Computational Optimization and Applications (2020) 75:701–737
https://doi.org/10.1007/s10589-019-00128-3

An inexact augmented Lagrangian method for computing strongly orthogonal decompositions of tensors

Shenglong Hu[1]

Received: 26 August 2018 / Published online: 31 August 2019

Abstract
A strongly orthogonal decomposition of a tensor is a rank one tensor decomposition with the two component vectors in each mode of any two rank one tensors are either colinear or orthogonal. A strongly orthogonal decomposition with few number of rank one tensors is favorable in applications, which can be represented by a matrix-tensor multiplication with orthogonal factor matrices and a sparse tensor; and such a decomposition with the minimum number of rank one tensors is a strongly orthogonal rank decomposition. Any tensor has a strongly orthogonal rank decomposition. In this article, computing a strongly orthogonal rank decomposition is equivalently reformulated as solving an optimization problem. Different from the ill-posedness of the usual optimization reformulation for the tensor rank decomposition problem, the optimization reformulation of the strongly orthogonal rank decomposition of a tensor is well-posed. Each feasible solution of the optimization problem gives a strongly orthogonal decomposition of the tensor; and a global optimizer gives a strongly orthogonal rank decomposition, which is however difficult to compute. An inexact augmented Lagrangian method is proposed to solve the optimization problem. The augmented Lagrangian subproblem is solved by a proximal alternating minimization method, with the advantage that each subproblem has a closed formula solution and the factor matrices are kept orthogonal during the iteration. Thus, the algorithm always can return a feasible solution and thus a strongly orthogonal decomposition for any given tensor. Global convergence of this algorithm to a critical point is established without any further assumption. Extensive numerical experiments are conducted, and show that the proposed algorithm is quite promising in both efficiency and accuracy.

Keywords Strongly orthogonal decomposition of a tensor · Augmented Lagrangian method · Strongly orthogonal rank

Mathematics Subject Classification 15A69 · 90C26

✉ Shenglong Hu
shenglonghu@hdu.edu.cn

Extended author information available on the last page of the article

1 Introduction

Tensors (a.k.a. hypermatrices) are natural generalizations of vectors and matrices, serving as much more apparent and convenient tools to encode, analyze, as well as represent data and information in diverse disciplines of applied science. As one of the two sides of a coin, opposite to their capability over matrices, computations with tensors are subject to the *curse of dimensionality* [29,32]. A vector can be represented as an array indexed by one subscript, a matrix by two, whereas a higher order tensor by several. Usually, we refer to *order* as the number of subscripts needed to express a tensor, each subscript represents one mode, whereas the range of each subscript is the *dimension* in that mode. As is well-known, almost all computations of tensors have complexity *exponential* with respect to their orders and dimensions [29]. This brings a heavy obstacle to warrant tensors in many real applications. This tough situation dramatically improves if the tensor has a rank one decomposition with few rank one components, since in which case the computational complexity becomes *linear* with respect to either the order, the dimensions, or the rank of this tensor. Therefore, there is a natural invitation to the study of tensor rank one decompositions. As a result, the focus of research on tensors has been putting on both theoretical and computational properties of tensor rank one decompositions. This article falls into this main stream as well, and will be devoted to a particular rank one decomposition of tensors—the *strongly orthogonal (rank) decomposition*.

The conception of strongly orthogonal rank and the corresponding decomposition were proposed in the thesis of Franc in 1992 [17]. It is a natural generalization of SVD for matrices from a restricted perspective (cf. Sect. 2.4). In [33], "free orthogonal rank" and "free rank decomposition" were employed instead of "strong orthogonal rank" and "strong orthogonal rank decomposition" respectively by Leibovici and Sabatier. We will follow the terminologies of [17,27]. The number of rank one tensors in a strongly orthogonal decomposition is referred as the *length* of this decomposition.

In the following, we want outline three aspects that motivate this work. The first one is the wide range applications of strongly orthogonal (rank) decompositions of tensors, please refer to [3,12,13,27,29,30,33] and references therein. Particularly, in statistical modelings for learning latent variables [3] and blind source separations [12, 13], orthogonality is a reasonable requirement. Unlike the second order case in which a complete orthogonality (a.k.a. diagonalization) can be assumed (guaranteed by SVD of matrices, cf. Sect. 2.4), complete orthogonality for higher order tensors is impossible in general [27,32,41]. Therefore, strong orthogonality becomes an appropriate tradeoff between orthogonality and diagonality—it preserves the orthogonality and seeks the most sparsity on the representation. As a result, strongly orthogonal decompositions as well as general orthogonal decompositions have being adopted in diverse applications [6,10,13,14,24,28,29,36].

The second one is the appealing theoretical and computational benefits by considering strong orthogonality over the general rank one decompositions. The most general tensor rank one decomposition is decomposing a tensor with rank one components without further requirements on the structures of these rank one tensors. The smallest number of rank one tensors in such a decomposition is the *rank* of that tensor. It is well-known that determining the rank, computing a rank decomposition of a ten-

sor is very difficulty [32]. From the computational complexity perspective, they are respectively NP-complete and NP-hard problems [15,19]. Given the freedom on the structures of the decomposed rank one tensors, the tensor rank one decomposition and approximation problems suffer from many numerical difficulties as well, such as the component matrices of the iteration become unbounded, etc, please see [15,29] and references therein. However, whenever strong orthogonality is imposed on the decomposed rank one tensors, both the decomposition and the approximation become well-possed (cf. Sect. 2.3). Thus, from the computational point of view, the strongly orthogonal decomposition is a good candidate for tensor decomposition.

The strongly orthogonal (rank) decomposition has a geometric meaning as well. One widely admitted rule of science is *simplification*. A tensor (hypermatrix) is actually the coordinate representation of an element in a certain tensor space under a chosen coordinate system. A representation with simple coordinates should be the goal in many tasks. SVD for matrices and principal component analysis are both ruled by this principle, and it can represent a matrix with the simplest coordinates (i.e., a diagonal matrix). With the matrix-tensor multiplication, it is easy to see that a strongly orthogonal rank decomposition of a tensor also falls into this principle (cf. Proposition 2.1). Thus, geometrically, computing a strongly orthogonal rank decomposition is equivalent to finding a coordinate system such that the representation of this tensor is the simplest in the sense of having the fewest nonzero coordinates. This concise mathematical interpretation will provide insights on investigations of strongly orthogonal rank decompositions.

The last but not the least is a direct motivation for this article—computing out a strongly orthogonal (rank) decomposition for a given tensor. There lacks a systematic study on this issue in the literature, since the works [17,27,33]. To compute a strongly orthogonal decomposition for a given tensor, in [27], a greedy tensor decomposition is proposed. In each step of this algorithm, a best rank one approximation problem subject to orthogonality constraints must be solved. As observed already by Kolda [27], solving such a sequence of optimization problems is a *very challenging task*. Moreover, it is pointed out that *the difficulty with this approach is in enforcing the constraints* [27, Page 252]. A purpose of this article is providing a numerical method enforcing the orthogonality constraints during the iteration and computing a strongly orthogonal decomposition with the rank one tensors all at once.

At present, there has no computational complexity of computing a strongly orthogonal rank decomposition for a given tensor [27–29]. However, it is suspected to be NP-hard, in viewing of the NP-hardness of some related problems [11,20,34]. Thus, it would be difficult and impossible (if this problem is indeed NP-hard and P$\neq$NP) to compute out a strongly orthogonal rank decomposition of a general tensor in polynomial time. Therefore, the primary purpose of this article is *presenting a heuristic approach for computing a strongly orthogonal decomposition of a given tensor with length as short as possible and a numerical method to realize it*. Hopefully, this heuristic approach could give a strongly orthogonal rank decomposition in several cases as well.

The main contributions of this article are

(1) With l_0-norm and matrix-tensor multiplication, the strongly orthogonal rank decomposition is reformulated as an optimization problem over the matrix mani-

folds (cf. (9)) the first time (Proposition 2.1). This should serve as a standard tool and accumulate a step towards a systematic way for analyzing strongly orthogonal rank decompositions.

(2) A *heuristic* optimization problem (cf. (12)) is proposed to approximate the hard problem (9) which has a discontinuous objective function. An inexact augmented Lagrangian method is proposed to solve (12), i.e., Algorithm 3.1. With a careful design, all the subproblems have closed formula solutions, all the iterations fulfill the orthogonality constraints, and thus this algorithm always give a strongly orthogonal decomposition (Algorithm 4.1). Global convergence to a critical point of (12) is established without any further hypothesis (Proposition 3.3). Extensive numerical experiments show that this approach is quite promising (Sect. 5), and in several cases strongly orthogonal rank decompositions can be found.

The equivalent reformulation (9) actually reflects the difficulty in the sense that the task is minimizing a discontinuous (letting alone differentiable) function over a system of a large number of highly nonlinear equations and a manifold constraint. Note that even the optimization of a differentiable function over a smooth manifold is a hard computational problem in general [2]. Therefore, there is no guarantee that an algorithm can always return a global optimizer for (9). Actually, advances on manifold optimization are more focused on smooth functions for finding critical points, see [2,25] and references therein; a call for study on nonsmooth manifold optimization is given by Absil and Hosseini very recently [1]. This gives a motivation for the particularly designed inexact augmented Lagrangian method to problem (12).

With the discontinuous objective function in (9), a common strategy is replacing it with a heuristic continuous surrogate [1,16]. The l_1-norm is applied in this article (cf. (12)). It is a reasonable choice, which can be viewed as a penalty approach (cf. Sect. 2.5). Although (12) belongs to the class of optimization problems with continuous objective function over a compact feasible set, there is no theoretical guarantee for a numerical optimization method on finding their global optimizers for such problems [2,7], since the constraints are highly nonlinear and nonconvex. Usually, a wisely designed optimization method could converge to a critical point [2,7].

The l_1-norm is a nonsmooth function. The nonsmoothness of (12) and the particular structure that the variables of the manifold constraints are not related directly to the objective function (cf. (12)) motivate us to a particularly designed algorithm. A general thought is employing the penalty technique. The augmented Lagrangian method is a modification of the penalty method in a wise way to avoid the penalty parameter being forced to going infinite. Thus, the augmented Lagrangian method has more stable numerical performance over the penalty method [7]. The general picture in this article is applying the augmented Lagrangian method to (12) and keeping the orthogonality of the iterations with a careful design. On the other hand, the "inexact version" is studied as (i) subproblems cannot always be guaranteed to be solved exactly and (ii) the subproblem can be solved only inexactly within appropriate required precision to improve the overall efficiency.

The rest paper is organized as follows. For the convenience of reading, several technical details are put in Appendixes. Preliminaries on matrix-tensor multiplication (Sect. 2.1), the orthogonal group and related optimization properties (Sect. 2.2),

strongly orthogonal decompositions of a tensor (Sect. 2.3), some optimization techniques (Sect. 2.6) will be given in Sect. 2. In Sect. 2.5, the problem of computing a strongly orthogonal rank decomposition of a tensor is equivalently reformulated as a nonlinear optimization problem with orthogonality constraints and l_0-norm objective function. Replacing the l_0-norm by the l_1-norm surrogate, we get a heuristic of the problem, i.e., (12). In Sect. 3, an inexact augmented Lagrangian method (ALM) is proposed to solve this optimization problem. Critical points of the augmented Lagrangian function (Sect. 3.1), KKT points of the problem (12) (Sect. 3.2) are investigated. In order to study the KKT points, a nonsmooth version of the standard Lagrange multiplier theory is reviewed in Appendix B, since the l_1-norm is a nonsmooth function. Global convergence of the algorithm is established without any further hypothesis in Sect. 3.3, whose proof is put in Appendix C. The augmented Lagrangian subproblem is discussed in Sect. 4. Section 4.1 presents more details on implementing the ALM algorithm. The augmented Lagrangian subproblem is solved by a proximal alternating minimization (PAM) method, whose global convergence is also established without any further hypothesis in Sect. 4.2 and the proof is put in Appendix D. To this end, the convergence theory for a general PAM method is reviewed in Appendix A. Extensive numerical experiments are reported in Sect. 5. Sect. 5.1 is for concrete examples taken from literatures, Sect. 5.2 is for completely orthogonal decomposable tensors, in which a kind of *condition numbers* for tensors are discussed, Sect. 5.3 is for random examples, whereas Sect. 5.4 draws some conclusions for the numerical computations. Some final conclusions are given in Sect. 6.

2 Preliminaries

In this section, we will review some basic notions on tensors and preliminaries on strongly orthogonal (rank) decompositions of a tensor, as well as those of optimization theory which will be conducted in this article.

Given a positive integer $k \geq 2$, and positive integers $n_1, \ldots, n_k$, the tensor space consisting of real tensors of dimension $n_1 \times \cdots \times n_k$ is denoted as $\mathbb{R}^{n_1} \otimes \cdots \otimes \mathbb{R}^{n_k}$. In this Euclidean space, inner product and the induced norm can be defined. The Hilbert–Schmidt inner product of two given tensors $\mathcal{A}, \mathcal{B} \in \mathbb{R}^{n_1} \otimes \cdots \otimes \mathbb{R}^{n_k}$ is defined as

$$\langle \mathcal{A}, \mathcal{B} \rangle := \sum_{i_1=1}^{n_1} \cdots \sum_{i_k=1}^{n_k} a_{i_1 \ldots i_k} b_{i_1 \ldots i_k}.$$

The Hilbert–Schmidt norm $\|\mathcal{A}\|$ is then defined as

$$\|\mathcal{A}\| := \sqrt{\langle \mathcal{A}, \mathcal{A} \rangle}.$$

2.1 Matrix-tensor multiplication

Given a tensor $\mathcal{A} \in \mathbb{R}^{n_1} \otimes \cdots \otimes \mathbb{R}^{n_k}$ and k matrices $B^{(i)} \in \mathbb{R}^{n_i \times n_i}$ for $i \in \{1, \ldots, k\}$, the *matrix-tensor multiplication* $(B^{(1)}, \ldots, B^{(k)}) \cdot \mathcal{A}$ results in a tensor in $\mathbb{R}^{n_1} \otimes \cdots \otimes \mathbb{R}^{n_k}$, defined entry-wisely as

$$\left[(B^{(1)},\ldots,B^{(k)})\cdot\mathcal{A}\right]_{i_1\ldots i_k}:=\sum_{j_1=1}^{n_1}\cdots\sum_{j_k=1}^{n_k}b^{(1)}_{i_1j_1}\ldots b^{(k)}_{i_kj_k}a_{j_1\ldots j_k} \tag{1}$$

for all $i_t\in\{1,\ldots,n_t\}$ and $t\in\{1,\ldots,k\}$.

Let $n^*:=\prod_{i=1}^k n_i$. The *mode-j matrix flattening* of $\mathcal{A}$ is a matrix $A^{(j)}\in\mathbb{R}^{n_j\times\frac{n^*}{n_j}}$ defined entry-wisely as

$$a^{(j)}_{i_ji^*_j}=a_{i_1\ldots i_k}\text{ for all }i_j\in\{1,\ldots,n_j\}\text{ and }i^*_j\in\left\{1,\ldots,\frac{n^*}{n_j}\right\} \tag{2}$$

with

$$i^*_j=(i_1-1)\frac{n^*}{n_1n_j}+\cdots+(i_{j-1}-1)\frac{n^*}{n_1\cdots n_j}+(i_{j+1}-1)\frac{n^*}{n_1\cdots n_{j+1}}+\cdots+i_k.$$

We will denote by $\mathcal{A}^{(\mathrm{f},i)}=A^{(i)}$ the mode-i matrix flattening of $\mathcal{A}$. Let I be the identity matrix of appropriate size. It follows that

$$\left[(B^{(1)},\ldots,B^{(k)})\cdot\mathcal{A}\right]^{(\mathrm{f},i)}=B^{(i)}\left[(B^{(1)},\ldots,B^{(i-1)},I,B^{(i+1)},\ldots,B^{(k)})\cdot\mathcal{A}\right]^{(\mathrm{f},i)}$$

for all $i\in\{1,\ldots,k\}$.

It follows from the Hilbert–Schmidt inner product that

$$\langle\mathcal{A},\mathcal{B}\rangle=\langle\mathcal{A}^{(\mathrm{f},i)},\mathcal{B}^{(\mathrm{f},i)}\rangle$$

for any pair $\mathcal{A},\mathcal{B}\in\mathbb{R}^{n_1}\otimes\cdots\otimes\mathbb{R}^{n_k}$ and any $i\in\{1,\ldots,k\}$.

