STOCHASTIC ML SIMPLEX-STRUCTURED MATRIX FACTORIZATION UNDER THE DIRICHLET MIXTURE MODEL

Ruiyuan Wu[†]*, Qiang Li*[‡]*, and Wing-Kin Ma*[†]

[†]Department of Electronic Engineering, The Chinese University of Hong Kong, Shatin, N.T., Hong Kong
[‡]School of Info. & Comm. Eng., University of Electronic Science and Technology of China, China

ABSTRACT

Simplex-structured matrix factorization (SSMF) is a problem of recovering a basis matrix and the corresponding coefficient vectors from data, where the coefficient vectors are constrained to lie in the unit simplex. SSMF has attracted growing attention in recent years, with numerous applications such as hyperspectral unmixing and document clustering. In this work, we develop a maximum-likelihood (ML) approach for SSMF. Specifically, by modeling the coefficient vectors as random variables following a Dirichlet mixture distribution—which allows us to model more complex data distributions in real-life data, a probabilistic model for SSMF is employed. We consider a marginalized likelihood with respect to the coefficient vectors, and use ML estimation to learn the basis matrix and unknown Dirichlet mixture parameters. The marginalized likelihood does not admit a closed form and is non-concave, and this makes the problem challenging to solve. To handle this challenge, an effective algorithm using sample average approximation and block successive upper-bound minimization is proposed. We consider the aforementioned two real-world applications by simulations. Numerical results show that the proposed algorithm delivers appealing performance in both applications.

Index Terms— Simplex-structured matrix factorization, Dirichlet mixture model, maximum-likelihood estimation, sample average approximation, block successive upper-bound minimization

1. INTRODUCTION

Structured matrix factorization (SMF), which aims at decomposing a data matrix into factor matrices with certain prescribed structures, is powerful in retrieving physically meaningful latent information from the data. SMF is a long-standing problem that can be traced back to classical topics such as eigendecomposition and singular value decomposition. In the past two decades, intrigued by new factorization models, there has been renewed interest in SMF. For instance, non-negative matrix factorization (NMF), arguably one of the most successful SMF models, has been extensively applied in many different areas [1–4]. More recently, another kind of SMF model called *simplex-SMF (SSMF)*, which restricts every column of one factor matrix (i.e., the coefficient vector) to lie within the unit simplex, has attracted increasing attention owing to its various applications in signal processing and machine learning, such as hyperspectral unmixing [5,6], document clustering [3,7], video summarization [8], blind separation of speech sources [9], to name just a few. In addition, an NMF problem can be transformed to an SSMF problem under appropriate data pre-processing [4]. Apart from the applications, SSMF also exhibits appealing theoretical identifiability guarantees. In particular, under some mild assumptions, the factor matrices of SSMF can be uniquely identified up to some trivial ambiguities [4]—this is crucial in many estimation and recovery applications.

In this paper, we consider SSMF with an emphasis on the algorithm design aspect. There are numerous approaches proposed for SSMF, such as pure-pixel search (or near-separable NMF and self-dictionary sparse regression), simplex volume minimization, and Bayesian methods [5–7]. Here, we resort to a stochastic maximum-likelihood (ML) framework, which is relatively less studied in the literature. To explain, we employ a probabilistic model to describe SSMF, where the coefficient vectors are modeled as random variables with a Dirichlet mixture distribution. Accordingly, by marginalizing these variables, we obtain the likelihood function of our ML formulation. In previous works, the stochastic ML was considered for the noise-free case [10] and the uniform Dirichlet case [11]. The Dirichlet mixture model herein (with noise) is more general, and it may better represent data in real world. On the other hand, the resulting ML problem is harder to tackle—the likelihood function cannot be expressed in closed form due to the marginalization, and it is non-concave. With a similar flavor of the algorithm proposed in [11], a combination of stochastic optimization, alternating optimization and majorization-minimization is proposed to tackle the ML problem. In simulations, we test the proposed algorithm under two real applications, namely, hyperspectral unmixing and document clustering. The numerical results demonstrate that our approach delivers very promising performance in both applications.

Our notations are standard. For a matrix $\boldsymbol{X}$, $\boldsymbol{X}^\dagger$ denotes its pseudo-inverse. For a vector $\boldsymbol{x} \in \mathbb{R}^n$, $\boldsymbol{x} \geq \boldsymbol{0}$ means that $\boldsymbol{x}$ is elementwise nonnegative; $\boldsymbol{x}^2 = [x_1^2, \ldots, x_n^2]^T$ denotes the elementwise square operation; both $[\boldsymbol{x}]_i$ and x_i denote the ith element of $\boldsymbol{x}$; $\mathrm{Diag}(\boldsymbol{x}) \in \mathbb{R}^{n\times n}$ is a diagonal matrix with the diagonal vector being $\boldsymbol{x}$. For a scalar $x > 0$, $\Gamma(x)$ is the gamma function; $\Psi(x) = \frac{\mathrm{d}\log\Gamma(x)}{\mathrm{d}x}$ is the digamma function [12]. Also, we use $\mathcal{N}(\boldsymbol{x};\boldsymbol{\mu},\boldsymbol{\Sigma}) \triangleq \frac{1}{\sqrt{(2\pi)^M \det(\boldsymbol{\Sigma})}} e^{-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu})^T\boldsymbol{\Sigma}^{-1}(\boldsymbol{x}-\boldsymbol{\mu})}$ to denote the probability density function (p.d.f.) of an M-dimensional Gaussian random variable $\boldsymbol{x}$ with mean $\boldsymbol{\mu}$ and covariance $\boldsymbol{\Sigma}$.

