JACOBI ALGORITHM FOR THE BEST LOW MULTILINEAR RANK APPROXIMATION OF SYMMETRIC TENSORS *

MARIYA ISHTEVA†, P.-A. ABSIL‡, AND PAUL VAN DOOREN‡

Abstract. The problem discussed in this paper is the symmetric best low multilinear rank approximation of third-order symmetric tensors. We propose an algorithm based on Jacobi rotations, for which symmetry is preserved at each iteration. Two numerical examples are provided indicating the need of such algorithms. An important part of the paper consists of proving that our algorithm converges to stationary points of the objective function. This can be considered an advantage of the proposed algorithm over existing symmetry-preserving algorithms in the literature.

Key words. multilinear algebra, higher-order tensor, rank reduction, singular value decomposition, Jacobi rotation.

AMS subject classifications. 15A69, 65F99

1. Introduction. Higher-order tensors (three-way arrays) have been used as a tool in higher-order statistics (HOS) [35, 32, 41, 34] and independent component analysis (ICA) [12, 13, 18, 8] for several decades already. Other application areas include chemometrics, scientific computing, biomedical signal processing, image processing and telecommunications. For an exhaustive list and references we refer to [40, 30, 29, 7, 11].

Let us first consider the general low multilinear rank approximation of third-order tensors. The problem consists of finding the best approximation of a given tensor $\mathcal{A} \in \mathbb{R}^{I_1 \times I_2 \times I_3}$, subject to a constraint on the multilinear rank of the approximation. The concept of multilinear rank was first introduced in [23, 24] and is simply a generalization of the row and column rank of matrices to higher-order tensors. We define the problem more precisely in the next section.

A closed-form solution of the best low multilinear rank approximation problem is not known. A generalization of the singular value decomposition (SVD) [21, §2.5] called higher-order SVD (HOSVD) has been studied in [14]. A variation of this decomposition is know as the Tucker decomposition [43, 44]. In general, truncation of the HOSVD leads to a good but not necessary to the best low multilinear rank approximation. Recent iterative algorithms solving the problem include geometric Newton [19, 27], quasi-Newton [39], trust-region [25] and particle swarm optimization [3] algorithms. In [38], a Krylov subspace algorithm is proposed for large sparse tensors. The most widely used algorithm is still the one based on alternating least squares [15, 30, 31, 2, 42] because of its simplicity and satisfying performance. We will refer to it as higher-order orthogonal iteration (HOOI). It is worth mentioning that the objective function associated with the problem may have several stationary

*Research supported by: (1) National Science Foundation grant CCF-0956517; (2) The Belgian Federal Science Policy Office, Interuniversity Attraction Poles P6/04, Dynamical Systems, Control, and Optimization (DYSCO), 2007–2011; (3) Communauté française de Belgique - Actions de Recherche Concertées. The scientific responsibility rests with the authors. Any opinions, findings and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

†College of Computing, Georgia Institute of Technology, 266 Ferst Drive, Atlanta, GA 30332 USA (mariya.ishteva@cc.gatech.edu).

‡Department of Mathematical Engineering, Université catholique de Louvain, Bâtiment Euler, Av. Georges Lemaître 4, B-1348 Louvain-la-Neuve, Belgium (http://www.inma.ucl.ac.be/~absil/, paul.vandooren@uclouvain.be).

points [26] and none of the iterative algorithms is guaranteed to converge to the global optimum. Most of the algorithms however converge to local optima.

In this paper, we deal with symmetric tensors and symmetric approximations. Symmetric (also called supersymmetric) tensors are tensors invariant to permutation of the indices. Symmetric tensors have been studied in [9] in relation to the parallel factor decomposition (PARAFAC) [22], also know as canonical decomposition (CANDECOMP) [6]. The goal there is to decompose a tensor into a (symmetric) sum of outer products of vectors (rank-1 terms). Symmetric tensors naturally appear, for example, when dealing with HOS in the context of ICA [8, 13]. The low multilinear rank approximation that is considered in this paper can then be used as a dimensionality reduction tool for ICA [18]. As mentioned in [39], finding the best approximation in the case of symmetric tensors is to a large extent different from the general case. Algorithms dedicated to the symmetric case are studied to a lesser extent. A symmetric version of HOOI for the special case of rank-1 tensors is mentioned in [15] and further studied in [28, 37]. In [15] the special case of symmetric $(2 \times 2 \times \cdots \times 2)$-tensors and their rank-1 approximation is studied as well. An algorithm for the general symmetric case, based on the quasi-Newton method is presented in [39].

We develop an algorithm for symmetric tensors, based on Jacobi rotations. The symmetry is preserved at each iteration. The main subproblem reduces to maximizing a polynomial of degree six and finding the value at which the maximum is reached. The main computational cost is due to updating (parts of) the tensor at each rotation. With respect to convergence speed and cost per iteration, our algorithm has similar properties as the general HOOI algorithm, i.e., linear convergence and low cost per iteration. Our algorithm is however especially designed for symmetric tensors. HOOI can be used to solve the symmetric problem as well but the intermediate steps are not symmetric in general. The solutions found by HOOI on the other hand are reported to be symmetric, although there is no proof that this would always be the case. There might be examples where non-symmetric solutions are at least as good as the best symmetric ones with respect to the associated objective function. It is also expected that if the symmetry is taken into account, the computational cost would be reduced. The symmetric version of HOOI [15, 28, 37] converges for the case of rank-1 approximations of even-order tensors under some additional constraints. However, examples for which it does not converge can be easily constructed (see Section 4.2). The quasi-Newton algorithm [39] is expected to converge to stationary points. Our algorithm solves the general case of third-order symmetric tensors and we prove that it converges to stationary points. An advantage of our algorithm over the one in [39] is our comprehensive convergence analysis.

Jacobi-like algorithms for approximating higher-order tensors under certain constraints are already available in the literature. However, to the best of our knowledge, this direction has never been explored for finding the best low multilinear rank approximation of tensors. Jacobi-based algorithms in the literature [8, 17, 16, 4, 5, 33] are designed to solve problems in the framework of ICA or in relation to PARAFAC/CANDECOMP. The main purpose of the algorithms is simultaneously diagonalizing a set of matrices or approximately diagonalizing a higher-order tensor. In [33], for example, the goal is to minimize the values of the off-diagonal elements of the given tensors via orthogonal transformations. Our algorithm on the other hand transforms the elements of the tensor so that, except for a small block, all other elements have small values. The main subproblems are also different. In [33] the main problem is reduced to solving subproblems for $(2 \times 2 \times 2)$-tensors. In our case more elements are involved. Finally, our algorithm is specially designed for symmetric tensors.

This paper is organized as follows. In Section 2 the problem of finding the best low multilinear rank approximation of higher-order tensors is formulated for the general and for the symmetric case. HOSVD and HOOI are briefly presented as well. In Section 3 we present the new algorithm. Some numerical experiments are shown in Section 4. In Section 5 we provide the convergence proof of the proposed algorithm. We summarize the results of the paper in Section 6.

2. Problem formulation. In this section we first provide some basic definitions and comment on the notation. Then the problem of finding the best low multilinear rank approximation of a given tensor is formulated in its general form and an invariance property of the objective function is presented. We also briefly present HOSVD and HOOI. Finally, the main problem for the case of symmetric tensors is formulated.

2.1. Basic definitions and notation. We adopt the notation from [14]. We denote tensors by calligraphic letters, matrices by bold-face capitals, vectors by bold-face letters, and scalars by lower-case letters. Special scalars, such as tensor dimensions, entries of the multilinear rank and upper bounds are written in capital letters. The symbols $\times$ and $\otimes$ stand for the Cartesian and the Kronecker product, respectively. The identity matrix is denoted by $\mathbf{I}$. Finally, O_p and $St(p, n)$ denote the orthogonal group (the set of orthogonal $(p \times p)$-matrices) and the Stiefel manifold (the set of column-wise orthonormal $(n \times p)$-matrices), respectively. In this paper we consider mainly third-order tensors.

It is useful to have *matrix representations* of a tensor. In this paper, $\mathbf{A}_{(1)}, \mathbf{A}_{(2)}, \mathbf{A}_{(3)}$ are matrix representations of a tensor $\mathcal{A} \in \mathbb{R}^{I_1 \times I_2 \times I_3}$, defined in the following way

$$(\mathbf{A}_{(1)})_{i_1,(i_2-1)I_3+i_3} = (\mathbf{A}_{(2)})_{i_2,(i_3-1)I_1+i_1} = (\mathbf{A}_{(3)})_{i_3,(i_1-1)I_2+i_2} = a_{i_1 i_2 i_3}\,, \quad 1 \leq i_n \leq I_n.$$

These matrices are thus obtained via the juxtaposition of the different "slices" of the tensor in the three different directions.

The following tensor-matrix products of a tensor $\mathcal{A} \in \mathbb{R}^{I_1 \times I_2 \times I_3}$ with matrices $\mathbf{M}^{(n)} \in \mathbb{R}^{J_n \times I_n}$ are used:

$$\begin{aligned}
&\textit{mode-1 product:} \quad (\mathcal{A} \bullet_1 \mathbf{M}^{(1)})_{j_1 i_2 i_3} = \textstyle\sum_{i_1} a_{i_1 i_2 i_3} m^{(1)}_{j_1 i_1}\,,\\
&\textit{mode-2 product:} \quad (\mathcal{A} \bullet_2 \mathbf{M}^{(2)})_{i_1 j_2 i_3} = \textstyle\sum_{i_2} a_{i_1 i_2 i_3} m^{(2)}_{j_2 i_2}\,,\\
&\textit{mode-3 product:} \quad (\mathcal{A} \bullet_3 \mathbf{M}^{(3)})_{i_1 i_2 j_3} = \textstyle\sum_{i_3} a_{i_1 i_2 i_3} m^{(3)}_{j_3 i_3}\,,
\end{aligned}$$

where $1 \leq i_n \leq I_n$, $1 \leq j_n \leq J_n$. These products can be considered as a generalization of the left and right multiplication of a matrix $\mathbf{A}$ with a matrix $\mathbf{M}$. The mode-1 product signifies multiplying the columns (mode-1 vectors) of $\mathcal{A}$ with the rows of $\mathbf{M}^{(1)}$ and similarly for the other tensor-matrix products. We also have $(\mathcal{A} \bullet_n \mathbf{M}^{(n)})_{(n)} = \mathbf{M}^{(n)} \mathbf{A}_{(n)}$, $n = 1, 2, 3$.

