A STOCHASTIC MAXIMUM-LIKELIHOOD FRAMEWORK FOR SIMPLEX STRUCTURED MATRIX FACTORIZATION

Ruiyuan Wu, Wing-Kin Ma*, and Xiao Fu†*

*Department of Electronic Engineering, The Chinese University of Hong Kong, Shatin, N.T., Hong Kong
†Department of Electrical and Computer Engineering, University of Minnesota, Minneapolis, MN, USA

ABSTRACT

Consider a structured matrix factorizaton (SMF) whose coefficient vectors are constrained to lie in the unit simplex. This kind of simplex SMF (SSMF) has received growing attention and has found many applications such as hyperspectral unmixing in remote sensing, text mining in machine learning, and blind source separation in signal processing. The aim of this paper is to establish a maximum-likelihood (ML) estimation framework for SSMF in the presence of Gaussian noise and outliers, and to demonstrate its potential. Our ML formulation has the coefficient vectors marginalized in accordance with a prescribed probabilistic model, and this leads to a likelihood function that contains multi-dimensional integrals. Unfortunately these integrals do not appear to have analytically tractable solutions, and this makes the ML problem challenging. We tackle the problem by using sample average approximation in stochastic optimization and majorization-minimization. Simulation results show that the resulting ML algorithm significantly outperforms several existing methods when noise and outliers are present.

Index Terms— Structured matrix factorization, majorization minimization, sample average approximation, hyperspectral unmixing

1. INTRODUCTION

Consider a data model as follows:

$$\boldsymbol{y}_n = \boldsymbol{A}\boldsymbol{s}_n + \boldsymbol{v}_n, \quad n = 1, \ldots, L, \tag{1}$$

where $\boldsymbol{y}_n \in \mathbb{R}^M$ is an observed data point, $\boldsymbol{A} \in \mathbb{R}^{M\times N}$ is a basis matrix, $\boldsymbol{s}_n \in \mathbb{R}^N$ is the coefficient vector for $\boldsymbol{y}_n$, and $\boldsymbol{v}_n \in \mathbb{R}^M$ is noise. Here, N describes the basis rank, and we assume $N < \max\{M, L\}$. The above data model can be expressed in a factored form $\boldsymbol{Y} = \boldsymbol{A}\boldsymbol{S}+\boldsymbol{V}$, where $\boldsymbol{Y} = [\, \boldsymbol{y}_1, \ldots, \boldsymbol{y}_L \,]$, $\boldsymbol{S} = [\, \boldsymbol{s}_1, \ldots, \boldsymbol{s}_L \,]$, $\boldsymbol{V} = [\, \boldsymbol{v}_1, \ldots, \boldsymbol{v}_L \,]$. The problem of structured matrix factorization (SMF) is to determine $\boldsymbol{A}$ and $\boldsymbol{S}$ from $\boldsymbol{Y}$, assuming some structures with $\boldsymbol{A}$ and/or $\boldsymbol{S}$.

For example, non-negative matrix factorization (NMF), which assumes $\boldsymbol{A} \geq \boldsymbol{0}$, $\boldsymbol{S} \geq \boldsymbol{0}$, is seen as an SMF; here, the notation $\boldsymbol{X} \geq \boldsymbol{0}$ means that $\boldsymbol{X}$ is element-wise non-negative. NMF is well known for its wide variety of applications [1, 2]. There is another type of SMF that has recently drawn significantly growing attention. This SMF has many names or is closely related to several independent developments from different areas, and we will use the name *simplex SMF (SSMF)* for the sake of simplicity. SSMF assumes that every $\boldsymbol{s}_n$ lies in the unit simplex, i.e., it satisfies $\boldsymbol{s}_n \geq \boldsymbol{0}$, $\boldsymbol{1}^T\boldsymbol{s}_n = 1$ ($\boldsymbol{1}$ denotes an all-one vector); the basis $\boldsymbol{A}$ does not need to be non-negative, although such a structure can also be added in the model if one desires. The developments of SSMF are probably most prominent in hyperspectral unmixing, a very active topic in remote sensing; see [3, 4] for the background. Lately, SSMF has also attracted much interest in machine learning [5, 6]. There, the application lies in text mining, and SSMF is often associated with, or studied under, a framework called separable NMF. Other than that, SSMF has found many applications such as video summarization in computer vision [7], blind separation of speech sources [8], blind spectra estimation [9], to name just a few. One appealing aspect of SSMF is its theoretical identifiability. It has been shown that under some mild assumptions such as separability [5] and a much more relaxed form of separability [8, 10, 11], SSMF exhibits very desirable identifiability results with the extracted factors $\boldsymbol{A}, \boldsymbol{S}$.

