE Transactions on Image Processing*, vol. 20, no. 4, pp. 1112–1125, 2011.

[8] A. Zare and K. Ho, "Endmember variability in hyperspectral analysis: Addressing spectral variability during spectral unmixing," *IEEE Signal Processing Magazine*, vol. 31, no. 1, pp. 95–104, 2014.

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

[10] M. T. Eismann and R. C. Hardie, "Application of the stochastic mixing model to hyperspectral resolution enhancement," *IEEE transactions on geoscience and remote sensing*, vol. 42, no. 9, pp. 1924–1933, 2004.

[11] M. T. Eismann and D. Stein, "Stochastic mixture modeling," *Hyperspectral Data Exploitation: Theory and Applications*, vol. 148, 2007.

[12] X. Du, A. Zare, P. Gader, and D. Dranishnikov, "Spatial and spectral unmixing using the beta compositional model," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 7, no. 6, pp. 1994–2003, 2014.

[13] Y. Zhou, A. Rangarajan, and P. D. Gader, "A gaussian mixture model representation of endmember variability in hyperspectral unmixing," *IEEE Transactions on Image Processing*, vol. 27, no. 5, pp. 2242–2256, 2018.

[14] A. Zare, P. Gader, and G. Casella, "Sampling piecewise convex unmixing and endmember extraction," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 51, no. 3, pp. 1655–1665, 2013.

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

[16] B. Zhang, L. Zhuang, L. Gao, W. Luo, Q. Ran, and Q. Du, "Psoem: A hyperspectral unmixing algorithm based on normal compositional model," *IEEE Transactions on Geoscience and Remote sensing*, vol. 52, no. 12, pp. 7782–7792, 2014.

[17] O. Eches, N. Dobigeon, C. Mailhes, and J.-Y. Tourneret, "Bayesian estimation of linear mixtures using the normal compositional model. application to hyperspectral imagery," *IEEE Transactions on Image Processing*, vol. 19, no. 6, pp. 1403–1413, 2010.

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

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

[20] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the em algorithm," *Journal of the royal statistical society. Series B (methodological)*, pp. 1–38, 1977.

[21] A. Giannandrea, N. Raqueno, D. W. Messinger, J. Faulring, J. P. Kerekes, J. van Aardt, K. Canham, S. Hagstrom, E. Ontiveros, A. Gerace, *et al.*, "The share 2012 data campaign," in *Proc. SPIE*, vol. 8743, p. 87430F, 2013.

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

[23] D. M. Titterington, A. F. Smith, and U. E. Makov, *Statistical analysis of finite mixture distributions*. Wiley,, 1985.

[24] S. Lloyd, "Least squares quantization in pcm," *IEEE transactions on information theory*, vol. 28, no. 2, pp. 129–137, 1982.

[25] G. McLachlan and T. Krishnan, *The EM algorithm and extensions*, vol. 382. John Wiley & Sons, 2007.

[26] S. Boyd and L. Vandenberghe, *Convex optimization*. Cambridge university press, 2004.

[27] O. Ledoit and M. Wolf, "Honey, i shrunk the sample covariance matrix," *The Journal of Portfolio Management*, vol. 30, no. 4, pp. 110–119, 2004.

[28] A. Wiesel, T. Zhang, *et al.*, "Structured robust covariance estimation," *Foundations and Trends® in Signal Processing*, vol. 8, no. 3, pp. 127–216, 2015.

[29] D. C. Heinz *et al.*, "Fully constrained least squares linear spectral mixture analysis method for material quantification in hyperspectral imagery," *IEEE transactions on geoscience and remote sensing*, vol. 39, no. 3, pp. 529–545, 2001.

[30] J. Li, J. M. Bioucas-Dias, A. J. Plaza, and L. Liu, "Robust collaborative nonnegative matrix factorization for hyperspectral unmixing.," *IEEE Trans. Geoscience and Remote Sensing*, vol. 54, no. 10, pp. 6076–6090, 2016.

[31] G. Vane, R. O. Green, T. G. Chrien, H. T. Enmark, E. G. Hansen, and W. M. Porter, "The airborne visible/infrared imaging spectrometer (aviris)," *Remote sensing of environment*, vol. 44, no. 2-3, pp. 127–143, 1993.

[32] C. Jiao and A. Zare, "Functions of multiple instances for learning target signatures," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 53, no. 8, pp. 4670–4686, 2015.

[33] A. K. Ziemann, J. Theiler, and D. W. Messinger, "Hyperspectral target detection using manifold learning and multiple target spectra," in *Applied Imagery Pattern Recognition Workshop (AIPR), 2015 IEEE*, pp. 1–7, IEEE, 2015.

[34] D. Manolakis and G. Shaw, "Detection algorithms for hyperspectral imaging applications," *IEEE signal processing magazine*, vol. 19, no. 1, pp. 29–43, 2002.

[35] D. Manolakis, R. Lockwood, T. Cooley, and J. Jacobson, "Is there a best hyperspectral detection algorithm?," The International Society for Optical Engineering, 2009.