Algebraic and Optimization-based Algorithms for Decomposing Tensors into Block Terms

Nico Vervliet, Ignat Domanov, Lieven De Lathauwer

Department of Electrical Engineering (ESAT)—Stadius, KU Leuven
https://homes.esat.kuleuven.be/~nvervlie

Abstract

Tensors, or multiway arrays of numerical values, and their decompositions have been applied successfully in a myriad of applications in, a.o., signal processing, data analysis and machine learning [1]. Key is the ability to use decompositions to extract simple components under mild uniqueness conditions. The simplest decomposition, the canonical polyadic decomposition (CPD), writes a tensor as a sum of rank-1 tensors. However, in some applications more complex terms are required and a block term decomposition (BTD) may be more suitable.

In this work, we consider the decomposition of a third-order $I \times J \times K$ tensor $\mathcal{T}$ into a sum of multilinear rank $(M_r, N_r, \cdot)$ terms:

$$\mathcal{T} = \sum_{r=1}^{R} \mathcal{D}_r \cdot_1 \mathbf{A}_r \cdot_2 \mathbf{B}_r \tag{1}$$

in which $\mathcal{D}_r$, $\mathbf{A}_r$ and $\mathbf{B}_r$ have dimensions $M_r \times N_r \times K$, $I \times M_r$ and $J \times N_r$, resp., $\cdot_n$ denotes the mode-n product, i.e., $\mathcal{T} = \mathcal{D} \cdot_n \mathbf{A} \Leftrightarrow \mathbf{T}_{(n)} = \mathbf{A}\mathbf{D}_{(n)}$, and the subscript $_{(n)}$ denotes the mode-n unfolding; see [1]. Alternatively, one can see this as the joint block diagonalization of the frontal slices $\mathbf{T}_k$, $k = 1, \ldots, K$:

$$\mathbf{T}_k = \mathbf{A} \cdot \operatorname{blockdiag}(\mathbf{D}_{1,k}, \ldots, \mathbf{D}_{R,k}) \cdot \mathbf{B}^{\mathrm{T}}, \tag{2}$$

in which $\mathbf{A} = [\mathbf{A}_1, \ldots, \mathbf{A}_R]$ and $\mathbf{B} = [\mathbf{B}_1, \ldots, \mathbf{B}_R]$. This problem has been studied for some special cases; see, e.g., [2, 3]. We tackle the general BTD, which is often solved via a nonconvex optimization problem which has many local optima. For the CPD and the decomposition into multilinear rank $(L_r, L_r, 1)$ terms (LL1), algebraic methods often provide suitable initializations. Here, we aim to derive an algebraic and an optimization-based method for problem (1).

Algebraic algorithm We show that the decomposition (1) can be cast as a coupled simultaneous eigenvalue decomposition (CS-EVD) problem under mild assumptions. Let $\mathcal{T} \in \mathbb{K}^{I \times J \times K}$ ($\mathbb{K}$ can be $\mathbb{R}$ or $\mathbb{C}$) admit a decomposition into a sum of R multilinear rank $(M_r, N_r, \cdot)$ terms (Eq. (1)) and assume $\sum_{r=1}^{R} M_r \leq I$ and $\sum_{r=1}^{R} N_r \leq J$. (Using compression by multilinear singular value decomposition [4], we can assume w.l.o.g. that these are equalities.) Let $\mathbf{A}$ and $\mathbf{B}$ have full column rank. Then, under some mild conditions, the decomposition in Eq. (1) is unique (see [5]), and can be reduced to a CS-EVD as follows. Let $\mathbf{M} \in \mathbb{K}^{IJK \times (I^2+J^2)}$ be defined as

$$\mathbf{M} = \begin{bmatrix} \mathbf{T}_1^{\mathrm{T}} \otimes \mathbf{I}_I & -\mathbf{I}_J \otimes \mathbf{T}_1 \\ \vdots & \vdots \\ \mathbf{T}_K^{\mathrm{T}} \otimes \mathbf{I}_I & -\mathbf{I}_J \otimes \mathbf{T}_K \end{bmatrix}. \tag{3}$$

We can show that $\operatorname{null}(\mathbf{M}) =: \mathbf{N}$ has dimension R and can be partitioned as $\begin{bmatrix} \mathbf{X} \\ \mathbf{Y} \end{bmatrix}$ in which $\mathbf{X} \in \mathbb{K}^{I^2 \times R}$ and $\mathbf{Y} \in \mathbb{K}^{J^2 \times R}$. Let $\mathbf{X}_r \in \mathbb{K}^{I \times I}$ ($\mathbf{Y}_r \in \mathbb{K}^{J \times J}$) be the reshaped rth column $\mathbf{x}_r$ ($\mathbf{y}_r$),

$r = 1, \ldots, R$, of $\mathbf{X}$ ($\mathbf{Y}$). We can show that $\mathbf{A}$ and $\mathbf{B}$ can be recovered from the CS-EVD problem

$$\mathbf{X}_r = \mathbf{A} \cdot \text{blockdiag}(g_{1r}\mathbf{I}_{M_1}, \ldots, g_{Rr}\mathbf{I}_{M_R}) \cdot \mathbf{A}^{-1}, \qquad r = 1, \ldots, R, \tag{4}$$

$$\mathbf{Y}_r = \mathbf{B}^{-\mathrm{T}} \cdot \text{blockdiag}(g_{1r}\mathbf{I}_{N_1}, \ldots, g_{Rr}\mathbf{I}_{N_R}) \cdot \mathbf{B}^{\mathrm{T}}, \qquad r = 1, \ldots, R. \tag{5}$$

Hence, we have two sets of simultaneous EVD problems that are coupled through the eigenvalues g_{rs}, $r = 1, \ldots, R$, $s = 1, \ldots, R$, which result in the factors $\mathbf{A}$ and $\mathbf{B}$ in Eq. (2). The core tensors $\mathcal{D}_r$ can be recovered from Eq. (2) as $\mathbf{A}$ and $\mathbf{B}$ are invertible by assumption. Note that R, M_r and N_r can be determined automatically from the dimension of the null space and the eigenvalues g_{rs}.

