Constrained Block-Term Tensor Decomposition-Based Hyperspectral Unmixing via Alternating Gradient Projection

Meng Ding
School of Math. Sci.
Univ. of Elec. Sci. & Tech. of China
Chengdu, China

Xiao Fu
School of Elec. Eng. & Computer Sci.
Oregon State Univ.
Corvallis, OR 97331, United States

Ting-Zhu Huang, Xi-Le Zhao
School of Math. Sci.
Univ. of Elec. Sci. & Tech. of China
Chengdu, China

***Abstract*—Block-term tensor decomposition (BTD)-based hyperspectral unmixing (HU) is well-motivated because of its identifiability of the endmembers and abundance maps under relatively mild conditions. However, algorithm design of BTD-based HU faces challenges in enforcing hyperspectral image-related structural constraints, e.g., the probability simplex constraints on the abundance vectors—while incorporating structural information is critical for combating noise and enhancing interpretability. Existing work uses a three-block alternating least square (ALS) framework, and employs the multiplicative update (MU) method to handle constraints, but this ALS-MU approach has high per-iteration complexity and often converges slowly. This work puts forth an alternating gradient projection (GP) algorithm for the problem of interest. Our method leverages a two-block parameterization of the BTD model to avoid encountering heavy updates, thereby exhibiting high efficiency. Another core contribution lies in a fast solver for computing a key step in the GP algorithm, namely, the orthogonal projection onto the set of matrices with low-rank and probability simplex structures. Simulations show that the GP framework attains order-of-magnitude speedup and accuracy improvement relative to the state-of-the-art.**

***Index Terms*—Hyperspectral unmixing, constrained block-term tensor decomposition, alternating gradient projection.**

I. Introduction

Hyperspectral images (HSIs) are often acquired with a limited spatial resolution, and thus a pixel may be a mixture of several materials. *Hyperspectral unmixing* (HU) techniques estimate the spectral signatures of the constituent materials (endmembers) and their corresponding proportions (abundances) from the mixtures [1]. The arguably most prominent model for HU is the so-called *linear mixture model* (LMM). Under the LMM, a hyperspectral pixel are expressed as a convex combination of the endmembers. Estimating the endmembers and the abundances of the materials is then recast as a blind linear unmixing problem, which is also often associated with (nonnegative) matrix factorization (NMF); see [1], [2].

The work of M. Ding, T. Huang, and X. Zhao is supported by the National Natural Science Foundation of China (61772003, 61876203), the Key Project of Applied Basic Research in Sichuan Province (2020YJ0216), the Applied Basic Research Project of Sichuan Province (21YYJC3042), and National Key Research and Development Program of China (2020YFA0714001). The work of X. Fu is supported by the National Science Foundation under Projects ECCS 1808159 and ECCS 2024058.

Under the NMF model, the identifiability of the endmembers and abundances holds under relatively restrictive or hard to check conditions [2]. A recent work [3] connected the LMM-based HU problem to the tensor decomposition model with multi-linear rank-$(L_r, L_r, 1)$ block terms (i.e., the LL1 model) [4]. Leveraging the uniqueness of the LL1 model, identifiability of the endmembers and abundance maps can be established under conditions that are fairly different from those used in the NMF model. Hence, the LL1 model is considered a valuable alternative to existing HU frameworks.

However, computing the LL1 decomposition under HU-related structural constraints gives rise to challenging optimization problems. The work in [3] adopts the classic *alternating least squares* (ALS) framework for unconstrained LL1 decomposition [4]. Nonnegativity and probability simplex constraints are added to reflect the physical meaning of the endmembers and abundance maps. Every subproblem in the ALS framework becomes a (large-scale) nonnegative least squares problem. The work in [3] uses the *multiplicative updates* (MU) for handling the subproblems, which requires a large amount of operations per iteration and often leads to slow convergence of the overall algorithm.

In this work, our interest lies in efficient constrained LL1 decomposition for HU. Our approach starts with an equivalent two-block re-parametrization of the matricized LL1 model. Per the block factors' physical meaning in the context of HU, we also impose nonnegativity and simplex constraints on the pertinent factors. The re-parametrization allows us to develop an alternating *gradient projection* (GP) algorithm that circumvents the large-scale subproblems that arise in the ALS framework [3]. The key challenge for realizing the GP framework is that one block's constraint is a nonconvex low-rank and simplex-structured matrix set, and no tractable projector exists. We propose a heuristic solver for this projection problem. The solver consists of simple operations, i.e., truncated SVD and water-filling, and thus is efficient. To support our design, we also show that the projection solver exhibits local linear convergence. Simulations show that our GP algorithm attains substantial efficiency improvement and produces more accurate HU results relative to state-of-the-art.