2. PROBLEM STATEMENT

Consider the following observation model:

$$\boldsymbol{y}_\ell = \boldsymbol{A}\boldsymbol{s}_\ell + \boldsymbol{v}_\ell, \;\; \ell = 1, \ldots, L, \tag{1}$$

where $\boldsymbol{y}_\ell \in \mathbb{R}^M$ is the ℓth observed data vector; $\boldsymbol{A} \in \mathbb{R}^{M\times N}$ is a basis matrix; $\boldsymbol{s}_\ell$ is the coefficient vector for $\boldsymbol{y}_\ell$; and $\boldsymbol{v}_\ell \in \mathbb{R}^M$ is

This work was supported by a General Research Fund (GRF) of the Research Grant Council (RGC), Hong Kong, under Project ID CUHK 14205717.

noise. We assume that every $\boldsymbol{s}_\ell$ lies in the unit simplex; specifically, $\boldsymbol{s}_\ell \in \mathcal{U}_N \triangleq \{\boldsymbol{s} \in \mathbb{R}^N | \mathbf{1}^T\boldsymbol{s} = 1, \boldsymbol{s} \geq \mathbf{0}\}$. The model (1) can be written in a matrix form as $\boldsymbol{Y} = \boldsymbol{A}\boldsymbol{S}+\boldsymbol{V}$, where $\boldsymbol{Y} = [\boldsymbol{y}_1, \ldots, \boldsymbol{y}_L]$, $\boldsymbol{S} = [\boldsymbol{s}_1, \ldots, \boldsymbol{s}_L]$ and $\boldsymbol{V} = [\boldsymbol{v}_1, \ldots, \boldsymbol{v}_L]$. Our problem is to retrieve $\boldsymbol{A}$ and $\boldsymbol{S}$ from the data matrix $\boldsymbol{Y}$.

We resort to a stochastic ML approach. To describe it, we start with the following statistical assumptions:

(A1) The noise vectors $\{\boldsymbol{v}_\ell\}_{\ell=1}^L$ follow independently and identically distributed (i.i.d.) Gaussian distribution with mean zero and covariance $\mathrm{Diag}(\boldsymbol{\sigma})$, where $\boldsymbol{\sigma} > \mathbf{0}$.

(A2) The coefficient vectors $\{\boldsymbol{s}_\ell\}_{\ell=1}^L$ follow an i.i.d. Dirichlet mixture distribution

$$\boldsymbol{s}_\ell \sim q(\boldsymbol{s}_\ell; \boldsymbol{\Theta}, \boldsymbol{\gamma}) = \sum_{p=1}^P \gamma_p \mathcal{D}(\boldsymbol{s}_\ell; \boldsymbol{\theta}_p),$$

where P is the number of Dirichlet mixtures, $\boldsymbol{\gamma} \in \mathcal{U}_P$ is the mixing weight vector, $\boldsymbol{\Theta} = [\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_P] \in \mathbb{R}_{++}^{N\times P}$, and

$$\mathcal{D}(\boldsymbol{s}; \boldsymbol{\theta}) = \begin{cases} \frac{\Gamma(\sum_{n=1}^N \theta_n)}{\prod_{n=1}^N \Gamma(\theta_n)} \prod_{n=1}^N s_n^{\theta_n - 1}, & \boldsymbol{s} \in \mathcal{U}_N \\ 0, & \boldsymbol{s} \notin \mathcal{U}_N \end{cases}$$

denotes the Dirichlet p.d.f. with concentration parameter $\boldsymbol{\theta}$.

We should point out that the Dirichlet mixture model is a reasonable choice—it intrinsically enforces the unit simplex constraint on $\boldsymbol{s}_\ell$'s, and it is capable of modeling complicated data distributions [10, 12]. Let $(\boldsymbol{A}, \boldsymbol{\sigma}, \boldsymbol{\Theta}, \boldsymbol{\gamma})$ be the parameter to be estimated. Under (A1)-(A2), the p.d.f. of $\boldsymbol{y}_\ell$ conditioned on $\boldsymbol{s}_\ell$ is given by

$$h(\boldsymbol{y}_\ell | \boldsymbol{s}_\ell; \boldsymbol{A}, \boldsymbol{\sigma}) = \mathcal{N}(\boldsymbol{y}_\ell; \boldsymbol{A}\boldsymbol{s}_\ell, \mathrm{Diag}(\boldsymbol{\sigma})),$$

and the likelihood function of $\{\boldsymbol{y}_\ell\}_{\ell=1}^L$ is $\prod_{\ell=1}^L p(\boldsymbol{y}_\ell; \boldsymbol{A}, \boldsymbol{\sigma}, \boldsymbol{\Theta}, \boldsymbol{\gamma})$, where

$$p(\boldsymbol{y}_\ell; \boldsymbol{A}, \boldsymbol{\sigma}, \boldsymbol{\Theta}, \boldsymbol{\gamma}) = \int_{\mathcal{U}_N} h(\boldsymbol{y}_\ell | \boldsymbol{s}_\ell; \boldsymbol{A}, \boldsymbol{\sigma}) q(\boldsymbol{s}_\ell; \boldsymbol{\Theta}, \boldsymbol{\gamma}) \mathrm{d}\boldsymbol{s}_\ell \quad (2)$$

is the marginalized p.d.f. of $\boldsymbol{y}_\ell$. We are interested in finding the $(\boldsymbol{A}, \boldsymbol{\sigma}, \boldsymbol{\Theta}, \boldsymbol{\gamma})$ that maximizes the likelihood function, or equivalently, a solution to the following problem

$$\min_{\substack{\boldsymbol{A}\in\mathbb{R}^{M\times N}, \boldsymbol{\sigma}\in\mathbb{R}_{++}^M \\ \boldsymbol{\Theta}\in\mathbb{R}_{++}^{M\times P}, \boldsymbol{\gamma}\in\mathcal{U}_P}} -\sum_{\ell=1}^L \log(p(\boldsymbol{y}_\ell; \boldsymbol{A}, \boldsymbol{\sigma}, \boldsymbol{\Theta}, \boldsymbol{\gamma})). \quad (3)$$

It should be noted that the stochastic ML formulation derived above has $\boldsymbol{S}$ marginalized, rather than treating $\boldsymbol{S}$ as a variable to be estimated. Once we solve (3) and obtain $\boldsymbol{A}$, we can retrieve $\boldsymbol{S}$ by solving an inverse problem. However, the ML problem (3) is challenging. First, the calculation of the likelihood function in (2) involves a multi-dimensional integral over the unit simplex, which has no analytic form in general. Second, even if $p(\boldsymbol{y}_\ell; \boldsymbol{A}, \boldsymbol{\sigma}, \boldsymbol{\Theta}, \boldsymbol{\gamma})$ could be analytically expressed, the objective function is still non-convex. In the ensuing section, we will propose our method to handle problem (3).

3. THE PROPOSED SOLUTION

Our approach relies on two key techniques, namely, *sample average approximation (SAA)* and *block successive upper-bound minimization (BSUM)*. The former is employed to tackle the intractable integrals in (2), and the latter is a technique for non-convex optimization.

