Electronic Transactions on Numerical Analysis.
Volume 44, pp. 124–139, 2015.

ISSN 1068–9613.

ETNA
Kent State University
http://etna.math.kent.edu

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 decompositions of a fully (partially) symmetric tensor into a sum of rank-one fully (partially) symmetric tensors. We present an iterative technique for third-order partially symmetric tensors and fourth-order fully and partially symmetric tensors. We include several numerical examples which indicate faster convergence for the new algorithms than for 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 [13, 14] 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 [1, 11, 16, 24, 25, 26]. Specifically, symmetric tensors are ubiquitous in many signal processing applications [6, 7, 9]. 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 a given symmetric tensor by a sum of rank-one symmetric tensors.

SOPD is common in independent component analysis (ICA) [5, 15] or blind source separation (BSS), which is used to separate the true signal from noise and interference in signal processing [7, 9]. When the order of the tensor is three and the tensor is symmetric in two modal dimensions, this is called the individual differences scaling (INDSCAL) model introduced by Carrol and Chang [4, 27].

There are very few numerical methods for finding SOPDs. For unsymmetric tensors, a well-known method for finding the sum of rank-one terms is the alternating least squares (ALS) technique [4, 12]. Since SOPD is a special case of the CP decomposition, the ALS method can be applied in this situation. 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 [2] for SOPDs reduces the problem to the decomposition of a linear form. For fourth-order fully symmetric tensors, De Lathauwer in [9] proposed the fourth-order-only blind identification (FOOBI) algorithm. Schultz [23] numerically solves SOPD problems using the best symmetric rank-one approximation of a symmetric tensor through the maximum of the associated homogeneous form over the unit sphere. In this paper, we study SOPDs for third-order partially symmetric tensors and fourth-order fully and partially symmetric tensors and propose a new method called partial column-wise least squares (PCLS) to compute the SOPD. It obviates the nonlinear least-squares subproblems through some tensor unfoldings and a root finding technique for polynomials in estimating the factor matrices.

*Received January 23, 2014. Accepted January 12, 2015. Published online on February 11, 2015. Recommended by S. Kindermann.
†Department of Mathematics, Clarkson University, Potsdam, NY 13699 (nalitensor@gmail.com).
‡Department of Mathematics, University of Alabama at Birmingham, 1300 University Boulevard, Birmingham, AL, 35294 (cnavasca@uab.edu, tina_creations@hotmail.com).

ETNA
Kent State University
http://etna.math.kent.edu

2. Preliminaries. Throughout this paper, a tensor is understood as a multidimensional array, i.e., an element of $\mathbb{R}^{I_1\times I_2\times\cdots\times I_N}$, $I_i\in\mathbb{N}$, $i=1,\ldots,N$. The number I_i is the i-th modal dimension, and the integer N is called the order of the tensor. We denote scalars in $\mathbb{R}$ by lower-case letters $(a,b,\ldots)$ and vectors by bold lower-case letters $(\mathbf{a},\mathbf{b},\ldots)$. Matrices are written as bold upper-case letters $(\mathbf{A},\mathbf{B},\ldots)$, and the symbols for tensors are calligraphic letters $(\mathcal{A},\mathcal{B},\ldots)$. 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 designated as $\mathbf{a}_r$.

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 is obtained by arranging the mode-n fibers as 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 $\mathcal{X}$ (using Matlab-type colon notation):

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

respectively. These matricizations can be attained in Matlab by these commands:

$$\begin{aligned}
\mathbf{X}_{(1)} &= \texttt{reshape(X, I, J*K)},\\
\mathbf{X}_{(2)} &= \texttt{reshape(permute(X, [2 1 3]), J, K*I)}, \qquad \text{and}\\
\mathbf{X}_{(3)} &= \texttt{reshape(permute(X, [3 2 1]), K, J*I)}.
\end{aligned}$$

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 $\mathrm{mat}(\mathcal{T})\in\mathbb{R}^{IJ\times KL}$ and is defined as*

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

See [3] for a generalization of square matricization in terms of tensor blocks. In Matlab, square matricization is obtained by the command $\mathbf{T}=$ `reshape(T, I*J, K*L)`.

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

$$\mathbf{w}_{ij}=v_{(j-1)I+i}, \qquad i,j=1,\ldots,I,$$

and

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

3. The 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*

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

If $\mathbf{x} = \mathbf{y}$, then we observe 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:

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

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 that $\mathcal{T}$ is a symmetric third-order rank-one tensor. We say that $\mathcal{T}$ 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.,*

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

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 $1 \le r \le k$.

DEFINITION 3.3 (Partially symmetric rank-one tensor). *A kth-order rank-one 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 $1 \le l, m \le k$ and $l \ne m$ in*

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

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

Furthermore, the modal indices corresponding to symmetry can be arranged into equivalence classes forming disjoint subsets S_i of subindices, where $\bigcup_{i=1}^{k} S_i = \{1, 2, \ldots, k\}$ and $\bigcap_{i=1}^{k} S_i = \emptyset$.