We will often use the following associativity properties

$$\begin{gathered}
(\mathcal{A} \bullet_n \mathbf{N}) \bullet_m \mathbf{M} = (\mathcal{A} \bullet_m \mathbf{M}) \bullet_n \mathbf{N} = \mathcal{A} \bullet_n \mathbf{N} \bullet_m \mathbf{M}, \quad m \neq n,\\
(\mathcal{A} \bullet_n \mathbf{N}) \bullet_n \mathbf{M} = \mathcal{A} \bullet_n (\mathbf{M}\,\mathbf{N}).
\end{gathered}$$

Finally, the *scalar product* of two tensors $\mathcal{A}, \mathcal{B} \in \mathbb{R}^{I_1 \times I_2 \times I_3}$ is

$$\langle \mathcal{A}, \mathcal{B} \rangle = \sum_{i_1} \sum_{i_2} \sum_{i_3} a_{i_1 i_2 i_3}\, b_{i_1 i_2 i_3}\,, \quad 1 \leq i_n \leq I_n$$

and the *Frobenius norm* of a tensor $\mathcal{A}$ is

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

2.2. The general problem. In the general low multilinear rank approximation problem we look for a tensor $\hat{\mathcal{A}}$ with bounded multilinear rank such that $\hat{\mathcal{A}}$ is a good approximation of a given tensor $\mathcal{A}$. In stricter terms, given $\mathcal{A} \in \mathbb{R}^{I_1 \times I_2 \times I_3}$ and values R_1, R_2, R_3, the problem is to minimize the least-squares cost function $F : \mathbb{R}^{I_1 \times I_2 \times I_3} \to \mathbb{R}$,

$$F : \hat{\mathcal{A}} \mapsto \|\mathcal{A} - \hat{\mathcal{A}}\|^2 \tag{2.1}$$

under the constraints $\operatorname{rank}(\hat{\mathbf{A}}_{(1)}) \leq R_1$, $\operatorname{rank}(\hat{\mathbf{A}}_{(2)}) \leq R_2$, $\operatorname{rank}(\hat{\mathbf{A}}_{(3)}) \leq R_3$.

An equivalent problem is [15] to maximize the function $\overline{g} : St(R_1, I_1) \times St(R_2, I_2) \times St(R_3, I_3) \to \mathbb{R}$,

$$\overline{g} : (\mathbf{U}, \mathbf{V}, \mathbf{W}) \mapsto \|\mathcal{A} \bullet_1 \mathbf{U}^T \bullet_2 \mathbf{V}^T \bullet_3 \mathbf{W}^T\|^2 \tag{2.2}$$

over the column-wise orthonormal matrices $\mathbf{U}, \mathbf{V}$ and $\mathbf{W}$. The link between the solutions of (2.1) and (2.2) is given by

$$\hat{\mathcal{A}} = \mathcal{B} \bullet_1 \mathbf{U} \bullet_2 \mathbf{V} \bullet_3 \mathbf{W}, \tag{2.3}$$

where $\mathcal{B} \in \mathbb{R}^{R_1 \times R_2 \times R_3}$ is given by

$$\mathcal{B} = \mathcal{A} \bullet_1 \mathbf{U}^T \bullet_2 \mathbf{V}^T \bullet_3 \mathbf{W}^T.$$

2.3. Invariance property. The cost function $\overline{g}$ in (2.2) can be reformulated in a matrix form as follows

$$\begin{aligned} \overline{g}(\mathbf{U}, \mathbf{V}, \mathbf{W}) = \|\mathcal{A} \bullet_1 \mathbf{U}^T \bullet_2 \mathbf{V}^T \bullet_3 \mathbf{W}^T\|^2 &= \|\mathbf{U}^T \mathbf{A}_{(1)} (\mathbf{V} \otimes \mathbf{W})\|^2 \\ &= \|\mathbf{V}^T \mathbf{A}_{(2)} (\mathbf{W} \otimes \mathbf{U})\|^2 \\ &= \|\mathbf{W}^T \mathbf{A}_{(3)} (\mathbf{U} \otimes \mathbf{V})\|^2. \end{aligned}$$

It is worth mentioning that $\overline{g}$ has the following invariance property

$$\overline{g}(\mathbf{U}\mathbf{Q}^{(1)}, \mathbf{V}\mathbf{Q}^{(2)}, \mathbf{W}\mathbf{Q}^{(3)}) = \overline{g}(\mathbf{U}, \mathbf{V}, \mathbf{W}), \tag{2.4}$$

where $\mathbf{Q}^{(i)}, i = 1, 2, 3$ are orthogonal matrices. This means that we are interested in the column space of the matrices $\mathbf{U}, \mathbf{V}, \mathbf{W}$ rather than in their exact elements.

2.4. Higher-order singular value decomposition. Matrix SVD [21] is widely used due to its numerous useful properties. One of them is that the best low rank approximation of a matrix is trivially obtained by truncating the SVD of the given matrix. Tensor HOSVD [14, 43, 44] is a generalization of SVD to higher-order tensors that has proven to be useful as well. Every tensor $\mathcal{A} \in \mathbb{R}^{I_1 \times I_2 \times I_3}$ can be decomposed as a product of a tensor $\mathcal{S} \in \mathbb{R}^{I_1 \times I_2 \times I_3}$, called core tensor, and three orthogonal matrices $\mathbf{U}^{(n)} \in \mathbb{R}^{I_n \times I_n}$, $n = 1, 2, 3$, i.e.,

$$\mathcal{A} = \mathcal{S} \bullet_1 \mathbf{U}^{(1)} \bullet_2 \mathbf{U}^{(2)} \bullet_3 \mathbf{U}^{(3)}.$$

The factors are taken is such a way that the matrix slices of $\mathcal{S}$ in any direction are orthogonal to each other and have decreasing norm (when increasing the indices).

These properties reduce to a having a diagonal core matrix if the original tensor is a second-order tensor, i.e., a matrix.

In general, it is impossible to obtain a diagonal core tensor (except in the case of second-order tensors). This is why HOSVD does not trivially lead to the best low multilinear rank approximation of higher-order tensors. However, due to the properties of the core tensor, truncation of HOSVD leads to a good approximation that can be used as a starting point of iterative algorithms.

Computing HOSVD is straightforward and requires computing three SVDs. The columns of the singular matrices $\mathbf{U}^{(n)}, n = 1, 2, 3$ are obtained as the left singular vectors of $\mathbf{A}_{(n)}, n = 1, 2, 3$. The core tensor can then be computed as

$$\mathcal{S} = \mathcal{A} \bullet_1 \mathbf{U}^{(1)^T} \bullet_2 \mathbf{U}^{(2)^T} \bullet_3 \mathbf{U}^{(3)^T} . \tag{2.5}$$

2.5. Higher-order orthogonal iteration. HOOI [15, 30, 31] is an alternating least-squares algorithm for solving the best low multilinear rank approximation problem. Initialization is often based on the truncated HOSVD. Only one of the matrices $\mathbf{U}, \mathbf{V}, \mathbf{W}$ in (2.2) is updated at a time. One iteration step is complete when all the matrices are updated once in a fixed order.

When, for example, the matrix $\mathbf{U}$ is being optimized, the solution is given by the left R_1-dimensional dominant subspace of $\mathbf{A}_{(1)}(\mathbf{V} \otimes \mathbf{W})$. Updates of $\mathbf{V}$ and $\mathbf{W}$ are obtained in a similar way. The convergence rate of HOOI is at most linear.

The main computational cost of one HOOI iteration is forming the matrices of the form $\mathbf{A}_{(1)}(\mathbf{V} \otimes \mathbf{W})$ and computing their first R_n left singular vectors. We assume for simplicity that $R_1 = R_2 = R_3 = R$ and $I_1 = I_2 = I_3 = I$. First, the expression $\mathbf{A}_{(1)}(\mathbf{V} \otimes \mathbf{W})$ is a matrix representation of $\mathcal{A} \bullet_2 \mathbf{V}^T \bullet_3 \mathbf{W}^T$. The cost for computing the product $\mathcal{A} \bullet_2 \mathbf{V}^T$ is of order $O(I^3 R)$ and the cost for computing the product $\mathcal{A} \bullet_2 \mathbf{V}^T \bullet_3 \mathbf{W}^T$, given $\mathcal{A} \bullet_2 \mathbf{V}^T$, is $O(I^2 R^2)$. Thus, the computational cost for $\mathbf{A}_{(1)}(\mathbf{V} \otimes \mathbf{W})$ is of order $O(I^3 R)$. Second, the computational cost for finding the singular vectors is approximately $3(6IR^4 + 11R^6)$ [21, §5.4.5], i.e., $O(IR^4 + R^6)$. The total cost for one iteration of HOOI is then $O(I^3 R + IR^4 + R^6)$ [19, 27].

2.6. The symmetric problem. We look for the best rank-(R, R, R) approximation $\hat{\mathcal{A}} \in \mathbb{R}^{I \times I \times I}$ of a symmetric third-order tensor $\mathcal{A} \in \mathbb{R}^{I \times I \times I}$. $\hat{\mathcal{A}}$ has to be a symmetric tensor that minimizes the least-squares cost function $F_s : \mathbb{R}^{I \times I \times I} \to \mathbb{R}$,

$$F_s : \hat{\mathcal{A}} \mapsto \|\mathcal{A} - \hat{\mathcal{A}}\|^2 \tag{2.6}$$

under the constraint $\mathrm{rank}_1(\hat{\mathcal{A}}) \leq R$ (i.e., $\mathrm{rank}(\hat{\mathbf{A}}_{(1)}) \leq R$).