Let $\mathbb{O}(n)\subset\mathbb{R}^{n\times n}$ be the group of $n\times n$ orthogonal matrices. Whenever $B^{(i)}\in\mathbb{O}(n_i)$ for each $i\in\{1,\ldots,k\}$, it is a direct calculation to see that

$$\|(B^{(1)},\ldots,B^{(k)})\cdot\mathcal{A}\|=\|\mathcal{A}\|.$$

2.2 Orthogonal group and its normal cone

For any $A\in\mathbb{O}(n)$, the Fréchet normal cone to $\mathbb{O}(n)$ at A is defined as

$$\hat{N}_{\mathbb{O}(n)}(A):=\{B\in\mathbb{R}^{n\times n}\mid\langle B,C-A\rangle\le o(\|C-A\|)\text{ for all }C\in\mathbb{O}(n)\}.$$

Usually, we set $\hat{N}_{\mathbb{O}(n)}(A)=\emptyset$ whenever $A\notin\mathbb{O}(n)$. The (limiting) normal cone to $\mathbb{O}(n)$ at $A\in\mathbb{O}(n)$ is denoted by $N_{\mathbb{O}(n)}(A)$ and is defined as

$$B\in N_{\mathbb{O}(n)}(A)\iff\exists A_k\in\mathbb{O}(n),\ A_k\to A,\ \exists B_k\in\hat{N}_{\mathbb{O}(n)}(A_k),\ \text{such that }B_k\to B.$$

It is easily seen from the definition that the normal cone $N_{\mathbb{O}(n)}(A)$ is always closed. The indicator function $\delta_{\mathbb{O}(n)}$ of $\mathbb{O}(n)$ is defined as

$$\delta_{\mathbb{O}(n)}(X) := \begin{cases} 0 & \text{if } X \in \mathbb{O}(n), \\ +\infty & \text{otherwise.} \end{cases}$$

Given a function $f : \mathbb{R}^n \to \mathbb{R} \cup \{\infty\}$, the *regular subdifferential* of f at $\mathbf{x} \in \mathbb{R}^n$ is defined as

$$\hat{\partial} f(\mathbf{x}) := \left\{ \mathbf{h} \in \mathbb{R}^n : \liminf_{\mathbf{x} \neq \mathbf{y} \to \mathbf{x}} \frac{f(\mathbf{y}) - f(\mathbf{x}) - \langle \mathbf{h}, \mathbf{y} - \mathbf{x} \rangle}{\|\mathbf{y} - \mathbf{x}\|} \geq 0 \right\}$$

and the *(limiting) subdifferential* of f at $\mathbf{x}$ is defined as

$$\partial f(\mathbf{x}) := \left\{ \mathbf{h} \in \mathbb{R}^n : \exists \{\mathbf{x}^k\} \to \mathbf{x} \text{ and } \{\mathbf{h}^k\} \to \mathbf{h} \text{ satisfying } \mathbf{h}^k \in \hat{\partial} f(\mathbf{x}^k) \text{ for all } k \right\}.$$

We refer to [40] for more details on variational analysis concepts. An important fact about normal cone $N_{\mathbb{O}(n)}(A)$ and the subdifferential of the indicator function $\delta_{\mathbb{O}(n)}$ of $\mathbb{O}(n)$ at A is (cf. [40])

$$\partial \delta_{\mathbb{O}(n)} = N_{\mathbb{O}(n)}. \tag{3}$$

Note that the group $\mathbb{O}(n)$ of orthogonal matrices of size $n \times n$ is a smooth manifold of dimension $\frac{n(n-1)}{2}$. It follows from [40, Chapter 6.C] that

$$N_{\mathbb{O}(n)}(A) = \hat{N}_{\mathbb{O}(n)}(A) = \{AS \mid S \in \mathrm{S}^{n \times n}\},$$

where $\mathrm{S}^{n \times n} \subset \mathbb{R}^{n \times n}$ is the subspace of $n \times n$ symmetric matrices.

Given a matrix $B \in \mathbb{R}^{n \times n}$, the projection of B onto the normal cone of $\mathbb{O}(n)$ at A is

$$\pi_{N_{\mathbb{O}(n)}(A)}(B) = A \left(\frac{A^{\mathsf{T}} B + B^{\mathsf{T}} A}{2} \right).$$

Therefore,

$$\begin{aligned} B \in N_{\mathbb{O}(n)}(A) &\iff B - A \left(\frac{A^{\mathsf{T}} B + B^{\mathsf{T}} A}{2} \right) = O \\ &\iff B - AB^{\mathsf{T}} A = O \iff A^{\mathsf{T}} B - B^{\mathsf{T}} A = O, \end{aligned} \tag{4}$$

since

$$\left(I - \frac{1}{2} A A^{\mathsf{T}} \right) AB^{\mathsf{T}} A = \frac{1}{2} AB^{\mathsf{T}} A$$

and

$$I - \frac{1}{2} A A^{\mathsf{T}}$$

is an invertible matrix. The invertibility follows from the fact that $\mathbf{x}^{\mathsf{T}}(I - \frac{1}{2}AA^{\mathsf{T}})\mathbf{x} = \frac{1}{2}\|\mathbf{x}\|^2$.

2.3 Strongly orthogonal decomposition

Two rank one tensors

$$\mathcal{U} = \mathbf{u}^{(1)} \otimes \cdots \otimes \mathbf{u}^{(k)} \text{ and } \mathcal{V} = \mathbf{v}^{(1)} \otimes \cdots \otimes \mathbf{v}^{(k)}$$

with unit component vectors $\mathbf{u}^{(i)}$'s and $\mathbf{v}^{(i)}$'s are *strongly orthogonal*, denoted as $\mathcal{U} \perp_{\mathrm{s}} \mathcal{V}$, if $\mathcal{U} \perp \mathcal{V}$, i.e.,

$$\langle \mathcal{U}, \mathcal{V} \rangle = \prod_{i=1}^{k} \langle \mathbf{u}^{(i)}, \mathbf{v}^{(i)} \rangle = 0,$$

and

$$\mathbf{u}^{(i)} = \pm \mathbf{v}^{(i)} \text{ or } \mathbf{u}^{(i)} \perp \mathbf{v}^{(i)} \text{ for all } i = 1, \ldots, k.$$

Given a tensor $\mathcal{A} \in \mathbb{R}^{n_1} \otimes \cdots \otimes \mathbb{R}^{n_k}$, a *strongly orthogonal decomposition* means a rank one decomposition of $\mathcal{A}$

$$\mathcal{A} = \sum_{i=1}^{r} \lambda_i \mathbf{u}_i^{(1)} \otimes \cdots \otimes \mathbf{u}_i^{(k)} \tag{5}$$

such that the set of rank one tensors $\{\mathbf{u}_i^{(1)} \otimes \cdots \otimes \mathbf{u}_i^{(k)}\}_{i=1}^{r}$ is a set of mutually strongly orthogonal rank one tensors. For each given tensor $\mathcal{A} \in \mathbb{R}^{n_1} \otimes \cdots \otimes \mathbb{R}^{n_k}$, there is a natural strongly orthogonal decomposition as

$$\mathcal{A} = \sum_{i_1=1}^{n_1} \cdots \sum_{i_k=1}^{n_k} a_{i_1 \ldots i_k} \mathbf{e}_{i_1}^{(1)} \otimes \cdots \otimes \mathbf{e}_{i_k}^{(k)},$$

where $\{\mathbf{e}_1^{(s)}, \ldots, \mathbf{e}_{n_s}^{(s)}\}$ is the *standard basis* of $\mathbb{R}^{n_s}$ for all $s \in \{1, \ldots, k\}$. It is immediate to see that this can be done for any given *orthogonal basis* of $\mathbb{R}^{n_s}$ for all $s \in \{1, \ldots, k\}$.

A strongly orthogonal decomposition of $\mathcal{A}$ with the minimum r is called a *strongly orthogonal rank decomposition* of $\mathcal{A}$ and the corresponding r is the *strongly orthogonal rank* of $\mathcal{A}$, denoted as $\mathrm{rank}_{\mathrm{SO}}(\mathcal{A})$ [27]. It is called *free orthogonal rank* by Leibovici and Sabatier [33]. Some properties on strongly orthogonal rank of a tensor are investigated in [22], including an upper bound of the strongly orthogonal ranks and expected strongly orthogonal rank for a given tensor space.

2.4 An SVD perspective of strongly orthogonal rank decomposition

Given a matrix $A \in \mathbb{R}^{n_1 \times n_2}$, the classical singular value decomposition (SVD) of A reads that there exist orthogonal matrices $U \in \mathbb{O}(n_1)$, $V \in \mathbb{O}(n_2)$ and a diagonal matrix $\Lambda \in \mathbb{R}^{n_1 \times n_2}$ such that (cf. [21])

$$A = U\Lambda V^{\mathsf{T}} = (U, V) \cdot \Lambda. \tag{6}$$

The diagonality of the matrix Λ ensures that we can take a nonnegative diagonal of Λ, and define these diagonals as *singular values* of the matrix A. Arrange these singular values in nonincreasing order along the diagonals. Suppose without loss of generality that $n_1 \leq n_2$, and the columns of U and V are denoted as $\mathbf{u}_i$'s and $\mathbf{v}_j$'s. Then the SVD (6) can be expanded as

$$A = \sum_{i=1}^{n_1} \lambda_i \mathbf{u}_i \mathbf{v}_i^{\mathsf{T}} =: \sum_{i=1}^{n_1} \lambda_i \mathbf{u}_i \otimes \mathbf{v}_i = \sum_{i=1}^{r} \lambda_i \mathbf{u}_i \otimes \mathbf{v}_i, \tag{7}$$

where $r = \operatorname{rank}(A)$ is the *rank* of the matrix A. Obviously, (7) is a strongly orthogonal rank decomposition of A. The importance of SVD for matrices and its fundamental influence are widely known [18].

The beautiful nature of the matrix case makes many essential features of orthogonal decompositions simple and vague to picture their higher order counterparts, i.e., tensors (a.k.a. hypermatrices). There exist various attempts to generalize this fundamental singular value decomposition of a matrix to tensors. As the philosophy suggests, each generalization has its own advantages with the sacrifice of losing some attracting properties of their matrix counterpart [14].

Perhaps the most important fact that people unwilling to see is the missing of *diagonalization* for a tensor. It is a commonly admitted fact that for a given tensor $\mathcal{A} \in \mathbb{R}^{n_1} \otimes \cdots \otimes \mathbb{R}^{n_k}$ there cannot always exist a set of orthogonal matrices $U_i \in \mathbb{O}(n_i)$ for all $i \in \{1, \ldots, k\}$ and a diagonal tensor $\Lambda \in \mathbb{R}^{n_1} \otimes \cdots \otimes \mathbb{R}^{n_k}$ (i.e., the only possible nonzero entries of Λ are $\lambda_{i_1 \ldots i_k}$ with $i_1 = \cdots = i_k = i$ for $i \in \{1, \ldots, \min\{n_1, \ldots, n_k\}\}$) such that (cf. [32])

$$\mathcal{A} = (U_1, \ldots, U_k) \cdot \Lambda. \tag{8}$$

Therefore, searching a decomposition of the form (8) keeping the orthogonality of U_i's and allowing possible a non-diagonal tensor Λ should be the main task in decomposing a tensor and a reasonable alternative orthogonal decomposition scheme for a tensor. Compared with the other decompositions of a tensor, the decomposition (8) has the advantage of interpolation with orthogonal coordinates change, parallelling to the discussion on geometry of vector spaces in the matrix case by Jordan [26].

2.5 Optimization reformulation for the strongly orthogonal rank decomposition

Given a tensor $\mathcal{A} \in \mathbb{R}^{n_1} \otimes \cdots \otimes \mathbb{R}^{n_k}$, we consider the following optimization problem

$$\begin{aligned} \min\ & \|\mathcal{B}\|_0 \\ \text{s.t. } & (U^{(1)}, \ldots, U^{(k)}) \cdot \mathcal{A} = \mathcal{B}, \\ & U^{(i)} \in \mathbb{O}(n_i) \text{ for all } i = 1, \ldots, k, \end{aligned} \tag{9}$$

where $\|\mathcal{B}\|_0$ is the *zero norm* of $\mathcal{B}$, i.e., counting the number of nonzero entries of $\mathcal{B}$.

Proposition 2.1 (Optimization Reformulation) *For any given tensor* $\mathcal{A} \in \mathbb{R}^{n_1} \otimes \cdots \otimes \mathbb{R}^{n_k}$*, the minimization problem* (9) *has an optimizer* $(\mathcal{B}, U^{(1)}, \ldots, U^{(k)})$ *with optimal value being* $\mathrm{rank}_{\mathrm{SO}}(\mathcal{A})$*, and a strongly orthogonal rank decomposition of* $\mathcal{A}$ *is given by*

$$\mathcal{A} = \sum_{i_1=1}^{n_1} \cdots \sum_{i_k=1}^{n_k} b_{i_1 \ldots i_k} \mathbf{u}_{i_1}^{(1)} \otimes \cdots \otimes \mathbf{u}_{i_k}^{(k)}, \tag{10}$$

where $\mathbf{u}_i^{(s)}$ *is the* i*-th row of* $U^{(s)}$ *for all* $i \in \{1, \ldots, n_s\}$ *and* $s \in \{1, \ldots, k\}$.

Proof We first show that the optimization problem (9) always has an optimizer. Note that each $\mathbb{O}(n_i)$ is a compact set. By the matrix-tensor multiplication, we have that

$$\|\mathcal{B}\| = \|(U^{(1)}, \ldots, U^{(k)}) \cdot \mathcal{A}\| = \|\mathcal{A}\|.$$

Thus, the feasible set of (9) is compact. Since the zero norm is lower semi-continuous, we conclude that the minimum in (9) is always attainable by an optimizer $(\mathcal{B}, U^{(1)}, \ldots, U^{(k)})$.

In the following, we assume that $(\mathcal{B}, U^{(1)}, \ldots, U^{(k)})$ is such an optimizer. It follows from the matrix-tensor multiplication that

$$\begin{aligned}
\mathcal{A} &= \left((U^{(1)})^{\mathsf{T}} U^{(1)}, \ldots, (U^{(k)})^{\mathsf{T}} U^{(k)}\right) \cdot \mathcal{A} \\
&= \left((U^{(1)})^{\mathsf{T}}, \ldots, (U^{(k)})^{\mathsf{T}}\right) \cdot \left((U^{(1)}, \ldots, U^{(k)}) \cdot \mathcal{A}\right) \\
&= \left((U^{(1)})^{\mathsf{T}}, \ldots, (U^{(k)})^{\mathsf{T}}\right) \cdot \mathcal{B} \\
&= \sum_{i_1=1}^{n_1} \cdots \sum_{i_k=1}^{n_k} b_{i_1 \ldots i_k} \mathbf{u}_{i_1}^{(1)} \otimes \cdots \otimes \mathbf{u}_{i_k}^{(k)},
\end{aligned}$$

where $\mathbf{u}_i^{(s)}$ is the i-th row of $U^{(s)}$ for all $i \in \{1, \ldots, n_s\}$ and $s \in \{1, \ldots, k\}$. Thus, each optimizer $(\mathcal{B}, U^{(1)}, \ldots, U^{(k)})$ of (9) gives a strongly orthogonal decomposition of $\mathcal{A}$, and the number of strongly orthogonal rank one tensors in this decomposition is exactly the number of nonzero entries of $\mathcal{B}$, i.e., $\|\mathcal{B}\|_0$. Consequently, we must have $\mathrm{rank}_{\mathrm{SO}}(\mathcal{A}) \leq \|\mathcal{B}\|_0$ and hence the optimal value of (9) is lower bounded by $\mathrm{rank}_{\mathrm{SO}}(\mathcal{A})$.

In the following, we complete the proof by constructing a feasible solution of (9) from a strongly orthogonal rank decomposition of $\mathcal{A}$. Suppose that (5) gives such a decomposition, i.e.,

$$\mathcal{A} = \sum_{i=1}^{r} \lambda_i \mathbf{u}_i^{(1)} \otimes \cdots \otimes \mathbf{u}_i^{(k)}. \tag{11}$$

By the definition of strong orthogonality, we have that each pair of the vectors $\mathbf{u}_1^{(s)}, \ldots, \mathbf{u}_r^{(s)}$ consists of either orthogonal or equal or two opposite vectors, for all $s \in \{1, \ldots, k\}$. Thus, without loss of generality (changing λ_i to $-\lambda_i$ if necessary), we can assume that the set $\{\mathbf{u}_1^{(s)}, \ldots, \mathbf{u}_r^{(s)}\}$ forms an orthnormal set of vectors for all $s \in \{1, \ldots, k\}$. Let p_s be the cardinality of the set $\{\mathbf{u}_1^{(s)}, \ldots, \mathbf{u}_r^{(s)}\}$ for all $s \in \{1, \ldots, k\}$. Note that $p_s \leq \min\{r, n_s\}$ and strict inequality can happen for all $s \in \{1, \ldots, k\}$.