REMARK 3.4. If a third-order tensor $\mathcal{T}$ is a 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 3.5 (Symmetric rank-one tensor). *A kth-order rank-one 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 $\mathbf{a} \in \mathbb{R}^I$.

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

REMARK 3.6. We say that a tensor is cubical if all modal dimensions are equal. Symmetric tensors are cubical. A fully symmetric tensor is invariant under all permutations of its indices. Let a 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

$$\mathcal{T}_{i_1, i_2, \ldots, i_k} = \mathcal{T}_{i_{m(1)} i_{m(2)} \ldots i_{m(k)}}$$

for all permutations σ of the indices $(i_1, i_2, \ldots, i_k)$.

ETNA
Kent State University
http://etna.math.kent.edu

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

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

This is called the canonical polyadic (CP) decomposition (also known as PARAFAC or CANDECOM). This decomposition first appeared in the papers of Hitchcock [13, 14]. The notion of tensor rank was also introduced by Hitchcock.

DEFINITION 3.7. *The rank of* $\mathcal{T} \in \mathbb{R}^{I_1 \times \cdots \times I_k}$ *is defined as*

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

Define $\mathsf{T}^k(\mathbb{R}^n)$ as the set of all order-k cubical tensors having modal dimensions equal to n: $I_i = n$, $i = 1, \ldots, k$. The set of symmetric tensors in $\mathsf{T}(\mathbb{R}^n)$ is denoted as $\mathsf{S}^k(\mathbb{R}^n)$.

DEFINITION 3.8. *If* $\mathcal{T} \in \mathsf{S}^k(\mathbb{R}^n)$*, then the rank of a symmetric tensor* $\mathcal{T} \in \mathbb{R}^{I_1 \times \cdots \times I_k}$ *is defined as*

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

LEMMA 3.9 ([6]). *Let* $\mathcal{T} \in \mathsf{S}^k(\mathbb{R}^n)$ *have* $\operatorname{rank}_{\mathsf{S}}(\mathcal{T}) = S$*. Then there exist linearly independent vectors* $\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_S \in \mathbb{R}^n$ *such that*

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

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, modal dimension-n, cubical tensors $\mathsf{T}^k(\mathbb{R}^n)$ and $R_{\mathsf{S}}(k,n)$ is the maximally attainable symmetric rank in the space of symmetric tensors $\mathsf{S}^k(\mathbb{R}^n)$. In [6, 17], numerous results on the symmetric rank over $\mathbb{C}$ can be found. For example in [6], for all $\mathcal{T}$, it holds that

$$\operatorname{rank}_{\mathsf{S}}(\mathcal{T}) \leq \binom{n+k-1}{k},$$
$$\operatorname{rank}(\mathcal{T}) \leq \operatorname{rank}_{\mathsf{S}}(\mathcal{T}).$$

We also refer the readers to the book by Landsberg [17] for discussions on the partially symmetric tensor rank and to the work of Stegeman [27] for uniqueness conditions for the minimum rank of the symmetric outer product.

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

$$\min_{\widetilde{\mathcal{T}}} \|\mathcal{T} - \widetilde{\mathcal{T}}\|_F^2,$$

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 procedure for tackling 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 a third-order tensor is to solve

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

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

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:

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

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 the tensor $\mathcal{T}$. The symbol $\odot$ denotes the Khatri-Rao product [22]. 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.,

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

By fixing two factor matrices alternately at each iteration, three coupled linear least squares subproblems are then formulated to find each factor matrix:

$$\begin{aligned} \mathbf{A}^{k+1} &= \underset{\widehat{\mathbf{A}}\in\mathbb{R}^{I\times R}}{\operatorname{argmin}} \left\| {\mathbf{T}_{(\mathbf{1})}}^{I\times JK} - \widehat{\mathbf{A}}(\mathbf{C}^k\odot\mathbf{B}^k)^T \right\|_F^2, \\ \mathbf{B}^{k+1} &= \underset{\widehat{\mathbf{B}}\in\mathbb{R}^{J\times R}}{\operatorname{argmin}} \left\| {\mathbf{T}_{(\mathbf{2})}}^{J\times IK} - \widehat{\mathbf{B}}(\mathbf{C}^k\odot\mathbf{A}^{k+1})^T \right\|_F^2, \\ \mathbf{C}^{k+1} &= \underset{\widehat{\mathbf{C}}\in\mathbb{R}^{K\times R}}{\operatorname{argmin}} \left\| {\mathbf{T}_{(\mathbf{3})}}^{K\times IJ} - \widehat{\mathbf{C}}(\mathbf{B}^{k+1}\odot\mathbf{A}^{k+1})^T \right\|_F^2, \end{aligned} \tag{4.1}$$

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

Although ALS has been applied extensively across engineering and science disciplines, it has some disadvantages. For non-degenerate problems, it may require a high number of iterations until a stopping criterion is satisfied (see the ”swamp“ in Figure 4.1), which can

ETNA
Kent State University
http://etna.math.kent.edu

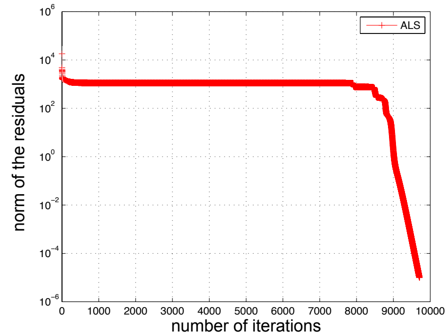

FIG. 4.1. *The long flat curve (*swamp*) in the ALS method. The error stays at* 10^3 *during the first 8000 iterations.*

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., [8, 20, 21]. A long flat 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 (4.1). 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 leads to non-unique solutions. Regularization methods [18, 19] do not drastically mitigate the requirement for a high number of iterations.