Instead of minimizing the cost function (2.6) we will solve the equivalent problem (see [15]) of maximizing the function $\overline{g}_s : St(R, I) \to \mathbb{R}$,

$$\overline{g}_s : \mathbf{U} \mapsto \|\mathcal{A} \bullet_1 \mathbf{U}^T \bullet_2 \mathbf{U}^T \bullet_3 \mathbf{U}^T\|^2 \tag{2.7}$$

over the column-wise orthonormal matrix $\mathbf{U}$. After determining $\mathbf{U}$, the optimal tensor $\hat{\mathcal{A}}$ can be computed as

$$\hat{\mathcal{A}} = \mathcal{B} \bullet_1 \mathbf{U} \bullet_2 \mathbf{U} \bullet_3 \mathbf{U}, \tag{2.8}$$

where $\mathcal{B} \in \mathbb{R}^{R \times R \times R}$ is given by

$$\mathcal{B} = \mathcal{A} \bullet_1 \mathbf{U}^T \bullet_2 \mathbf{U}^T \bullet_3 \mathbf{U}^T .$$

Note that the function (2.7) also has the invariance property described in Section 2.3.

3. Jacobi algorithm. The main idea of the proposed algorithm is to apply rotations on the given tensor $\mathcal{A}$ in order to increase the norm of its $(R \times R \times R)$-subtensor with smallest indices (visualized as an $(R \times R \times R)$ top-left-front block in Figure 3.1).

Let $\mathcal{A}$ be a given tensor. We define $f : O(I) \to \mathbb{R}$ to be the function

$$f : \mathbf{Q} \mapsto \overline{g}_s(\mathbf{U}), \tag{3.1}$$

where $\mathbf{Q} = \begin{bmatrix}\mathbf{U} & \mathbf{U}_\perp\end{bmatrix}$, $\mathbf{U} \in St(R, I)$ and $\mathbf{U}_\perp$ is the orthogonal complement of $\mathbf{U}$. (Recall that $O(I)$ is the set of all orthogonal matrices of size $(I \times I)$.)

Then, maximizing $\overline{g}_s$ from (2.7) with respect to $\mathbf{U}$ is equivalent to maximizing f with respect to $\mathbf{Q} = \begin{bmatrix}\mathbf{U} & \mathbf{U}_\perp\end{bmatrix}$. Note that we can rewrite the function f as

$$\begin{aligned} f(\mathbf{Q}) &= \|\mathcal{A} \bullet_1 \mathbf{M}\mathbf{Q}^T \bullet_2 \mathbf{M}\mathbf{Q}^T \bullet_3 \mathbf{M}\mathbf{Q}^T\|^2 \\ &= \|\mathbf{M}\mathbf{Q}^T\, \mathbf{A}_{(1)}((\mathbf{Q}\mathbf{M}) \otimes (\mathbf{Q}\mathbf{M}))\|^2, \end{aligned}$$

where $\mathbf{M} = \begin{pmatrix} \mathbf{I}_R & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{pmatrix} \in \mathbb{R}^{I \times I}$.

3.1. Main idea. It is difficult to find such $\mathbf{Q}$ directly. On the other hand, let $1 \leq m \leq R < n \leq I$ and let $\mathbf{G}_{m,n,\theta}$ be a modified $(I \times I)$ identity matrix with $\mathbf{G}_{m,n,\theta}(m, m) = \mathbf{G}_{m,n,\theta}(n, n) = \cos\theta$, $\mathbf{G}_{m,n,\theta}(m, n) = -\sin\theta$, $\mathbf{G}_{m,n,\theta}(n, m) = \sin\theta$, i.e.,

$$\mathbf{G}_{m,n,\theta} = \begin{pmatrix} 1 & & & & & & \\ & \ddots & & & & \mathbf{0} & \\ & & \cos\theta & & -\sin\theta & & \\ & & & \ddots & & & \\ & & \sin\theta & & \cos\theta & & \\ & \mathbf{0} & & & & \ddots & \\ & & & & & & 1 \end{pmatrix}.$$

For a fixed pair (m, n) it is possible to exactly maximize $f(\mathbf{G}_{m,n,\theta})$ with respect to the angle θ. The procedure is explained in Section 3.2.

This can be used to develop an iterative algorithm for the best symmetric low multilinear rank approximation of a symmetric $\mathcal{A} \in \mathbb{R}^{I \times I \times I}$ in the following way. We initialize by setting $\mathbf{Q}_1 = \mathbf{I}_I$ and $\mathcal{T}_1 = \mathcal{A}$, where $\mathbf{I}_I$ is the $(I \times I)$ identity matrix. At each step we find an optimal $\mathbf{G}_{m_k,n_k,\theta_k}$ for the current $\mathcal{T}_k$. The current approximation $\mathbf{Q}_k$ of $\mathbf{Q}$ is updated by

$$\mathbf{Q}_{k+1} = \mathbf{Q}_k\, \mathbf{G}_{m_k,n_k,\theta_k}.$$

We also modify the current working tensor $\mathcal{T}_k$ by

$$\mathcal{T}_{k+1} \quad = \quad \mathcal{T}_k \bullet_1 \mathbf{G}^T_{m_k,n_k,\theta_k} \bullet_2 \mathbf{G}^T_{m_k,n_k,\theta_k} \bullet_3 \mathbf{G}^T_{m_k,n_k,\theta_k}.$$

Note that

$$\begin{aligned} \mathcal{T}_{k+1} &= (\mathcal{A} \bullet_1 \mathbf{Q}_k^T \bullet_2 \mathbf{Q}_k^T \bullet_3 \mathbf{Q}_k^T) \bullet_1 \mathbf{G}^T_{m_k,n_k,\theta_k} \bullet_2 \mathbf{G}^T_{m_k,n_k,\theta_k} \bullet_3 \mathbf{G}^T_{m_k,n_k,\theta_k} \\ &= \mathcal{A} \bullet_1 (\mathbf{Q}_k \mathbf{G}_{m_k,n_k,\theta_k})^T \bullet_2 (\mathbf{Q}_k \mathbf{G}_{m_k,n_k,\theta_k})^T \bullet_3 (\mathbf{Q}_k \mathbf{G}_{m_k,n_k,\theta_k})^T \\ &= \mathcal{A} \bullet_1 \mathbf{Q}^T_{k+1} \bullet_2 \mathbf{Q}^T_{k+1} \bullet_3 \mathbf{Q}^T_{k+1}. \end{aligned}$$

The pairs (m_k, n_k) are chosen from the following possibilities

$$(1, R+1), (1, R+2), \ldots, (1, I), (2, R+1), \ldots, (R, I). \tag{3.2}$$

In order to ensure convergence to stationary points, we need to choose the pairs (m, n) carefully. As it will be shown in Section 5, it is sufficient to choose each pair (m, n) such that

$$|\langle \operatorname{grad} f(\mathbf{Q}), d_{m,n}(\mathbf{Q})\rangle| \geq \varepsilon \|\operatorname{grad} f(\mathbf{Q})\|, \tag{3.3}$$

where $d_{m,n}(\mathbf{Q}) = \mathbf{Q}\dot{\mathbf{G}}_{m,n,\theta=0}$, $1 \leq m \leq R < n \leq I$ and ε is small. It is always possible to find m and n that satisfy the above inequality (see Lemma 5.2 in Section 5). In practice, however, to speed up the algorithm, one could just cycle through all pairs in (3.2).

The whole procedure is summarized in Algorithm 1.[1]

Algorithm 1 Jacobi based algorithm for minimizing (2.6)

Input: Higher-order tensor $\mathcal{A} \in \mathbb{R}^{I \times I \times I}$, a number R, $R < I$ and initial $\mathbf{Q}_1$ and $\mathcal{T}_1$. ($\mathbf{Q}_1 = \mathbf{I}_I$ and $\mathcal{T}_1 = \mathcal{A}$ can be used as default values.)

Output: Projection matrix $\mathbf{U} \in St(R, I)$ and a rank-(R, R, R) approximation $\hat{\mathcal{A}} = \mathcal{A} \bullet_1 (\mathbf{U}\mathbf{U}^T) \bullet_2 (\mathbf{U}\mathbf{U}^T) \bullet_3 (\mathbf{U}\mathbf{U}^T)$ of $\mathcal{A}$, corresponding to a stationary point of (2.6).

Set ε such that $0 < \varepsilon \leq \frac{2}{I}$ (see remark below).
for $k = 1, 2, \ldots$ until a stopping criterion is satisfied **do**
  Choose (m_k, n_k), $1 \leq m_k \leq R < n_k \leq I$, such that

$$|\langle \operatorname{grad} f(\mathbf{Q}_k), d_{m_k,n_k}(\mathbf{Q}_k)\rangle| \geq \varepsilon \|\operatorname{grad} f(\mathbf{Q}_k)\|,$$

  where $d_{m_k,n_k}(\mathbf{Q}_k) = \mathbf{Q}_k \dot{\mathbf{G}}_{m_k,n_k,\theta=0}$, and $\mathbf{G}_{m_k,n_k,\theta}$ is a modified $(I \times I)$ identity matrix with

$$\begin{aligned} \mathbf{G}_{m_k,n_k,\theta}(m_k, m_k) &= \mathbf{G}_{m_k,n_k,\theta}(n_k, n_k) = \cos\theta, \\ \mathbf{G}_{m_k,n_k,\theta}(m_k, n_k) &= -\sin\theta,\ \mathbf{G}_{m_k,n_k,\theta}(n_k, m_k) = \sin\theta. \end{aligned}$$

  Maximize $f(\mathbf{G}_{m_k,n_k,\theta})$ for the tensor $\mathcal{T}_k$ with respect to θ (Algorithm 2).
  Set $\mathbf{Q}_{k+1} = \mathbf{Q}_k \mathbf{G}_{m_k,n_k,\theta_k}$ where θ_k is a choice of an optimal θ from the previous step.
  Set $\mathcal{T}_{k+1} = \mathcal{T}_k \bullet_1 \mathbf{G}^T_{m_k,n_k,\theta_k} \bullet_2 \mathbf{G}^T_{m_k,n_k,\theta_k} \bullet_3 \mathbf{G}^T_{m_k,n_k,\theta_k}$.
end for
Take the first R columns of $\mathbf{Q}_{k+1}$ as columns of $\mathbf{U}$, i.e., $\mathbf{Q}_{k+1} = \begin{bmatrix}\mathbf{U} & \mathbf{U}_\perp\end{bmatrix}$.
Set $\hat{\mathcal{A}} = (\mathcal{A} \bullet_1 \mathbf{U}^T \bullet_2 \mathbf{U}^T \bullet_3 \mathbf{U}^T) \bullet_1 \mathbf{U} \bullet_2 \mathbf{U} \bullet_3 \mathbf{U}$.