In this paper we consider a maximum-likelihood (ML) framework for SSMF. The framework we adopted is not to estimate $\boldsymbol{A}$ and $\boldsymbol{S}$ jointly. Instead, a probabilistic model is applied on $\boldsymbol{S}$, and the marginalized likelihood with respect to $\boldsymbol{S}$ is used as the ML metric. Consequently, we care only about the estimation of $\boldsymbol{A}$ (once $\boldsymbol{A}$ is obtained, one can easily estimate $\boldsymbol{S}$ by solving an inverse problem). Such an ML formulation was considered only in the noiseless case in previous work [12]. Here, we assume not only the noisy case, but also the possible presence of outliers. The resulting ML problem is challenging, and we tackle it via a combination of stochastic optimization and majorization-minimization techniques. Our numerical results will show that the ML framework holds great potential in enhancing estimation performance in the noisy case.

2. THE ML FORMULATION

Let us describe the probabilistic model that will lead to our ML formulation. Given each index n, we assume that there is a probability that $\boldsymbol{y}_n$ is an outlier and does not follow the nominal model (1). Also, for the nominal model (1), the noise vectors $\boldsymbol{v}_n$'s are independent and identically distributed (i.i.d.) and follow a Gaussian distribution. The probability density function (p.d.f.) of $\boldsymbol{y}_n$ conditioned on $\boldsymbol{s}_n$ and given $\boldsymbol{A}$ may therefore be modeled as

$$p(\boldsymbol{y}_n|\boldsymbol{s}_n;\boldsymbol{A}) = (1-\beta)\mathcal{N}(\boldsymbol{y}_n;\boldsymbol{A}\boldsymbol{s}_n,\boldsymbol{\Sigma}) + \beta h(\boldsymbol{y}_n),$$

where $0 \leq \beta < 1$ is the probability that $\boldsymbol{y}_n$ is an outlier,

$$\mathcal{N}(\boldsymbol{x};\boldsymbol{\mu},\boldsymbol{\Sigma}) = \frac{1}{\sqrt{(2\pi)^M \det(\boldsymbol{\Sigma})}} e^{-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu})^T\boldsymbol{\Sigma}^{-1}(\boldsymbol{x}-\boldsymbol{\mu})}$$

denotes the Gaussian p.d.f. with mean $\boldsymbol{\mu}$ and covariance $\boldsymbol{\Sigma}$, and h is the outlier p.d.f. which is assumed to be independent of $\boldsymbol{A}$

This work was supported by the Hong Kong Research Grants Council (RGC) General Research Fund (GRF) under Project CUHK 14205414.

and $\boldsymbol{s}_n$. The outlier p.d.f. h can take any positive p.d.f. function, and usually we would choose an uniform distribution function for h. The coefficient vectors $\boldsymbol{s}_n$'s are modeled to be i.i.d. and follow a Dirichlet distribution

$$q(\boldsymbol{s}_n) = \frac{\Gamma(\sum_{i=1}^N \alpha_i)}{\prod_{i=1}^N \Gamma(\alpha_i)} \prod_{i=1}^N s_{n,i}^{\alpha_i - 1}, \quad \boldsymbol{s}_n \in \mathcal{S},$$

where Γ is the gamma function, $\boldsymbol{\alpha} > \boldsymbol{0}$ is the so-called concentration parameter of the Dirichlet distribution, and $\mathcal{S} = \{\boldsymbol{s} \mid \boldsymbol{s} \geq \boldsymbol{0}, \boldsymbol{1}^T \boldsymbol{s} = 1\}$ is the unit simplex.