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

- Problem 1: find a/the best minimum-rank sum of rank-one *symmetric* tensor
$$\min_{\widetilde{\mathcal{T}}} \|\mathcal{T} - \widetilde{\mathcal{T}}\|_F^2,$$
where $\widetilde{\mathcal{T}} = \sum_{r=1}^{R} \mathbf{a}_r \otimes \mathbf{a}_r \otimes \cdots \otimes \mathbf{a}_r$,
- Problem 2: find a/the best minimum-rank sum of rank-one *partially symmetric* tensor
$$\min_{\widetilde{\mathcal{T}}} \|\mathcal{T} - \widetilde{\mathcal{T}}\|_F^2,$$
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 $\mathbf{a}_r^{(j)} = \mathbf{a}_r^{(l)}$ with $1 \leq j$, $l \leq k$, and $j \neq l$.

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

We describe the decomposition methods for third-order and fourth-order tensors with partial and full symmetries. Later, we outline 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 $t_{ijk} = t_{jik}$. Then 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

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

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

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

If we employ the ALS iteration, we are faced with the following subproblems:

$$\begin{aligned} \mathbf{A}^{k+1} &= \operatorname*{argmin}_{\widehat{\mathbf{A}} \in \mathbb{R}^{I \times R_{ps}}} \left\| \mathbf{T}_{(\mathbf{3})} - \mathbf{C}^k (\widehat{\mathbf{A}} \odot \widehat{\mathbf{A}})^T \right\|_F^2 , \\ \mathbf{C}^{k+1} &= \operatorname*{argmin}_{\widehat{\mathbf{C}} \in \mathbb{R}^{K \times R_{ps}}} \left\| \mathbf{T}_{(\mathbf{3})} - \widehat{\mathbf{C}} (\mathbf{A}^{k+1} \odot \mathbf{A}^{k+1})^T \right\|_F^2 . \end{aligned} \tag{5.2}$$

Directly applying the ALS method to (5.1) does not work. For symmetric problems, at least one of the subproblems is a nonlinear least squares problem, e.g., here (5.2). The ALS approach leads to factor matrices that do not satisfy tensor symmetries, and/or it needs a high number of iterations (swamps can appear), if the procedure 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 $\mathbf{C}$ can be handled by linear least squares methods. Recall that $\mathbf{T}_{(\mathbf{3})} = \mathbf{C}^k(\widehat{\mathbf{A}} \odot \widehat{\mathbf{A}})^T$ can be solved for $\widehat{\mathbf{A}} \odot \widehat{\mathbf{A}}$, i.e.,

$$\widehat{\mathbf{A}} \odot \widehat{\mathbf{A}} = ((\mathbf{C}^k)^\dagger \mathbf{T}_{(\mathbf{3})})^T, \tag{5.3}$$

where $(\cdot)^\dagger$ denotes the Moore-Penrose pseudoinverse. Equivalently, (5.3) can be written as

$$\widehat{\mathbf{a}}_r \otimes \widehat{\mathbf{a}}_r = ((\mathbf{C}^k)^\dagger \mathbf{T}_{(\mathbf{3})})^T(:,r) \quad \Leftrightarrow \quad \widehat{\mathbf{a}}_r \cdot \widehat{\mathbf{a}}_r^T = \operatorname{unvec}\left(((\mathbf{C}^k)^\dagger \mathbf{T}_{(\mathbf{3})})^T(:,r) \right), \tag{5.4}$$

where $r = 1, \ldots, R_{ps}$, $\widehat{\mathbf{a}}_r$ is the rth column of the matrix $\widehat{\mathbf{A}}$, and $\operatorname{unvec}\left(((\mathbf{C}^k)^\dagger \mathbf{T}_{(\mathbf{3})})^T(:,r) \right)$ is a matrix of size $I \times I$ obtained from the vector $((\mathbf{C}^k)^\dagger \mathbf{T}_{(\mathbf{3})})^T(:,r)$ via column vector stacking of size I. With (5.4), we can obtain $\widehat{\mathbf{A}}$ by calculating each of its column $\widehat{\mathbf{a}}_r$ at a time. We call this approach the partial column-wise least squares method (PCLS), a Cholesky-like factorization for a symmetric Khatri-Rao product.