---

[1] Note that $R(I-R)$ is also the number of degrees of freedom that we have for the unknown matrix $\mathbf{U}$, see the invariance property (2.4). $R(I-R)$ is actually the dimension of the Grassmann manifold $Gr(R, I)$.

Remarks:

• Algorithm 1 is a descent algorithm. A proof of convergence to stationary points is presented in Section 5.

• As explained in Section 2.4, truncated HOSVD generally provides a good starting point for iterative algorithms. We can use it in Algorithm 1 as well by setting $\mathbf{Q}_1 = \mathbf{U}^{(1)}$ and $\mathcal{T}_1 = \mathcal{S}$, where $\mathbf{U}^{(1)}$ and $\mathcal{S}$ are as in (2.5). (Note that in the symmetric case, $\mathbf{U}^{(1)} = \mathbf{U}^{(2)} = \mathbf{U}^{(3)}$.)

• There are different options concerning the stopping criterion. A simple approach is to stop after a prespecified number of iterations has been reached or when there is (almost) no change in the column space of the $\mathbf{U}$ matrix. Another strategy would be to consider the gradient of the objective function and to stop when it reaches (almost) zero. The latter would guarantee convergence to a stationary point (see Section 5) but would increase the computational time since it involves additional computations that are not necessary for the working of the algorithm.

• The angle condition (3.3) on (m_k, n_k) ensures convergence to stationary points (see Theorem 5.4). As we will see in Lemma 5.2, the condition $0 < \varepsilon \leq \frac{2}{I}$ guarantees that there is at least one admissible (m_k, n_k). Note that for generic $\mathcal{A}$, cycling through all pairs in (3.2) can be made admissible for an arbitrarily large number of steps by choosing ε small. In practice, we recommend to take $\varepsilon \ll 1$.

• If we assume that all pairs (m, n) are acceptable, the cost per sweep ($R(I - R)$ iterations) is $O(I^3 R)$, which is similar to the cost of one HOOI iteration. (It is also expected that one Jacobi sweep does a similar job as one HOOI iteration.) The computationally heaviest operation in the Jacobi-based algorithm is the update of the tensor (step 6). Taking into account that we need to update only six matrices (and not all the elements of the tensor due to the sparse structure of the rotation matrices), the cost for one update is $O(I^2)$. Within one sweep, there are $R(I - R)$ iterations, which leads to the total cost of $O(I^3 R)$ per sweep.

• Since most of the computational time is spent for updating the tensor (step 6), one may decide to skip certain updates if their effect is much smaller than recently computed updates. An estimation for the impact of an update can be the value $f(\mathbf{G}_{m_k,n_k,\theta_k}) - f(\mathbf{I})$ for the current tensor $\mathcal{T}_k$.

3.2. Choosing a rotation angle θ. Note first that the Frobenius norm of a tensor $\mathcal{T}$ does not change under orthogonal transformations, i.e., if $\mathbf{Q} \in \mathbb{R}^{I \times I}$ is an orthogonal matrix,

$$\|\mathcal{T} \bullet_1 \mathbf{Q}^T \bullet_2 \mathbf{Q}^T \bullet_3 \mathbf{Q}^T\| = \|\mathcal{T}\|.$$

Note also that the matrices $\mathbf{G}_{m,n,\theta}$ in Section 3.1 are orthogonal matrices. Let $\mathcal{T} = \mathcal{A} \bullet_1 \mathbf{Q}_k^T \bullet_2 \mathbf{Q}_k^T \bullet_3 \mathbf{Q}_k^T$. The goal is to maximize the sum of squares of the elements whose indices are smaller than or equal to R, see the gray cube in Figure 3.1(a). This is equivalent to minimizing the sum of squares of the elements having at least one index greater than R. Because of the structure of $\mathbf{G}_{m,n,\theta}^T$, its application to $\mathcal{T}$ (on all three modes) changes only the elements having an index equal to m or to n. These elements form six matrices, two in each direction. The three matrices with an m index are shown in Figure 3.1(b) for $m = R$. From these elements, we are only interested in maximizing the ones that do not have an index greater than R, Figure 3.1(c). These elements form three $(R \times R)$ matrices (in gray). Each two of the three matrices intersect in a vector (dotted line) and the three vectors have one common element.

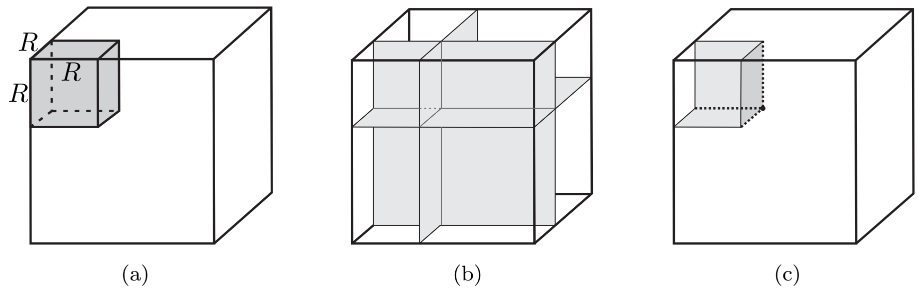

FIGURE 3.1. *Elements taking part in the Jacobi update*

For fixed m and n the optimization problem of maximizing the elements from Figure 3.1(c) by applying $\mathbf{G}^T_{m,n,\theta}$ can be solved exactly for θ. Let $c = \cos\theta$ and $s = \sin\theta$. By applying $\mathbf{G}^T_{m,n,\theta}$ to $\mathcal{T}$ on all three modes, the sum of the squares of the elements we want to maximize (Figure 3.1(c)) changes from

$$3\sum_{i,j=1}^{R-1}\mathcal{T}(i,j,m)^2 + 3\sum_{i=1}^{R-1}\mathcal{T}(i,m,m)^2 + \mathcal{T}(m,m,m)^2$$

to

$$\begin{aligned}\psi(c,s) \quad = \quad & 3\sum_{i,j=1}^{R-1}[c\mathcal{T}(i,j,m)+s\mathcal{T}(i,j,n)]^2 \\ + \quad & 3\sum_{i=1}^{R-1}[c^2\mathcal{T}(i,m,m)+s^2\mathcal{T}(i,n,n)+2cs\mathcal{T}(i,m,n)]^2 \\ + \quad & [c^3\mathcal{T}(m,m,m)+s^3\mathcal{T}(n,n,n)+3c^2s\mathcal{T}(m,m,n)+3cs^2\mathcal{T}(m,n,n)]^2.\end{aligned} \tag{3.4}$$

Let $\varphi(c,s) = \varphi_2(c,s)+\varphi_4(c,s)+\varphi_6(c,s)$ be the derivative of $\psi(c,s)$ with respect to θ, where $\varphi_i(c,s)$ is a homogeneous polynomial of degree i in c and s, $i = 2,4,6$. It is possible to transform $\varphi(c,s) = 0$ to a homogeneous equation of degree 6 in the following way

$$\varphi_2(c,s)(c^2+s^2)^2 + \varphi_4(c,s)(c^2+s^2) + \varphi_6(c,s) = 0.$$

This can now be reduced to a polynomial equation

$$P(t(\theta)) = 0. \tag{3.5}$$

of degree 6 in $t = s/c$. We then look for the solution of (3.5) that maximizes $\psi(c,s)$. If $\psi(c(\theta),s(\theta))$ achieves its maximum at two points θ_1 and θ_2, we choose the one with smaller $|\theta|$. (If both θ and $-\theta$ maximize $\psi(c,s)$, we choose the positive one.) The algorithm is summarized in Algorithm 2.

In the special case where we are looking for the best rank-1 approximation of symmetric $(2\times2\times2)$-tensors, our formulas reduce to the formulas derived in [15, §3.5].

Algorithm 2 Choosing the optimal angle θ for step 4 in Algorithm 1

Input: $\mathcal{T}$, R, m, n.
Output: θ.
1: Form $\psi(c, s)$ as in (3.4).
2: Compute $\varphi(c, s)$, the derivative of $\psi(c, s)$ with respect to θ.
3: Compute the coefficients of $P(t)$ from (3.5).
4: Find all zeros of $P(t) = 0$.
5: Choose θ as the zero of $P(t(\theta)) = 0$ for which $\psi(c, s)$ has the highest value.

4. Examples.

The advantages of our algorithm are that it serves a specific purpose and has well-understood convergence behavior. We do not claim speed improvement with respect to HOOI or symmetric quasi-Newton.

The examples proposed in this section illustrate the need of reliable symmetric algorithms. The convergence proof of the algorithm will be given in the next section.

4.1. Partial symmetry.

Consider the following example [10]

$$\mathcal{A} = \mathbf{a} \circ \mathbf{b} \circ \mathbf{c} + \mathbf{b} \circ \mathbf{c} \circ \mathbf{a} + \mathbf{c} \circ \mathbf{a} \circ \mathbf{b},$$

where $\circ$ stands for the outer product of vectors and $\mathbf{a}, \mathbf{b}, \mathbf{c} \in \mathbb{R}^n$ have unit norm and are orthogonal to each other. We have

$$\mathcal{A}_{i,j,k} = \mathcal{A}_{j,k,i} = \mathcal{A}_{k,i,j},$$

i.e, $\mathcal{A}$ is partially symmetric. However, it appears that its best rank-$(1, 1, 1)$ approximation does not have the same partial symmetry in general.