II. Background: LMM-Based HU

Under the LMM, in the noise-free case, a spectral pixel $\boldsymbol{y}_\ell \in \mathbb{R}^K$ contained in the HSI can be expressed as $\boldsymbol{y}_\ell = \boldsymbol{C}\boldsymbol{s}_\ell$, where $\boldsymbol{C} = [\boldsymbol{c}_1, \ldots, \boldsymbol{c}_R] \in \mathbb{R}^{K\times R}$ denotes the spectral signatures of R endmembers contained in the pixel, and $\boldsymbol{s}_\ell \in \mathbb{R}^R$ is the abundance vector such that $\mathbf{1}^\top \boldsymbol{s}_\ell = 1,\ \boldsymbol{s}_\ell \geq \mathbf{0}$. Collecting all pixels together, we have

$$\boldsymbol{Y} = \boldsymbol{C}\boldsymbol{S}, \tag{1}$$

where $\boldsymbol{S} = [\boldsymbol{s}_1, \ldots, \boldsymbol{s}_N]$, and $\boldsymbol{Y} = [\boldsymbol{y}_1, \ldots, \boldsymbol{y}_N]$ [1]. The LMM can also be expressed as

$$\underline{\boldsymbol{Y}} = \sum_{r=1}^{R} \boldsymbol{S}_r \circ \boldsymbol{C}(:,r),\ \sum_{r=1}^{R} \boldsymbol{S}_r = \mathbf{1}\mathbf{1}^\top,\ \boldsymbol{S}_r \geq \mathbf{0}, \tag{2}$$

where $\circ$ is the outer product, $\boldsymbol{C}(:,r) = \boldsymbol{c}_r$ and $\underline{\boldsymbol{Y}} \in \mathbb{R}^{I\times J\times K}$. Note that $\boldsymbol{Y}$ is often obtained by re-arranging pixels in $\underline{\boldsymbol{Y}}$ via $\boldsymbol{y}_\ell = \underline{\boldsymbol{Y}}(i,j,:)$ with $\ell = i + (j-1)I$. The matrix $\boldsymbol{S}_r = \mathrm{mat}(\boldsymbol{S}(r,:))$ is interpreted as the *abundance map* of endmember r, where the "matricization" operator $\mathrm{mat}(\cdot) : \mathbb{R}^{IJ} \to \mathbb{R}^{I\times J}$ is the inverse operation of "vectorization"; see Fig. 1. LMM-based HU aims at finding $\boldsymbol{S}$ and $\boldsymbol{C}$.

A. NMF-based HU and Identifiability

Since $\boldsymbol{C}$ and $\boldsymbol{S}$ are both nonnegative per their physical meaning, a plethora of NMF based methods were proposed for LMM based HU; see [1], [2]. As a blind estimation problem, the soundness of HU methods is built upon the *identifiability* of $\boldsymbol{C}$ and $\boldsymbol{S}$ from $\boldsymbol{Y}$. The NMF model is in general not identifiable [2], since one can often find invertible $\boldsymbol{Q}$ such that $\widetilde{\boldsymbol{C}} = \boldsymbol{C}\boldsymbol{Q} \geq \mathbf{0}$ and $\widetilde{\boldsymbol{S}} = \boldsymbol{Q}^{-1}\boldsymbol{S} \geq \mathbf{0}$—and $\boldsymbol{Y} = \widetilde{\boldsymbol{C}}\widetilde{\boldsymbol{S}}$ still holds. The identifiability consideration often leads to more complex NMF criteria, e.g., the volume minimization based NMF; see, e.g., [5]. In addition, even with complex regularization terms, NMF methods admit identifiability only when some conditions are satisfied by $\boldsymbol{C}$ and/or $\boldsymbol{S}$—e.g., the separability and sufficiently scattered conditions [2]. These conditions are related to how "spread" is the *conic hull* of $\boldsymbol{C}$ (and/or $\boldsymbol{S}$) in the nonnegative orthant, which are nontrivial to satisfy or even to check.

B. LL1-Based HU and Challenges

The work in [3] connected the LL1 tensor model with the HU problem. To be specific, if the abundance maps are low-rank matrices, i.e., $\mathrm{rank}(\boldsymbol{S}_r) = L_r < \min\{I, J\}$, one can re-write the first term in (2) as follows:

$$\underline{\boldsymbol{Y}} = \sum_{r=1}^{R} \left(\boldsymbol{A}_r \boldsymbol{B}_r^\top\right) \circ \boldsymbol{C}(:,r), \tag{3}$$

where $\boldsymbol{A}_r \in \mathbb{R}^{I\times L_r}$, $\boldsymbol{B}_r \in \mathbb{R}^{J\times L_r}$, and $\boldsymbol{S}_r = \boldsymbol{A}_r\boldsymbol{B}_r^\top$—which is exactly the block-term decomposition with multilinear rank-$(L, L, 1)$ terms [4]. The LL1 model has the following property:

Theorem 1 *Assume that $\boldsymbol{A}_r$, $\boldsymbol{B}_r$, and $\boldsymbol{C}$ in* (3) *are drawn from any absolutely continuous distributions. Then, the* LL1 *decomposition of $\underline{\boldsymbol{Y}}$ is essentially unique almost surely, if $L_r = L$, $IJ \geq L^2R$ and*

$$\min\left(\left\lfloor \frac{I}{L} \right\rfloor, R\right) + \min\left(\left\lfloor \frac{J}{L} \right\rfloor, R\right) + \min(K, R) \geq 2R + 2.$$