In detail, let $\mathbf{x} \in \mathbb{R}^I = [x_1\ x_2\ \cdots\ x_I]^T$ denote the unknown vector $\widehat{\mathbf{a}}_r$, and let $\mathbf{Y} = \operatorname{unvec}\left(((\mathbf{C}^k)^\dagger \mathbf{T}_{(\mathbf{3})})^T(:,r) \right) \in \mathbb{R}^{I \times I}$. Then (5.4) 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

$$x_1^* = \operatorname*{argmin}_{x_1} \ (y_{11} - x_1^2)^2 + \sum_{i=2}^{I} \left[(y_{i1} - x_i x_1)^2 + (y_{1i} - x_i x_1)^2 \right]. \tag{5.5}$$

ETNA
Kent State University
http://etna.math.kent.edu

This cost function in (5.5) is a fourth-order polynomial in one variable x_1, and each component x_i is obtained in the same manner by minimizing a fourth-order polynomial.

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

$$(x_m^*)^{k+1} = \underset{x_m}{\operatorname{argmin}} \; (y_{mm} - (x_m^k)^2)^2 + \sum_{\substack{i=1 \\ i \neq m}}^{I} \left[(y_{im} - x_i^k x_m^k)^2 + (y_{mi} - x_i^k x_m^k)^2 \right].$$

Thus, in each iteration we have to solve a system of fourth-order optimization problems as outlined in Algorithm 5.1 below.

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 of the numerical examples discussed in Section 6. It is fast and more reliable than implementing singular value/eigenvalue decompositions (SVD/EVD)s. The SVD/EVD approximations often lead to a high number of iterations.

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

$$\text{(5.6)} \quad \mathbf{a}_r^{k+1} = \underset{\widehat{a}_r \in \mathbb{R}^I}{\operatorname{argmin}} \left\| \operatorname{unvec} \left((((\mathbf{C}^k)^\dagger \mathbf{T}_{(\mathbf{3})})^T(:, r) \right) - \widehat{\mathbf{a}}_r \cdot \widehat{\mathbf{a}}_r^T \right\|_F^2, \qquad r = 1, \dots, R_{ps},$$

and

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

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 Algorithm 5.1. Then in the second subproblem, we fixed $\mathbf{A}$ to solve for $\mathbf{C}$. This process continues iteratively until some convergence criterion, based on upper bounds for the residual and the maximal number of iterations, is satisfied.

ALGORITHM 5.1 (Partial Column-wise Least-Squares Method (PCLS)).
Find $\mathbf{A}^* = \operatorname{argmin}_{\mathbf{A}} \|\mathbf{T} - \mathbf{C}(\mathbf{A} \odot \mathbf{A})^T\|_F^2$

%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: $\mathbf{a}_r \otimes \mathbf{a}_r = \mathbf{Y}(:, r) \to \mathbf{a}_r \cdot \mathbf{a}_r^T = \operatorname{unvec}(\mathbf{Y}(:, r))$
  %Solve $\mathbf{a}_r^{k+1} = \operatorname{argmin}_{\widehat{a}_r \in \mathbb{R}^I} \|\operatorname{unvec}(\mathbf{Y})(:, r)) - \mathbf{a}_r \cdot \mathbf{a}_r^T\|_F^2$
    FOR m=1:I
      $$(a_r)_m^* = \operatorname{argmin}_{(a_r)_m} \; (y_{mm} - ((a_r)_m)^2)^2 + \sum_{i=1, i \neq m}^{I} \big[(y_{im} - (a_r)_i (a_r)_m)^2 + (y_{mi} - x_i (a_r)_m)^2 \big]$$
    END
END

If the given tensors exhibits symmetries in other modes, the procedure is analogous. For the cases $t_{ijk} = t_{ikj}$ ($\mathbf{B} = \mathbf{C}$) and $t_{ijk} = t_{jki}$ ($\mathbf{A} = \mathbf{C}$), the optimization problems are

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

ETNA
Kent State University
http://etna.math.kent.edu

and

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

respectively. Here are the corresponding subproblems:

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

and

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

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 is also not guaranteed that the solution satisfies the symmetries. The ALS method solves three linear least squares problems in each iteration, while PCLS solves in each iteration one linear least squares problem and minimizes R_{ps} quartic polynomials. The latter is equivalent to finding the roots of cubic polynomials. The operational cost of running PCLS on a third-order tensor is less than the requirement of ALS since in each iteration only one linear least squares problem is solved 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 multiplication for n-digit precision.

5.2. SOPD for fourth-order partially symmetric tensors. We can apply PCLS on a fourth-order partial symmetric tensor. Here we consider the following cases.