For example, let us take

$$\mathbf{a} = \begin{pmatrix} -0.6060 \\ 0.3195 \\ 0.7285 \end{pmatrix}, \quad \mathbf{b} = \begin{pmatrix} 0.7955 \\ 0.2491 \\ 0.5524 \end{pmatrix}, \quad \mathbf{c} = \begin{pmatrix} -0.0050 \\ 0.9143 \\ -0.4051 \end{pmatrix}.$$

Then

$$\begin{aligned}
\mathcal{A}(:, :, 1) &= \begin{pmatrix} 0.0072 & -0.4413 & 0.1941 \\ -0.4413 & 0.0940 & 0.5901 \\ 0.1941 & -0.4099 & -0.1012 \end{pmatrix}, \\
\mathcal{A}(:, :, 2) &= \begin{pmatrix} -0.4413 & 0.0940 & -0.4099 \\ 0.0940 & 0.2183 & 0.2950 \\ 0.5901 & 0.2950 & 0.2229 \end{pmatrix}, \\
\mathcal{A}(:, :, 3) &= \begin{pmatrix} 0.1941 & 0.5901 & -0.1012 \\ -0.4099 & 0.2950 & 0.2229 \\ -0.1012 & 0.2229 & -0.4891 \end{pmatrix},
\end{aligned}$$

where we have used MATLAB's notation $\mathcal{A}(:, :, i)$ to denote the i-th frontal slice of $\mathcal{A}$. If we initialize with the truncated HOSVD and run HOOI on this example, for the rank-$(1,1,1)$ approximation $\hat{\mathcal{A}}$ of $\mathcal{A}$ we get

$$\hat{\mathcal{A}}(:,:,1) = \begin{pmatrix} 0.0024 & -0.0013 & -0.0029 \\ -0.4408 & 0.2324 & 0.5299 \\ 0.1953 & -0.1030 & -0.2348 \end{pmatrix},$$

$$\hat{\mathcal{A}}(:,:,2) = \begin{pmatrix} 0.0008 & -0.0004 & -0.0009 \\ -0.1380 & 0.0728 & 0.1659 \\ 0.0612 & -0.0322 & -0.0735 \end{pmatrix},$$

$$\hat{\mathcal{A}}(:,:,3) = \begin{pmatrix} 0.0017 & -0.0009 & -0.0020 \\ -0.3061 & 0.1614 & 0.3679 \\ 0.1356 & -0.0715 & -0.1630 \end{pmatrix}.$$

As it can be seen, the partial symmetry has been lost.

Whether it is possible to construct similar examples for (super)symmetric tensors remains an open question; see [20]. However, the above example indicates that it is not trivial to assume that the best rank-(R,R,R) approximation of a symmetric tensor would be symmetric. Note also that in applications involving symmetric tensors, one looks for a symmetric approximation in general. Thus, algorithms seeking the best rank-(R,R,R) symmetric approximation of a given symmetric tensors should be used.

4.2. Symmetric example. In this section we present an example for which the symmetric HOOI algorithm does not converge.

Consider the following symmetric tensor

$$\mathcal{A}(:,:,1) = \begin{pmatrix} 0.1182 & 0.3220 & 1.3986 \\ 0.3220 & 1.2843 & -0.8889 \\ 1.3986 & -0.8889 & -0.8642 \end{pmatrix},$$

$$\mathcal{A}(:,:,2) = \begin{pmatrix} 0.3220 & 1.2843 & -0.8889 \\ 1.2843 & -0.2796 & -1.4854 \\ -0.8889 & -1.4854 & 0.2368 \end{pmatrix},$$

$$\mathcal{A}(:,:,3) = \begin{pmatrix} 1.3986 & -0.8889 & -0.8642 \\ -0.8889 & -1.4854 & 0.2368 \\ -0.8642 & 0.2368 & 0.7504 \end{pmatrix}$$

and let $R = 1$. Note that in the rank-$(1,1,1)$ case HOOI simplifies to higher-order power method (HOPM) [14]. The convergence behavior of HOPM, symmetric HOPM and the Jacobi algorithm for computing the rank-$(1,1,1)$ approximation of $\mathcal{A}$ is presented in Figure 4.1. The algorithms were initialized with truncated HOSVD and stopped after computing 100 iterations each. HOPM and Jacobi converged to the same solution. Symmetric HOPM did not converge for this example. Such examples are easily obtained. However, we mention that there are also examples where the symmetric HOPM converges as well.

The Jacobi-type algorithm proposed in this paper is of the same class of algorithms as HOOI with respect to convergence speed and computational time but our algorithm is guaranteed to converge to stationary points (of the symmetric problem) as shown in Section 5.

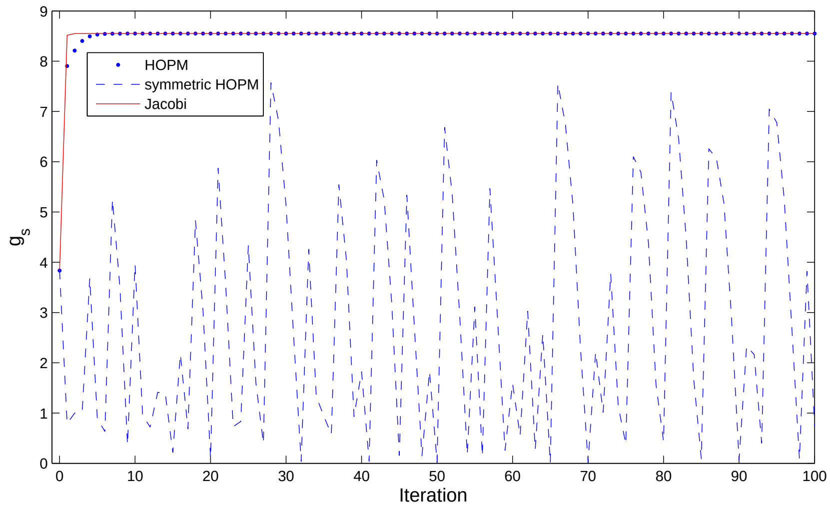

FIGURE 4.1. *Convergence behavior of HOPM, symmetric HOPM and the Jacobi algorithm for the tensor in Example 2. Truncated HOSVD was used for the initialization.*

5. Convergence proof. In this section we prove that every accumulation point of a sequence generated by the proposed algorithm is a stationary point of (3.1). We first prove three lemmas. Lemma 5.1 presents the structure of the gradient of f from (3.1). This is used in Lemma 5.2 in order to prove that if a point $\mathbf{Q}$ is not a stationary point, then there exists a rotation $\mathbf{G}_{m,n,\theta}$ that would increase the value of f. Lemma 5.3 uses Lemma 5.2 to show that if $\mathbf{Q}$ is not a stationary point, then for any point in a small enough neighborhood around $\mathbf{Q}$, applying one step of the algorithm would improve the value of f by an increment that is bounded away from zero.

Finally, based on Polak's theorem [36, §1.3, Th. 3] we can prove our main result by contradiction. Assume that there is an accumulation point $\overline{\mathbf{Q}}$ that is not a stationary point and take a subsequence $\{\mathbf{Q}_j\}_{j\in K}$ that converges to $\overline{\mathbf{Q}}$. Then after some step, the distance between each point of the subsequence and $\overline{\mathbf{Q}}$ would stay small enough and from Lemma 5.3 we have that the value of $f(\mathbf{Q}_j)$ would continue to increase without bound as $j \to \infty$. However, since f is continuous and $\{\mathbf{Q}_j\}_{j\in K}$ converges, $\{f(\mathbf{Q}_j)\}_{j\in K}$ should converge too, which is a contradiction.

LEMMA 5.1. *The gradient of f has the following structure*

$$\operatorname{grad} f(\mathbf{Q}) = \mathbf{Q}\left(\begin{array}{c|c} \mathbf{0}_R & \mathbf{C} \\ \hline -\mathbf{C}^T & \mathbf{0} \end{array}\right),$$

for some matrix $\mathbf{C} \in \mathbb{R}^{R\times(I-R)}$.

Proof. In order to compute $\operatorname{grad} f(\mathbf{Q})$ we first need to compute $\operatorname{grad} \tilde{f}(\mathbf{Q})$, where $\tilde{f} : \mathbb{R}^{I\times I} \to \mathbb{R}$ is the function

$$\tilde{f} : \mathbf{Q} \mapsto \|\mathcal{A} \bullet_1 \mathbf{M}\mathbf{Q}^T \bullet_2 \mathbf{M}\mathbf{Q}^T \bullet_3 \mathbf{M}\mathbf{Q}^T\|^2.$$

Then $\operatorname{grad} f(\mathbf{Q})$ is obtained by projecting $\operatorname{grad} \tilde{f}(\mathbf{Q})$ onto the tangent space at $\mathbf{Q}$ to the manifold O_I, i.e.,

$$\operatorname{grad} f(\mathbf{Q}) = P_{\mathbf{Q}}(\operatorname{grad} \tilde{f}(\mathbf{Q})).$$

From the theory of matrix manifolds, it is known that the tangent space at $\mathbf{Q}$ to O_I is

$$T_{\mathbf{Q}} O_I = \{\mathbf{Z} = \mathbf{Q}\Omega : \Omega^T = -\Omega\} = \mathbf{Q}\, S_{\text{skew}}.$$

Let $\operatorname{skew}(\mathbf{B}) = (\mathbf{B} - \mathbf{B}^T)/2$. The projection onto the tangent space is [1, Example 3.6.2]

$$P_{\mathbf{Q}}\mathbf{Z} = \mathbf{Q} \operatorname{skew}(\mathbf{Q}^T\mathbf{Z}) = \mathbf{Q}(\mathbf{Q}^T\mathbf{Z} - \mathbf{Z}^T\mathbf{Q})/2.$$

It remains to compute $\operatorname{grad} \tilde{f}(\mathbf{Q})$. From the definition of $\tilde{f}(\mathbf{Q})$ we have

$$\tilde{f}(\mathbf{Q}) = \langle \mathcal{A} \bullet_1 \mathbf{M}\mathbf{Q}^T \bullet_2 \mathbf{M}\mathbf{Q}^T \bullet_3 \mathbf{M}\mathbf{Q}^T,\, \mathcal{A} \bullet_1 \mathbf{M}\mathbf{Q}^T \bullet_2 \mathbf{M}\mathbf{Q}^T \bullet_3 \mathbf{M}\mathbf{Q}^T \rangle.$$

