ITERATIVE METHODS FOR SYMMETRIC OUTER PRODUCT TENSOR DECOMPOSITION

NA LI*, CARMELIZA NAVASCA†, AND CHRISTINA GLENN‡

Abstract. We study the symmetric outer product for tensors. Specifically, we look at decomposition of fully (partially) symmetric tensor into a sum of rank-one fully (partially) symmetric tensors. We present an iterative technique for the third-order partially symmetric tensor and fourth-order fully and partially symmetric tensor. We included several numerical examples which indicate a faster convergence rate for the new algorithms than the standard method of alternating least squares.

Key words. multilinear algebra, tensor products, factorization of matrices

AMS subject classifications. 15A69, 15A23

1. Introduction.

In 1927, Hitchcock [15][16] proposed the idea of the polyadic form of a tensor, that is, expressing a tensor as a sum of a finite number of rank-one tensors. Today, this is called the canonical polyadic (CP) decomposition; it is also known as CANDECOMP or PARAFAC. It has been extensively applied to many problems in various engineering and science disciplines [26, 27, 1, 13, 28, 18]. Specifically, symmetric tensors are ubiquitous in many signal processing applications [6, 8, 12]. In this paper, we look at the symmetric outer product decomposition (SOPD), a summation of rank-one fully (partially) symmetric tensors. More specifically, we provide some iterative methods for approximating the sum of rank-one symmetric tensors for a given symmetric tensor.

SOPD is common in independent component analysis (ICA) [17, 7] or blind source separation (BSS), which is used to separate the true signal from noise and interference in signal processing [8, 12]. When the order of the tensor is three and it is symmetric in two mode dimensions, this is called the individual differences scaling (INDSCAL) model introduced by Carrol and Chang [5, 29].

There are very few numerical methods for finding SOPD. For unsymmetric tensors, a well-known method for finding the sum of rank one terms is the Alternating Least-Squares (ALS) technique [5, 14]. Since SOPD is a special case of CP decomposition, the ALS method can be applied to obtain SOPD. However, this approach is not efficient and is not guaranteed to work since all alternating least squares subproblems lead to the same equation. In addition, the subproblems are now nonlinear least squares problems in the factor matrices. A different method proposed by Comon [3] for SOPD reduces the problem to the decomposition of a linear form. For the fourth-order fully symmetric tensor, De Lathauwer in [12] proposed the Fourth-Order-Only Blind Identification (FOOBI) algorithm. Schultz [25] numerically solves SOPD using the best symmetric rank-1 approximation of a symmetric tensor through the maximum of the associated homogeneous form over the unit sphere. In this paper, we study the SOPD for the third-order partially symmetric tensors and the fourth-order fully and partially symmetric tensors and propose a new method called Partial Column-wise Least-squares (PCLS) to solve the SOPD. It obviates the nonlinear least-squares sub-

*Department of Mathematics, Clarkson University, Potsdam, NY 13699.

†Department of Mathematics, University of Alabama at Birmingham, 1300 University Boulevard, Birmingham, AL, 35294 (cnavasca@uab.edu).

‡Department of Mathematics, University of Alabama at Birmingham, 1300 University Boulevard, Birmingham, AL, 35294.

problems through some tensor unfoldings and a root finding technique for polynomials in estimating factor matrices.

2. Preliminaries. We denote the scalars in $\mathbb{R}$ with lower-case letters $(a, b, \ldots)$ and the vectors with bold lower-case letters $(\mathbf{a}, \mathbf{b}, \ldots)$. The matrices are written as bold upper-case letters $(\mathbf{A}, \mathbf{B}, \ldots)$ and the symbols for tensors are calligraphic letters $(\mathcal{A}, \mathcal{B}, \ldots)$. The subscripts represent the following scalars: $(\mathcal{A})_{ijk} = a_{ijk}$, $(\mathbf{A})_{ij} = a_{ij}$, $(\mathbf{a})_i = a_i$. The r-th column of a matrix $\mathbf{A}$ is $\mathbf{a}_r$.

DEFINITION 2.1 (Mode-n matricization). *Matricization is the process of reordering the elements of a tensor of Nth order into a matrix. The mode-n matricization of a tensor $\mathcal{T} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$ is denoted by $\mathbf{T}_{(n)}$ and arranges the mode-n fibers as the columns of the resulting matrix. The mode-n fiber, $\mathbf{t}_{i_1 \cdots i_{n-1} : i_{n+1} \cdots i_N}$, is a vector obtained by fixing every index with the exception of the nth index.* For example, a third-order tensor $\mathcal{X} \in \mathbb{R}^{I \times J \times K}$ has the following mode-1, mode-2 and mode-3 matricizations of $\mathcal{X}$:

$$\begin{aligned} \mathbf{X}_{(1)} &= [\mathbf{x}_{:11}, \ldots, \mathbf{x}_{:J1}, \mathbf{x}_{:12} \ldots, \mathbf{x}_{:J2}, \ldots, \mathbf{x}_{:1K}, \ldots, \mathbf{x}_{:JK}], \\ \mathbf{X}_{(2)} &= [\mathbf{x}_{1:1}, \ldots, \mathbf{x}_{I:1}, \mathbf{x}_{1:2} \ldots, \mathbf{x}_{I:2}, \ldots, \mathbf{x}_{1:K}, \ldots, \mathbf{x}_{I:K}], \\ \mathbf{X}_{(3)} &= [\mathbf{x}_{11:}, \ldots, \mathbf{x}_{I1:}, \mathbf{x}_{12:} \ldots, \mathbf{x}_{I2:}, \ldots, \mathbf{x}_{1J:}, \ldots, \mathbf{x}_{IJ:}], \end{aligned} \tag{2.1}$$

respectively. These matricizations can be attained through these matlab commands: $\mathbf{X}_{(1)} =$ `reshape(X, I, J*K)`, $\mathbf{X}_{(2)} =$ `reshape(permute(X, [2 1 3]), J, K*I)` and $\mathbf{X}_{(3)} =$ `reshape(permute(X, [3 2 1]), K, J*I)`.

DEFINITION 2.2 (square matricization). *For a fourth-order tensor $\mathcal{T} \in \mathbb{R}^{I \times J \times K \times L}$, the square matricization is denoted by $mat(\mathcal{T}) \in \mathbb{R}^{IJ \times KL}$ and is defined as*

$$\mathbf{T} = mat(\mathcal{T}) \Leftrightarrow (\mathbf{T})_{(i-1)J+j,(k-1)L+l} = \mathcal{T}_{ijkl}. \tag{2.2}$$

See [4] for the generalizations of square matricization in terms of tensor blocks. This is equivalent to the matlab command: $\mathbf{T} =$ `reshape(T, I*J, K*L)`.

DEFINITION 2.3 (unvec). *Given a vector $\mathbf{v} \in \mathbb{R}^{I^2}$, $unvec(\mathbf{v}) = \mathbf{W}$ is a square matrix of size $I \times I$ obtained from matricizing $\mathbf{v}$ through its column vectors $\mathbf{w}_j \in \mathbb{R}^I$, $j = 1, 2, \ldots, I$; i.e.*

$$\mathbf{w}_{ij} = v((j-1) \cdot I + i), \quad i = 1, 2, \ldots, I$$

and

$$unvec(\mathbf{v}) = \begin{bmatrix} \mathbf{w}_1 & \mathbf{w}_2 & \ldots & \mathbf{w}_I \end{bmatrix}.$$

3. Symmetric Outer Product Decomposition. DEFINITION 3.1. *Let $\mathbf{x}, \mathbf{y} \in \mathbb{R}^n$. The outer product of $\mathbf{x}$ and $\mathbf{y}$ is*

$$\mathbf{M} = \begin{bmatrix} x_1y_1 & x_1y_2 & \cdots & x_1y_n \\ x_2y_1 & & & \vdots \\ \vdots & & & \vdots \\ x_ny_1 & & & y_ny_n \end{bmatrix}. \tag{3.1}$$

If $\mathbf{x} = \mathbf{y}$, then we see that $\mathbf{M}$ is a symmetric matrix.

The outer product of the vectors $\mathbf{x}, \mathbf{y}, \mathbf{z} \in \mathbb{R}^n$ is the following:

$$(\mathbf{x} \otimes \mathbf{y} \otimes \mathbf{z})_{ijk} = x_i y_j z_k. \tag{3.2}$$

The outer product of three nonzero vectors is a third-order rank-one tensor; the outer product of k nonzero vectors is a kth-order rank-one tensor. Given $\mathcal{T} = \mathbf{x} \otimes \mathbf{y} \otimes \mathbf{z}$. If $\mathbf{x} = \mathbf{y} = \mathbf{z}$, then we say $\mathcal{T}$ is a symmetric third-order rank-one tensor. We say $\mathcal{T}$ is a partially symmetric third-order rank-one tensor if either $\mathbf{x} = \mathbf{y}$, $\mathbf{x} = \mathbf{z}$ or $\mathbf{y} = \mathbf{z}$ holds.

