Probabilistic Simplex Component Analysis

Ruiyuan Wu[†], Wing-Kin Ma[†], Yuening Li[†], Anthony Man-Cho So[‡],
and Nicholas D. Sidiropoulos[⋆]

[†]Department of Electronic Engineering, The Chinese University of Hong Kong,
Hong Kong SAR of China

[‡]Department of Systems Engineering and Engineering Management,
The Chinese University of Hong Kong, Hong Kong SAR of China

[⋆]Department of Electrical and Computer Engineering,
University of Virginia, Charlottesville, Virginia 22904, USA

January 20, 2022

Abstract

This study presents PRISM, a probabilistic simplex component analysis approach to identifying the vertices of a data-circumscribing simplex from data. The problem has a rich variety of applications, the most notable being hyperspectral unmixing in remote sensing and non-negative matrix factorization in machine learning. PRISM uses a simple probabilistic model, namely, uniform simplex data distribution and additive Gaussian noise, and it carries out inference by maximum likelihood. The inference model is sound in the sense that the vertices are provably identifiable under some assumptions, and it suggests that PRISM can be effective in combating noise when the number of data points is large. PRISM has strong, but hidden, relationships with simplex volume minimization, a powerful geometric approach for the same problem. We study these fundamental aspects, and we also consider algorithmic schemes based on importance sampling and variational inference. In particular, the variational inference scheme is shown to resemble a matrix factorization problem with a special regularizer, which draws an interesting connection to the matrix factorization approach. Numerical results are provided to demonstrate the potential of PRISM.

Keywords: Simplex-structured matrix factorization, maximum likelihood, identifiability, simplex volume minimization, variational inference, hyperspectral unmixing

Acknowledgments

This work is dedicated to the late professor José Bioucas-Dias. Wing-Kin Ma is greatly indebted to him for his many inspirations, support, encouragement, and occasionally hard time in the form of thought-provoking challenges; the same goes for the many wonderful interactions with him over a decade. Had they not met, this work could have never existed.

1 Introduction

Consider this problem: We have a collection of multi-dimensional data points that are circumscribed by a simplex; see Fig. 1 for an illustration. Can we learn the vertices of that simplex from the data points? The pursuit of a solution to such vertex-finding problem is termed *simplex component analysis* in this study. In particular, we will consider a probabilistic simplex component analysis approach that employs the same inference formulation as probabilistic principal component analysis (PCA) [1] and independent component analysis (ICA) [2–4].

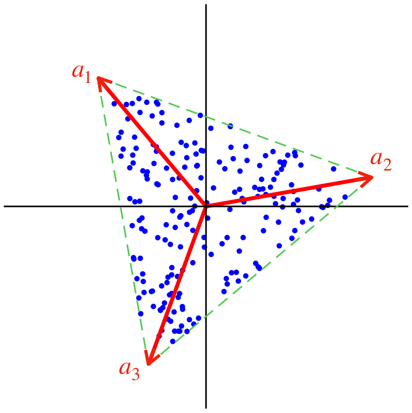
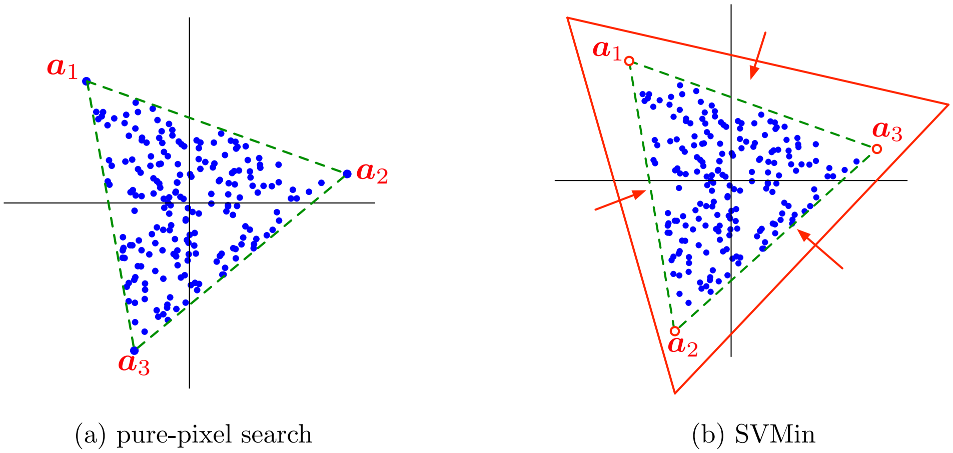


Figure 1: Illustration of simplex component analysis by a scatterplot. Blue dots: data points $\boldsymbol{y}_t$'s, green line: the data circumscribing simplex, red arrows: vertices $\boldsymbol{a}_i$'s of the data circumscribing simplex.

1.1 Background and State of the Art

Simplex component analysis arose in different fields and has many names. It appears in hyperspectral unmixing (HU) in remote sensing [5,6], a topic that has more than 30 years of history [7]. It emerges in non-negative matrix factorization (NMF) in machine learning [8–10], with application to topic modeling. Some other areas also stumbled on the same problem; see, e.g., [6,9,10] and the references therein, for details. It is now recognized that simplex component analysis covers a rich variety of applications—in addition to the aforementioned HU and topic modeling applications, it has been applied to biomedical imaging [11], blind audio source separation [12], finding representatives from data in computer vision [13], community detection [14,15], and crowdsourcing [16], to name a few.

In solving the simplex component analysis problem, the majority of the existing studies follow, or turn out to be related to, the notion of *convex geometry* (CG). Its idea is to exploit certain geometric structures with the data points. There are two main approaches with CG. One, called *pure-pixel search* in HU or *separable NMF* in machine learning, assumes that some data points lie exactly at the vertices; see Fig. 2(a) for an illustration. The problem is then to identify those vertex points algorithmically; see [6,8–10,13,17–22] for the many different ways to do so. Another, called *simplex volume minimization* (SVMin), amounts to finding a simplex that encloses the data points and yields the minimum volume [23–25]. As visualized in Fig. 2(b), the minimum volume data-enclosing simplex seems to coincide with the true data simplex when the data points are adequately well-spread on the simplex. This intuition has recently been confirmed to be mathematically sound—SVMin can identify the vertices under a geometric assumption that is likely to hold for sufficiently well-spread data points [12, 26, 27]. That geometric assumption is much more relaxed than the assumption of having vertex points in pure-pixel search or separable NMF, and thus SVMin is arguably more powerful. We also refer the reader to [28, 29] for a few other original geometric approaches.



(a) pure-pixel search

(b) SVMin

Figure 2: Illustration of the CG concepts. Blue dots: data points $\boldsymbol{y}_t$'s, green line: the true data circumscribing simplex, red line: a data enclosing simplex.

While the notions of CG are elegant, they were established under the noiseless case—at least in the beginning of most of the developments. In the noisy case some researchers developed "provably good" schemes by equipping their CG algorithms with recovery accuracy analyses, typically under the separable NMF approach; see, e.g., [8, 20–22]. Such analyses are fundamentally intriguing in pinning down the noise robustness of separable NMF. Others altered the formulations to make the solutions more robust against noise in practice, and this is more often seen for SVMin; see, e.g., [19,27,30–32]. Such alternations usually introduce new parameters, typically for regularization. Those parameters are usually tuned in a manual fashion, with no strong theory to guide.

Probabilistic approaches, such as Bayesian and maximum-likelihood (ML) inference, are arguably more pertinent when there is noise. In HU we have seen applications of probabilistic methods to simplex component analysis, and here we mention two representative developments. Dobigeon *et. al* [33] studied Bayesian inference. The difficulty in that work is that some probability density functions (PDFs) appear as intractable integrals, and the issue is tackled by Markov chain Monte Carlo (MCMC) sampling which is known to be computationally expensive for large problem dimensions. Nascimento and Bioucas-Dias studied an ML inference scheme called dependent component analysis (DECA) [34]. DECA considers the noiseless case, and it employs expectation maximization to realize ML.

1.2 Contributions of the Present Study

In this study, we are interested in probabilistic simplex component analysis. Named PRISM (PRobabIlistic SiMplex), our approach considers ML inference under the model of uniform simplex distribution with the noise-free components and Gaussian distribution with noise. The ML inference formulation is principally the same as that by DECA, but with a key difference—*noise*. The likelihood function in the noiseless case, or in DECA, has a closed form. In the noisy case we no longer have this prestige—the likelihood function appears as an integral that has no known analytical expression in general. This is an obstacle to be overcome or circumvented.

Our study spans theoretical identifiability, drawing connections to the CG approach, and exploration of algorithmic realizations. Our contributions are summarized as follows.

1) Identifiability: Can PRISM correctly identify the vertices, particularly in the noisy case? Understanding identifiability is vital in confirming whether we are working on a sound inference model, and it has been a key aspect in the CG studies. We will confirm that the answer is yes in theory, as far as we have a large amount of data points (technically, infinite). We pin down the identifiability result by leveraging insight from a recent paper on ICA identifiability [4]. The main challenge lies in making the proof mathematically rigorous.

2) Connections with SVMin in CG: We will show that several representative SVMin methods in CG can be derived from PRISM, either in the noiseless case or as approximations in the noisy case. These relationships, which are not obvious at first sight, enrich our understanding—the deterministic CG and the stochastic PRISM are not disparate subjects; rather, they are intimately related.

We are obliged to commend Bioucas-Dias who informally, but insightfully, mentioned one of the aforementioned relationships in his WHISPERS 2009 oral presentation [35]. Our task entails consolidating and expanding his initial insight to discover more connections. It is interesting to note that, coincidentally, Dobigeon *et al.* touched on a result similar to Bioucas-Dias' reporting in the same year [33, Appendix].

3) New Algorithmic Schemes: We take inspiration from statistical inference and develop two schemes to realize PRISM algorithmically. The first is importance sampling approximation (ISA) via Monte Carlo expectation maximization [36]. The second, which is arguably more interesting, is variational inference approximation (VIA) [37]. VIA has recently become popular in statistics, data science and machine learning, and its idea is to approximate the intractable likelihood function by optimization. We will propose, and custom-derive, a VIA scheme for PRISM. Also we will reveal a connection between VIA-PRISM and matrix factorization—VIA-PRISM can be seen as an instance of simplex-structured matrix factorization, with a special regularization not seen in the previous matrix factorization literature.

1.3 Comparison with Related Studies and Organization

Let us further elaborate on the similarities and differences between PRISM and DECA. As mentioned, the key difference is that PRISM and DECA consider the noisy and noiseless cases, respectively. Another difference is in their respective aims. In PRISM we aim to understand basic aspects by adopting a plain uniform simplex distribution model. In DECA the authors want to learn complex phenomena of real-world data by applying a complex Dirichlet mixture (and non-uniform) distribution model. Furthermore, and beyond the scope of simplex component analysis, the inference formulations of PRISM and DECA fall into the same genre as those in probabilistic PCA [1] and ICA [2–4].[1]

[1] Specifically, they all employ latent variable models, and each postulates a different latent prior—independent Gaussian for probabilistic PCA, independent non-Gaussian for probabilistic ICA, simplex-uniform for PRISM, and Dirichlet mixture for DECA.

Since we will consider VIA for PRISM, it is worth noting that VIA was used in related contexts such as ICA [3], latent Dirichlet allocation [38] and nonlinear (or deep) ICA [4]. However, the VIA of PRISM takes a different appearance from those of the previous studies, due to the different model. In fact, technically we will need to devise a specialized method to solve the new optimization problem arising from VIA-PRISM.

We should note that most of the results to be presented appear for the first time; the exception is the results for ISA, which were reported in conferences [39–41]. The nature of our study is fundamental, exploring the potential of PRISM as a framework. We will focus less on computational or implementation aspects, which will be future work.

This article is organized as follows. Section 2 concisely reviews the necessary concepts to understand this work. Section 3 describes the PRISM model and formulation. Section 4 studies the PRISM identifiability. Section 5 shows the hidden relationships of PRISM and SVMin. Sections 6–7 turn to stochastic and variational approximations of PRISM and suggest algorithmic solutions. This is followed by a set of numerical experiments in Section 8, and then by conclusions in Section 9.

2 Preliminaries

2.1 Notations

Most of our notations are standard. Vectors and matrices are represented by boldfaced lowercase and capital letters, e.g., $\boldsymbol{x}$ and $\boldsymbol{X}$, respectively (resp.); unless otherwise specified, $\boldsymbol{x}_i$ denotes the ith column of $\boldsymbol{X}$; $\mathbb{R}, \mathbb{R}_+, \mathbb{R}_{++}$ are the sets of all real, non-negative and positive numbers, resp.; given $\mathcal{X} \subseteq \mathbb{R}^n$, $\boldsymbol{y} \in \mathbb{R}^n$, we denote $\mathcal{X} + \boldsymbol{y} = \{\boldsymbol{x} + \boldsymbol{y} \mid \boldsymbol{x} \in \mathcal{X}\}$; the superscripts $^\top$, $^{-1}$ and † denote transpose, inverse and pseudo-inverse, resp.; $\boldsymbol{x} = (x_1, \ldots, x_n)$ means that $\boldsymbol{x} = [\ x_1, \ldots, x_n\ ]^\top$; $\|\cdot\|$ denotes the Euclidean norm; $\mathrm{tr}(\cdot)$ is the trace of a matrix; $\mathrm{Diag}(\boldsymbol{x})$ is a diagonal matrix whose (i,i)th element is x_i; $\boldsymbol{0}$ is an all-zero vector; $\boldsymbol{1}$ is an all-one vector; $\boldsymbol{I}$ is an identity matrix; $\boldsymbol{e}_i$ is a unit vector, i.e., $\boldsymbol{e}_i = (0, \ldots, 0, 1, 0, \ldots, 0)$ with 1 being at the ith element; given $\boldsymbol{A} \in \mathbb{R}^{m \times n}$,

$$\begin{aligned} \mathrm{span}(\boldsymbol{A}) &= \{\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x} \mid \boldsymbol{x} \in \mathbb{R}^n\}, \\ \mathrm{aff}(\boldsymbol{A}) &= \{\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x} \mid \boldsymbol{x} \in \mathbb{R}^n, \boldsymbol{1}^\top \boldsymbol{x} = 1\}, \\ \mathrm{conv}(\boldsymbol{A}) &= \{\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x} \mid \boldsymbol{x} \in \mathbb{R}^n_+, \boldsymbol{1}^\top \boldsymbol{x} = 1\} \end{aligned}$$

denote the span, affine hull and convex hull of $\{\boldsymbol{a}_1, \ldots, \boldsymbol{a}_n\}$, resp.; as a less standard notation,

$$\overline{\mathrm{conv}}(\boldsymbol{A}) = \{\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x} \mid \boldsymbol{x} \in \mathbb{R}^n_{++}, \boldsymbol{1}^\top \boldsymbol{x} = 1\}$$

denotes the open convex hull of $\{\boldsymbol{a}_1, \ldots, \boldsymbol{a}_n\}$ (it is open on $\mathrm{aff}(\boldsymbol{A})$, not on $\mathbb{R}^m$); $\mathbb{1}_{\mathcal{X}}$ is the indicator function of a set $\mathcal{X}$:

$$\mathbb{1}_{\mathcal{X}}(\boldsymbol{x}) = \left\{ \begin{array}{ll} 1, & \boldsymbol{x} \in \mathcal{X} \\ 0, & \boldsymbol{x} \notin \mathcal{X} \end{array} \right. ;$$

$\boldsymbol{x} \sim D$ means that $\boldsymbol{x}$ is a random variable with distribution D; $\mathbb{E}_{\boldsymbol{x} \sim p}[\cdot]$ denotes expectation of a random variable $\boldsymbol{x}$ with distribution p; $\mathrm{var}(x)$ and $\mathrm{Cov}(\boldsymbol{x})$ denote the covariance of a random variable x and $\boldsymbol{x}$, resp.

The following specialized notations will be used frequently. Given $\boldsymbol{A} \in \mathbb{R}^{m \times n}$, $\boldsymbol{s} \in \mathbb{R}^n$, we define, resp.,

$$\bar{\boldsymbol{A}} = [\ \boldsymbol{a}_1 - \boldsymbol{a}_n, \ldots, \boldsymbol{a}_{n-1} - \boldsymbol{a}_n\ ], \quad \bar{\boldsymbol{s}} = (s_1, \ldots, s_{n-1}). \tag{1}$$

2.2 Simplex

To describe simplex, we need to first review affine independence. Let $\boldsymbol{A} \in \mathbb{R}^{m \times n}$. We say that $\boldsymbol{A}$ is affinely independent if $\bar{\boldsymbol{A}}$ (cf. (1)) has full column rank. A simplex is defined as a convex hull $\mathrm{conv}(\boldsymbol{A})$ with affinely independent $\boldsymbol{A}$. Additionally, a simplex is called full-dimensional if $m = n-1$. A simplex $\mathrm{conv}(\boldsymbol{A})$ has the following properties: its set of vertices is $\{\boldsymbol{a}_1, \ldots, \boldsymbol{a}_n\}$; its volume, according to [42], is

$$\mathrm{vol}(\boldsymbol{A}) = \frac{(\det(\bar{\boldsymbol{A}}^\top \bar{\boldsymbol{A}}))^{1/2}}{(n-1)!}. \tag{2}$$

2.3 Dirichlet Distribution

The Dirichlet distribution is commonly used to model on-unit-simplex random variables [43, 44]. Let

$$\Delta = \{\boldsymbol{s} \in \mathbb{R}^N_+ \mid \mathbf{1}^\top \boldsymbol{s} = 1\}, \quad \bar{\Delta} = \{\boldsymbol{s} \in \mathbb{R}^N_{++} \mid \mathbf{1}^\top \boldsymbol{s} = 1\}$$

be the unit simplex of $\mathbb{R}^N$ and its open counterpart, resp. A random variable $\boldsymbol{s} \in \Delta$ is said to be Dirichlet distributed with concentration parameter $\boldsymbol{\alpha} \in \mathbb{R}^N_{++}$, or simply $\boldsymbol{\alpha}$-Dirichlet distributed, if $\bar{\boldsymbol{s}} = (s_1, \ldots, s_{N-1})$ has the density

$$D(\bar{\boldsymbol{s}}; \boldsymbol{\alpha}) = \frac{1}{B(\boldsymbol{\alpha})} \left(\prod_{i=1}^{N-1} s_i^{\alpha_i - 1} \right) \left(1 - \sum_{i=1}^{N-1} s_i \right)^{\alpha_N - 1} \mathbb{1}_{\tilde{\Delta}}(\bar{\boldsymbol{s}}), \tag{3}$$

where

$$\tilde{\Delta} = \{\bar{\boldsymbol{s}} \in \mathbb{R}^{N-1}_{++} \mid 1 - \mathbf{1}^\top \bar{\boldsymbol{s}} > 0\};$$

$B(\boldsymbol{\alpha}) = (\prod_{i=1}^N \Gamma(\alpha_i))/\Gamma(\sum_{i=1}^N \alpha_i)$; $\Gamma(x) = \int_0^\infty t^{x-1} e^{-t} \mathrm{d}t$ is the Gamma function. Note that $\bar{\boldsymbol{s}}$ is the truly operating random variable, as the last element $s_N = 1 - \mathbf{1}^\top \bar{\boldsymbol{s}}$ of $\boldsymbol{s}$ is completely determined by $\bar{\boldsymbol{s}}$. For convenience, however, it is common to write

$$D(\boldsymbol{s}; \boldsymbol{\alpha}) = \frac{1}{B(\boldsymbol{\alpha})} \left(\prod_{i=1}^{N} s_i^{\alpha_i - 1} \right) \mathbb{1}_{\bar{\Delta}}(\boldsymbol{s}) \tag{4}$$

and write $\boldsymbol{s} \sim D(\cdot; \boldsymbol{\alpha})$ to specify a Dirichlet random variable. The parameter $\boldsymbol{\alpha}$ governs the shape of the Dirichlet distribution, and the reader is referred to the literature [43, 44] for illustrations. A well-known case is

$$D(\boldsymbol{s}; \mathbf{1}) = (N-1)! \cdot \mathbb{1}_{\bar{\Delta}}(\boldsymbol{s}),$$

which is the uniform unit-simplex distribution.