The differential of $\tilde{f}$ is then

$$\begin{aligned}
\mathrm{D}\tilde{f}(\mathbf{Q})[\mathbf{Z}_{\mathbf{Q}}] = \; & \langle \mathcal{A} \bullet_1 \mathbf{M}\mathbf{Z}_{\mathbf{Q}}^T \bullet_2 \mathbf{M}\mathbf{Q}^T \bullet_3 \mathbf{M}\mathbf{Q}^T,\, \mathcal{A} \bullet_1 \mathbf{M}\mathbf{Q}^T \bullet_2 \mathbf{M}\mathbf{Q}^T \bullet_3 \mathbf{M}\mathbf{Q}^T \rangle \\
& + \langle \mathcal{A} \bullet_1 \mathbf{M}\mathbf{Q}^T \bullet_2 \mathbf{M}\mathbf{Z}_{\mathbf{Q}}^T \bullet_3 \mathbf{M}\mathbf{Q}^T,\, \mathcal{A} \bullet_1 \mathbf{M}\mathbf{Q}^T \bullet_2 \mathbf{M}\mathbf{Q}^T \bullet_3 \mathbf{M}\mathbf{Q}^T \rangle \\
& + \langle \mathcal{A} \bullet_1 \mathbf{M}\mathbf{Q}^T \bullet_2 \mathbf{M}\mathbf{Q}^T \bullet_3 \mathbf{M}\mathbf{Z}_{\mathbf{Q}}^T,\, \mathcal{A} \bullet_1 \mathbf{M}\mathbf{Q}^T \bullet_2 \mathbf{M}\mathbf{Q}^T \bullet_3 \mathbf{M}\mathbf{Q}^T \rangle \\
& + \langle \mathcal{A} \bullet_1 \mathbf{M}\mathbf{Q}^T \bullet_2 \mathbf{M}\mathbf{Q}^T \bullet_3 \mathbf{M}\mathbf{Q}^T,\, \mathcal{A} \bullet_1 \mathbf{M}\mathbf{Z}_{\mathbf{Q}}^T \bullet_2 \mathbf{M}\mathbf{Q}^T \bullet_3 \mathbf{M}\mathbf{Q}^T \rangle \\
& + \langle \mathcal{A} \bullet_1 \mathbf{M}\mathbf{Q}^T \bullet_2 \mathbf{M}\mathbf{Q}^T \bullet_3 \mathbf{M}\mathbf{Q}^T,\, \mathcal{A} \bullet_1 \mathbf{M}\mathbf{Q}^T \bullet_2 \mathbf{M}\mathbf{Z}_{\mathbf{Q}}^T \bullet_3 \mathbf{M}\mathbf{Q}^T \rangle \\
& + \langle \mathcal{A} \bullet_1 \mathbf{M}\mathbf{Q}^T \bullet_2 \mathbf{M}\mathbf{Q}^T \bullet_3 \mathbf{M}\mathbf{Q}^T,\, \mathcal{A} \bullet_1 \mathbf{M}\mathbf{Q}^T \bullet_2 \mathbf{M}\mathbf{Q}^T \bullet_3 \mathbf{M}\mathbf{Z}_{\mathbf{Q}}^T \rangle \big).
\end{aligned} \tag{5.1}$$

All summands in (5.1) have the same value. Consider for example the first two of them. They are equivalent since $\mathcal{A}$ is symmetric and thus $\mathcal{A} \bullet_1 \mathbf{M}\mathbf{Q}^T \bullet_2 \mathbf{M}\mathbf{Z}_{\mathbf{Q}}^T \bullet_3 \mathbf{M}\mathbf{Q}^T$ can be obtained from $\mathcal{A} \bullet_1 \mathbf{M}\mathbf{Z}_{\mathbf{Q}}^T \bullet_2 \mathbf{M}\mathbf{Q}^T \bullet_3 \mathbf{M}\mathbf{Q}^T$ by permuting its elements. (The corresponding permuted version of $\mathcal{A} \bullet_1 \mathbf{M}\mathbf{Q}^T \bullet_2 \mathbf{M}\mathbf{Q}^T \bullet_3 \mathbf{M}\mathbf{Q}^T$ is still $\mathcal{A} \bullet_1 \mathbf{M}\mathbf{Q}^T \bullet_2 \mathbf{M}\mathbf{Q}^T \bullet_3 \mathbf{M}\mathbf{Q}^T$.) Similarly, we can show that the rest of the summands have the same value as well. Note that $\mathbf{M}^T = \mathbf{M}$ and $\mathbf{M}\mathbf{M} = \mathbf{M}$. To simplify the notation, we will write $(\mathbf{QM} \otimes \mathbf{QM})$ for $((\mathbf{QM}) \otimes (\mathbf{QM}))$. Taking into account that $\operatorname{trace}(\mathbf{AB}) = \operatorname{trace}(\mathbf{BA})$ for any two matrices $\mathbf{A}$ and $\mathbf{B}$,

$$\begin{aligned}
\mathrm{D}\tilde{f}(\mathbf{Q})[\mathbf{Z}_{\mathbf{Q}}] &= 6\, \langle \mathcal{A} \bullet_1 \mathbf{M}\mathbf{Z}_{\mathbf{Q}}^T \bullet_2 \mathbf{M}\mathbf{Q}^T \bullet_3 \mathbf{M}\mathbf{Q}^T,\, \mathcal{A} \bullet_1 \mathbf{M}\mathbf{Q}^T \bullet_2 \mathbf{M}\mathbf{Q}^T \bullet_3 \mathbf{M}\mathbf{Q}^T \rangle \\
&= 6\, \langle \mathbf{M}\mathbf{Z}_{\mathbf{Q}}^T \mathbf{A}_{(1)} (\mathbf{QM} \otimes \mathbf{QM}),\, \mathbf{M}\mathbf{Q}^T \mathbf{A}_{(1)} (\mathbf{QM} \otimes \mathbf{QM}) \rangle \\
&= 6 \operatorname{trace}\big(\mathbf{M}\mathbf{Z}_{\mathbf{Q}}^T \mathbf{A}_{(1)} (\mathbf{QM} \otimes \mathbf{QM})(\mathbf{QM} \otimes \mathbf{QM})^T \mathbf{A}_{(1)}^T \mathbf{QM}\big) \\
&= 6 \operatorname{trace}\big(\mathbf{Z}_{\mathbf{Q}}^T \mathbf{A}_{(1)} (\mathbf{QM} \otimes \mathbf{QM})(\mathbf{QM} \otimes \mathbf{QM})^T \mathbf{A}_{(1)}^T \mathbf{QMM}\big) \\
&= 6\, \langle \mathbf{Z}_{\mathbf{Q}},\, \mathbf{A}_{(1)} (\mathbf{QM} \otimes \mathbf{QM})(\mathbf{QM} \otimes \mathbf{QM})^T \mathbf{A}_{(1)}^T \mathbf{QM} \rangle.
\end{aligned}$$

From the latter expression we can conclude that

$$\operatorname{grad} \tilde{f}(\mathbf{Q}) = 6\, \mathbf{A}_{(1)} (\mathbf{QM} \otimes \mathbf{QM})(\mathbf{QM} \otimes \mathbf{QM})^T \mathbf{A}_{(1)}^T \mathbf{QM}.$$

Finally, for $\operatorname{grad} f(\mathbf{Q})$ we have

$$
\begin{aligned}
\operatorname{grad} f(\mathbf{Q}) &= P_{\mathbf{Q}}(\operatorname{grad} \tilde{f}(\mathbf{Q})) \\
&= P_{\mathbf{Q}}\big(6\, \mathbf{A}_{(1)}(\mathbf{QM} \otimes \mathbf{QM})(\mathbf{QM} \otimes \mathbf{QM})^T \mathbf{A}_{(1)}^T \mathbf{QM}\big) \\
&= 6\, \mathbf{Q} \operatorname{skew}\big(\mathbf{Q}^T \mathbf{A}_{(1)}(\mathbf{QM} \otimes \mathbf{QM})(\mathbf{QM} \otimes \mathbf{QM})^T \mathbf{A}_{(1)}^T \mathbf{QM}\big) \\
&= 6\, \mathbf{Q}\big(\mathbf{Q}^T \mathbf{A}_{(1)}(\mathbf{QM} \otimes \mathbf{QM})(\mathbf{QM} \otimes \mathbf{QM})^T \mathbf{A}_{(1)}^T \mathbf{QM} \\
&\qquad -\mathbf{MQ}^T \mathbf{A}_{(1)}(\mathbf{QM} \otimes \mathbf{QM})(\mathbf{QM} \otimes \mathbf{QM})^T \mathbf{A}_{(1)}^T \mathbf{Q}\big)/2 \\
&= 3\, \mathbf{Q}\big(\mathbf{SM} - \mathbf{MS}\big),
\end{aligned}
$$

where $\mathbf{S} = \mathbf{Q}^T \mathbf{A}_{(1)}(\mathbf{QM} \otimes \mathbf{QM})(\mathbf{QM} \otimes \mathbf{QM})^T \mathbf{A}_{(1)}^T \mathbf{Q}$. Note that $\mathbf{S}^T = \mathbf{S}$, i.e., $\mathbf{S} = \begin{pmatrix} \mathbf{S}_{11} & \mathbf{S}_{12} \\ \mathbf{S}_{12}^T & \mathbf{S}_{22} \end{pmatrix}$ with $\mathbf{S}_{11} \in \mathbb{R}^{R\times R}$, $\mathbf{S}_{12} \in \mathbb{R}^{R\times(I-R)}, \mathbf{S}_{22} \in \mathbb{R}^{(I-R)\times(I-R)}$. Then

$$
\mathbf{SM} - \mathbf{MS} = \begin{pmatrix} \mathbf{S}_{11} & \mathbf{0} \\ \mathbf{S}_{12}^T & \mathbf{0} \end{pmatrix} - \begin{pmatrix} \mathbf{S}_{11} & \mathbf{S}_{12} \\ \mathbf{0} & \mathbf{0} \end{pmatrix} = \begin{pmatrix} \mathbf{0} & -\mathbf{S}_{12} \\ \mathbf{S}_{12}^T & \mathbf{0} \end{pmatrix}
$$

and thus if we set $\mathbf{C} = -3\,\mathbf{S}_{12}$,

$$
\operatorname{grad} f(\mathbf{Q}) = \mathbf{Q} \left(\begin{array}{c|c} \mathbf{0}_R & \mathbf{C} \\ \hline -\mathbf{C}^T & \mathbf{0} \end{array} \right),
$$

which concludes the proof. ∎

Lemma 5.2. *For every orthogonal* $\mathbf{Q}$ *and every* $0 < \varepsilon \leq \frac{2}{I}$ *there exists a direction* $d_{m,n}(\mathbf{Q})$ *such that*

$$
|\langle \operatorname{grad} f(\mathbf{Q}), d_{m,n}(\mathbf{Q})\rangle| \geq \varepsilon \|\operatorname{grad} f(\mathbf{Q})\|.
$$