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 $x_{ijkl} = x_{jikl}$ and $x_{ijkl} = x_{ijlk}$. With the given symmetries, the unknown factor matrices are reduced to two, $\mathbf{A}$ and $\mathbf{C}$, since $\mathbf{A} = \mathbf{B}$ and $\mathbf{C} = \mathbf{D}$. Then, the task is to find factor matrices $\mathbf{A}$ and $\mathbf{C}$ solving the following minimization problem:

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

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 square matricization, we obtain

$$\operatorname{mat}(\mathcal{X}) = (\mathbf{A} \odot \mathbf{A})(\mathbf{C} \odot \mathbf{C})^T. \tag{5.7}$$

ETNA
Kent State University
http://etna.math.kent.edu

To solve equation (5.7) for $\mathbf{A}$ and $\mathbf{C}$, we apply PCLS with respect to the optimality conditions

$$\mathbf{a}_r \otimes \mathbf{a}_r = \mathrm{mat}(\mathcal{X})((\mathbf{C} \odot \mathbf{C})^T)^\dagger(:,r), \qquad r = 1,\dots,R,$$
$$\mathbf{c}_r \otimes \mathbf{c}_r = \mathrm{mat}(\mathcal{X})^T((\mathbf{A} \odot \mathbf{A})^T)^\dagger(:,r), \qquad r = 1,\dots,R,$$

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 task is to find factor matrices $\mathbf{A}$ and $\mathbf{B}$ solving

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

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 matricizing, 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 square matricization, we obtain

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

which leads to

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

Similarly, in case of 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,

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

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 1 and mode 3. 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 matricizing, 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 square matricization, we obtain

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

From this matricized equation, we get two optimality conditions:

$$\mathbf{a}_r \otimes \mathbf{a}_r = \mathrm{mat}(\mathcal{X})((\mathbf{B} \odot \mathbf{B})^T)^\dagger(:,r), \qquad r = 1,\dots,R, \tag{5.8}$$

and

$$\mathbf{b}_r \otimes \mathbf{c}_r = \mathrm{mat}(\mathcal{X})^T((\mathbf{A} \odot \mathbf{A})^T)^{\dagger}(:,r), \qquad r = 1, \ldots, R. \tag{5.9}$$

One can apply PCLS to extract $\mathbf{a}_r$ from the symmetric Khatri-Rao product (5.8). A rank-one SVD can be applied to find the decomposition of the asymmetric Khatri-Rao product (5.9), while a rank-one EVD may be used for (5.8). 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 $t_{ijkl} = t_{\sigma(i,j,k,l)}$ for any permutation σ of the indices (i, j, k, l). We want to a find a factor matrix $\mathbf{A} \in \mathbb{R}^{I \times R_s}$ solving

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

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

By using square matricization, we have

$$\mathbf{T} = (\mathbf{A} \odot \mathbf{A})(\mathbf{A} \odot \mathbf{A})^T. \tag{5.10}$$

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

$$\mathbf{T} = \mathbf{E}\mathbf{E}^T. \tag{5.11}$$

Comparing equations (5.10) and (5.11), we know that there exists an orthogonal matrix $\mathbf{Q}$ such that

$$\mathbf{E} = (\mathbf{A} \odot \mathbf{A})\mathbf{Q}, \tag{5.12}$$

where $\mathbf{Q} \in \mathbb{R}^{R_s \times R_s}$. In equation (5.12), 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 according to the following subproblems:

$$\mathbf{A}^{k+1} = \underset{\widehat{\mathbf{A}} \in \mathbb{R}^{I \times R_s}}{\operatorname{argmin}} \left\| \mathbf{E} - (\widehat{\mathbf{A}} \odot \widehat{\mathbf{A}})\mathbf{Q}^k \right\|_F^2, \tag{5.13}$$

and

$$\mathbf{P} = \underset{\widehat{\mathbf{Q}} \in \mathbb{R}^{R_s \times R_s}}{\operatorname{argmin}} \left\| \mathbf{E} - (\mathbf{A}^{k+1} \odot \mathbf{A}^{k+1})\widehat{\mathbf{Q}} \right\|_F^2. \tag{5.14}$$

Since the solution in (5.14) is not guaranteed to be orthogonal, we perform a QR factorization of $\mathbf{P}$ to obtain an orthogonal matrix $\mathbf{O}$. Let

$$\mathbf{Q}^{k+1} = \mathbf{O},$$

where $\mathbf{P} = \mathbf{O}\mathbf{R}$ and $\mathbf{R}$ is an upper triangular matrix. To solve equation (5.13), we apply PCLS (5.6) to compute $\mathbf{A}$ column by column,

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

We summarize the iterative method via PCLS for a fourth-order fully symmetric tensor. Given the tensor $\mathcal{T} \in \mathbb{R}^{I\times I\times I\times I}$, we first calculate the 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 that the updated $\mathbf{Q}$ is orthogonal, we apply a QR factorization of $\mathbf{P}$ to obtain an orthogonal matrix and set it to be the updated $\mathbf{Q}$. This process continues iteratively until the absolute residual $\|\mathcal{T} - \mathcal{T}_{est}\|_F$ drops below a given tolerance.

6. Numerical examples. In this section, we compare the performance of ALS against our iterative method via PCLS for third-order partially symmetric tensors and fourth-order fully symmetric tensors. According to 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: I) third-order partially symmetric tensors, II) fourth-order fully symmetric tensors, and III) a fourth-order cumulant tensor in a blind source separation problem. In the all the experiments, the iteration is stopped using the tolerance $\epsilon = 10^{-10}$ for the absolute residual in the criterion $\|\mathcal{X} - \mathcal{X}_{est}\|_F^2 < \epsilon$.