Let $U^{(s)} \in \mathbb{O}(n_s)$ with the first p_s columns being these of $\{\mathbf{u}_1^{(s)}, \ldots, \mathbf{u}_r^{(s)}\}$ for all $s \in \{1, \ldots, k\}$, and tensor $\mathcal{B} \in \mathbb{R}^{n_1} \otimes \cdots \otimes \mathbb{R}^{n_k}$ with entries

$$b_{i_1 \ldots i_k} := \begin{cases} \lambda_i & \text{if } \mathbf{u}_i^{(s)} \text{ is the } i_s\text{-th column of } U^{(s)} \text{ for all } s = 1, \ldots, k, \\ 0 & \text{otherwise.} \end{cases}$$

Immediately, we see that $\|\mathcal{B}\|_0 = r$ and

$$\mathcal{A} = (U^{(1)}, \ldots, U^{(k)}) \cdot \mathcal{B}.$$

Therefore, the above constructed $(\mathcal{B}, U^{(1)}, \ldots, U^{(k)})$ is a feasible solution to (9) with objective function value $\|\mathcal{B}\|_0 = r = \text{rank}_{\text{SO}}(\mathcal{A})$. As a result, the optimal value of (9) is upper bounded by $\text{rank}_{\text{SO}}(\mathcal{A})$.

Hence, an optimizer $(\mathcal{B}, U^{(1)}, \ldots, U^{(k)})$ of (9) exists with $\|\mathcal{B}\|_0 = \text{rank}_{\text{SO}}(\mathcal{A})$ and a strongly orthogonal rank decomposition of $\mathcal{A}$ is given as (10) with this optimizer. The proof is then complete. □

With Proposition 2.1, computing a strongly orthogonal rank decomposition of a given tensor $\mathcal{A}$ can be done, if one can solve the optimization problem (9). Moreover, optimizers of (9) will give such decompositions. Therefore, we will concentrate on solving the problem (9) in the rest article.

The constraint is nonlinear in $U^{(i)}$'s and linear in $\mathcal{B}$. However, the objective function is not even continuous. In order to resolve this, we would like to use a continuous sparsity measure $\|\mathcal{B}\|_1$ (i.e., the absolute sum of all the entries of $\mathcal{B}$) as a surrogate for $\|\mathcal{B}\|_0$.

$$\begin{aligned} \min \ & \|\mathcal{B}\|_1 \\ \text{s.t.} \ & (U^{(1)}, \ldots, U^{(k)}) \cdot \mathcal{A} = \mathcal{B}, \\ & U^{(i)} \in \mathbb{O}(n_i) \text{ for all } i = 1, \ldots, k. \end{aligned} \tag{12}$$

The heuristic of employing l_1-norm $\|\mathcal{B}\|_1$ for l_0-norm $\|\mathcal{B}\|_0$ is quite popular and useful in compressive sensing as well as in convex mathematical optimization models [16]. Theoretical justification for this convex relaxation has been established in the literature. For our problem (9), (12) is still nonconvex, and hence it is hard to give a theoretical guarantee on exactness of the relaxation. However, this heuristic is quite reasonable: If $\mathcal{B}$ is an optimizer of (9) such that $b_{i_1 \ldots i_k} = 0$ for $(i_1, \ldots, i_k) \in S$ with a subset $S \subseteq \{1, \ldots, n_1\} \times \cdots \times \{1, \ldots, n_k\}$, then the constraint $b_{i_1 \ldots i_k} = 0$ for $(i_1, \ldots, i_k) \in S$ can be added into (9). The objective function value is then constant and irrelevant to the optimization, and hence it can be removed. On the other hand, $|b_{i_1 \ldots i_k}|$ is a nonsmooth penalty for the constraint $b_{i_1 \ldots i_k} = 0$. By adding $\sum_{(i_1, \ldots, i_k) \in S} |b_{i_1 \ldots i_k}|$ into the objective function, we get a penalized problem with equal penalty weights. $\|\mathcal{B}\|_1 - \sum_{(i_1, \ldots, i_k) \in S} |b_{i_1 \ldots i_k}|$ can be added into the objective function as a regularizer for selecting an optimizer with the minimum absolute sum from the set of optimizers with $b_{i_1 \ldots i_k} = 0$ for $(i_1, \ldots, i_k) \in S$.

2.6 Optimization techniques

In this section, we give some basic optimization techniques that will be used in the sequel.

For a given $\alpha \in \mathbb{R}$, $\operatorname{sign}(\alpha)$ is the *sign* of α, defined as

$$\operatorname{sign}(\alpha) := \begin{cases} 1 & \text{if } \alpha > 0, \\ 0 & \text{if } \alpha = 0, \\ -1 & \text{if } \alpha < 0. \end{cases}$$

Given a vector $\tilde{\mathbf{x}} \in \mathbb{R}^n$ and parameter $\gamma > 0$, the optimizer of the following optimization problem

$$\min_{\mathbf{x}\in\mathbb{R}^n} \|\mathbf{x}\|_1 + \frac{\gamma}{2}\|\mathbf{x} - \tilde{\mathbf{x}}\|^2$$

is given by

$$\mathbf{x}^* = \mathrm{T}_{\frac{1}{\gamma}}(\tilde{\mathbf{x}}) := \left(\operatorname{sign}((\tilde{\mathbf{x}})_1)\max\left\{0, |(\tilde{\mathbf{x}})_1| - \frac{1}{\gamma}\right\}, \ldots, \operatorname{sign}((\tilde{\mathbf{x}})_n)\max\left\{0, |(\tilde{\mathbf{x}})_n| - \frac{1}{\gamma}\right\}\right)^{\mathsf{T}}.$$

It is known as the *soft-thresholding*, and T_α is the soft-thresholding operator for a given $\alpha > 0$.

Given a matrix $A \in \mathbb{R}^{n\times n}$, an optimizer of the following optimization problem

$$\min_{X\in\mathbb{O}(n)} \|X - A\|^2$$

is given by $X^* = UV^{\mathsf{T}}$ [18], where $U, V \in \mathbb{O}(n)$ are taken from the full singular value decomposition of A, i.e., $U\Sigma V^{\mathsf{T}} = A$ for some nonnegative diagonal matrix $\Sigma \in \mathbb{R}^{n\times n}$.

3 Augmented Lagrangian method

In this section, we apply the classical augmented Lagrangian method (ALM) for solving problem (12).

A standard reformulation of (12) by putting the orthogonality constraints into the objective function is

$$\begin{aligned} &\min \ \|\mathcal{B}\|_1 + \textstyle\sum_{i=1}^k \delta_{\mathbb{O}(n_i)}(U^{(i)}) \\ &\text{s.t. } (U^{(1)}, \ldots, U^{(k)}) \cdot \mathcal{A} = \mathcal{B}. \end{aligned} \tag{13}$$

With Lagrangian multiplier $\mathcal{X}$ and penalty parameter ρ, the augmented Lagrangian function of problem (13) is (cf. [7])

$$\begin{aligned} L_\rho(\mathbb{U}, \mathcal{B}; \mathcal{X}) = \|\mathcal{B}\|_1 + \sum_{i=1}^k \delta_{\mathbb{O}(n_i)}(U^{(i)}) + \langle \mathcal{X}, (U^{(1)}, \ldots, U^{(k)}) \cdot \mathcal{A} - \mathcal{B}\rangle \\ + \frac{\rho}{2}\|(U^{(1)}, \ldots, U^{(k)}) \cdot \mathcal{A} - \mathcal{B}\|^2. \end{aligned} \tag{14}$$

For given matrices $U_s^{(i)} \in \mathbb{O}(n_i)$ for all $i \in \{1, \ldots, k\}$ and $s = 1, 2, \ldots$, let

$$\mathbb{U}_s := (U_s^{(1)}, \ldots, U_s^{(k)}).$$

For convenience, we define

$$\|\mathbb{U}_s\| := \|U_s^{(1)}\| + \cdots + \|U_s^{(k)}\|.$$

Note that minimization problem (13) is an equality constrained nonlinear optimization problem with nonsmooth objective function. The classical *inexact augmented Lagrangian method* for solving (13) is

Algorithm 3.1 *inexact ALM*

The input is a tensor $\mathcal{A} \in \mathbb{R}^{n_1} \otimes \cdots \otimes \mathbb{R}^{n_k}$.
Given a sequence of positive numbers $\{\epsilon_s\}$ *such that* $\sum_{s=1}^{\infty} \epsilon_s < \infty$, *a penalty parameter* $\rho_1 > 0$, *a penalty adjustment parameter* $\gamma > 1$, *and a penalty adjustment threshold parameter* $\tau \in (0, 1)$.

Step 0: Initialization: choose initial guess $\mathcal{X}_1 \in \mathbb{R}^{n_1} \otimes \cdots \otimes \mathbb{R}^{n_k}$. *Set* $s := 1$.

Step 1: Solve the subproblem

$$(\mathbb{U}_s, \mathcal{B}_s) \approx \operatorname{argmin}_{\mathbb{U},\mathcal{B}} L_{\rho_s}(\mathbb{U}, \mathcal{B}; \mathcal{X}_s) \tag{15}$$

such that

$$U_s^{(i)} \in \mathbb{O}(n_i) \text{ for all } i \in \{1, \ldots, k\}, \tag{16}$$

and

$$\operatorname{dist}(0, \partial L_{\rho_s}(\mathbb{U}_s, \mathcal{B}_s; \mathcal{X}_s)) \le \epsilon_s. \tag{17}$$

Step 2: Update the multiplier as

$$\mathcal{X}_{s+1} = \mathcal{X}_s + \rho_s((U_s^{(1)}, \ldots, U_s^{(k)}) \cdot \mathcal{A} - \mathcal{B}_s). \tag{18}$$

Step 3: Update the penalty parameter as

$$\rho_{s+1} = \begin{cases} \rho_s & \text{if } \frac{\rho_{s-1}\|\mathcal{X}_{s+1} - \mathcal{X}_s\|}{\rho_s \|\mathcal{X}_s - \mathcal{X}_{s-1}\|} \le \tau, \\ \gamma\rho_s & \text{otherwise.} \end{cases} \tag{19}$$

Step 4: Unless a termination criterion is fulfilled, set $s := s + 1$ *and go to Step 1.*

Several problems should be addressed for Algorithm 3.1: (i) Step 1 is well-defined, i.e., there exists a solution for (15), (16) and (17); and (ii) efficient computation of such

a solution as $\epsilon_s \downarrow 0$. For a succinct representation, these algorithmic implementations will be discussed later. Convergence properties will be established first, assuming Algorithm 3.1 is well-defined.

To that end, critical points of the augmented Lagrangian function and KKT points of the original optimization problem (12) will be discussed.

3.1 Critical points

In the following, we study the critical points of the augmented Lagrangian. For a given multiplier $\mathcal{X}$, we can split $L_\rho(\mathbb{U}, \mathcal{B}; \mathcal{X})$ as

$$L_\rho(\mathbb{U}, \mathcal{B}; \mathcal{X}) := f(\mathbb{U}) + Q(\mathbb{U}, \mathcal{B}) + g(\mathcal{B}), \tag{20}$$

where

$$f(\mathbb{U}) := \sum_{i=1}^{k} \delta_{\mathbb{O}(n_i)}(U^{(i)}),\ g(\mathcal{B}) := \|\mathcal{B}\|_1,$$

and

$$Q(\mathbb{U}, \mathcal{B}) := \langle \mathcal{X}, (U^{(1)}, \ldots, U^{(k)}) \cdot \mathcal{A} - \mathcal{B} \rangle + \frac{\rho}{2} \|(U^{(1)}, \ldots, U^{(k)}) \cdot \mathcal{A} - \mathcal{B}\|^2$$

with

$$\mathbb{U} := (U^{(1)}, \ldots, U^{(k)}) \in \mathbb{R}^{n_1 \times n_1} \times \cdots \times \mathbb{R}^{n_k \times n_k}.$$

With the natural structure of the variables, we can partition the subdifferentials or gradients of the functions involved so far accordingly. The subdifferentials and gradients are kept aligned as the variables without vectorizing, e.g., the partial derivatives of $Q(\mathbb{U}, \mathcal{B})$ are collected in the same tensor format as $\mathcal{B}$ and the block matrix structure of $\mathbb{U}$. This notation can be easily understood from the linear operator perspective of subdifferentials and gradients. In particular,

$$\partial f(\mathbb{U}) = \left\{ \begin{bmatrix} B_1 \\ \vdots \\ B_k \end{bmatrix} \middle| B_i \in \partial \delta_{\mathbb{O}(n_i)}(U^{(i)}) \right\} = \left\{ \begin{bmatrix} B_1 \\ \vdots \\ B_k \end{bmatrix} \middle| B_i \in N_{\mathbb{O}(n_i)}(U^{(i)}) \right\},$$

and

$$\nabla_{\mathbb{U}} Q(\mathbb{U}, \mathcal{B}) = \rho \begin{bmatrix} U^{(1)} V^{(1)} \left[V^{(1)}\right]^{\mathsf{T}} - \mathcal{B}^{(\mathrm{f},1)} \left[V^{(1)}\right]^{\mathsf{T}} + \frac{1}{\rho} \mathcal{X}^{(\mathrm{f},1)} \left[V^{(1)}\right]^{\mathsf{T}} \\ \vdots \\ U^{(k)} V^{(k)} \left[V^{(k)}\right]^{\mathsf{T}} - \mathcal{B}^{(\mathrm{f},k)} \left[V^{(k)}\right]^{\mathsf{T}} + \frac{1}{\rho} \mathcal{X}^{(\mathrm{f},k)} \left[V^{(k)}\right]^{\mathsf{T}} \end{bmatrix}$$

where

$$V^{(i)} := \left[(U^{(1)}, \ldots, U^{(i-1)}, I, U^{(i+1)}, \ldots, U^{(k)}) \cdot \mathcal{A}\right]^{(\mathrm{f},i)} \text{ for all } i \in \{1, \ldots, k\}. \tag{21}$$

Likewise,

$$\nabla_{\mathcal{B}} Q(\mathbb{U}, \mathcal{B}) = \rho \left(\mathcal{B} - (U^{(1)}, \ldots, U^{(k)}) \cdot \mathcal{A} - \frac{1}{\rho} \mathcal{X} \right).$$

We also have that

$$\mathcal{W} \in \partial g(\mathcal{B}) \iff w_{i_1 \ldots i_k} \in \partial |b_{i_1 \ldots i_k}| = \begin{cases} 1 & \text{if } b_{i_1 \ldots i_k} > 0 \\ [-1, 1] & \text{if } b_{i_1 \ldots i_k} = 0 \\ -1 & \text{if } b_{i_1 \ldots i_k} < 0 \end{cases} \tag{22}$$

for all $i_j \in \{1, \ldots, n_j\}$ and $j \in \{1, \ldots, k\}$.

Given a lower semicontinuous function f, a critical point of f is a point $\mathbf{x}$ such that $0 \in \partial f(\mathbf{x})$.

Proposition 3.2 (Critical Points) *With notation as above, a point* $(\mathbb{U}, \mathcal{B}) \in (\mathbb{R}^{n_1 \times n_1} \times \cdots \times \mathbb{R}^{n_k \times n_k}) \times (\mathbb{R}^{n_1} \otimes \cdots \otimes \mathbb{R}^{n_k})$ *is a critical point of the minimization problem*

$$\min_{\mathbb{U}, \mathcal{B}} L_\rho(\mathbb{U}, \mathcal{B}; \mathcal{X})$$

if and only if

$$\left(U^{(i)}\right)^T \left(\mathcal{B}^{(\mathrm{f},i)} - \frac{1}{\rho} \mathcal{X}^{(\mathrm{f},i)}\right) \left[V^{(i)}\right]^T - V^{(i)} \left(\mathcal{B}^{(\mathrm{f},i)} - \frac{1}{\rho} \mathcal{X}^{(\mathrm{f},i)}\right)^T U^{(i)} = O \text{ for all } i \in \{1, \ldots, k\}$$

and

$$\rho((U^{(1)}, \ldots, U^{(k)}) \cdot \mathcal{A} - \mathcal{B}) + \mathcal{X} \in \partial g(\mathcal{B}).$$

Proof By the structure of the augmented Lagrangian (cf. (20)), we have that

$$\partial_{\mathbb{U}} L_\rho(\mathbb{U}, \mathcal{B}; \mathcal{X}) = \partial f(\mathbb{U}) + \nabla_{\mathbb{U}} Q(\mathbb{U}, \mathcal{B}) \text{ and } \partial_{\mathcal{B}} L_\rho(\mathbb{U}, \mathcal{B}; \mathcal{X}) = \nabla_{\mathcal{B}} Q(\mathbb{U}, \mathcal{B}) + \partial g(\mathcal{B}).$$

The rest then follows from (4). □

3.2 KKT points

In this section, we study KKT points of the optimization problem (12). A general theory on Lagrange multiplier rule for optimization problems with nonsmooth objective functions is presented in Appendix B, for more details, we refer to [40].

