This file is a preprint. The original publication is available at
http://www.springer.com/engineering/book/978-3-642-12597-3,
Recent Advances in Optimization and its Applications in Engineering, Springer,
November 2010, pp. 145–164. (invited overview paper)

On the best low multilinear rank approximation of higher-order tensors *

Mariya Ishteva[1], P.-A. Absil[1],
Sabine Van Huffel[2], and Lieven De Lathauwer[2,3]

[1] Department of Mathematical Engineering, Université catholique de Louvain, Bâtiment Euler - P13, Av. Georges Lemaître 4, 1348 Louvain-la-Neuve, Belgium, mariya.ishteva@uclouvain.be, http://www.inma.ucl.ac.be/~absil
[2] Department of Electrical Engineering - ESAT/SCD, K.U.Leuven, Kasteelpark Arenberg 10/2446, 3001 Leuven, Belgium, sabine.vanhuffel@esat.kuleuven.be
[3] Group Science, Engineering and Technology, K.U.Leuven Campus Kortrijk, E. Sabbelaan 53, 8500 Kortrijk, Belgium, lieven.delathauwer@kuleuven-kortrijk.be

This paper deals with the best low multilinear rank approximation of higher-order tensors. Given a tensor, we are looking for another tensor, as close as possible to the given one and with bounded multilinear rank. Higher-order tensors are used in higher-order statistics, signal processing, telecommunications and many other fields. In particular, the best low multilinear rank approximation is used as a tool for dimensionality reduction and signal subspace estimation.

Computing the best low multilinear rank approximation is a nontrivial task. Higher-order generalizations of the singular value decomposition lead to suboptimal solutions. The higher-order orthogonal iteration is a widely used linearly convergent algorithm for further refinement. We aim for conceptually faster algorithms. However, applying standard optimization algorithms directly is not a good idea since there are infinitely many equivalent solutions. Nice convergence properties are observed when the solutions are isolated. The present invariance can be removed by working on quotient manifolds. We discuss three algorithms, based on Newton's method, the trust-region scheme and conjugate gradients. We also comment on the local minima of the problem.

* Research supported by: (1) The Belgian Federal Science Policy Office: IUAP P6/04 (DYSCO, "Dynamical systems, control and optimization", 2007–2011), (2) Communauté française de Belgique - Actions de Recherche Concertées, (3) Research Council K.U.Leuven: GOA-AMBioRICS, GOA-MaNet, CoE EF/05/006 Optimization in Engineering (OPTEC), (4) F.W.O. project G.0321.06, "Numerical tensor methods for spectral analysis". (5) "Impulsfinanciering Campus Kortrijk (2007–2012)(CIF1)" and STRT1/08/023. Part of this research was carried out while M. Ishteva was with K.U.Leuven, supported by OE/06/25, OE/07/17, OE/08/007, OE/09/004. The scientific responsibility rests with the authors.

1 Introduction

Multilinear algebra deals with higher-order tensors, generalizations of vectors and matrices to higher-dimensional tables of numbers. Tensor algebra is more involved than matrix algebra but can model more complex processes. Higher-order tensors are used in many application fields so efficient and reliable algorithms for handling them are required.

Matrices are second-order tensors with well-studied properties. The matrix rank is a well-understood concept. In particular, the low-rank approximation of a matrix is essential for various results and algorithms. However, the matrix rank and its properties are not easily or uniquely generalizable to higher-order tensors. The rank, the row rank and the column rank of a matrix are equivalent whereas in multilinear algebra these are in general different.

Of main concern for this paper is the multilinear rank [40, 41] of a tensor, which is a generalization of column and row rank of a matrix. In particular, we discuss algorithms for the best low multilinear rank approximation of a higher-order tensor. The result is a higher-order tensor, as close as possible to the original one and having bounded multilinear rank. In the matrix case, the solution is given by the truncated singular value decomposition (SVD) [34, §2.5]. In multilinear algebra, the truncated higher-order SVD (HOSVD) [22] gives a suboptimal approximation, which can be refined by iterative algorithms. The traditional algorithm for this purpose is the higher-order orthogonal iteration (HOOI) [23, 52, 53]. In this paper, we discuss conceptually faster algorithms based on the Newton method, trust-region scheme and conjugate gradients. We also comment on the fact that numerical methods converge to local minimizers [44] of the function associated with the best low multilinear approximation.

It will be shown that the cost function has an invariance property by the action of the orthogonal group. Conceptually speaking, the solutions are not isolated, i.e., there are whole groups of infinitely many equivalent elements. This is a potential obstacle for algorithms since in practice, convergence to one particular point has to be achieved. Differential geometric techniques remove successfully the mentioned invariance. The working spaces are quotient manifolds. The elements of such spaces are sets containing points that are in some sense equivalent. For our particular problem, we work with matrices but in practice we are only interested in their column space. There are infinitely many matrices with the same column space that can be combined in one compound element of a quotient space. Another possibility is to first restrict the set of all considered matrices to the set of matrices with column-wise orthonormal columns and then combine all equivalent matrices from the selected ones in one element. This is justified by the fact that any subspace can be represented by the column space of a column-wise orthonormal matrix. We consider both options. We can summarize that in this paper, a multilinear algebra optimization problem is solved using optimization on manifolds.

This paper is an overview of recent publications and technical reports

[47, 46, 43, 44, 45] and the PhD thesis [42]. We present a digest of current research results, a survey of the literature on the best low multilinear rank approximation problem and other tensor approximations and discuss some applications. The paper is organized as follows. In Section 2, some definitions and properties of higher-order tensors are given. The main problem is formulated, HOSVD and HOOI are presented and we also mention some other related algorithms from the literature. Some applications are demonstrated in Section 3. Three differential-geometric algorithms are discussed in Section 4. In Section 5, we talk about local minima. Conclusions are drawn in Section 6.

In this paper we consider third-order tensors. The differences in the properties and algorithms for third-order tensors and for tensors of order higher than three are mainly technical, whereas the differences between the matrix case and the case of third-order tensors are fundamental.

2 Background material

2.1 Basic definitions

An Nth-order tensor is an element of the tensor product of N vector spaces. When the choice of basis is implicit, we think of a tensor as its representation as an N-way array [28]. Each "direction" of an Nth order tensor is called a mode. The vectors, obtained by varying the nth index, while keeping the other indices fixed are called mode-n vectors ($n = 1, 2, \ldots, N$). For a tensor $\mathcal{A} \in \mathbb{R}^{6\times5\times4}$ they are visualized in Fig. 1. The mode-n rank of a tensor $\mathcal{A}$,

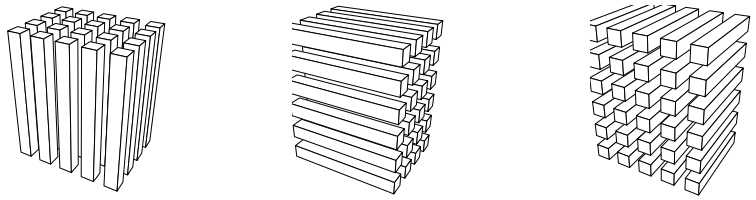


Fig. 1. Mode-n vectors of a $(6 \times 5 \times 4)$-tensor.

denoted by $\text{rank}_n(\mathcal{A})$, is defined as the number of linearly independent mode-n vectors. The multilinear rank of a tensor is then the n-tuple of the mode-n ranks. An essential difference with the matrix case is that the mode-n ranks are in general different from each other.