We generate our tensor examples satisfying the symmetric constraints by randomly generating factor matrices accordingly. 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 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 tensors. We generate a partially symmetric tensor $\mathcal{X} \in \mathbb{R}^{17\times 17\times 18}$ by random data for which $x_{ijk} = x_{jik}$. For the results in Figures 6.1–6.2, we consider a 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

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

For the computations, we used the stopping criterion mentioned above with $\epsilon = 10^{-10}$. Moreover, ALS needs three initial guesses, so we set $\mathbf{B}^0 = \mathbf{A}^0$.

Figure 6.1 indicates that both algorithms work well with a particular initial guesses, but our iterative method performs better than the ALS algorithm. It takes only 120 iterations in comparison to the 1129 ALS iterations. Moreover, our method is faster than ALS since the CPU time is 3.9919s while ALS took 6.4126s. Figure 6.2 shows that our method did not enter a swamp regime and converged after 205 iterations with a residual less than 10^{-10}. ALS did not converge after 20000 iterations saturating at a constant residual of the order of 1.

We furthermore tested the algorithms with 50 different random initial starters given the same tensor $\mathcal{X}$, performing a decomposition with a rank $R_{ps} = 17$. The average results in terms of the number of iterations and CPU time are shown in Table 6.1.

Considering the performance of the algorithms with respect to the number of variables, we apply the ALS method and our iterative PCLS method to third-order partially symmetric tensors of varying sizes as follows: $\mathcal{X}_1 \in \mathbb{R}^{10\times 10\times 10}$ with $R_{ps} = 10$, $\mathcal{X}_2 \in \mathbb{R}^{20\times 20\times 20}$ with $R_{ps} = 20$, and continuing this pattern up to $\mathcal{X}_9 \in \mathbb{R}^{90\times 90\times 90}$ 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

ETNA
Kent State University
http://etna.math.kent.edu

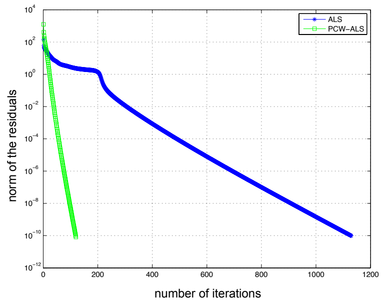

FIG. 6.1. *Example I: good initial guess.*

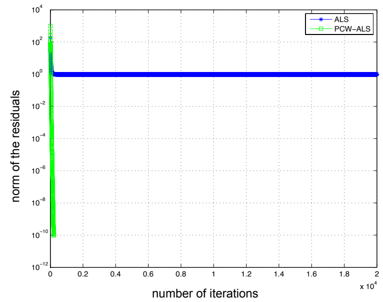

FIG. 6.2. *Example I: random initial guess.*

TABLE 6.1
ALS and iterative PCLS: mean of the CPU time and the number of iterations of 50 *random initial starters.*

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

the mean average of the CPU times and the number of iterations for each method as in the previous experiments. Figure 6.3 shows that as the tensor size increases, the average CPU time of ALS grows faster than that of PCLS as the dimension increases.

6.2. Example II: fourth-order fully symmetric tensor. For this example, the first test case corresponds to a given fully symmetric fourth-order tensor $\mathcal{X} \in \mathbb{R}^{10\times 10\times 10\times 10}$ 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 graphs in Figure 6.4 indicate that swamps occur for the ALS method, while the iterative PCLS converges very fast.

In a second test case, a fully symmetric fourth-order tensor $\mathcal{X} \in \mathbb{R}^{15\times 15\times 15\times 15}$ with $R = 10$ is used. Given the initial guess $\mathbf{A}^0$, both ALS and iterative PCLS are applied to solve the SOPD for this fourth-order tensor. Figure 6.5 shows that both method works well. The CPU time of the ALS method is 27.2149s while that of the iterative PCLS method is 4.2763s. The iterative PCLS is superior to ALS both in terms of CPU time and the number of iterations.

6.3. Example III: blind source separation problem. From a given mixture of signals $\mathbf{Z}(t)$ displayed in Figure 6.6, we would like to recover the two original source signals $\mathbf{X}(t)$ (cf. [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 a matrix $\mathbf{V}$ so that $\mathbf{VZ}(t) = \mathbf{X}(t)$. The matrix $\mathbf{V} \in \mathbb{R}^{2\times 2}$ can be obtained from

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

where $\mathcal{C}_Z$ and $\mathcal{C}_X$ are fourth-order cumulant tensors of size $2 \times 2 \times 2 \times 2$ with respect to $\mathbf{Z}$ 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