Definition 3.2 (Rank-one tensor). *A kth order tensor $\mathcal{T} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_k}$ is called rank-one if it can be written as an outer product of k vectors; i.e.*

$$\mathcal{T}_{i_1 i_2 \cdots i_k} = a^{(1)}_{i_1} a^{(2)}_{i_2} \cdots a^{(k)}_{i_k}, \quad \textit{for all } 1 \leq i_r \leq I_r.$$

Conveniently, a rank-one tensor is expressed as

$$\mathcal{T} = \mathbf{a}^{(1)} \otimes \mathbf{a}^{(2)} \otimes \cdots \otimes \mathbf{a}^{(k)},$$

where $\mathbf{a}^{(r)} \in \mathbb{R}^{I_r}$ with $1 \leq r \leq k$.

Definition 3.3 (Partially symmetric rank-one tensor). *A rank-one kth-order tensor $\mathcal{T} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_k}$ is partially symmetric if it can be written as an outer product of k vectors and if there exist modes l and m such that $\mathbf{a}^{(l)} = \mathbf{a}^{(m)}$ where $1 \leq l, m \leq k$ and $l \neq m$ in*

$$\mathcal{T} = \mathbf{a}^{(1)} \otimes \mathbf{a}^{(2)} \otimes \ldots \otimes \mathbf{a}^{(k)}$$

with $\mathbf{a}^{(r)} \in \mathbb{R}^{I_r}$. Moreover, the equivalent mode indices form into disjoint subsets S_i of subindices where $\cup_{i=1}^{\bar{k}} S_i = \{1, 2, \ldots, k\}$ and $\cap_{i=1}^{\bar{k}} S_i = \emptyset$.

Remark 3.1. *If a third-order tensor $\mathcal{T}$ is partially symmetric tensor with $\mathbf{a}^{(1)} = \mathbf{a}^{(2)}$, then*

$$\mathcal{T}_{i_1 i_2 i_3} = \mathcal{T}_{i_2 i_1 i_3}.$$

Definition 3.4 (Symmetric rank-one tensor). *A rank-one kth-order tensor $\mathcal{T} \in \mathbb{R}^{I \times I \times \cdots \times I}$ is symmetric if it can be written as an outer product of k identical vectors; i.e.*

$$\mathcal{T} = \underbrace{\mathbf{a} \otimes \mathbf{a} \otimes \cdots \otimes \mathbf{a}}_{k}$$

where $\mathbf{a} \in \mathbb{R}^I$.

Symmetric rank-one tensor is a special partially rank-one tensor where for any $l \in \{1, 2, \ldots, k\}$, $\mathbf{a} = \mathbf{a}^{(l)}$.

Remark 3.2. *We say a tensor is cubical if the modal dimensions have equal length. Symmetric tensors are cubical. A fully symmetric tensor is invariant under all permutations of its indices. Let the permutation σ be defined as $\sigma(i_1, i_2, \ldots, i_k) = i_{m(1)} i_{m(2)} \ldots i_{m(k)}$ where $m(j) \in \{1, 2, \ldots, k\}$. If $\mathcal{T}$ is a symmetric tensor, then*

$$\mathcal{T}_\sigma = \mathcal{T}_{i_{m(1)} i_{m(2)} \cdots i_{m(k)}}$$

for all permutation σ on the index set $\{i_1, i_2, \ldots, i_k\}$.

A kth-order tensor $\mathcal{T}$ can be decomposed into a sum of outer products of vectors if there exists a positive number R such that

$$\mathcal{T} = \sum_{r=1}^{R} \underbrace{\mathbf{a}_r^{(1)} \otimes \mathbf{a}_r^{(2)} \otimes \cdots \otimes \mathbf{a}_r^{(k)}}_{k} \tag{3.3}$$

exists. This is called the Canonical Polyadic (CP) decomposition (also known as PARAFAC and CANDECOM). This decomposition first appeared in the papers of Hitchcock [15, 16]. The notion of tensor rank was also introduced by Hitchcock.

DEFINITION 3.5. *The rank of $\mathcal{T} \in \mathbb{R}^{I_1 \times \cdots \times I_k}$ is defined as*

$$rank(\mathcal{T}) := \min_{R} \left\{ R \middle| \mathcal{T} = \sum_{r=1}^{R} \mathbf{a}_r^{(1)} \otimes \mathbf{a}_r^{(2)} \otimes \cdots \otimes \mathbf{a}_r^{(k)} \right\}$$

Define $\mathsf{T}^k(\mathbb{R}^n)$ as the set of all order-k dimensional n cubical tensors. A set of symmetric tensors in $\mathsf{T}(\mathbb{R}^n)$ is denoted as $\mathsf{S}^k(\mathbb{R}^n)$.

DEFINITION 3.6. *If $\mathcal{T} \in \mathsf{S}^k(\mathbb{R}^n)$, then the rank of a symmetric $\mathcal{T} \in \mathbb{R}^{I_1 \times \cdots \times I_k}$ is defined as*

$$rank_{\mathsf{S}}(\mathcal{T}) := \min_{S} \left\{ S \middle| \mathcal{T} = \sum_{s=1}^{S} \underbrace{\mathbf{a}_s \otimes \mathbf{a}_s \otimes \cdots \otimes \mathbf{a}_s}_{k} \right\}$$

LEMMA 3.7. *[6] Let $\mathcal{T} \in \mathsf{S}^k(\mathbb{R}^n)$, there exist $\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_S \in \mathbb{R}^n$ linearly independent vectors such that*

$$\mathcal{T} = \sum_{i=1}^{S} \underbrace{\mathbf{x}_i \otimes \mathbf{x}_i \otimes \cdots \otimes \mathbf{x}_i}_{k}$$

has $rank_{\mathsf{S}}(\mathcal{T}) = S$.

Note that $\mathsf{S}^k(\mathbb{R}^n) \subset \mathsf{T}^k(\mathbb{R}^n)$. We have that $R(k,n) \geq R_{\mathsf{S}}(k,n)$ where $R(k,n)$ is the maximally attainable rank in the space of order-k dimension-n cubical tensors $\mathsf{T}^k(\mathbb{R}^n)$ and $R_{\mathsf{S}}(k,n)$ be the maximally attainable symmetric rank in the space of symmetric tensors $\mathsf{S}^k(\mathbb{R}^n)$. In [6, 20], there are numerous results on symmetric rank over $\mathbb{C}$. For example in [6], for all $\mathcal{T}$

- $rank_{\mathsf{S}}(\mathcal{T}) \leq \binom{n+k-1}{k}$
- $rank(\mathcal{T}) \leq rank_{\mathsf{S}}(\mathcal{T})$

We also refer the readers to the book by Landsberg [20] on some discussions on partially symmetric tensor rank and the work of Stegeman [29] on some uniqueness conditions for a minimum rank of the symmetric outer product.

4. Alternating Least-Squares.

Our goal is to approximate a minimum sum of rank-one kth-order symmetric tensors from a given symmetric tensor $\mathcal{T}$. The unsymmetric general problem is given a kth-order tensor $\mathcal{T} \in \mathbb{R}^{I_1 \times I_2 \times \ldots \times I_k}$, find the best minimum sum of rank-one kth-order tensor

$$\min_{R} \|\mathcal{T} - \widetilde{\mathcal{T}}\|_F^2 \tag{4.1}$$

where $\widetilde{\mathcal{T}} = \sum_{r=1}^{R} \mathbf{a}_r^{(1)} \otimes \mathbf{a}_r^{(2)} \otimes \cdots \otimes \mathbf{a}_r^{(k)}$. The ALS method is a popular method for this general problem.

ALS is a numerical method for approximating the canonical decomposition of a given tensor. For simplicity, we describe ALS for third-order tensors. The ALS problem for third order tensor is the following

$$\min_{\mathbf{A},\mathbf{B},\mathbf{C}} \left\| \mathcal{T} - \sum_{r=1}^{R} \mathbf{a}_r \otimes \mathbf{b}_r \otimes \mathbf{c}_r \right\|_F^2$$

where $\mathcal{T} \in \mathbb{R}^{I\times J\times K}$. Define the factor matrices $\mathbf{A}$, $\mathbf{B}$ and $\mathbf{C}$ as the concatenation of the vectors $\mathbf{a}_r$, $\mathbf{b}_r$ and $\mathbf{c}_r$, respectively; i.e., $\mathbf{A} = [\mathbf{a}_1\ \mathbf{a}_2\ \dots \mathbf{a}_R] \in \mathbb{R}^{I\times R}$, $\mathbf{B} = [\mathbf{b}_1\ \mathbf{b}_2\ \dots \mathbf{b}_R] \in \mathbb{R}^{J\times R}$ and $\mathbf{C} = [\mathbf{c}_1\ \mathbf{c}_2\ \dots \mathbf{c}_R] \in \mathbb{R}^{K\times R}$.