We use the following definition of mode-n products $\mathcal{A} \bullet_n \mathbf{M}^{(n)}$, $n = 1, 2, 3$ of a tensor $\mathcal{A} \in \mathbb{R}^{I_1\times I_2\times I_3}$ and matrices $\mathbf{M}^{(n)} \in \mathbb{R}^{J_n\times I_n}$:

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

where $1 \leq i_n \leq I_n$, $1 \leq j_n \leq J_n$. In this notation, $\mathbf{A} = \mathbf{U}\mathbf{M}\mathbf{V}^T$ is presented as $\mathbf{A} = \mathbf{M} \bullet_1 \mathbf{U} \bullet_2 \mathbf{V}$. This is reasonable since the columns of $\mathbf{U}$ correspond to the column space of $\mathbf{A}$ in the same way as the columns of $\mathbf{V}$ correspond to the row space of $\mathbf{A}$. The mode-n product has the following properties

$$(\mathcal{A} \bullet_n \mathbf{U}) \bullet_m \mathbf{V} = (\mathcal{A} \bullet_m \mathbf{V}) \bullet_n \mathbf{U} = \mathcal{A} \bullet_n \mathbf{U} \bullet_m \mathbf{V}, \quad m \neq n,$$
$$(\mathcal{A} \bullet_n \mathbf{U}) \bullet_n \mathbf{V} = \mathcal{A} \bullet_n (\mathbf{V}\,\mathbf{U}).$$

It is often useful to represent a tensor in a matrix form, e.g., by putting all mode-n vectors one after the other in a specific order. For a tensor $\mathcal{A} \in \mathbb{R}^{I_1 \times I_2 \times I_3}$, the matrix representations $\mathbf{A}_{(n)}, n = 1, 2, 3$ that we use are

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

where $1 \leq i_n \leq I_n$. This definition is illustrated in Fig. 2 for $I_1 > I_2 > I_3$.

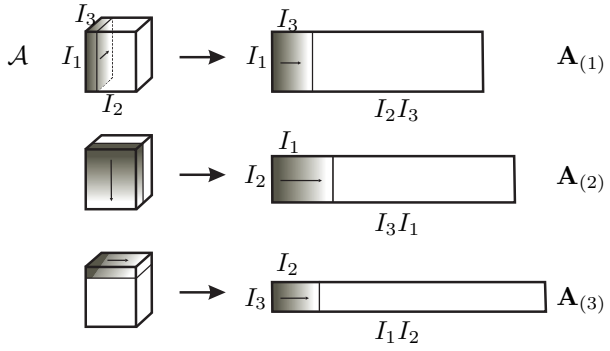


Fig. 2. Matrix representations of a tensor.

2.2 Best low multilinear rank approximation

Given $\mathcal{A} \in \mathbb{R}^{I_1 \times I_2 \times I_3}$, its best rank-$(R_1, R_2, R_3)$ approximation is a tensor $\hat{\mathcal{A}} \in \mathbb{R}^{I_1 \times I_2 \times I_3}$, such that it *minimizes* the cost function $f : \mathbb{R}^{I_1 \times I_2 \times I_3} \to \mathbb{R}$,

$$f : \hat{\mathcal{A}} \mapsto \|\mathcal{A} - \hat{\mathcal{A}}\|^2 \tag{1}$$

under the constraints $\text{rank}_1(\hat{\mathcal{A}}) \leq R_1$, $\text{rank}_2(\hat{\mathcal{A}}) \leq R_2, \text{rank}_3(\hat{\mathcal{A}}) \leq R_3$. This problem is equivalent [23, 52, 53] to the problem of *maximizing* the function

$$\begin{aligned} &\overline{g} : St(R_1, I_1) \times St(R_2, I_2) \times St(R_3, I_3) \to \mathbb{R}, \\ &\quad (\mathbf{U}, \mathbf{V}, \mathbf{W}) \mapsto \|\mathcal{A} \bullet_1 \mathbf{U}^T \bullet_2 \mathbf{V}^T \bullet_3 \mathbf{W}^T\|^2 = \|\mathbf{U}^T \mathbf{A}_{(1)}(\mathbf{V} \otimes \mathbf{W})\|^2 \end{aligned} \tag{2}$$

over the matrices $\mathbf{U}, \mathbf{V}$ and $\mathbf{W}$ ($St(p, n)$ stands for the set of column-wise orthonormal $(n \times p)$-matrices, $\|\cdot\|$ is the Frobenius norm and $\otimes$ denotes the Kronecker product). This equivalence is a direct generalization of the

matrix case where finding the best rank-R approximation $\hat{\mathbf{A}} = \mathbf{U}\,\mathbf{B}\,\mathbf{V}^T$ of $\mathbf{A} \in \mathbb{R}^{I_1 \times I_2}$, where $\mathbf{B} \in \mathbb{R}^{R \times R}$, $\mathbf{U} \in St(R, I_1)$, $\mathbf{V} \in St(R, I_2)$ and $\|\mathbf{A} - \hat{\mathbf{A}}\|$ is minimized, is equivalent to the maximization of $\|\mathbf{U}^T\,\mathbf{A}\,\mathbf{V}\| = \|\mathbf{A} \bullet_1 \mathbf{U}^T \bullet_2 \mathbf{V}^T\|$. Having estimated $\mathbf{U}, \mathbf{V}$ and $\mathbf{W}$ in (2), the solution of (1) is computed by

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

Thus, in this paper, our goal is to solve the maximization problem (2). In practice, the function $-\overline{g}$ will be minimized.

2.3 Higher-order singular value decomposition

The SVD [34, §2.5] gives the best low-rank approximation of a matrix. In the sense of multilinear rank, a generalization of the SVD is the higher-order SVD (HOSVD) [22]. With possible variations it is also known as Tucker decomposition [72, 73]. HOSVD decomposes a tensor $\mathcal{A} \in \mathbb{R}^{I_1 \times I_2 \times I_3}$ as

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

where $\mathcal{S} \in \mathbb{R}^{I_1 \times I_2 \times I_3}$ and where $\mathbf{U}^{(n)} \in \mathbb{R}^{I_n \times I_n}$, $n = 1, 2, 3$, are orthogonal, see Fig. 3. The matrices obtained from $\mathcal{S}$ by fixing any of the indices are

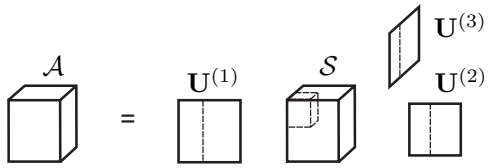


Fig. 3. Higher-order singular value decomposition.

orthogonal to each other and their norm is decreasing with increasing the fixed index. The mode-n singular values of $\mathcal{A}$ are the singular values of $\mathbf{A}_{(n)}$.

For second-order tensors, i.e., matrices, HOSVD reduces to the well-known SVD. However, truncation of HOSVD results in a suboptimal solution of the best low multilinear rank approximation problem. This is due to the fact that in general, it is impossible to obtain a diagonal $\mathcal{S}$ tensor. The number of degrees of freedom in such a decomposition would be smaller than the number of the elements of the tensor that needs to be decomposed. However, the truncated HOSVD can serve as a good starting point for iterative algorithms.

Other generalizations of the matrix SVD have been discussed in the literature, focusing on different properties of the SVD. The tensor corresponding to $\mathcal{S}$ can be made as diagonal as possible (in a least squares sense) under orthogonal transformations [12, 24, 56, 10], or the original tensor can be decomposed in a minimal number of rank-1 terms (CANDECOMP/PARAFAC) [13, 37, 9, 25, 17], on which orthogonal [50] or symmetry [14] constraints can be imposed. A unifying framework for Tucker/HOSVD and CANDECOMP/PARAFAC is given by the block term decompositions [18, 19, 26].