In the following, we first rewrite (12) in the form (43) as [1]

$$\begin{aligned} &\min \ \|\mathcal{B}\|_1 \\ &\text{s.t. } (U^{(1)}, \ldots, U^{(k)}) \cdot \mathcal{A} - \mathcal{B} = \mathcal{O}, \\ &\quad (U^{(1)}, \ldots, U^{(k)}, \mathcal{B}) \in \mathbb{O}(n_1) \times \cdots \times \mathbb{O}(n_k) \times \left(\mathbb{R}^{n_1} \otimes \cdots \otimes \mathbb{R}^{n_k}\right), \end{aligned} \tag{23}$$

where $\mathcal{O} \in \mathbb{R}^{n_1} \otimes \cdots \otimes \mathbb{R}^{n_k}$ is the zero tensor. Note that all requirements in Appendix B for the objective and constraint functions and abstract set X are satisfied. In the following, we will show that the constraint qualification (44) is satisfied at each feasible point of (23).

Note that in this case

$$X = \mathbb{O}(n_1) \times \cdots \times \mathbb{O}(n_k) \times \left(\mathbb{R}^{n_1} \otimes \cdots \otimes \mathbb{R}^{n_k}\right),$$

and

$$N_X(U^{(1)}, \ldots, U^{(k)}, \mathcal{B}) = N_{\mathbb{O}(n_1)}(U^{(1)}) \times \cdots \times N_{\mathbb{O}(n_k)}(U^{(k)}) \times \{\mathcal{O}\}.$$

Let matrices $V^{(i)} \in \mathbb{R}^{n_i \times \frac{n^*}{n_i}}$ be defined as (21) for all $i \in \{1, \ldots, k\}$. Let $\mathcal{X} \in \mathbb{R}^{n_1} \otimes \cdots \otimes \mathbb{R}^{n_k}$. With a direct calculation, the system (44) for problem (23) becomes

$$\nabla(\langle \mathcal{X}, (U^{(1)}, \ldots, U^{(k)}) \cdot \mathcal{A} - \mathcal{B} \rangle) = \begin{bmatrix} \mathcal{X}^{(\mathrm{f},1)}\left(V^{(1)}\right)^{\mathsf{T}} \\ \vdots \\ \mathcal{X}^{(\mathrm{f},k)}\left(V^{(k)}\right)^{\mathsf{T}} \\ -\mathcal{X} \end{bmatrix} \in N_X(U^{(1)}, \ldots, U^{(k)}, \mathcal{B}), \tag{24}$$

which implies directly $\mathcal{X} = \mathcal{O}$ from the last relation. Therefore, the constraint qualification (44) is satisfied at each feasible, and hence local minimum, solution of (23).

Next, we present the KKT system of (23) in an explicit form. With (24) and the fact that each $N_{\mathbb{O}(n_i)}(U^{(i)})$ is a linear space (cf. Sect. 2.2), it is easy to see that a feasible point $(\mathbb{U}, \mathcal{B}) = (U^{(1)}, \ldots, U^{(k)}, \mathcal{B})$ is a KKT point of (23) (as well as (12)) if and only if the following system is fulfilled

$$\mathcal{X} \in \partial\|\mathcal{B}\|_1, \text{ and } \mathcal{X}^{(\mathrm{f},i)}\left(V^{(i)}\right)^{\mathsf{T}} \in N_{\mathbb{O}(n_i)}(U^{(i)}) \text{ for all } i \in \{1, \ldots, k\}. \tag{25}$$

3.3 Global convergence

Suppose that $\{(\mathbb{U}_s, \mathcal{B}_s, \mathcal{X}_s)\}$ is a sequence generated by Algorithm 3.1. We will show first that the sequence $\{(\mathbb{U}_s, \mathcal{B}_s, \mathcal{X}_s)\}$ is bounded. Then, suppose that $(\mathbb{U}_*, \mathcal{B}_*, \mathcal{X}_*)$ is

[1] We can reformulate (23) as an optimization problem with a smooth objective function by packing $\|\mathcal{B}\|_1$ into the constraints as well. Then, optimality conditions can be derived as [39]. While, it seems that it is not a wise choice here to destroy the smooth nature of the constraints and introduce a heavy task on computing the normal cone of a feasible set whose constraints involve nonsmooth functions.

one of its limit points. We will next show that $(\mathbb{U}_*, \mathcal{B}_*, \mathcal{X}_*)$ is a KKT point of (12), i.e., the system (25) is satisfied.

Proposition 3.3 *Let* $\{(\mathbb{U}_s, \mathcal{B}_s, \mathcal{X}_s)\}$ *be the iteration sequence generated by Algorithm* 3.1. *Then it is a bounded sequence, every limit point* $(\mathbb{U}_*, \mathcal{B}_*, \mathcal{X}_*)$ *of this sequence satisfies the feasibility of problem* (12)*, i.e.,*

$$(U_*^{(1)}, \ldots, U_*^{(k)}) \cdot \mathcal{A} = \mathcal{B}_* \text{ and } U_*^{(i)} \in \mathbb{O}(n_i) \text{ for all } i \in \{1, \ldots, k\},$$

and $(\mathbb{U}_*, \mathcal{B}_*)$ *is a KKT point of problem* (12).

The proof of Proposition 3.3 is given in Appendix C. Proposition 3.3 gives the global convergence result for Algorithm 3.1. In the following Sect. 4, we will address the well-definiteness and computation issues.

4 Augmented Lagrangian subproblem

In this section, we apply a proximal alternating minimization (PAM) method to solve the augmented Lagrangian subproblem (15), (16) and (17). For the sake of notational simplicity, we will omit the outer iteration indices of Algorithm 3.1 and present the algorithm for the problem

$$(\mathbb{U}_*, \mathcal{B}_*) \approx \operatorname{argmin}_{\mathbb{U},\mathcal{B}} L_\rho(\mathbb{U}, \mathcal{B}; \mathcal{X})$$

for given multiplier $\mathcal{X}$ and penalty parameter ρ. The initial guess for this problem is denoted as $(\mathbb{U}_0, \mathcal{B}_0)$, which can be taken as the previous outer iteration of Algorithm 3.1.

4.1 Proximal alternating minimization algorithm

The algorithm is a regularized proximal multi-block nonlinear Gauss–Seidel method: starting from $s = 1$, iteratively solve the following problems

$$\begin{cases} \mathcal{B}_s = \operatorname{argmin}_{\mathcal{B}} L_\rho(\mathbb{U}_{s-1}, \mathcal{B}; \mathcal{X}) + \frac{c_s^{(0)}}{2}\|\mathcal{B} - \mathcal{B}_{s-1}\|^2, \\ \qquad\qquad \text{and for } j = 1, \ldots, k, \text{ solve} \\ U_s^{(j)} \in \operatorname{argmin}_{U^{(j)}} L_\rho\big((U_s^{(1)}, \ldots, U_s^{(j-1)}, U^{(j)}, U_{s-1}^{(j+1)}, \ldots, U_{s-1}^{(k)}), \mathcal{B}_s; \mathcal{X}\big) \\ \qquad\qquad\qquad + \frac{c_s^{(j)}}{2}\|U^{(j)} - U_{s-1}^{(j)}\|^2, \end{cases} \tag{26}$$

in which $c_s^{(j)} \geq 0$ for all $j \in \{0, \ldots, k\}$ and $s = 1, 2, \ldots$ are proximal parameters chosen by the user. There are $k+1$ subproblems in (26), whereas the last k subproblems are of the same structure. A good news is that all the subproblems have optimal solutions in closed formulae. In the sequel, we will derive these closed formulae.

The first subproblem is equivalent to

$$\begin{aligned}&\min_{\mathcal{B}\in\mathbb{R}^{n_1}\otimes\cdots\otimes\mathbb{R}^{n_k}} \|\mathcal{B}\|_1+\frac{\rho}{2}\|\mathcal{B}-(U_{s-1}^{(1)},\ldots,U_{s-1}^{(k)})\cdot\mathcal{A}-\frac{1}{\rho}\mathcal{X}\|^2+\frac{c_s^{(0)}}{2}\|\mathcal{B}-\mathcal{B}_{s-1}\|^2\\ &\simeq \min_{\mathcal{B}\in\mathbb{R}^{n_1}\otimes\cdots\otimes\mathbb{R}^{n_k}} \|\mathcal{B}\|_1+\frac{\rho+c_s^{(0)}}{2}\left\|\mathcal{B}-\frac{\rho(U_{s-1}^{(1)},\ldots,U_{s-1}^{(k)})\cdot\mathcal{A}+\mathcal{X}+c_s^{(0)}\mathcal{B}_{s-1}}{\rho+c_s^{(0)}}\right\|^2\end{aligned}$$

whose solution is analytic and obtained by the soft-thresholding (cf. Sect. 2.6)

$$\mathcal{B}_s=\mathrm{T}_{\frac{1}{\rho+c_s^{(0)}}}\left(\frac{\rho(U_{s-1}^{(1)},\ldots,U_{s-1}^{(k)})\cdot\mathcal{A}+\mathcal{X}+c_s^{(0)}\mathcal{B}_{s-1}}{\rho+c_s^{(0)}}\right).$$

In the next, we derive optimal solutions for the rest subproblems. Let $j\in\{1,\ldots,k\}$, and

$$V_s^{(j)}:=\left[(U_s^{(1)},\ldots,U_s^{(j-1)},I,U_{s-1}^{(j+1)},\ldots,U_{s-1}^{(k)})\cdot\mathcal{A}\right]^{(\mathrm{f},j)}.$$

Then the subproblem for computing $U_s^{(j)}$ is

$$\begin{aligned}&\min_{U^{(j)}\in\mathbb{O}(n_j)}\langle\mathcal{X},(U_s^{(1)},\ldots,U_s^{(j-1)},U^{(j)},U_{s-1}^{(j+1)},\ldots,U_{s-1}^{(k)})\cdot\mathcal{A}\rangle\\ &+\frac{\rho}{2}\|(U_s^{(1)},\ldots,U_s^{(j-1)},U^{(j)},U_{s-1}^{(j+1)},\ldots,U_{s-1}^{(k)})\cdot\mathcal{A}-\mathcal{B}_s\|^2+\frac{c_s^{(j)}}{2}\|U^{(j)}-U_{s-1}^{(j)}\|^2.\end{aligned}\tag{27}$$

Note that

$$\begin{aligned}&\langle\mathcal{X},(U_s^{(1)},\ldots,U_s^{(j-1)},U^{(j)},U_{s-1}^{(j+1)},\ldots,U_{s-1}^{(k)})\cdot\mathcal{A}\rangle\\ &=\langle\mathcal{X}^{(\mathrm{f},j)},U^{(j)}V_s^{(j)}\rangle=\langle\mathcal{X}^{(\mathrm{f},j)}(V_s^{(j)})^{\mathsf{T}},U^{(j)}\rangle.\end{aligned}$$

Likewise,

$$\langle\mathcal{B}_s,(U_s^{(1)},\ldots,U_s^{(j-1)},U^{(j)},U_{s-1}^{(j+1)},\ldots,U_{s-1}^{(k)})\cdot\mathcal{A}\rangle=\langle\mathcal{B}_s^{(\mathrm{f},j)}(V_s^{(j)})^{\mathsf{T}},U^{(j)}\rangle.$$

With these facts and $U^{(j)}\in\mathbb{O}(n_j)$, the subproblem (27) is equivalent to

$$\min_{U^{(j)}\in\mathbb{O}(n_j)}\|U^{(j)}-c_s^{(j)}U_{s-1}^{(j)}+\left(\mathcal{X}^{(\mathrm{f},j)}-\rho\mathcal{B}_s^{(\mathrm{f},j)}\right)(V_s^{(j)})^{\mathsf{T}}\|^2,$$

which in turn can be solved by singular value decomposition (SVD) or polar decomposition [18] (cf. Sect. 2.6).

We are now in the position to present the proximal alternating minimization algorithm for the augmented Lagrangian subproblem.

Algorithm 4.1 *PAM*

The inputs are tensors $\mathcal{A}, \mathcal{X} \in \mathbb{R}^{n_1} \otimes \cdots \otimes \mathbb{R}^{n_k}$, *parameters* $\rho > 0, \epsilon > 0, 0 < \underline{c} < \overline{c}$.

Step 0: Initialization: choose initial guess $\mathcal{B}_0 \in \mathbb{R}^{n_1} \otimes \cdots \otimes \mathbb{R}^{n_k}$, *and* $\mathbb{U}_0 := (U_0^{(1)}, \ldots, U_0^{(k)}) \in \mathbb{O}(n_1) \times \cdots \times \mathbb{O}(n_k)$. *Set* $s := 1$.

Step 1: Choose $c_s^{(0)} \in [\underline{c}, \overline{c}]$, *and compute*

$$\mathcal{B}_s = \mathrm{T}_{\frac{1}{\rho + c_s^{(0)}}} \left(\frac{\rho(U_{s-1}^{(1)}, \ldots, U_{s-1}^{(k)}) \cdot \mathcal{A} + \mathcal{X} + c_s^{(0)} \mathcal{B}_{s-1}}{\rho + c_s^{(0)}} \right). \tag{28}$$

Step 2: For $j = 1, \ldots, k$, *choose* $c_s^{(j)} \in [\underline{c}, \overline{c}]$ *and compute the full singular value decomposition*

$$U \Sigma V^{\mathsf{T}} = c_s^{(j)} U_{s-1}^{(j)} - \big(\mathcal{X}^{(\mathrm{f},j)} - \rho \mathcal{B}_s^{(\mathrm{f},j)}\big)(V_s^{(j)})^{\mathsf{T}}, \tag{29}$$

and let $U_s^{(j)} = U V^{\mathsf{T}}$.

Step 3: Unless $\|\Theta_s\| \le \epsilon$ *(see* (30) *for* Θ_s*), set* $s := s + 1$ *and go to Step 1.*

4.2 Convergence analysis

In this section, we will establish the global convergence of Algorithm 4.1. To that end, optimality conditions of problem (26) will be derived first.

A direct calculation shows that the optimality conditions for the subproblem (26) are the following system:

$$\begin{cases} \rho\Big(\mathcal{B}_s - (U_{s-1}^{(1)}, \ldots, U_{s-1}^{(k)}) \cdot \mathcal{A} - \frac{1}{\rho}\mathcal{X}\Big) + c_s^{(0)}(\mathcal{B}_s - \mathcal{B}_{s-1}) \in -\partial \|\mathcal{B}_s\|_1, \\ \mathcal{X}^{(\mathrm{f},j)}(V_s^{(j)})^{\mathsf{T}} + \rho\big(U_s^{(j)} V_s^{(j)} - \mathcal{B}_s^{(\mathrm{f},j)}\big)(V_s^{(j)})^{\mathsf{T}} + c_s^{(j)}(U_s^{(j)} - U_{s-1}^{(j)}) \in -\partial_{\mathbb{O}(n_j)}(U_s^{(j)}) \\ \qquad \text{for all } j = 1, \ldots, k. \end{cases}$$

With the fact that the normal cones of the orthogonal groups are linear subspaces, the above system can be simplified as

$$\begin{cases} \rho\Big(\mathcal{B}_s - (U_{s-1}^{(1)}, \ldots, U_{s-1}^{(k)}) \cdot \mathcal{A} - \frac{1}{\rho}\mathcal{X}\Big) + c_s^{(0)}(\mathcal{B}_s - \mathcal{B}_{s-1}) \in -\partial \|\mathcal{B}_s\|_1, \\ \mathcal{X}^{(\mathrm{f},j)}(V_s^{(j)})^{\mathsf{T}} - \rho \mathcal{B}_s^{(\mathrm{f},j)}(V_s^{(j)})^{\mathsf{T}} + c_s^{(j)}(U_s^{(j)} - U_{s-1}^{(j)}) \in -\partial_{\mathbb{O}(n_j)}(U_s^{(j)}) \text{ for all } j = 1, \ldots, k. \end{cases}$$

Let

$$\Theta_s := \begin{bmatrix} \mathcal{X}^{(\mathrm{f},1)}(V_s^{(1)} - \tilde{V}_s^{(1)})^{\mathsf{T}} - \rho \mathcal{B}_s^{(\mathrm{f},1)}(V_s^{(1)} - \tilde{V}_s^{(1)})^{\mathsf{T}} + c_s^{(1)}(U_s^{(1)} - U_{s-1}^{(1)}) \\ \vdots \\ \mathcal{X}^{(\mathrm{f},k)}(V_s^{(k)} - \tilde{V}_s^{(k)})^{\mathsf{T}} - \rho \mathcal{B}_s^{(\mathrm{f},k)}(V_s^{(k)} - \tilde{V}_s^{(k)})^{\mathsf{T}} + c_s^{(k)}(U_s^{(k)} - U_{s-1}^{(k)}) \\ \rho\Big((U_{s-1}^{(1)}, \ldots, U_{s-1}^{(k)}) \cdot \mathcal{A} - (U_s^{(1)}, \ldots, U_s^{(k)}) \cdot \mathcal{A}\Big) - c_s^{(0)}(\mathcal{B}_s - \mathcal{B}_{s-1}) \end{bmatrix}, \tag{30}$$

where for $j \in \{1, \ldots, k\}$

$$\tilde{V}_s^{(j)} := \left[(U_s^{(1)}, \ldots, U_s^{(j-1)}, I, U_s^{(j+1)}, \ldots, U_s^{(k)}) \cdot \mathcal{A}\right]^{(\mathrm{f},j)}.$$

Compared with the critical points of the augmented Lagrangian function (cf. Sect. 3.1), we have that

$$\Theta_s \in \partial L_\rho(\mathbb{U}_s, \mathcal{B}_s, \mathcal{X}). \tag{31}$$

We will show that Algorithm 4.1 converges. It is based on a general result established in [4, Theorem 6.2]. We summarized the convergence theory for this general PAM in Appendix A.