3.1. SAA of Problem (3)

We employ a sample average strategy called *importance sampling* [13] to approximately evaluate the integral in (2). Let $\mu_\ell(\boldsymbol{s}_\ell)$ be some pre-specified sampling distribution for $\boldsymbol{s}_\ell$. Importance sampling approximates $p(\boldsymbol{y}_\ell; \boldsymbol{A}, \boldsymbol{\sigma}, \boldsymbol{\Theta}, \boldsymbol{\gamma})$ by

$$\begin{aligned} p(\boldsymbol{y}_\ell; \boldsymbol{A}, \boldsymbol{\sigma}, \boldsymbol{\Theta}, \boldsymbol{\gamma}) &= \int_{\mathcal{U}_N} \frac{h(\boldsymbol{y}_\ell | \boldsymbol{s}_\ell; \boldsymbol{A}, \boldsymbol{\sigma}) q(\boldsymbol{s}_\ell; \boldsymbol{\Theta}, \boldsymbol{\gamma})}{\mu_\ell(\boldsymbol{s}_\ell)} \mu_\ell(\boldsymbol{s}_\ell) \mathrm{d}\boldsymbol{s}_\ell \\ &\approx \frac{1}{R} \sum_{r=1}^R \frac{h(\boldsymbol{y}_\ell | \boldsymbol{\xi}_\ell^r; \boldsymbol{A}, \boldsymbol{\sigma}) q(\boldsymbol{\xi}_\ell^r; \boldsymbol{\Theta}, \boldsymbol{\gamma})}{\mu_\ell(\boldsymbol{\xi}_\ell^r)}, \end{aligned} \quad (4)$$

where $\boldsymbol{\xi}_\ell^1, \ldots, \boldsymbol{\xi}_\ell^R$ are R samples sampled from $\mu_\ell(\boldsymbol{s}_\ell)$. The reasons for using $\mu_\ell(\boldsymbol{s}_\ell)$ as the sampling distribution are twofold: Firstly, it is impossible to directly sample from the true prior $q(\boldsymbol{s}_\ell; \boldsymbol{\Theta}, \boldsymbol{\gamma})$, since $q(\boldsymbol{s}_\ell; \boldsymbol{\Theta}, \boldsymbol{\gamma})$ depends on the unknown variables $(\boldsymbol{\Theta}, \boldsymbol{\gamma})$. Secondly, with a carefully selected $\mu_\ell(\boldsymbol{s}_\ell)$, a reliable approximation can be obtained with limited samples [13], thereby improving the sampling efficiency. This is the crux of importance sampling, and we will discuss how to choose $\mu_\ell(\boldsymbol{s}_\ell)$ in details in Sec.3.3 after we present the whole algorithm.

Plugging the above approximation into (3), we obtain:

$$\min_{\substack{\boldsymbol{A}\in\mathbb{R}^{M\times N}, \boldsymbol{\sigma}\in\mathbb{R}_{++}^M \\ \boldsymbol{\Theta}\in\mathbb{R}_{++}^{M\times P}, \boldsymbol{\gamma}\in\mathcal{U}_P}} -\sum_{\ell=1}^L \log\left(\sum_{r=1}^R \frac{h(\boldsymbol{y}_\ell | \boldsymbol{\xi}_\ell^r; \boldsymbol{A}, \boldsymbol{\sigma}) q(\boldsymbol{\xi}_\ell^r; \boldsymbol{\Theta}, \boldsymbol{\gamma})}{R\mu_\ell(\boldsymbol{\xi}_\ell^r)}\right). \quad (5)$$

The above strategy is referred to as SAA [14] in stochastic optimization. Under some mild regularity conditions, the SAA solution converges to that of the original problem (with high probability) as the sample size increases [14]. In the sequel, we will focus on problem (5).

3.2. BSUM for Problem (5)

It is difficult to simultaneously optimize $(\boldsymbol{A}, \boldsymbol{\sigma}, \boldsymbol{\Theta}, \boldsymbol{\gamma})$ in problem (5). Thus, we employ an alternating optimization scheme called BSUM [15], which cyclically optimizes $(\boldsymbol{A}, \boldsymbol{\sigma})$ and $(\boldsymbol{\Theta}, \boldsymbol{\gamma})$ by majorization-minimization (MM) [16]. To explain, a function $g(\boldsymbol{x}; \bar{\boldsymbol{x}})$ is called a *majorizer* of a function $f(\boldsymbol{x})$ at $\bar{\boldsymbol{x}}$ if, for all $\boldsymbol{x}$, it holds that

$$f(\boldsymbol{x}) \leq g(\boldsymbol{x}; \bar{\boldsymbol{x}}), \;\; f(\bar{\boldsymbol{x}}) = g(\bar{\boldsymbol{x}}; \bar{\boldsymbol{x}}).$$

The BSUM algorithm incorporates the idea of MM into the update of each block, so that each block may be more efficiently computed. Readers are referred to [15] for details. Back to problem (5), denote its objective function as $f(\boldsymbol{A}, \boldsymbol{\sigma}, \boldsymbol{\Theta}, \boldsymbol{\gamma})$. Let $f_{\bar{\boldsymbol{\Theta}}, \bar{\boldsymbol{\gamma}}}(\boldsymbol{A}, \boldsymbol{\sigma}) = f(\boldsymbol{A}, \boldsymbol{\sigma}, \bar{\boldsymbol{\Theta}}, \bar{\boldsymbol{\gamma}})$ and $f_{\bar{\boldsymbol{A}}, \bar{\boldsymbol{\sigma}}}(\boldsymbol{\Theta}, \boldsymbol{\gamma}) = f(\bar{\boldsymbol{A}}, \bar{\boldsymbol{\sigma}}, \boldsymbol{\Theta}, \boldsymbol{\gamma})$. Applying BSUM to problem (5) amounts to recursively performing the following updates:

$$(\boldsymbol{A}^{k+1}, \boldsymbol{\sigma}^{k+1}) = \arg \min_{\substack{\boldsymbol{A}\in\mathbb{R}^{M\times N} \\ \boldsymbol{\sigma}\in\mathbb{R}_{++}^M}} g_{\boldsymbol{\Theta}^k, \boldsymbol{\gamma}^k}(\boldsymbol{A}, \boldsymbol{\sigma}; \boldsymbol{A}^k, \boldsymbol{\sigma}^k), \quad (6a)$$