2.4 Higher-order orthogonal iteration

The traditional iterative algorithm for maximizing (2) and thus minimizing (1) is the higher-order orthogonal iteration (HOOI) [23, 52, 53]. It is an alternating least-squares (ALS) algorithm. At each step the estimate of one of the matrices $\mathbf{U}, \mathbf{V}, \mathbf{W}$ is optimized, while the other two are kept constant. The function $\overline{g}$ from (2) is thought of as a quadratic expression in the components of the matrix that is being optimized. For fixed $\mathbf{V}$ and $\mathbf{W}$, since

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

the columns of the optimal $\mathbf{U} \in \mathbb{R}^{I_1 \times R_1}$ build an orthonormal basis for the left R_1-dimensional dominant subspace of $\mathbf{A}_{(1)}(\mathbf{V} \otimes \mathbf{W})$. It can be obtained from the SVD of $\mathbf{A}_{(1)}(\mathbf{V} \otimes \mathbf{W})$. The optimization with respect to the other two unknown matrices is performed by analogy.

Initial matrices for HOOI are often taken from the truncated HOSVD. These matrices usually belong to the attraction region of (2) but there are exceptions. Moreover, convergence to the global maximum is not guaranteed.

HOOI is a simple concept and easy to implement. Therefore it is the most widely used algorithm at the moment [51]. If we assume for simplicity that $R_1 = R_2 = R_3 = R$ and $I_1 = I_2 = I_3 = I$, the total cost for one iteration of HOOI is then $O(I^3R + IR^4 + R^6)$ [32, 47]. However, the convergence speed of HOOI is at most linear.

2.5 Other methods in the literature

Recently, a Newton-type algorithm for the best low multilinear rank approximation of tensors has been proposed in [32]. It works on the so-called Grassmann manifold whereas the Newton-type algorithm considered in this paper is a generalization of the ideas behind the geometric Newton method for Oja's vector field [2]. Quasi-Newton methods have been suggested in [64].

We also mention other related methods. A Krylov method for large sparse tensors has been proposed in [63]. In [23, 75, 49], specific algorithms for the best rank-1 approximation have been discussed. Fast HOSVD algorithms for symmetric, Toeplitz and Hankel tensors have been proposed in [7]. For tensors with large dimensions, Tucker-type decompositions are developed in [59, 8, 54].

3 Some applications

The best low multilinear rank approximation of tensors is used for signal subspace estimation [60, 61, 52, 67, 51, 35] and as a dimensionality reduction tool for tensors with high dimensions [27, 4, 29, 30, 52, 67, 51], including simultaneous dimensionality reduction of a matrix and a tensor [27].

Independent component analysis (ICA) [27] extracts statistically independent sources from a linear mixture in fields like electroencephalography

(EEG), magnetoencephalography (MEG) and nuclear magnetic resonance (NMR). Sometimes only a few sources have significant contributions. A principal component analysis (PCA)-based prewhitening step for reducing the dimensionality is often used. This is beneficial if white Gaussian noise is present but is not applicable in case of colored Gaussian noise. In the latter case, low multilinear rank approximation of a higher-order cumulant tensor of the observation vector can be performed instead. The dimensionality of the problem is reduced from the number of observation channels to the number of sources.

A rank-1 tensor is an outer product of a number of vectors. The decomposition of higher-order tensors in rank-1 terms is called parallel factor decomposition (PARAFAC) [37] or canonical decomposition (CANDECOMP) [9]. It has applications in chemometrics [67], wireless communication [66, 21], and can also be used for epileptic seizure onset localization [30, 29, 4], since only one of the rank-1 terms is related to the seizure activity. The best low multilinear rank approximation of tensors is often used as a dimensionality reduction step preceding the actual computation of PARAFAC. Such a preprocessing step is implemented for example in the N-way toolbox for MATLAB [6].

Dimensionality reduction works as illustrated in Fig. 4. See also [16, Remark 6.2.2]. Let the rank-R decomposition of $\mathcal{A} \in \mathbb{R}^{I_1 \times I_2 \times I_3}$ be required. If

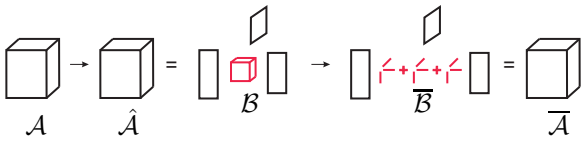


Fig. 4. Dimensionality reduction.

$R < \max(I_1, I_2, I_3)$, then a reduction of $\mathcal{A}$ to a tensor $\mathcal{B} \in \mathbb{R}^{I_1' \times I_2' \times I_3'}$, $I_n' = \min(I_n, R), n = 1, 2, 3$ can be used for the actual computation of PARAFAC. This can be done as follows. Let $\hat{\mathcal{A}}$ be the best rank-(I_1', I_2', I_3') approximation of $\mathcal{A}$. If $\mathbf{U}, \mathbf{V}, \mathbf{W}$ are the matrices as in (2), i.e., if

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

then a rank-R approximation $\overline{\mathcal{A}}$ of $\mathcal{A}$ is computed from the best rank-R approximation $\overline{\mathcal{B}}$ of $\mathcal{B}$ in the following way

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

Tensor $\mathcal{B}$ has smaller dimensions than $\mathcal{A}$ so that computing $\overline{\mathcal{B}}$ is much less expensive than directly computing $\overline{\mathcal{A}}$. In practice, due to numerical problems, in some applications $I_n' = \min(I_n, R + 2), n = 1, 2, 3$ are used instead of the dimensions $I_n' = \min(I_n, R)$. In general, it is advisable to examine the mode-n singular values for gaps between them and use a corresponding low multilinear rank approximation. It might also be useful to perform a few additional PARAFAC steps on $\overline{\mathcal{A}}$ in order to find an even better approximation of $\mathcal{A}$.

In signal processing applications, a signal is often modeled as a sum of exponentially damped sinusoids (EDS). The parameters of the model have to be estimated given only samples of the signal. In the literature there are both matrix [31, 74] and tensor-based algorithms [60, 61]. The latter are based on the best rank-(R_1, R_2, R_3) approximation. In [48], the EDS model in the multi-channel case is considered in the case of closely spaced poles. This problem is more difficult than the case where the poles are well separated. A comparison of the performance of a matrix-based and a tensor-based method was performed. None of them always outperforms the other one. However, in the tensor-based algorithm, one can choose the mode-3 rank in such a way that the performance is optimal. Numerical experiments indicate that if ill-conditioning is present in the mode corresponding to the complex amplitudes, taking a lower value for the mode-3 rank than for the mode-1 and mode-2 ranks improves the performance of the tensor method to the extent that it outperforms the matrix method.

For more references and application areas, we refer to the books [67, 52, 11], to the overview papers [51, 20] and to the references therein.

4 Algorithms

In this section, we will review three classical optimization algorithms adapted for quotient matrix manifolds. We will then show how these algorithms can be applied on the best low multilinear rank approximation problem.