In the above, "essential uniqueness" means that if $\underline{\boldsymbol{Y}} = \sum_{r=1}^{R}(\boldsymbol{A}_r^\star(\boldsymbol{B}_r^\star)^\top) \circ \boldsymbol{C}^\star(:,r)$, then, it must hold that $\boldsymbol{S}^\star = \boldsymbol{S}\boldsymbol{\Pi}\boldsymbol{\Lambda}$, $\boldsymbol{C}^\star = \boldsymbol{C}\boldsymbol{\Pi}\boldsymbol{\Lambda}^{-1}$, where $\boldsymbol{\Pi}$ and $\boldsymbol{\Lambda}$ denote a permutation matrix and a nonsingular scaling matrix, respectively, $\boldsymbol{S}^\star = [\mathrm{vec}(\boldsymbol{S}_1^\star), \ldots, \mathrm{vec}(\boldsymbol{S}_R^\star)]^\top$, $\boldsymbol{S}_r^\star = \boldsymbol{A}_r^\star(\boldsymbol{B}_r^\star)^\top$ and $\boldsymbol{S} = [\mathrm{vec}(\boldsymbol{S}_1), \ldots, \mathrm{vec}(\boldsymbol{S}_R)]^\top$, $\boldsymbol{S}_r = \boldsymbol{A}_r\boldsymbol{B}_r^\top$.

Simply speaking, Theorem 1 means that if the abundance maps ($\boldsymbol{S}_r$'s) have low rank and the endmembers are linearly independent, then their identifiability is guaranteed under the LL1 framework. In the context of HU, the abundance maps are often (approximately) low-rank matrices. Table I shows that about 90% energy is captured in the first 50 principal components of the Terrain data's abundance maps (with a size of 500×307) [6]. Some more numerical evidence can be found in [7]. The identifiability conditions in Theorem 1 are different from those geometric conditions used in NMF [2], and thus LL1-based HU is a valuable complement to existing NMF approaches. Notably, the conditions in Theorem 1 is checkable, which is also a sharp contrast to the NMF cases.

To utilize Theorem 1 together with the HU model in (2), the work in [3] proposed the following criterion:

$$\min_{\{\boldsymbol{A}_r, \boldsymbol{B}_r\}, \boldsymbol{C}} \left\| \underline{\boldsymbol{Y}} - \sum_{r=1}^{R} \left(\boldsymbol{A}_r\boldsymbol{B}_r^\top\right) \circ \boldsymbol{C}(:,r) \right\|_F^2 + \lambda g(\{\boldsymbol{A}_r, \boldsymbol{B}_r\})$$
$$\text{s.t. } \boldsymbol{A}_r \geq \mathbf{0},\ \boldsymbol{B}_r \geq \mathbf{0},\ \boldsymbol{C} \geq \mathbf{0}, \tag{4}$$

where $g(\{\boldsymbol{A}_r, \boldsymbol{B}_r\}) = \| \sum_{r=1}^{R} \boldsymbol{A}_r\boldsymbol{B}_r^\top - \mathbf{1}\mathbf{1}^\top \|_F^2$. Note that the nonnegativity constraints are added according to the physical meaning of the endmembers and the abundance maps. The second term in the criterion is an approximation to

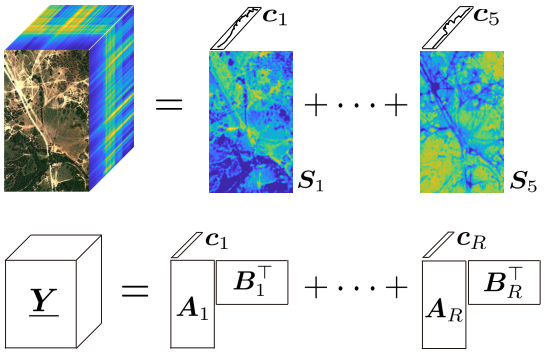


Fig. 1. Illustration of the LMM (top) and the LL1 tensor model (bottom).

TABLE I
THE ENERGY PROPORTION CONTAINED IN THE FIRST 50 PRINCIPAL COMPONENTS OF THE ABUNDANCE MAPS $\boldsymbol{S}_r$ (SIZE: 500×307) OF THE 5 PROMINENT MATERIALS IN THE TERRAIN DATA (SEE FIG. 1).

Ab. map ($\boldsymbol{S}_r$)	Soil1	Soil2	Tree	Shadow	Grass
Ratio of energy	93.56%	93.41%	89.48%	91.92%	94.60%

$\sum_{r=1}^{R} \boldsymbol{S}_r = \mathbf{1}\mathbf{1}^\top$ in (2). In real-data analysis, adding these physical meaning-motivated constraints is critical to fend against noise and modeling error, and often helps enhance interpretability of the HU results.