$$(\boldsymbol{\Theta}^{k+1}, \boldsymbol{\gamma}^{k+1}) = \arg \min_{\substack{\boldsymbol{\Theta}\in\mathbb{R}_{++}^{M\times P} \\ \boldsymbol{\gamma}\in\mathcal{U}_P}} g_{\boldsymbol{A}^{k+1}, \boldsymbol{\sigma}^{k+1}}(\boldsymbol{\Theta}, \boldsymbol{\gamma}; \boldsymbol{\Theta}^k, \boldsymbol{\gamma}^k), \quad (6b)$$

where $g_{\boldsymbol{\Theta}^k, \boldsymbol{\gamma}^k}(\boldsymbol{A}, \boldsymbol{\sigma}; \boldsymbol{A}^k, \boldsymbol{\sigma}^k)$ and $g_{\boldsymbol{A}^{k+1}, \boldsymbol{\sigma}^{k+1}}(\boldsymbol{\Theta}, \boldsymbol{\gamma}; \boldsymbol{\Theta}^k, \boldsymbol{\gamma}^k)$ are majorizers of $f_{\boldsymbol{\Theta}^k, \boldsymbol{\gamma}^k}(\boldsymbol{A}, \boldsymbol{\sigma})$ and $f_{\boldsymbol{A}^{k+1}, \boldsymbol{\sigma}^{k+1}}(\boldsymbol{\Theta}, \boldsymbol{\gamma})$, respectively (resp.). In the following, we will elaborate on how to choose $g_{\boldsymbol{\Theta}^k, \boldsymbol{\gamma}^k}(\boldsymbol{A}, \boldsymbol{\sigma}; \boldsymbol{A}^k, \boldsymbol{\sigma}^k)$ and $g_{\boldsymbol{A}^{k+1}, \boldsymbol{\sigma}^{k+1}}(\boldsymbol{\Theta}, \boldsymbol{\gamma}; \boldsymbol{\Theta}^k, \boldsymbol{\gamma}^k)$ so that problems (6a) and (6b) can be solved in closed form. For

notational convenience, we temporarily use $(\bar{\boldsymbol{A}}, \bar{\boldsymbol{\sigma}}, \bar{\boldsymbol{\Theta}}, \bar{\boldsymbol{\gamma}})$ [resp. $(\boldsymbol{A}^+, \boldsymbol{\sigma}^+, \boldsymbol{\Theta}^+, \boldsymbol{\gamma}^+)$] to denote the last [resp. next] estimation of $(\boldsymbol{A}, \boldsymbol{\sigma}, \boldsymbol{\Theta}, \boldsymbol{\gamma})$.

3.2.1. Problem (6a) *w.r.t.* $(\boldsymbol{A}, \boldsymbol{\sigma})$.

The following lemma provides us with an easy-to-optimize majorizer for $f_{\bar{\boldsymbol{\Theta}},\bar{\boldsymbol{\gamma}}}(\boldsymbol{A}, \boldsymbol{\sigma})$:

Lemma 1 *Denote*

$$g_{\bar{\boldsymbol{\Theta}},\bar{\boldsymbol{\gamma}}}(\boldsymbol{A}, \boldsymbol{\sigma}; \bar{\boldsymbol{A}}, \bar{\boldsymbol{\sigma}}) = -\sum_{\ell=1}^{L}\sum_{r=1}^{R} \alpha_\ell^r \log\left(\frac{h(\boldsymbol{y}_\ell|\boldsymbol{\xi}_\ell^r; \boldsymbol{A}, \boldsymbol{\sigma})}{\alpha_\ell^r R \mu_\ell(\boldsymbol{\xi}_\ell^r)/q(\boldsymbol{\xi}_\ell^r; \bar{\boldsymbol{\Theta}}, \bar{\boldsymbol{\gamma}})}\right)$$

where $\alpha_\ell^r = \frac{q(\boldsymbol{\xi}_\ell^r;\bar{\boldsymbol{\Theta}},\bar{\boldsymbol{\gamma}})h(\boldsymbol{y}_\ell|\boldsymbol{\xi}_\ell^r;\bar{\boldsymbol{A}},\bar{\boldsymbol{\sigma}})/R\mu_\ell(\boldsymbol{\xi}_\ell^r)}{\sum_{\tilde{r}=1}^{R} q(\boldsymbol{\xi}_\ell^{\tilde{r}};\bar{\boldsymbol{\Theta}},\bar{\boldsymbol{\gamma}})h(\boldsymbol{y}_\ell|\boldsymbol{\xi}_\ell^{\tilde{r}};\bar{\boldsymbol{A}},\bar{\boldsymbol{\sigma}})/R\mu_\ell(\boldsymbol{\xi}_\ell^{\tilde{r}})}$. *The function* $g_{\bar{\boldsymbol{\Theta}},\bar{\boldsymbol{\gamma}}}(\boldsymbol{A}, \boldsymbol{\sigma}; \bar{\boldsymbol{A}}, \bar{\boldsymbol{\sigma}})$ *is a majorizer of* $f_{\bar{\boldsymbol{\Theta}},\bar{\boldsymbol{\gamma}}}(\boldsymbol{A}, \boldsymbol{\sigma})$.

Lemma 1 can be easily shown by using Jensen's inequality; we skip the proof. Invoking Lemma 1, problem (6a) becomes

$$\min_{\substack{\boldsymbol{A}\in\mathbb{R}^{M\times N}\\ \boldsymbol{\sigma}\in\mathbb{R}^{M}_{++}}} \sum_{\ell=1}^{L}\sum_{r=1}^{R} \alpha_\ell^r \left(\sum_{m=1}^{M} \log(\sigma_m) + \sigma_m^{-1}[\boldsymbol{y}_\ell - \boldsymbol{A}\boldsymbol{\xi}_\ell^r]_m^2\right).$$

It can be shown that the optimal $(\boldsymbol{A}, \boldsymbol{\sigma})$ is given by

$$\boldsymbol{A}^+ = \left[\sum_{\ell=1}^{L} \boldsymbol{y}_\ell \left(\sum_{r=1}^{R} \alpha_\ell^r \boldsymbol{\xi}_\ell^r\right)^T\right]\left[\sum_{\ell=1}^{L}\sum_{r=1}^{R} \alpha_\ell^r \boldsymbol{\xi}_\ell^r (\boldsymbol{\xi}_\ell^r)^T\right]^\dagger,$$

$$\boldsymbol{\sigma}^+ = \frac{1}{L}\sum_{\ell=1}^{L}\sum_{r=1}^{R} \alpha_\ell^r (\boldsymbol{y}_\ell - \boldsymbol{A}^+\boldsymbol{\xi}_\ell^r)^2.$$