Proof. If $\operatorname{grad} f(\mathbf{Q}) = 0$ the equality holds trivially. Let $\operatorname{grad} f(\mathbf{Q}) = \mathbf{QN} \neq 0$, let $1 \leq m \leq R < n \leq I$ be such that $|\mathbf{N}(m,n)| = \max\limits_{1\leq i,j\leq I} |\mathbf{N}(i,j)| > 0$ and let $x := \mathbf{N}(m,n)$. Then

$$
\|\operatorname{grad} f(\mathbf{Q})\| = \|\mathbf{QN}\| = \|\mathbf{N}\| = \sqrt{\sum_{1\leq i,j\leq I} (\mathbf{N}(i,j))^2} \leq \sqrt{I^2x^2} = I|x|. \tag{5.2}
$$

Consider next $\dot{\mathbf{G}}_{m,n,\theta=0}$:

$$
\dot{\mathbf{G}}_{m,n,\theta=0} = \frac{d}{d\theta}\mathbf{G}_{m,n,\theta}\Big|_{\theta=0} = \begin{pmatrix} 0 & & & \\ & & & \mathbf{0} \\ & -\sin 0 & -\cos 0 & \\ & \cos 0 & -\sin 0 & \\ \mathbf{0} & & & \\ & & & 0 \end{pmatrix} = \begin{pmatrix} 0 & & & \\ & & & \mathbf{0} \\ & 0 & -1 & \\ & 1 & 0 & \\ \mathbf{0} & & & \\ & & & 0 \end{pmatrix}.
$$

Then

$$d_{m,n}(\mathbf{Q}) = \mathbf{Q}\,\dot{\mathbf{G}}_{m,n,\theta=0} = \mathbf{Q}\begin{pmatrix} 0 & & & & \mathbf{0} \\ & & & -1 & \\ & & \ddots & & \\ & 1 & & & \\ \mathbf{0} & & & & 0 \end{pmatrix}.$$

Recall that $\operatorname{grad} f(\mathbf{Q}) = \mathbf{Q}\mathbf{N}$ and recall the structure of $\mathbf{N}$ from Lemma 5.1. We have

$$(\operatorname{grad} f(\mathbf{Q}))^T d_{m,n} = \underbrace{\left(\begin{array}{c|c} \mathbf{0} & -x \quad * \\ \hline * \quad x & \mathbf{0} \end{array}\right)}_{\mathbf{N}^T} \mathbf{Q}^T\,\mathbf{Q} \underbrace{\left(\begin{array}{c|c} \mathbf{0} & -1 \quad \mathbf{0} \\ \hline \mathbf{0} \quad 1 & \mathbf{0} \end{array}\right)}_{\dot{\mathbf{G}}_{m,n,\theta=0}}$$

and thus

$$\begin{aligned} \langle \operatorname{grad} f(\mathbf{Q}), d_{m,n}(\mathbf{Q})\rangle &= \operatorname{trace}((\operatorname{grad} f(\mathbf{Q}))^T d_{m,n}) = \operatorname{trace}(\mathbf{N}^T\mathbf{Q}^T\mathbf{Q}\,\dot{\mathbf{G}}_{m,n,\theta=0}) \\ &= \operatorname{trace}(\mathbf{N}^T\dot{\mathbf{G}}_{m,n,\theta=0}) = -2x. \end{aligned} \tag{5.3}$$

We now have

$$|\langle \operatorname{grad} f(\mathbf{Q}), d_{m,n}(\mathbf{Q})\rangle| = 2|x| = \frac{2}{I} I|x| \geq \frac{2}{I}\|\operatorname{grad} f(\mathbf{Q})\| \geq \varepsilon \|\operatorname{grad} f(\mathbf{Q})\|,$$

which completes the proof of the lemma. ▯

Let $a : O(I) \to 2^{O(I)}$ be the set-valued function such that, for all $\mathbf{Q} \in O(I)$, $a(\mathbf{Q})$ is the set of all $\mathbf{Q}_{k+1}$ that can be generated from $\mathbf{Q}_k = \mathbf{Q}$ by Steps 3–5 of Algorithm 1. Note that function a is set-valued because there is, in general, more than one possible choice for (m_k, n_k) in Step 3 and for θ_k in Step 5.

LEMMA 5.3. *For every orthogonal* $\overline{\mathbf{Q}}$ *such that* $\operatorname{grad} f(\overline{\mathbf{Q}}) \neq 0$ *there exist* $\eta > 0$ *and* $\delta > 0$ *such that*

$$f(\mathbf{Q}') - f(\mathbf{Q}) \geq \delta > 0, \quad \textit{for all } \mathbf{Q} \in B_\eta(\overline{\mathbf{Q}}) \textit{ and all } \mathbf{Q}' \in a(\mathbf{Q}),$$

where $B_\eta(\overline{\mathbf{Q}}) = \{\mathbf{Q} : \|\mathbf{Q} - \overline{\mathbf{Q}}\| \leq \eta\}$.

Proof. Let $\overline{\mathbf{Q}}$ be such that $\operatorname{grad} f(\overline{\mathbf{Q}}) \neq 0$. Since $f \in C^\infty$, there exists $\eta > 0$ such that $\varepsilon_1 := \min\limits_{\mathbf{Q}\in B_\eta(\overline{\mathbf{Q}})} \|\operatorname{grad} f(\mathbf{Q})\| > 0$. Let ε be as in Algorithm 1. Define $h_{\mathbf{Q},m,n}$: $\mathbb{R} \to \mathbb{R} : \theta \mapsto f(\mathbf{Q}\mathbf{G}_{m,n,\theta})$. Let $M = \max_{\mathbf{Q}\in B_\eta(\overline{\mathbf{Q}}), 1\leq m\leq R<n\leq I, \theta\in\mathbb{R}} |h''_{\mathbf{Q},m,n}(\theta)|$; since $h_{\mathbf{Q},m,n}$ is smooth and 2π periodic, $M < \infty$. Let $\delta = \frac{\varepsilon^2\varepsilon_1^2}{2M}$.

Let $\mathbf{Q}_k \in B_\eta(\overline{\mathbf{Q}})$, and let $\mathbf{Q}_{k+1}$, m_k, n_k and θ_k be obtained from Steps 3–5 of Algorithm 1. We show that $f(\mathbf{Q}_{k+1}) - f(\mathbf{Q}_k) \geq \delta$, hence the claim.

To lighten the notation, we denote $h_{\mathbf{Q}_k,m_k,n_k}$ by h_k, i.e., $h_k(\theta) := f(\mathbf{Q}_k\mathbf{G}_{m_k,n_k,\theta})$. Note that

$$f(\mathbf{Q}_k) = h_k(0) \quad \text{and} \quad f(\mathbf{Q}_{k+1}) = \max_\theta h_k(\theta). \tag{5.4}$$

In view of Step 3 of Algorithm 1, we have

$$\|h_k'(0)\| = |\langle \operatorname{grad} f(\mathbf{Q}_k\, \mathbf{G}_{m_k,n_k,0}), \mathbf{Q}_k\, \dot{\mathbf{G}}_{m_k,n_k,0}\rangle| = |\langle \operatorname{grad} f(\mathbf{Q}_k), d_{m_k,n_k}(\mathbf{Q}_k)\rangle| \geq \varepsilon \|\operatorname{grad} f(\mathbf{Q}_k)\|.$$

Thus

$$\|h_k'(0)\| \geq \varepsilon \min_{\mathbf{Q}\in B_\eta(\overline{\mathbf{Q}})} \|\operatorname{grad} f(\mathbf{Q})\| = \varepsilon\varepsilon_1 > 0. \tag{5.5}$$

The Taylor expansion of h_k around 0 is given by

$$\begin{aligned} h_k(\theta) &= h_k(0) + h_k'(0)\theta + \frac{1}{2}h_k''(\xi)\theta^2 \\ &\geq h_k(0) + h_k'(0)\theta - \frac{1}{2}M\theta^2, \end{aligned}$$

where $0 \leq \xi \leq \theta$. For $\theta = \left(\frac{h_k'(0)}{M}\right)$, this yields

$$h_k\left(\frac{h_k'(0)}{M}\right) - h_k(0) \geq h_k'(0)\frac{h_k'(0)}{M} - \frac{1}{2}M\left(\frac{h_k'(0)}{M}\right)^2 = \frac{1}{2}\frac{(h_k'(0))^2}{M} \geq \frac{\varepsilon^2\varepsilon_1^2}{2M} = \delta.$$

Hence, in view of (5.4), we have that $f(\mathbf{Q}_{k+1}) - f(\mathbf{Q}_k) = \max_\theta h_k(\theta) - h_k(0) \geq h_k\left(\frac{h_k'(0)}{M}\right) - h_k(0) \geq \delta$, and the claim is proven. $\square$

THEOREM 5.4. *Every accumulation point of the sequence $\{\mathbf{Q}_j\}_{j\geq 1}$ constructed by Algorithm 1 is a stationary point of f.*

Proof. The proof is based on Polak's theorem [36, §1.3, Th. 3].

Suppose $\overline{\mathbf{Q}}$ is an accumulation point of Algorithm 1. Then there exist a subsequence of $\{\mathbf{Q}_j\}_{j\geq 1}$ converging to $\overline{\mathbf{Q}}$, i.e., $\{\mathbf{Q}_j\}_{j\in K} \to \overline{\mathbf{Q}}$, where K is the index set of the subsequence.

Suppose that $\overline{\mathbf{Q}}$ is not a stationary point of f, i.e., $\operatorname{grad} f(\overline{\mathbf{Q}}) \neq 0$. From Lemma 5.3, there exist $\eta > 0$ and $\delta > 0$ such that

$$f(\mathbf{Q}_{k+1})) - f(\mathbf{Q}_k) \geq \delta > 0,$$

for all k such that $\|\overline{\mathbf{Q}} - \mathbf{Q}_k\| \leq \eta$. Since $\{\mathbf{Q}_j\}_{j\in K}$ converges to $\overline{\mathbf{Q}}$, there exists an $l \in K$ such that for all $j \geq l, j \in K$,

$$\|\overline{\mathbf{Q}} - \mathbf{Q}_j\| \leq \eta.$$

Thus for any two consecutive points $\mathbf{Q}_j, \mathbf{Q}_{j+i}$ of the subsequence, with $j, j+i \in K$; $j \geq l$ we have

$$f(\mathbf{Q}_{j+i}) - f(\mathbf{Q}_j) \geq f(\mathbf{Q}_{j+1}) - f(\mathbf{Q}_j) \geq \delta > 0.$$

Thus, the sequence $\{f(\mathbf{Q}_j)\}_{j\in K}$ is not a Cauchy sequence so it does not converge. On the other hand, since f is continuous and $\{\mathbf{Q}_j\}_{j\in K}$ converges to $\overline{\mathbf{Q}}$, $\{f(\mathbf{Q}_j)\}_{j\in K}$ should converge to $f(\overline{\mathbf{Q}})$. This is a contradiction, which proves the theorem. $\square$

6. Conclusions. In this paper, we have developed an algorithm for solving the best low multilinear rank approximation problem in the symmetric case. The main idea of the algorithm is to modify the given symmetric tensor by simultaneously applying Jacobi rotations on all modes of the tensor. The main subproblem reduces to finding the point at which a polynomial of degree six is maximized. The main computational cost is due to the need of updating (parts of) the tensor at each rotation. With respect to computational efficiency, our algorithm is of the same class as the currently widely used HOOI.