In [3], the ALS framework is used for handling (4). The three blocks $\boldsymbol{A} = [\boldsymbol{A}_1, \ldots, \boldsymbol{A}_R]$, $\boldsymbol{B} = [\boldsymbol{B}_1, \ldots, \boldsymbol{B}_R]$ and $\boldsymbol{C}$ are updated using matrix unfoldings of $\underline{\boldsymbol{Y}}$ using MU for accommodating the nonegativity constraints. Several notable challenges are in order: First, the ALS-MU algorithm in [3] takes $O(IJKLR+IKL^2R^2+JKL^2R^2)$ flops per iteration—which is fairly costly in the context of HU. The large number of flops is induced by the parametrization using $\boldsymbol{A} \in \mathbb{R}^{I\times LR}$, $\boldsymbol{B} \in \mathbb{R}^{J\times LR}$ and $\boldsymbol{C} \in \mathbb{R}^{K\times R}$ and the ALS framework. This parametrization essentially treats the LL1 decomposition problem as a *canonical polyadic decomposition* (CPD) problem with a tensor rank of LR, which is known to be hard when LR is large. Second, the employment of MU perhaps worsens the efficiency, since MU is known to be less effective for nonnegative factor analysis and often requires a large number of iterations to reach sensible results [8]. Third, the penalty based treatment for the sum-to-one condition in (4) does not necessarily produce a solution satisfying the LMM model in (2), and tuning λ is often nontrivial.

III. Constrained LL1 for HU

In this work, we employ the parametrization in (2) with $\mathrm{rank}(\boldsymbol{S}_r) \le L_r$ for LL1-based HU. Note that this model was shown to be equivalent to the three-block representation in (3) [9]. By the link between (1) and (2), we propose the following two-block parametrization-based constrained LL1 decomposition criterion:

$$\min_{\boldsymbol{S},\boldsymbol{C}} \ \frac{1}{2}\|\boldsymbol{Y} - \boldsymbol{C}\boldsymbol{S}\|_F^2 \tag{5a}$$

$$\text{s.t. } \mathrm{rank}(\mathrm{mat}(\boldsymbol{S}(r,:))) \le L_r, \ r = 1, \ldots, R, \tag{5b}$$

$$\boldsymbol{S} \ge \mathbf{0}, \ \mathbf{1}^\top\boldsymbol{S} = \mathbf{1}^\top, \ \boldsymbol{C} \ge \mathbf{0}. \tag{5c}$$

Our motivation for using this reformulation is as follows: As shown in [9], the low-rank $\boldsymbol{S}_r$ parameterization of the LL1 model can effectively avoid large-size subproblems in the ALS framework (in particular, the $\boldsymbol{A}$ and $\boldsymbol{B}$ blocks with sizes of $I \times LR$ and $J \times LR$, respectively, are circumvented), and thus could potentially substantially reduces the per-iteration complexity. The work in [9] did not consider nonnegativity and simplex constraints on the latent factors—with which the LL1 decomposition problem in (5) is a much more challenging optimization problem.

A. Proposed Approach: Alternating Gradient Projection

We propose to employ an alternating GP algorithm. To begin with, in iteration t, we update $\boldsymbol{C}$ using gradient projection:

$$\boldsymbol{C}^{(t+1)} \leftarrow \max\left\{\boldsymbol{C}^{(t)} - \alpha^{(t)}\boldsymbol{G}_{\boldsymbol{C}}^{(t)}, \ \mathbf{0}\right\}, \tag{6}$$

where $\max\{\cdot, \ \mathbf{0}\}$ is the orthogonal projector onto the nonnegativity orthant and $\alpha^{(t)}$ is a pre-defined step size used at iteration t, and the gradient can be computed using $\boldsymbol{G}_{\boldsymbol{C}}^{(t)} = \boldsymbol{C}^{(t)}\boldsymbol{S}^{(t)}(\boldsymbol{S}^{(t)})^\top - \boldsymbol{Y}(\boldsymbol{S}^{(t)})^\top$. In this work, we set $\alpha^{(t)} \le \frac{1}{\sigma_{\max}^2(\boldsymbol{S}^{(t)})}$ that ensures the cost to be decreased in each iteration.

For the $\boldsymbol{S}$-subproblem, we hope to use the same GP-based rule, i.e.,

$$\boldsymbol{S}^{(t+1)} \leftarrow \mathsf{Proj}_{\mathcal{S}}\left(\boldsymbol{S}^{(t)} - \beta^{(t)}\boldsymbol{G}_{\boldsymbol{S}}^{(t)}\right), \tag{7}$$

where $\beta^{(t)} \le \frac{1}{\sigma_{\max}^2(\boldsymbol{C}^{(t+1)})}$, $\boldsymbol{G}_{\boldsymbol{S}}^{(t)} = (\boldsymbol{C}^{(t+1)})^\top\boldsymbol{C}^{(t+1)}\boldsymbol{S}^{(t)} - (\boldsymbol{C}^{(t+1)})^\top\boldsymbol{Y}$, and the set $\mathcal{S} \subseteq \mathbb{R}^{R\times IJ}$ is defined as

$$\mathcal{S} = \{\boldsymbol{S} | \boldsymbol{S} \ge \mathbf{0}, \mathbf{1}^\top\boldsymbol{S} = \mathbf{1}^\top, \mathrm{rank}(\mathrm{mat}(\boldsymbol{S}(r,:))) \le L_r\}. \tag{8}$$

The above is conceptually simple. However, the critical challenge is that there is no known tractable algorithm that can solve the projection problem in (7). To address this issue, in the next subsection, we propose a simple heuristic for handling this projection problem.