3.2.2. Problem (6b) *w.r.t.* $(\boldsymbol{\Theta}, \boldsymbol{\gamma})$.

We introduce the following majorizer for $f_{\bar{\boldsymbol{A}},\bar{\boldsymbol{\sigma}}}(\boldsymbol{\Theta}, \boldsymbol{\gamma})$:

Lemma 2 *Denote* $g_{\bar{\boldsymbol{A}},\bar{\boldsymbol{\sigma}}}(\boldsymbol{\Theta}, \boldsymbol{\gamma}; \bar{\boldsymbol{\Theta}}, \bar{\boldsymbol{\gamma}})$ *as the following function:*

$$-\sum_{\ell=1}^{L}\sum_{r=1}^{R}\sum_{p=1}^{P} \beta_{\ell,p}^r \log\left(\frac{\gamma_p u(\boldsymbol{\theta}_p, \bar{\boldsymbol{\theta}}_p, \boldsymbol{\xi}_\ell^r) C(\bar{\boldsymbol{\theta}}_p, \boldsymbol{\xi}_\ell^r)}{\beta_{\ell,p}^r R \mu_\ell(\boldsymbol{\xi}_\ell^r)/h(\boldsymbol{y}_\ell|\boldsymbol{\xi}_\ell^r; \bar{\boldsymbol{A}}, \bar{\boldsymbol{\sigma}})}\right),$$

where $\beta_{\ell,p}^r = \frac{\bar{\gamma}_p \mathcal{D}(\boldsymbol{\xi}_\ell^r;\bar{\boldsymbol{\theta}}_p)h(\boldsymbol{y}_\ell|\boldsymbol{\xi}_\ell^r;\bar{\boldsymbol{A}},\bar{\boldsymbol{\sigma}})/R\mu_\ell(\boldsymbol{\xi}_\ell^r)}{\sum_{\tilde{r}=1}^{R}\sum_{\tilde{p}=1}^{P} \bar{\gamma}_{\tilde{p}} \mathcal{D}(\boldsymbol{\xi}_\ell^{\tilde{r}};\bar{\boldsymbol{\theta}}_{\tilde{p}})h(\boldsymbol{y}_\ell|\boldsymbol{\xi}_\ell^{\tilde{r}};\bar{\boldsymbol{A}},\bar{\boldsymbol{\sigma}})/R\mu_\ell(\boldsymbol{\xi}_\ell^{\tilde{r}})}$,

$$u(\boldsymbol{\theta}, \bar{\boldsymbol{\theta}}, \boldsymbol{s}) = \exp\left\{(\mathbf{1}^T\boldsymbol{\theta})\Psi(\mathbf{1}^T\bar{\boldsymbol{\theta}}) - \sum_{n=1}^{N} \log\left(\frac{\Gamma(\theta_n)}{s_n^{\theta_n}}\right)\right\},$$

$$C(\bar{\boldsymbol{\theta}}, \boldsymbol{s}) = \exp\left\{\log\left(\frac{\Gamma(\mathbf{1}^T\bar{\boldsymbol{\theta}})}{\prod_{n=1}^{N} s_n}\right) - (\mathbf{1}^T\bar{\boldsymbol{\theta}})\Psi(\mathbf{1}^T\bar{\boldsymbol{\theta}})\right\}.$$

Then, $g_{\bar{\boldsymbol{A}},\bar{\boldsymbol{\sigma}}}(\boldsymbol{\Theta}, \boldsymbol{\gamma}; \bar{\boldsymbol{\Theta}}, \bar{\boldsymbol{\gamma}})$ *is a majorizer of* $f_{\bar{\boldsymbol{A}},\bar{\boldsymbol{\sigma}}}(\boldsymbol{\Theta}, \boldsymbol{\gamma})$.

Lemma 2 is non-trivial. Its proof leverages on the property of the gamma function [12] and Jensen's inequality; we skip the proof due to the page limit. Invoking Lemma 2, problem (6b) can be equivalently expressed as (after dropping the constant terms):

$$\min_{\boldsymbol{\Theta}\in\mathbb{R}^{M\times P}_{++},\boldsymbol{\gamma}\in\mathcal{U}_P} -\sum_{\ell=1}^{L}\sum_{r=1}^{R}\sum_{p=1}^{P} \beta_{\ell,p}^r \log(\gamma_p u(\boldsymbol{\theta}_p, \bar{\boldsymbol{\theta}}_p, \boldsymbol{\xi}_\ell^r)). \quad (7)$$

Notice that $\boldsymbol{\Theta}$ and $\boldsymbol{\gamma}$ are decoupled in problem (7). It can be verified that both $\boldsymbol{\Theta}$ and $\boldsymbol{\gamma}$ can be computed in closed form:

$$[\boldsymbol{\theta}_p^+]_n = \Psi^{-1}\left(\Psi\left(\mathbf{1}^T\bar{\boldsymbol{\theta}}_p\right) + \frac{\sum_{\ell=1}^{L}\sum_{r=1}^{R} \beta_{\ell,p}^r \log([\boldsymbol{\xi}_\ell^r]_n)}{\sum_{\ell=1}^{L}\sum_{r=1}^{R} \beta_{\ell,p}^r}\right),$$

$$\gamma_p^+ = \frac{\sum_{\ell=1}^{L}\sum_{r=1}^{R} \beta_{\ell,p}^r}{\sum_{\ell=1}^{L}\sum_{r=1}^{R}\sum_{\tilde{p}=1}^{P} \beta_{\ell,\tilde{p}}^r},$$

for $p = 1, \ldots, P$, $n = 1, \ldots, N$, where $\Psi^{-1}(\cdot)$ is the inverse digamma function. We should mention that both $\Psi(\cdot)$ and $\Psi^{-1}(\cdot)$ can be numerically computed efficiently; see [17].

Combining the above pieces together, we obtain an SAA-based BSUM algorithm for the ML problem (3), which is named *Saa-bsuM-mL (SML)* and summarized in Algorithm 1.