4.1 Geometric Newton algorithm

In order to apply Newton's method, the solutions of the optimization problem (2) have to be reformulated as zeros of a suitable function. The matrix $\mathbf{U} \in St(R_1, I_1)$ is optimal if and only if [38, Th. 3.17] its column space is the R_1-dimensional left dominant subspace of $\mathbf{A}_{(1)}(\mathbf{V} \otimes \mathbf{W})$. A necessary condition for this is that the column space of $\mathbf{U}$ is an invariant subspace of $\mathbf{A}_{(1)}(\mathbf{V} \otimes \mathbf{W})(\mathbf{V} \otimes \mathbf{W})^T \mathbf{A}_{(1)}^T$. Defining $\mathbf{X} = (\mathbf{U}, \mathbf{V}, \mathbf{W})$ and

$$\mathbf{R}_1(\mathbf{X}) = \mathbf{U}^T \mathbf{A}_{(1)}(\mathbf{V} \otimes \mathbf{W}),$$

this condition can be written as

$$F_1(\mathbf{X}) \equiv \mathbf{U}\, \mathbf{R}_1(\mathbf{X})\mathbf{R}_1(\mathbf{X})^T - \mathbf{A}_{(1)}(\mathbf{V} \otimes \mathbf{W})\mathbf{R}_1(\mathbf{X})^T = \mathbf{0}.$$

In the same way two more conditions are obtained for the matrices $\mathbf{V}$ and $\mathbf{W}$. The new function is then

$$\begin{aligned} F : &\mathbb{R}^{I_1 \times R_1} \times \mathbb{R}^{I_2 \times R_2} \times \mathbb{R}^{I_3 \times R_3} \to \mathbb{R}^{I_1 \times R_1} \times \mathbb{R}^{I_2 \times R_2} \times \mathbb{R}^{I_3 \times R_3}, \\ &\mathbf{X} \mapsto (F_1(\mathbf{X}),\, F_2(\mathbf{X}),\, F_3(\mathbf{X})). \end{aligned} \tag{3}$$

Newton's method can be applied for finding the zeros of F. However, F_1 has an invariance property

$$F_1(\mathbf{XQ}) = F_1(\mathbf{X})\,\mathbf{Q}_1, \tag{4}$$

where $\mathbf{XQ} = (\mathbf{UQ}_1, \mathbf{VQ}_2, \mathbf{WQ}_3)$ and $\mathbf{Q}_i \in O_{R_i}, i = 1, 2, 3$ are orthogonal matrices. The functions F_2 and F_3 have similar properties, i.e.,

$$F(\mathbf{X}) = \mathbf{0} \quad \Longleftrightarrow \quad F(\mathbf{XQ}) = \mathbf{0}.$$

Thus, the zeros of F are not isolated, which means that the plain Newton method is expected to have difficulties (see, for example, [3, Prop. 2.1.2], [2]).

A solution to this problem is to combine equivalent solutions in one element and work on the obtained quotient manifold (see [3] for the general theory on optimization on matrix manifolds). For information on differential-geometric version of Newton's method see also [5]. If we perform as little quotienting as possible in order to isolate the zeros, we obtain the quotient set

$$M = \mathbb{R}_*^{I_1 \times R_1}/O_{R_1} \times \mathbb{R}_*^{I_2 \times R_2}/O_{R_2} \times \mathbb{R}_*^{I_3 \times R_3}/O_{R_3}. \tag{5}$$

$\mathbb{R}_*^{n \times p}$ is the set of all full-rank $(n \times p)$-matrices, $n \geq p$ and each element $[\mathbf{U}]$ of $\mathbb{R}_*^{I_1 \times R_1}/O_{R_1}$ is a set of all matrices that can be obtained by multiplying $\mathbf{U}$ from the right by an orthogonal matrix. Any two sets $[\mathbf{U}_1]$ and $[\mathbf{U}_2]$ are either disjoint or coincide and the union of all such sets equals $\mathbb{R}_*^{n \times p}$. They are called equivalence classes. In each equivalence class all elements have the same column space.

For our problem (2), working on the manifold M removes the invariance and leads to a differential-geometric Newton algorithm [47]. The Newton algorithm has local quadratic convergence to the nondegenerate zeros of the vector field ξ on M (5) represented by the horizontal lift $P^h F$,

$$P^h_{\mathbf{U}}(\mathbf{Z}_\mathbf{U}) = \mathbf{Z}_\mathbf{U} - \mathbf{U}\,\mathrm{skew}((\mathbf{U}^T\mathbf{U})^{-1}\mathbf{U}^T\mathbf{Z}_\mathbf{U}),$$

where $\mathrm{skew}(\mathbf{B}) = (\mathbf{B} - \mathbf{B}^T)/2$. If $\mathbf{X}_*$ is a zero of F (3), then $[\mathbf{X}_*]$ is a zero of ξ. Numerical results indicate that that nondegeneracy holds under generic conditions.

Numerical examples also confirmed the fast quadratic convergence of the algorithm in the neighborhood of the solution. However, the cost per iteration of the geometric Newton algorithm $O(I^3R^3)$ is higher than the cost $O(I^3R + IR^4 + R^6)$ for one HOOI iteration. Another possible disadvantage of the proposed algorithm is that it does not necessarily converge to a local maximum of (2) since not all zeros of F correspond to local maxima of (2). In theory, Newton's method can even diverge. However, this was not observed in numerical experiments. To increase the chances of converging to a maximum of (2), one can first perform an HOSVD followed by a few iterations of HOOI and additionally check for the negative definiteness of the Hessian before starting the Newton algorithm.

4.2 Trust-region based algorithm

Another iterative method for minimizing a cost function is the trust-region method [15, 58]. At each step, instead of working with the original function, a quadratic model is obtained. This model is assumed to be accurate in a neighborhood (the trust-region) of the current iterate. The solution of the quadratic minimization problem is suggested as a solution of the original problem. The quality of the updated iterate is evaluated and is accepted or rejected. The trust-region radius is also adjusted.

On a Riemannian manifold, the trust-region subproblem at a point $\mathbf{x} \in M$ is moved to the tangent plane $T_{\mathbf{x}}M$. The tangent plane is a Euclidean space so the minimization problem can be solved with standard algorithms. The update vector $\xi \in T_{\mathbf{x}}M$ is a tangent vector, giving the direction in which the next iterate is to be found and the size of the step. However, the new iterate has to be on the manifold and not on the tangent plane. The correspondence between vectors on the tangent plane and points on the manifold is given by a retraction [65, 5], Fig. 5.

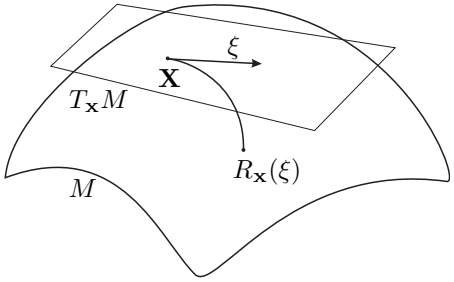


Fig. 5. Retraction.

The choice of retraction is important. The first obvious choice is the exponential map. However, depending on the manifold, this choice may be computationally inefficient [55]. A retraction can be thought of as a cheap approximation of the exponential map, without destroying the convergence behavior of the optimization methods.

As suggested in [70, 71], an approximate but sufficiently accurate solution to the trust-region subproblem (the minimization of the quadratic model) is given by the truncated conjugate gradient algorithm (tCG). An advantage here is that the Hessian matrix is not computed explicitly but only its application to a tangent vector is required. For other possible methods for (approximately) solving the trust-region subproblem see [57, 15].