Proposition 4.2 *For any given tensors $\mathcal{A}, \mathcal{X} \in \mathbb{R}^{n_1} \otimes \cdots \otimes \mathbb{R}^{n_k}$, and parameters $\rho > 0$, $\epsilon > 0$, $0 < \underline{c} < \overline{c}$, we have that the sequence $\{(\mathbb{U}_s, \mathcal{B}_s)\}$ produced by Algorithm 4.1 converges and*

$$\|\Theta_s\| \to 0 \text{ as } s \to \infty$$

for the sequence $\{\Theta_s\}$ generated by (28), (29) and (30). Then Algorithm 4.1 is finite termination.

The proof of Proposition 4.2 is given in Appendix D.

5 Numerical experiments

In this section, we test Algorithm 3.1 for several classes of tensors. All the tests were conducted on a Dell PC with RAM 4GB and 3.2GHz CPU in 64bt Windows operation system. All codes were written in MatLab with Tensor ToolBox by Bader and Kolda [5]. Some default parameters were chosen as $\gamma = 1.05$, $\tau = 0.8$, and

$$\epsilon_s = \max\{10^{-10} * \max\{n_i \mid 1 \le i \le k\}, \min\{0.8^s, 0.8 * \hat{\epsilon}_{s-1}\}\} \text{ with } \hat{\epsilon}_0 = 0.8,$$

with $\hat{\epsilon}_{s-1} := \|\Theta_{s-1}\| \le \epsilon_{s-1}$ for $s \ge 2$. The initial guess $\mathcal{X}_1 = \mathcal{O}$, and $\mathcal{B}_1 = ((U_1^{(1)})^\mathsf{T}, \ldots, (U_1^{(k)})^\mathsf{T}) \cdot \mathcal{A}$ with each $U_1^{(i)} \in \mathbb{O}(n_i)$ randomly chosen for all $i \in \{1, \ldots, k\}$ (unless otherwise stated). The other parameters are given concretely. The maximum outer iteration number (i.e., the maximum iteration number allowed for Algorithm 3.1) is 1000. The termination rule is

$$\begin{array}{c} \frac{\|(U_s^{(1)}, \ldots, U_s^{(k)}) \cdot \mathcal{A} - \mathcal{B}_s\|}{\|\mathcal{A}\| \sqrt{\prod_{i=1}^k n_i}} < 10^{-8} \text{ and} \\ \sqrt{\frac{\|\max\{|\mathcal{X}_s| - 1, 0\}\|^2 + \frac{1}{\|\mathcal{A}\|^2} \sum_{i=1}^k \|(U_s^{(i)})^\mathsf{T} \mathcal{X}_s^{(\mathrm{f},i)} (V_s^{(i)})^\mathsf{T} - V_s^{(i)} (\mathcal{X}_s^{(\mathrm{f},i)})^\mathsf{T} (U_s^{(i)})^\mathsf{T}\|^2}{\prod_{i=1}^k n_i}} < 10^{-8}, \end{array} \tag{32}$$

representing the relative feasibility and optimality residuals. Unless otherwise stated, the inner iteration Algorithm 4.1 is terminated whenever either the optimality condition

(17) is satisfied or the number of iterations exceeds 1000. If the number of the outer iteration exceeds 1000 and (32) is not satisfied, then the algorithm *fails* for this test; otherwise the problem is solved successfully.

As problem (12) is nonlinear and nonconvex, the solution found heavily depends on the initial point. Thus, each instance is tested 10 times (unless otherwise stated) with randomly generated initializations. In the tables, the column **sucp** indicates the probability of successful computations in the 10 simulations; **max(len)** is the maximum length of the strongly orthogonal decomposition computed among the simulations, and **min(len)** for the minimum; **prob** is the probability the algorithm computes out a strongly orthogonal decomposition with length **min(len)**. All the other columns are the *mean* of the successfully solved simulations: **len** is the length of the strongly orthogonal decomposition, **out-it** and **inner-it** are respectively the numbers of the outer and inner iterations, **cpu** is the cpu time consumed, **in**(10^{-9}) and **op**(10^{-9}) are respectively the final infeasibility and optimality residuals in the magnitude 10^{-9}.

5.1 Examples

In the following, we test several classes of examples from known literatures.

Example 5.1 This example is taken from Kolda [28, Section 3]. Let $\mathbf{a}, \hat{\mathbf{a}} \in \mathbb{R}^n$ be two unit vectors and orthogonal to each other, and $\sigma_1 > \sigma_2 > 0$. The tensor $\mathcal{A} \in \mathbb{R}^n \otimes \mathbb{R}^n \otimes \mathbb{R}^n$ is given by

$$\mathcal{A} = \sigma_1 \mathbf{a}^{\otimes 3} + \sigma_2 \mathbf{b} \otimes \mathbf{b} \otimes \hat{\mathbf{a}}, \tag{33}$$

where $\mathbf{b} = \frac{1}{\sqrt{2}}(\mathbf{a} + \hat{\mathbf{a}})$. The rank of $\mathcal{A}$ is two, and (33) is a rank decomposition, while it is not a strongly orthogonal decomposition. Obviously, $\mathrm{rank}_{\mathrm{SO}}(\mathcal{A}) \leq 5$, by expanding $\mathbf{b}$ into $\mathbf{a}$ and $\hat{\mathbf{a}}$.

We tested sampled examples with the dimensions n varying from 2 to 8. Usually, we get a strongly orthogonal decomposition with length 5; sometimes we get 4. The starting penalty parameter is 10 and the maximum inner iteration number is $(n-1) * 100$. For each case, 10 simulations were generated with random $[\mathbf{a}, \hat{\mathbf{a}}] \in \mathrm{St}(n, 2)$, and random $\sigma_2 < \sigma_1 \in (0, 1)$. The results are collected in Table 1.

Example 5.2 This example is taken from Ishteva et al. [24, Section 4.1]. Let $\mathbf{a}, \mathbf{b}, \mathbf{c} \in \mathbb{R}^n$ be three unit vectors and orthogonal to each other. The tensor $\mathcal{A} \in \mathbb{R}^n \otimes \mathbb{R}^n \otimes \mathbb{R}^n$ is given by

$$\mathcal{A} = \mathbf{a} \otimes \mathbf{b} \otimes \mathbf{c} + \mathbf{b} \otimes \mathbf{c} \otimes \mathbf{a} + \mathbf{c} \otimes \mathbf{a} \otimes \mathbf{b}.$$

Obviously, this already gives a strongly orthogonal rank decomposition of $\mathcal{A}$.

For each case, 10 simulations were generated with random $[\mathbf{a}, \mathbf{b}, \mathbf{c}] \in \mathrm{St}(n, 3)$. The penalty parameter is chosen as 10. The computational results are listed in Table 2. All simulations were solved by the algorithm successfully. Thus the column **sucp** is omitted. It is easily seen from Table 2 that in most cases the algorithm can find out a strongly orthogonal rank decomposition.

Table 1 Computational results for Example 5.1

n	sucp	len	out-it	inner-it	cpu	in(10^{-9})	op(10^{-9})
2	1	4.9	26.6	110.9	1.4	1.1	5.1
3	1	4.8	46.4	230.5	2.8	1.6	7.6
4	1	4.8	51.6	342.8	4.0	1.5	7.6
5	1	4.9	50.9	443.8	5.0	0.5	0.6
6	1	4.9	51.1	2015.6	19.9	0.4	5.8
7	0.8	4.875	65.5	6129.6	55.4	1.2	6.1
8	0.8	5	108.6	30579.1	285.7	2.3	3.5

Table 2 Computational results for Example 5.2

n	prob	len	max(len)	Out-it	Inner-it	cpu	in(10^{-9})	op(10^{-9})
3	0.7	3.6	5	20.3	50.7	0.8	0.4	5.2
4	0.9	3.2	5	25.1	51.7	0.9	1.3	6.8
5	0.8	3.4	5	28.7	272.2	2.9	3.1	4.6
8	0.8	3.4	5	28.7	598	5.9	1.3	4.5
10	0.8	3.4	5	27.6	279.9	3.1	2.2	5.0
15	1	3	3	20.2	399.3	11.2	0.9	6.4
20	0.8	3.4	5	16.1	105.5	4.0	0.9	5.4
25	1	3	3	15.1	142	5.7	0.4	6.6
30	1	3	3	28.4	4955.7	199.8	0.4	6.0
35	1	3	3	44.2	8971.7	462.5	0.7	1.0
40	1	3	3	52.8	13281.4	858.3	0.05	0.004
45	1	3	3	54.8	13097.9	1106.8	0.04	0.003
50	1	3	3	58.9	18292.3	3415.6	0.004	0.0004

Example 5.3 This example is taken from Kolda [27, Example 3.3]. Let $\mathbf{a}, \mathbf{b} \in \mathbb{R}^n$ be two unit vectors and orthogonal to each other, and $\sigma_1 > \sigma_2 > \sigma_3 > 0$. The tensor $\mathcal{A} \in \mathbb{R}^n \otimes \mathbb{R}^n \otimes \mathbb{R}^n$ is given by

$$\mathcal{A} = \sigma_1 \mathbf{a} \otimes \mathbf{b} \otimes \mathbf{b} + \sigma_2 \mathbf{b} \otimes \mathbf{b} \otimes \mathbf{b} + \sigma_3 \mathbf{a} \otimes \mathbf{a} \otimes \mathbf{a}.$$

The definition of $\mathcal{A}$ gives a strongly orthogonal rank decomposition already. The parameters are chosen the same as Example 5.2. The initialization for the orthogonal matrices is by the factor matrices of the higher order singular value decomposition (HOSVD) of the underlying tensor [14]. Computational results are shown in Table 3.

Example 5.4 This example is taken from Kolda [27, Example 3.6]. Let $\mathcal{U}_1, \mathcal{U}_2 \in \mathbb{R}^{n_1} \otimes \mathbb{R}^{n_2} \otimes \mathbb{R}^{n_3}$ be two unit rank one tensors that are not orthogonal to each other, and $\sigma_1 \geq \sigma_2$. The tensor $\mathcal{A} \in \mathbb{R}^{n_1} \otimes \mathbb{R}^{n_2} \otimes \mathbb{R}^{n_3}$ is given by

$$\mathcal{A} = \sigma_1 \mathcal{U}_1 + \sigma_2 \mathcal{U}_2.$$

Table 3 Computational results for Example 5.3

n	sucp	prob	len	max(len)	out-it	inner-it	cpu	in(10^{-9})	op(10^{-9})
5	0.9	1	3	3	55.1	737.2	7.7	1.9	7.9
10	1	1	3	3	61.1	9840.8	94.0	3.9	5.8
15	0.8	1	3	3	61.5	9883.2	340.0	2.6	6.2
20	0.6	1	3	3	52	10603.1	562.1	2.8	5.9

Table 4 Computational results for Example 5.4

n	sucp	prob	len	max(len)	out-it	inner-it	cpu	in(10^{-9})	op(10^{-9})
(5, 5, 5)	1	1	5	5	51	2155.7	21.0	3.3	5.4
(10, 10, 10)	1	1	5	5	51.7	4764.4	46.0	1.9	3.4
(15, 15, 15)	0.8	1	5	5	43	2210	82.5	4.7	3.0
(10, 15, 20)	0.9	1	5	5	48.4	2279	65.1	3.8	4.5
(15, 20, 30)	0.9	1	5	5	34.2	1429	44.8	3.8	4.8
(20, 30, 40)	1.0	1	5	5	28.9	2108.1	74.2	1.7	6.3

The penalty parameter is chosen as 10. The computational results are given in Table 4.

Example 5.5 This example is taken from Kolda [27, Example 5.1]. Let $\mathbf{a}, \mathbf{b}, \mathbf{c}, \mathbf{d} \in \mathbb{R}^n$ be four unit vectors and orthogonal to each other,

$$\sigma_1 = 1,\ \sigma_2 = 0.75,\ \sigma_3 = \sigma_4 = 0.7,\ \sigma_5 = \sigma_6 = 0.65,$$

and

$$\mathcal{U}_1 = \mathbf{a}^{\otimes 3},\ \mathcal{U}_2 = \mathbf{b}^{\otimes 3},\ \mathcal{U}_3 = \mathbf{a} \otimes \mathbf{c} \otimes \mathbf{d},\ \mathcal{U}_4 = \mathbf{a} \otimes \mathbf{d} \otimes \mathbf{c},$$
$$\mathcal{U}_5 = \mathbf{b} \otimes \mathbf{c} \otimes \mathbf{d},\ \mathcal{U}_6 = \mathbf{b} \otimes \mathbf{d} \otimes \mathbf{c}.$$

The tensor $\mathcal{A} \in \mathbb{R}^n \otimes \mathbb{R}^n \otimes \mathbb{R}^n$ is given by

$$\mathcal{A} = \sum_{i=1}^{6} \sigma_i \mathcal{U}_i.$$

The definition of $\mathcal{A}$ gives a strongly orthogonal rank decomposition already. The penalty parameter is chosen as 10. The computational results are given in Table 5.

Example 5.6 This example is taken from Nie [37, Example 5.6]. The tensor $\mathcal{A} \in \otimes^5 \mathbb{R}^3$ is given by

$$\mathcal{A} = \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix}^{\otimes 5} + \begin{bmatrix} 1 \\ -2 \\ 3 \end{bmatrix}^{\otimes 5} + \frac{1}{3}\begin{bmatrix} 1 \\ -12 \\ -3 \end{bmatrix}^{\otimes 5} + \frac{1}{5}\begin{bmatrix} 1 \\ 12 \\ -13 \end{bmatrix}^{\otimes 5}.$$

Table 5 Computational results for Example 5.5

n	sucp	prob	len	max(len)	out-it	inner-it	cpu	in(10^{-9})	op(10^{-9})
5	1	0.7	6.7	9	54.9	1048.4	10.3	3.8	5.0
10	1	0.7	6.7	9	57.9	3968.9	37.3	1.7	4.5
15	1	0.9	6.2	8	48.5	2311.9	76.3	3.6	4.0
20	1	0.5	7.3	9	40.1	4424.2	150.1	3.9	3.7

Table 6 Computational results for Example 5.7

n	sucp	prob	len	max(len)	out-it	inner-it	cpu	in(10^{-9})	op(10^{-9})
2	1	1	5	5	38.9	197.1	2.3	4.1	4.6
5	1	1	5	5	103.6	39810.7	399.0	5.0	0.006
8	1	1	5	5	374.6	279100.5	2575.2	1.9	0.01
10	1	1	5	5	482	454104.7	3995.8	0.4	0.07

This tensor is a hard one, since the entries of the tensor varies from $\frac{38}{15}$ to $-\frac{369268}{5} = -7.38 \times 10^4$. The computed orthogonal decomposition has rank 213, slightly smaller than the upper bound $3^5 - 15 = 228$ (cf. [22]). We take the penalty parameter 10, and run 100 simulations. Each simulation finds a decomposition with length 213. The average outer iteration number is 83.15, the inner iteration number is 5435.59, the cpu time is 78.771, the infeasibility is 1.7597×10^{-11}, and the optimality residual is 3.6463×10^{-9}.

Example 5.7 This example is taken from Nie [37, Example 5.8]. The tensor $\mathcal{A} \in \mathrm{S}^3(\mathbb{R}^n)$, the subspace of symmetric tensors in $\otimes^3\mathbb{R}^n$, is given by

$$a_{ijk} = ijk - i - j - k, \text{ for all } i, j, k \in \{1, \ldots, n\}.$$

The penalty parameter is chosen as 10. The computational results are given in Table 6.

Example 5.8 This example is taken from Nie[37, Example 5.9]. The tensor $\mathcal{A} \in \mathrm{S}^4(\mathbb{R}^n)$ is given by

$$a_{ijkl} = \tan(ijkl), \text{ for all } i, j, k, l \in \{1, \ldots, n\}.$$

This example is also not easy to solve, since the entries of the tensor vary with large magnitudes. The penalty parameter is chosen as 10. The computational results are given in Table 7.