B. Heuristic Simplex-Constrained Low-Rank Projector

To see our approach, let us define $\mathcal{S}_1 = \{\boldsymbol{S} \in \mathbb{R}^{R\times IJ} \mid \mathbf{1}^\top\boldsymbol{S} = \mathbf{1}^\top, \ \boldsymbol{S} \ge \mathbf{0}\}$ and $\mathcal{S}_2 = \{\boldsymbol{S} \in \mathbb{R}^{R\times IJ} \mid \mathrm{rank}(\mathrm{mat}(\boldsymbol{S}(r,:))) \le L_r, \ \forall r\}$. The goal then boils down to computing the projection on to $\mathcal{S} = \mathcal{S}_1 \cap \mathcal{S}_2$. We propose the following *alternating projection* (AP) algorithm for computing the projection:

$$\boldsymbol{F}^{(k+1)} \leftarrow \mathsf{Proj}_{\mathcal{S}_2}\left(\boldsymbol{W}^{(k)}\right), \tag{9a}$$

$$\boldsymbol{W}^{(k+1)} \leftarrow \mathsf{Proj}_{\mathcal{S}_1}(\boldsymbol{F}^{(k+1)}), \tag{9b}$$

where $\boldsymbol{W}^{(0)} = \boldsymbol{S}^{(t)} - \beta^{(t)}\boldsymbol{G}_{\boldsymbol{S}}^{(t)}$ and we have used k as the iteration index for the AP algorithm. Note that the above projections can be readily computed: Eq. (9a) can be solved optimally by truncated SVD, following the *Eckart–Young–Mirsky* theorem. Eq. (9b) can be solved efficiently by water-filling type algorithms [5].

The AP algorithm has a long history in convex feasibility problems. However, since the low-rank constraint is nonconvex, it is unclear if the AP algorithm can always converges to a "good" solution. Nonetheless, in our extensive experiments, we observe that AP algorithm converges quickly and works under various scenarios. In this work, we provide local convergence analysis to support our observation:

Proposition 1 *(Local Linear Convergence) Denote $\boldsymbol{W}^{(k)} = \widetilde{\boldsymbol{S}} + \boldsymbol{E}^{(k)}$ where $\widetilde{\boldsymbol{S}} \in \mathcal{S}_1 \cap \mathcal{S}_2$ is the sought projection of $\boldsymbol{W}^{(0)}$. Also assume that for any iterate $\boldsymbol{W}^{(k)} \in \mathcal{S}_1/\mathcal{S}_2$, there is a uniform upper bound $\rho < 1$ such that:*

$$\left(\frac{\sum_{r=1}^{R}\|\boldsymbol{E}_r^{(k)} - \boldsymbol{U}_2^r(\boldsymbol{U}_2^r)^\top\boldsymbol{E}_r^{(k)}\boldsymbol{V}_2^r(\boldsymbol{V}_2^r)^\top\|^2}{\sum_{r=1}^{R}\|\boldsymbol{E}_r^{(k)}\|^2}\right) \le \rho, \tag{10}$$

where $\boldsymbol{E}_r^{(k)} = \mathrm{mat}(\boldsymbol{E}^{(k)}(r,:))$, $\mathcal{R}(\boldsymbol{U}_2^r) = \mathcal{R}(\widetilde{\boldsymbol{S}}_r)^\perp$ and $\mathcal{R}(\boldsymbol{V}_2^r) = \mathcal{R}(\widetilde{\boldsymbol{S}}_r^\top)^\perp$ and $\boldsymbol{U}_2^r$ and $\boldsymbol{V}_2^r$ are semi-orthogonal bases. Then, if $\|\boldsymbol{E}^{(0)}\|$ is small enough, the algorithm converges linearly to $\widetilde{\boldsymbol{S}}$; i.e.,

$$\|\boldsymbol{E}^{(k+1)}\| \le \sqrt{\rho}\|\boldsymbol{E}^{(k)}\|. \tag{11}$$

The proposition asserts that the algorithm converges quickly to a feasible solution, if initialized properly. The condition (10) is reasonable. When $\boldsymbol{W} \notin \mathcal{S}_2$, it means that some $\mathrm{mat}(\boldsymbol{W}(r,:))$'s are not low-rank, and thus there is non-negligible energy in the "noise subspaces" spanned by $\boldsymbol{U}_2^r$ and $\boldsymbol{V}_2^r$.