Notice that $\overline{g}$ from (2) has the following invariance property

$$\overline{g}(\mathbf{U}, \mathbf{V}, \mathbf{W}) = \overline{g}(\mathbf{U}\mathbf{Q}_1, \mathbf{V}\mathbf{Q}_2, \mathbf{W}\mathbf{Q}_3)\,, \tag{6}$$

where $\mathbf{Q}_i \in O_{R_i}, i = 1, 2, 3$ are orthogonal matrices. This means that we are not interested in the exact elements of the matrices $\mathbf{U}, \mathbf{V}, \mathbf{W}$ but in the

subspaces that their columns span. For the Newton algorithm in Section 4.1 we worked on the manifold defined in (5). Here we choose the Grassmann manifold which removes more unused information from the cost function. In (2) we optimize three matrices so we need the product manifold

$$M = St(R_1, I_1)/O_{R_1} \times St(R_2, I_2)/O_{R_2} \times St(R_3, I_3)/O_{R_3}, \tag{7}$$

which can be thought of as a product of three Grassmann manifolds. A natural choice of a retraction is [3, §4.1.2]

$$R_{\mathbf{X}O_p}(\mathbf{Z}) = \mathrm{qf}(\mathbf{X} + \overline{\mathbf{Z}})O_p\,, \tag{8}$$

where qf denotes the **Q** factor of the thin **QR** decomposition [34, §5.2] and **Z** is a tangent vector. This choice is also motivated by the fact that we are only interested in column spaces of the matrices $\mathbf{U}, \mathbf{V}$ and $\mathbf{W}$ from (2) and not in their actual values.

In order to apply the Riemannian trust-region scheme to the problem (2), we need to go through the "checklist" in [1, §5.1] and give closed-form expressions for all the necessary components. A summary of the first version of the trust-region algorithm has been proposed in [45]. The algorithm is described in detail in [46].

The trust-region method has superlinear convergence. On the other hand, the cost for one iteration $O(I^3R^3)$ is higher than the cost for one HOOI iteration $O(I^3R + IR^4 + R^6)$ [32, 47]. However, it should be taken into account that in applications, the multilinear rank is often much smaller than the dimensions of the tensor. Moreover, one can reduce the computational cost of the trust-region algorithm without losing its fast local convergence rate. This can be done by choosing a stopping criterion based on the gradient of the cost function for the inner iteration [1]. In this case, few inner tCG steps are taken when the current iterate is far away from the solution (when the gradient is large) and more inner tCG steps are taken close to the solution. Thus, the overall performance of the trust-region method is to be preferred to HOOI in many cases.

Newton-type methods (see [47, 32, 64] and Section 4.1) also have local quadratic convergence rate and their computational cost per iteration is of the same order as the one of the trust-region method. However, they are not globally convergent and strongly depend on the initialization point. Although the truncated HOSVD often gives good initial values, sometimes these values are not good enough. These methods might even diverge in practice. On the other hand, the trust-region method converges globally (i.e., for all initial points) to stationary points [1] except for very special examples that are artificially constructed. Moreover, since the trust-region method is decreasing the cost function at each step, convergence to saddle points or local maxima is not observed in practice. Newton methods do not distinguish between minima, maxima and saddle points. Thus, if the stationary points are close to each other, even if a relatively good starting point is chosen, these algorithms might converge to a maximum or to a saddle point instead of to a minimum.

4.3 Conjugate gradient based algorithm

The linear conjugate gradient (CG) method [39] is used for solving large systems of linear equations having a symmetric positive definite matrix. One can also regard CG as a method to minimize a convex quadratic cost function. The initial search direction is taken equal to the steepest descent direction. Every subsequent search direction is required to be conjugate to all previously generated search directions. The step length is chosen as the exact minimizer in the search direction and indicates where to take the next iterate. The optimal solution is found in n steps, where n is the dimension of the problem.

Nonlinear CG methods [33, 62] use the same idea as linear CG but apply it to general nonlinear functions. A few adjustments are necessary. The step size is obtained by a line search algorithm. The computation of the next search direction is not uniquely defined as in the linear CG. The main approaches are those provided by Fletcher-Reeves [33] and Polak-Ribière [62], both having advantages and disadvantages. The nonlinear CG methods reduce to the linear CG if the function is convex quadratic and if the step size is the exact minimizer along the search direction. However, since the cost function is in general not convex quadratic, convergence is obtained after more than n iterations. Some convergence results can be found in [58, §5] and the references therein.

In order to generalize the nonlinear CG from functions in $\mathbb{R}^n$ to functions defined on Riemannian manifolds, the expressions for the step length and search direction have to be adjusted. Exact line search for the step length could be extremely expensive. In that case, the step size could be computed using a backtracking procedure, searching for an Armijo point [3, §4.2].

When computing the new search direction η_{k+1}, another obstacle appears. The formula for η_{k+1} involves the gradient at the new point $\mathbf{x}_{k+1}$ and the previous search direction η_k, which are two vectors in two different tangent spaces. A solution for this problem is to carry η_k over to the tangent space of $\mathbf{x}_{k+1}$. Nonlinear CG on Riemannian manifolds was first proposed in [68, 69]. This algorithm makes use of the exponential map and parallel translation, which might be inefficient. The algorithm proposed in [3] works with the more general concepts of retraction and vector transport. The vector transport is a mapping that transports a tangent vector from one tangent plane to another. The vector transport has a different purpose than a retraction but is a similar concept in the sense that it is a cheap version of parallel translation, being just as useful as the parallel translation at the same time. We refer to [3, Def. 8.1.1] for the precise formulation. The concept is illustrated in Fig. 6. The vector ξ is transported to the tangent plane of $R_{\mathbf{x}}(\eta)$ and the result is $\mathcal{T}_{\eta}\xi$.

As in the trust-region algorithm, here, for solving (2) we work again on the Grassmann manifold. A simple vector transport in this case is

$$\overline{(\mathcal{T}_{\eta_{\mathbf{x}}}\xi_{\mathbf{x}})}_{\mathrm{qf}(\mathbf{X}+\overline{\eta}_{\mathbf{x}})} = P^h_{\mathrm{qf}(\mathbf{X}+\overline{\eta}_{\mathbf{x}})}\overline{\xi}_{\mathbf{x}}\,, \tag{9}$$

where $\eta_{\mathbf{x}}$ and $\xi_{\mathbf{x}}$ are two tangent vectors at point $[\mathbf{X}]$ and $\overline{\xi}_{\mathbf{x}}$ and $\overline{\eta}_{\mathbf{x}}$ are the horizontal lifts [3, §3.5.8] at $\mathbf{X}$ of $\xi_{\mathbf{x}}$ and $\eta_{\mathbf{x}}$ respectively. $P^h_{\mathbf{Y}}$ is the orthogonal

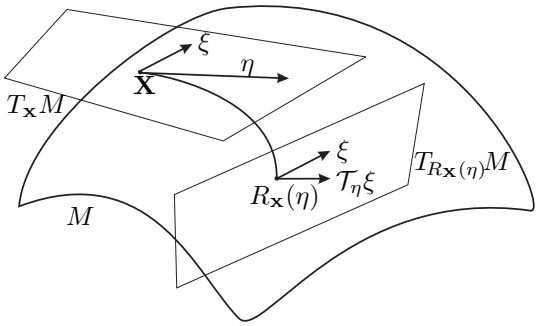


Fig. 6. Vector transport.

projection

$$P^h_{\mathbf{Y}}(\mathbf{Z}) = (\mathbf{I} - \mathbf{Y}\mathbf{Y}^T)\mathbf{Z}$$

onto the horizontal space of the point $\mathbf{Y}$. Note that $[\mathrm{qf}(\mathbf{X} + \overline{\eta}_{\mathbf{X}})] = R_{[\mathbf{X}]}\eta_x$.