Matricizing the equation

$$\mathcal{T} = \sum_{r=1}^{R} \mathbf{a}_r \otimes \mathbf{b}_r \otimes \mathbf{c}_r$$

on both sides, we obtain three equivalent matrix equations:

$$\begin{aligned} \mathbf{T}_{(\mathbf{1})} &= \mathbf{A}(\mathbf{C}\odot\mathbf{B})^{\mathrm{T}}, \\ \mathbf{T}_{(\mathbf{2})} &= \mathbf{B}(\mathbf{C}\odot\mathbf{A})^{\mathrm{T}}, \\ \mathbf{T}_{(\mathbf{3})} &= \mathbf{C}(\mathbf{B}\odot\mathbf{A})^{\mathrm{T}}. \end{aligned}$$

where ${\mathbf{T}_{(\mathbf{1})}}^{I\times JK}$, ${\mathbf{T}_{(\mathbf{2})}}^{J\times IK}$ and ${\mathbf{T}_{(\mathbf{3})}}^{K\times IJ}$ are the mode-1, mode-2 and mode-3 matricizations of tensor $\mathcal{T}$. The symbol $\odot$ denotes the Khatri-Rao product [24]. Given matrices $\mathbf{A} \in \mathbb{R}^{I\times R}$ and $\mathbf{B} \in \mathbb{R}^{J\times R}$, the Khatri-Rao product of $\mathbf{A}$ and $\mathbf{B}$ is the "matching columnwise" Kronecker product; i.e.,

$$\mathbf{A}\odot\mathbf{B} = [\mathbf{a_1}\otimes\mathbf{b_1}\quad \mathbf{a_2}\otimes\mathbf{b_2}\quad \dots] \in \mathbb{R}^{IJ\times R}.$$

By fixing two factor matrices but one at each minimization, three coupled linear least-squares subproblems are then formulated to find each factor matrices:

$$\begin{aligned} \mathbf{A}^{k+1} &= \underset{\widehat{\mathbf{A}}\in\mathbb{R}^{I\times R}}{\operatorname{argmin}} \left\| {\mathbf{T}_{(\mathbf{1})}}^{I\times JK} - \widehat{\mathbf{A}}(\mathbf{C}^k\odot\mathbf{B}^k)^{\mathrm{T}} \right\|_F^2, \\ \mathbf{B}^{k+1} &= \underset{\widehat{\mathbf{B}}\in\mathbb{R}^{J\times R}}{\operatorname{argmin}} \left\| {\mathbf{T}_{(\mathbf{2})}}^{J\times IK} - \widehat{\mathbf{B}}(\mathbf{C}^k\odot\mathbf{A}^{k+1})^{\mathrm{T}} \right\|_F^2, \\ \mathbf{C}^{k+1} &= \underset{\widehat{\mathbf{C}}\in\mathbb{R}^{K\times R}}{\operatorname{argmin}} \left\| {\mathbf{T}_{(\mathbf{3})}}^{K\times IJ} - \widehat{\mathbf{C}}(\mathbf{B}^{k+1}\odot\mathbf{A}^{k+1})^{\mathrm{T}} \right\|_F^2. \end{aligned} \tag{4.2}$$

where $\mathbf{T}_{(\mathbf{1})}$, $\mathbf{T}_{(\mathbf{2})}$ and $\mathbf{T}_{(\mathbf{3})}$ are the standard tensor flattennings described in (2.1). To start the iteration, the factor matrices are initialized with $\mathbf{A}^0$, $\mathbf{B}^0$, $\mathbf{C}^0$. ALS fixes $\mathbf{B}$ and $\mathbf{C}$ to solve for $\mathbf{A}$, then it fixes $\mathbf{A}$ and $\mathbf{C}$ to solve for $\mathbf{B}$. And then ALS finally fixes $\mathbf{A}$ and $\mathbf{B}$ to solve for $\mathbf{C}$. This Gauss-Seidel sweeping process continues iteratively until some convergence criterion is satisfied. Thus the original nonlinear optimization problem can be solved with a sequence three linear least squares problems.

Although ALS has been applied extensively across engineering and science disciplines. However, ALS has some disadvantages. For non-degenerate problems, convergence may require a high number of iterations (see Figure 4.1) which can be attributed to the non-uniqueness in the solutions of the subproblems, collinearity of the columns in the factor matrices and initialization of the factor matrices; see e.g. [9, 23, 30]. The long curve in the residual plot is also an indication of a degeneracy problem.

The ALS algorithm can be applied to find symmetric and partially symmetric outer product decompositions for third order tensors by setting $\mathbf{A} = \mathbf{B} = \mathbf{C}$ and $\mathbf{A} = \mathbf{B}$ or $\mathbf{A} = \mathbf{C}$, respectively, in (4.2). The swamps are prevalent in these cases. Moreover, the factor matrices obtained often do not reflect the symmetry of the tensor. In addition, when ALS is applied to symmetric tensors, the least-squares subproblems can be highly ill-conditioned which lead to non-unique solutions. The regularization methods [22, 21] does not drastically mitigate the requirement for a higher number of iterations.

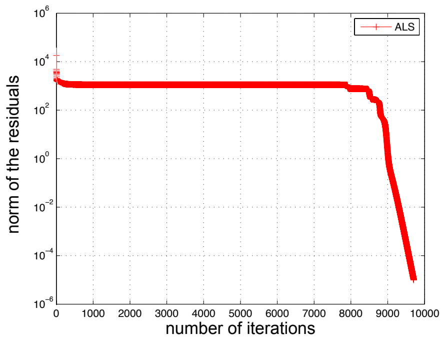


Fig. 4.1: The long flat curve (*swamp*) in the ALS method. The error stays at 10^3 during the first 8000 iterations.

5. Symmetric and partially symmetric tensor decompositions. Here are the problem formulations: given an order-kth tensor $\mathcal{T} \in \mathbb{R}^{I_1 \times I_2 \times \ldots \times I_k}$,

(1) find the best minimum sum of rank-one *symmetric* tensor

$$\min \|\mathcal{T} - \widetilde{\mathcal{T}}\|_F^2 \qquad \text{(Problem 1)}$$

where $\widetilde{\mathcal{T}} = \sum_{r=1}^{R} \mathbf{a}_r \otimes \mathbf{a}_r \otimes \cdots \otimes \mathbf{a}_r$

(2) find the best minimum sum of rank-one *partially symmetric* tensor

$$\min \|\mathcal{T} - \widetilde{\mathcal{T}}\|_F^2 \qquad \text{(Problem 2)}$$

where $\widetilde{\mathcal{T}} = \sum_{r=1}^{R} \mathbf{a}_r^{(1)} \otimes \mathbf{a}_r^{(2)} \otimes \ldots \otimes \mathbf{a}_r^{(k)}$ for some modes $\mathbf{a}_r^{(j)} = \mathbf{a}_r^{(l)}$ where $1 \leq j, l \leq k$ and $j \neq l$.

We refer to these decomposition symmetric outer product decompositions (SOPD).

We describe the decomposition methods for third-order and fourth-order tensors with partial and full symmetries. Later, we outline on how these methods can be extended to the general case in our future line of research.

5.1. SOPD for Third-order Partially Symmetric Tensor. Given a third-order tensor $\mathcal{T} \in \mathbb{R}^{I\times I\times K}$ with $t_{ijk} = t_{jik}$, Problem 2 becomes

$$\min_{\mathbf{A},\mathbf{C}} \left\| \mathcal{T} - \sum_{r=1}^{R_{ps}} \mathbf{a}_r \otimes \mathbf{a}_r \otimes \mathbf{c}_r \right\|_F^2, \tag{5.1}$$

with R_{ps} summands of rank-one partial symmetric tensors and $\widehat{\mathcal{T}} = \sum_{r=1}^{R_{ps}} \mathbf{a}_r \otimes \mathbf{a}_r \otimes \mathbf{c}_r$. The unknown vectors are arranged into two factor matrices $\mathbf{A} = [\mathbf{a}_1\ \mathbf{a}_2\ \cdots\ \mathbf{a}_{R_{ps}}]$ and $\mathbf{C} = [\mathbf{c}_1\ \mathbf{c}_2\ \cdots\ \mathbf{c}_{R_{ps}}]$ in this case. Matricization of $\widehat{\mathcal{T}}$ leads to

$$\widehat{\mathbf{T}}_{(\mathbf{3})} = \mathbf{C}(\mathbf{A} \odot \mathbf{A})^{\mathrm{T}},$$

where $\widehat{\mathbf{T}}_{(\mathbf{3})} \in \mathbb{R}^{K\times I^2}$ is the mode-3 matricization of tensor $\widehat{\mathcal{T}}$. Thus (5.1) becomes

$$\min_{\mathbf{A},\mathbf{C}} \left\| \mathbf{T}_{(\mathbf{3})} - \mathbf{C}(\mathbf{A} \odot \mathbf{A})^{\mathrm{T}} \right\|_F^2. \tag{5.2}$$