The Dirichlet distribution has a number of friendly properties. First, it is easy to generate samples from it [43]. Second, many of its moments admit explicit expressions [44].

Fact 1 (Dirichlet moments; see, e.g., [44]) *Let $\boldsymbol{s} \sim D(\cdot; \boldsymbol{\alpha})$. We have*

(a) $\mathbb{E}[\boldsymbol{s}] = \tilde{\boldsymbol{\alpha}}$, where $\tilde{\boldsymbol{\alpha}} = \boldsymbol{\alpha}/\alpha_0$, $\alpha_0 = \sum_{i=1}^N \alpha_i$;

(b) the covariance of $\boldsymbol{s}$ is

$$\mathrm{Cov}(\boldsymbol{s}) = \frac{1}{1+\alpha_0} (\mathrm{Diag}(\tilde{\boldsymbol{\alpha}}) - \tilde{\boldsymbol{\alpha}} \tilde{\boldsymbol{\alpha}}^\top);$$

(c) the entropy of $\boldsymbol{s}$ equals

$$\begin{aligned} H(\boldsymbol{s}) &:= \mathbb{E}[-\log D(\boldsymbol{s};\boldsymbol{\alpha})] \\ &= \log B(\boldsymbol{\alpha}) - \sum_{i=1}^{N} (\alpha_i - 1)(\psi(\alpha_i) - \psi(\alpha_0)), \end{aligned}$$

where $\psi(x) = \frac{\mathrm{d} \log \Gamma(x)}{\mathrm{d}x}$ is the digamma function.

Third, the Dirichlet distribution can be used to construct distributions on a simplex.

Fact 2 (uniform distribution on a full-dimensional simplex) *Let $\boldsymbol{x} = \boldsymbol{B}\boldsymbol{s}$, where $\boldsymbol{B} \in \mathbb{R}^{(N-1)\times N}$ is affinely independent and $\boldsymbol{s} \sim D(\cdot, \boldsymbol{1})$. The PDF of $\boldsymbol{x}$ is*

$$p(\boldsymbol{x}) = \frac{1}{\mathrm{vol}(\boldsymbol{B})} \mathbb{1}_{\overline{\mathrm{conv}}(\boldsymbol{B})}(\boldsymbol{x}). \tag{5}$$

The proof of Fact 2 is shown in Appendix A.

We will encounter integration involving the Dirichlet distribution. The problem, in a general sense, is to integrate a function $f : \mathbb{R}^N \to \mathbb{R}$ over $\mathcal{A} = \{\boldsymbol{s} \in \mathbb{R}^N \mid \boldsymbol{1}^\top \boldsymbol{s} = 1\}$. A proper way to do so is

$$\int_{\mathbb{R}} \cdots \int_{\mathbb{R}} f(s_1, \ldots, s_{N-1}, 1 - \textstyle\sum_{i=1}^{N-1} s_i) \mathrm{d}s_1 \cdots \mathrm{d}s_{N-1}, \tag{6}$$

where we apply $s_N = 1 - \sum_{i=1}^{N-1} s_i$ directly. Writing out (6) is clumsy, and we will use the Lebesgue integral

$$\int f(\boldsymbol{s}) \mathrm{d}\mu(\boldsymbol{s})$$

to compactly represent (6); here μ is the Lebesgue measure on $\mathcal{A}$.

3 PRISM Formulation

As described in the Introduction and as illustrated in Fig. 1, we consider the following problem: We have a collection of data points $\boldsymbol{y}_1, \ldots, \boldsymbol{y}_T \in \mathbb{R}^M$ that are posited to distribute on a simplex. The vertices undergirding the simplex are unknown, and there is noise in the data points. Our aim is to identify the vertices from the data points. In our probabilistic simplex component analysis approach, or PRISM, we model the data points as

$$\boldsymbol{y}_t = \boldsymbol{x}_t + \boldsymbol{v}_t, \quad \boldsymbol{x}_t = \boldsymbol{A}_0 \boldsymbol{s}_t \in \mathrm{conv}(\boldsymbol{A}_0), \quad t = 1, \ldots, T, \tag{7}$$

where $\boldsymbol{x}_t \in \mathbb{R}^M$ is the noise-free part of $\boldsymbol{y}_t$; $\boldsymbol{A}_0 \in \mathbb{R}^{M \times N}$ collects the vertices of the simplex and is called the *true vertex matrix*; $\boldsymbol{s}_t \in \Delta$ is a latent variable; $\boldsymbol{v}_t \in \mathbb{R}^M$ is noise. The model (7) is accompanied by the following assumptions:

Assumption 1 *The matrix $\boldsymbol{A}_0$ is affinely independent.*

Assumption 2 *The latent variables $\boldsymbol{s}_t$'s are independently and identically distributed (i.i.d.). Every $\boldsymbol{s}_t$ is uniformly distributed on $\bar{\Delta}$, or, $\boldsymbol{1}$-Dirichlet distributed.*

Assumption 3 *The noise variables $\boldsymbol{v}_t$'s are i.i.d. and independent of the $\boldsymbol{s}_t$'s. Every $\boldsymbol{v}_t$ is Gaussian distributed with mean $\mathbf{0}$ and covariance $\sigma^2\boldsymbol{I}$, $\sigma > 0$.*

Note that we consider a basic model wherein the $\boldsymbol{s}_t$'s are treated as uniformly distributed random variables. Such model is arguably reasonable when we have no prior information on the latent-variable distribution.

PRISM considers the following ML inference

$$\hat{\boldsymbol{A}}_{\mathsf{ML}} \in \arg \max_{\boldsymbol{A} \in \mathbb{R}^{M \times N}} \mathcal{L}_T(\boldsymbol{A}) := \frac{1}{T} \sum_{t=1}^{T} \log p(\boldsymbol{y}_t; \boldsymbol{A}), \tag{8}$$

where $p(\boldsymbol{y}; \boldsymbol{A})$ is the PDF of a data point $\boldsymbol{y}$ parameterized by $\boldsymbol{A}$. Under the above data model, $p(\boldsymbol{y}; \boldsymbol{A})$ is given by

$$\begin{aligned} p(\boldsymbol{y}; \boldsymbol{A}) &= \int p(\boldsymbol{y} | \boldsymbol{s}; \boldsymbol{A}) p(\boldsymbol{s}) \mathrm{d}\mu(\boldsymbol{s}) \\ &= (N-1)! \int \varphi_\sigma(\boldsymbol{y} - \boldsymbol{A}\boldsymbol{s}) \mathbb{1}_{\bar{\Delta}}(\boldsymbol{s}) \mathrm{d}\mu(\boldsymbol{s}), \end{aligned} \tag{9}$$

where $\varphi_\sigma(\boldsymbol{y}) = e^{-\|\boldsymbol{y}\|^2/2\sigma^2}/(\sqrt{2\pi}\sigma)^M$ is a multivariate i.i.d. Gaussian function; $p(\boldsymbol{s})$ is the PDF of a latent variable $\boldsymbol{s}$; $p(\boldsymbol{y}|\boldsymbol{s}; \boldsymbol{A})$ is the PDF of a data point $\boldsymbol{y}$ conditioned on $\boldsymbol{s}$ and parameterized by $\boldsymbol{A}$. There is no known closed-form solution for the integral (9) in general. The intractability of (9) presents a challenge for realizing the ML estimator (8), which we shall address.

The reader may wonder: How about the ML alternative of maximizing the log likelihood over both $\boldsymbol{A}$ and $\boldsymbol{s}_1, \ldots, \boldsymbol{s}_T$? To be precise, consider modeling $\boldsymbol{S} = [\ \boldsymbol{s}_1, \ldots, \boldsymbol{s}_T\ ]$ as a deterministic unknown with simplex support $\Delta^T = \{\boldsymbol{S} \in \mathbb{R}^{N \times T} \mid \boldsymbol{s}_t \in \Delta, \forall t\}$, and deal with the ML estimator

$$\max_{\boldsymbol{A} \in \mathbb{R}^{M \times N}, \boldsymbol{S} \in \Delta^T} \log p(\boldsymbol{Y}; \boldsymbol{S}, \boldsymbol{A}),$$

where $p(\boldsymbol{Y}; \boldsymbol{S}, \boldsymbol{A})$ is the PDF of $\boldsymbol{Y} = [\ \boldsymbol{y}_1, \ldots, \boldsymbol{y}_T\ ]$ parameterized by $\boldsymbol{A}$ and $\boldsymbol{S}$. It is easy to show that the above ML problem equals

$$\min_{\boldsymbol{A} \in \mathbb{R}^{M \times N}, \boldsymbol{S} \in \Delta^T} \|\boldsymbol{Y} - \boldsymbol{A}\boldsymbol{S}\|^2, \tag{10}$$

which is a simplex-structured matrix factorization (SSMF) problem and looks easier to handle than the ML estimator (8). But there is an issue.

Fact 3 *Let $\boldsymbol{R}$ be any invertible matrix in Δ^N. If $(\boldsymbol{A}, \boldsymbol{S})$ is a solution to the SSMF problem* (10)*, then $(\boldsymbol{A}\boldsymbol{R}^{-1}, \boldsymbol{R}\boldsymbol{S})$ is also a solution to the SSMF problem* (10)*.*

The proof of Fact 3 is trivial and omitted for brevity. Fact 3 indicates that, even without noise, a solution to problem (10) does not necessarily equal the true vertex matrix $\boldsymbol{A}_0$ or its column permuted counterparts.[2]

[2] Note that Fact 3 assumes general $\boldsymbol{A}$. In the NMF context it is known that if $\boldsymbol{A}$ is non-negative and we incorporate non-negative constraints with $\boldsymbol{A}$ in the matrix factorization problem (10), then (10) may provide some form of identifiability guarantees; see, e.g., [10] and the references therein.

4 ML Identifiability

While the ML estimator (10) fails to guarantee identifiable solutions with the vertices, we will show that the more difficult ML estimator (8) can provide identifiable solutions.

4.1 The Identifiability Result

The identifiability problem in question is classic in statistical inference. Consider $T \to \infty$ such that, by the law of large numbers, the log likelihood function $\mathcal{L}_T$ in (8) converges to

$$\mathcal{L}(\boldsymbol{A}) = \mathbb{E}_{\boldsymbol{y} \sim p(\cdot;\boldsymbol{A}_0)}[\log p(\boldsymbol{y};\boldsymbol{A})] = \int_{\mathbb{R}^M} p(\boldsymbol{y};\boldsymbol{A}_0) \log p(\boldsymbol{y};\boldsymbol{A}) \mathrm{d}\boldsymbol{y}.$$

Consider the ML problem

$$\max_{\boldsymbol{A} \in \mathbb{R}^{M \times N}} \mathcal{L}(\boldsymbol{A}), \tag{11}$$

which may intuitively be seen as the ML problem (8) for large data size T. By Kullback-Leibler divergence, we have

$$\mathcal{L}(\boldsymbol{A}_0) \geq \mathcal{L}(\boldsymbol{A}), \tag{12}$$

where equality in (12) holds if and only if

$$p(\boldsymbol{y};\boldsymbol{A}_0) = p(\boldsymbol{y};\boldsymbol{A}), \quad \text{for all } \boldsymbol{y}. \tag{13}$$

Eqs. (12)–(13) suggest that the true vertex matrix $\boldsymbol{A}_0$ is an ML solution in (11), and $\boldsymbol{A}$ is an ML solution in (11) if and only if (13) holds. Hence, our identifiability problem is to confirm whether (13) does *not* hold for any non-trivial choice of $\boldsymbol{A}$. Our identifiability result is shown below.

Theorem 1 *Eq.* (13) *holds if and only if $\boldsymbol{A} = \boldsymbol{A}_0\boldsymbol{\Pi}$, where $\boldsymbol{\Pi}$ is a permutation matrix. Consequently, $\boldsymbol{A}$ is a solution to the ML problem* (11) *if and only if $\boldsymbol{A} = \boldsymbol{A}_0\boldsymbol{\Pi}$.*

Note that the underlying assumptions with Theorem 1 are Assumptions 1–3 and $T \to \infty$. Theorem 1 confirms that the ML estimator (11) can exactly identify the vertices. It also gives an intuitive implication that the finite-data ML estimator (8) should suppress the impact of noise better as we have more data points. In the next subsection we will show an intuitive proof of Theorem 1 to provide insight. The formal proof of Theorem 1 is relegated to Appendix B.

Remark 1 It is natural to question whether the PRISM identifiability result in Theorem 1 provides new insights compared to the known CG identifiability results [8,12,20–22,26–29]. Analyses in CG and PRISM operate under different assumptions (one deterministic, another stochastic), and it is hard to compare fairly. Still, let us compare one aspect, namely, whether we can reduce the impact of noise by increasing the data size T. The currently available analyses in CG are unable to confirm the aforementioned aspect, although they can confirm how noise-robust an algorithm is in the worst-case sense and for any T. The PRISM identifiability result in Theorem 1, in comparison, requires infinite T but can confirm elimination of the noise effects under infinite T.

4.2 An Intuitive Proof of ML Identifiability

The intuitive proof of Theorem 1 is as follows. From (9), it is easy to see that $\boldsymbol{A} = \boldsymbol{A}_0\boldsymbol{\Pi}$ implies $p(\boldsymbol{y}; \boldsymbol{A}) = p(\boldsymbol{y}; \boldsymbol{A}_0)$. To show the converse, consider the special case of $M = N - 1$ and affinely independent $\boldsymbol{A}$. By Fact 2, the noise-free components $\boldsymbol{x}_t$'s in (7) follow a uniform simplex distribution

$$p(\boldsymbol{x}; \boldsymbol{A}) = \frac{1}{\mathrm{vol}(\boldsymbol{A})} \mathbb{1}_{\overline{\mathrm{conv}}(\boldsymbol{A})}(\boldsymbol{x}). \tag{14}$$

Applying (14) to the model (7), we can write

$$p(\boldsymbol{y}; \boldsymbol{A}) = \int_{\mathbb{R}^{N-1}} \varphi_\sigma(\boldsymbol{y} - \boldsymbol{x}) p(\boldsymbol{x}; \boldsymbol{A}) \mathrm{d}\boldsymbol{x}. \tag{15}$$

By defining the Fourier transform (FT) of $f : \mathbb{R}^n \to \mathbb{R}$ as $\check{f}(\boldsymbol{\xi}) = \int_{\mathbb{R}^n} f(\boldsymbol{x}) e^{-j2\pi\boldsymbol{\xi}^\top \boldsymbol{x}} \mathrm{d}\boldsymbol{x}$ and the inverse FT as $\int_{\mathbb{R}^n} \check{f}(\boldsymbol{\xi}) e^{j2\pi\boldsymbol{\xi}^\top \boldsymbol{x}} \mathrm{d}\boldsymbol{\xi}$, we have

$$p(\boldsymbol{y}; \boldsymbol{A}) = p(\boldsymbol{y}; \boldsymbol{A}_0) \;\text{ for all } \boldsymbol{y} \tag{16a}$$
$$\Longrightarrow \check{\varphi}_\sigma(\boldsymbol{\xi}) \check{p}(\boldsymbol{\xi}; \boldsymbol{A}) = \check{\varphi}_\sigma(\boldsymbol{\xi}) \check{p}(\boldsymbol{\xi}; \boldsymbol{A}_0) \;\text{ for all } \boldsymbol{\xi} \tag{16b}$$
$$\Longrightarrow \check{p}(\boldsymbol{\xi}; \boldsymbol{A}) = \check{p}(\boldsymbol{\xi}; \boldsymbol{A}_0) \;\text{ for all } \boldsymbol{\xi} \tag{16c}$$
$$\Longrightarrow p(\boldsymbol{x}; \boldsymbol{A}) = p(\boldsymbol{x}; \boldsymbol{A}_0) \;\text{ for all } \boldsymbol{x} \tag{16d}$$
$$\Longrightarrow \overline{\mathrm{conv}}(\boldsymbol{A}) = \overline{\mathrm{conv}}(\boldsymbol{A}_0) \tag{16e}$$
$$\Longrightarrow \mathrm{conv}(\boldsymbol{A}) = \mathrm{conv}(\boldsymbol{A}_0) \tag{16f}$$
$$\Longrightarrow \{\boldsymbol{a}_1, \ldots, \boldsymbol{a}_N\} = \{\boldsymbol{a}_{0,1}, \ldots, \boldsymbol{a}_{0,N}\}. \tag{16g}$$

Here, (16b) is obtained by taking FT on both sides of (16a) and by noting the convolution relation in (15); (16c) is due to the fact that $\check{\varphi}_\sigma(\boldsymbol{\xi}) = e^{-2\pi^2\|\boldsymbol{\xi}\|^2} \neq 0$ for all $\boldsymbol{\xi}$; (16d) is obtained by taking inverse FT on both sides of (16c); (16e) is the direct consequence of (14); (16f) is obtained by taking closure on both sides of (16e); (16g) is due to the fact that the set of all vertices of $\mathrm{conv}(\boldsymbol{A})$ is $\{\boldsymbol{a}_1, \ldots, \boldsymbol{a}_N\}$. Our intuitive proof is complete.

The above intuitive proof takes insight from the ICA identifiability proof in [4, Theorem 1], particularly, the FT and inverse FT steps in (16b)–(16d). In the formal proof, shown in Appendix B, we will generalize the result to any $\boldsymbol{A}$ and to any $M \geq N - 1$. Also we will fix a subtle issue—namely, using inverse FT to obtain (16d) is not rigorous. As $p(\boldsymbol{x}; \boldsymbol{A})$ is discontinuous, $\check{p}(\boldsymbol{\xi}; \boldsymbol{A})$ may not be integrable and its inverse FT may not exist.