The ML formulation is as follows. Let

$$\begin{aligned} p(\boldsymbol{y}_n; \boldsymbol{A}) &= \int_{\mathcal{S}} p(\boldsymbol{y}_n | \boldsymbol{s}_n; \boldsymbol{A}) q(\boldsymbol{s}_n) d\boldsymbol{s}_n \\ &= (1-\beta) \int_{\mathcal{S}} \mathcal{N}(\boldsymbol{y}_n; \boldsymbol{A}\boldsymbol{s}_n, \boldsymbol{\Sigma}) q(\boldsymbol{s}_n) d\boldsymbol{s}_n + \beta h(\boldsymbol{y}_n) \end{aligned} \tag{2}$$

be the marginalized p.d.f. of $\boldsymbol{y}_n$ with respect to $\boldsymbol{s}_n$. We seek to maximize the likelihood function $\prod_{n=1}^L p(\boldsymbol{y}_n; \boldsymbol{A})$, or equivalently,

$$\min_{\boldsymbol{A} \in \mathbb{R}^{M \times N}} \; -\textstyle\sum_{n=1}^L \log\left(p(\boldsymbol{y}_n; \boldsymbol{A})\right). \tag{3}$$

Solving the ML problem (3) is the main focus of this paper. At this point, we should mention that the ML problem (3) is hard to handle (at least directly). The main difficulty is that the integral in (2) does not appear to have an explicit or closed-form expression in general; the only known exception is the noiseless case [12].

3. THE PROPOSED SOLUTION

In this section we develop a method for handling the ML problem (3). There are two ingredients with our approach, namely, sample average approximation (SAA) and majorization-minimization (MM). SAA is a widely-used technique in stochastic optimization and its idea is as follows. Suppose that we can generate random samples in accordance with $q(\boldsymbol{s}_n)$, which is true for our problem (Dirichlet random variables are very easy to generate). Let $\boldsymbol{\xi}_n^1, \ldots, \boldsymbol{\xi}_n^R$ be a collection of i.i.d. random samples drawn from $q(\boldsymbol{s}_n)$, where R is the sample size. We approximate (2) by the sample average

$$p(\boldsymbol{y}_n; \boldsymbol{A}) \approx (1-\beta) \frac{1}{R} \sum_{i=1}^R \mathcal{N}(\boldsymbol{y}_n; \boldsymbol{A}\boldsymbol{\xi}_n^i, \boldsymbol{\Sigma}) + \beta h(\boldsymbol{y}_n),$$

and plug the above approximation into the ML problem (3) to obtain an SAA-ML problem

$$\min_{\boldsymbol{A} \in \mathbb{R}^{M \times N}} \; -\textstyle\sum_{n=1}^L \log\left(\frac{1-\beta}{R} \sum_{i=1}^R \mathcal{N}\left(\boldsymbol{y}_n; \boldsymbol{A}\boldsymbol{\xi}_n^i, \boldsymbol{\Sigma}\right) + \beta h(\boldsymbol{y}_n)\right). \tag{4}$$

Generally, SAA would require a large number of samples R to provide good approximation. Our empirical experience is that for a basis rank of 5 or less, we can obtain reasonably good results by using a sample size of about $R = 500$. Moreover, in the optimization context it has been shown that SAA exhibits certain desirable properties; e.g., solution proximity before and after SAA when R is large. Readers are referred to [13] for details.

However, SAA alone is not enough. The SAA-ML problem (4) is non-convex, and to tackle this issue we need another technique. We use MM [14, 15] and the operating principle is as follows. Let $f(\boldsymbol{A})$ be the objective function of problem (4), and let $u(\boldsymbol{A}, \bar{\boldsymbol{A}})$ be a function that satisfies the following two properties:

$$f(\boldsymbol{A}) \leq u(\boldsymbol{A}, \bar{\boldsymbol{A}}), \quad \text{for any } \boldsymbol{A}, \bar{\boldsymbol{A}}, \tag{5a}$$

$$f(\boldsymbol{A}) = u(\boldsymbol{A}, \boldsymbol{A}), \quad \text{for any } \boldsymbol{A}. \tag{5b}$$

Such a function is called a *majorizer* of f. A majorizer requires design in a problem-specific sense, and it is usually easier to minimize than f. MM works by running an iterative update

$$\boldsymbol{A}^{t+1} = \arg \min_{\boldsymbol{A} \in \mathbb{R}^{M \times N}} \; u(\boldsymbol{A}, \boldsymbol{A}^t), \quad t = 1, 2, \ldots,$$

As a desirable property, it has been shown that a limit point of $\{\boldsymbol{A}^t\}$ is guaranteed to converge to a stationary point of the problem if f and u are smooth; see [15].