If we apply the ALS method, the problem reduces to the following subproblems:

$$\mathbf{A}^{k+1} = \underset{\widehat{\mathbf{A}}\in\mathbb{R}^{I\times R_{ps}}}{\operatorname{argmin}} \left\| \mathbf{T}_{(\mathbf{3})} - \mathbf{C}^k(\widehat{\mathbf{A}} \odot \widehat{\mathbf{A}})^{\mathrm{T}} \right\|_F^2, \tag{5.3}$$

$$\mathbf{C}^{k+1} = \underset{\widehat{\mathbf{C}}\in\mathbb{R}^{K\times R_{ps}}}{\operatorname{argmin}} \left\| \mathbf{T}_{(\mathbf{3})} - \widehat{\mathbf{C}}(\mathbf{A}^{k+1} \odot \mathbf{A}^{k+1})^{\mathrm{T}} \right\|_F^2. \tag{5.4}$$

Directly applying the ALS method to (5.1) does not work. For symmetric problems, at least one is a nonlinear least squares problem; e.g. see (5.12). The consequences of the ALS approach lead to factor matrices that do not satisfy tensor symmetries and/or it takes a high number of iterations (swamps) if it converges at all.

To obviate this problem, we find an alternative method to solve for the factor matrix $\mathbf{A}$. Note that once $\mathbf{A}$ is solved, then $\mathbf{C}$ is solved via linear least squares. Recall that $\mathbf{T}_{(\mathbf{3})} = \mathbf{C}^k(\widehat{\mathbf{A}} \odot \widehat{\mathbf{A}})^{\mathrm{T}}$ can be solved for $\widehat{\mathbf{A}} \odot \widehat{\mathbf{A}}$; i.e.

$$\widehat{\mathbf{A}} \odot \widehat{\mathbf{A}} = ((\mathbf{C}^k)^{\dagger}\mathbf{T}_{(\mathbf{3})})^{\mathrm{T}} \tag{5.5}$$

where $(\cdot)^{\dagger}$ denotes the Moore-Penrose pseudoinverse. Equivalently, (5.5) can be written as

$$\widehat{\mathbf{a}}_r \otimes \widehat{\mathbf{a}}_r = ((\mathbf{C}^k)^{\dagger}\mathbf{T}_{(\mathbf{3})})^{\mathrm{T}}(:,r) \Leftrightarrow \widehat{\mathbf{a}}_r \cdot \widehat{\mathbf{a}}_r^{\mathrm{T}} = \mathit{unvec}\left(((\mathbf{C}^k)^{\dagger}\mathbf{T}_{(\mathbf{3})})^{\mathrm{T}}(:,r)\right) \tag{5.6}$$

where $r = 1, 2, \ldots, R_{ps}$, $\widehat{\mathbf{a}}_r$ is the rth column of matrix $\widehat{\mathbf{A}}$ and $\mathit{unvec}\left(((\mathbf{C}^k)^{\dagger}\mathbf{T}_{(\mathbf{3})})^{\mathrm{T}}(:,r)\right)$ is a matrix $(I\times I)$ obtained from the vector $((\mathbf{C}^k)^{\dagger}\mathbf{T}_{(\mathbf{3})})^{\mathrm{T}}(:,r)$ via column vector stacking of size I. With (5.6), we can obtain $\widehat{\mathbf{A}}$ by calculating each of its column $\widehat{\mathbf{a}}_r$ at a time. We call this approach as the partial column-wise least squares (PCLS) method, a Cholesky-like factorization for a symmetric Khatri-Rao product.

Let $\mathbf{x} \in \mathbb{R}^I = [x_1\ x_2\ \cdots\ x_I]^{\mathrm{T}}$ denote the unknown vector $\widehat{\mathbf{a}}_r$ and $\mathbf{Y} = \mathit{unvec}\left(((\mathbf{C}^k)^{\dagger}\mathbf{T}_{(\mathbf{3})})^{\mathrm{T}}(:,r)\right) \in \mathbb{R}^{I\times I}$. Then (5.6) becomes

$$\begin{bmatrix} x_1^2 & x_1x_2 & \cdots & x_1x_I \\ x_1x_2 & x_2^2 & & \\ \vdots & & \ddots & \\ x_1x_I & & & x_I^2 \end{bmatrix} = \mathbf{Y}.$$

Notice that the unknown x_1 is only involved in the first column and first row, so we only take the first column and first row elements of $\mathbf{Y}$. Thus, the least-squares formulation for these elements is

$$x_1^* = \arg\min_{x_1} \ (y_{11} - x_1^2)^2 + \sum_{i=2}^{I} \left[(y_{i1} - x_i x_1)^2 + (y_{1i} - x_i x_1)^2\right]. \tag{5.7}$$

This cost function in (5.8) is a fourth-order polynomial in one variable x_1. Thus each component x_i can be solved in the same manner of minimizing a fourth-order polynomial.

Generally, for each $m = 1, \cdots, I$, the least-squares formulation at $(k+1)$th iteration is

$$(x_m^*)^{k+1} = \arg\min_{x_m} \ (y_{mm} - (x_m^k)^2)^2 + \sum_{\substack{i=1\\ i\neq m}}^{I} \left[(y_{im} - x_i^k x_m^k)^2 + (y_{mi} - x_i^k x_m^k)^2\right] \tag{5.8}$$

Thus, we have a system of fourth-order optimization problems. See Table 5.1 In practice, a fast root finding method is used to solve for the zeros of a cubic polynomial. Specifically, `roots` in matlab is used in the implementation in the numerical examples discussed in Section 6. It is fast and more reliable than implementing SVD/EVD. The SVD/EVD approximations often lead to a high number of iterations for cases where the numerical experiment converges.

Here are the two subproblems with two initial factor matrices $\mathbf{A}^0$ and $\mathbf{C}^0$:

$$\mathbf{a}_r^{k+1} = \underset{\widehat{a}_r \in \mathbb{R}^I}{\text{argmin}} \left\| unvec\left(((\mathbf{C}^k)^\dagger \mathbf{T}_{(\mathbf{3})})^{\mathrm{T}}(:, r)\right) - \widehat{\mathbf{a}}_r \cdot \widehat{\mathbf{a}}_r^{\mathrm{T}} \right\|_F^2, \quad r = 1, \ldots, R_{ps}, \tag{5.9}$$

and (5.10)

$$\mathbf{C}^{k+1} = \underset{\widehat{\mathbf{C}} \in \mathbb{R}^{K \times R_{ps}}}{\text{argmin}} \left\| \mathbf{T}_{(\mathbf{3})} - \widehat{\mathbf{C}}(\mathbf{A}^{k+1} \odot \mathbf{A}^{k+1})^{\mathrm{T}} \right\|_F^2 \tag{5.11}$$

for approximating $\mathbf{A}$ and $\mathbf{C}$. Starting from the initial guesses, the first subproblem is solved for each column $\mathbf{a}_r$ of $\mathbf{A}$ while $\mathbf{C}$ is fixed; this method is called the iterative Partial Column-wise Least-Squares (PCLS). See Table 5.1. Then in the second subproblem, we fixed $\mathbf{A}$ to solve for $\mathbf{C}$. This process continues iteratively until some convergence criterion, the upperbound for the residual and the maximum number of iterations, are satisfied.

For the cases $t_{ijk} = t_{ikj}$ ($\mathbf{B} = \mathbf{C}$) and $t_{ijk} = t_{jki}$ ($\mathbf{A} = \mathbf{C}$), the optimization problems are

$$\min_{\mathbf{A},\mathbf{C}} \left\| \mathcal{T} - \sum_{r=1}^{R_{ps}} \mathbf{a}_r \otimes \mathbf{c}_r \otimes \mathbf{c}_r \right\|_F^2 \Longleftrightarrow \min_{\mathbf{A},\mathbf{C}} \left\| \mathbf{T}_{(\mathbf{1})} - \mathbf{A}(\mathbf{C} \odot \mathbf{C})^{\mathrm{T}} \right\|_F^2$$

