DIMENSIONALITY REDUCTION FOR HIGHER-ORDER TENSORS: ALGORITHMS AND APPLICATIONS

Mariya Ishteva[1], Lieven De Lathauwer[1,2], P.-A. Absil[3], and
Sabine Van Huffel[1]
[1]ESAT/SCD, Katholieke Universiteit Leuven,
Kasteelpark Arenberg 10, bus 2446, B-3001 Leuven, BELGIUM.
e-mail:
{mariya.ishteva,lieven.delathauwer,sabine.vanhuffel}@esat.kuleuven.be
[2]Subfaculty Sciences, Katholieke Universiteit Leuven Campus Kortrijk,
Kortrijk, Belgium.
[3]INMA, Université catholique de Louvain,
Av. Georges Lemaître 4, B-1348 Louvain-la-Neuve, BELGIUM.
web: http://www.inma.ucl.ac.be/∼absil/

Abstract: Higher-order tensors have applications in many areas such as biomedical engineering, image processing, and signal processing. For example, dimensionality reduction of a multi-way problem can be achieved by the best rank-(R_1,R_2,. . . ,R_N) approximation of tensors. Contrary to the matrix case, the tensor best rank-$(R_1, R_2, \ldots, R_N)$ approximation cannot be computed in a straightforward way. In this paper, we present the higher-order orthogonal iterations and outline two new algorithms, based on the trust-region and conjugate gradient methods on manifolds. We touch on some of the applications.

AMS Subject Classification: 15A69, 15A18
Key Words: multilinear algebra, higher-order tensor, rank reduction, trust-region, conjugate gradients, Grassmann manifold

1 Motivation

Independent component analysis applications, such as electro-encephalography, magneto-encephalography, nuclear magnetic resonance, often involve high-dimensional data in which only a few sources have significant contributions. To reduce the dimensionality of the problem from the number of observation channels to the number of sources the best rank-$(R_1, R_2, \ldots, R_N)$ approximation of tensors can be used [6].

Parallel factor decomposition (PARAFAC) [9] is a decomposition of higher-order tensors in rank-1 terms. This decomposition is widely used in chemometrics. It can also be used for epileptic seizure onset localisation [7], since only one of its components is related to the seizure activity. However, computing PARAFAC is a difficult problem, especially if the dimensions of the tensor are large. Dimensionality reduction is useful in this case as well.

Another field in which the best rank-$(R_1, R_2, \ldots, R_N)$ approximation of tensors can be applied is image synthesis, analysis and recognition. A set of facial images can be represented as a higher-order tensor, where different modes correspond to different factors, such as face expression, position of the head relative to the camera, and illumination [10].

Finally, we mention that in many signal processing applications a signal is decomposed as a sum of exponentially damped sinusoids. Tensor-based algorithms for the estimation of the poles and the complex amplitudes, given only samples of the signal are proposed in [11]. They are based on the best rank-$(R_1, R_2, \ldots, R_N)$ approximation of a tensor.

This paper is organized as follows: Section 2 introduces the problem of the best rank-(R_1, R_2, R_3) approximation. For simplicity we only consider real-valued third-order tensors. The generalization to higher-order tensors is straightforward. One algorithm for solving the problem is the higher-order orthogonal iteration (HOOI) [5] which is an alternating least-squares algorithm. It is summarized in Section 3. Two new algorithms, based on the trust-region and on the conjugate gradients methods on manifolds are outlined in Section 4 and Section 5 respectively. Another manifold-based algorithm was recently proposed in [8].

2 Problem Formulation

Nth-order tensors are generalizations of vectors (order 1) and matrices (order 2). Their elements are referred to by N indices. The columns of a tensor are called mode-1 vectors, the rows are called mode-2 vectors. In general, the mode-n vectors ($n = 1, 2, \ldots, N$) are the vectors, obtained by varying the n-th index, while keeping the other indices fixed. The maximal number of linearly independent mode-n vectors is called the mode-n rank. It is a generalization of the column and row rank of a matrix but different mode-n ranks are not necessarily equal to each other.