One advantage of the Jacobi-based algorithm is that it preserves symmetry. HOOI on the other hand destroys the symmetry during the iterations although it seems to converge to a symmetric solution. Whether HOOI always converges to a symmetric solution remains an open question. Empirical evidence as well as Friedland's recent result for rank-$(1, 1, 1)$ approximations [20] suggest that the answer might be positive. But the counterexample for the case of partial symmetry (Section 4.2) suggests that symmetry preservation should not be seen as granted. In any case, preserving symmetry throughout the iteration can be viewed as an asset in case of early stopping, and it is known that HOOI does not have this property while the proposed Jacobi-based algorithm does. Although it is easy to acquire symmetric versions of existing algorithms, these versions are not necessarily reliable. The symmetric version of HOOI where the symmetry is preserved at each step has convergence problems in several cases.

Another benefit of our algorithm is its convergence behavior. We have proved that it converges to stationary points. Moreover, the algorithm converges generically to local maxima of the maximization problem. The convergence theory of the Jacobi-based algorithm can also be seen as an advantage over the recently proposed symmetric quasi-Newton algorithm [39].

REFERENCES

[1] P.-A. Absil, R. Mahony, and R. Sepulchre, *Optimization Algorithms on Matrix Manifolds*, Princeton University Press, Princeton, NJ, January 2008.

[2] C. A. Andersson and R. Bro, *Improving the speed of multi-way algorithms: Part I. Tucker3*, Chemometrics and Intelligent Laboratory Systems, 42 (1998), pp. 93–103.

[3] P. B. Borckmans, M. Ishteva, and P.-A. Absil, *A modified particle swarm optimization algorithm for the best low multilinear rank approximation of higher-order tensors*, in 7th International Conference on Swarm Intelligence (ANTS 2010), Brussels, Belgium, 2010, pp. 13–23.

[4] J.-F. Cardoso and A. Souloumiac, *Blind beamforming for non gaussian signals*, IEE Proceedings-F, 140 (1993), pp. 362–370.

[5] J.-F. Cardoso and A. Souloumiac, *Jacobi angles for simultaneous diagonalization*, SIAM J. Matrix Anal. Appl., 17 (1996), pp. 161–164.

[6] J. Carroll and J. Chang, *Analysis of individual differences in multidimensional scaling via an N-way generalization of "Eckart-Young" decomposition*, Psychometrika, 35 (1970), pp. 283–319.

[7] A. Cichocki, R. Zdunek, A. Phan, and S. Amari, *Nonnegative Matrix and Tensor Factorizations*, Wiley, 2009.

[8] P. Comon, *Independent component analysis, a new concept?*, Signal Processing, 36 (1994), pp. 287–314. Special issue on Higher-Order Statistics.

[9] 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. Special Issue on Tensor Decompositions and Applications.

[10] L. De Lathauwer, *First-order perturbation analysis of the best rank-(R_1, R_2, R_3) approximation in multilinear algebra*, J. Chemometrics, 18 (2004), pp. 2–11.

[11] ———, *A survey of tensor methods*, in Proc. of the 2009 IEEE International Symposium on Circuits and Systems (ISCAS 2009), Taipei, Taiwan, May 2009, pp. 2773–2776.

[12] L. De Lathauwer and J. Castaing, *Tensor-based techniques for the blind separation of DS-CDMA signals*, Signal Processing, 87 (2007), pp. 322–336. Special Issue on Tensor Signal Processing.

[13] L. De Lathauwer, B. De Moor, and J. Vandewalle, *An introduction to independent component analysis*, J. Chemometrics, 14 (2000), pp. 123–149. Special Issue: Multi-way Analysis.

[14] ———, *A multilinear singular value decomposition*, SIAM J. Matrix Anal. Appl., 21 (2000), pp. 1253–1278.

[15] ———, *On the best rank-1 and rank-$(R_1, R_2, \ldots, R_N)$ approximation of higher-order tensors*, SIAM J. Matrix Anal. Appl., 21 (2000), pp. 1324–1342.

[16] ———, *Independent component analysis and (simultaneous) third-order tensor diagonalization*, IEEE Trans. Signal Process., 49 (2001), pp. 2262–2271.

[17] ———, *Computation of the canonical decomposition by means of a simultaneous generalized Schur decomposition*, SIAM J. Matrix Anal. Appl., 26 (2004), pp. 295–327.

[18] L. De Lathauwer and J. Vandewalle, *Dimensionality reduction in higher-order signal processing and rank-$(R_1, R_2, \ldots, R_N)$ reduction in multilinear algebra*, Linear Algebra Appl., 391 (2004), pp. 31–55. Special Issue on Linear Algebra in Signal and Image Processing.

[19] L. Eldén and B. Savas, *A Newton–Grassmann method for computing the best multi-linear rank-(r_1, r_2, r_3) approximation of a tensor*, SIAM J. Matrix Anal. Appl., 31 (2009), pp. 248–271.

[20] S. Friedland, *Best rank one approximation of real symmetric tensors can be chosen symmetric*, Tech. Report arXiv:1110.5689v1, 2011. http://arxiv.org/abs/1110.5689v1.

[21] G. H. Golub and C. F. Van Loan, *Matrix Computations*, Johns Hopkins University Press, Baltimore, Maryland, 3rd ed., 1996.

[22] R. A. Harshman, *Foundations of the PARAFAC procedure: Model and conditions for an "explanatory" multi-mode factor analysis*, UCLA Working Papers in Phonetics, 16 (1970), pp. 1–84.

[23] F. L. Hitchcock, *The expression of a tensor or a polyadic as a sum of products*, Journal of Mathematical Physics, 6 (1927), pp. 164–189.

[24] ———, *Multiple invariants and generalized rank of a p-way matrix or tensor*, Journal of Mathematical Physics, 7 (1927), pp. 39–79.

[25] M. Ishteva, P.-A. Absil, S. Van Huffel, and L. De Lathauwer, *Best low multilinear rank approximation of higher-order tensors, based on the Riemannian trust-region scheme*, SIAM J. Matrix Anal. Appl., 32 (2011), pp. 115–135.

[26] ———, *Tucker compression and local optima*, Chemometrics and Intelligent Laboratory Systems, 106 (2011), pp. 57–64.

[27] M. Ishteva, L. De Lathauwer, P.-A. Absil, and S. Van Huffel, *Differential-geometric Newton method for the best rank-(R_1, R_2, R_3) approximation of tensors*, Numerical Algorithms, 51 (2009), pp. 179–194. Tributes to Gene H. Golub Part II.

[28] E. Kofidis and P. A. Regalia, *On the best rank-1 approximation of higher-order supersymmetric tensors*, SIAM J. Matrix Anal. Appl., 23 (2002), pp. 863–884.

[29] T. G. Kolda and B. W. Bader, *Tensor decompositions and applications*, SIAM Review, 51 (2009), pp. 455–500.

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

[31] P. M. Kroonenberg and J. de Leeuw, *Principal component analysis of three-mode data by means of alternating least squares algorithms*, Psychometrika, 45 (1980), pp. 69–97.

[32] P. McCullagh, *Tensor Methods in Statistics*, Chapman and Hall, London, 1987.

[33] C. D. Moravitz Martin and C. F. Van Loan, *A Jacobi-type method for computing orthogonal tensor decompositions*, SIAM J. Matrix Anal. Appl., 30 (2008), pp. 1219–1232. Special Issue on Tensor Decompositions and Applications.

[34] C. L. Nikias and J. M. Mendel, *Signal processing with higher-order spectra*, IEEE Signal Process. Mag., 10 (1993), pp. 10–37.

[35] C. L. Nikias and A. P. Petropulu, *Higher-Order Spectra Analysis. A Nonlinear Signal Processing Framework*, Prentice Hall, Englewood Cliffs, New Jersey, 1993.

[36] E. Polak, *Computational Methods in Optimization. A Unified Approach*, Mathematics in Science and Engineering, Vol. 77, Academic Press, New York, 1971.

[37] P. A. Regalia and E. Kofidis, *The higher-order power method revisited: convergence proofs and effective initialization*, in Proceedings of the Acoustics, Speech, and Signal Processing, 2000. on IEEE International Conference - Volume 05, Washington, DC, USA, 2000, IEEE Computer Society, pp. 2709–2712.

[38] B. Savas and L. Eldén, *Krylov subspace methods for tensor computations*, Tech. Report LITH-MAT-R-2009-02-SE, Department of Mathematics, Linköping University, 2009.

[39] B. Savas and L.-H. Lim, *Quasi-Newton methods on Grassmannians and multilinear approxi-*

mations of tensors, SIAM Journal on Scientific Computing, 32 (2010), pp. 3352–3393.

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

[41] A. Swami and G. Giannakis, *Bibliography on HOS*, Signal Process., 60 (1997), pp. 65–126.

[42] J. ten Berge, J. de Leeuw, and P. Kroonenberg, *Some additional results on principal components analysis of three-mode data by means of alternating least squares algorithms*, Psychometrika, 52 (1987), pp. 183–191.

[43] L. R. Tucker, *The extension of factor analysis to three-dimensional matrices*, in Contributions to mathematical psychology, H. Gulliksen and N. Frederiksen, eds., Holt, Rinehart & Winston, NY, 1964, pp. 109–127.

[44] ———, *Some mathematical notes on three-mode factor analysis*, Psychometrika, 31 (1966), pp. 279–311.