ETNA
Kent State University
http://etna.math.kent.edu

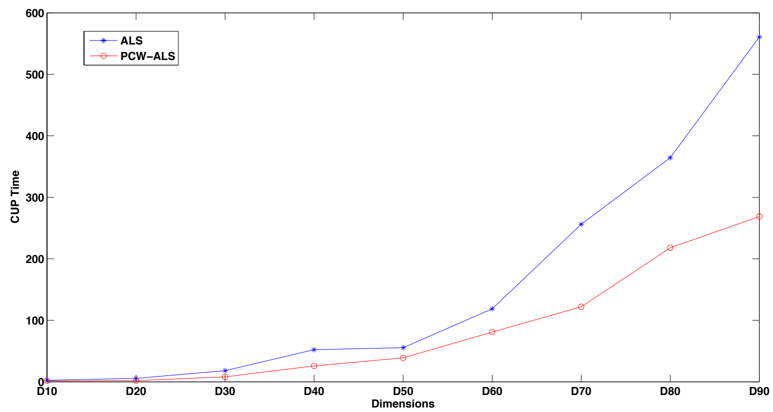

FIG. 6.3. *Graph of the mean CPU times versus tensor dimensions.*

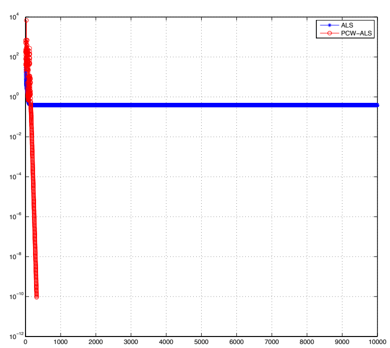

FIG. 6.4. *Example II (dimension $n = 10$).*

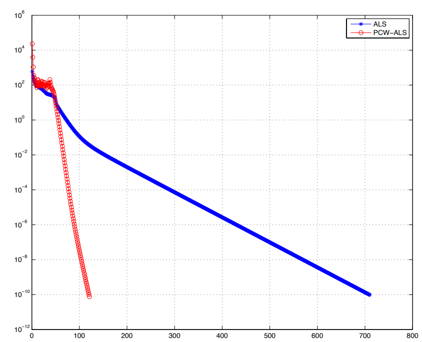

FIG. 6.5. *Example II (dimension $n = 15$).*

entries of $\mathcal{C}_Z$ are as follows:

$$(\mathcal{C}_Z)_{1111} = -\frac{39}{32},$$

$$(\mathcal{C}_Z)_{1112} = (\mathcal{C}_Z)_{2111} = (\mathcal{C}_Z)_{1211} = (\mathcal{C}_Z)_{1121} = -\frac{9\sqrt{3}}{32},$$

$$(\mathcal{C}_Z)_{1122} = (\mathcal{C}_Z)_{2121} = (\mathcal{C}_Z)_{1221} = (\mathcal{C}_Z)_{2211} = (\mathcal{C}_Z)_{1212} = (\mathcal{C}_Z)_{1122} = -\frac{21}{32},$$

$$(\mathcal{C}_Z)_{1222} = (\mathcal{C}_Z)_{2122} = (\mathcal{C}_Z)_{2212} = (\mathcal{C}_Z)_{2221} = \frac{5\sqrt{3}}{32},$$

$$(\mathcal{C}_Z)_{2222} = -\frac{31}{32}.$$

In Figure 6.6 we display the results when applying our iterative PCLS and ALS to find the decomposition (6.1). Using the iterative PCLS, we were able to find the factor matrix $\mathbf{V}$ and to unmix the source signals, in contrast to ALS, which converged, but the factor matrix solution did not unmix the signals.

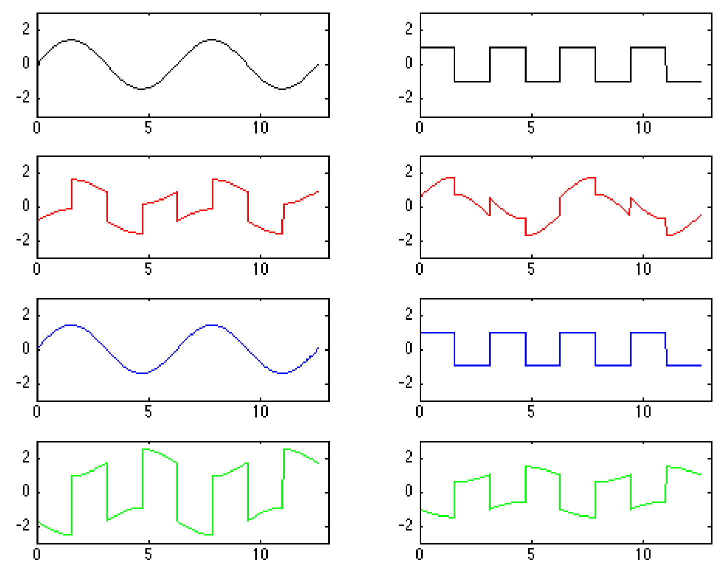

FIG. 6.6. *Top row: original source signals. 2nd row: mixed signals. 3rd row: source signals separated via PCLS. Bottom row: source signals separated via ALS.*

7. Conclusion. We presented an iterative algorithm which implements the partially column-wise least squares method (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 factorizing a symmetric Khatri-Rao product into two similar factor matrices. For symmetric third-order and fourth-order tensors, these symmetric 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 with the ALS approach. In these examples, swamps are not common for the iterative PCLS in contrast to some instances where the ALS method 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 on increasing the speed and efficiency of the current methods.