5 Relationships Between PRISM and SVMin

Having pinned down the identifiability of PRISM, we continue by exploring the connection of PRISM and the SVMin approach in convex geometry. Our study will focus on the case of $M = N-1$; such assumption can be justified and has been used in the literature (e.g., [5, 6]), and it will also be discussed in Appendix F.

5.1 SVMin is PRISM in the Noiseless Case

In the preceding section, we showed in (14)–(15) that the PDF $p(\boldsymbol{y}; \boldsymbol{A})$ in the case of $M = N - 1$ takes the form

$$p(\boldsymbol{y}; \boldsymbol{A}) = \frac{1}{\mathrm{vol}(\boldsymbol{A})} \int_{\mathbb{R}^{N-1}} \varphi_\sigma(\boldsymbol{y} - \boldsymbol{x}) \mathbb{1}_{\overline{\mathrm{conv}}(\boldsymbol{A})}(\boldsymbol{x}) \mathrm{d}\boldsymbol{x}, \tag{17}$$

for an affinely independent $\boldsymbol{A}$. Let $\mathcal{A}$ be the set of all $(N-1) \times N$ affinely independent matrices, and restrict the ML problem (8) as

$$\max_{\boldsymbol{A} \in \mathcal{A}} \mathcal{L}_T(\boldsymbol{A}) = \frac{1}{T} \sum_{t=1}^{T} \log p(\boldsymbol{y}_t; \boldsymbol{A}) \tag{18}$$

so that (17) applies. Consider the noiseless case where

$$\log p(\boldsymbol{y}; \boldsymbol{A}) = -\log \mathrm{vol}(\boldsymbol{A}) + \log(\mathbb{1}_{\overline{\mathrm{conv}}(\boldsymbol{A})}(\boldsymbol{y})).$$

Since

$$\log(\mathbb{1}_{\overline{\mathrm{conv}}(\boldsymbol{A})}(\boldsymbol{y})) = \begin{cases} 0, & \boldsymbol{y} \in \overline{\mathrm{conv}}(\boldsymbol{A}) \\ -\infty, & \boldsymbol{y} \notin \overline{\mathrm{conv}}(\boldsymbol{A}) \end{cases}$$

we may rewrite (18) as

$$\begin{aligned} \min_{\boldsymbol{A} \in \mathcal{A}} \ & \log \mathrm{vol}(\boldsymbol{A}) \\ \text{s.t. } & \boldsymbol{y}_t \in \overline{\mathrm{conv}}(\boldsymbol{A}), \quad t = 1, \ldots, T. \end{aligned} \tag{19}$$

We see that problem (19) aims to find a data enclosing simplex $\mathrm{conv}(\boldsymbol{A})$ that yields the minimum volume—which is SVMin [25]. We thereby have the revelation that *PRISM reduces to SVMin in the noiseless case.*

The above identity was mentioned by Bioucas-Dias in his WHISPERS 2009 oral presentation [35]. It was not shown in his papers [34,35], although, for experts, it was alluded to. A somewhat similar result was also mentioned by Dobigeon *et al.* [33, Appendix].

5.2 Connection to Volume-Regularized Matrix Factorization

Let us turn to the noisy case. From (17), we see the following: $p(\boldsymbol{y}; \boldsymbol{A})$ is the convolution of a multivariate Gaussian function $\varphi_\sigma(\boldsymbol{y})$ and the on-off function $\mathbb{1}_{\overline{\mathrm{conv}}(\boldsymbol{A})}(\boldsymbol{y})$ (ignoring the scale $1/\mathrm{vol}(\boldsymbol{A})$); see Fig. 3. Or, $p(\boldsymbol{y}; \boldsymbol{A})$ is a "blurred" version of $\mathbb{1}_{\overline{\mathrm{conv}}(\boldsymbol{A})}(\boldsymbol{y})$ with edges being smoothed. This observation leads us to consider an edge-smooth approximation

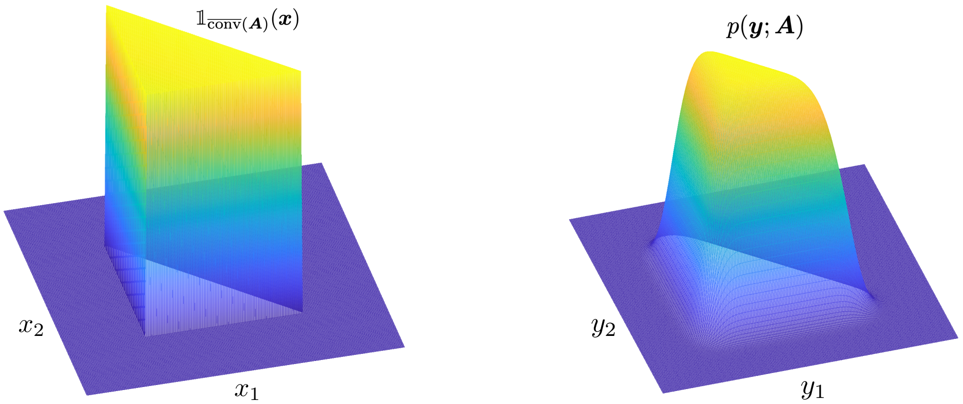
$$p(\boldsymbol{y}; \boldsymbol{A}) \approx \frac{1}{\mathrm{vol}(\boldsymbol{A})} e^{-\frac{1}{\lambda} \cdot \mathrm{dist}(\boldsymbol{y}, \mathrm{conv}(\boldsymbol{A}))^2}, \tag{20}$$

where $\mathrm{dist}(\boldsymbol{x}, \mathcal{X}) := \inf_{\boldsymbol{x}' \in \mathcal{X}} \|\boldsymbol{x} - \boldsymbol{x}'\|$ is the distance of a point $\boldsymbol{x}$ and a set $\mathcal{X}$; $\lambda > 0$ determines the smoothness level which should scale with σ^2. The right-hand side of (20) mimics $p(\boldsymbol{y}; \boldsymbol{A})$ in the sense that it is constant if $\boldsymbol{y} \in \mathrm{conv}(\boldsymbol{A})$, and it gradually goes down as $\boldsymbol{y}$ moves away from $\mathrm{conv}(\boldsymbol{A})$. We argue that (20) is reasonable for high SNRs.

Let us examine the approximation of the ML problem (18) under (20):

$$\begin{aligned} -\max_{\boldsymbol{A} \in \mathcal{A}} \mathcal{L}_T(\boldsymbol{A}) &\approx \min_{\boldsymbol{A} \in \mathcal{A}} \log \mathrm{vol}(\boldsymbol{A}) + \frac{1}{\lambda T} \sum_{t=1}^{T} \mathrm{dist}(\boldsymbol{y}_t, \mathrm{conv}(\boldsymbol{A}))^2 \\ &= \min_{\boldsymbol{A} \in \mathcal{A}, \boldsymbol{S} \in \Delta^T} \log \mathrm{vol}(\boldsymbol{A}) + \frac{1}{\lambda T} \|\boldsymbol{Y} - \boldsymbol{A}\boldsymbol{S}\|^2; \end{aligned} \tag{21}$$

note $\mathrm{dist}(\boldsymbol{y}, \mathrm{conv}(\boldsymbol{A}))^2 = \min_{\boldsymbol{s} \in \Delta} \|\boldsymbol{y} - \boldsymbol{A}\boldsymbol{s}\|^2$. Problem (21) appears as a volume-regularized SSMF, a typical altered formulation of the noiseless SVMin formulation (19) in the noisy case [27,30]. For convenience, we will call (21) *SVMin-SSMF.* To conclude, we can interpret SVMin-SSMF as an approximate PRISM. Intuitively, the approximation should be good for high SNRs.



(a) (b)

Figure 3: Illustration of $\mathbb{1}_{\overline{\mathrm{conv}}(\boldsymbol{A})}(\boldsymbol{x})$ and $p(\boldsymbol{y};\boldsymbol{A})$ in (17).

5.3 Connection to Soft-Constrained SVMin

By the same argument as above, we can also see SISAL [31], a popularly-used algorithm in SVMin, as an approximate PRISM. To put into context, note the following result.

Fact 4 (polyhedral form of a full-dimensional simplex [25]) *Let* $\boldsymbol{A} \in \mathbb{R}^{(N-1)\times N}$ *be affinely independent. We have*

$$\mathrm{conv}(\boldsymbol{A}) = \cap_{i=1}^{N} \mathcal{H}_i(\boldsymbol{A}),$$

where each $\mathcal{H}_i(\boldsymbol{A}) = \{\boldsymbol{y} \mid \boldsymbol{b}_i^\top \boldsymbol{y} \geq c_i\}$ *is a halfspace;*

$$\boldsymbol{B} := [\ \boldsymbol{b}_1, \dots \boldsymbol{b}_{N-1}\ ] = \bar{\boldsymbol{A}}^{-\top}, \tag{22}$$

$$\boldsymbol{c} := (c_1, \dots, c_{N-1}) = \bar{\boldsymbol{A}}^{-1}\boldsymbol{a}_N, \tag{23}$$

$$c_N := -\boldsymbol{1}^\top \boldsymbol{c} - 1,\ \ \boldsymbol{b}_N := -\boldsymbol{B}\boldsymbol{1}. \tag{24}$$

Our intuition is to build a variant of the approximation (20) by penalizing points that lie outside $\mathcal{H}_i(\boldsymbol{A})$. To this end, consider a change of variables $\boldsymbol{A} \in \mathcal{A} \rightarrow (\boldsymbol{B}, \boldsymbol{c}) \in \mathcal{B} \times \mathbb{R}^{N-1}$ according to (22)–(23), where $\mathcal{B}$ denotes the set of all invertible $(N-1)\times(N-1)$ matrices. We adopt the following approximation

$$p(\boldsymbol{y};\boldsymbol{A}) \approx \frac{1}{\mathrm{vol}(\boldsymbol{A})} e^{-\frac{1}{\lambda}\sum_{i=1}^{N}(c_i - \boldsymbol{b}_i^\top \boldsymbol{y})_+} \tag{25}$$

for some $\lambda > 0$, where $(x)_+ := \max\{0, x\}$; $(c_i - \boldsymbol{b}_i^\top \boldsymbol{y})_+$ serves as a penalty function for the violation of $\boldsymbol{y} \in \mathcal{H}_i(\boldsymbol{A})$. The resulting approximation of the ML problem (18) is

$$\begin{aligned} \min_{\boldsymbol{B},\boldsymbol{c}} \ & -\log|\det(\boldsymbol{B})| + \frac{1}{\lambda T}\sum_{t=1}^{T}\sum_{i=1}^{N}(c_i - \boldsymbol{b}_i^\top \boldsymbol{y}_t)_+ \\ \text{s.t.}\ & \boldsymbol{B} \in \mathcal{B},\ \text{(24) holds.} \end{aligned} \tag{26}$$

Problem (26) is similar to the formulation of SISAL [31], whose rationale is to replace the hard constraints $\boldsymbol{y} \in \mathrm{conv}(\boldsymbol{A})$ in the original SVMin formulation (19) with "soft constraints" to make the solution robust against noise.

5.4 Connection to Chance-Constrained SVMin

We can also draw a connection to robust SVMin via chance constraints [32]. Observe

$$\int_{\mathbb{R}^{N-1}} \varphi_\sigma(\boldsymbol{y}-\boldsymbol{x})\mathbb{1}_{\mathrm{conv}(\boldsymbol{A})}(\boldsymbol{x})\mathrm{d}\boldsymbol{x} = \mathrm{Prob}(\boldsymbol{y}-\boldsymbol{v} \in \mathrm{conv}(\boldsymbol{A})),$$

where $\boldsymbol{v} \sim \varphi_\sigma$. Using the polyhedral representation in Fact 4, we get

$$\begin{aligned}\mathrm{Prob}(\boldsymbol{y}-\boldsymbol{v} \in \mathrm{conv}(\boldsymbol{A})) &\leq \mathrm{Prob}(\boldsymbol{y}-\boldsymbol{v} \in \mathcal{H}_i(\boldsymbol{A})) \\ &= \mathrm{Prob}(-\boldsymbol{b}_i^\top \boldsymbol{v} \geq c_i - \boldsymbol{b}_i^\top \boldsymbol{y}) \\ &= \Phi\left(\frac{\boldsymbol{b}_i^\top \boldsymbol{y} - c_i}{\sigma\|\boldsymbol{b}_i\|}\right)\end{aligned}$$

for any i, where $\Phi(x) = \frac{1}{\sqrt{2\pi}}\int_{-\infty}^{x} e^{-z^2/2}\mathrm{d}z$. The above equations lead to an upper-bound approximation

$$p(\boldsymbol{y};\boldsymbol{A}) \leq \frac{1}{\mathrm{vol}(\boldsymbol{A})} \min_{i=1,\ldots,N} \Phi\left(\frac{\boldsymbol{b}_i^\top \boldsymbol{y} - c_i}{\sigma\|\boldsymbol{b}_i\|}\right). \tag{27}$$

By applying (27) and the change of variables in (22)–(23), we obtain an approximation of the ML problem (18) as follows

$$\begin{aligned}&\min_{\boldsymbol{B},\boldsymbol{c}} \ -\log|\det(\boldsymbol{B})| + \frac{1}{T}\sum_{t=1}^{T} \max_{i=1,\ldots,N} -\log\Phi\left(\frac{\boldsymbol{b}_i^\top \boldsymbol{y}_t - c_i}{\sigma\|\boldsymbol{b}_i\|}\right) \\ &\text{s.t. } \boldsymbol{B} \in \mathcal{B}, \ (24) \text{ holds.}\end{aligned} \tag{28}$$

Problem (28) is reminiscent of an existing SVMin formulation, namely, the chance-constrained SVMin

$$\begin{aligned}&\min_{\boldsymbol{B}\in\mathcal{B},\boldsymbol{c}} \ -\log|\det(\boldsymbol{B})| \\ &\text{s.t. } \Phi((\boldsymbol{b}_i^\top \boldsymbol{y}_t - c_i)/(\sigma\|\boldsymbol{b}_i\|)) \geq \eta, \text{ for all } i,t\end{aligned} \tag{29}$$

for a pre-specified $\eta \in [0,1]$ [32]. Problem (29) was proposed as an alteration of the noiseless SVMin formulation (19), wherein the data enclosing constraints of (19) are modified as chance constraints to improve robustness to noise. We see that by changing the penalty terms of problem (28) as constraints, we get the chance-constrained SVMin problem (29).

5.5 Further Discussion

In Appendix G, we further discuss aspects arising from the PRISM-SVMin relationships revealed above.

6 Importance Sampling Approximation (ISA)

Having shown the relationships of PRISM and SVMin in the previous section, we now turn our attention to designated schemes for realizing PRISM. This section will consider ISA, while the next section will be devoted to variational inference approximation (VIA). Note that, unlike the previous section, we do not assume $M = N - 1$.

6.1 A Variational Reformulation of ML

The ISA scheme to be presented is an instance of Monte Carlo expectation maximization (MCEM). To describe, we consider a slightly non-standard presentation that will cover expectation maximization (EM), MCEM, and later, VIA. Let q be any PDF that is measurable on $\{\boldsymbol{s} \in \mathbb{R}^N \mid \mathbf{1}^\top \boldsymbol{s} = 1\}$ and has support $\bar{\Delta}$. Let $p(\boldsymbol{y}, \boldsymbol{s}; \boldsymbol{A}) = p(\boldsymbol{y}|\boldsymbol{s}; \boldsymbol{A})p(\boldsymbol{s})$. Consider the Jensen inequality

$$\begin{aligned} \log p(\boldsymbol{y}; \boldsymbol{A}) &= \log \left(\int p(\boldsymbol{y}, \boldsymbol{s}; \boldsymbol{A}) \frac{q(\boldsymbol{s})}{q(\boldsymbol{s})} \mathrm{d}\mu(\boldsymbol{s}) \right) \\ &= \log \left(\mathbb{E}_{\boldsymbol{s}\sim q}[p(\boldsymbol{y}, \boldsymbol{s}; \boldsymbol{A})/q(\boldsymbol{s})] \right) \\ &\geq \mathbb{E}_{\boldsymbol{s}\sim q}[\log(p(\boldsymbol{y}, \boldsymbol{s}; \boldsymbol{A})/q(\boldsymbol{s}))] \\ &:= \hat{\ell}(\boldsymbol{A}, q; \boldsymbol{y}), \end{aligned} \tag{30}$$

where equality in (30) holds if and only if

$$q(\boldsymbol{s}) \propto p(\boldsymbol{y}, \boldsymbol{s}; \boldsymbol{A}) \propto \varphi_\sigma(\boldsymbol{y} - \boldsymbol{A}\boldsymbol{s}) \mathbb{1}_{\bar{\Delta}}(\boldsymbol{s}), \tag{31}$$

i.e., q is a unit-simplex truncated Gaussian distribution. Note that (31) is equivalent to

$$q(\boldsymbol{s}) = \frac{p(\boldsymbol{y}, \boldsymbol{s}; \boldsymbol{A})}{\int p(\boldsymbol{y}, \boldsymbol{s}; \boldsymbol{A}) \mathrm{d}\mu(\boldsymbol{s})} = p(\boldsymbol{s}|\boldsymbol{y}; \boldsymbol{A}). \tag{32}$$

Using (30), we can reformulate the ML problem (8) as

$$\max_{\boldsymbol{A}\in\mathbb{R}^{M\times N}, q_t \in \mathcal{D}\, \forall t} \hat{\mathcal{L}}_T(\boldsymbol{A}, \{q_t\}) := \tfrac{1}{T} \textstyle\sum_{t=1}^T \hat{\ell}(\boldsymbol{A}, q_t; \boldsymbol{y}_t), \tag{33}$$

where $\mathcal{D}$ is the family of all distributions with support $\bar{\Delta}$.

We should justify why we are interested in (33), a seemingly more complex ML formulation. The difficulty with the original ML problem (8) is that $\log p(\boldsymbol{y}; \boldsymbol{A})$ is an intractable integral. We want to see if (33), or its approximations, will circumvent the difficulty. To put into context, consider an alternating maximization (AM) method for problem (33):

$$\boldsymbol{A}^{k+1} \in \arg \max_{\boldsymbol{A}\in\mathbb{R}^{M\times N}} \hat{\mathcal{L}}_T(\boldsymbol{A}, \{q_t^k\}), \tag{34a}$$

$$q_t^{k+1} \in \arg\max_{q_t \in \mathcal{D}} \hat{\ell}(\boldsymbol{A}^{k+1}, q_t; \boldsymbol{y}_t), \quad t = 1, \ldots, T, \tag{34b}$$

for $k = 0, 1, \cdots$. The ISA and VIA schemes to be developed seek two different ways to approximate (34).