and

$$\min_{\mathbf{A},\mathbf{B}} \left\| \mathcal{T} - \sum_{r=1}^{R_{ps}} \mathbf{b}_r \otimes \mathbf{a}_r \otimes \mathbf{a}_r \right\|_F^2 \Longleftrightarrow \min_{\mathbf{A},\mathbf{B}} \left\| \mathbf{T}_{(\mathbf{2})} - \mathbf{B}(\mathbf{A} \odot \mathbf{A})^{\mathrm{T}} \right\|_F^2.$$

```
Find $\mathbf{A}^* = \mathrm{argmin}_{\mathbf{A}} \|\mathbf{T} - \mathbf{C}(\mathbf{A} \odot \mathbf{A})^T\|_F^2$
```

```
%Solve for $\mathbf{A} \in \mathbb{R}^{I \times R}$ in $\mathbf{A} \odot \mathbf{A} = \mathbf{Y}$ where $\mathbf{Y} = (\mathbf{C}^\dagger \mathbf{T})^T$
Input: $\mathbf{T} \in \mathbb{R}^{K \times I^2}$, $\mathbf{C} \in \mathbb{R}^{K \times R}$.
for r=1:R
  Matricize column equation: $\mathbf{a}_r \otimes \mathbf{a}_r = \mathbf{Y}(:,r) \rightarrow \mathbf{a}_r \cdot \mathbf{a}_r^T = unvec(\mathbf{Y}(:,r))$
  %Solve $\mathbf{a}_r^{k+1} = \underset{\widehat{a}_r \in \mathbb{R}^I}{\mathrm{argmin}} \|unvec(\mathbf{Y})(:,r)) - \mathbf{a}_r \cdot \mathbf{a}_r^T\|_F^2$
    for m=1:I
      $(a_r)_m^* = \arg\min_{(a_r)_m} \; (y_{mm} - ((a_r)_m)^2)^2 + \sum_{\substack{i=1 \\ i \neq m}}^{I} \big[(y_{im} - (a_r)_i (a_r)_m)^2$
                $+ (y_{mi} - x_i (a_r)_m)^2\big]$
    end
end
```

Table 5.1: Partial Column-wise Least-Squares (PCLS)

Here are the corresponding subproblems:

$$\mathbf{C}^{k+1} = \underset{\widehat{\mathbf{C}} \in \mathbb{R}^{K \times R_{ps}}}{\mathrm{argmin}} \left\| \mathbf{T}_{(\mathbf{1})} - \mathbf{A}^k (\widehat{\mathbf{C}} \odot \widehat{\mathbf{C}})^{\mathrm{T}} \right\|_F^2,$$

$$\mathbf{A}^{k+1} = \underset{\widehat{\mathbf{A}} \in \mathbb{R}^{I \times R_{ps}}}{\mathrm{argmin}} \left\| \mathbf{T}_{(\mathbf{1})} - \widehat{\mathbf{A}} (\mathbf{C}^{k+1} \odot \mathbf{C}^{k+1})^{\mathrm{T}} \right\|_F^2$$

and

$$\mathbf{A}^{k+1} = \underset{\widehat{\mathbf{A}} \in \mathbb{R}^{J \times R_{ps}}}{\mathrm{argmin}} \left\| \mathbf{T}_{(\mathbf{2})} - \mathbf{B}^k (\widehat{\mathbf{A}} \odot \widehat{\mathbf{A}})^{\mathrm{T}} \right\|_F^2,$$

$$\mathbf{B}^{k+1} = \underset{\widehat{\mathbf{B}} \in \mathbb{R}^{I \times R_{ps}}}{\mathrm{argmin}} \left\| \mathbf{T}_{(\mathbf{2})} - \widehat{\mathbf{B}} (\mathbf{A}^{k+1} \odot \mathbf{A}^{k+1})^{\mathrm{T}} \right\|_F^2.$$

The advantage of our iterative PCLS over ALS is that it directly computes two factor matrices. If the ALS method is applied to this problem, then one has to update three factor matrices even though there are only two distinct factors in each iteration. In addition, a very high number of iterations is required for this ALS problem to converge and it also not guaranteed that the solution satisfies the symmetries. The ALS method solves three linear least squares problems in each iteration, while PCLS solves one linear least squares and minimizes R_{ps} quartic polynomials one iteration. This is equivalent to finding the roots of a cubic polynomials. The operational cost of running PCLS on a third-order tensor is less than the requirement of ALS since only one linear least-squares is performed with an operational count of $\mathcal{O}(n^3)$ where n reflects the size of the system. A root-finding solver for a cubic polynomial is implemented. Fast numerical methods like Newton's method could be implemented with a complexity of $\mathcal{O}(M(n))$ where $M(n)$ is the operational cost for the choice of multiplication for n-digit precision.

5.2. SOPD for Fourth-order Partially Symmetric Tensors. We can apply PCLS on the fourth-order partial symmetric tensor. Here we consider the following cases.

5.2.1. Case 1: two pairs of similar factor matrices. Let us consider the fourth-order partially symmetric tensor $\mathcal{X} \in \mathbb{R}^{I\times I\times J\times J}$ with $x_{ijkl} = x_{jill}$ and $x_{ijkl} = x_{ijlk}$. With the given symmetries, the number of unknown factors have been reduced to two, $\mathbf{A}$ and $\mathbf{C}$, since $\mathbf{A} = \mathbf{B}$ and $\mathbf{C} = \mathbf{D}$. Then, the problem is to find factor matrices $\mathbf{A}$ and $\mathbf{C}$ through the following minimization

$$\min_{\mathbf{A},\mathbf{C}} \left\| \mathcal{X} - \sum_{r=1}^{R} \mathbf{a}_r \otimes \mathbf{a}_r \otimes \mathbf{c}_r \otimes \mathbf{c}_r \right\|_F^2 ,$$

where $\mathbf{A} = [\mathbf{a}_1\ \mathbf{a}_2\ \cdots\ \mathbf{a}_R]$ and $\mathbf{C} = [\mathbf{c}_1\ \mathbf{c}_2\ \cdots\ \mathbf{c}_R]$. By using the square matricization, we obtain

$$mat(\mathcal{X}) = (\mathbf{A} \odot \mathbf{A})(\mathbf{C} \odot \mathbf{C})^T. \tag{5.12}$$

To solve the equation (5.12) for $\mathbf{A}$ and $\mathbf{C}$, we apply PCLS on

$$\begin{aligned} \mathbf{a}_r \otimes \mathbf{a}_r &= mat(\mathcal{X})((\mathbf{C} \odot \mathbf{C})^T)^\dagger(:, r), \quad r = 1, \dots R \\ \mathbf{c}_r \otimes \mathbf{c}_r &= mat(\mathcal{X})^T((\mathbf{A} \odot \mathbf{A})^T)^\dagger(:, r), \quad r = 1, \dots R \end{aligned}$$

iteratively. Again, we only need to solve for the global minima of two fourth-order polynomials.

Now consider the fourth-order partially symmetric tensor $\mathcal{X} \in \mathbb{R}^{I\times J\times I\times J}$ with $x_{ijkl} = x_{kjil}$ and $x_{ijkl} = x_{ilkj}$. The problem is to find factor matrices $\mathbf{A}$ and $\mathbf{B}$ through the following minimization

$$\min_{\mathbf{A},\mathbf{B}} \left\| \mathcal{X} - \sum_{r=1}^{R} \mathbf{a}_r \otimes \mathbf{b}_r \otimes \mathbf{a}_r \otimes \mathbf{b}_r \right\|_F^2 ,$$

where $\mathbf{A} = [\mathbf{a}_1\ \mathbf{a}_2\ \cdots\ \mathbf{a}_R]$ and $\mathbf{B} = [\mathbf{b}_1\ \mathbf{b}_2\ \cdots\ \mathbf{b}_R]$.

Before matrcizing, we permute the indices of $\mathcal{X}$ where the modes are reordered from $1, 2, 3, 4$ to $1, 3, 2, 4$. Then, by using the square matricization, we obtain

$$mat(\mathcal{X}) = (\mathbf{A} \odot \mathbf{A})(\mathbf{B} \odot \mathbf{B})^T$$

which leads to

$$\begin{aligned} \mathbf{a}_r \otimes \mathbf{a}_r &= mat(\mathcal{X})((\mathbf{B} \odot \mathbf{B})^T)^\dagger(:, r), \quad r = 1, \dots R \\ \mathbf{b}_r \otimes \mathbf{b}_r &= mat(\mathcal{X})^T((\mathbf{A} \odot \mathbf{A})^T)^\dagger(:, r), \quad r = 1, \dots R. \end{aligned}$$

Similarly, for the case when the symmetries, $x_{ijkl} = x_{ljki}$ and $x_{ijkl} = x_{ikjl}$, we minimize

$$\min_{\mathbf{A},\mathbf{B}} \left\| \mathcal{X} - \sum_{r=1}^{R} \mathbf{a}_r \otimes \mathbf{b}_r \otimes \mathbf{b}_r \otimes \mathbf{a}_r \right\|_F^2 ,$$

where $\mathbf{A} = [\mathbf{a}_1\ \mathbf{a}_2\ \cdots\ \mathbf{a}_R]$ and $\mathbf{B} = [\mathbf{b}_1\ \mathbf{b}_2\ \cdots\ \mathbf{b}_R]$. We permute the indices of $\mathcal{X}$ from $[1, 2, 3, 4]$ to $[1, 4, 2, 3]$ to achieve the matricization,

$$mat(\mathcal{X}) = (\mathbf{A} \odot \mathbf{A})(\mathbf{B} \odot \mathbf{B})^T.$$

5.2.2. Case 2: one pair of similar factor matrices. Consider the fourth-order partially symmetric tensor $\mathcal{X} \in \mathbb{R}^{I\times J\times I\times K}$ with $x_{ijkl} = x_{kjil}$. Tensor $\mathcal{X}$ is partially symmetric in mode one and mode three. We find factor matrices $\mathbf{A}$, $\mathbf{B}$ and $\mathbf{C}$ via

$$\min_{\mathbf{A},\mathbf{B},\mathbf{C}} \left\| \mathcal{X} - \sum_{r=1}^{R} \mathbf{a}_r \otimes \mathbf{b}_r \otimes \mathbf{a}_r \otimes \mathbf{c}_r \right\|_F^2 ,$$

where $\mathbf{A} = [\mathbf{a}_1\ \mathbf{a}_2\ \cdots\ \mathbf{a}_R]$, $\mathbf{B} = [\mathbf{b}_1\ \mathbf{b}_2\ \cdots\ \mathbf{b}_R]$ and $\mathbf{C} = [\mathbf{c}_1\ \mathbf{c}_2\ \cdots\ \mathbf{c}_R]$.