Proof: A sketch of the proof is as follows. By [10, Theorem 1], we have $\boldsymbol{F}^{(k+1)} - \widetilde{\boldsymbol{S}} = \boldsymbol{M}^{(k)} + \boldsymbol{Q}(\boldsymbol{E}^{(k)})$, where $\boldsymbol{M}^{(k)}(r,:) = \mathrm{vec}(\boldsymbol{E}_r^{(k)} - \boldsymbol{U}_2^r(\boldsymbol{U}_2^r)^\top \boldsymbol{E}_r^{(k)} \boldsymbol{V}_2^r (\boldsymbol{V}_2^r)^\top)^\top$, if $\|\boldsymbol{E}^{(k)}\|$ is smaller than a certain threshold, where $\|\boldsymbol{Q}(\boldsymbol{Z})\| = O(\|\boldsymbol{Z}\|^2)$. Then, by the non-expansive property of convex sets:

$$\begin{aligned}\|\boldsymbol{E}^{(k+1)}\| &\le \|\boldsymbol{M}^{(k)}\| + O(\|\boldsymbol{E}^{(k)}\|^2) \\ &\le \rho^{1/2}\|\boldsymbol{E}^{(k)}\| + O(\|\boldsymbol{E}^{(k)}\|^2),\end{aligned}$$

where we have used (10). Hence, by [11, Lemma 10], we reach the conclusion in (11). ∎

Our overall algorithm consists of (6) and (7), where the $\boldsymbol{S}$ projection step in (7) leverages the subproblem solver in (9a)-(9b). This algorithm is referred to as the *gradient projection alternating projection algorithm* (GradPAPA). Note that as an alternating gradient projection algorithm, the standard Nesterov's extrapolation technique is also adopted in our implementation; see details in [12].

C. Complexity

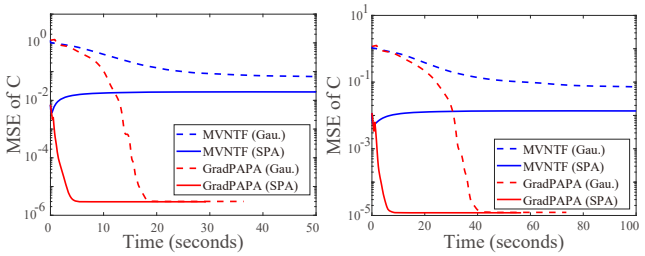
Computing the gradients for $\boldsymbol{C}$ and $\boldsymbol{S}$ both costs $O(IJKR)$ flops. The step sizes of $\alpha^{(t)}$ and $\beta^{(t)}$ take $O(R^3)$ flops, but R is normally small. In the AP solver, the SVD in (9a) takes $O(L^2 \min\{K, IJ\})$, and (9b) takes $O(IJR\log(R))$ flops by using water-filling type algorithms. In summary, the per-iteration complexity of the proposed algorithm is dominated by $O(IJKR + m(IJR\log(R) + L^2\min\{K, IJ\}) + R^3)$, where m is the number of AP iterations—which is normally only 3 to 6 (see Table III). Recall the ALS-MU algorithm [3] takes $O(IJKLR + IKL^2R^2 + JKL^2R^2)$ flops per iteration, which is much higher when $LR \approx I \approx J$ (which does often happen in HU). To summarize, our two-block parametrization helps effectively avoid operations that need $O(IJKLR)$ flops—which may easily dominate the computation time.

IV. Experiments

We compare our algorithm with the ALS-MU algorithms MVNTF [3] and MVNTFTV [13], where the latter has an additional spatial total variation regularization for performance enhancement. We terminate the algorithms when the relative error of the cost value is smaller than 10^{-5}. Besides, the AP is stopped when the relative change of the iterates is smaller than 10^{-3}. The *mean squared error* (MSE) of $\boldsymbol{C}$ and $\boldsymbol{S}$ is used as the performance metric; see definition in [5].

A. Synthetic Data

In this synthetic experiment, we generate $\boldsymbol{C}$ and $\boldsymbol{S}$ using the following procedure: 1) we draw the entries of $\boldsymbol{C}$ and $\boldsymbol{S}$ from the unit-variance zero-mean Gaussian distribution; 2) we employ the AP (9) on the Gaussian $\boldsymbol{S}$ to produce $\boldsymbol{S}$ that satisfies our problem structure; similarly, we use thresholding to obtain the nonnegative $\boldsymbol{C}$. We add i.i.d. zero-mean Gaussian noise to the synthetic tensors and make the signal-to-noise ratio (SNR) 25 dB. We set $I = J = K = 100$, $L = 30$, $R = 5$ or 10. We also test two initialization strategies, i.e., i.i.d. Gaussian initialization and successive projection algorithm (SPA)-based initialization; see [2] and references therein.



(a) $R = 5$ (b) $R = 10$

Fig. 2. The average MSEs with Gaussian and SPA initializations.

TABLE II
FEASIBILITY RATE OF RECOVERED $\boldsymbol{S}$.