Algorithm 1 SML Algorithm for Problem (3)

1: **input** , $\{\boldsymbol{y}_\ell\}_{\ell=1}^L, \boldsymbol{A}^0, \boldsymbol{\sigma}^0, \boldsymbol{\Theta}^0, \boldsymbol{\gamma}^0$;
2: $k = 0$
3: **repeat**
4: sample $\{\boldsymbol{\xi}_\ell^r\}_{r=1}^R$ from $\mu_\ell(\boldsymbol{s}_\ell; \boldsymbol{A}^k, \boldsymbol{\sigma}^k) \propto h(\boldsymbol{y}_\ell|\boldsymbol{s}_\ell, \boldsymbol{A}^k, \boldsymbol{\sigma}^k)$ over $\boldsymbol{s}_\ell \in \mathcal{U}_N$, $\ell = 1, \ldots, L$ (cf. Sec.3.3);
5: *% update* $(\boldsymbol{A}, \boldsymbol{\sigma})$
6: $\alpha_\ell^r \leftarrow \frac{q(\boldsymbol{\xi}_\ell^r;\boldsymbol{\Theta}^k,\boldsymbol{\gamma}^k)h(\boldsymbol{y}_\ell|\boldsymbol{\xi}_\ell^r;\boldsymbol{A}^k,\boldsymbol{\sigma}^k)/R\mu_\ell(\boldsymbol{\xi}_\ell^r)}{\sum_{\tilde{r}=1}^{R} q(\boldsymbol{\xi}_\ell^{\tilde{r}};\boldsymbol{\Theta}^k,\boldsymbol{\gamma}^k)h(\boldsymbol{y}_\ell|\boldsymbol{\xi}_\ell^{\tilde{r}};\boldsymbol{A}^k,\boldsymbol{\sigma}^k)/R\mu_\ell(\boldsymbol{\xi}_\ell^{\tilde{r}})}$, $r = 1, \ldots, R$, $\ell = 1, \ldots, L$;
7: $\boldsymbol{A}^{k+1} = \left[\sum_{\ell=1}^{L} \boldsymbol{y}_\ell \left(\sum_{r=1}^{R} \alpha_\ell^r \boldsymbol{\xi}_\ell^r\right)^T\right]\left[\sum_{\ell=1}^{L}\sum_{r=1}^{R} \alpha_\ell^r \boldsymbol{\xi}_\ell^r (\boldsymbol{\xi}_\ell^r)^T\right]^\dagger$;
8: $\boldsymbol{\sigma}^{k+1} = \frac{1}{L}\sum_{\ell=1}^{L}\sum_{r=1}^{R} \alpha_\ell^r (\boldsymbol{y}_\ell - \boldsymbol{A}^{k+1}\boldsymbol{\xi}_\ell^r)^2$;
9: *% update* $(\boldsymbol{\Theta}, \boldsymbol{\gamma})$
10: $\beta_{\ell,p}^r \leftarrow \frac{\gamma_p^k \mathcal{D}(\boldsymbol{\xi}_\ell^r;\boldsymbol{\theta}_p^k)h(\boldsymbol{y}_\ell|\boldsymbol{\xi}_\ell^r;\boldsymbol{A}^{k+1},\boldsymbol{\sigma}^{k+1})/R\mu_\ell(\boldsymbol{\xi}_\ell^r)}{\sum_{\tilde{r}=1}^{R}\sum_{\tilde{p}=1}^{P} \gamma_{\tilde{p}}^k \mathcal{D}(\boldsymbol{\xi}_\ell^r;\boldsymbol{\theta}_{\tilde{p}}^k)h(\boldsymbol{y}_\ell|\boldsymbol{\xi}_\ell^r;\boldsymbol{A}^{k+1},\boldsymbol{\sigma}^{k+1})/R\mu_\ell(\boldsymbol{\xi}_\ell^r)}$, $p = 1, \ldots, P$, $\ell = 1, \ldots, L$, $r = 1, \ldots, R$;
11: $\gamma_p^{k+1} = \frac{\sum_{\ell=1}^{L}\sum_{r=1}^{R} \beta_{\ell,p}^r}{\sum_{\ell=1}^{L}\sum_{r=1}^{R}\sum_{\tilde{p}=1}^{P} \beta_{\ell,\tilde{p}}^r}$, $p = 1, \ldots, P$;
12: $[\boldsymbol{\theta}_p^{k+1}]_n = \Psi^{-1}\left(\Psi\left(\mathbf{1}^T\boldsymbol{\theta}_p^k\right) + \frac{\sum_{\ell=1}^{L}\sum_{r=1}^{R} \beta_{\ell,p}^r \log([\boldsymbol{\xi}_\ell^r]_n)}{\sum_{\ell=1}^{L}\sum_{r=1}^{R} \beta_{\ell,p}^r}\right)$, $n = 1, \ldots, N$, $p = 1 \ldots, P$;
13: $\boldsymbol{\Theta}^{k+1} = [\boldsymbol{\theta}_1^{k+1}, \ldots, \boldsymbol{\theta}_P^{k+1}]$;
14: $k = k + 1$;
15: **until** some stopping criterion is satisfied.
16: **output** $(\boldsymbol{A}^k, \boldsymbol{\sigma}^k, \boldsymbol{\Theta}^k, \boldsymbol{\gamma}^k)$.

3.3. Choice of $\mu_\ell(\boldsymbol{s}_\ell)$

In this subsection, we discuss how to choose the sampling distribution $\mu_\ell(\boldsymbol{s}_\ell)$ in the approximation (4). One typical choice of $\mu_\ell(\boldsymbol{s}_\ell)$ would be $\mathcal{D}(\boldsymbol{s}_\ell; \mathbf{1})$; i.e., $\boldsymbol{s}_\ell$ follows a uniform Dirichlet distribution. However, such a choice cannot incorporate the up-to-date information $(\boldsymbol{A}^k, \boldsymbol{\sigma}^k, \boldsymbol{\Theta}^k, \boldsymbol{\gamma}^k)$ into the sampling process. Empirically, we found that a more effective way is to adaptively change $\mu_\ell(\boldsymbol{s}_\ell)$ with the iteration. To be specific, we iteratively update $\mu_\ell(\boldsymbol{s}_\ell; \boldsymbol{A}^k, \boldsymbol{\sigma}^k) \propto h(\boldsymbol{y}_\ell|\boldsymbol{s}_\ell, \boldsymbol{A}^k, \boldsymbol{\sigma}^k)$ over $\boldsymbol{s}_\ell \in \mathcal{U}_N$ and re-sample $\{\boldsymbol{\xi}_\ell^r\}_{r=1}^R$ from $\mu_\ell(\boldsymbol{s}_\ell; \boldsymbol{A}^k, \boldsymbol{\sigma}^k)$ at the $(k+1)$th iteration; cf., line 4 in Algorithm 1. Notice that $\mu_\ell(\boldsymbol{s}_\ell; \boldsymbol{A}^k, \boldsymbol{\sigma}^k)$ is a Gaussian distribution truncated in the unit simplex, which is not hard to sample; see [18] for a review of related sampling techniques. According to our numerical experience, this adaptive and re-sampling strategy leads to much better results than uniform Dirichlet distribution without re-sampling.