6.2 The ISA-PRISM Scheme

Before we describe our ISA scheme, we should first note that the AM (34) for realizing the ML is identical to EM. By the Jensen inequality result in (30)–(32), the solution to (34b) is $q_t^{k+1} = p(\cdot|\boldsymbol{y}_t; \boldsymbol{A}^{k+1})$. By putting this solution to (34a), we can simplify the AM (34) to

$$\boldsymbol{A}^{k+1} \in \arg \max_{\boldsymbol{A}\in\mathbb{R}^{M\times N}} \textstyle\sum_{t=1}^T \mathbb{E}_{\boldsymbol{s}_t \sim p(\cdot|\boldsymbol{y}_t; \boldsymbol{A}^k)}[\log p(\boldsymbol{y}_t|\boldsymbol{s}_t; \boldsymbol{A})]. \tag{35}$$

Eq. (35) takes the same form as EM, which was derived by a minorization-maximization methodology. In considering (35), we wish that either (35) would be easy to solve, or (35) would admit a tractable objective function. But none of the above is true in our problem. A natural idea is then to apply ISA, using a large amount of randomly drawn samples to approximate the objective function. Such idea is identical to MCEM [36].

Let us go into the details. Let $\boldsymbol{\xi}_t^1, \ldots, \boldsymbol{\xi}_t^{R_t}$ be an R_t number of randomly drawn samples from $p(\cdot|\boldsymbol{y}_t; \boldsymbol{A}^k)$.[3] This requires us to generate samples from a unit-simplex truncated Gaussian distribution, and it can be done by rejection sampling or MCMC methods; see, e.g., [45, 46]. We apply the ISA

$$\mathbb{E}_{\boldsymbol{s}_t \sim p(\cdot|\boldsymbol{y}_t; \boldsymbol{A}^k)}[\log p(\boldsymbol{y}_t|\boldsymbol{s}_t; \boldsymbol{A})] \approx \tfrac{1}{R_t} \textstyle\sum_{r=1}^{R_t} \log p(\boldsymbol{y}_t|\boldsymbol{\xi}_t^r; \boldsymbol{A}).$$

The subsequent approximation of (35) can be shown to be

$$\boldsymbol{A}^{k+1} \in \arg \max_{\boldsymbol{A} \in \mathbb{R}^{M \times N}} - \textstyle\sum_{t=1}^T \tfrac{1}{R_t} \sum_{r=1}^{R_t} \|\boldsymbol{y}_t - \boldsymbol{A}\boldsymbol{\xi}_t^r\|^2, \tag{36}$$

which is a least squares problem with solution

$$\begin{aligned} \boldsymbol{A}^{k+1} &= \left(\textstyle\sum_{t=1}^T \boldsymbol{y}_t \boldsymbol{m}_t^\top\right) \left(\textstyle\sum_{t=1}^T \boldsymbol{R}_t\right)^\dagger, \\ \boldsymbol{m}_t &= \tfrac{1}{R_t} \textstyle\sum_{r=1}^{R_t} \boldsymbol{\xi}_t^r, \quad \boldsymbol{R}_t = \tfrac{1}{R_t} \textstyle\sum_{r=1}^{R_t} \boldsymbol{\xi}_t^r (\boldsymbol{\xi}_t^r)^\top. \end{aligned} \tag{37}$$

To summarize, our ISA-PRISM scheme is given by (37), where, at each iteration k, we generate $\{\boldsymbol{\xi}_t^r\}$ from $p(\cdot|\boldsymbol{y}_t; \boldsymbol{A}^k)$ by a sampling method.

6.3 Discussion

Let us discuss the advantages and drawbacks of the above ISA-PRISM scheme. The implementation of ISA-PRISM is very simple. It can also deal with more complex models such as models under the presence of outlying data points, Dirichlet mixture models, nonlinear and variability models, non-negative $\boldsymbol{A}$, etc.; such extensions will not be pursued here, and the reader is referred to [39–41]. However, using ISA also means that we need a massive amount of samples to approximate well. This is particularly a concern for large problem dimension N. This issue is coupled with another issue, namely, the low efficiencies of known sampling methods for large problem dimension N; e.g., rejection sampling tends to reject many samples, or take many iterations to generate one sample, when N is large. Empirically we found that ISA-PRISM works very well for N less than 10, but performs poorly for larger N. We should note that the aforementioned merits and limitations are common in Monte Carlo-based inference methods.

7 Variational Inference Approximation (VIA)

The ISA-PRISM scheme in the last section implements the ML estimator by applying Monte Carlo approximation to the intractable integral in the alternating maximization (34). VIA attacks (34) by a different route, namely, restricting the distribution family $\mathcal{D}$ so that the objective function is tractable.

[3] Actually $\boldsymbol{\xi}_t^1, \ldots, \boldsymbol{\xi}_t^{R_t}$ and R_t depend on the iteration k, but we shall suppress the latter for brevity.

7.1 Dirichlet VIA

Consider restricting $\mathcal{D}$, the family of all $\bar{\Delta}$-supported distributions, in the ML formulation (33) by the Dirichlet family

$$\mathcal{D} = \{q = D(\cdot; \boldsymbol{\alpha}) \mid \boldsymbol{\alpha} \in \mathbb{R}^N_{++}\}. \tag{38}$$

Such restriction will lead to a lower-bound approximation of the ML. We have two reasons for this. First, among all $\bar{\Delta}$-supported distributions, the Dirichlet distribution is the most well-understood. Second, it can be shown that, under $q = D(\cdot; \boldsymbol{\alpha})$, the function $\hat{\ell}(\boldsymbol{A}, q; \boldsymbol{y})$ in (30) can be written as

$$\begin{aligned} -\hat{\ell}(\boldsymbol{A}, q; \boldsymbol{y}) &\propto \frac{1}{2\sigma^2}\mathbb{E}[\|\boldsymbol{y} - \boldsymbol{A}\boldsymbol{s}\|^2] - H(\boldsymbol{s}) \\ &= \frac{1}{2\sigma^2}(\|\boldsymbol{y} - \boldsymbol{A}\mathbb{E}[\boldsymbol{s}]\|^2 + \mathrm{tvar}(\boldsymbol{A}\boldsymbol{s})) - H(\boldsymbol{s}) := f(\boldsymbol{A}, \boldsymbol{\alpha}; \boldsymbol{y}), \end{aligned} \tag{39}$$

where we denote $\mathbb{E}_{\boldsymbol{s}\sim q}[\cdot] = \mathbb{E}[\cdot]$ for brevity; $H(\boldsymbol{s}) = \mathbb{E}[-\log q(\boldsymbol{s})]$ is the entropy; $\mathrm{tvar}(\boldsymbol{x}) = \sum_{i=1}^n \mathrm{var}(x_i) = \mathrm{tr}(\mathrm{Cov}(\boldsymbol{x}))$. We know from Fact 1 that $\mathbb{E}[\boldsymbol{s}]$, $\mathrm{Cov}(\boldsymbol{s})$ and $H(\boldsymbol{s})$ have explicit expressions; we will examine the details later. Consequently, the ML problem (33) under the restrictive approximation (38), or VIA-ML for short, has a tractable objective function.

7.2 VIA-ML is Regularized Matrix Factorization

It is worthwhile to pause a moment to draw connections. From (33) and (39), we can write the VIA-ML problem as

$$\min_{\boldsymbol{\Theta}\in\mathcal{C}} \frac{1}{2\sigma^2}(\|\boldsymbol{Y} - \boldsymbol{A}\mathbb{E}[\boldsymbol{S}]\|^2 + \mathrm{tvar}(\boldsymbol{A}\boldsymbol{S})) - H(\boldsymbol{S}), \tag{40}$$

where $\boldsymbol{s}_t \sim D(\cdot; \boldsymbol{\alpha}_t)$; $\boldsymbol{\Theta} = \{\boldsymbol{A}, \boldsymbol{\alpha}_1, \ldots, \boldsymbol{\alpha}_T\}$; $\mathcal{C} = \mathbb{R}^{M\times N} \times \mathbb{R}^N_{++} \times \cdots \times \mathbb{R}^N_{++}$; $\mathrm{tvar}(\boldsymbol{X}) = \sum_{j=1}^n \mathrm{tvar}(\boldsymbol{x}_j)$; $H(\boldsymbol{S}) = \sum_{t=1}^T H(\boldsymbol{s}_t)$. We see that *the VIA-ML problem* (40) *resembles a matrix factorization problem*—the first term of (40) is a data fitting term in matrix factorizaton; the second term is a penalty term on variances, encouraging smaller variances; the third term is a negative-entropy penalty term, discouraging smaller variances (entropy tends to be larger for more diversely distributed distributions). Note that, from a matrix factorization viewpoint, problem (40) has no regularization parameter to tune. We also show the following result:

Proposition 1 *If we remove the entropy term H from the VIA-ML problem* (40)*, the resulting problem is equivalent to the plain SSMF*

$$\min_{\boldsymbol{A}\in\mathbb{R}^{M\times N}, \boldsymbol{\Xi}\in\bar{\Delta}^T} \|\boldsymbol{Y} - \boldsymbol{A}\boldsymbol{\Xi}\|^2. \tag{41}$$

Proof of Proposition 1: Define $\eta_t = \boldsymbol{1}^\top\boldsymbol{\alpha}_t$, $\boldsymbol{\xi}_t = \boldsymbol{\alpha}_t/\eta_t \in \bar{\Delta}$. From Fact 1 one readily gets $\mathbb{E}[\boldsymbol{s}_t] = \boldsymbol{\xi}_t$,

$$\mathrm{Cov}(\boldsymbol{s}_t) = \frac{1}{1+\eta_t}\boldsymbol{C}(\boldsymbol{\xi}_t), \quad \boldsymbol{C}(\boldsymbol{\xi}_t) = \mathrm{Diag}(\boldsymbol{\xi}_t) - \boldsymbol{\xi}_t\boldsymbol{\xi}_t^\top.$$

Problem (40) without H can be expressed as

$$\min_{\boldsymbol{\Theta}\in\mathcal{C}} f(\boldsymbol{\Theta}) := \|\boldsymbol{Y} - \boldsymbol{A}\boldsymbol{\Xi}\|^2 + \textstyle\sum_{t=1}^T \frac{1}{1+\eta_t}\mathrm{tr}(\boldsymbol{A}\boldsymbol{C}(\boldsymbol{\xi}_t)\boldsymbol{A}^\top). \tag{42}$$

Let $\boldsymbol{\Theta}^*$ be a solution to (42). Let $\eta_t^* = \mathbf{1}^\top \boldsymbol{\alpha}_t^*$, $\boldsymbol{\xi}_t^* = \boldsymbol{\alpha}_t^*/\eta_t^*$, and choose $\boldsymbol{\alpha}_t = \eta_t \boldsymbol{\xi}_t^*$ for any $\eta_t > \eta_t^*$. Then we see from (42) that $\boldsymbol{\Theta} = \{\boldsymbol{A}^*, \boldsymbol{\alpha}_1, \ldots, \boldsymbol{\alpha}_T\}$ has $f(\boldsymbol{\Theta}^*) \geq f(\boldsymbol{\Theta})$. This implies that there always exists a solution for which $\eta_t \to \infty$ for all t. As the second term of f in (42) diminishes as $\eta_t \to \infty$, we can reduce (42) to (41). This completes the proof. ■

Remark 2 Proposition 1 reveals a limitation. Suppose σ^2 is very small. The entropy term H in the VIA-ML problem (40) would have negligible effects, and, by Proposition 1, the VIA-ML problem should be close to the plain SSMF. Moreover the plain SSMF is an unidentifiable formulation, as indicated in Fact 3. This implies that VIA-ML may not work well for high SNRs. Our empirical results to be presented seem to be in agreement with the above argument. But our empirical results will also indicate that VIA-ML works well for low SNRs.

7.3 The VIA-PRISM Scheme

We now turn to algorithmic realization. Our VIA-PRISM scheme is the realization of AM (34) under the aforementioned VIA. By applying Fact 1, we express the right-hand side of (39) as

$$
\begin{aligned}
f(\boldsymbol{A}, \boldsymbol{\alpha}; \boldsymbol{y}) &= g(\boldsymbol{A}, \boldsymbol{\alpha}, \mathbf{1}^\top \boldsymbol{\alpha}; \boldsymbol{y}) + \textstyle\sum_{i=1}^N h(\alpha_i) + \iota(\mathbf{1}^\top \boldsymbol{\alpha}) + C, && \text{(43a)} \\
g(\boldsymbol{A}, \boldsymbol{\alpha}, \eta; \boldsymbol{y}) &= \tfrac{1}{\sigma^2} \left(-\tfrac{1}{\eta} \boldsymbol{y}^\top \boldsymbol{A} \boldsymbol{\alpha} + \tfrac{1}{2(1+\eta)\eta} \mathrm{tr}(\boldsymbol{A} \boldsymbol{R}(\boldsymbol{\alpha}) \boldsymbol{A}^\top) \right), && \text{(43b)} \\
\boldsymbol{R}(\boldsymbol{\alpha}) &= \mathrm{Diag}(\boldsymbol{\alpha}) + \boldsymbol{\alpha} \boldsymbol{\alpha}^\top, && \text{(43c)} \\
h(\alpha) &= -\log \Gamma(\alpha) + (\alpha - 1) \psi(\alpha), && \text{(43d)} \\
\iota(\eta) &= \log \Gamma(\eta) - (\eta - N) \psi(\eta), && \text{(43e)}
\end{aligned}
$$

where C is a constant; recall that $\Gamma(x) = \int_0^\infty t^{x-1} e^{-t} \mathrm{d}t$ is the Gamma function, and $\psi(x) = \frac{\mathrm{d} \log \Gamma(x)}{\mathrm{d}x}$ is the digamma function. The AM (34) under the VIA can be written as

$$
\begin{aligned}
\boldsymbol{A}^{k+1} &\in \arg \min_{\boldsymbol{A} \in \mathbb{R}^{M \times N}} \textstyle\sum_{t=1}^T g(\boldsymbol{A}, \boldsymbol{\alpha}_t^k, \mathbf{1}^\top \boldsymbol{\alpha}_t^k; \boldsymbol{y}_t), && \text{(44a)} \\
\boldsymbol{\alpha}_t^{k+1} &\in \arg \min_{\boldsymbol{\alpha} \in \mathbb{R}_{++}^N} f(\boldsymbol{A}^{k+1}, \boldsymbol{\alpha}; \boldsymbol{y}_t), \quad t = 1, \ldots, T. && \text{(44b)}
\end{aligned}
$$

Problem (44a) is a least squares problem with solution

$$
\boldsymbol{A}^{k+1} = \left[\sum_{t=1}^T \tfrac{1}{\eta_t^k} \boldsymbol{y}_t (\boldsymbol{\alpha}_t^k)^\top \right] \left[\sum_{t=1}^T \tfrac{1}{(1+\eta_t^k)\eta_t^k} \boldsymbol{R}(\boldsymbol{\alpha}_t^k) \right]^{-1}, \tag{45}
$$

where $\eta_t^k = \mathbf{1}^\top \boldsymbol{\alpha}_t^k$. The problems in (44b) are not easy and will be treated next.

7.4 Are the Variational Problems in (44b) Solvable?

For notational convenience, let us rewrite (44b) as

$$
\min_{\boldsymbol{\alpha} \in \mathbb{R}_{++}^N} f(\boldsymbol{A}, \boldsymbol{\alpha}; \boldsymbol{y}), \tag{46}
$$

Problem (46) is non-convex; the term $\eta = \mathbf{1}^\top \boldsymbol{\alpha}$ in (43b) is particularly troublesome. As a fundamental study, we beg this question: can problem (46) be solvable? To answer that, we reformulate problem (46) as

$$
\min_{\eta \in \mathbb{R}_{++}} r(\eta) + \iota(\eta), \tag{47}
$$

where

$$\begin{aligned} r(\eta) = \min_{\boldsymbol{\alpha} \in \mathbb{R}_{++}^N} \ & g(\boldsymbol{A}, \boldsymbol{\alpha}, \eta; \boldsymbol{y}) + \textstyle\sum_{i=1}^N h(\alpha_i) \\ \text{s.t.}\ & \mathbf{1}^\top \boldsymbol{\alpha} = \eta. \end{aligned} \tag{48}$$

Suppose that $r(\eta)$ is efficiently computable for any given $\eta > 0$. Then we may argue that problem (47) is not that difficult—for we can use grid search to find the solution to problem (47) (up to an accuracy). In practice it is more pragmatic to employ line search, rather than grid search. We would expect, at least intuitively, that the chance for line search to find the optimal solution to a one-dimensional problem should be high. Hence the question boils down to whether problem (48) is efficiently solvable. Observe that g in (43b) is convex in $\boldsymbol{\alpha}$. If h is convex on $\mathbb{R}_{++}$ then problem (48) is convex. We show that this is true.

Proposition 2 *The function h is strictly convex on $\mathbb{R}_{++}$. As a direct corollary, problem* (48) *is strictly convex.*

The proof of Proposition 2 is relegated to Appendix I. We also custom-develop an efficient solver for problem (48) via the augmented direction method of multipliers (ADMM) [47]; the details are also relegated to Appendix J.

Let us summarize how we numerically solve problem (46). There are two levels. The first level applies line search to problem (47); we employ Golden search. At each line search iteration, the computation of $r(\eta)$ for a specific η is required. This is done at the second level, where we solve problem (48) by the ADMM solver in Appendix J. The pseudo code is provided in Algorithm 1.

Algorithm 1 Solver for the varational problem in (46)

```
given: a search interval $[a, b]$
set $[\eta_1, \eta_2] = [a, b]$, $\mu = \frac{1+\sqrt{5}}{2}$ (the golden ratio)
repeat
    let $\eta_3 = \eta_2 - (\eta_2 - \eta_1)/\mu$, $\eta_4 = \eta_1 + (\eta_2 - \eta_1)/\mu$
    compute $r(\eta_3)$ and $r(\eta_4)$ by solving problem (48) using the ADMM solver in Appendix J
    if $r(\eta_3) + \iota(\eta_3) < r(\eta_4) + \iota(\eta_4)$ then
        update $\eta_2 = \eta_4$
    else
        update $\eta_1 = \eta_3$
    end if
until a stopping rule is satisfied
set $\eta = (\eta_1 + \eta_2)/2$
find the solution $\boldsymbol{\alpha}^\star$ to problem (48) using the ADMM solver in Appendix J
output $\boldsymbol{\alpha}^\star$
```

8 Numerical Experiments

We performed numerical experiments to examine the potential of PRISM.

algorithm	formulation & reference	initialization	parameter settings
SVMAX	pure-pixel search [19]	/	/
SISAL	soft-constrained SVMin, cf. (19) [31]	VCA [18]	volume regularization parameter $\lambda = 0.1$
SISAL-t			MSE vs SNR: $\lambda = 0.02 \times (\mathrm{SNR} - 7)/3$ $(N = 5)$ MSE vs SNR: $\lambda = 0.02 \times \mathrm{SNR}/3$ $(N = 20)$ MSE vs T: $\lambda = \frac{0.01}{[T/1000]+1}$ (SNR=10dB) MSE vs T: $\lambda = \frac{0.2}{[T/1000]+1}$ (SNR=20dB)
RVolMin	SVMin-SSMF in (49) [27]	SISAL-t	$\lambda = 0.5, p = 0.5$
MVES	noiseless SVMin in (19) [25]	solve a feasibility problem	/
RMVES	chance-constrained SVMin in (29) [32]	VCA	$\eta = 0.16$
ISA-PRISM	ML, importance sampling approx., Section 6	SVMAX	rejection sampling from 500 samples
VIA-PRISM	ML, variational inference approx., Section 7	SVMAX	/

Table 1: Settings of the various schemes.