Example 5.9 This example is taken from Nie [37, Example 5.10]. The tensor $\mathcal{A} \in \mathrm{S}^4(\mathbb{R}^n)$ is given by

$$a_{ijkl} = \sin(i + j + k + l) + \cos(ijkl), \text{ for all } i, j, k, l \in \{1, \ldots, n\}.$$

Table 7 Computational results for Example 5.8

n	sucp	prob	min(len)	len	max(len)	out-it	inner-it	cpu	in(10^{-9})	op(10^{-9})
2	1	1	12	12	12	63.8	699.1	9.4	8.7	0.08
3	1	0.2	10	11.6	12	56.2	524.6	6.5	7.2	1.6
4	1	0.3	51	54.8	57	70.5	3537.9	39.2	2.8	6.1
5	1	0.1	178	191.7	198	105.7	21171.7	233.1	2.2	6.8
6	0.2	0.5	495	508	521	113	41670.5	461.9	4.4	7.3

Table 8 Computational results for Example 5.9

n	sucp	prob	min(len)	len	max(len)	out-it	inner-it	cpu	in(10^{-9})	op(10^{-9})
2	1	1	12	12	12	43.5	209	3.0	1.9	6.6
3	1	1	53	53	53	75.2	2191.7	26.1	0.7	7.5
4	1	0.1	180	193.2	208	108	26385.9	285.3	5.8	6.0
5	0.8	0.375	499	507.5	527	139.8	57917.1	637.6	3.4	7.4

Table 9 Computational results for Example 5.10

n	sucp	prob	min(len)	len	max(len)	out-it	inner-it	cpu	in(10^{-9})	op(10^{-9})
2	1	1	5	5	5	18.4	167.2	1.6	0.2	0.4
3	1	1	15	15	15	70.1	14889.3	121.4	1.3	1.2

The penalty parameter is chosen as 10. The computational results are given in Table 8.

Example 5.10 The last concrete example is taken from Batselier et al. [6, Example 6]. The tensor $\mathcal{A} \in \otimes^3 \mathbb{R}^n$ is given by

$$a_{ijk} = \frac{1}{i+j+k}, \text{ for all } i, j, k \in \{1, \ldots, n\}.$$

Similar reason as that in Example 5.8, this class of tensors is hard. The penalty parameter is 100. The computational results are given in Table 9.

5.2 Completely orthogonally decomposable tensors

We would like to separate a section for the class of completely orthogonally decomposable tensors (CODT), as it is of particular interest [3,23]. For this class of tensors, we can consider an analogue *condition number* for tensors.

A tensor $\mathcal{A} \in \mathbb{R}^{n_1} \otimes \cdots \otimes \mathbb{R}^{n_k}$ is called *completely orthogonally decomposable* (cf. [27,38,41]) if there exist orthogonal matrices

$$A_i = [\mathbf{a}_{i,1}, \ldots, \mathbf{a}_{i,n_i}] \in \mathbb{R}^{n_i \times n_i} \text{ for all } i = 1, \ldots, k$$

and numbers $\lambda_d \in \mathbb{R}_+$ for $d = 1, \ldots, D_0 := \min\{n_1, \ldots, n_k\}$ such that

$$\mathcal{A} = \sum_{d=1}^{D_0} \lambda_d \mathbf{a}_{1,d} \otimes \cdots \otimes \mathbf{a}_{k,d}. \tag{34}$$

Note that some of the λ_d's can be zeros. By eliminating the zeros, we can further assume that a nonzero completely orthogonally decomposable tensor takes the form

$$\mathcal{A} = \sum_{d=1}^{D} \lambda_d \mathbf{a}_{1,d} \otimes \cdots \otimes \mathbf{a}_{k,d}.$$

with $D \le D_0 := \min\{n_1, \ldots, n_k\}$ and $\lambda_d > 0, d = 1, \ldots, D$. It is easy to see that D is then the strongly orthogonal rank of $\mathcal{A}$.

Unlike the matrix case (i.e., $k = 2$), the rank one decomposition of a completely orthogonally decomposable tensor is always unique [41], regardless of the possibility of equal λ_i's. It can be derived from Kruskal's uniqueness theorem [31] as well, in which the prerequisity is the trivial inequality $k - 1 \le (k - 2)D$ when $D \ge 2$ and the case $D = 1$ is obvious. Therefore, in this case, problem (13) has an optimizer with a diagonal tensor $\mathcal{B}$.

For this class of tensors, we implement two tests for stability of the algorithm. The first test is on tensors with respectively small, medium, and large strongly orthogonal ranks. All the tensors are third order, and the dimensions are listed in Table 10 case by case. In this table, rk indicates the rank of the generated tensor. In each case, 10 simulations were generated with the factor orthogonal matrices being the orthogonalization of the columns of randomly generated matrices. The penalty parameter for tensors with dimensions 30, 50, 80, and 100 are chosen respectively as 10, 20, 30, and 40.

Denote by $\lambda_{\max} := \max\{\lambda_d \mid 1 \le d \le D\}$, and $\lambda_{\min} := \min\{\lambda_d \mid 1 \le d \le D\}$. Then

$$\kappa := \frac{\lambda_{\max}}{\lambda_{\min}} \tag{35}$$

will serve as the role of *condition number* in the tensor case. The other test is on third order tensors of strongly orthogonal rank 3 and dimension 30 with different levels of condition numbers, which are indicated as **cond** in Table 11. Computational tests are given in Table 11 for the performance. The penalties for these nine cases are respectively ranging from 20 to 100 with equi-gap 10.

5.3 Random examples

In this section, we test random examples to see the performance of Algorithm 3.1.

We generate two sets of random examples, the first one is generated with each entry being drawn randomly in $[-1, 1]$. The second one is generated as the sum of rk rank one tensors, with each component vector in the rank one tensor being drawn componentwisely in $[-1, 1]$. In this set, we also generate symmetric tensors. The penalty parameter is chosen as 100 for all simulations, and the results are listed in Tables 12, 13 and 14. In Table 14, **sym** indicates the symmetry of the tensors; it is symmetric when it takes value 1, and nonsymmetric otherwise.

Table 10 Computational results for third order CODTs with different ranks

(n, rk)	sucp	prob	min(len)	len	max(len)	out-it	inner-it	cpu	in(10^{-9})	op(10^{-9})
(30,2)	1	1	2	2	2	18	531.5	45.5	3.9	1.7
(50,2)	1	1	2	2	2	11.5	178	34.5	8.6	5.3
(80,2)	1	0.9	2	2.1	3	13	469.5	366.5	1.9	3.0
(100,2)	1	0.9	2	2.2	4	10.3	305.4	425.8	0.9	5.2
(30,15)	1	0.2	15	17.2	19	35.6	1047.6	49.5	2.8	7.0
(50,25)	1	0.4	25	27	33	42.5	1524.3	220.7	0.3	9.1
(80,40)	1	0.1	40	45.8	54	54.8	3646	2690.5	3.8	6.5
(100,50)	1	0.1	52	62.7	79	72.8	13649.8	15725.1	4.6	5.2
(30,28)	1	0.3	32	35.2	38	45	1071	63.9	0.6	8.7
(50,48)	1	0.1	56	60.4	68	47.6	1774.5	241.1	0.3	8.5
(80,78)	1	0.1	80	90.4	96	58.5	3783.7	2331.2	0.3	9.0
(100,98)	1	0.1	102	116.3	134	67.9	10284.9	11312.8	3.4	4.9

Table 11 Computational results for third order CODTs with different condition numbers

(n, cond)	sucp	prob	min(len)	len	max(len)	out-it	inner-it	cpu	in(10^{-9})	op(10^{-9})
(30,2)	1	1	3	3	3	29	140.2	10.0	0.4	6.6
(30,3)	1	1	3	3	3	38.9	266.8	17.5	0.6	8.4
(30,5)	1	1	3	3	3	50.3	606.9	36.3	0.7	8.4
(30,8)	1	1	3	3	3	59.5	2820.1	146.0	0.5	9.1
(30,10)	1	1	3	3	3	57.7	3222.5	161.1	0.9	8.7
(30,15)	1	1	3	3	3	86.9	22528.9	1086.1	0.7	5.7
(30,20)	0.8	1	3	3	3	123.2	49390	2522.1	0.7	0.01
(30,25)	0.7	1	3	3	3	199.4	85041.8	4015.2	1.3	0.08
(30,30)	0.3	1	3	3	3	153.3	64728.6	4866.3	0.9	0.08

Table 12 Computational results for randomly generated nonsymmetric third order tensors

(n, rk)	sucp	prob	min(len)	len	max(len)	out-it	inner-it	cpu	in(10^{-9})	op(10^{-9})
(3,3)	1	1	18	18	18	56.7	822.7	5.3	0.1	6.9
(5,3)	1	0.1	15	17.7	18	71.5	3314.1	18.6	0.7	5.4
(8,3)	1	0.1	13	16.3	18	65.7	681.8	4.3	0.09	7.9
(10,3)	1	0.1	14	17.3	18	78	1423.5	8.5	0.09	6.5
(15,3)	1	0.1	14	16.6	18	59.6	1005.6	27.4	0.8	6.3
(20,3)	1	0.1	13	16	18	66.6	1583.4	44.5	0.1	7.8
(3,10)	1	1	18	18	18	62.5	5387.3	29.3	1.1	7.2
(4,10)	1	1	46	46	46	90.4	38876.7	211.4	3.5	6.0
(5,10)	0.7	1	95	95	95	124.1	88120.4	473.9	4.6	8.0
(6,10)	0.9	1	171	171	171	172.1	145383.1	802.8	2.9	5.4
(7,10)	0.4	1	280	280	280	169.7	146378.2	803.7	2.2	5.3

Table 13 Computational results for randomly generated symmetric third order tensors

(n, rk)	sucp	prob	min(len)	len	max(len)	out-it	inner-it	cpu	in(10^{-9})	op(10^{-9})
(3,3)	1	0.3	15	17	18	51.7	586	3.7	0.7	5.7
(5,3)	1	0.1	15	17.4	18	65.1	2481	14.3	9.9	6.7
(8,3)	1	0.1	15	17.6	18	60.4	843.9	5.4	0.8	8.4
(10,3)	1	0.1	14	17.1	18	62.8	7665.4	43.3	1.3	7.8
(15,3)	1	0.1	14	17.2	18	50.1	661.4	19.1	0.9	7.8
(20,3)	1	0.2	16	18	23	68.5	2279.7	63.0	0.6	7.6
(3,10)	1	0.2	15	17.3	18	56.5	615.6	3.9	0.6	7.7
(4,10)	1	0.2	37	42.7	46	63.8	2994.5	17.0	1.0	6.7
(5,10)	1	0.1	77	86.7	95	122.7	69607.9	381.4	3.5	5.3
(6,10)	0.9	0.111	144	157.111	169	112.1	63797.3	345.5	2.7	3.9
(7,10)	0.8	0.125	259	273.625	280	165	136584.5	744.8	1.5	6.2

Table 14 Performance for randomly generated higher order tensor examples: symmetric and nonsymmetric

$(n, k, \text{rk}, \text{sym})$	sucp	prob	min(len)	len	max(len)	out-it	inner-it	cpu	in(10^{-9})	op(10^{-9})
(30,4,1,0)	1	1	1	1	1	22	230	313.7	0.1	4.1
(30,4,1,1)	1	1	1	1	1	21.9	250.7	339.1	0.07	4.4
(40,4,1,0)	1	1	1	1	1	18.3	196.7	779.1	0.1	5.7
(40,4,1,1)	1	1	1	1	1	16.7	237.1	912.3	0.1	4.4
(10,5,1,0)	1	1	1	1	1	22.5	245.8	43.2	0.1	5.2
(10,5,1,1)	1	1	1	1	1	22.7	207.9	34.4	0.1	5.4
(20,5,1,0)	1	1	1	1	1	19.9	205.2	1392.1	0.1	4.7
(20,5,1,1)	1	0.8	1	1.2	2	18	207.2	1390.0	0.2	6.0
(10,5,2,0)	1	0.5	11	15.3	27	42.1	352.7	60.6	0.9	6.1
(10,5,3,0)	1	0.1	51	93	174	85.7	4775.8	1073.6	3.5	3.6
(20,5,2,0)	1	0.3	11	14.6	18	46.1	797.7	6034.9	1.1	3.8
(10,5,2,1)	1	0.3	11	20.9	27	51.8	767.2	159.0	0.9	5.7
(10,5,3,1)	1	0.1	57	124.3	213	106.4	13165	2639.9	3.3	1.7
(20,5,2,1)	1	0.1	11	20.7	28	34.8	384.4	2672.7	1.5	4.9

5.4 Conclusions

We see that problem (13) is highly nonlinear, especially when the tensor size is large, thus solutions of Algorithm 3.1 depend on the initializations heavily. While, for most cases, the convergence is fast with high accuracy. These can be seen from Tables 1, 2, 10, 11, 12, 13 and 14. We can also see from these computations that when the strongly orthogonal ranks are small relative to the tensor sizes or the tensor sizes are small, the computational performance is very well, which can be seen from Tables 3, 4, 5, 6 and 11. On the other hand, when the tensor components have a large deviation in magnitude or the strongly orthogonal ranks are large, the performance is reduced, which can be seen from Tables 7, 8, 9, 10, 12 and 13.

We want to emphasize Table 11, from which we can see that the condition number defined as (35) plays a key role in the performance of the algorithm. We think this is an intrinsic property of the underlying tensor, which will domonstrate the efficiency of most computations. Since the rank one decomposition for CODT is unique, and then the condition number as (35) for CODTs is well-defined. More sophisticated definition and investigation for general tensors should be carried out in the future. Also, theoretical justification of the dependence of the efficiency of an algorithm on the condition number should be investigated in the next.

6 Conclusions

In this article, computing the strongly orthogonal rank decomposition of a given tensor is formulated as solving an optimization problem with the help of matrix-tensor multiplication. This optimization problem has discrete-valued objective function (the l_0-norm of the tensor) subject to a system of nonlinear equality constraints and a set of orthogonal constraints. As the number of components of a tensor increases exponentially with respect to the tensor size, the number of nonlinear equality constraints becomes huge for tensors with large sizes. For example, it is one million for a sixth order ten dimensional tensor. As we can image, this class of problem is very difficult to solve in general, partly due to (i) the huge number of equality constraints, (ii) the orthogonality constraints, and (iii) the discrete-valued objective function.

Nevertheless, we propose to replace the objective function by a widely used surrogate–the l_1-norm of the tensor. Then, we apply an inexact augmented Lagrangian multiplier method to solve the resulting optimization problem. During the iterations, the orthogonality requirements are always guaranteed. Thus, the algorithm will always return a strongly orthogonal decomposition whatever the termination situations were met. Moreover, the augmented Lagrangian subproblem is solved by a proximal alternating minimization method with the benefit being that each subproblem has a closed formula solution. This is one key ingredient to keep the orthogonality constraints. Global convergence of the ALM algorithm is established without any further assumptions. Surprisingly, though as simple this algorithm as it sounds, the performance of the proposed algorithm is quite well. Extensive numerical computations were conducted with quite sounding efficiency as well as high accuracy. Note that in Table 14, the number of nonlinear equality constraints is 3,200,000 for the case $(n, k) = (20, 5)$.

It follows from the computations that several issues need further investigation. The first is the exactness of the l_1-norm surrogate with respect to the original l_0-norm. It can be seen from the computations that quite often, l_1-norm can realize the strongly orthogonal rank decomposition. Thus, theoretical justifications should be established. The second would be a more efficient method to deal with tensors with larger strongly orthogonal ranks, which are the hard ones in the present computations. Another is that various other surrogates for the l_0-norm should be studied.

Acknowledgements This work is partially supported by National Science Foundation of China (Grant No. 11771328). The author is very grateful for the annoynous referees for their helpful suggestions and comments in revising this paper.

Appendix A. Convergence theorem for PAM

Let $f : \mathbb{R}^{n_1} \times \cdots \times \mathbb{R}^{n_k} \to \mathbb{R} \cup \{+\infty\}$ be a function of the following structure

$$f(\mathbf{x}) = Q(\mathbf{x}_1, \ldots, \mathbf{x}_k) + \sum_{i=1}^{k} g_i(\mathbf{x}_i),$$

where Q is a C^1 (continuously differentiable) function with locally Lipschitz continuous gradient, and $g_i : \mathbb{R}^{n_i} \to \mathbb{R} \cup \{+\infty\}$ is a proper lower semicontinuous function for each $i \in \{1, \ldots, k\}$.