Now, we claim that the following function is a good majorizer of f:

$$u(\boldsymbol{A}, \bar{\boldsymbol{A}}) = \\ -\sum_{n=1}^L \left[\sum_{i=1}^R \theta_n^i \log\left(\frac{\mathcal{N}(\boldsymbol{y}_n; \boldsymbol{A}\boldsymbol{\xi}_n^i, \boldsymbol{\Sigma})}{R\theta_n^i/(1-\beta)}\right) + \theta_n^{R+1} \log\left(\frac{\beta h(\boldsymbol{y}_n)}{\theta_n^{R+1}}\right) \right], \tag{6}$$

where

$$\theta_n^i = \begin{cases} \dfrac{\frac{1-\beta}{R} \mathcal{N}(\boldsymbol{y}_n; \bar{\boldsymbol{A}}\boldsymbol{\xi}_n^i, \boldsymbol{\Sigma})}{\frac{1-\beta}{R} \sum_{j=1}^R \mathcal{N}(\boldsymbol{y}_n; \bar{\boldsymbol{A}}\boldsymbol{\xi}_n^j, \boldsymbol{\Sigma}) + \beta h(\boldsymbol{y}_n)}, & i = 1, \ldots, R, \\ \dfrac{\beta h(\boldsymbol{y}_n)}{\frac{1-\beta}{R} \sum_{j=1}^R \mathcal{N}(\boldsymbol{y}_n; \bar{\boldsymbol{A}}\boldsymbol{\xi}_n^j, \boldsymbol{\Sigma}) + \beta h(\boldsymbol{y}_n)}, & i = R+1. \end{cases} \tag{7}$$

It can be verified that (6) is a majorizer of f (or satisfies (5)). The proof, which we skip here, is nothing more than applying Jensen's inequality and using the fact that $\theta_n^i \geq 0$ for all n, i, $\sum_{i=1}^{R+1} \theta_n^i = 1$ for all n. To see why (6) is a good majorizer, consider the minimization $\min_{\boldsymbol{A}} u(\boldsymbol{A}, \bar{\boldsymbol{A}})$. Through careful algebraic manipulations, it can be shown that

$$u(\boldsymbol{A}, \bar{\boldsymbol{A}}) = \mathrm{Tr}(\boldsymbol{\Phi}\boldsymbol{A}^T\boldsymbol{\Sigma}^{-1}\boldsymbol{A}) - 2\mathrm{Tr}(\boldsymbol{C}\boldsymbol{A}^T\boldsymbol{\Sigma}^{-1}) + c(\bar{\boldsymbol{A}}), \tag{8}$$

where $c(\bar{\boldsymbol{A}})$ is a term that does not depend on $\boldsymbol{A}$,

$$\boldsymbol{\Phi} = \sum_{n=1}^L \sum_{i=1}^R \theta_n^i \boldsymbol{\xi}_n^i (\boldsymbol{\xi}_n^i)^T, \quad \boldsymbol{C} = \sum_{n=1}^L \boldsymbol{y}_n \left(\sum_{i=1}^R \theta_n^i \boldsymbol{\xi}_n^i\right)^T.$$

In particular, note that u is convex quadratic in $\boldsymbol{A}$. It is easy to show that the solution to $\min_{\boldsymbol{A}} u(\boldsymbol{A}, \bar{\boldsymbol{A}})$ is simply

$$\arg\min_{\boldsymbol{A}} u(\boldsymbol{A}, \bar{\boldsymbol{A}}) = \boldsymbol{C}\boldsymbol{\Phi}^\dagger; \tag{9}$$

(the superscript "†" denotes the pseudo-inverse).

Putting the above components together, we obtain an SAA-MM algorithm for the ML problem in Algorithm 1. As seen, the algorithm takes an iteratively reweighted least squares form and is simple to implement.

We should discuss the relationship of the above SAA-MM approach and some existing ML approach. SAA-MM is, in fact, very similar to the implementation of expectation maximization (EM) via Monte-Carlo (MC) averaging [16,17]. However, the concept of MM is more general and flexible than that of EM. Also, expert readers should note that the MC-EM approach does not allow one to straightforwardly derive our algorithm in Algorithm 1 (Hint: the obstacle lies in the outlier term, and how Jensen's inequality is applied turns out to make a significant difference).