Acknowledgments. 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 (2007), pp. i10–i18.

[2] J. BRACHAT, P. COMON, B. MOURRAIN, AND E. TSIGARIDAS, *Symmetric tensor decomposition*, Linear Algebra Appl., 433 (2010), pp. 1851–1872.

[3] M. BRAZELL, N. LI, C. NAVASCA, AND C. TAMON, *Solving multilinear systems via tensor inversion*, SIAM J. Matrix Anal. Appl., 34 (2013), pp. 542–570.

[4] J. CARROL AND J. CHANG, *Analysis of individual differences in multidimensional scaling via an n-way heneralization of "Eckart-Young" decomposition*, Psychometrika, 35 (1970), pp. 283–319.

ETNA
Kent State University
http://etna.math.kent.edu

[5] P. COMON, *Independent component analysis, a new concept?*, Signal Process., 36 (1994), pp. 287–314.
[6] P. COMON, G. GOLUB, L-H. LIM, AND B. MOURRAIN, *Symmetric tensors and symmetric tensor rank*, SIAM J. Matrix Anal. Appl., 30 (2008), pp. 1254–1279.
[7] P. COMON AND C. JUTTEN, *Handbook of Blind Source Separation: Independent Component Analysis and Applications*, Academic Press, Oxford, 2010.
[8] P. COMON, X. LUCIANI, AND A. L. F. DE ALMEIDA, *Tensor decompositions, alternating least squares and other tales*, J. Chemometrics, 23 (2009), pp. 393–405.
[9] L. DE LATHAUWER, J. CASTAING, AND J-F. CARDOSO, *Fourth-order cumulant-based blind identification of underdetermined mixtures*, IEEE Trans. Signal Process., 55 (2007), pp. 2965–2973.
[10] L. DE LATHAUWER, B. DE MOOR, AND J. VANDEWALLE, *An introduction to independent component analysis*, J. Chemometrics, 14 (2000), pp. 123–149.
[11] 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 (2007), pp. 844–854.
[12] R. A. HARSHMAN, *Foundations of the PARAFAC procedure: models and conditions for an "explanatory" multi-code factor analysis,* UCLA Working Papers in Phonetics 16, UCLA Linguistics Department, UCLA, Los Angeles, 1970 (84 pages).
[13] F. L. HITCHCOCK, *The expression of a tensor or a polyadic as a sum of products*, J. Math. Phys., 6 (1927), pp. 164–189.
[14] ———, *Multilple invariants and generalized rank of a p-way matrix or tensor*, J. Math. Phys, 7 (1927), pp. 39–79.
[15] A. HYVARINEN, J. KARHUNEN, AND E. OJA, *Independent component analysis,* Stud. Inform. Control, 11 (2002), pp. 205–207.
[16] P. M. KROONENBERG, *Applied Multiway Data Analysis*, Wiley-Interscience, New York, 2008.
[17] J. M. LANDSBERG, *Tensors: Geometry and Applications*, Amer. Math. Soc., Providence, 2010.
[18] N. LI, S. KINDERMANN, AND C. NAVASCA, *Some convergent results of the regularized alternating least-squares method for tensor decomposition*, Linear Algebra Appl., 438 (2013), pp. 796–812.
[19] C. NAVASCA, L. DE LATHAUWER, AND S. KINDERMANN, *Swamp reducing technique for tensor decomposition*, in Proceedings of the European Signal Processing Conference 2008, EUSIPCO online proceedings, EURASIP, 2008 (5 pages).
[20] P. PAATERO, *Construction and analysis of degenerate Parafac models*, J. Chemometrics, 14 (2000), pp. 285–299.
[21] M. RAJIH, P. COMON, AND R. HARSHMAN, *Enchanced line search: a novel nethod to accelerate PARAFAC*, SIAM J. Matrix Anal. Appl., 30 (2008), pp. 1128–1147.
[22] C. R. RAO AND S. K. MITRA, *Generalized Inverse of Matrices and Its Applications*, Wiley, New York, 1971.
[23] T. SCHULTZ AND H. P. SEIDEL, *Estimating crossing fibers: a tensor decomposition approach*, IEEE Trans. Vis. Comput. Graph., 14 (2008), pp. 1635–1642.
[24] N. SIDIROPOULOS, R. BRO, AND G. B. GIANNAKIS, *Parallel factor analysis in sensor array processing*, IEEE Trans. Signal Process., 48 (2000), pp. 2377–2388.
[25] N. SIDIROPOULOS, G. B. GIANNAKIS, AND R. BRO, *Blind PARAFAC receivers for DS-CDMA systems*, IEEE Trans. Signal Process., 48 (2000), pp. 810–823.
[26] A. SMILDE, R. BRO, AND P. GELADI, *Multi-way Analysis. Applications in the Chemical Sciences*, Wiley, Chichester, 2004.
[27] A. STEGEMAN, *On uniqueness of the canonical tensor decomposition with some form of symmetry*, SIAM J. Matrix Anal. Appl., 32 (2011), pp. 561–583.