4. SIMULATIONS

We test our proposed algorithm in two different applications, namely, hyperspectral unmixing and document clustering.

4.1. Hyperspectral Unmixing

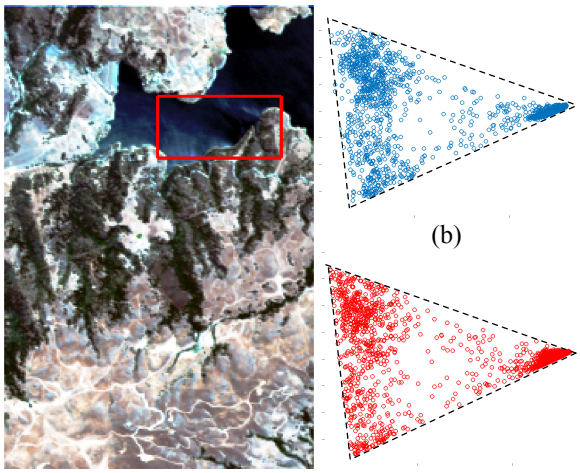

Fig. 1. (a) AVIRIS "Moffett Field" image (red rectangle denotes the selected subimage); (b) scatterplot of groundtruth abundances; (c) scatterplot of samples generated from the estimated distribution.

Hyperspectral unmixing (HU) is a classical problem in remote sensing. Its goal is to identify materials of a scene from a hyperspectral image. The model of HU can be described by (1), where $\boldsymbol{y}_\ell$ represents the ℓth spectral pixel of a hyperspectral image, each column of $\boldsymbol{A}$ represents a material spectral signature, and $\boldsymbol{s}_\ell$ denotes the abundance of the materials in the ℓth pixel; see [5, 6] for details. Our simulations are conducted using synthetic data. Specifically, the procedure is as follows: 1) randomly select 1,000 abundances $\boldsymbol{s}_\ell$'s from an abundance pool retrieved from a real hyperspectral image called AVIRIS Moffett Field [19] (there are three materials in the selected area; see Fig. 1(a)); 2) randomly select spectral signatures from the USGS library [20] to construct $\boldsymbol{A}$; 3) generate $\boldsymbol{y}_\ell$'s according to (1) with a specified SNR. The problem sizes are $(M, N, L) = (224, 3, 1000)$. For comparison, four classical HU algorithms are considered, namely, SISAL, RVolMin, MVC-NMF and Bayesian MCMC (B-MCMC) [21–24]. For SML, we consider two versions: the first one, denoted as SML-U, assumes uniform Dirichlet distribution on $\boldsymbol{s}_\ell$'s (i.e., fixing $(\boldsymbol{\Theta}, \boldsymbol{\gamma}) = (\mathbf{1}, 1)$), and estimates $\boldsymbol{A}$ from problem (6a) by MM; the second one, denoted as SML-E, implements Algorithm 1 with mixture number $P = 10$. A sample size of $R = 100$ is used in both SML-U and SML-E.

We first test whether SML-E can correctly learn the distribution of $\boldsymbol{s}_\ell$'s. To this end, we run SML-E in a 20dB SNR scenario to estimate the parameters of the Dirichlet mixture distribution $(\boldsymbol{\Theta}, \boldsymbol{\gamma})$, and generate samples from the estimated distribution. The scatterplot of the groundtruth abundances and the sampled abundances are shown in Figs. 1 (b) and (c), resp. We see that the estimated distribution is a reasonable approximation of the true. Next, we compare the average mean square errors of different algorithms. The results are shown in Fig. 2; the number of trials is 50. Clearly, SML-E achieves the best performance in all cases, while SML-U suffers from around 4dB performance loss. This makes sense since the latter does not model the abundance distribution well. Finally, we provide the runtime performance of the considered algorithms in Table 1.

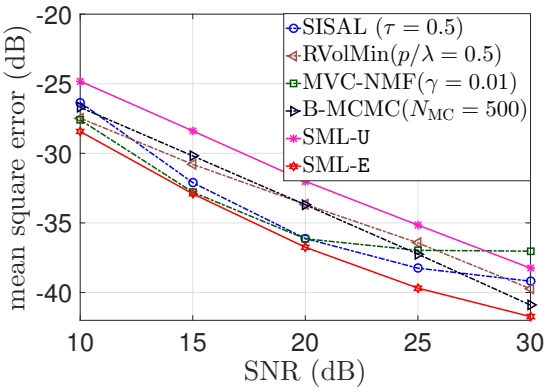

Fig. 2. MSE performance under various SNRs.

Table 1. Runtime performance of considered algorithms

Algorithm	SISAL	RVolMin	MVC-NMF
Runtime (sec.)	0.472±0.022	2.565±0.095	3.623±1.198
Algorithm	B-MCMC	SML-U	SML-E
Runtime (sec.)	299.398±1.846	46.482±1.301	97.042±3.989

Table 2. Clustering accuracy of considered algorithms

Algorithm	Clustering Accuracy			
	N=3	N=4	N=5	N=6
LCCF	0.81±0.16	0.73±0.12	0.71±0.09	0.65±0.08
SISAL ($\tau = 0.5$)	0.79±0.15	0.72±0.12	0.70±0.09	0.66±0.07
RVolMin ($p/\lambda = 1.5/1$)	0.82±0.13	0.78±0.11	0.75±0.09	0.73±0.09
SML	**0.87±0.11**	**0.80±0.12**	**0.78±0.08**	**0.74±0.08**

4.2. Document Clustering

The second application is document clustering, where $\boldsymbol{y}_\ell$ is the term-frequency-inverse-document-frequency (tf-idf) vector of the ℓth document; each column of $\boldsymbol{A}$ represents the tf-idf vector of a topic; $\boldsymbol{s}_\ell$ is the topic weight of the ℓth document. Instead of directly clustering the tf-idf vectors, SSMF-based methods consider retrieving $\boldsymbol{s}_\ell$'s first from $\boldsymbol{y}_\ell$'s, and then applying clustering (e.g. k-means) on $\boldsymbol{s}_\ell$'s to cluster documents.