Some remarks are in order. Since the step size is not the optimal one along η_k, it is possible that the new direction is not a descent direction. If this is the case, we set the new direction to be the steepest descent direction. A generalization of the computation of the search directions based on the Fletcher-Reeves and Polak-Ribière formulas is given in [3, §8.3]. The precision of CG was discussed in [3, 36]. When the distance between the current iterate and the local minimum is close to the square root of the machine precision, the Armijo condition within the line-search procedure can never be satisfied. This results in CG having maximum precision equal to the square root of the machine precision. To overcome this problem, an approximation of the Armijo condition was proposed in [36]. Finally, we mention that for better convergence results, it is advisable to "restart" the CG algorithm, i.e., to take as a search direction the steepest descent direction. This should be done at every n steps, where n is the number of unknown parameters, in order to erase unnecessary old information. The convergence of CG in $\mathbb{R}^n$ is then n-step quadratic. However, n is often too large in the sense that the algorithm already converges in less than n iterations.

The convergence properties of nonlinear CG methods are difficult to analyze. Under mild assumptions on the cost function, nonlinear CG converges to stationary points. Descent directions are guaranteed if we take the steepest descent direction when the proposed direction is not a descent direction itself. Thus, CG converges to local minima unless very special initial values are started from. The advantage of the nonlinear CG methods is their low computational cost and the fact that they do not require a lot of storage space. At each iteration, the cost function and the gradient are evaluated but the computation of the Hessian is not required, as it was the case for the trust-region algorithm from Section 4.2.

It is expected that the proposed geometric CG algorithm [43] has properties similar to those of nonlinear CG although theoretical results are difficult

to prove. Numerical experiments indicate that the performance of CG strongly depends on the problem. If the tensor has a well-determined part with low multilinear rank, CG performs well. The difficulty of the problem is related to the distribution of the multilinear singular values of the original tensor. As far as the computational time is concerned, CG seems to be competitive with HOOI and the trust-region algorithm for examples that are not too easy and not too difficult, such as tensors with elements taken from a normal distribution with zero mean and unit standard deviation.

In our study of algorithms for the low multilinear rank approximation of tensors, it was important to investigate a CG-based algorithm. The convergence speed of the algorithm is not favorable but this is compensated by the fact that the iterations are extremely fast.

4.4 Remarks

HOOI is a simple algorithm with cheap iterations but linear convergence rate. This suggests to use it if the precision or the computational time are not critical. On the other hand, the Newton based algorithm has local quadratic convergence rate but has expensive iterations and convergence issues. Thus, this algorithm can be used if a *good* starting point is available. The trust-region based algorithm has also fast (up to quadratic) convergence rate and cost per iteration smaller or equal to the one of the Newton based algorithm. Its computational time per iteration is competitive with the one of HOOI for approximations with small multilinear rank. Finally, the CG based algorithm converges after a large amount of cheap iterations. The cost for one iteration is similar to the cost of one HOOI iteration. Numerical experiments suggest that the CG algorithm has best performance for easy problems, i.e., for approximations where the original tensor is close to a tensor with low multilinear rank. We summarize the most important features of the algorithms in Table 1. Some numerical examples can be found in [42].

	HOOI	Newton	TR	CG
global/local convergence	(global)	local	global	(global)
convergence to	min, (saddle point), ((max))	stationary point	min, (saddle point), ((max))	min, (saddle point), ((max))
local convergence speed	linear	quadratic	superlinear up to quadratic	$\left(\begin{smallmatrix} n\text{-step} \\ \text{quadratic} \end{smallmatrix}\right)$
cost/iteration	$O(I^3R+IR^4+R^6)$	$O(I^3R^3)$	$\leq O(I^3R^3)$	$(\sim O(I^3R))$
monotonically decreasing?	yes	no	yes	yes

Table 1. Summary of the main features of HOOI, the Newton's algorithm, the trust-region algorithm and the conjugate gradient algorithm.

The low multilinear rank approximation problem (1) may have many local minima. Searching for distinct minima, all available algorithms could be run with a number of initial points. Because of the different functioning of the algorithms, they often find different solutions even if initialized in the same way.

5 Local minima

The best low multilinear rank approximation problem (1) has local minima [16, 23, 44, 42]. This is a key observation since the best low-rank approximation of a matrix has a unique minimum.

For tensors with low multilinear rank perturbed by a small amount of additive noise, algorithms converge to a small number of local minima. After increasing the noise level, the tensors become less structured and more local minima are found [44]. This behavior is related to the distribution of the mode-n singular values. In the first case, there is a large gap between the singular values. If the gap is small or nonexistent, the best low multilinear rank approximation is a difficult problem since we are looking for a structure that is not present. In this case, there are many equally good, or equally bad, solutions.

The values of the cost function at different local minima seem to be similar [44]. Thus, in applications where the multilinear rank approximation is merely used as a compression tool for memory savings, taking a nonglobal local minimum is not too different from working with the global minimum itself.

On the other hand, the column spaces of the matrices $\mathbf{U}_1$ and $\mathbf{U}_2$ corresponding to two different local minima are very different and the same holds for $\mathbf{V}$ and $\mathbf{W}$ [44]. In applications where these subspaces are important, local minima may be an issue. This concerns in particular the dimensionality reduction prior to computing a PARAFAC decomposition. One should inspect the gap between the mode-n singular values in each mode in order to choose meaningful values for the multilinear rank of the approximation.

An additional problem appears when the subspaces are important but the global minimum is not the desired one. This could happen when a tensor with low multilinear rank is affected by noise. The subspaces corresponding to the global minimum of (1) are not necessarily the closest to the subspaces corresponding to the original noise-free tensor, especially for high noise levels. This further stresses that solutions of the approximation problem have to be interpreted with care. It may even be impossible to obtain a meaningful solution.

It is usually a good idea to start from the truncated HOSVD. However, convergence to the global optimum is not guaranteed [16, 23, 44]. In some examples, a better (in the sense of yielding a smaller cost function value) local minimum is obtained from another initial point. Considering different algorithms with different initial values could improve the change to find the global minimum.

Finally, we describe a procedure for dimensionality reduction of large-scale problems. As an initial step, the HOSVD of the original tensor can be trun-

cated so that the mode-n singular values close to zero be discarded. In this way, the dimensions of the original tensor are reduced without losing much precision. As a second step prior to computing e.g., a PARAFAC decomposition, an essential dimensionality reduction via low multilinear rank approximation on an already smaller scale can be performed. The latter needs to take into account gaps between mode-n singular values.

6 Conclusions

This paper combines several topics. The main problem, the best low multilinear rank approximation of higher-order tensors, is a key problem in multilinear algebra having various applications. We considered solutions based on optimization on manifolds. The fact that the cost function is invariant under right multiplication of the matrices $\mathbf{U}, \mathbf{V}$ and $\mathbf{W}$ by orthogonal matrices prohibits potential algorithms from converging to a particular solution. Working on quotient manifolds isolates the solutions and makes the work of "standard" optimization algorithms easier.

The optimization methods on which the discussed methods are based are Newton's method, trust-region and conjugate gradients. There are also other methods in the literature. It is difficult to say which algorithm is the best. All algorithms have their advantages and disadvantages. Depending on the application, the dimensions of the tensor, the required precision and the time restrictions, one of the algorithms can be the method of choice. The Newton algorithm has local quadratic convergence rate but might diverge or converge to a saddle point or a maximum instead of a minimum. Moreover, it needs a good starting point. A well-chosen stopping criterion for the inner iteration of the trust-region algorithm leads to an algorithm with local quadratic convergence. The computational cost per iteration is competitive with the one of HOOI, which has only linear local convergence. Moreover, convergence of the trust-region algorithm to a minimum is (almost always) guaranteed. On the other hand, the conjugate gradient based algorithm has much cheaper iterations but lacks solid theoretical proofs.

It can make sense to apply several algorithms to the same problem. For example, if one wishes to inspect several local minima, one strategy would be to run all available algorithms, starting from enough initial points and in this way to obtain a more complete set of solutions. Due to the different character of the algorithms, they often find different solutions even when starting from the same initial values.