Initialization		Gaussian Init.		SPA Init.	
Constraints	Methods	$R=5$	$R=10$	$R=5$	$R=10$
$\mathcal{S}_1$ (Simplex)	MVNTF ($q=10^{-2}$)	10.48%	12.36%	10.20%	11.77%
	MVNTF ($q=10^{-6}$)	0.0005%	0.001%	0.001%	0.003%
	GradPAPA ($q=10^{-6}$)	100%	100%	100%	100%
$\mathcal{S}_2$ (Low-Rank)	MVNTF	100%	100%	100%	100%
	GradPAPA	99.88%	99.90%	99.88%	99.90%

TABLE III
THE AVERAGE NUMBER OF AP ITERATIONS OF DIFFERENT R AND INITIALIZATION METHODS.

Initialization	Gaussian Init.		SPA Init.	
	$R=5$	$R=10$	$R=5$	$R=10$
Ave. AP iterations	5	6	3	4

Fig. 2 shows the averaged MSEs of $\boldsymbol{C}$ from 20 independent trials. It is observed that for different cases, the proposed GradPAPA algorithm largely outperforms ALS-based MVNTF in both accuracy and speed. The SPA initialization further improves the speed of GradPAPA by about 75%, which presents a promising combination of the simple greedy NMF algorithm and an LL1 based algorithm. Although MVNTF works to a certain extent, its MSE is more than three orders of magnitude higher compared to that of GradPAPA in all cases.

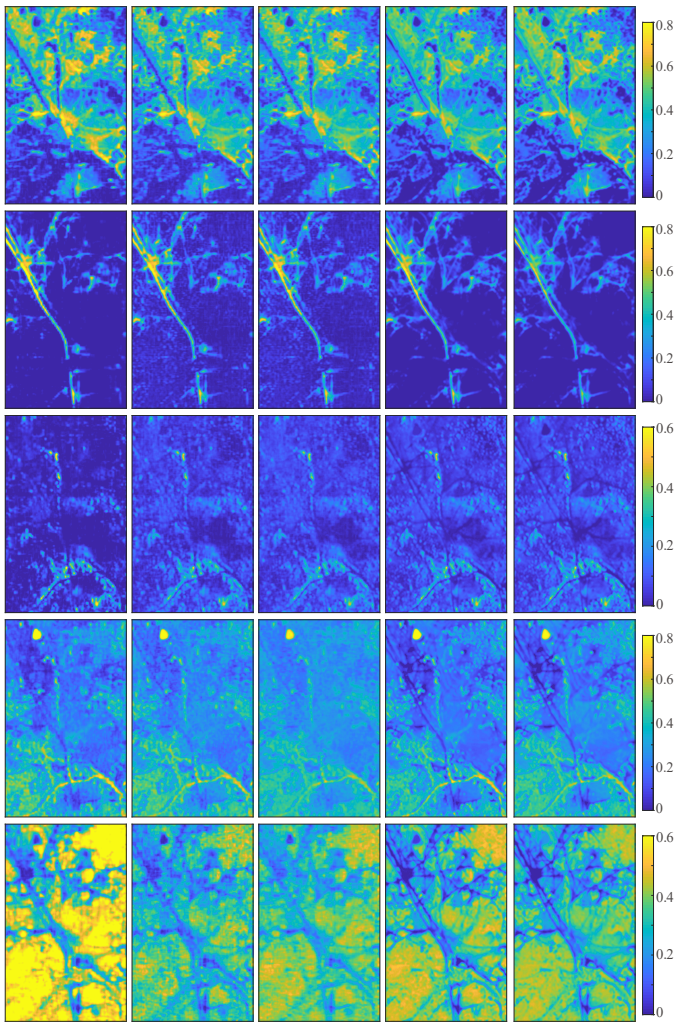
Table II shows the percentages of trials where the solutions obtained by the algorithms satisfy the constraints in the context HU. These constraints are important for interpreting the results. The low-rank constraint satisfaction is measured by averaging $(\sum_{i=1}^{L}\sigma_i / \sum_{i=1}^{\min\{I,J\}}\sigma_i) \times 100\%$, where σ_i is the ith singular value of the estimated $\boldsymbol{S}_r$ over all r. The $\mathcal{S}_1$ feasibility is measured by counting the percentage of the nonnegative columns of the estimated $\boldsymbol{S}$ satisfying $|\boldsymbol{1}^\top \boldsymbol{s}_\ell - 1| \le q$, where $q = 10^{-2}$ or 10^{-6}. One can see that MVNTF struggles to satisfy the probability simplex constraint, perhaps because it uses a "soft" penalty for this requirement [cf. Eq. (4)]. Nonetheless, GradPAPA almost achieves 100% feasibility. More importantly, such feasibility is enforced with relatively small cost: Table III shows that only about 5 AP iterations are needed for the $\boldsymbol{S}$ projection problem.

B. Semi-Real Data

In this experiment, we employ the semi-real Terrain data whose "space×space×spectrum" dimensions are $500 \times 307 \times$

TABLE IV
THE AVERAGE MSES OF $\boldsymbol{C}$ AND $\boldsymbol{S}$, AND RUNNING TIME (IN MINUTES) OF TERRAIN DATA BY DIFFERENT METHODS.