The Reuters21578 corpus in [25] is used, which contains preprocessed tf-idf vectors of documents that are clustered into 41 topics. In each trial, we randomly select N topics; and for each topic, 100 tf-idf vectors are randomly selected. We compare SML with SISAL, RVolMin and LCCF [25], where the last algorithm is considered as the benchmark algorithm in document clustering. The settings are identical to those in the last subsection. For SML, we set $P = N$. We run 50 independent trials for each N, and use the clustering accuracy (see [25] for definition) to quantitatively assess the performance. The results are summarized in Table 2. As seen, SML achieves promising performance for all cases.

5. CONCLUSION

To conclude, we proposed a stochastic ML framework for the SSMF problem. The resulting ML problem is challenging, and an SAA-based alternating optimization approach is employed to handle it. The efficacy of the proposed method was corroborated by two real-world applications from remote sensing and document clustering.

6. REFERENCES

[1] D. D. Lee and H. S. Seung, "Learning the parts of objects by non-negative matrix factorization," *Nature*, vol. 401, no. 6755, p. 788, 1999.

[2] A. Cichocki, R. Zdunek, A. H. Phan, and S.-i. Amari, *Nonnegative Matrix and Tensor Factorizations: Applications to Exploratory Multi-Way Data Analysis and Blind Source Separation*. John Wiley & Sons, 2009.

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

[4] X. Fu, K. Huang, N. D. Sidiropoulos, and W.-K. Ma, "Nonnegative matrix factorization for signal and data analytics: Identifiability, algorithms, and applications," *arXiv preprint arXiv:1803.01257*, 2018.

[5] 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 J. Sel. Topics Appl. Earth Observ. in Remote Sens.*, vol. 5, no. 2, pp. 354–379, 2012.

[6] W.-K. Ma, J. M. Bioucas-Dias, T.-H. Chan, N. Gillis, P. Gader, A. J. Plaza, A. Ambikapathi, and C.-Y. Chi, "A signal processing perspective on hyperspectral unmixing," *IEEE Signal Process. Mag.*, vol. 31, no. 1, pp. 67–81, 2014.

[7] S. Arora, R. Ge, R. Kannan, and A. Moitra, "Computing a nonnegative matrix factorization–provably," in *Proc. 44th ACM Symp. Theory Comput.* ACM, 2012, pp. 145–162.

[8] E. Elhamifar, G. Sapiro, and R. Vidal, "See all by looking at a few: Sparse modeling for finding representative objects," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2012, pp. 1600–1607.

[9] X. Fu, W.-K. Ma, K. Huang, and N. D. Sidiropoulos, "Blind separation of quasi-stationary sources: Exploiting convex geometry in covariance domain," *IEEE Trans. Signal Process.*, vol. 63, no. 9, pp. 2306–2320, 2015.

[10] J. M. Nascimento and J. M. Bioucas-Dias, "Hyperspectral unmixing based on mixtures of Dirichlet components," *IEEE Trans. Geosci. Remote Sens.*, vol. 50, no. 3, pp. 863–878, 2012.

[11] R. Wu, W.-K. Ma, and X. Fu, "A stochastic maximum-likelihood framework for simplex structured matrix factorization," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process.* IEEE, 2017, pp. 2557–2561.

[12] T. Minka, "Estimating a Dirichlet distribution," *Technical Report, MIT*, 2000.

[13] R. M. Neal, "Annealed importance sampling," *Stat. Comput.*, vol. 11, no. 2, pp. 125–139, 2001.

[14] A. Shapiro, D. Dentcheva, and A. Ruszczynski, *Lectures on Stochastic Programming: Modeling and Theory*. SIAM, 2014, vol. 16.

[15] M. Razaviyayn, M. Hong, and Z.-Q. Luo, "A unified convergence analysis of block successive minimization methods for nonsmooth optimization," *SIAM J. Optim.*, vol. 23, no. 2, pp. 1126–1153, 2013.

[16] D. R. Hunter and K. Lange, "A tutorial on MM algorithms," *Am. Stat.*, vol. 58, no. 1, pp. 30–37, 2004.

[17] B. C. Berndt, *Ramanujans Notebooks*. Springer Science & Business Media, 2012.

[18] Y. Altmann, S. McLaughlin, and N. Dobigeon, "Sampling from a multivariate Gaussian distribution truncated on a simplex: a review," in *Proc. 2014 IEEE Stat. Signal Process. Workshop (SSP)*, 2014.

[19] AVIRIS, "Jet Propulsion Lab," *California Inst. Technol., Pasadena.* Available [online]: http://aviris.jpl.nasa.gov/html/aviris.freedata.html.

[20] R. N. Clark, G. A. Swayze, R. Wise, K. E. Livo, T. Hoefen, R. F. Kokaly, and S. J. Sutley, "USGS digital spectral library splib06a," *U.S. Geol. Surv., Digital Data Ser.*, vol. 231, 2007.

[21] J. M. Bioucas-Dias, "A variable splitting augmented Lagrangian approach to linear spectral unmixing," in *Proc. Workshop Hyperspectral Image Signal Process.: Evol. Remote Sens.*, Aug. 2009, pp. 1–4.

[22] X. Fu, K. Huang, B. Yang, W.-K. Ma, and N. D. Sidiropoulos, "Robust volume minimization-based matrix factorization for remote sensing and document clustering," *IEEE Trans. Signal Process.*, vol. 64, no. 23, pp. 6254–6268, 2016.

[23] L. Miao and H. Qi, "Endmember extraction from highly mixed data using minimum volume constrained nonnegative matrix factorization," *IEEE Trans. Geosci. Remote Sens.*, vol. 45, no. 3, pp. 765–777, Mar. 2007.

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

[25] D. Cai, X. He, and J. Han, "Locally consistent concept factorization for document clustering," *IEEE Trans. Knowl. Data Eng.*, vol. 23, no. 6, pp. 902–913, 2011.