We also discussed the issue of local minima of the low multilinear rank approximation problem. It concerns the problem itself and does not depend on the actual algorithm. There are important consequences for whole classes of applications. One should be very careful when deciding whether or not it is meaningful to use such an approximation. The higher-order singular values may provide relevant information in this respect.

References

1. P.-A. Absil, C. G. Baker, K. A. Gallivan. Trust-region methods on Riemannian manifolds. *Found. Comput. Math.*, 7(3):303–330, 2007.
2. P.-A. Absil, M. Ishteva, L. De Lathauwer, S. Van Huffel. A geometric Newton method for Oja's vector field. *Neural Comput.*, 21(5):1415–1433, 2009.
3. P.-A. Absil, R. Mahony, R. Sepulchre. *Optimization Algorithms on Matrix Manifolds*. Princeton University Press, Princeton, NJ, 2008.
4. E. Acar, C. A. Bingol, H. Bingol, R. Bro, B. Yener. Multiway analysis of epilepsy tensors. *ISMB 2007 Conference Proc., Bioinformatics*, 23(13):i10–i18, 2007.
5. R. L. Adler, J.-P. Dedieu, J. Y. Margulies, M. Martens, M. Shub. Newton's method on Riemannian manifolds and a geometric model for the human spine. *IMA J. Numer. Anal.*, 22(3):359–390, 2002.
6. C. A. Andersson, R. Bro. The N-way toolbox for MATLAB. *Chemometrics and Intelligent Laboratory Systems*, 52(1):1–4, 2000. See also http://www.models.kvl.dk/source/nwaytoolbox/.
7. R. Badeau, R. Boyer. Fast multilinear singular value decomposition for structured tensors. *SIAM J. Matrix Anal. Appl.*, 30(3):1008–1021, 2008.
8. C. Caiafa, A. Cichocki. Reconstructing matrices and tensors from few rows and columns. In *Proc. of 2009 International Symposium on Nonlinear Theory and its Applications*, 2009. In press.
9. J. Carroll, J. Chang. Analysis of individual differences in multidimensional scaling via an N-way generalization of "Eckart-Young" decomposition. *Psychometrika*, 35(3):283–319, 1970.
10. J. Chen, Y. Saad. On the tensor SVD and the optimal low rank orthogonal approximation of tensors. *SIAM J. Matrix Anal. Appl.*, 30(4):1709–1734, 2009.
11. A. Cichocki, R. Zdunek, A. H. Phan, and S.-I. Amari. *Nonnegative Matrix and Tensor Factorizations*. Wiley, 2009.
12. P. Comon. Independent component analysis, a new concept? *Signal Processing*, 36(3):287–314, 1994.
13. P. Comon. Tensor decompositions. In J. G. McWhirter, I. K. Proudler (eds), *Mathematics in Signal Processing V*, pp. 1–24. Clarendon Press, Oxford, 2002.
14. P. Comon, G. Golub, L.-H. Lim, B. Mourrain. Symmetric tensors and symmetric tensor rank. *SIAM J. Matrix Anal. Appl.*, 30(3):1254–1279, 2008.
15. A. R. Conn, N. I. M. Gould, P. L. Toint. *Trust-Region Methods*. MPS-SIAM Series on Optimization. SIAM, Philadelphia, PA, 2000.
16. L. De Lathauwer. *Signal Processing Based on Multilinear Algebra*. PhD thesis, Dept. of Electrical Engineering, Katholieke Universiteit Leuven, 1997.
17. L. De Lathauwer. A link between the canonical decomposition in multilinear algebra and simultaneous matrix diagonalization. *SIAM J. Matrix Anal. Appl.*, 28(3):642–666, 2006.
18. L. De Lathauwer. Decompositions of a higher-order tensor in block terms — Part I: Lemmas for partitioned matrices. *SIAM J. Matrix Anal. Appl.*, 30(3): 1022–1032, 2008.
19. L. De Lathauwer. Decompositions of a higher-order tensor in block terms — Part II: Definitions and uniqueness. *SIAM J. Matrix Anal. Appl.*, 30(3):1033–1066, 2008.

20. L. De Lathauwer. A survey of tensor methods. In *Proc. of the 2009 IEEE International Symposium on Circuits and Systems (ISCAS 2009)*, pp. 2773–2776, Taipei, Taiwan, 2009.
21. L. De Lathauwer, J. Castaing. Tensor-based techniques for the blind separation of DS-CDMA signals. *Signal Processing*, 87(2):322–336, 2007.
22. L. De Lathauwer, B. De Moor, J. Vandewalle. A multilinear singular value decomposition. *SIAM J. Matrix Anal. Appl.*, 21(4):1253–1278, 2000.
23. L. De Lathauwer, B. De Moor, J. Vandewalle. 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(4):1324–1342, 2000.
24. L. De Lathauwer, B. De Moor, J. Vandewalle. Independent component analysis and (simultaneous) third-order tensor diagonalization. *IEEE Trans. Signal Process.*, 49(10):2262–2271, 2001.
25. L. De Lathauwer, B. De Moor, J. Vandewalle. Computation of the canonical decomposition by means of a simultaneous generalized Schur decomposition. *SIAM J. Matrix Anal. Appl.*, 26(2):295–327, 2004.
26. L. De Lathauwer, D. Nion. Decompositions of a higher-order tensor in block terms — Part III: Alternating least squares algorithms. *SIAM J. Matrix Anal. Appl.*, 30(3):1067–1083, 2008.
27. L. De Lathauwer, 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.*, 391:31–55, 2004.
28. V. de Silva, L.-H. Lim. Tensor rank and the ill-posedness of the best low-rank approximation problem. *SIAM J. Matrix Anal. Appl.*, 30(3):1084–1127, 2008.
29. M. De Vos, L. De Lathauwer, B. Vanrumste, S. Van Huffel, W. Van Paesschen. Canonical decomposition of ictal scalp EEG and accurate source localization: Principles and simulation study. *Journal of Computational Intelligence and Neuroscience*, 2007(Article ID 58253):1–10, 2007.
30. M. De Vos, A. Vergult, L. De Lathauwer, W. De Clercq, S. Van Huffel, P. Dupont, A. Palmini, W. Van Paesschen. Canonical decomposition of ictal scalp EEG reliably detects the seizure onset zone. *NeuroImage*, 37(3):844–854, 2007.
31. M. Elad, P. Milanfar, G. H. Golub. Shape from moments — an estimation theory perspective. *IEEE Trans. on Signal Processing*, 52(7):1814–1829, 2004.
32. L. Eldén, 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(2):248–271, 2009.
33. R. Fletcher, C. M. Reeves. Function minimization by conjugate gradients. *Comput. J.*, 7:149–154, 1964.
34. G. H. Golub, C. F. Van Loan. *Matrix Computations*. Johns Hopkins University Press, Baltimore, Maryland, 3rd edition, 1996.
35. M. Haardt, F. Roemer, G. Del Galdo. Higher-order SVD-based subspace estimation to improve the parameter estimation accuracy in multidimensional harmonic retrieval problems. *IEEE Trans. on Signal Processing*, 56(7):3198–3213, 2008.
36. W. W. Hager, H. Zhang. A new conjugate gradient method with guaranteed descent and an efficient line search. *SIAM Journal on Optimization*, 16(1):170–192, 2005.