Before matrcizing, we permute the indices of $\mathcal{X}$ where the modes are reordered from $1, 2, 3, 4$ to $1, 3, 2, 4$. Then by using the square matricization, we obtain

$$mat(\mathcal{X}) = (\mathbf{A} \odot \mathbf{A})(\mathbf{B} \odot \mathbf{C})^{\mathrm{T}}.$$

From this matricized equation, we can get two equations:

$$\mathbf{a}_r \otimes \mathbf{a}_r = mat(\mathcal{X})((\mathbf{B} \odot \mathbf{B})^T)^{\dagger}(:, r), \quad r = 1, \ldots R \tag{5.13}$$

and

$$\mathbf{b}_r \otimes \mathbf{c}_r = mat(\mathcal{X})^T((\mathbf{A} \odot \mathbf{A})^T)^{\dagger}(:, r), \quad r = 1, \ldots R. \tag{5.14}$$

One can apply PCLS to extract $\mathbf{a}_r$ from the symmetric Khatri-Rao product (5.13). A rank-one SVD can be applied to find the decomposition of the asymmetric Khatri-Rao product (5.14) while rank-one EVD may be used for (5.13). In practice, these approximations lead to a high number of outer loop iterations.

5.3. SOPD for Fourth-order Fully Symmetric Outer Product Decomposition. Given a fourth-order fully symmetric tensor $\mathcal{T} \in \mathbb{R}^{I\times I\times I\times I}$ with $t_{ijkl} = t_{\sigma(ijkl)}$ for any permutation σ on the index set $\{ijkl\}$. We want to a find factor matrix $\mathbf{A} \in \mathbb{R}^{I\times R_s}$ such that

$$\min_{\mathbf{A}} \left\| \mathcal{T} - \sum_{r=1}^{R_s} \mathbf{a}_r \otimes \mathbf{a}_r \otimes \mathbf{a}_r \otimes \mathbf{a}_r \right\|_F^2 , \tag{5.15}$$

where $\mathbf{A} = [\mathbf{a}_1\ \mathbf{a}_2\ \cdots\ \mathbf{a}_{R_s}]$.

By using the square matricization, we have

$$\mathbf{T} = (\mathbf{A} \odot \mathbf{A})(\mathbf{A} \odot \mathbf{A})^{\mathrm{T}}. \tag{5.16}$$

Since $\mathcal{T}$ is symmetric, then $\mathbf{T}$ is a symmetric matrix. Then it follows that there exists a matrix $\mathbf{E}$ such that

$$\mathbf{T} = \mathbf{E}\mathbf{E}^{\mathrm{T}}. \tag{5.17}$$

Comparing the equations (5.16) and (5.17), we know that there exists an orthogonal matrix $\mathbf{Q}$ such that

$$\mathbf{E} = (\mathbf{A} \odot \mathbf{A})\mathbf{Q}, \tag{5.18}$$

where $\mathbf{Q} \in \mathbb{R}^{R_s\times R_s}$ is an orthogonal matrix. In equation (5.18), the unknowns are $\mathbf{A}$ and $\mathbf{Q}$ while $\mathbf{E}$ is known.

Therefore, given the the initial guess matrix $\mathbf{A}^0$ and any starting orthogonal matrix $\mathbf{Q}^0$, we can update the factor matrix by following subproblems:

$$\mathbf{A}^{k+1} = \underset{\widehat{\mathbf{A}} \in \mathbb{R}^{I \times R_s}}{\operatorname{argmin}} \left\| \mathbf{E} - (\widehat{\mathbf{A}} \odot \widehat{\mathbf{A}}) \mathbf{Q}^k \right\|_F^2, \tag{5.19}$$

and

$$\mathbf{P} = \underset{\widehat{\mathbf{Q}} \in \mathbb{R}^{R_s \times R_s}}{\operatorname{argmin}} \left\| \mathbf{E} - (\mathbf{A}^{k+1} \odot \mathbf{A}^{k+1}) \widehat{\mathbf{Q}} \right\|_F^2. \tag{5.20}$$

Since the solution in (5.20) is not guaranteed to be orthogonal, we perform a QR factorization to $\mathbf{P}$ to obtain an orthogonal matrix $\mathbf{O}$. Let

$$\mathbf{Q}^{k+1} = \mathbf{O}. \tag{5.21}$$

where $\mathbf{P} = \mathbf{OR}$ and $\mathbf{R}$ is an upper triangular matrix. To solve equation (5.19), we apply the PCLS (5.9) to compute $\mathbf{A}$ column by column,

$$\mathbf{a}_r^{k+1} = \underset{\widehat{a}_r \in \mathbb{R}^I}{\operatorname{argmin}} \left\| unvec \left(\mathbf{E}(\mathbf{Q}^k)^T(:, r) \right) - \widehat{\mathbf{a}}_r \cdot \widehat{\mathbf{a}}_r^{\mathrm{T}} \right\|_F^2, r = 1, \ldots, R_s. \tag{5.22}$$

We summarize the iterative method via PCLS method for fourth-order fully symmetric tensor. Given the tensor $\mathcal{T} \in \mathbb{R}^{I \times I \times I \times I}$, we first calculate matrix $\mathbf{E} \in \mathbb{R}^{I^2 \times R_s}$ through $\mathbf{T}$, the matricization of $\mathcal{T}$. Then starting from the initial guesses, we fix $\mathbf{Q}$ to solve for each column $\mathbf{a}_r$ of $\mathbf{A}$, then $\mathbf{A}$ is fixed to compute a temporary matrix $\mathbf{P}$. In order to make sure the updated $\mathbf{Q}$ is orthogonal, we apply QR factorization on $\mathbf{P}$ to obtain an orthogonal matrix and set it to be the updated $\mathbf{Q}$. This process continues iteratively until the absolute residual $\|\mathcal{T} - \mathcal{T}_{est}\|_F$ has reached a set tolerance.

6. Numerical Examples.

In this section, we compare the performance of ALS against our iterative method via PCLS for the third-order partially symmetric tensors and the fourth-order fully symmetric tensors. From these numerical examples, our iterative method outperformed ALS in terms of the number of iterations until convergence and the CPU time. We prepared three types of examples: (1) third order partially symmetric tensor, (2) fourth-order fully symmetric tensor and (3) fourth-order cumulant tensor in blind source separation problem. In the all of the experiments, the stopping criterion is set at $\epsilon = 10^{-10}$ in $\|\mathcal{X} - \mathcal{X}_{est}\|_F^2 < \epsilon$.

We generated our tensor examples by randomly generating factor matrices which satisfy the symmetric constraints. For example, we create a third-order tensor $\mathcal{T} \in \mathbb{R}^{I \times I \times K}$ with partial symmetry $t_{ijk} = t_{jik}$ by randomly generating a matrices $\mathbf{A} \in \mathbb{R}^{I \times R}$ and $\mathbf{C} \in \mathbb{R}^{K \times R}$ with i.i.d. gaussian entries in

$$(\mathcal{T})_{ijk} = \sum_{r=1}^{R} (\mathbf{A})_{ir} (\mathbf{A})_{jr} (\mathbf{C})_{kr}.$$

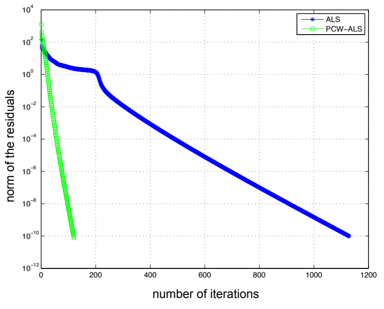
The random matrices are generated in matlab via $\mathbf{A} =$ `randn(I,R)` and $\mathbf{C} =$ `randn(K,R)`.