Algorithm 1 SAA-MM

```
1: input: the observed data $\{\boldsymbol{y}_n\}_{n=1}^{L}$, a sample size $R$, and a starting point $\boldsymbol{A}^0$
2: i.i.d. generate $\{\boldsymbol{\xi}_n^i\}_{i=1}^{R}$ from $q(\boldsymbol{s}_n)$ for $n = 1, \ldots, L$;
3: $t = 0$;
4: repeat
5:     $\theta_n^i \leftarrow \frac{\frac{1-\beta}{R}\mathcal{N}(\boldsymbol{y}_n;\boldsymbol{A}^t\boldsymbol{\xi}_n^i,\boldsymbol{\Sigma})}{\frac{1-\beta}{R}\sum_{j=1}^{R}\mathcal{N}(\boldsymbol{y}_n;\boldsymbol{A}^t\boldsymbol{\xi}_n^j,\boldsymbol{\Sigma})+\beta h(\boldsymbol{y}_n)}, i = 1, \ldots, R, n = 1, \ldots, L$;
6:     $\theta_n^{R+1} \leftarrow \frac{\beta h(\boldsymbol{y}_n)}{\frac{1-\beta}{R}\sum_{j=1}^{R}\mathcal{N}(\boldsymbol{y}_n;\boldsymbol{A}^t\boldsymbol{\xi}_n^j,\boldsymbol{\Sigma})+\beta h(\boldsymbol{y}_n)}, n = 1, \ldots, L$;
7:     $\boldsymbol{A}^{t+1} = \left[\sum_{n=1}^{L}\boldsymbol{y}_n\left(\sum_{i=1}^{R}\theta_n^i\boldsymbol{\xi}_n^i\right)^T\right]\left(\sum_{n=1}^{L}\sum_{i=1}^{R}\theta_n^i\boldsymbol{\xi}_n^i(\boldsymbol{\xi}_n^i)^T\right)^{\dagger}$;
8:     $t = t + 1$;
9: until a stopping rule is satisfied.
10: output: $\boldsymbol{A}^t$.
```

4. EXTENSIONS AND MODIFICATION

In this section we describe several extensions and modification of our SAA-MM approach.

4.1. Unknown Outlier Probability β

Previously, the probability of occurrence of outliers β is assumed to be given. Suppose that β is unknown, and we wish to estimate both $\boldsymbol{A}$ and β from the above ML framework. As it turns out, this is easy. Let $u(\boldsymbol{A}, \beta, \bar{\boldsymbol{A}}, \bar{\beta})$ be a majorizer that takes the same expression as (6), except that $\theta_{n,i}$ in (7) has β replaced by $\bar{\beta}$. It can be shown that the minimization $\min_{\boldsymbol{A},\beta} u(\boldsymbol{A}, \beta, \bar{\boldsymbol{A}}, \bar{\beta})$ can be decoupled; the solution with respect to $\boldsymbol{A}$ is the same as (9), while the solution with respect to β is

$$\beta = \frac{1}{L}\sum_{n=1}^{L}\theta_n^{R+1}.$$

Hence, we only need to add the above equation in Algorithm 1 (specifically, between Step 6–7) to handle the unknown outlier probability case.

4.2. Non-Negative $\boldsymbol{A}$

Suppose that $\boldsymbol{A}$ is also required to be non-negative (or ML subject to $\boldsymbol{A} \geq \boldsymbol{0}$ is sought). We can apply the same SAA-MM in the last section, although the caveat is that we now need to solve a non-negatively constrained quadratic program (NCQP) $\min_{\boldsymbol{A}\geq\boldsymbol{0}} u(\boldsymbol{A}, \bar{\boldsymbol{A}})$ at every iteration—which has no closed-form solution in general. While we can employ efficient off-the-shelf solvers to handle the NCQP, the resulting SAA-MM algorithm may be computationally demanding at least in a per-iteration sense.

We use a different trick that leverages once again on MM principles. Let

$$\tilde{u}(\boldsymbol{A}, \bar{\boldsymbol{A}}) = u(\bar{\boldsymbol{A}}, \bar{\boldsymbol{A}}) + 2\mathrm{Tr}(\boldsymbol{\Sigma}^{-1}(\bar{\boldsymbol{A}}\boldsymbol{\Phi} - \boldsymbol{C})^T(\boldsymbol{A} - \bar{\boldsymbol{A}})) + \lambda\|\boldsymbol{A} - \bar{\boldsymbol{A}}\|_F^2, \tag{10}$$