8.1 Algorithm Settings

The algorithm settings of the ISA-PRISM and VIA-PRISM schemes in Sections 6 and 7 are described as follows. Both ISA-PRISM and VIA-PRISM deal with non-convex optimization, and a reasonable initialization for them would be desirable. We initialize the two schemes by SVMAX [19], a computationally light CG algorithm by pure-pixel search.[4] ISA-PRISM is implemented by rejection sampling. Specifically, for each data point $\boldsymbol{y}_t$, we give $R = 500$ random samples and pick up the accepted samples by rejection sampling. Note that the number of accepted samples R_t for each data point $\boldsymbol{y}_t$ can vary from one point to another. For VIA-PRISM, we set the parameter ρ of the ADMM solver (see Appendix J) as 0.01. We stop the ADMM solver when the dual error is less than 0.005. Furthermore, we stop ISA-PRISM and VIA-PRISM when their iteration numbers exceed 100.

We benchmark PRISM against the following state-of-the-art schemes: i) the pure-pixel search algorithm SVMAX; ii) MVES [25], which realizes the noiseless SVMin in (19); iii) the famous SISAL [31], which adopts a soft-constrained SVMin formulation similar to (26); iv) RVolMin [27], which considers the SVMin-SSMF formulation

$$\min_{\boldsymbol{A}, \boldsymbol{S} \in \Delta^T} \lambda \log\det(\boldsymbol{A}^\top \boldsymbol{A}) + \textstyle\sum_{t=1}^T \|\boldsymbol{y}_t - \boldsymbol{A}\boldsymbol{s}_t\|^p \tag{49}$$

for some given $p > 0$, $\lambda > 0$; v) RMVES [32], which is based on the chance-constrained SVMin formulation in (29). Some key settings of the above schemes are shown in Table 1. For SISAL we consider two implementations. One, simply called "SISAL", has the regularization parameter λ fixed for all experiments.[5] Another, called "SISAL-t", has λ manually tuned for better performance; our tuning is heuristic and heavily empirical based. Also, we stop SISAL and RVolMin when the iteration numbers exceed 250 and $1,000$, resp.; we stop MVES and RMVES when the relative objective value changes are less than 10^{-8} and 10^{-6}, resp.

8.2 Synthetic Data Experiments

We performed a collection of synthetic data experiments. The data model (7) and the accompanied assumptions are used to generate the data points. In each simulation trial, we randomly generate

[4] SVMAX belongs to a representative type of pure-pixel search methods. It resembles VCA [18], a very widely-used pure-pixel search algorithm. It is nearly identical to SPA [22], which is equipped with noise robustness analyses.

[5] Here, λ refers to the volume regularization parameter presented in the SISAL paper [31], not the one in the closely-related formulation (26).

$\boldsymbol{A}_0$ by the element-wise independent $[0,1]$-uniform distribution. We measure the estimation performance by mean square error (MSE) $\mathsf{MSE}(\boldsymbol{A}_0, \hat{\boldsymbol{A}}) = \min_{\boldsymbol{\pi} \in \Pi_N} \frac{1}{MN} \sum_{n=1}^{N} \|\boldsymbol{a}_{0,\pi_n} - \hat{\boldsymbol{a}}_n\|^2$, where $\hat{\boldsymbol{A}}$ is the estimated vertex matrix; Π_N is the set of all permutations of $\{1, 2, \ldots, N\}$. We fix $M = 50$, and we use 500 independent trials to obtain the results below.

8.2.1 Varying the Data Length T

We are interested in examining how the various schemes perform as the number of available data points, T, increases; intuitively, one would expect that the MSE improves with T in a consistent manner. In Figs. 4(a) and 4(b) we show the MSEs for different T. Note that the lines are average MSEs, while the shadows indicate the standard deviations of the MSEs. Let us first set our eyes on the case of $N = 5$ in Fig. 4(a). We observe the following:

1. First, the MSEs of ISA-PRISM and VIA-PRISM improve as T increases. This is in agreement with the identifiability theorem in Theorem 1, which says that PRISM can perfectly identify the vertices when T approaches infinity. That being said, we also notice that the MSE improvement is slow as T becomes very large, say, $T \geq 4,000$.

2. Second, the benchmark schemes do not seem to show consistent MSE improvement with T. As such, ISA-PRISM and VIA-PRISM are able to perform better than the benchmark schemes for large T. We also observe that SISAL, which has no parameter tuning, behaves peculiarly. SISAL-t, which has parameter tuning, gives better performance. The heuristic nature of the tuning should however be noted.

3. Third, there is a performance gap between ISA-PRISM and VIA-PRISM. The reason could be with the approximation errors of VIA. The gap is seen to be larger for the higher SNR case (SNR= 20dB). In this regard we should recall that, as indicated in Proposition 1 and discussed in Remark 2, VIA may not work well for very high SNRs.

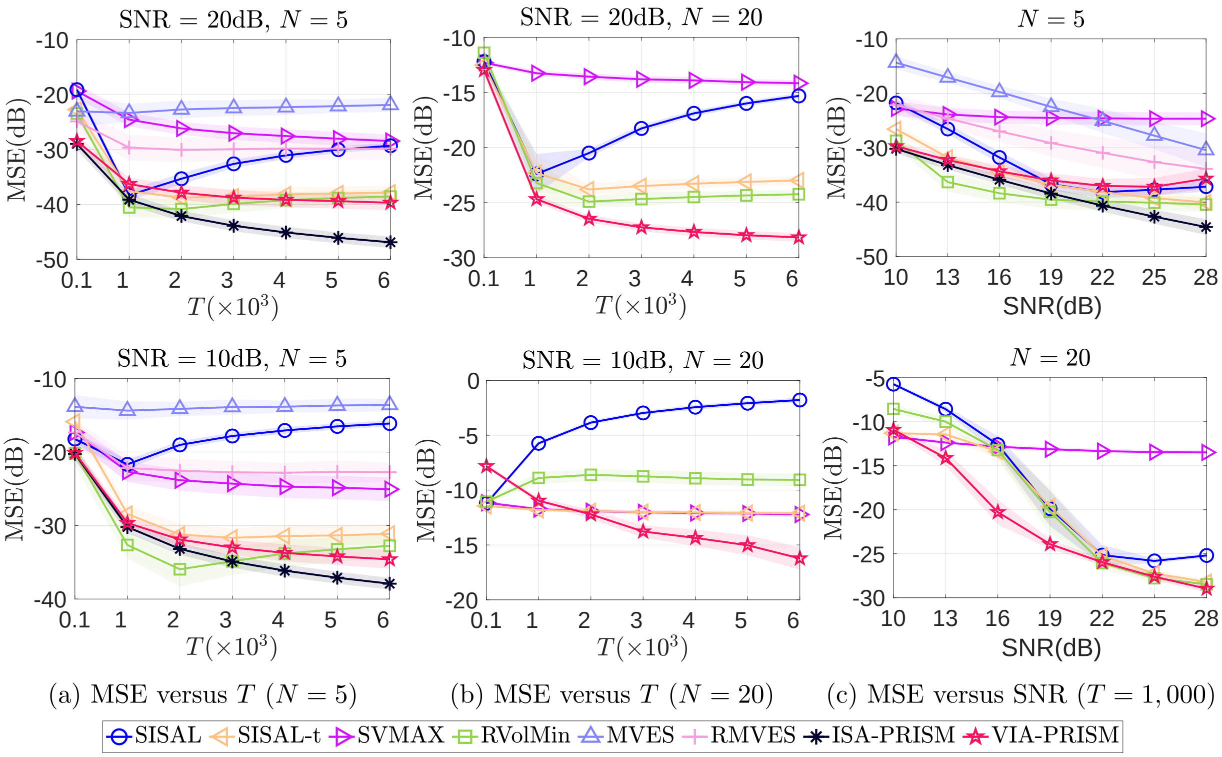
Next, we turn to the case of $N = 20$ in Fig. 4(b). We did not try MVES and RMVES because they run too slowly for large N. We also did not consider ISA-PRISM because it fails to work; rejection sampling generates almost no sample. We observe similar performance behaviors as the case of $N = 5$. Moreover, VIA-PRISM is seen to perform better than the benchmark schemes for large T.

8.2.2 Varying the Noise Level

Fig. 4(c) shows the MSEs for different SNRs. Note that we fix $T = 1,000$. We see that, for $N = 5$, ISA-PRISM generally outperforms the other schemes. For $N = 20$, VIA-PRISM generally works better than the benchmark schemes; this is particularly so for lower SNRs. In addition, it is worth mentioning that SISAL-t gives reasonably good performance. But note that SISAL-t requires heuristic parameter tuning, while VIA-PRISM does not.

8.2.3 Runtime Comparison

We compare the runtimes of some of the schemes in Table 2. We consider $T = 1,000$, SNR= 10dB, and the runtimes were recorded on a desktop computer with Intel Core i7 3.20GHz CPU and 64GB memory, and under MATLAB2019. The computational costs of ISA-PRISM and VIA-PRISM are



(a) MSE versus T ($N = 5$) (b) MSE versus T ($N = 20$) (c) MSE versus SNR ($T = 1,000$)

Figure 4: MSE performance. $M = 50$.

seen to be on the high side. We should say that this study focuses on fundamental aspects and sets aside efficient implementations. The latter will be future work.

Table 2: Average runtimes (sec.) of the various schemes.

	SVMAX	SISAL	RVolMin	ISA-PRISM	VIA-PRISM
$N = 5$	0.003	0.06	0.65	30.76	15.93
$N = 20$	0.005	0.27	1.37	-	189.89

8.3 Real Data Experiment

We apply PRISM to real data. The application of interest is hyperspectral unmixing (HU) in remote sensing. The problem is well-known in the remote sensing literature (see, e.g., [5, 6]), and it is concisely described as follows. We are given a hyperspectral image of a captured scene, which has a few hundreds of spectral bands and has high spectral resolution. That image is represented by $\boldsymbol{y}_1, \ldots, \boldsymbol{y}_T$, where each $\boldsymbol{y}_t \in \mathbb{R}^M$ is a pixel collecting measurements over a number of M spectral bands, and T is the number of pixels. We posit that the $\boldsymbol{y}_t$'s follow the model in (7), where the columns of $\boldsymbol{A}_0$ are the spectral signatures of different materials that underlie the scene, and $\boldsymbol{s}_t$ describes the materials' distribution at pixel t. The HU problem is to recover $\boldsymbol{A}_0$ from the $\boldsymbol{y}_t$'s, thereby identifying the materials.

The dataset we use is the Cuprite AVIRIS dataset, taken in 1997 in the Cuprite area by airborne visible/infrared imaging spectrometer (AVIRIS) [48]. It is widely used in HU. Previous studies revealed that, for the Cuprite AVIRIS dataset, pure-pixel search already gives very good results [18,19]. Still, it is interesting to use this dataset to demonstrate whether PRISM can provide reasonable results.

Our experiment setups are as follows. We take a preprocessed subimage of the Cuprite AVIRIS dataset with 250×190 pixels and with $M = 186$ bands; see the left figure in Fig. 5. It is believed that there are 12 materials (see, e.g., [49] for a discussion); their names are displayed in Table 3. As a real-data problem, we do not know the ground truth of the materials' spectral signatures. As a standard procedure, the reference spectral signatures from the USGS library [50] (which records the spectral signatures of numerous materials) corresponding to those 12 materials are used as our believed ground truth. We measure the performance by the spectral angle distance (SAD) $\theta_i(\boldsymbol{A}_0, \hat{\boldsymbol{A}}) = \cos^{-1}(\boldsymbol{a}_{0,\pi_i}^\top \hat{\boldsymbol{a}}_i / (\|\boldsymbol{a}_{0,\pi_i}\| \|\hat{\boldsymbol{a}}_i\|))$, where $\boldsymbol{\pi}$ is the solution to $\min_{\boldsymbol{\pi} \in \Pi_N} \frac{1}{N} \sum_{i=1}^{N} \cos^{-1}(\boldsymbol{a}_{0,\pi_i}^\top \hat{\boldsymbol{a}}_i / (\|\boldsymbol{a}_{0,\pi_i}\| \|\hat{\boldsymbol{a}}_i\|))$.

We consider SVMAX, SISAL, RVolMin and VIA-PRISM. We set $N = 12$. SISAL has its regularization parameter tuned as $\lambda = 0.001$. RVolMin has its parameters set as $\lambda = 6$, $p = 1.5$. We employ an improved version of VIA-PRISM, wherein we estimate the vertex matrix $\boldsymbol{A}$ and the noise variance σ^2 jointly by incorporating σ^2 as an extra optimization variable in the ML problem (8). AM is used to deal with the extended ML problem (see, e.g., [40], for its ISA counterpart); we shall omit the details here. Doing so frees us from pre-determining the noise variance σ^2 when using PRISM.

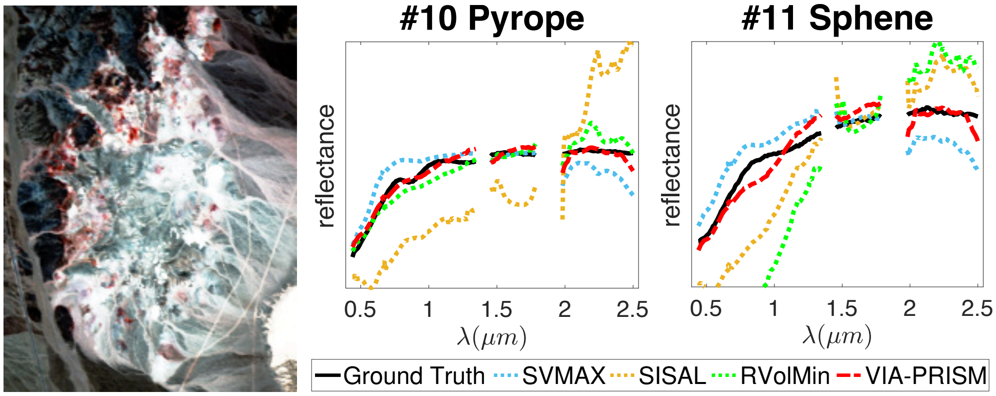


Figure 5: Color composite image from Cuprite Area (left) and the recovered spectra for Pyrope and Sphene (middle and right).

The SAD performance is listed in Table 3. We see that all the schemes show good performance in general. Upon a closer look, SISAL and RVolMin have relatively large estimation errors with "#10 Pyrope" and "#11 Sphene"; see Fig. 5 for the recovered spectral signatures. We suspect that this may be due to noise sensitivity. In comparison, VIA-PRISM appears to be less sensitive. We should reiterate that there is no parameter tuning with VIA-PRISM.

9 Conclusion and Discussion

We studied simplex component analysis under a probabilistic paradigm. Our study revealed the following results.

Table 3: SAD performance on the Cuprite AVIRIS dataset. Blue: the best SAD among all the tested schemes; red: SADs that are larger than 10 degrees.

SAD (degree)	SVMAX	SISAL	RVolMin	VIA-PRISM
average performance	**2.89**	4.55	3.69	3.07
#1 Alunite	**2.33**	2.76	3.12	3.25
#2 Andradite	2.97	2.61	**1.87**	2.27
#3 Buddingtonite	3.28	**3.21**	3.30	3.57
#4 Dumortierite	**2.12**	3.11	2.32	4.88
#5 Kaolinite$_1$	**2.36**	3.38	3.51	4.88
#6 Kaolinite$_2$	2.27	**2.24**	2.71	2.79
#7 Muscovite	**2.49**	3.65	3.30	2.50
#8 Montmorillonite	3.39	2.39	**1.65**	2.09
#9 Nontronite	**2.07**	2.61	2.21	2.36
#10 Pyrope	4.07	14.78	2.54	**1.40**
#11 Sphene	4.86	9.38	15.34	**2.28**
#12 Chalcedony	2.48	4.52	**2.44**	4.53

1. Our framework, called PRISM, works on an identifiable model, provably. Both theory and numerical results suggested that PRISM can leverage large data size to reduce noise.

2. We showed how some powerful state-of-the-art methods, namely, simplex volume minimization (SVMin) methods, are related to PRISM. They also appeared to be good approximations of PRISM for the high-SNR regime.

3. PRISM is a difficult problem; it is an optimization problem that has intractable integrals appearing in the objective function. We studied variational inference approximation (VIA) for PRISM. It was shown that VIA-PRISM resembles regularized matrix factorization, and there is no parameter to manually tune, unlike some regularized matrix factorization and SVMin methods. Our analysis indicated that VIA-PRISM may be a poor approximation under very high SNRs. However our numerical results illustrated that VIA-PRISM works well under the medium- or low-SNR regime, and that VIA-PRISM performs better than the state-of-the-art methods for large latent-variable size N.

4. We examined importance sampling approximation (ISA) for PRISM. ISA-PRISM is easy to use, and empirically it works very well for small N. But, computationally, it does not scale well with N.

As future work we will be interested in efficient realizations of PRISM, e.g., by VIA or by SVMin (improved forms driven by PRISM). Extension of the current probabilistic model to more complex data models also appears promising.

Appendix

A Proof of Fact 2

By $s_N = 1 - \mathbf{1}^\top \bar{\boldsymbol{s}}$, we can write

$$\boldsymbol{x} = (\textstyle\sum_{i=1}^{N-1} \boldsymbol{b}_i s_i) + \boldsymbol{b}_N (1 - \textstyle\sum_{i=1}^{N-1} s_i) = \bar{\boldsymbol{B}} \bar{\boldsymbol{s}} + \boldsymbol{b}_N,$$

where $\bar{\boldsymbol{B}}$ is invertible due to the affine independence of $\boldsymbol{B}$. Since the mapping from $\bar{\boldsymbol{s}}$ to $\boldsymbol{x}$ is bijective, we can apply transformation of random variables to obtain

$$p(\boldsymbol{x}) = \frac{1}{|\det(\bar{\boldsymbol{B}})|} D(\bar{\boldsymbol{B}}^{-1}(\boldsymbol{x} - \boldsymbol{b}_N); \mathbf{1}),$$

where D is defined in (3). It can be verified that

$$\bar{\boldsymbol{B}}^{-1}(\boldsymbol{x} - \boldsymbol{b}_N) \in \tilde{\Delta} \iff \boldsymbol{x} \in \overline{\text{conv}}(\boldsymbol{B}),$$

and hence $D(\bar{\boldsymbol{B}}^{-1}(\boldsymbol{x} - \boldsymbol{b}_N); \mathbf{1}) = (N-1)! \cdot \mathbb{1}_{\overline{\text{conv}}(\boldsymbol{B})}(\boldsymbol{x})$. Also, it can seen from (2) that $\text{vol}(\boldsymbol{B}) = |\det(\bar{\boldsymbol{B}})|/(N-1)!$ when $\bar{\boldsymbol{B}}$ is square. The proof is complete.