Our goal is the best rank-(R_1, R_2, R_3) approximation $\hat{\mathcal{A}} \in \mathbb{R}^{I_1 \times I_2 \times I_3}$ of a third-order tensor $\mathcal{A} \in \mathbb{R}^{I_1 \times I_2 \times I_3}$ as a tool for dimensionality reduction. $\hat{\mathcal{A}}$ needs to minimize the least-squares cost function $F : \mathbb{R}^{I_1 \times I_2 \times I_3} \to \mathbb{R}$,

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

under the constraints $\text{rank}_1(\hat{\mathcal{A}}) \leq R_1$, $\text{rank}_2(\hat{\mathcal{A}}) \leq R_2, \text{rank}_3(\hat{\mathcal{A}}) \leq R_3$, where $\text{rank}_n(.)$ stands for the mode-n rank and $\|.\|$ is the Frobenius norm. The truncated SVD gives the best low-rank approximation of a matrix. However, truncation of the higher-order equivalent of the SVD (HOSVD) [4] usually results in a suboptimal tensor rank-$(R_1, R_2, \ldots, R_N)$ approximation, which can be refined by iterative algorithms.

In this paper, we consider maximizing $\bar{g}$,

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

over the orthonormal matrices $\mathbf{U}, \mathbf{V}, \mathbf{W}$. Here, $\mathcal{A} \bullet_n \mathbf{M}$, $n = 1, 2, 3$ is the product of a tensor $\mathcal{A}$ and a matrix $\mathbf{M}$ with respect to the n-th mode of the tensor [4]. This problem is equivalent to minimizing (1) [5]. It is also possible to perform computations on a matrix level due to the following property of $\bar{g}$

$$\bar{g}(\mathbf{U}, \mathbf{V}, \mathbf{W}) = \|\mathcal{A} \bullet_1 \mathbf{U}^T \bullet_2 \mathbf{V}^T \bullet_3 \mathbf{W}^T\|^2 = \|\mathbf{U}^T (\mathbf{A}_{(1)}(\mathbf{V} \otimes \mathbf{W}))\|^2, \tag{3}$$

where $\mathbf{A}_{(1)}$ is a matrix representation of the tensor $\mathcal{A}$, obtained by putting the columns of $\mathcal{A}$ one after the other in a specific order [4]. The symbol "$\otimes$" stands for the Kronecker product. Having estimated $\mathbf{U}, \mathbf{V}, \mathbf{W}$, the optimal tensor $\hat{\mathcal{A}}$ is computed [5] by

$$\hat{\mathcal{A}} = \mathcal{B} \bullet_1 \mathbf{U} \bullet_2 \mathbf{V} \bullet_3 \mathbf{W}, \tag{4}$$

where $\mathcal{B} = \mathcal{A} \bullet_1 \mathbf{U}^T \bullet_2 \mathbf{V}^T \bullet_3 \mathbf{W}^T \in \mathbb{R}^{R_1 \times R_2 \times R_3}$.

3 Higher-Order Orthogonal Iteration

HOOI [5] is an alternating least-squares algorithm for optimizing (2). At each step the estimate of one of the matrices $\mathbf{U}, \mathbf{V}, \mathbf{W}$ is optimized, while the other two stay fixed. In order to maximize with respect to the unknown orthonormal matrix $\mathbf{U}$, $\bar{g}(\mathbf{U}, \mathbf{V}, \mathbf{W})$ is thought of as a quadratic expression in the components of $\mathbf{U}$. From (3) it can be deduced that the columns of $\mathbf{U} \in \mathbb{R}^{I_1 \times R_1}$ should build an orthonormal basis for the left R_1-dimensional dominant singular subspace of $\mathbf{A}_{(1)}(\mathbf{V} \otimes \mathbf{W})$, which can be obtained from the SVD of $\mathbf{A}_{(1)}(\mathbf{V} \otimes \mathbf{W})$. The optimization with respect to $\mathbf{V}$ and $\mathbf{W}$ is performed by analogy.

The **HOOI Algorithm** can be summarized as follows

1. Obtain initial estimates $\mathbf{U}_0, \mathbf{V}_0, \mathbf{W}_0$, e.g., from HOSVD
2. Iterate until convergence ($k = 0, 1, 2, \ldots$)
   - Compute the columns of $\mathbf{U}_{k+1}$ as orthonormal basis vectors of the R_1-dimensional left dominant singular subspace of $\mathbf{A}_{(1)}(\mathbf{V}_k \otimes \mathbf{W}_k)$.
   - Compute the columns of $\mathbf{V}_{k+1}$ as orthonormal basis vectors of the R_2-dimensional left dominant singular subspace of $\mathbf{A}_{(2)}(\mathbf{W}_k \otimes \mathbf{U}_{k+1})$.
   - Compute the columns of $\mathbf{W}_{k+1}$ as orthonormal basis vectors of the R_3-dimensional left dominant singular subspace of $\mathbf{A}_{(3)}(\mathbf{U}_{k+1} \otimes \mathbf{V}_{k+1})$.
3. Compute $\hat{\mathcal{A}}$ using (4) with the converged $\mathbf{U}, \mathbf{V}$, and $\mathbf{W}$ from Step 2.