where u is defined by the same way as before (see (6)), and

$$\lambda = \|\boldsymbol{\Sigma}^{-1}\|_2\|\boldsymbol{\Phi}\|_2;$$

(note that $\|\cdot\|_2$ denotes the matrix 2-norm). Eq. (10) is obtained by applying majorization on the quadratic terms of u in (8), and the majorization method is the same as that used in MM for $\ell_1 - \ell_2$ optimization in compressive sensing; see [18, 19] for details. Being an outcome of applying majorizations twice, $\tilde{u}$ is still a majorizer of f as one can verify from the definition (5). The advantage of using $\tilde{u}$ to build an SAA-MM algorithm is that the minimization $\min_{\boldsymbol{A}\geq\boldsymbol{0}} \tilde{u}(\boldsymbol{A}, \bar{\boldsymbol{A}})$ has a closed-form solution

$$\arg\min_{\boldsymbol{A}\geq\boldsymbol{0}} \tilde{u}(\boldsymbol{A}, \bar{\boldsymbol{A}}) = \left(\bar{\boldsymbol{A}} - \frac{1}{\lambda}\boldsymbol{\Sigma}^{-1}(\bar{\boldsymbol{A}}\boldsymbol{\Phi} - \boldsymbol{C})\right)_+, \tag{11}$$

where $(\cdot)_+$ denotes the projection onto the non-negative orthant. To summarize, we can handle the non-negative $\boldsymbol{A}$ case by replacing Step 7 of Algorithm 1 with (11).

4.3. An Accelerated Scheme

Let us continue our development for the non-negative $\boldsymbol{A}$ case in the last subsection. Empirically we found that the MM update in (11) leads to slow convergence with the SAA-MM algorithm. We handle this issue by taking insight from Nesterov's accelerated method [20] and some recent related development. Specifically, we replace $\bar{\boldsymbol{A}}$ in (11) with an extrapolated point. To describe it accurately, consider the algorithm description in Algorithm 1, and replace Step 7 by

$$\boldsymbol{A}^{t+1} = \left(\boldsymbol{H}^t - \frac{1}{\lambda}\boldsymbol{\Sigma}^{-1}(\boldsymbol{H}^t\boldsymbol{\Phi} - \boldsymbol{C})\right)_+,$$

where

$$\boldsymbol{H}^t = \boldsymbol{A}^t + \left(\frac{\gamma^t - 1}{\gamma^{t+1}}\right)(\boldsymbol{A}^t - \boldsymbol{A}^{t-1}),$$

$$\gamma^{t+1} = \frac{1 + \sqrt{1 + 4(\gamma^t)^2}}{2},$$

with $\gamma^0 = 1$. Note that for large t, γ^t approaches a constant and the algorithm will act almost like MM. It has been shown in other applications that such extrapolation can improve the convergence rate significantly [11, 21], and we found the same phenomena with our problem.

5. SIMULATION RESULTS

5.1. Synthetic Data Experiment

We first test our SAA-MM algorithm on synthetic data. We use the same probabilistic model as in Section 2 to generate the observed data $\boldsymbol{Y}$, with the following model parameters: data size $(M, L) = (50, 3000)$, basis rank $N = 5$, outlier probability $\beta = 0.01$, noise covariance $\boldsymbol{\Sigma} = \sigma^2\boldsymbol{I}$, concentration parameter $\boldsymbol{\alpha} = \boldsymbol{1}$ (which means that $\boldsymbol{s}_n$'s are uniformly distributed on the unit simplex), h being an element-wise i.i.d. uniform function with interval $[0, 1.6]$. Also, in each simulation trial, $\boldsymbol{A}$ is randomly generated following an element-wise i.i.d. uniform distribution on $[0, 1]$.

We employ the extended version of our SAA-MM algorithm described in Section 4 (i.e., it assumes unknown outlier probability β and non-negative $\boldsymbol{A}$); the sample size is $R = 500$, and the algorithm is initialized by another algorithm called SISAL [22]. The algorithms to be compared are SISAL, RVolMin [11], MVC-NMF [23], MVES [24] and Bayesian MCMC (B-MCMC) [25]. The parameters of these algorithms were tuned for optimized performance in our comparison. Also, for B-MCMC, 100 samples are used.

Fig. 1 shows the mean square errors (MSEs) of the estimated $\boldsymbol{A}$ provided by the various algorithms. We observe that the MSE performance of SAA-MM is better than that of the other algorithms, and the performance improvement is particularly significant for SNR $\leq$ 20dB. Another observation is that for SNR $\geq$ 15dB, the MSE performance of SAA-MM does not improve consistently with the SNR; instead it goes up slightly. The reason is due to the finite sample effects in SAA, and the effects may be reduced by increasing the sample size.