We introduce the following algorithmic scheme to solve the optimization problem

$$\min_{\mathbf{x} \in \mathbb{R}^{n_1} \times \cdots \times \mathbb{R}^{n_k}} f(\mathbf{x}).$$

Since f is proper and lower semicontinuous, $\mathbf{x}$ is an optimizer of this minimization problem only if it is a critical point of f, i.e., $0 \in \partial f(\mathbf{x})$.

Algorithm A.1 *general PAM*

Given parameters $\alpha_j > 0$ with $j \in \{1, \ldots, k\}$, $0 < \underline{c} < \overline{c}$, k symmetric matrices P_j such that $\underline{c} I \preceq P_j \preceq \overline{c} I$ for each $j \in \{1, \ldots, k\}$.

Step 0: Initialization: choose initial guess $\mathbf{x}^0 \in \mathbb{R}^{n_1} \times \cdots \times \mathbb{R}^{n_k}$. Set $s := 1$.

Step 1: For $j = 1, \ldots, k$, find $\mathbf{x}^s, \mathbf{v}^s \in \mathbb{R}^{n_1} \times \cdots \times \mathbb{R}^{n_k}$ such that

$$g_j(\mathbf{x}_j^s) + Q(\mathbf{x}_{s,j}) + \frac{1}{2}\|\mathbf{x}_j^s - \mathbf{x}_j^{s-1}\|_{P_j}^2 \leq g_j(\mathbf{x}_j^{s-1}) + Q(\mathbf{x}_{s,j-1}); \tag{36}$$

$$\mathbf{v}_j^s \in \partial g_j(\mathbf{x}_j^s); \tag{37}$$

$$\|\mathbf{v}_j^s + \nabla_{\mathbf{x}_j} Q(\mathbf{x}_{s,j})\| \leq \alpha_j \|\mathbf{x}_j^s - \mathbf{x}_j^{s-1}\|. \tag{38}$$

where

$$\mathbf{x}_{s,j} = (\mathbf{x}_1^s, \ldots, \mathbf{x}_j^s, \mathbf{x}_{j+1}^{s-1}, \ldots, \mathbf{x}_k^{s-1}),$$

and

$$\|\mathbf{z}\|_{P_j}^2 := \langle P_j \mathbf{z}, \mathbf{z} \rangle.$$

Step 3: If a termination criterion is not reached, set $s := s + 1$ and go to Step 1.

Step 1 can be implemented through several methods. In particular, (36), (37) and (38) are fulfilled if for all $j \in \{1, \ldots, k\}$, $\mathbf{x}_j^s$ is taken as a minimizer of the optimization problem

$$\min_{\mathbf{z}\in\mathbb{R}^{n_j}} g_j(\mathbf{z}) + Q(\mathbf{x}_1^s, \ldots, \mathbf{x}_{j-1}^s, \mathbf{z}, \mathbf{x}_{j+1}^{s-1}, \ldots, \mathbf{x}_k^{s-1}) + \frac{1}{2}\|\mathbf{z} - \mathbf{x}_j^{s-1}\|_{P_j}. \tag{39}$$

We now state the global convergence of Algorithm A.1 for a wide class of objective functions [4, Theorem 6.2].

Theorem A.2 (Proximal Alternating Minimization) *Let f be a Kurdyka–Łojasiewicz function and bounded from below. Let $\{\mathbf{x}^s\}$ be a sequence produced by Algorithm A.1. If $\{\mathbf{x}^s\}$ is bounded, then it converges to a critical point of f.*

Appendix B. Nonsmooth Lagrange multiplier

The following materials can be found in [40, Chapter 10].

Let $X \subseteq \mathbb{R}^n$ be nonempty and closed, $f_0 : \mathbb{R}^n \to \mathbb{R}$ be locally Lipschitz continuous, $F : \mathbb{R}^n \to \mathbb{R}^m$ with $F := (f_1, \ldots, f_m)$ and each f_i locally Lipschitz continuous, and $\theta : \mathbb{R}^m \to \mathbb{R} \cup \{\pm\infty\}$ be proper, lower semicontinuous, convex with effective domain D.

Consider the following optimization problem

$$\min f_0(\mathbf{x}) + \theta(F(\mathbf{x})) \text{ s.t. } \mathbf{x} \in X. \tag{40}$$

If $\overline{\mathbf{x}}$ is a local optimal solution to (40) such that the following constraint qualification being satisfied

$$\mathbf{0} \in \partial(\mathbf{y}^{\mathsf{T}} F)(\overline{\mathbf{x}}) + N_X(\overline{\mathbf{x}}) \text{ and } \mathbf{y} \in N_D(F(\overline{\mathbf{x}})) \Longrightarrow \mathbf{y} = \mathbf{0}, \tag{41}$$

then there exists a vector $\overline{\mathbf{y}}$ such that

$$\mathbf{0} \in \partial(f_0 + \overline{\mathbf{y}}^{\mathsf{T}} F)(\overline{\mathbf{x}}) + N_X(\overline{\mathbf{x}}) \text{ and } \overline{\mathbf{y}} \in \partial\theta(F(\overline{\mathbf{x}})). \tag{42}$$

A vector $\overline{\mathbf{y}}$ satisfying (42) is called a *Lagrange multiplier*, and the pair $(\overline{\mathbf{x}}, \overline{\mathbf{y}})$ satisfying (42) is a Karush–Kuhn–Tucker pair with $\overline{\mathbf{x}}$ a KKT point. Let $M(\overline{\mathbf{x}})$ be the set of Lagrange multipliers for a KKT point $\overline{\mathbf{x}}$. Under the constraint qualification (41), the set $M(\overline{\mathbf{x}})$ is compact.

A particular case is $\theta = \delta_{\{\mathbf{0}\}}$, the indicator function of the set $\{\mathbf{0}\} \subset \mathbb{R}^m$. Then problem (40) reduces to

$$\min_{\mathbf{x}\in X} f_0(\mathbf{x}) \text{ s.t. } f_i(\mathbf{x}) = 0, \text{ for all } i = 1, \ldots, m. \tag{43}$$

If each f_i is continuously differentiable for $i \in \{1, \ldots, m\}$, then the constraint qualification is

$$y_1 \nabla f_1(\overline{\mathbf{x}}) + \cdots + y_m \nabla f_m(\overline{\mathbf{x}}) \in N_X(\overline{\mathbf{x}}) \Longrightarrow \mathbf{y} = \mathbf{0}. \tag{44}$$

It is the *basic constraint qualification* discussed in [39], an extension of the Mangasarian–Fromovitz constraint qualification [35].

The optimality condition (42) becomes

$$\overline{y}_1 \nabla f_1(\overline{\mathbf{x}}) + \cdots + \overline{y}_m \nabla f_m(\overline{\mathbf{x}}) \in \partial f_0(\overline{\mathbf{x}}) + N_X(\overline{\mathbf{x}}),$$

or in a more familiar form as

$$\mathbf{v} + y_1 \nabla f_1(\overline{\mathbf{x}}) + \cdots + y_m \nabla f_m(\overline{\mathbf{x}}) \in N_X(\overline{\mathbf{x}}) \text{ for some } \mathbf{v} \in \partial f_0(\overline{\mathbf{x}}).$$

Appendix C. Proof of Proposition 3.3

Proof It follows from (16) that $U_s^{(i)} \in \mathbb{O}(n_i)$ for all $i \in \{1, \ldots, k\}$ and $s = 1, 2, \ldots$ and hence the sequence $\{\mathbb{U}_s\}$ is bounded.

Let $\Xi_s \in \partial L_{\rho_s}(\mathbb{U}_s, \mathcal{B}_s; \mathcal{X}_s)$ be such that $\|\Xi_s\| \le \epsilon_s$ which is guaranteed by (17). Thus,

$$\Xi_s = \begin{bmatrix} B_s^{(1)} \\ \vdots \\ B_s^{(k)} \\ \mathcal{W}_s \end{bmatrix} + \rho_s \begin{bmatrix} U_s^{(1)} V_s^{(1)} \big[V_s^{(1)}\big]^{\mathsf{T}} - \mathcal{B}_s^{(\mathrm{f},1)} \big[V_s^{(1)}\big]^{\mathsf{T}} + \frac{1}{\rho_s} \mathcal{X}_s^{(\mathrm{f},1)} \big[V_s^{(1)}\big]^{\mathsf{T}} \\ \vdots \\ U_s^{(k)} V_s^{(k)} \big[V_s^{(k)}\big]^{\mathsf{T}} - \mathcal{B}_s^{(\mathrm{f},k)} \big[V_s^{(k)}\big]^{\mathsf{T}} + \frac{1}{\rho_s} \mathcal{X}_s^{(\mathrm{f},k)} \big[V_s^{(k)}\big]^{\mathsf{T}} \\ \mathcal{B}_s - (U_s^{(1)}, \ldots, U_s^{(k)}) \cdot \mathcal{A} - \frac{1}{\rho_s} \mathcal{X}_s \end{bmatrix} \tag{45}$$

for some $\mathcal{W}_s \in \partial \|\mathcal{B}_s\|_1$, and $B_s^{(i)} \in N_{\mathbb{O}(n_i)}(U_s^{(i)})$ for all $i \in \{1, \ldots, k\}$.

It follows from the last row in (45) and (17) that

$$\left\| \rho_s \big(\mathcal{B}_s - (U_s^{(1)}, \ldots, U_s^{(k)}) \cdot \mathcal{A} - \frac{1}{\rho_s} \mathcal{X}_s \big) + \mathcal{W}_s \right\| \le \|\Xi_s\| \le \epsilon_s. \tag{46}$$

By the fact that $\mathcal{W}_s$ is uniformly bounded (cf. (22)), and $\epsilon_s \to 0$, we conclude that $\rho_s \big(\mathcal{B}_s - (U_s^{(1)}, \ldots, U_s^{(k)}) \cdot \mathcal{A} - \frac{1}{\rho_s} \mathcal{X}_s \big)$ is bounded. Therefore, the sequence $\{\mathcal{X}_{s+1}\}$ is bounded by the multiplier update rule (18).

Since $\mathcal{W}_s$ and $\mathcal{X}_s$ are both bounded, it follows from (46) that $\rho_s \big(\mathcal{B}_s - (U_s^{(1)}, \ldots, U_s^{(k)}) \cdot \mathcal{A} \big)$ is bounded. As $\{\rho_s\}$ is a nondecreasing sequence of positive numbers and $\{\mathbb{U}_s\}$ is bounded, we must have that the sequence $\{\mathcal{B}_s\}$ is bounded.

In a conclusion, the sequence $\{\mathbb{U}_s, \mathcal{B}_s, \mathcal{X}_s\}$ is bounded.

For the feasibility, note that $\mathbb{U}_*$ satisfies the orthogonality by (16). The rest proof is divided into two parts, according to the boundedness of the sequence $\{\rho_s\}$.

Part I. Suppose first that the penalty sequence $\{\rho_s\}$ is bounded. By the penalty parameter update rule (19), it follows that ρ_s stabilizes after some s_0, i.e., $\rho_s = \rho_{s_0}$ for all $s \ge s_0$. Thus,

$$\|(U_s^{(1)}, \ldots, U_s^{(k)}) \cdot \mathcal{A} - \mathcal{B}_s\| \le \tau \|(U_{s-1}^{(1)}, \ldots, U_{s-1}^{(k)}) \cdot \mathcal{A} - \mathcal{B}_{s-1}\| \text{ for all } s \ge s_0 + 1. \tag{47}$$

The feasibility result then follows from a standard continuity argument.

Part II. In the following, we assume that $\rho_s \to \infty$ as $s \to \infty$.

Likewise, it follows from the last row in (45) and (17) that

$$\left\| \mathcal{B}_s - (U_s^{(1)}, \ldots, U_s^{(k)}) \cdot \mathcal{A} - \frac{1}{\rho_s}\mathcal{X}_s + \frac{1}{\rho_s}\mathcal{W}_s \right\| \leq \frac{\|\Xi_s\|}{\rho_s} \leq \frac{\epsilon_s}{\rho_s}.$$

By the fact that $\mathcal{W}_s$ and $\mathcal{X}_s$ are both bounded, $\rho_s \to \infty$, and $\epsilon_s \to 0$, we have that

$$\left\| \mathcal{B}_s - (U_s^{(1)}, \ldots, U_s^{(k)}) \cdot \mathcal{A} \right\| \to 0. \tag{48}$$

Thus, by continuity, we have that $(\mathbb{U}_*, \mathcal{B}_*)$ is a feasible point.

In the following, we show that $(\mathbb{U}_*, \mathcal{B}_*)$ is a KKT point. Let

$$\mathcal{M}_s := \rho_s\big((U_s^{(1)}, \ldots, U_s^{(k)}) \cdot \mathcal{A} - \mathcal{B}_s\big).$$

It follows from the above analysis that $\{\mathcal{M}_s\}$ is bounded. By the multiplier update rule (18), the system (45) can be rewritten as

$$\Xi_s = \begin{bmatrix} B_s^{(1)} \\ \vdots \\ B_s^{(k)} \\ \mathcal{W}_s \end{bmatrix} + \begin{bmatrix} \big(\mathcal{M}_s + \mathcal{X}_s\big)^{(\mathrm{f},1)}\big[V_s^{(1)}\big]^{\mathsf{T}} \\ \vdots \\ \big(\mathcal{M}_s + \mathcal{X}_s\big)^{(\mathrm{f},k)}\big[V_s^{(k)}\big]^{\mathsf{T}} \\ -\mathcal{M}_s - \mathcal{X}_s \end{bmatrix}. \tag{49}$$

The boundedness of $\{\mathbb{U}_s, \mathcal{B}_s, \mathcal{X}_s, \mathcal{M}_s\}$ and $\{\Xi_s\}$ implies the boundedness of each $\{B_s^{(i)}\}$ for all $i \in \{1, \ldots, k\}$ as well. We assume without loss of generality that

$$\{\mathbb{U}_s, \mathcal{B}_s, \mathcal{X}_s, \mathcal{M}_s, \mathcal{W}_s, \mathbb{B}u_s\} \to \{\mathbb{U}_*, \mathcal{B}_*, \mathcal{X}_*, \mathcal{W}_*, \mathcal{M}_*, \mathbb{B}_*\} \text{ as } s \to \infty \text{ and } s \in \mathcal{K}$$

for an infinite index set $\mathcal{K} \subseteq \{1, 2, \ldots\}$, and in where

$$\mathbb{B}_s := (B_s^{(1)}, \ldots, B_s^{(k)}) \text{ and } \mathbb{B}_* := (B_*^{(1)}, \ldots, B_*^{(k)}).$$

Taking limitations on both sides of (49) within $\mathcal{K}$, we have then

$$\begin{bmatrix} \big(\mathcal{M}_* + \mathcal{X}_*\big)^{(\mathrm{f},1)}\big[V_*^{(1)}\big]^{\mathsf{T}} \\ \vdots \\ \big(\mathcal{M}_* + \mathcal{X}_*\big)^{(\mathrm{f},1)}\big[V_*^{(1)}\big]^{\mathsf{T}} \\ -(\mathcal{M}_* + \mathcal{X}_*) \end{bmatrix} = - \begin{bmatrix} B_*^{(1)} \\ \vdots \\ B_*^{(k)} \\ \mathcal{W}_* \end{bmatrix},$$

where $V_*^{(i)}$ is defined as (21) with $U^{(i)}$'s being replaced by $U_*^{(i)}$'s. By the closedness of subdifferentials, we have

$$\mathcal{W}_* \in \partial \|\mathcal{B}_*\|_1,$$

and

$$B_*^{(i)} \in N_{\mathbb{O}(n_i)}(U_*^{(i)}) \text{ for all } i \in \{1, \ldots, k\}.$$

Since each $N_{\mathbb{O}(n_i)}(U_*^{(i)})$ is a linear subspace, we have shown that $(\mathbb{U}_*, \mathcal{B}_*)$ is a KKT point of (12) with Lagrange multiplier $\mathcal{X}_* + \mathcal{M}_*$ (cf. (25)). The proof is complete. □

Appendix D. Proof of Proposition 4.2

Proof It is known that for all $i \in \{1, \ldots, k\}$ each orthogonal group $\mathbb{O}(n_i)$ is an algebraic set, defined by a system of polynomial equations. Therefore, $\mathbb{O}(n_i)$ is a semi-algebraic set and its indicator function is semi-algebraic [8]. The l_1-norm $\|\cdot\|_1$ is also semi-algebraic. Also known is that each semi-algebraic function is a Kurdyka–Łojasiewicz function (cf. [9, Appendix]). Thus, as a summation of the l_1-norm, the indicator functions of the orthogonal groups, and polynomials, the augmented Lagrangian function $L_\rho(\cdot, \cdot; \mathcal{X})$ is a Kurdyka–Łojasiewicz function.