Methods	MSE of $\boldsymbol{C}$	MSE of $\boldsymbol{S}$	Time (min.)
SPA	0.0317 $\pm$ 0.0001	0.0715 $\pm$ 0.0023	—
MVNTF	0.0289 $\pm$ 0.0047	0.0452 $\pm$ 0.0114	81.1 $\pm$ 0.8
MVNTFTV	0.0288 $\pm$ 0.0033	0.0488 $\pm$ 0.0127	102.6 $\pm$ 1.0
GradPAPA	**0.0104** $\pm$ **0.0002**	**0.0047** $\pm$ **0.0001**	**5.5** $\pm$ **0.1**



(a) SPA (b) MVNTF (c) MVNTFTV (d) GradPAPA (e) Ground-truth

Fig. 3. The estimated abundance maps by different methods. From top to bottom: `Soil1`, `Soil2`, `Tree`, `Shadow`, and `Grass`.

166. The data contains five endmembers, namely, `Soil1`, `Soil2`, `Tree`, `Shadow`, and `Grass`; see details in [6]. We set $L = 100$ and employ the SPA initialization. We also consider i.i.d zero-mean Gaussian noise with SNR=45dB. Since MVNTF and MVNTFTV often run with extra lengthy time in such large-scale problems, we also set the maximum number of the iteration as 2,500 for all algorithms.

Table IV shows the MSE performance of the algorithms. Note that since the dataset is semi-real, ground-truth $\boldsymbol{C}$ and $\boldsymbol{S}$ are known and can be used for evaluation. One can see GradPAPA improves upon SPA by one order of magnitude in terms of MSE, while MVNTF and MVNTFTV could not offer such improvements. In terms of runtime, GradPAPA uses less than 6 minutes for this task, while the baselines both use more than 1 hour. The abundance maps produced by GradPAPA are also visually much closer to the ground-truth maps; see Fig. 3.

V. CONCLUSION

We proposed a constrained LL1 decomposition algorithm tailored for HU. Unlike existing algorithms that use three block parameterization of the LL1 tensor and ALS-MU type updates, our method employs a two-block parameterization and a GP algorithmic framework. As a consequence, the proposed algorithm effectively avoids heavy computations in its iterations. To realize the GP framework, we proposed an AP-based solver for a nonconvex orthogonal projection problem that is essential for enforcing HU-related low-rank and simplex constraints. Equipped with the AP algorithm, our GP framework exhibits largely improved HU performance (in terms of both accuracy and speed) on synthetic and semi-real datasets, compared to existing LL1 based HU algorithms.

REFERENCES

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

[2] 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.

[3] Y. Qian, F. Xiong, S. Zeng, J. Zhou, and Y. Y. Tang, "Matrix-vector nonnegative tensor factorization for blind unmixing of hyperspectral imagery," *IEEE Trans. Geosci. Remote Sens.*, vol. 55, no. 3, pp. 1776–1792, 2017.

[4] L. De Lathauwer, "Decompositions of a higher-order tensor in block terms—Part II: Definitions and uniqueness," *SIAM J. Matrix Anal. Appl.*, vol. 30, no. 3, pp. 1033–1066, 2008.

[5] 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.

[6] L. Zhuang, C. Lin, M. A. T. Figueiredo, and J. M. Bioucas-Dias, "Regularization parameter selection in minimum volume hyperspectral unmixing," *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 12, pp. 9858–9877, 2019.

[7] M. Ding, X. Fu, T.-Z. Huang, J. Wang, and X.-L. Zhao, "Hyperspectral super-resolution via interpretable block-term tensor modeling," *IEEE J. Sel. Topics Signal Process.*, vol. 15, no. 3, pp. 641–656, 2021.

[8] K. Huang and N. D. Sidiropoulos, "Putting nonnegative matrix factorization to the test: a tutorial derivation of pertinent Cramer–Rao bounds and performance benchmarking," *IEEE Signal Process. Mag.*, vol. 31, no. 3, pp. 76–86, 2014.

[9] X. Fu and K. Huang, "Block-term tensor decomposition via constrained matrix factorization," in *MLSP*, 2019, pp. 1–6.

[10] T. Vu and R. Raich, "Accelerating iterative hard thresholding for low-rank matrix completion via adaptive restart," in *Proc. IEEE ICASSP*, 2019, pp. 2917–2921.

[11] B. T. Polyak, "Some methods of speeding up the convergence of iteration methods," *USSR Computational Mathematics and Mathematical Physics*, vol. 4, no. 5, pp. 1–17, 1964.

[12] Y. Xu and W. Yin, "Block stochastic gradient iteration for convex and nonconvex optimization," *SIAM J. Optimiz.*, vol. 25, no. 3, pp. 1686–1716, 2015.

[13] F. Xiong, Y. Qian, J. Zhou, and Y. Y. Tang, "Hyperspectral unmixing via total variation regularized nonnegative tensor factorization," *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 4, pp. 2341–2357, 2019.