Table 1 shows the runtime performance of the various algorithms. The runtimes were evaluated on a desktop computer with Intel Core i7 3GHz CPU and 32GB memory, and under MATLAB. We see that SAA-MM has a higher runtime requirement than SISAL, RVolMin, MVC-NMF and MVES, although it is faster than B-MCMC.

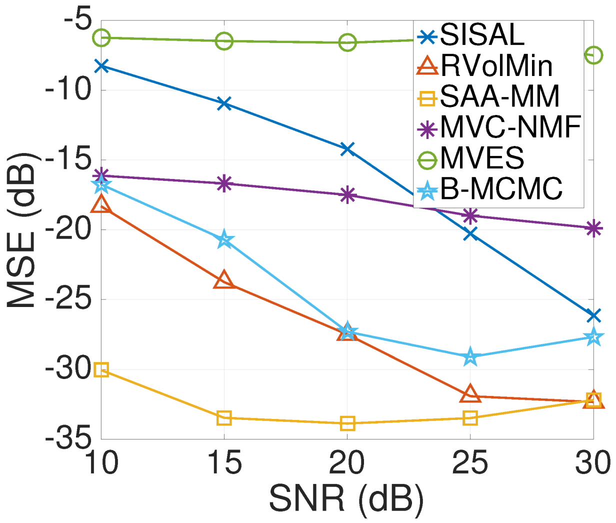


Fig. 1. MSE performance

Table 1. Runtime performance

Algorithm	SAA-MM	SISAL	RVolMin	MVC-NMF	MVES	B-MCMC
Time(s)	26.50	0.09	2.40	11.26	1.66	435.51

5.2. Real Data Experiment

We apply SAA-MM to hyperspectral unmixing (HU). Our experiment is based on a real hyperspectral remote-sensing image called AVIRIS Moffett Field 1997 [26]. The image is illustrated in Fig. 2. This hyperspectral image is composed of three main materials, namely, water, soil and vegetation (thereby $N = 3$). The HU problem aims at unmixing these materials and retrieving their abundance maps. According to the previous domain study [27], some pixels like those around the lakeshore are subjected to heavy nonlinear effects and can be seen as outliers.

Our experimental settings are similar to those in [11]. Additionally, we choose $\boldsymbol{\Sigma} = 0.1\boldsymbol{I}$, $\boldsymbol{\alpha} = \boldsymbol{1}$ and $R = 500$ for the SAA-MM algorithm. Fig. 3 illustrates the abundance maps and outlier map recovered by SAA-MM. The result is considered successful since the recovered maps look consistently with those in previous work [11, 27]. Also, most of the outliers identified by SAA-MM are around the lakeshore, which agrees with the domain study.

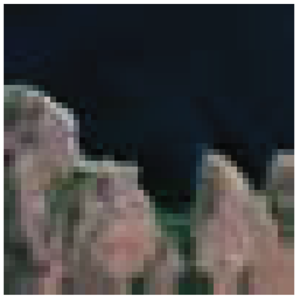
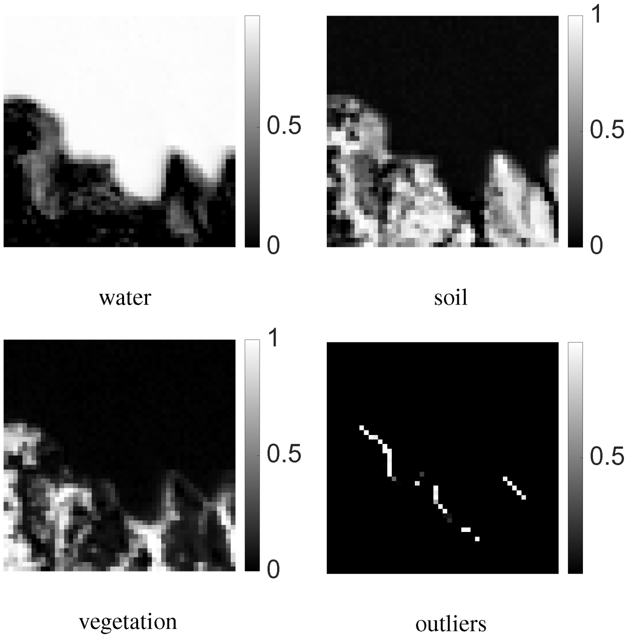