B The Formal Proof of ML Identifiability

The formal proof of Theorem 1 is as follows. Consider the following propositions, whose proofs are shown in Appendix C and E.

Proposition 3 *If $\boldsymbol{A}$ is such that $p(\boldsymbol{y}; \boldsymbol{A}) = p(\boldsymbol{y}; \boldsymbol{A}_0)$ for all $\boldsymbol{y}$, then (a)* $\text{aff}(\boldsymbol{A}) = \text{aff}(\boldsymbol{A}_0)$*; (b)* $\text{vol}(\boldsymbol{A}) = \text{vol}(\boldsymbol{A}_0)$.

Proposition 4 *Let $(\boldsymbol{Q}, \boldsymbol{d}) \in \mathbb{R}^{M \times (N-1)} \times \mathbb{R}^M$, with $\boldsymbol{Q}$ being semi-orthogonal (i.e., $\boldsymbol{Q}^\top \boldsymbol{Q} = \boldsymbol{I}$), such that* $\text{aff}(\boldsymbol{A}_0) = \text{span}(\boldsymbol{Q}) + \boldsymbol{d}$*; such $(\boldsymbol{Q}, \boldsymbol{d})$ exists. Let*

$$q(\boldsymbol{z}; \boldsymbol{A}) = (\sqrt{2\pi}\sigma)^{M-N+1} p(\boldsymbol{y} = \boldsymbol{Q}\boldsymbol{z} + \boldsymbol{d}; \boldsymbol{A}), \ \boldsymbol{z} \in \mathbb{R}^{N-1},$$

where, on the right-hand side of the above equation, $p(\boldsymbol{y}; \boldsymbol{A})$ is given in (9)*. If $\boldsymbol{A}$ is such that* $\text{aff}(\boldsymbol{A}) = \text{aff}(\boldsymbol{A}_0)$*, then $q(\boldsymbol{z}; \boldsymbol{A})$ takes the same form as* (15)*, specifically,*

$$q(\boldsymbol{z}; \boldsymbol{A}) = \int_{\mathbb{R}^{N-1}} \varphi_\sigma(\boldsymbol{z} - \boldsymbol{x}) p(\boldsymbol{x}; \boldsymbol{B}) \text{d}\boldsymbol{x} := q(\boldsymbol{z}; \boldsymbol{B}), \tag{50}$$

where $\boldsymbol{B} \in \mathbb{R}^{(N-1) \times N}$ is affinely independent and satisfies $\boldsymbol{A} = \boldsymbol{Q}\boldsymbol{B} + \boldsymbol{d}\mathbf{1}^\top$; $p(\boldsymbol{x}; \boldsymbol{B})$ is defined in (14)*.*

Proposition 3 is obtained by observing the first- and second-order moments of $\boldsymbol{y} \sim p(\cdot; \boldsymbol{A})$. Proposition 4 is a dimensionality reduction result which is reminiscent of [25, Lemma 2] in spirit. Invoking Propositions 3.(a) and 4, we have

$$p(\boldsymbol{y}; \boldsymbol{A}) = p(\boldsymbol{y}; \boldsymbol{A}_0) \ \text{ for all } \boldsymbol{y} \quad \Longrightarrow \quad q(\boldsymbol{z}; \boldsymbol{B}) = q(\boldsymbol{z}; \boldsymbol{B}_0) \ \text{ for all } \boldsymbol{z}, \tag{51}$$

where $\boldsymbol{B}$ and $\boldsymbol{B}_0$, defined by the way as in Proposition 4, are affinely independent. As $q(\boldsymbol{z}; \boldsymbol{B})$ (cf. (50)) takes the same form as $p(\boldsymbol{y}; \boldsymbol{A})$ in (15) in the intuitive proof in Section 4.2, most of the steps in the intuitive proof apply. The exception is the implication

$$\check{p}(\boldsymbol{\xi}; \boldsymbol{B}) = \check{p}(\boldsymbol{\xi}; \boldsymbol{B}_0) \ \text{ for all } \boldsymbol{\xi} \quad \Longrightarrow \quad p(\boldsymbol{x}; \boldsymbol{B}) = p(\boldsymbol{x}; \boldsymbol{B}_0) \ \text{ for all } \boldsymbol{x}. \tag{52}$$

We complete the proof by confirming the implication (52). Consider the FT result below.

Fact 5 *[51, Theorem 1.16] If $f : \mathbb{R}^n \rightarrow \mathbb{R}$ belongs to $L^1(\mathbb{R}^n)$, the space of all measurable functions defined on $\mathbb{R}^n$ and with $\int_{\mathbb{R}^n} \|f(\boldsymbol{x})\|_1 \mathrm{d}\boldsymbol{x} < \infty$ ($\| \cdot \|_1$ denotes the 1-norm), then*

$$\int_{\mathbb{R}^n} \check{\varphi}_\varepsilon(\boldsymbol{\xi}) \check{f}(\boldsymbol{\xi}) e^{j2\pi \boldsymbol{\xi}^\top \boldsymbol{z}} \mathrm{d}\boldsymbol{\xi} = \int_{\mathbb{R}^n} \varphi_\varepsilon(\boldsymbol{z} - \boldsymbol{x}) f(\boldsymbol{x}) \mathrm{d}\boldsymbol{x}$$

for all $\varepsilon > 0$.

Applying Fact 5 to both sides of (52) gives

$$\int_{\mathbb{R}^{N-1}} \varphi_\varepsilon(\boldsymbol{z} - \boldsymbol{x}) p(\boldsymbol{x}; \boldsymbol{B}) \mathrm{d}\boldsymbol{x} = \int_{\mathbb{R}^{N-1}} \varphi_\varepsilon(\boldsymbol{z} - \boldsymbol{x}) p(\boldsymbol{x}; \boldsymbol{B}_0) \mathrm{d}\boldsymbol{x}. \tag{53}$$

Also, we apply Proposition 3.(b) to (51) (with $\boldsymbol{A} = \boldsymbol{B}$) to get $\mathrm{vol}(\boldsymbol{B}) = \mathrm{vol}(\boldsymbol{B}_0)$. Eq. (53) can thus be reduced to

$$\int_{\overline{\mathrm{conv}}(\boldsymbol{B})} \varphi_\varepsilon(\boldsymbol{z} - \boldsymbol{x}) \mathrm{d}\boldsymbol{x} = \int_{\overline{\mathrm{conv}}(\boldsymbol{B}_0)} \varphi_\varepsilon(\boldsymbol{z} - \boldsymbol{x}) \mathrm{d}\boldsymbol{x}.$$

It can be shown that, for an open convex set $\mathcal{C}$, we have[6]

$$\lim_{\varepsilon \rightarrow 0} \int_{\mathcal{C}} \varphi_\varepsilon(\boldsymbol{z} - \boldsymbol{x}) \mathrm{d}\boldsymbol{x} \begin{cases} = 1, & \boldsymbol{z} \in \mathcal{C}, \\ \leq 1/2, & \boldsymbol{z} \notin \mathcal{C}. \end{cases} \tag{54}$$

The above two equations lead to

$$\boldsymbol{z} \in \overline{\mathrm{conv}}(\boldsymbol{B}) \quad \Longleftrightarrow \quad \boldsymbol{z} \in \overline{\mathrm{conv}}(\boldsymbol{B}_0),$$

and subsequently, $p(\boldsymbol{x}; \boldsymbol{B}) = p(\boldsymbol{x}; \boldsymbol{B}_0)$. The proof is complete.

C Proof of Proposition 3

First, consider the mean and covariance of $\boldsymbol{y}$ for a given $\boldsymbol{A}$. Recall from the model in Section 3 of the main manuscript that $\boldsymbol{y} = \boldsymbol{A}\boldsymbol{s} + \boldsymbol{v}$, $\boldsymbol{s} \sim D(\cdot; \boldsymbol{1})$, $\boldsymbol{v} \sim \varphi_\sigma$, with $\boldsymbol{v}$ independent of $\boldsymbol{s}$. Using Fact 1, we get

$$\begin{aligned} &\mathbb{E}[\boldsymbol{y}] = \tfrac{1}{N} \boldsymbol{A}\boldsymbol{1}, \qquad (55) \\ &\boldsymbol{C}_{\boldsymbol{y}} := \mathrm{Cov}(\boldsymbol{y}) = \boldsymbol{A}\boldsymbol{C}_{\boldsymbol{s}}\boldsymbol{A}^\top + \sigma^2 \boldsymbol{I}, \\ &\boldsymbol{C}_{\boldsymbol{s}} := \mathrm{Cov}(\boldsymbol{s}) = \tfrac{1}{(N+1)N} \left(\boldsymbol{I} - \tfrac{1}{N} \boldsymbol{1}\boldsymbol{1}^\top \right). \end{aligned}$$

Let $\boldsymbol{U} \in \mathbb{R}^{N \times (N-1)}$ be any semi-orthogonal matrix such that $\boldsymbol{U}^\top \boldsymbol{1} = \boldsymbol{0}$. It can be verified that $\boldsymbol{U}\boldsymbol{U}^\top = \boldsymbol{I} - \boldsymbol{1}\boldsymbol{1}^\top / N$. The covariance $\boldsymbol{C}_{\boldsymbol{y}}$ can thus be simplified to

$$\boldsymbol{C}_{\boldsymbol{y}} = \tfrac{1}{(N+1)N} (\boldsymbol{A}\boldsymbol{U})(\boldsymbol{A}\boldsymbol{U})^\top + \sigma^2 \boldsymbol{I}. \tag{56}$$

Second, suppose $p(\boldsymbol{y}; \boldsymbol{A}) = p(\boldsymbol{y}; \boldsymbol{A}_0)$ for all $\boldsymbol{y}$. We see from (55) and (56) that

$$\tfrac{1}{N} \boldsymbol{A}\boldsymbol{1} = \tfrac{1}{N} \boldsymbol{A}_0 \boldsymbol{1}, \tag{57}$$

$$(\boldsymbol{A}\boldsymbol{U})(\boldsymbol{A}\boldsymbol{U})^\top = (\boldsymbol{A}_0\boldsymbol{U})(\boldsymbol{A}_0\boldsymbol{U})^\top. \tag{58}$$

Consider the following results.

[6] Hint: for the uppercase of (54), note that $\boldsymbol{z} \in \mathcal{C}$ is an interior point of $\mathcal{C}$. For the lowercase, consider $\int_{\mathcal{H}} \varphi_\varepsilon(\boldsymbol{z} - \boldsymbol{x}) \mathrm{d}\boldsymbol{x}$ where $\mathcal{H}$, $\mathcal{C} \subset \mathcal{H}$, is a halfspace associated with the separating hyperplane of $\boldsymbol{z}$ and $\mathcal{C}$.

Lemma 1 *Let* $\boldsymbol{A} \in \mathbb{R}^{m\times n}$ *be any matrix. Let* $\boldsymbol{U} \in \mathbb{R}^{n\times(n-1)}$ *be a semi-orthogonal matrix such that* $\boldsymbol{U}^\top \boldsymbol{1} = \boldsymbol{0}$. *Then*

(a) $\mathrm{aff}(\boldsymbol{A}) = \mathrm{span}(\bar{\boldsymbol{A}}) + \boldsymbol{d}$ *for any* $\boldsymbol{d} \in \mathrm{aff}(\boldsymbol{A})$*;*

(b) $\mathrm{span}(\bar{\boldsymbol{A}}) = \mathrm{span}(\boldsymbol{A}\boldsymbol{U})$*;*

(c) $\det(\bar{\boldsymbol{A}}^\top \bar{\boldsymbol{A}}) = C \det((\boldsymbol{A}\boldsymbol{U})^\top(\boldsymbol{A}\boldsymbol{U}))$ *for some constant* C*.*

The proof of Lemma 1 will be shown in Appendix D. We will also need the following results.

Lemma 2 *Let* $\boldsymbol{F}, \boldsymbol{G} \in \mathbb{R}^{m\times n}$, $m \geq n$. *If* $\boldsymbol{F}\boldsymbol{F}^\top = \boldsymbol{G}\boldsymbol{G}^\top$,

(a) $\mathrm{span}(\boldsymbol{F}) = \mathrm{span}(\boldsymbol{G})$*;*

(b) $\det(\boldsymbol{F}^\top\boldsymbol{F}) = \det(\boldsymbol{G}^\top\boldsymbol{G})$*.*

We omit the proof of Lemma 2 as it is basic in matrix analysis (hint: use singular value decomposition). By Lemma 1.(a)–(b) we have

$$\begin{aligned} \mathrm{aff}(\boldsymbol{A}) &= \mathrm{span}(\boldsymbol{A}\boldsymbol{U}) + \tfrac{1}{N}\boldsymbol{A}\boldsymbol{1}, \\ \mathrm{aff}(\boldsymbol{A}_0) &= \mathrm{span}(\boldsymbol{A}_0\boldsymbol{U}) + \tfrac{1}{N}\boldsymbol{A}_0\boldsymbol{1}. \end{aligned} \tag{59}$$

By (58) and Lemma 2.(a), we have $\mathrm{span}(\boldsymbol{A}\boldsymbol{U}) = \mathrm{span}(\boldsymbol{A}_0\boldsymbol{U})$. In addition, since $\frac{1}{N}\boldsymbol{A}\boldsymbol{1} = \frac{1}{N}\boldsymbol{A}_0\boldsymbol{1}$ (cf. (57)), we have $\mathrm{aff}(\boldsymbol{A}) = \mathrm{aff}(\boldsymbol{A}_0)$. Moreover, from (58), Lemma 1.(c) and Lemma 2.(b), we obtain $\det(\bar{\boldsymbol{A}}^\top\bar{\boldsymbol{A}}) = \det(\bar{\boldsymbol{A}}_0^\top\bar{\boldsymbol{A}}_0)$. This leads to $\mathrm{vol}(\boldsymbol{A}) = \mathrm{vol}(\boldsymbol{A}_0)$. The proof is complete.

D Proof of Lemma 1

Lemma 1.(a) is basic. Concisely, from the definition of aff, it is immediate that $\mathrm{aff}(\boldsymbol{A}) = \mathrm{span}(\bar{\boldsymbol{A}}) + \boldsymbol{a}_n$. As $\boldsymbol{d} \in \mathrm{aff}(\boldsymbol{A})$, we can write $\boldsymbol{d} = \boldsymbol{b} + \boldsymbol{a}_n$ for some $\boldsymbol{b} \in \mathrm{span}(\bar{\boldsymbol{A}})$. As $\mathrm{span}(\bar{\boldsymbol{A}}) + \boldsymbol{b} = \mathrm{span}(\bar{\boldsymbol{A}})$, we have Lemma 1.(a).

For Lemma 1.(b) it suffices to show $\mathrm{span}(\boldsymbol{U}) = \mathrm{span}(\bar{\boldsymbol{I}})$. By noting $\bar{\boldsymbol{I}} = [\, \boldsymbol{e}_1 - \boldsymbol{e}_n, \ldots, \boldsymbol{e}_{n-1} - \boldsymbol{e}_n \,]$, we see that

$$\begin{aligned} \boldsymbol{x} \in \mathrm{span}(\bar{\boldsymbol{I}}) &\iff \boldsymbol{x} = (x_1, \ldots, x_{n-1}, -\textstyle\sum_{i=1}^{n-1} x_i) \\ &\iff \boldsymbol{x} \in \mathcal{N}(\boldsymbol{1}^\top) := \{\boldsymbol{x} \in \mathbb{R}^n \mid \boldsymbol{1}^\top\boldsymbol{x} = 0\}. \end{aligned}$$

The set $\mathcal{N}(\boldsymbol{1}^\top)$ equals the orthogonal complement of $\mathrm{span}(\boldsymbol{1})$, which is $\mathrm{span}(\boldsymbol{U})$. The proof is done.

For Lemma 1.(c), we have the following. Since $\mathrm{span}(\boldsymbol{U}) = \mathrm{span}(\bar{\boldsymbol{I}})$ in the preceding proof, we can write $\bar{\boldsymbol{I}} = \boldsymbol{U}\boldsymbol{\Theta}$ for some $\boldsymbol{\Theta} \in \mathbb{R}^{(n-1)\times(n-1)}$. By equaling $\bar{\boldsymbol{A}} = \boldsymbol{A}\bar{\boldsymbol{I}} = \boldsymbol{A}\boldsymbol{U}\boldsymbol{\Theta}$,

$$\begin{aligned} \det(\bar{\boldsymbol{A}}^\top\bar{\boldsymbol{A}}) &= \det(\boldsymbol{\Theta}^\top((\boldsymbol{A}\boldsymbol{U})^\top(\boldsymbol{A}\boldsymbol{U}))\boldsymbol{\Theta}) \\ &= (\det(\boldsymbol{\Theta}))^2 \det((\boldsymbol{A}\boldsymbol{U})^\top(\boldsymbol{A}\boldsymbol{U})), \end{aligned}$$

where we exploit the fact that $\boldsymbol{\Theta}$ and $(\boldsymbol{A}\boldsymbol{U})^\top(\boldsymbol{A}\boldsymbol{U})$ are square. Letting $C = (\det(\boldsymbol{\Theta}))^2$ completes the proof.

E Proof of Proposition 4

First it is clear from Lemma 1.(a) that we can write $\mathrm{aff}(\boldsymbol{A}_0) = \mathrm{span}(\boldsymbol{Q}) + \boldsymbol{d}$; here $\boldsymbol{Q}$ is an orthogonal basis for $\mathrm{span}(\bar{\boldsymbol{A}}_0)$, and $\boldsymbol{Q}$ has $N-1$ columns because $\bar{\boldsymbol{A}}_0$ has full-column rank. The identity $\mathrm{aff}(\boldsymbol{A}) = \mathrm{aff}(\boldsymbol{A}_0)$ implies that i) $\boldsymbol{A}$ is affinely independent, and ii) we can write $\boldsymbol{a}_i = \boldsymbol{Q}\boldsymbol{b}_i + \boldsymbol{d}$ for some coefficient $\boldsymbol{b}_i \in \mathbb{R}^{N-1}$. This leads to

$$\boldsymbol{A} = \boldsymbol{Q}\boldsymbol{B} + \boldsymbol{d}\mathbf{1}^\top \iff \boldsymbol{B} = \boldsymbol{Q}^\top(\boldsymbol{A} - \boldsymbol{d}\mathbf{1}^\top).$$

Also, $\boldsymbol{B}$ must be affinely independent, for otherwise $\boldsymbol{A}$ will be affinely dependent as one may verify.