37. R. A. Harshman. Foundations of the PARAFAC procedure: Model and conditions for an "explanatory" multi-mode factor analysis. *UCLA Working Papers in Phonetics*, 16(1):1–84, 1970.
38. U. Helmke, J. B. Moore. *Optimization and Dynamical Systems*. Springer-Verlag, 1993.
39. M. R. Hestenes, E. Stiefel. Methods of conjugate gradients for solving linear systems. *J. Research Nat. Bur. Standards*, 49:409–436 (1953), 1952.
40. F. L. Hitchcock. The expression of a tensor or a polyadic as a sum of products. *Journal of Mathematical Physics*, 6(1):164–189, 1927.
41. F. L. Hitchcock. Multiple invariants and generalized rank of a p-way matrix or tensor. *Journal of Mathematical Physics*, 7(1):39–79, 1927.
42. M. Ishteva. *Numerical methods for the best low multilinear rank approximation of higher-order tensors*. PhD thesis, Dept. of Electrical Engineering, Katholieke Universiteit Leuven, 2009.
43. M. Ishteva, P.-A. Absil, S. Van Huffel, L. De Lathauwer. Best low multilinear rank approximation with conjugate gradients. Tech. Rep. 09-246, ESAT-SISTA, K.U.Leuven, Belgium, 2009.
44. M. Ishteva, P.-A. Absil, S. Van Huffel, L. De Lathauwer. Tucker compression and local optima. Tech. Rep. UCL-INMA-2010.012, Université catholique de Louvain and 09-247, ESAT-SISTA, K.U.Leuven, Belgium, 2010.
45. M. Ishteva, L. De Lathauwer, P.-A. Absil, S. Van Huffel. Dimensionality reduction for higher-order tensors: algorithms and applications. *International Journal of Pure and Applied Mathematics*, 42(3):337–343, 2008.
46. M. Ishteva, L. De Lathauwer, P.-A. Absil, S. Van Huffel. Best low multilinear rank approximation of higher-order tensors, based on the Riemannian trust-region scheme. Tech. Rep. 09-142, ESAT-SISTA, K.U.Leuven, Belgium, 2009.
47. M. Ishteva, L. De Lathauwer, P.-A. Absil, S. Van Huffel. Differential-geometric Newton method for the best rank-(R_1, R_2, R_3) approximation of tensors. *Numerical Algorithms*, 51(2):179–194, 2009.
48. M. Ishteva, L. De Lathauwer, S. Van Huffel. Comparison of the performance of matrix and tensor based multi-channel harmonic analysis. In *7th International Conf. on Mathematics in Signal Processing, Cirencester, UK*, pp. 77–80, 2006.
49. E. Kofidis, P. A. Regalia. On the best rank-1 approximation of higher-order supersymmetric tensors. *SIAM J. Matrix Anal. Appl*, 23(3):863–884, 2002.
50. T. Kolda. Orthogonal tensor decompositions. *SIAM J. Matrix Anal. Appl.*, 23:243–255, 2001.
51. T. G. Kolda, B. W. Bader. Tensor decompositions and applications. *SIAM Review*, 51(3):455–500, 2009.
52. P. M. Kroonenberg. *Applied Multiway Data Analysis*. Wiley, 2008.
53. P. M. Kroonenberg, J. de Leeuw. Principal component analysis of three-mode data by means of alternating least squares algorithms. *Psychometrika*, 45(1):69–97, 1980.
54. M. W. Mahoney, M. Maggioni, P. Drineas. Tensor-CUR decompositions for tensor-based data. *SIAM J. Matrix Anal. Appl.*, 30(3):957–987, 2008.
55. J. H. Manton. Optimization algorithms exploiting unitary constraints. *IEEE Trans. Signal Process.*, 50(3):635–650, 2002.
56. C. D. Moravitz Martin, C. F. Van Loan. A Jacobi-type method for computing orthogonal tensor decompositions. *SIAM J. Matrix Anal. Appl.*, 30(3):1219–1232, 2008.

57. J. J. Moré, D. C. Sorensen. Computing a trust region step. *SIAM J. Sci. Statist. Comput.*, 4(3):553–572, 1983.
58. J. Nocedal, S. J. Wright. *Numerical Optimization*. Springer Verlag, New York, 2nd edition, 2006. Springer Series in Operations Research.
59. I. V. Oseledets, D. V. Savostianov, E. E. Tyrtyshnikov. Tucker dimensionality reduction of three-dimensional arrays in linear time. *SIAM J. Matrix Anal. Appl.*, 30(3):939–956, 2008.
60. J.-M. Papy, L. De Lathauwer, S. Van Huffel. Exponential data fitting using multilinear algebra: The single-channel and the multichannel case. *Numer. Linear Algebra Appl.*, 12(8):809–826, 2005.
61. J.-M. Papy, L. De Lathauwer, S. Van Huffel. Exponential data fitting using multilinear algebra: The decimative case. *J. Chemometrics*, 23(7–8):341–351, 2009.
62. E. Polak, G. Ribière. Note sur la convergence de méthodes de directions conjuguées. *Rev. Française Informat. Recherche Opérationnelle*, 3(16):35–43, 1969.
63. B. Savas, L. Eldén. Krylov subspace methods for tensor computations. Tech. Rep. LITH-MAT-R-2009-02-SE, Dept. of Mathematics, Linköping University, 2009.
64. B. Savas, L.-H. Lim. Best multilinear rank approximation of tensors with quasi-Newton methods on Grassmannians. Tech. Rep. LITH-MAT-R-2008-01-SE, Dept. of Mathematics, Linköping University, 2008.
65. M. Shub. Some remarks on dynamical systems and numerical analysis. In L. Lara-Carrero, J. Lewowicz (eds), *Proc. VII ELAM.*, pp. 69–92. Equinoccio, U. Simón Bolívar, Caracas, 1986.
66. N. Sidiropoulos, R. Bro, G. Giannakis. Parallel factor analysis in sensor array processing. *IEEE Trans. Signal Process.*, 48:2377–2388, 2000.
67. A. Smilde, R. Bro, P. Geladi. *Multi-way Analysis. Applications in the Chemical Sciences*. John Wiley and Sons, Chichester, U.K., 2004.
68. S. T. Smith. *Geometric Optimization Methods for Adaptive Filtering*. PhD thesis, Division of Applied Sciences, Harvard University, Cambridge, MA, 1993.
69. S. T. Smith. Optimization techniques on Riemannian manifolds. In A. Bloch (ed), *Hamiltonian and gradient flows, algorithms and control*, volume 3 of *Fields Inst. Commun.*, pp. 113–136. Amer. Math. Soc., Providence, RI, 1994.
70. T. Steihaug. The conjugate gradient method and trust regions in large scale optimization. *SIAM J. Numer. Anal.*, 20(3):626–637, 1983.
71. P. L. Toint. Towards an efficient sparsity exploiting Newton method for minimization. In I. S. Duff (ed), *Sparse Matrices and Their Uses*, pp. 57–88. Academic Press, London, 1981.
72. L. R. Tucker. The extension of factor analysis to three-dimensional matrices. In H. Gulliksen, N. Frederiksen (eds), *Contributions to mathematical psychology*, pp. 109–127. Holt, Rinehart & Winston, NY, 1964.
73. L. R. Tucker. Some mathematical notes on three-mode factor analysis. *Psychometrika*, 31:279–311, 1966.
74. L. Vanhamme, S. Van Huffel. Multichannel quantification of biomedical magnetic resonance spectroscopy signals. In F. Luk (ed), *Proc. of SPIE, Advanced Signal Processing Algorithms, Architectures, and Implementations VIII*, volume 3461, pp. 237–248, San Diego, California, 1998.
75. T. Zhang, G. H. Golub. Rank-one approximation to high order tensors. *SIAM J. Matrix Anal. Appl.*, 23:534–550, 2001.