6.1. Example I: third-order partially symmetric tensor.

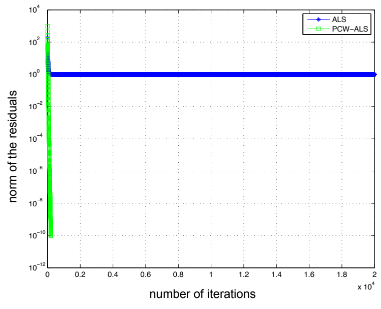
We generate a partially symmetric tensor $\mathcal{X} \in \mathbb{R}^{17 \times 17 \times 18}$ by random data, in which $x_{ijk} = x_{jik}$. In

Figure 6.1, we consider SOPD for $\mathcal{X}$ with $R_{ps} = 17$ with two different factor matrices $\mathbf{A} \in \mathbb{R}^{17\times 17}$ and $\mathbf{C} \in \mathbb{R}^{18\times 17}$ and the decomposition is

$$\mathcal{X} = \sum_{r=1}^{R_{ps}} \mathbf{a}_r \otimes \mathbf{a}_r \otimes \mathbf{c}_r.$$



(a) good initial guess



(b) random initial guess

Fig. 6.1: **Plots for the Example 6.1**

In the two plots, the stopping criteria for both ALS and our method are satisfied when the error $\|\mathcal{X} - \mathcal{X}_{est}\|_F^2$ is less than $\epsilon = 10^{-10}$ where $\mathcal{X}_{est}$ denotes the obtained tensor after every iteration. ALS needs three initial guesses, so we set $\mathbf{B}^0 = \mathbf{A}^0$.

In Figure 6.1a, both algorithms work well with a particular initial guesses, but our iterative method performed better than the ALS algorithm. It required only takes 120 iterations in comparison to that of 1129 ALS iterations. Moreover, our method is faster than ALS since the CPU time is 3.9919s while ALS took 6.4126s. Figure 6.1b shows that our method did not enter a swamp regime and converged after 205 iterations at an error within 10^{-10}. ALS did not converge after 20000 iterations with a constant error at 1.

Simulation. We tested the algorithms with 50 different random initial starters given the same tensor $\mathcal{X}$ set-up. Both the ALS and our iterative algorithms are used to decompose $\mathcal{X}$ with rank $R_{ps} = 17$. The average results in terms of the number of iterations and CPU time are shown in the Table 6.1.

	ALS	Iterative PCLS
average CPU time	17.1546s	6.1413s
average number of iterations	3445.0	258.7

Table 6.1: **ALS and Iterative PCLS: Mean of the CPU time and the number of iterations of** 50 **random initial starters**

CPU time vs tensor size. We apply the ALS and our iterative PCLS methods on the third-order partially symmetric tensors with varying sizes: $\mathcal{X}_1 \in \mathbb{R}^{10\times 10\times 10}$ with $R_{ps} = 10$, $\mathcal{X}_2 \in \mathbb{R}^{20\times 20\times 20}$ with $R_{ps} = 20$, ..., $\mathcal{X}_9 \in \mathbb{R}^{90\times 90\times 90}$ with $R_{ps} = 90$. We compare the CPU times of both methods for the same tensor size. For each tensor

$\mathcal{X}_i$, we calculated the mean average of the CPU times and the number of iterations from each method as in the previous experiments above. The following Figure 6.2 shows that as the tensor size increases, the rate of CPU time of the ALS is faster than the rate of the iterative PCLS's CPU time.

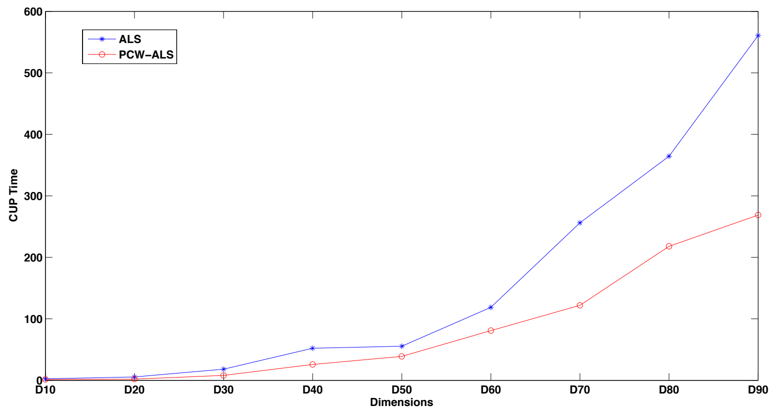


Fig. 6.2: **Plots of the mean of CPU times vs tensor dimensions**

6.2. Example II: fourth-order fully symmetric tensor. The first fully symmetric fourth-order tensor example is $\mathcal{X} \in \mathbb{R}^{10\times 10\times 10\times 10}$ with $R = 10$. With the given initial guess $\mathbf{A}^0$, both ALS and iterative PCLS are applied to solve the SOPD for this fourth-order tensor. The following Figure 6.3 shows that the swamp occurs in the ALS method while the iterative PCLS converges very fast.

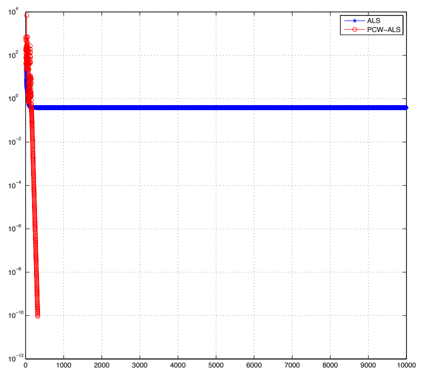


Fig. 6.3: **Plot for the Example 6.2 (first)**

The second example is a fully symmetric fourth-order tensor $\mathcal{X} \in \mathbb{R}^{15\times 15\times 15\times 15}$ with $R = 10$. Given with the initial guess $\mathbf{A}^0$, both ALS and iterative PCLS are applied to solve the SOPD for this fourth-order tensor. Figure 6.4 shows that both method works well. The CPU time of the ALS method is 27.2149s while the iterative PCLS method is 4.2763s. The iterative PCLS performed faster than ALS in terms of CPU time and the number of iterations.

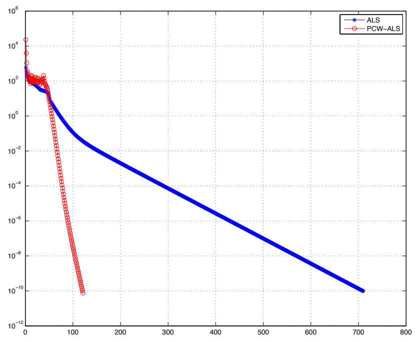


Fig. 6.4: **Plot for the Example 6.2 (second)**

6.3. Example III: blind source separation problem. From a mixture of signals $\mathbf{Z}(t)$ in Figure 6.5, we would like to recover the two original source signals $\mathbf{X}(t)$ [10],

$$x_1(t) = \sqrt{2}\sin t$$
$$x_2(t) = \begin{cases} 1 & \text{if } t \in [k\pi, k\pi + \frac{\pi}{2}),\ k \in \mathbb{Z} \\ -1 & \text{if } t \in [k\pi + \frac{\pi}{2}, (k+1)\pi),\ k \in \mathbb{Z}. \end{cases}$$

The goal is to find matrix $\mathbf{V}$ so that $\mathbf{V}\mathbf{Z}(t) = \mathbf{X}(t)$. The matrix $\mathbf{V} \in \mathbb{R}^{2\times 2}$ can be obtained from

$$\mathcal{C}_Z = \sum_{r=1}^{2} (\mathcal{C}_X)_{rrrr} \mathbf{v}_r \otimes \mathbf{v}_r \otimes \mathbf{v}_r \otimes \mathbf{v}_r \qquad (6.1)$$

where $\mathcal{C}_Z$ and $\mathcal{C}_X$ are fourth-order cumulant tensor of size $2\times 2\times 2\times 2$ w.r.t. $\mathbf{Z}$ and $\mathbf{X}$, respectively, and $\mathbf{v}_r$ is a column of $\mathbf{V}$. Note that $\mathcal{C}_X$ and $\mathbf{V}$ are the unknowns. The entries of $\mathcal{C}_Z$ are the following: $(\mathcal{C}_Z)_{1111} = -\frac{39}{32}$, $(\mathcal{C}_Z)_{1112} = (\mathcal{C}_Z)_{2111} = (\mathcal{C}_Z)_{1211} = (\mathcal{C}_Z)_{1121} = -\frac{9\sqrt{3}}{32}$, $(\mathcal{C}_Z)_{1122} = (\mathcal{C}_Z)_{2121} = (\mathcal{C}_Z)_{1221} = (\mathcal{C}_Z)_{2211} = (\mathcal{C}_Z)_{1212} = (\mathcal{C}_Z)_{1122} = -\frac{21}{32}$, $(\mathcal{C}_Z)_{1222} = (\mathcal{C}_Z)_{2122} = (\mathcal{C}_Z)_{2212} = (\mathcal{C}_Z)_{2221} = \frac{5\sqrt{3}}{32}$ and $(\mathcal{C}_Z)_{2222} = -\frac{31}{32}$. In Figure (6.5), we apply our iterative PCLS and ALS to find the decomposition (6.1). The iterative PCLS was able to find the factor matrix $\mathbf{V}$ to unmix the source signals. On the other hand, ALS converged but the factor matrix solution did not unmix the signals.