Second, consider $\|\boldsymbol{y} - \boldsymbol{A}\boldsymbol{s}\|^2$ for $\boldsymbol{y} = \boldsymbol{Q}\boldsymbol{z} + \boldsymbol{d}$, $\boldsymbol{s} \in \Delta$:

$$\begin{aligned}\|\boldsymbol{y} - \boldsymbol{A}\boldsymbol{s}\|^2 &= \|\boldsymbol{Q}\boldsymbol{z} + \boldsymbol{d} - (\boldsymbol{Q}\boldsymbol{B} + \boldsymbol{d}\mathbf{1}^\top)\boldsymbol{s}\|^2 \\ &= \|\boldsymbol{Q}(\boldsymbol{z} - \boldsymbol{B}\boldsymbol{s})\|^2 = \|\boldsymbol{z} - \boldsymbol{B}\boldsymbol{s}\|^2,\end{aligned}$$

where we have used $\mathbf{1}^\top \boldsymbol{s} = 1$. Putting the above equality to the PDF $p(\boldsymbol{y}; \boldsymbol{A})$ in (9), we obtain

$$\begin{aligned}q(\boldsymbol{z}; \boldsymbol{A}) &= (\sqrt{2\pi}\sigma)^{M-N+1} p(\boldsymbol{y} = \boldsymbol{Q}\boldsymbol{z} + \boldsymbol{d}; \boldsymbol{A}) \\ &= (N-1)! \int \varphi_\sigma(\boldsymbol{z} - \boldsymbol{B}\boldsymbol{s}) \mathbb{1}_{\bar{\Delta}}(\boldsymbol{s}) \mathrm{d}\mu(\boldsymbol{s}).\end{aligned}$$

By the change of variables $\boldsymbol{x} = \boldsymbol{B}\boldsymbol{s} = \bar{\boldsymbol{B}}\bar{\boldsymbol{s}} + \boldsymbol{b}_N$, one will find that

$$q(\boldsymbol{z}; \boldsymbol{A}) = \frac{(N-1)!}{\det(\bar{\boldsymbol{B}})} \int_{\mathbb{R}^{N-1}} \varphi_\sigma(\boldsymbol{z} - \boldsymbol{x}) \mathbb{1}_{\overline{\mathrm{conv}}(\boldsymbol{B})}(\boldsymbol{x}) \mathrm{d}\boldsymbol{x};$$

the proof is identical to that in Fact 2. The proof is complete.

F Dimensionality Reduction

Here we describe the dimensionality reduction (DR) procedure for simplex component analysis and explain why it works.

Algorithm 2 DR for simplex component analysis

```
given a collection of data points $\boldsymbol{y}_1, \ldots, \boldsymbol{y}_T \in \mathbb{R}^M$ and the model order $N$
compute the sample mean $\hat{\boldsymbol{\mu}}_y = (\sum_{t=1}^T \boldsymbol{y}_t)/T$
compute the sample covariance $\hat{\boldsymbol{C}}_y = (\sum_{t=1}^T (\boldsymbol{y}_t - \hat{\boldsymbol{\mu}}_y)(\boldsymbol{y}_t - \hat{\boldsymbol{\mu}}_y)^\top)/T$
compute the eigenvectors of $\hat{\boldsymbol{C}}_{\boldsymbol{y}}$ associated with the first $N-1$ principal eigenvalues, store them
  in $\hat{\boldsymbol{Q}} \in \mathbb{R}^{M \times (N-1)}$
apply the DR
    $\boldsymbol{z}_t = \hat{\boldsymbol{Q}}^\top(\boldsymbol{y}_t - \hat{\boldsymbol{\mu}}_y), \quad t = 1, \ldots, T,$        (60)
  where $\boldsymbol{z}_t \in \mathbb{R}^{N-1}$ is a dimension-reduced point of $\boldsymbol{y}_t$
output $\boldsymbol{z}_1, \ldots \boldsymbol{z}_T$
```

We begin by posting the DR procedure in Algorithm 2. It is the same as the standard PCA, but there is a difference with its result and explanation in the context of simplex component analysis. First, we describe the result. Recall the simplex component analysis model

$$\boldsymbol{y}_t = \boldsymbol{A}_0 \boldsymbol{s}_t + \boldsymbol{v}_t. \tag{61}$$

We argue that the dimension-reduced points $\boldsymbol{z}_t$'s outputted by Algorithm 2 can be modeled as

$$\boldsymbol{z}_t = \boldsymbol{B}_0 \boldsymbol{s}_t + \boldsymbol{w}_t, \tag{62}$$

where $\boldsymbol{B}_0 \in \mathbb{R}^{(N-1)\times N}$ is affinely independent and should satisfy the relation

$$\boldsymbol{A}_0 = \hat{\boldsymbol{Q}} \boldsymbol{B}_0 + \hat{\boldsymbol{\mu}}_y \boldsymbol{1}^\top; \tag{63}$$

$\boldsymbol{w}_t$ is Gaussian noise with mean $\boldsymbol{0}$ and covariance $\sigma^2 \boldsymbol{I}$. Since (62) takes the same form as (61), we can perform simplex component analysis by i) estimating $\boldsymbol{B}_0$ from the dimension-reduced points $\boldsymbol{z}_1, \ldots, \boldsymbol{z}_T$ via a simplex component analysis algorithm, and then ii) forming an estimated $\boldsymbol{A}_0$ from the estimated $\boldsymbol{B}_0$ via (63).

Second, we explain why the above result holds. According to the derivations in Appendix C, the mean and covariance of $\boldsymbol{y}_t$ under (61) are

$$\begin{aligned} \boldsymbol{\mu}_y &:= \mathbb{E}[\boldsymbol{y}_t] = \frac{1}{N} \boldsymbol{A}_0 \boldsymbol{1}, \\ \boldsymbol{C}_y &:= \mathrm{Cov}(\boldsymbol{y}_t) = \tfrac{1}{(N+1)N} (\boldsymbol{A}_0 \boldsymbol{U})(\boldsymbol{A}_0 \boldsymbol{U})^\top + \sigma^2 \boldsymbol{I}, \end{aligned} \tag{64}$$

for a semi-orthogonal $\boldsymbol{U} \in \mathbb{R}^{N \times (N-1)}$ such that $\boldsymbol{U}^\top \boldsymbol{1} = \boldsymbol{0}$. Let $\boldsymbol{Q} \in \mathbb{R}^{M \times (N-1)}$ be a matrix whose columns are the eigenvectors of $\boldsymbol{C}_y$ associated with the first $N-1$ principal eigenvalues. It can be shown from (64) that $\mathrm{span}(\boldsymbol{Q}) = \mathrm{span}(\boldsymbol{A}_0 \boldsymbol{U}) = \mathrm{span}(\bar{\boldsymbol{A}}_0)$; note that we use the assumption that $\boldsymbol{A}_0$ is affinely independent. Moreover, as a corollary of the derivations in Appendix E, $\boldsymbol{A}_0$ can be expressed as

$$\boldsymbol{A}_0 = \boldsymbol{Q} \boldsymbol{B}_0 + \boldsymbol{\mu}_y \boldsymbol{1}^\top \iff \boldsymbol{B}_0 = \boldsymbol{Q}^\top (\boldsymbol{A} - \boldsymbol{\mu}_y \boldsymbol{1}^\top), \tag{65}$$

with $\boldsymbol{B}_0$ being affinely independent. Applying the left-hand side of (65) to (60) yields the dimension-reduced data model (62). Note that Algorithm 2 uses sample mean and sample covariance to do the above DR task. Our explanation is done.

As a minor remark, the above explanation is slightly new. The prior study [6, 25] derived Algorithm 2 by using the deterministic CG notion, which assumes no noise. The above explanation is statistical and assumes the presence of noise.

G Further Discussion with the SVMin-PRISM Relationships

We discuss two issues arising from the SVMin-PRISM relationships shown in Section 5 of the main manuscript. The first issue is that we focused on the case of $M = N-1$, and one may ask whether this can be lifted to $M \geq N-1$. We give two answers, one simple and one complicated. The simple one is no, but the issue can be easily sidestepped. Our derivations exploit friendly properties of full-dimensional simplices, and that restricts us to $M = N-1$. But it is well-known in CG that we can handle the issue by dimensionality reduction [6, 25]; e.g., the one in Appendix F. The complicated answer is that it is possible, but the result will not be simple. In Appendix H, we show that the extension of the edge-smooth approximation in Section 5.2 to $M \geq N-1$ gives rise to an approximate ML

$$\min_{\boldsymbol{A} \in \mathcal{A}, \boldsymbol{S} \in \Delta^T} \log \mathrm{vol}(\boldsymbol{A}) + \tfrac{1}{\lambda T} \|\boldsymbol{Y} - \boldsymbol{A}\boldsymbol{S}\|^2 + \tfrac{1}{T} \left(\tfrac{1}{2\sigma^2} - \tfrac{1}{\lambda} \right) \min_{\boldsymbol{\Xi}^\top \boldsymbol{1} = \boldsymbol{1}} \|\boldsymbol{Y} - \boldsymbol{A}\boldsymbol{\Xi}\|^2.$$

We see that the above problem resembles the SVMin-SSMF (21), but it has an additional term (the third term).

The second issue we want to discuss is with the affine independence constraint; specifically, $\boldsymbol{A} \in \mathcal{A}$ or $\boldsymbol{B} \in \mathcal{B}$ in (19), (21), (26), (28), etc. This constraint is necessary in order to be mathematically correct in our development, but it is often discarded for ease of realization in practice. It is generally safe to remove the affine independence constraint from the original noiseless SVMin (19), SISAL (26) and the chance-promoting SVMin (28); e.g., if the $\boldsymbol{y}_t$'s are well spread such that any data enclosing simplex $\mathrm{conv}(\boldsymbol{A})$ has to be full-dimensional, then the affine independence of $\boldsymbol{A}$ will be automatically satisfied for (19). For SVMin-SSMF (21), its objective value approaches $-\infty$ as $\boldsymbol{A}$ approaches affine dependence. The trick to get this around is to replace the simplex volume (2) by $(\det(\bar{\boldsymbol{A}}^\top \bar{\boldsymbol{A}} + \varepsilon \boldsymbol{I}))^{1/2}/((N-1)!)$ for some small $\varepsilon > 0$ [27].

H Edge-Smooth Approximation for $M \geq N-1$

Here we extend the edge-smooth approximation in Section 5.2 of the main manuscript to the more general case of $M \geq N-1$. First we reformulate $p(\boldsymbol{y}; \boldsymbol{A})$ in (9) such that we can apply edge-smooth approximation. Let $\boldsymbol{A}$ be affinely independent. According to the proof of Proposition 4, we know that i) the affine hull of $\boldsymbol{A}$ can be characterized as $\mathrm{aff}(\boldsymbol{A}) = \mathrm{span}(\boldsymbol{Q}) + \boldsymbol{d}$ for some semi-orthogonal $\boldsymbol{Q} \in \mathbb{R}^{M \times (N-1)}$ and for some $\boldsymbol{d}$; ii) $\boldsymbol{A}$ can be characterized as $\boldsymbol{A} = \boldsymbol{Q}\boldsymbol{B} + \boldsymbol{d}\boldsymbol{1}^\top$ for some affinely independent $\boldsymbol{B} \in \mathbb{R}^{(N-1)\times N}$. Let $\tilde{\boldsymbol{Q}} \in \mathbb{R}^{M \times (M-N+1)}$ be such that $\boldsymbol{U} = [\ \boldsymbol{Q}\ \tilde{\boldsymbol{Q}}\ ]$ is orthogonal. For any $\boldsymbol{s} \in \Delta$ we have

$$\begin{aligned} \|\boldsymbol{y} - \boldsymbol{A}\boldsymbol{s}\|^2 &= \|\boldsymbol{U}^\top[\boldsymbol{y} - (\boldsymbol{Q}\boldsymbol{B} + \boldsymbol{d}\boldsymbol{1}^\top)\boldsymbol{s}]\|^2 \\ &= \|\boldsymbol{z}_1 - \boldsymbol{B}\boldsymbol{s}\|^2 + \|\boldsymbol{z}_2\|^2, \end{aligned} \tag{66}$$

where $\boldsymbol{z}_1$ and $\boldsymbol{z}_2$ are defined as

$$\begin{bmatrix} \boldsymbol{z}_1 \\ \boldsymbol{z}_2 \end{bmatrix} = \begin{bmatrix} \boldsymbol{Q}^\top(\boldsymbol{y} - \boldsymbol{d}) \\ \tilde{\boldsymbol{Q}}^\top(\boldsymbol{y} - \boldsymbol{d}) \end{bmatrix} = \boldsymbol{U}^\top(\boldsymbol{y} - \boldsymbol{d}).$$

With (66), we can write $\varphi_\sigma(\boldsymbol{y} - \boldsymbol{A}\boldsymbol{s}) = \varphi_\sigma(\boldsymbol{z}_2) \cdot \varphi_\sigma(\boldsymbol{z}_1 - \boldsymbol{B}\boldsymbol{s})$. The PDF $p(\boldsymbol{y}; \boldsymbol{A})$ in (9) can be rewritten as

$$\begin{aligned} p(\boldsymbol{y}; \boldsymbol{A}) &= \varphi_\sigma(\boldsymbol{z}_2) \cdot \int \varphi_\sigma(\boldsymbol{z}_1 - \boldsymbol{B}\boldsymbol{s}) p(\boldsymbol{s}) \mathrm{d}\mu(\boldsymbol{s}) & (67) \\ &= \varphi_\sigma(\boldsymbol{z}_2) \cdot \int_{\mathbb{R}^{N-1}} \varphi_\sigma(\boldsymbol{z}_1 - \boldsymbol{x}) p(\boldsymbol{x}; \boldsymbol{B}) \mathrm{d}\boldsymbol{x}, & (68) \end{aligned}$$

where $p(\boldsymbol{x}; \boldsymbol{B})$ is given in (14); here we transform (67) to (68) by the same formulation as in (15).

Second we apply the same edge-smooth approximation in (20) to the integral (68); i.e.,

$$\int_{\mathbb{R}^{N-1}} \varphi_\sigma(\boldsymbol{z}_1 - \boldsymbol{x}) p(\boldsymbol{x}; \boldsymbol{B}) \mathrm{d}\boldsymbol{x} \approx \frac{1}{\mathrm{vol}(\boldsymbol{B})} e^{-\frac{1}{\lambda} \cdot \mathrm{dist}(\boldsymbol{z}_1, \mathrm{conv}(\boldsymbol{B}))^2}.$$

This results in

$$-\log p(\boldsymbol{y}; \boldsymbol{A}) \approx \frac{1}{2\sigma^2}\|\boldsymbol{z}_2\|^2 + \log \mathrm{vol}(\boldsymbol{B}) + \frac{1}{\lambda} \cdot \mathrm{dist}(\boldsymbol{z}_1, \mathrm{conv}(\boldsymbol{B}))^2. \tag{69}$$

Now, observe that

$$\begin{aligned}
\mathrm{dist}(\boldsymbol{y}, \mathrm{conv}(\boldsymbol{A}))^2 &= \min_{\boldsymbol{s}\in\Delta} \|\boldsymbol{y} - \boldsymbol{A}\boldsymbol{s}\|^2 \\
&= \min_{\boldsymbol{s}\in\Delta} \|\boldsymbol{z}_1 - \boldsymbol{B}\boldsymbol{s}\|^2 + \|\boldsymbol{z}_2\|^2 \\
&= \mathrm{dist}(\boldsymbol{z}_1, \mathrm{conv}(\boldsymbol{B}))^2 + \|\boldsymbol{z}_2\|^2 \qquad (70) \\
\mathrm{dist}(\boldsymbol{y}, \mathrm{aff}(\boldsymbol{A}))^2 &= \min_{\boldsymbol{1}^\top\boldsymbol{s}=1} \|\boldsymbol{y} - \boldsymbol{A}\boldsymbol{s}\|^2 \\
&= \min_{\boldsymbol{1}^\top\boldsymbol{s}=1} \|\boldsymbol{z}_1 - \boldsymbol{B}\boldsymbol{s}\|^2 + \|\boldsymbol{z}_2\|^2 \\
&= \min_{\bar{\boldsymbol{s}}\in\mathbb{R}^{N-1}} \|\boldsymbol{z}_1 - \boldsymbol{b}_N - \bar{\boldsymbol{B}}\bar{\boldsymbol{s}}\|^2 + \|\boldsymbol{z}_2\|^2 \\
&= \|\boldsymbol{z}_2\|^2, \qquad (71) \\
\det(\bar{\boldsymbol{A}}^\top\bar{\boldsymbol{A}}) &= \det((\boldsymbol{Q}\bar{\boldsymbol{B}})^\top(\boldsymbol{Q}\bar{\boldsymbol{B}})) = \det(\bar{\boldsymbol{B}}^\top\bar{\boldsymbol{B}}), \qquad (72)
\end{aligned}$$

where (70) is due to (66); (71) is due to (66) and the fact that $\bar{\boldsymbol{B}}$ is invertible; (72) is due to $\bar{\boldsymbol{a}}_i = \boldsymbol{a}_i - \boldsymbol{a}_N = \boldsymbol{Q}(\boldsymbol{b}_i - \boldsymbol{b}_N) = \boldsymbol{Q}\bar{\boldsymbol{b}}_i$. Substituting (70)–(72) into (69), and then applying it to the ML problem (8), we obtain the edge-smooth approximation of the ML problem as

$$\min_{\boldsymbol{A}\in\mathcal{A},\boldsymbol{S}\in\Delta^T} \log\mathrm{vol}(\boldsymbol{A}) + \tfrac{1}{\lambda T}\|\boldsymbol{Y} - \boldsymbol{A}\boldsymbol{S}\|^2 + \tfrac{1}{T}\left(\tfrac{1}{2\sigma^2} - \frac{1}{\lambda}\right) \min_{\boldsymbol{\Xi}^\top\boldsymbol{1}=\boldsymbol{1}} \|\boldsymbol{Y} - \boldsymbol{A}\boldsymbol{\Xi}\|^2.$$

I Proof of Proposition 2

We will assume $\alpha > 0$ without mentioning. By the expansion

$$\log\Gamma(\alpha) = -\gamma\alpha - \log\alpha + \sum_{k=1}^{\infty}\left[\frac{\alpha}{k} - \log\left(1 + \frac{\alpha}{k}\right)\right],$$

where γ is the Euler–Mascheroni constant (see, e.g., [52, p. 204]), one can show that the function h in (43d) has its double differentiation given by

$$h''(\alpha) = \sum_{k=0}^{\infty}\left[\frac{1}{(\alpha+k)^2} - \frac{2(\alpha-1)}{(\alpha+k)^3}\right].$$

We will show that

$$h''(\alpha) \geq \frac{1}{(n+\alpha)^2} > 0, \quad n = \max\{0, \lceil 2\alpha - 3\rceil\},$$

and thereby confirms the strict convexity of h (here $\lceil x\rceil$ denotes the ceiling of x). To facilitate, let

$$f_\alpha(y) = \frac{1}{(\alpha+y)^2} - \frac{2(\alpha-1)}{(\alpha+y)^3}$$

and write

$$h''(\alpha) = \underbrace{\sum_{k=0}^{n-1} f_\alpha(k)}_{:=p(\alpha)} + \underbrace{\sum_{k=n}^{\infty} f_\alpha(k)}_{:=q(\alpha)}.$$

Our task is to derive lower bounds of p and q.