4 Trust Region Based Algorithm

The algorithm that we present in this section is based on the trust-region (TR) method on manifolds [1]. Our motivation for using manifolds is the invariance property of the function $\bar{g}$ from (2),

$$\bar{g}(\mathbf{U},\, \mathbf{V},\, \mathbf{W}) = \bar{g}(\mathbf{U}\mathbf{Q}^{(1)},\, \mathbf{V}\mathbf{Q}^{(2)},\, \mathbf{W}\mathbf{Q}^{(3)})\,, \tag{5}$$

where $\mathbf{Q}^{(i)}$, $i = 1, 2, 3$ are orthogonal matrices. (5) holds for any $\mathbf{U}, \mathbf{V}, \mathbf{W}$ (not necessarily orthonormal), so we can define a function $g : M \to \mathbb{R}$,

$$g([\mathbf{U}], [\mathbf{V}], [\mathbf{W}]) = \bar{g}(\mathbf{U},\, \mathbf{V},\, \mathbf{W}), \tag{6}$$

on the product manifold $M = Gr(R_1, I_1) \times Gr(R_2, I_2) \times Gr(R_3, I_3)$, where $Gr(R, I)$ denotes the Grassmann manifold of R-dimensional subspaces of $\mathbb{R}^I$ and $[\mathbf{A}]$ stands for all matrices whose columns span the same subspace as the ones of $\mathbf{A}$. The elements of M are more complex but simplifications in the formulas and better convergence results follow, which justify (6).

The trust-region method [3] is an iterative algorithm for minimizing a cost function f. At each step a quadratic model m of f is minimized in a trust-region around the current iterate and thus an update η is computed. The quality of the model is evaluated and as a consequence, the new iterate is accepted or rejected and the trust-region radius is updated. Below, we summarize the TR-based algorithm for the cost function g from (6).

The **TR-based algorithm**

1. Obtain initial iterate $\mathbf{X}_0 = \{\mathbf{U}_0, \mathbf{V}_0, \mathbf{W}_0\} \in M$.
2. Iterate until convergence ($k = 0, 1, 2, \ldots$)
   - Compute η_k as the solution of the trust-region subproblem $\min\limits_{\eta \in T_{\mathbf{X}_k} M} m_{\mathbf{X}_k}(\eta)$, subject to $\langle \eta, \eta \rangle \leq \Delta_k^2$, where

$$m_{\mathbf{X}_k}(\eta) = g(\mathbf{X}_k) + \langle \operatorname{grad} g(\mathbf{X}_k), \eta \rangle + \frac{1}{2}\langle \operatorname{Hess} g(\mathbf{X}_k)[\eta], \eta \rangle.$$

- Compute the quotient $\rho_k = \dfrac{g(\mathbf{X}_k) - g(R_{\mathbf{X}_k}(\eta_k))}{m_{\mathbf{X}_k}(0_{\mathbf{X}_k}) - m_{\mathbf{X}_k}(\eta_k)}$.
- Set $\mathbf{X}_{k+1}$ to be $R_{\mathbf{X}_k}(\eta_k)$ if ρ_k is big enough and $\mathbf{X}_k$ otherwise.
- Set Δ_{k+1} to be $\frac{1}{4}\Delta_k$ if ρ_k is too small, $2\Delta_k$ if ρ_k is large, else Δ_k.

3. Compute $\hat{\mathcal{A}}$ using (4) with the converged $\mathbf{X}=\{\mathbf{U},\mathbf{V},\mathbf{W}\}$ from Step 2.

The function R is called *retraction.* It specifies how a new iterate in the direction of the tangent vector $\eta = (\mathbf{Z_U}, \mathbf{Z_V}, \mathbf{Z_W})$ at a point $\mathbf{X} = (\mathbf{U}, \mathbf{V}, \mathbf{W})$ on the manifold is computed (for details see [1]). For the manifold M the following retraction can be used:

$$R_{(\mathbf{U},\mathbf{V},\mathbf{W})}(\mathbf{Z_U}, \mathbf{Z_V}, \mathbf{Z_W}) = (\mathrm{qf}(\mathbf{U} + \mathbf{Z_U}), \mathrm{qf}(\mathbf{V} + \mathbf{Z_V}), \mathrm{qf}(\mathbf{W} + \mathbf{Z_W})),$$

where qf denotes the Q factor of the thin QR decomposition. Finally, we mention that closed-form expressions for the gradient $\mathrm{grad}\, g$, and the Hessian $\mathrm{Hess}\, g$ can be obtained.