7. Conclusion. We presented an iterative algorithm which implements the partially column-wise least-squares (PCLS) for the SOPD for third-order partially symmetric tensors and fourth-order fully and partially symmetric tensors. PCLS is a column-wise approach for factoring symmetric Khatri-Rao product into two similar factor matrices. For symmetric third-order and fourth-order tensors, these symmetric

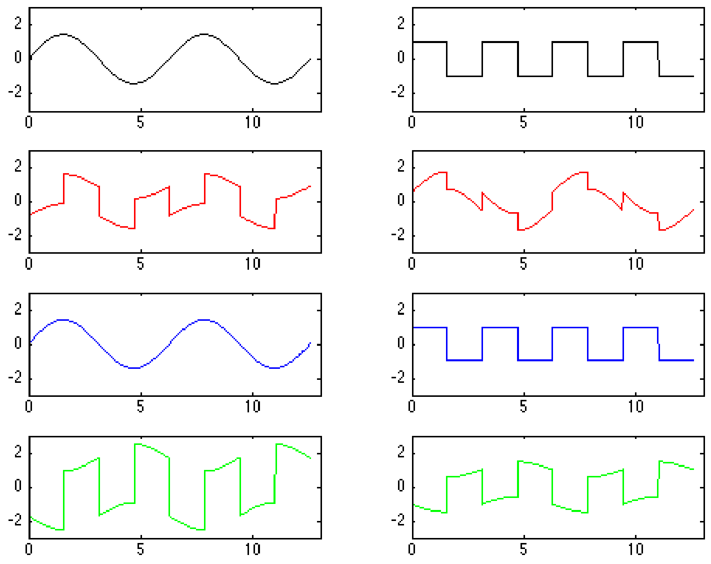


Fig. 6.5: **Top row: original source signals. Middle 2nd row: mixed signals. Middle 3rd row: source signals separated via PCLS. Bottom row: source signals separated via ALS.**

Khatri-Rao products are prevalent. With the PCLS method, the nonlinear least-squares subproblems which are present in the ALS formulation for symmetric tensors are avoided. We also provide several numerical examples to compare the performance of the iterative PCLS method to the ALS approach. In these examples, the swamps are not common when implementing our iterative PCLS as they are present in the examples where ALS was applied. Future work will focus on the generalization of SOPD to even-order and odd-order partially and fully symmetric tensors as well as increasing the speed and efficiency of the current methods.

Acknowledgements. C.N. and N.L. were both in part supported by the U.S. National Science Foundation DMS-0915100. C.G. is supported by the U.S. National Science Foundation HRD-325347.

REFERENCES

[1] E. Acar, C. A. Bingol, H. Bingol, R. Bro, and B. Yener, *Multiway analysis of epilepsy tensors*, Bioinformatics, **23** (13) (2007), pp. i10-i18.

[2] G. Beylkin and M.J. Mohlenkamp, *Algorithms for numerical analysis in high dimensions.* SIAM Journal on Scientific Computing, **26** (2005), 2133-2159.

[3] J. Brachat, P. Comon, B. Mourrain and E. Tsigaridas, *Symmetric tensor decomposition.* Linear Algebra and its Applications, **433**(11) (2010), 1851-1872, 2010.

[4] M. Brazell, N. Li, C. Navasca and C. Tamon, *Solving Multilinear Systems via Tensor Inversion.* SIAM Journal on Matrix Analysis and Applications, **34**(2) (2013), 542-570.

[5] J. Carrol and J. Chang, *Analysis of Individual Differences in Multidimensional Scaling via an N-way Generalization of "Eckart-Young" Decomposition.* Psychometrika, 9 (1970),

267-283.

[6] P. Comon, G. Golub, L-H. Lim and B. Mourrain, *Symmetric tensors and symmetric tensor rank.* SIAM Journal on Matrix Analysis and Applications, **30**(3) (2008), 1254-1279.

[7] P. Comon, *Independent component analysis, a new concept?* Signal processing, **36**(3) (1994), 287-314.

[8] P. Comon and C. Jutten, *Handbook of Blind Source Separation: Independent component analysis and applications.* Academic press, 2010.

[9] P. Comon, X. Luciani and A.L.F. De Almeida. *Tensor Decompositions, Alternating Least Squares and other Tales.* Journal of Chemometrics, **23** (2009) 393-405.

[10] L. De Lathauwer, B. De Moor and J. Vandewalle, *An introduction to independent component analysis.* Journal of Chemometrics, **14** (2000), 123-149.

[11] L. De Lathauwer, B. De Moor and J. Vandewalle, *A multilinear singular value decomposition.* SIAM journal on Matrix Analysis and Applications, **21**(4) (2000), 1253-1278.

[12] L. De Lathauwer, J. Castaing and J-F. Cardoso, *Fourth-order cumulant-based blind identification of underdetermined mixtures.* IEEE Transactions of Signal Processing, **55**(6), June 2007.

[13] M. De Vos, A. Vergult, L. De Lathauwer, W. De Clercq, S. Van Huffel, P. Dupont, A. Palmini, and W. Van Paesschen, *Canonical decomposition of ictal EEG reliably detects the seizure onset zone.* Neuroimage, **37**(3) (2007), 844-854.

[14] R. A. Harshman, *Foundations of the PARAFAC procedure: Model and Conditions for an "Explanatory" Multi-code Factor Analysis.* UCLA Working Papers in Phonetics, **16** (1970), 1-84.

[15] F.L. Hitchcock, *The expression of a tensor or a polyadic as a sum of products.* Journal of Mathematics and Physics, **6** (1927), 164-189.

[16] F.L. Hitchcock, *Multilple invariants and generalized rank of a p-way matrix or tensor.* Journal of Mathematics and Physics, **7** (1927), 39-79.

[17] A. Hyvarinen, J. Karhunen and E. Oja, *Independent component analysis.* Studies in Informatics and Control, **11**(2) (2002), 205-207.

[18] P.M. Kroonenberg, *Applied Multiway Data Analysis.* Wiley, 2008.

[19] J. B. Kruskal, *Three-way arrays: rank and uniqueness or trilinear decompositions, with application to arithmetic complexity and statistics.* Linear algebra and its applications, **18**(2) (1977), 95-138.

[20] J.M. Landsberg, *Tensors: Geometry and Applications* AMS, Providence, Rhode Island, 2010.

[21] N. Li, S. Kindermann and C. Navasca, *Some Convergent Results of the Regularized Alternating LeastSquares for Tensor Decomposition.* Linear Algebra and Applications, **438** (2) (2013), 796-812.

[22] C. Navasca, L. De Lathauwer and S. Kindermann, *Swamp reducing technique for tensor decomposition.* Proceedings of the European Signal Processing Conference, Lausanne, August 2008.

[23] M. Rajih, P. Comon and R. Harshman, *Enchanced Line Search: A Novel Method to Accelerate PARAFAC* SIMAX, **30** (3) (2008), pp. 1148-1171

[24] C.R. Rao and S.K. Mitra, *Generalized Inverse of Matrices and Its Applications.* Wiley, New York, 1971.

[25] T. Schultz and H.P. Seidel, *Estimating Crossing Fibers: A Tensor Decomposition Approach.* IEEE Transactions on Visualization and Computer Graphics, **14**(6) (2008), 1635-1642.

[26] N.D. Sidiropoulos, G.B. Giannakis, and R. Bro, *Blind PARAFAC receivers for DS-CDMA systems.* IEEE Trans. on Signal Processing, **48** (3) (2000), 810-823.

[27] N. Sidiropoulos, R. Bro, and G. Giannakis, *Parallel factor analysis in sensor array processing.* IEEE Trans. Signal Processing, **48** (2000), 2377-2388.

[28] A. Smilde, R. Bro, and P. Geladi, *Multi-way Analysis. Applications in the Chemical Sciences.* Chichester, U.K., John Wiley and Sons, 2004.

[29] A. Stegeman, *On Uniqueness of The Canonical tensor Decomposition with Some Form of Symmetry.* SIAM J. Matrix Anal. Appl., **32**(2) (2011), 561-583.

[30] P. Paatero, *Construction and analysis of degenerate Parafac models.* J. Chemometrics, **14**, (2000) pp. 285-299.