To reduce the computational cost of computing the null space $\mathbf{N}$ of the $IJK \times (I^2+J^2)$ matrix $\mathbf{M}$, an iterative EVD algorithm is used to compute $\mathbf{N}$ as the subspace corresponding to the zero eigenvalues of $\mathbf{M}^{\mathrm{H}}\mathbf{M}$. The bottleneck computation is solving $\mathbf{M}^{\mathrm{H}}\mathbf{M}\mathbf{x} = \mathbf{y}$ for $\mathbf{x}$ every iteration. We show that the QR factorization of $\mathbf{M} = \mathbf{QR}$ can be computed efficiently by exploiting the Kronecker and block structures using ideas from the block-QR and quasi-QR factorizations [6]. This way, $\mathbf{M}^{\mathrm{H}}\mathbf{M}\mathbf{x} = \mathbf{y}$ can be written as $\mathbf{R}^{\mathrm{H}}\mathbf{R}\mathbf{x} = \mathbf{y}$ which can be solved using forward and backward substitution as $\mathbf{R}$ is upper triangular. The cost of computing $\mathbf{N}$ then is $\mathcal{O}(J^4 + I^2J^2 + I^3)$ flop per iteration for the EVD step and $\mathcal{O}(J^6 + IJ^4K + I^2J^3K)$ flop for the QR step.

Optimization-based algorithm The CS-EVD problem (Eqs. (4)–(5)) is a coupled LL1 decomposition: by stacking $\mathbf{X}_r$ and $\mathbf{Y}_r$ as frontal slices in $\mathcal{X}$ and $\mathcal{Y}$, resp., we have (using CPD notation):

$$\mathcal{X} = [\![\mathbf{A}, \mathbf{A}^{-\mathrm{T}}, \mathbf{GP}_1]\!], \qquad \mathcal{Y} = [\![\mathbf{B}^{-\mathrm{T}}, \mathbf{B}, \mathbf{GP}_2]\!], \tag{6}$$

in which $\mathbf{G}$ collects the eigenvalues in Eqs. (4)–(5) and $\mathbf{P}_1 = \text{blockdiag}(\mathbf{1}_{M_1}^{\mathrm{T}}, \ldots, \mathbf{1}_{M_R}^{\mathrm{T}})$ and $\mathbf{P}_2 = \text{blockdiag}(\mathbf{1}_{N_1}^{\mathrm{T}}, \ldots, \mathbf{1}_{N_R}^{\mathrm{T}})$. We propose a single-step optimization-based approach that exploits this and computes the null space and the decomposition simultaneously by solving

$$\min_{\mathbf{A},\mathbf{B},\mathbf{G}} \frac{1}{2} \sum_{k=1}^{K} \left\| [\![\mathbf{A}, \mathbf{T}_k^{\mathrm{T}}\mathbf{A}^{-\mathrm{T}}, \mathbf{GP}_1]\!] - [\![\mathbf{T}_k\mathbf{B}^{-\mathrm{T}}, \mathbf{B}, \mathbf{GP}_2]\!] \right\|_{\mathrm{F}}^2. \tag{7}$$

This single-step approach has several advantages: increased robustness to noise, reduced accumulation of numerical error, and the possibility to exploit structure in both $\mathbf{M}$ and $\mathbf{N}$ simultaneously. The problem can be solved efficiently using Gauss–Newton in combination with standard CPD techniques and parametric constraints; see [7]. To avoid zero solutions, we can set $\mathbf{G} = \mathbf{I}$.

[1] N. D. Sidiropoulos, L. De Lathauwer, X. Fu, K. Huang, E. E. Papalexakis, and C. Faloutsos. Tensor decomposition for signal processing and machine learning. *IEEE Trans. Signal Process.*, 65 (13): pp. 3551–3582, 2017

[2] L. De Lathauwer. Decompositions of a higher-order tensor in block terms—Part II: Definitions and uniqueness. *SIAM J. Matrix Anal. Appl.*, 30 (3): pp. 1033–1066, 2008

[3] G. Chabriel, M. Kleinsteuber, E. Moreau, H. Shen, P. Tichavský, and A. Yeredor. Joint matrices decompositions and blind source separation. A survey of methods, identification and applications. *IEEE Signal Process. Mag.*, 31: pp. 34-43, 2014

[4] L. De Lathauwer, B. De Moor, and J. Vandewalle. A multilinear singular value decomposition. *SIAM J. Matrix Anal. Appl.*, 21 (4): pp. 1253-1278, 2000

[5] I. Domanov, N. Vervliet, and L. De Lathauwer. Decomposition of a tensor into multilinear rank $(M_r, N_r, \cdot)$ terms. *Internal Report 18-51*, ESAT-STADIUS, KU Leuven (Leuven, Belgium), 2018

[6] G. Stewart. Error analysis of the quasi-Gram–Schmidt algorithm. *SIAM J. Matrix Analysis Applications*, 27: pp. 493-506, 2005

[7] N. Vervliet, and L. De Lathauwer. Numerical optimization-based algorithms for data fusion. Chapter 4 of *Data Fusion Methodology and Applications*, Cocchi M., ed., vol 33 of *Data Handling in Science and Technology*, Elsevier, pp. 81-128