5 Conjugate Gradients Based Algorithm

In this section, we propose a conjugate gradient (CG) based algorithm for solving (1). Again we make use of the invariance property (5) and manifold idea (6). The CG method is an iterative method for minimizing a cost function. At each step a search direction is computed, which takes into account the previous ones. More details about nonlinear CG on manifolds can be found in [2].

The **CG-based algorithm** can be summarized as follows

1. Obtain initial iterate $\mathbf{X}_0 = \{\mathbf{U}_0, \mathbf{V}_0, \mathbf{W}_0\} \in M$.
2. Set $\eta_0 = -\mathrm{grad}\, g(\mathbf{X}_0)$.
3. Iterate until convergence ($k = 0, 1, 2, \ldots$)
   - Compute α_k; e.g., use Armijo stepsize.
   - Set $\mathbf{X}_{k+1} = R_{\mathbf{X}_k}(\alpha_k \eta_k)$.
   - Compute β_{k+1}, e.g., via the Fletcher-Reeves formula
   $$\beta_{k+1} = \frac{\langle \mathrm{grad}\, g(\mathbf{X}_{k+1}), \mathrm{grad}\, g(\mathbf{X}_{k+1})\rangle}{\langle \mathrm{grad}\, g(\mathbf{X}_k), \mathrm{grad}\, g(\mathbf{X}_k)\rangle} .$$
   - Set $\eta_{k+1} = -\mathrm{grad}\, g(\mathbf{X}_{k+1}) + \beta_{k+1}\mathcal{T}_{\alpha_k\eta_k}(\eta_k)$.
4. Compute $\hat{\mathcal{A}}$ using (4) with the converged $\mathbf{X}=\{\mathbf{U},\mathbf{V},\mathbf{W}\}$ from Step 3.

To perform a CG algorithm on a manifold we need to work simultaneously with tangent vectors at two different points on the manifold.

For this purpose we define a function, called *vector transport* [2]. For our problem, it can be the following function

$$\mathcal{T}_{\eta_{\mathbf{X}}}\xi_{\mathbf{X}} = P^h_{\mathbf{X}+\bar{\eta}_{\mathbf{X}}}\bar{\xi}_{\mathbf{X}}\,,$$

where $\eta_{\mathbf{X}}$ and $\xi_{\mathbf{X}}$ are two tangent vectors at point $[\mathbf{X}]$, $\bar{\xi}$ is a matrix representation of ξ, called horizontal lift, and $P^h_{\mathbf{Y}}$ is the projection onto the orthogonal complement of the column space of $\mathbf{Y}$.

Acknowledgements

The research reported in this paper is supported by: (i) FWO-G.0321.06 (Tensors/Spectral Analysis), (ii) IUAP P6/04 (DYSCO), 2007-2011, (iii) Impulsfinanciering Campus Kortrijk (2007-2012)(CIF1), (iv) Research Council KUL: OE/06/25, OE/07/17, GOA-AMBioRICS, CoE EF/05/006.

References

[1] P.-A. Absil, C.G. Baker, K.A. Gallivan, Trust-region methods on Riemannian Manifolds, *Found. Comput. Math.*, **7** (2007), 303-330.

[2] P.-A. Absil, R. Mahony, R. Sepulchre, *Optimization Algorithms on Matrix Manifolds*, Princeton University Press, Princeton, NJ (2008).

[3] A.R. Conn, N.I.M. Gould, P.L. Toint, *Trust-Region Methods*, Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA (2000).

[4] L. De Lathauwer, B. De Moor, J. Vandewalle, A Multilinear Singular Value Decomposition, *SIAM J. Matrix Anal. Appl.*, **21** (2000), 1253-1278.

[5] 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** (2000), 1324-1342.

[6] 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 and its Applications*, **391** (2004), 31-55.

[7] M. De Vos, A. Vergult, et al., Canonical Decomposition of Ictal Scalp EEG Reliably Detects the Seizure Onset Zone, *NeuroImage*, accepted.

[8] 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, Techreport LiTH-MAT-R-2007-6-SE, Linköping university (2007).

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

[10] M.A.O. Vasilescu, D. Terzopoulos, Multilinear Subspace Analysis for Image Ensembles, *Proc. Computer Vision and Pattern Recognition Conf. (CVPR '03), Madison, WI*, **2** (2003), 93-99.

[11] J.M. Papy, L. De Lathauwer, S. Van Huffel, Exponential data fitting using multilinear algebra: The single-channel and the multichannel case, *Numerical Linear Algebra and Applications*, **12** (2005), 809-826.