Fig. 2. "Moffett Field 1997" sub-image



water

soil

vegetation

outliers

Fig. 3. Abundance maps and outlier map

6. CONCLUSION AND DISCUSSION

In this paper we demonstrated the potential of using a stochastic ML framework to perform SSMF. In particular, numerical experiments showed that the ML approach gives promising estimation performance. It also outperforms several existing state-of-the-art methods when the data are noisy and contaminated by outliers. The drawback of our currently proposed ML algorithm is that it has relatively higher computational requirements than the other existing methods. At this point, we should mention that the running-time issue can be mitigated through parallel or multi-core computations since our algorithm (Algorithm 1) is quite amenable to such implementations. As future work, it would be worthwhile to study complexity reduction schemes for the ML framework.

7. REFERENCES

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

[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] 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.

[4] 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.

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

[6] N. Gillis and S. A. Vavasis, “Fast and robust recursive algorithmsfor separable nonnegative matrix factorization,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36, no. 4, pp. 698–714, 2014.

[7] E. Elhamifar, G. Sapiro, and R. Vidal, “See all by looking at a few: Sparse modeling for finding representative objects,” in *Proc. 2012 IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, 2012, pp. 1600–1607.

[8] 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.

[9] X. Fu, N. Sidiropoulos, and W.-K. Ma, “Power spectra separation via structured matrix factorization,” *IEEE Trans. Signal Process.*, vol. 64, no. 17, pp. 4592–4605, Sep. 2016.

[10] C.-H. Lin, W.-K. Ma, W.-C. Li, C.-Y. Chi, and A. Ambikapathi, “Identifiability of the simplex volume minimization criterion for blind hyperspectral unmixing: The no-pure-pixel case,” *IEEE Trans. Geosci. Remote Sens.*, vol. 53, no. 10, pp. 5530–5546, 2015.

[11] 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.

[12] 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.

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

[14] D. R. Hunter and K. Lange, “A tutorial on mm algorithms,” *The American Statistician*, vol. 58, no. 1, pp. 30–37, 2004.

[15] M. Razaviyayn, M. Y. Hong, and Z. Q. Luo, “A unified convergence analysis of block successive minimization methods for nonsmooth optimization,” *SIAM Journal on Optimization*, vol. 23, no. 2, pp. 1126–1153, 2013.

[16] G. C. Wei and M. A. Tanner, “A Monte Carlo implementation of the EM algorithm and the poor man’s data augmentation algorithms,” *Journal of the American Statistical Association*, vol. 85, no. 411, pp. 699–704, 1990.

[17] J. G. Booth and J. P. Hobert, “Maximizing generalized linear mixed model likelihoods with an automated monte carlo em algorithm,” *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, vol. 61, no. 1, pp. 265–285, 1999.

[18] M. Zibulevsky and M. Elad, “L1-l2 optimization in signal and image processing,” *IEEE Trans. Signal Process.*, vol. 27, no. 3, pp. 76–88, 2010.

[19] I. Daubechies, M. Defrise, and C. De Mol, “An iterative thresholding algorithm for linear inverse problems with a sparsity constraint,” *Communications on Pure and Applied Mathematics*, vol. 57, no. 11, pp. 1413–1457, 2004.

[20] Y. Nesterov, “Gradient methods for minimizing composite objective function,” UCL, Tech. Rep., 2007.

[21] C. Qian, X. Fu, N. D. Sidiropoulos, L. Huang, and J. H. Xie, “Inexact alternating optimization for phase retrieval in the presence of outliers,” *arXiv preprint arXiv:1605.00973*, 2016.

[22] J. M. Bioucas-Dias, “A variable splitting augmented lagrangian approach to linear spectral unmixing,” in *Proc. First Workshop Hyperspectral Image and Signal Processing: Evolution in Remote Sensing*, Aug. 2009, pp. 1–4.

[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] T. H. Chan, C. Y. Chi, Y. M. Huang, and W. K. Ma, “A convex analysis-based minimum-volume enclosing simplex algorithm for hyperspectral unmixing,” *IEEE Trans. Signal Process.*, vol. 57, no. 11, pp. 4418–4432, Nov. 2009.

[25] 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.

[26] A. F. Data, “Jet propulsion lab,” *California Inst. Technol., Pasadena.[Online]. Available: http://aviris. jpl. nasa. gov/html/aviris. freedata. html.*

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