First we deal with q. From

$$f'_\alpha(y) = -\frac{2}{(\alpha+y)^3} + \frac{6(\alpha-1)}{(\alpha+y)^4} = -\frac{2}{(\alpha+y)^4}(y - 2\alpha + 3),$$

we see that $f_\alpha(y)$ is nonincreasing for $y \geq n$ (note $n \geq 2\alpha - 3$). This implies that, for $k \geq n$, we have $f_\alpha(k) \geq f_\alpha(y)$ for any $y \in [k, k+1]$. We can therefore write

$$\begin{aligned} q(\alpha) &\geq \sum_{k=n}^{\infty} \int_k^{k+1} f_\alpha(y)\mathrm{d}y = \int_n^\infty f_\alpha(y)\mathrm{d}y \\ &= \frac{1}{n+\alpha} - \frac{\alpha-1}{(n+\alpha)^2} = \frac{n+1}{(n+\alpha)^2}. \end{aligned} \tag{73}$$

Second we handle p. Let $p_m(\alpha) = \sum_{k=0}^{m-1} f_\alpha(k)$, $m \geq 1$. We claim that

$$p_m(\alpha) \geq -\frac{m}{(m+\alpha)^2}, \tag{74}$$

and consequently, $p(\alpha) = p_n(\alpha) = -n/(n+\alpha)^2$. To show it we use the following inequality

$$-\frac{1}{(m+\alpha)^2} + \frac{2}{(m+\alpha)^3} \geq -\frac{1}{(m+1+\alpha)^2} \tag{75}$$

for any integer $m \geq 1$. The proof of (75) will be provided later. Using (75) with $m = 0$, we observe that

$$p_1(\alpha) = f_\alpha(0) = -\frac{1}{\alpha^2} + \frac{2}{\alpha^3} \geq -\frac{1}{(1+\alpha)^2},$$

which is (74) for the case of $m = 1$. To show the other cases, suppose $p_m(\alpha) \geq -m/(m+\alpha)^2$ is true. Then

$$\begin{aligned} p_{m+1}(\alpha) &= p_m(\alpha) + f_\alpha(m) \\ &\geq -\frac{m}{(m+\alpha)^2} + \frac{1}{(m+\alpha)^2} - \frac{2(\alpha-1)}{(m+\alpha)^3} \\ &= (m+1)\left[-\frac{1}{(m+\alpha)^2} + \frac{2}{(m+\alpha)^3}\right] \\ &\geq -(m+1)\frac{1}{(m+1+\alpha)^2}, \end{aligned}$$

where the last inequality is due to (75). Hence, by induction, we confirm that (74) is true for all $m \geq 1$.

Third, by (73) and (74), we obtain the desired result $h''(\alpha) \geq 1/(n+\alpha)^2$. Before we finish, we should provide the proof of (75). Let $g(y) = 1/(1+y)^2$. Since g is convex on $\mathbb{R}_+$, by the first-order condition of convex functions, i.e., $g(y) \geq g(x) + g'(x)(y-x)$ for $x, y \in \mathbb{R}_+$, we have

$$\begin{aligned} & g(\alpha+m) \geq g(\alpha+m-1) + g'(\alpha+m-1) \\ \iff \quad & \frac{1}{(\alpha+m+1)^2} \geq \frac{1}{(\alpha+m)^2} - \frac{2}{(\alpha+m)^3}. \end{aligned}$$

The proof is complete.

J An ADMM Algorithm for Problem (48)

Here we design an ADMM algorithm for solving the convex problem (48) efficiently. To reduce notational overheads, let $g(\boldsymbol{\alpha}) = g(\boldsymbol{A}, \boldsymbol{\alpha}, \eta; \boldsymbol{y})$. We reformulate problem (48) as

$$\begin{aligned} \min_{\boldsymbol{\alpha}, \boldsymbol{\beta} \in \mathbb{R}^N} \; & g(\boldsymbol{\alpha}) + \textstyle\sum_{i=1}^N h(\beta_i) \\ \text{s.t.}\; & \boldsymbol{1}^\top \boldsymbol{\alpha} = \eta, \;\; \boldsymbol{\beta} > \boldsymbol{0}, \;\; \boldsymbol{\alpha} = \boldsymbol{\beta}, \end{aligned} \tag{76}$$

where we split the variable into two, one for g and another for h. Denote the augmented Lagrangian of the above problem by

$$L_\rho(\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\lambda}) = g(\boldsymbol{\alpha}) + \textstyle\sum_{i=1}^N h(\beta_i) + \boldsymbol{\lambda}^\top(\boldsymbol{\alpha} - \boldsymbol{\beta}) + \frac{\rho}{2}\|\boldsymbol{\alpha} - \boldsymbol{\beta}\|^2,$$

where $\rho > 0$ is given. Following the ADMM literature [47], the ADMM routine for solving problem (76) is

$$\boldsymbol{\alpha}^{k+1} = \arg \min_{\boldsymbol{1}^\top \boldsymbol{\alpha} = \eta} \; L_\rho(\boldsymbol{\alpha}, \boldsymbol{\beta}^k, \boldsymbol{\lambda}^k), \tag{77a}$$

$$\boldsymbol{\beta}^{k+1} = \arg \min_{\boldsymbol{\beta} > \boldsymbol{0}} L_\rho(\boldsymbol{\alpha}^{k+1}, \boldsymbol{\beta}, \boldsymbol{\lambda}^k), \tag{77b}$$

$$\boldsymbol{\lambda}^{k+1} = \boldsymbol{\lambda}^k + \rho(\boldsymbol{\alpha}^{k+1} - \boldsymbol{\beta}^{k+1}), \tag{77c}$$

for $k = 0, 1, \ldots$ and given a starting point $(\boldsymbol{\alpha}^0, \boldsymbol{\beta}^0, \boldsymbol{\lambda}^0)$.

The update step in (77b) is handled as follows. The problem in (77b) collapses into a number of scalar problems

$$\beta_i^{k+1} = \arg \min_{\beta > 0} \; h(\beta) - \lambda_i^k \beta + \frac{\rho}{2}(\alpha_i^{k+1} - \beta)^2, \tag{78}$$

for $i = 1, \ldots, N$. The solution to problem (78) can be obtained by finding $\beta > 0$ such that the derivative of the objective function of (78) is zero; i.e.,

$$0 = h'(\beta) - \lambda_i^k - \rho(\alpha_i^{k+1} - \beta). \tag{79}$$

The derivative h' does not admit a simple expression, although it is computable. It can be shown that $h'(\beta) = (\beta - 1)\psi'(\beta)$, where ψ' is the trigamma function. The trigamma function does not have a closed form, but major numerical software (such as MATLAB) has specialized routines for computing the value of $\psi'(\beta)$ of a given β. We numerically find the solution to (79) by line search, specifically, the bisection search.

The update step in (77a) is a convex quadratic program with one linear equality constraint, and it has a closed form. To formulate properly, rewrite g in (43b) as a quadratic function

$$g(\boldsymbol{\alpha}) = -\boldsymbol{\alpha}^\top \underbrace{\left[\tfrac{1}{\sigma^2 \eta}\left(\boldsymbol{A}^\top \boldsymbol{y} - \tfrac{1}{2(\eta+1)}\mathrm{diag}(\boldsymbol{A}^\top \boldsymbol{A})\right)\right]}_{:=\boldsymbol{b}} + \tfrac{1}{2}\boldsymbol{\alpha}^\top \underbrace{\left(\tfrac{1}{\sigma^2(\eta+1)\eta}\boldsymbol{A}^\top \boldsymbol{A}\right)}_{:=\boldsymbol{C}} \boldsymbol{\alpha},$$

where $\mathrm{diag}(\boldsymbol{X}) = (x_{11}, \ldots, x_{nn})$. The problem in (77a) can be written as

$$\min_{\boldsymbol{1}^\top \boldsymbol{\alpha} = \eta} \; -\boldsymbol{\alpha}^\top(\boldsymbol{b} + \rho \boldsymbol{\beta}^k - \boldsymbol{\lambda}^k) + \tfrac{1}{2}\boldsymbol{\alpha}^\top(\boldsymbol{C} + \rho \boldsymbol{I})\boldsymbol{\alpha}. \tag{80}$$

To derive a closed-form solution to problem (80), note that any $\boldsymbol{\alpha}$ satisfying $\mathbf{1}^\top \boldsymbol{\alpha} = \eta$ can be equivalently represented by

$$\boldsymbol{\alpha} = \tfrac{\eta}{N}\mathbf{1} + \boldsymbol{U}\boldsymbol{\zeta}, \quad \boldsymbol{\zeta} \in \mathbb{R}^{N-1}, \tag{81}$$

where $\boldsymbol{U} \in \mathbb{R}^{N\times(N-1)}$ is a semi-orthogonal matrix such that $\boldsymbol{U}^\top \mathbf{1} = \mathbf{0}$. By the change of variables in (81), we can rewrite problem (80) as

$$\min_{\boldsymbol{\zeta}\in\mathbb{R}^{N-1}} -\boldsymbol{\zeta}^\top \boldsymbol{U}^\top(\boldsymbol{b} + \rho\boldsymbol{\beta}^k - \boldsymbol{\lambda}^k - \tfrac{\eta}{N}\boldsymbol{C}\mathbf{1}) + \tfrac{1}{2}\boldsymbol{\zeta}^\top \boldsymbol{U}^\top(\boldsymbol{C} + \rho\boldsymbol{I})\boldsymbol{U}\boldsymbol{\zeta}.$$

The above problem, as an unconstrained quadratic program, has a closed-form solution

$$\boldsymbol{\zeta}^{k+1} = (\boldsymbol{U}^\top \boldsymbol{C}\boldsymbol{U} + \rho\boldsymbol{I})^{-1}\boldsymbol{U}^\top(\boldsymbol{b} + \rho\boldsymbol{\beta}^k - \boldsymbol{\lambda}^k - \tfrac{\eta}{N}\boldsymbol{C}\mathbf{1});$$

and we obtain $\boldsymbol{\alpha}^{k+1}$ by $\boldsymbol{\alpha}^{k+1} = \frac{\eta}{N}\mathbf{1} + \boldsymbol{U}\boldsymbol{\zeta}^{k+1}$.

References

[1] M. E. Tipping and C. M. Bishop, “Probabilistic principal component analysis,” *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, vol. 61, no. 3, pp. 611–622, 1999.

[2] D. T. Pham and P. Garat, “Blind separation of mixture of independent sources through a quasi-maximum likelihood approach,” *IEEE Trans. Signal Process.*, vol. 45, no. 7, pp. 1712–1725, 1997.

[3] H. Attias, “Independent factor analysis,” *Neural Computation*, vol. 11, no. 4, pp. 803–851, 1999.

[4] I. Khemakhem, D. Kingma, R. Monti, and A. Hyvarinen, “Variational autoencoders and nonlinear ICA: A unifying framework,” in *Proc. 23th Int. Conf. Artificial Intelligence and Statistics*, vol. 108, 2020, pp. 2207–2217.

[5] J. 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.*, 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, Jan 2014.

[7] M. Craig, “Unsupervised unmixing of remotely sensed images,” in *Proc. 5th Australasian Remote Sensing Conf.*, vol. 8, no. 12, 1990, pp. 324–330.

[8] S. Arora, R. Ge, R. Kannan, and A. Moitra, “Computing a nonnegative matrix factorization—Provably,” *SIAM J. Comput.*, vol. 45, no. 4, pp. 1582–1611, 2016.

[9] N. Gillis, “The why and how of nonnegative matrix factorization,” in *Regularization, Optimization, Kernels, and Support Vector Machines.* Chapman and Hall/CRC, 2014, pp. 257–291.

[10] X. Fu, K. Huang, N. D. Sidiropoulos, and W.-K. Ma, "Nonnegative matrix factorization for signal and data analytics: Identifiability, algorithms, and applications," *IEEE Signal Process. Mag.*, vol. 36, no. 2, pp. 59–80, 2019.

[11] L. Chen, P. L. Choyke, T.-H. Chan, C.-Y. Chi, G. Wang, and Y. Wang, "Tissue-specific compartmental analysis for dynamic contrast-enhanced MR imaging of complex tumors," *IEEE Trans. Med. Imag.*, vol. 30, no. 12, pp. 2044–2058, 2011.

[12] 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.

[13] 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.

[14] M. Panov, K. Slavnov, and R. Ushakov, "Consistent estimation of mixed memberships with successive projections," in *Proc. Int. Conf. Complex Networks and their Applications.* Springer, 2017, pp. 53–64.

[15] K. Huang and X. Fu, "Detecting overlapping and correlated communities without pure nodes: Identifiability and algorithm," in *Proc. 36th Int. Conf. Machine Learning*, 2019, pp. 2859–2868.

[16] S. Ibrahim, X. Fu, N. Kargas, and K. Huang, "Crowdsourcing via pairwise co-occurrences: Identifiability and algorithms," in *Advances in Neural Information Processing Systems*, 2019, pp. 7847–7857.

[17] M. E. Winter, "N-findr: An algorithm for fast autonomous spectral end-member determination in hyperspectral data," in *Proc. SPIE Conf. Imaging Spectrometry*, Pasadena, CA, Oct. 1999, pp. 266–275.

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

[19] T.-H. Chan, W.-K. Ma, A. Ambikapathi, and C.-Y. Chi, "A simplex volume maximization framework for hyperspectral endmember extraction," *IEEE Trans. Geosci. Remote Sens.*, vol. 49, no. 11, pp. 4177–4193, 2011.

[20] S. Arora, R. Ge, Y. Halpern, D. Mimno, A. Moitra, D. Sontag, Y. Wu, and M. Zhu, "A practical algorithm for topic modeling with provable guarantees," in *Proc. Int. Conf. Mach. Learn.*, vol. 28(2), 2013, pp. 280–288.

[21] B. Recht, C. Re, J. Tropp, and V. Bittorf, "Factoring nonnegative matrices with linear programs," in *Advances in Neural Information Processing Systems*, 2012, pp. 1214–1222.

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

[23] M. D. Craig, "Minimum-volume transforms for remotely sensed data," *IEEE Trans. Geosci. Remote Sens.*, vol. 32, no. 3, pp. 542–552, May 1994.

[24] J. Li and J. Bioucas-Dias, "Minimum volume simplex analysis: A fast algorithm to unmix hyperspectral data," in *Proc. IEEE IGARSS*, Aug. 2008.

[25] 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, 2009.

[26] 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, Oct. 2015.

[27] 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.

[28] R. Ge and J. Zou, "Intersecting faces: Non-negative matrix factorization with new guarantees," in *Int. Conf. Machine Learning*, 2015, pp. 2295–2303.

[29] C.-H. Lin, R. Wu, W.-K. Ma, C.-Y. Chi, and Y. Wang, "Maximum volume inscribed ellipsoid: A new simplex-structured matrix factorization framework via facet enumeration and convex optimization," *SIAM J. Imag. Sciences*, vol. 11, no. 2, pp. 1651–1679, 2018.

[30] 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, 2007.

[31] J. Bioucas-Dias, "A variable splitting augmented Lagrangian approach to linear spectral unmixing," in *Proc. 2009 First Workshop on Hyperspectral Image and Signal Processing: Evolution in Remote Sensing*, Aug. 2009.

[32] A. Ambikapathi, T.-H. Chan, W.-K. Ma, and C.-Y. Chi, "Chance-constrained robust minimum-volume enclosing simplex algorithm for hyperspectral unmixing," *IEEE Trans. Geosci. Remote Sens.*, vol. 49, no. 11, pp. 4194–4209, 2011.

[33] 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.

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

[35] J. M. Nascimento and J. M. Bioucas-Dias, "Learning dependent sources using mixtures of dirichlet: Applications on hyperspectral unmixing," in *Proc. 2009 First Workshop on Hyperspectral Image and Signal Processing: Evolution in Remote Sensing*, 2009.

[36] 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.

[37] C. M. Bishop, *Pattern Recognition and Machine Learning.* Springer, 2006.

[38] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent Dirichlet allocation," *Journal of Machine Learning Research*, vol. 3, pp. 993–1022, Jan 2003.

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

[40] R. Wu, Q. Li, and W.-K. Ma, "Stochastic ML simplex-structured matrix factorization under the Dirichlet mixture model," in *2019 IEEE Int. Conf. Acoustics Speech Signal Process. (ICASSP)*, 2019, pp. 5561–5565.

[41] Y. Li, R. Wu, and W.-K. Ma, "Stochastic ML estimation for hyperspectral unmixing under endmember variability and nonlinear models," in *2020 IEEE Int. Conf. Acoustics Speech Signal Process. (ICASSP)*, 2020, pp. 5315–5319.

[42] P. Gritzmann, V. Klee, and D. Larman, "Largest j-simplices in n-polytopes," *Discrete and Computational Geometry*, vol. 13, no. 1, pp. 477–515, 1995.

[43] B. A. Frigyik, A. Kapila, and M. R. Gupta, "Introduction to the Dirichlet distribution and related processes," Department of Electrical Engineering, University of Washington, Tech. Rep. UWEETR-2010-0006, 2010.

[44] K. W. Ng, G.-L. Tian, and M.-L. Tang, *Dirichlet and Related Distributions: Theory, Methods and Applications.* John Wiley & Sons, 2011.

[45] Y. Altmann, S. McLaughlin, and N. Dobigeon, "Sampling from a multivariate gaussian distribution truncated on a simplex: a review," in *2014 IEEE Workshop Stat. Signal Process. (SSP)*, 2014, pp. 113–116.

[46] Y. Cong, B. Chen, and M. Zhou, "Fast simulation of hyperplane-truncated multivariate normal distributions," *Bayesian Analysis*, vol. 12, no. 4, pp. 1017–1037, 2017.

[47] S. Boyd, N. Parikh, and E. Chu, *Distributed Optimization and Statistical Learning via the Alternating Direction Method of Multipliers.* Now Publishers Inc, 2011.

[48] 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.

[49] F. Zhu, "Hyperspectral unmixing: ground truth labeling, datasets, benchmark performances and survey," *arXiv preprint arXiv:1708.05125*, 2017.

[50] 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. Geological Survey, Digital Data Series 231*, 2007.

[51] E. M. Stein and G. Weiss, *Introduction to Fourier Analysis on Euclidean Spaces (PMS-32).* Princeton University Press, 2016.

[52] G. Boros and V. Moll, *Irresistible Integrals: Symbolics, Analysis and Experiments in the Evaluation of Integrals.* Cambridge University Press, 2004.