If the iteration sequence $\{(\mathbb{U}_s, \mathcal{B}_s)\}$ generated by Algorithm 4.1 is bounded, and the function $L_\rho(\cdot, \cdot; \mathcal{X})$ is bounded from below, then the sequence $\{(\mathbb{U}_s, \mathcal{B}_s)\}$ converges by Theorem A.2.

For any given $\mathcal{X}$, it follows immediately from (14) that the function $L_\rho(\cdot, \cdot; \mathcal{X})$ is bounded from below, since

$$L_\rho(\mathbb{U}, \mathcal{B}; \mathcal{X}) = \|\mathcal{B}\|_1 + \sum_{i=1}^{k} \delta_{\mathbb{O}(n_i)}(U^{(i)}) + \frac{\rho}{2}\left\|(U^{(1)}, \ldots, U^{(k)}) \cdot \mathcal{A} - \mathcal{B} + \frac{1}{\rho}\mathcal{X}\right\|^2 - \frac{1}{2\rho}\|\mathcal{X}\|^2. \tag{50}$$

In the language of Appendix A, the variable $\mathcal{B}$ refers to the $j = 0$-th block variable, and $U^{(j)}$ the j-th block for $j \in \{1, \ldots, k\}$. Then,

$$g_0(\mathcal{B}) := \|\mathcal{B}\|_1, \text{ and } g_j(U^{(j)}) = \delta_{\mathbb{O}(n_j)}(U^{(j)}) \text{ for all } j \in \{1, \ldots, k\},$$

and the function Q is defined naturally to comprise L_ρ in (50).

We first show that (38) is satisfied. By (26), we know that (38) is satisfied by $\alpha_j = \overline{c}$ for all $j \in \{0, 1, \ldots, k\}$.

It follows from (26), (28) and (29) that

$$\begin{aligned}
&L_\rho(\mathbb{U}_{s-1}, \mathcal{B}_s; \mathcal{X}) + \frac{c_s^{(0)}}{2}\|\mathcal{B}_s - \mathcal{B}_{s-1}\|^2 \le L_\rho(\mathbb{U}_{s-1}, \mathcal{B}_{s-1}; \mathcal{X}),\\
&L_\rho\left(\left(U_s^{(1)}, U_{s-1}^{(2)}, \ldots, U_{s-1}^{(k)}\right), \mathcal{B}_s; \mathcal{X}\right) + \frac{c_s^{(1)}}{2}\|U_s^{(1)} - U_{s-1}^{(1)}\|^2 \le L_\rho(\mathbb{U}_{s-1}, \mathcal{B}_s; \mathcal{X}),\\
&\cdots\\
&L_\rho(\mathbb{U}_s, \mathcal{B}_s; \mathcal{X}) + \frac{c_s^{(k)}}{2}\|U_s^{(k)} - U_{s-1}^{(k)}\|^2 \le L_\rho\left(\left(U_s^{(1)}, \ldots, U_s^{(k-1)}, U_{s-1}^{(k)}\right), \mathcal{B}_s; \mathcal{X}\right).
\end{aligned}$$

Summing up these inequalities, we have

$$L_\rho(\mathbb{U}_s, \mathcal{B}_s; \mathcal{X}) + \frac{c}{2}\left(\sum_{i=1}^{k} \|U_s^{(i)} - U_{s-1}^{(i)}\|^2 + \|\mathcal{B}_s - \mathcal{B}_{s-1}\|^2\right) \le L_\rho(\mathbb{U}_{s-1}, \mathcal{B}_{s-1}; \mathcal{X}).$$

Therefore, the sequence $\{L_\rho(\mathbb{U}_s, \mathcal{B}_s; \mathcal{X})\}$ monotonically decreases to a finite limit.

On the other hand, since each component matrix of $\mathbb{U}_s$ is an orthogonal matrix, the sequence $\{\mathbb{U}_s\}$ is bounded. Suppose that the sequence $\{\mathcal{B}_s\}$ is unbounded. Then, it follows from (50) that the sequence $\{L_\rho(\mathbb{U}_s, \mathcal{B}_s; \mathcal{X})\}$ should diverge to infinity, which is an immediate contradiction. Thus, the iteration sequence $\{(\mathbb{U}_s, \mathcal{B}_s)\}$ must be bounded, and hence converges by Theorem A.2.

In the following, we show that $\|\Theta_s\| \to 0$ as $s \to \infty$. First of all, we derive an upper bound estimate for $\|V_s^{(j)} - \tilde{V}_s^{(j)}\|$ as

$$\begin{aligned}
&\|V_s^{(j)} - \tilde{V}_s^{(j)}\| \\
&\quad = \|\big[(U_s^{(1)}, \ldots, U_s^{(j-1)}, I, U_{s-1}^{(j+1)}, \ldots, U_{s-1}^{(k)}) \cdot \mathcal{A}\big]^{(\mathrm{f},j)} \\
&\qquad - \big[(U_s^{(1)}, \ldots, U_s^{(j-1)}, I, U_s^{(j+1)}, \ldots, U_s^{(k)}) \cdot \mathcal{A}\big]^{(\mathrm{f},j)}\| \\
&\quad = \|(U_s^{(1)}, \ldots, U_s^{(j-1)}, I, U_{s-1}^{(j+1)}, \ldots, U_{s-1}^{(k)}) \cdot \mathcal{A} \\
&\qquad - (U_s^{(1)}, \ldots, U_s^{(j-1)}, I, U_s^{(j+1)}, \ldots, U_s^{(k)}) \cdot \mathcal{A}\| \\
&\quad \le \|(U_s^{(1)}, \ldots, U_s^{(j-1)}, I, U_{s-1}^{(j+1)}, \ldots, U_{s-1}^{(k)}) \cdot \mathcal{A} \\
&\qquad - (U_s^{(1)}, \ldots, U_s^{(j-1)}, I, U_{s-1}^{(j+1)}, \ldots, U_{s-1}^{(k-1)}, U_s^{(k)}) \cdot \mathcal{A}\| \\
&\qquad + \|(U_s^{(1)}, \ldots, U_s^{(j-1)}, I, U_{s-1}^{(j+1)}, \ldots, U_{s-1}^{(k-1)}, U_s^{(k)}) \cdot \mathcal{A} \\
&\qquad - (U_s^{(1)}, \ldots, U_s^{(j-1)}, I, U_s^{(j+1)}, \ldots, U_s^{(k)}) \cdot \mathcal{A}\| \\
&\quad = \|(U_{s-1}^{(k)} - U_s^{(k)})\big[(U_s^{(1)}, \ldots, U_s^{(j-1)}, I, U_{s-1}^{(j+1)}, \ldots, U_{s-1}^{(k-1)}, I) \cdot \mathcal{A}\big]^{(\mathrm{f},k)}\| \\
&\qquad + \|(U_s^{(1)}, \ldots, U_s^{(j-1)}, I, U_{s-1}^{(j+1)}, \ldots, U_{s-1}^{(k-1)}, U_s^{(k)}) \cdot \mathcal{A} \\
&\qquad - (U_s^{(1)}, \ldots, U_s^{(j-1)}, I, U_s^{(j+1)}, \ldots, U_s^{(k)}) \cdot \mathcal{A}\| \\
&\quad \le \|\mathcal{A}\|\|U_{s-1}^{(k)} - U_s^{(k)}\| \\
&\qquad + \|(U_s^{(1)}, \ldots, U_s^{(j-1)}, I, U_{s-1}^{(j+1)}, \ldots, U_{s-1}^{(k-1)}, U_s^{(k)}) \cdot \mathcal{A} \\
&\qquad - (U_s^{(1)}, \ldots, U_s^{(j-1)}, I, U_s^{(j+1)}, \ldots, U_s^{(k)}) \cdot \mathcal{A}\| \\
&\quad \le \|\mathcal{A}\|\big(\|U_{s-1}^{(j+1)} - U_s^{(j+1)}\| + \cdots + \|U_{s-1}^{(k)} - U_s^{(k)}\|\big) \\
&\quad \le \|\mathcal{A}\|\|\mathbb{U}_s - \mathbb{U}_{s-1}\|,
\end{aligned}$$

where the second inequality follows from the fact that $U_t^{(i)} \in \mathbb{O}(n_i)$ for all $i \in \{1, \ldots, k\}$ and $t = 1, 2, \ldots$, and the third from a standard induction.

Likewise, we have

$$\|(U_{s-1}^{(1)}, \ldots, U_{s-1}^{(k)}) \cdot \mathcal{A} - (U_s^{(1)}, \ldots, U_s^{(k)}) \cdot \mathcal{A}\| \le \|\mathcal{A}\|\|\mathbb{U}_s - \mathbb{U}_{s-1}\|.$$

Thus, we have

$$\begin{aligned}\|\Theta_s\| &\le (\rho + k\|\mathcal{X}\| + k\rho\|\mathcal{B}_s\|)\|\mathcal{A}\|\|\mathbb{U}_s - \mathbb{U}_{s-1}\| + c_s^{(0)}\|\mathcal{B}_s - \mathcal{B}_{s-1}\| \\ &\quad + \sum_{i=1}^{k} c_s^{(i)}\|U_s^{(i)} - U_{s-1}^{(i)}\| \\ &\le \left[(\rho + k\|\mathcal{X}\| + k\rho\|\mathcal{B}_s\|)\|\mathcal{A}\| + \overline{c}\right]\|\mathbb{U}_s - \mathbb{U}_{s-1}\| + \overline{c}\|\mathcal{B}_s - \mathcal{B}_{s-1}\|.\end{aligned}$$

Since the iteration sequence $\{(\mathbb{U}_s, \mathcal{B}_s)\}$ converges, we conclude that $\|\Theta_s\| \to 0$ as $s \to \infty$. As $\epsilon > 0$ is a given parameter, Algorithm 4.1 terminates after finitely many iterations. □

References

1. Absil, P.-A., Hosseini, S.: A collection of nonsmooth Riemannian optimization problems. In: Hosseini, S., Mordukhovich, B., Uschmajew, A. (eds.) Nonsmooth Optimization and Its Applications, International Series of Numerical Mathematics, vol. 170, pp. 1–15. Birkhäuser, Cham (2019)
2. Absil, P.-A., Mahony, R., Sepulchre, R.: Optimization Algorithms on Matrix Manifolds. Princeton University Press, Princeton (2008)
3. Anandkumar, A., Ge, R., Hsu, D., Kakade, S.M., Telgarsky, M.: Tensor decompositions for learning latent variable models. J. Mach. Learn. Res. **15**, 2773–2832 (2014)
4. Attouch, H., Bolte, J., Svaiter, B.F.: Convergence of descent methods for semi-algebraic and tame problems: proximal algorithms, forward-backward splitting, and regularized Gauss–Seidel methods. Math. Program. **137**, 91–129 (2013)
5. Bader, B.W., Kolda, T.G.: MATLAB Tensor Toolbox Version 2.6, February 2015. http://www.sandia.gov/~tgkolda/TensorToolbox/
6. Batselier, K., Liu, H., Wong, N.: A constructive algorithm for decomposing a tensor into a finite sum of orthonormal rank-1 terms. SIAM J. Matrix Anal. Appl. **36**, 1315–1337 (2015)
7. Bertsekas, D.P.: Constrained Optimization and Lagrange Multiplier Methods. Athena Scientific, Belmont (1982)
8. Bochnak, J., Coste, M., Roy, M.-F.: Real Algebraic Geometry. Ergebnisse der Mathematik und ihrer Grenzgebiete, vol. 36. Springer, Berlin (1998)
9. Bolte, J., Sabach, S., Teboulle, M.: Proximal alternating linearized minimization for nonconvex and nonsmooth problems. Math. Program. **146**, 459–494 (2014)
10. Chen, J., Saad, Y.: On the tensor SVD and the optimal low rank orthogonal approximation of tensors. SIAM J. Matrix Anal. Appl. **30**, 1709–1734 (2009)
11. Chen, Y., Ye, Y., Wang, M.: Approximation hardness for a class of sparse optimization problems. J. Mach. Learn. Res. **20**, 1–27 (2019)
12. Comon, P.: MA identification using fourth order cumulants. Signal Process. **26**, 381–388 (1992)
13. Comon, P.: Independent component analysis, a new concept? Signal Process. **36**, 287–314 (1994)
14. De Lathauwer, L., De Moor, B., Vandewalle, J.: A multilinear singular value decomposition. SIAM J. Matrix Anal. Appl. **21**, 1253–1278 (2000)
15. De Silva, V., Lim, L.-H.: Tensor rank and the ill-posedness of the best low-rank approximation problem. SIAM J. Matrix Anal. Appl. **30**, 1084–1127 (2008)
16. Donoho, D.L.: For most large underdetermined systems of linear equations the minimal 1-norm solution is also the sparsest solution. Commun. Pure Appl. Math. **59**, 797–829 (2006)
17. Franc, A.: Etude Algébrique des Multitableaux: Apports de l'Algébre Tensorielle, Thèse de Doctorat, Spécialité Statistiques. Univ. de Montpellier II, Montpellier (1992)
18. Golub, G.H., Van Loan, C.F.: Matrix Computations, 4th edn. Johns Hopkins University Press, Baltimore (2013)
19. Håstad, J.: Tensor rank is NP-complete. J. Algorithms **11**, 644–654 (1990)
20. Hillar, C.J., Lim, L.-H.: Most tensor problems are NP-hard. J. ACM **60**(6), 1–39 (2013)
21. Horn, R.A., Johnson, C.R.: Matrix Analysis. Cambridge University Press, New York (1985)

22. Hu, S.: Bounds on strongly orthogonal ranks of tensors, revised manuscript (2019)
23. Hu, S., Li, G.: Convergence rate analysis for the higher order power method in best rank one approximations of tensors. Numer. Math. **140**, 993–1031 (2018)
24. Ishteva, M., Absil, P.-A., Van Dooren, P.: Jacobi algorithm for the best low multilinear rank approximation of symmetric tensors. SIAM J. Matrix Anal. Appl. **34**, 651–672 (2013)
25. Jiang, B., Dai, Y.H.: A framework of constraint preserving update schemes for optimization on Stiefel manifold. Math. Program. **153**, 535–575 (2015)
26. Jordan, C.: Essai sur la géométrie à n dimensions. Bull. Soc. Math. **3**, 103–174 (1875)
27. Kolda, T.G.: Orthogonal tensor decompositions. SIAM J. Matrix Anal. Appl. **23**, 243–255 (2001)
28. Kolda, T.G.: A counterexample to the possibility of an extension of the Eckart–Young low-rank approximation theorem for the orthogonal rank tensor decomposition. SIAM J. Matrix Anal. Appl. **24**, 762–767 (2003)
29. Kolda, T.G., Bader, B.W.: Tensor decompositions and applications. SIAM Rev. **51**, 455–500 (2009)
30. Kroonenberg, P.M., De Leeuw, J.: Principal component analysis of three-mode data by means of alternating least squares algorithms. Psychometrika **45**, 69–97 (1980)
31. Kruskal, J.B.: Three-way array: rank and uniqueness of trilinear decompositions, with application to arithmetic complexity and statistics. Linear Algebra Appl. **18**, 95–138 (1977)
32. Landsberg, J.M.: Tensors: Geometry and Applications, Graduate Studies in Mathematics, vol. 128. AMS, Providence (2012)
33. Leibovici, D., Sabatier, R.: A singular value decomposition of a k-way array for principal component analysis of multiway data, PTA-k. Linear Algebra Appl. **269**, 307–329 (1998)
34. Liu, Y.F., Dai, Y.H., Luo, Z.Q.: On the complexity of leakage interference minimization for interference alignment. In: 2011 IEEE 12th International Workshop on Signal Processing Advances in Wireless Communications, pp. 471–475 (2011)
35. Mangasarian, O.L., Fromovitz, S.: The Fritz–John necessary optimality conditions in the presence of equality and inequality constraints. J. Math. Ana. Appl. **7**, 34–47 (1967)
36. Martin, C.D.M., Van Loan, C.F.: A Jacobi-type method for computing orthogonal tensor decompositions. SIAM J. Matrix Anal. Appl. **30**, 1219–1232 (2008)
37. Nie, J.: Generating polynomials and symmetric tensor decompositions. Found. Comput. Math. **17**, 423–465 (2017)
38. Robeva, E.: Orthogonal decomposition of symmetric tensors. SIAM J. Matrix Anal. Appl. **37**, 86–102 (2016)
39. Rockafellar, R.T.: Lagrange multipliers and optimality. SIAM Rev. **35**, 183–238 (1993)
40. Rockafellar, R.T., Wets, R.: Variational Analysis. Grundlehren der Mathematischen Wissenschaften, vol. 317. Springer, Berlin (1998)
41. Zhang, T., Golub, G.H.: Rank-one approximation to high order tensors. SIAM J. Matrix Anal. Appl. **23**, 534–550 (2001)

Affiliations

Shenglong Hu[1]

[1] Department of Mathematics, School of Science, Hangzhou Dianzi University, Hangzhou 310018, China