Rank Splitting for CANDECOMP/PARAFAC

Anh-Huy Phan[1](✉), Petr Tichavský[2], and Andrzej Cichocki[1]

[1] Lab for Advanced Brain Signal Processing, Brain Science Institute - RIKEN, Saitama, Japan
phan@brain.riken.jp
[2] Institute of Information Theory and Automation, Prague, Czech Republic
tichavsk@utia.cas.cz

Abstract. CANDECOMP/PARAFAC (CP) approximates multiway data by a sum of rank-1 tensors. Our recent study has presented a method to rank-1 tensor deflation, i.e. sequential extraction of rank-1 tensor components. In this paper, we extend the method to block deflation problem. When at least two factor matrices have full column rank, one can extract two rank-1 tensors simultaneously, and rank of the data tensor is reduced by 2. For decomposition of order-3 tensors of size $R \times R \times R$ and rank-R, the block deflation has a complexity of $\mathcal{O}(R^3)$ per iteration which is lower than the cost $\mathcal{O}(R^4)$ of the ALS algorithm for the overall CP decomposition.

Keywords: Canonical polyadic decomposition · PARAFAC · Tensor deflation

1 Introduction

An important property in matrix factorisations like eigenvalue decomposition, is that rank-1 matrix components can be sequentially estimated via a deflation method, such as the power iteration method. The matrix deflation procedure is possible because subtracting the best rank-1 term from a matrix reduces the matrix rank. Unfortunately, this sequential extraction procedure in general is not applicable to decompose a rank-R tensor [1].

In our recent study [2,3], we have introduced a tensor decomposition which is able to extract a rank-1 tensor from a high rank tensor. The method is based on the rank-1 plus multilinear-$(R-1, R-1, R-1)$ block tensor decomposition, but with a smaller number of parameters, basically two vectors per modes. This paper extends the rank-1 tensor extraction to block tensor deflation or rank splitting which splits a high rank-R tensor into two tensors of smaller ranks. In particular, we develop an alternating subspace update (ASU) algorithm to

P. Tichavský—The work of P. Tichavský was supported by The Czech Science Foundation through Project No. 14-13713S.

A. Cichocki—Also affiliated with the EE Dept., Warsaw University of Technology and with Systems Research Institute, Polish Academy of Science, Poland.

© Springer International Publishing Switzerland 2015
E. Vincent et al. (Eds.): LVA/ICA 2015, LNCS 9237, pp. 31–40, 2015.
DOI: 10.1007/978-3-319-22482-4_4

extract a multilinear rank-(2,2,2) tensor from a rank-R tensor. Since decomposition of a $2 \times 2 \times 2$ tensor can be found in closed-form, we can straightforwardly obtain the desired rank-1 components. The proposed algorithm estimates only 4 vectors and two scalars per dimension with a computational complexity of $\mathcal{O}(R^3)$. Moreover, it also requires a lower space cost than algorithms for the ordinary CANDECOMP/PARAFAC (CPD).

The paper is organised as follows. A tensor decomposition for the block tensor deflation or rank splitting is presented in Sect. 2. The proposed algorithm is presented in Sect. 3. Simulations in Sect. 4 will verify validity and performance of the proposed algorithm. Section 5 concludes the paper.

2 Preliminaries

Throughout the paper, we shall denote tensors by bold calligraphic letters, e.g., $\boldsymbol{\mathcal{A}} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$, matrices by bold capital letters, e.g., $\mathbf{A} = [\boldsymbol{a}_1, \boldsymbol{a}_2, \ldots, \boldsymbol{a}_R] \in \mathbb{R}^{I \times R}$, and vectors by bold italic letters, e.g., $\boldsymbol{a}_j$. The Kronecker product is denoted by $\otimes$. Inner product of two tensors is denoted by $\langle \boldsymbol{\mathcal{X}}, \boldsymbol{\mathcal{Y}} \rangle = \operatorname{vec}(\boldsymbol{\mathcal{X}})^T \operatorname{vec}(\boldsymbol{\mathcal{Y}})$. Contraction between two tensors along modes-$\boldsymbol{m}$, where $\boldsymbol{m} = [m_1, \ldots, m_K]$, is denoted by $\langle \boldsymbol{\mathcal{X}}, \boldsymbol{\mathcal{Y}} \rangle_{\boldsymbol{m}}$, whereas $\langle \boldsymbol{\mathcal{X}}, \boldsymbol{\mathcal{Y}} \rangle_{-n}$ represents contraction along all modes but mode-n [4].

The mode-n matricization of tensor $\boldsymbol{\mathcal{Y}}$ is denoted by $\mathbf{Y}_{(n)}$. The mode-n multiplication of a tensor $\boldsymbol{\mathcal{Y}} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$ by a matrix $\mathbf{U} \in \mathbb{R}^{I_n \times R}$ is denoted by $\boldsymbol{\mathcal{Z}} = \boldsymbol{\mathcal{Y}} \times_n \mathbf{U} \in \mathbb{R}^{I_1 \times \cdots \times I_{n-1} \times R \times I_{n+1} \times \cdots \times I_N}$. Products of a tensor $\boldsymbol{\mathcal{Y}}$ with a set of N matrices $\{\mathbf{U}^{(n)}\} = \left\{\mathbf{U}^{(1)}, \mathbf{U}^{(2)}, \ldots, \mathbf{U}^{(N)}\right\}$ are denoted by $\boldsymbol{\mathcal{Y}} \times \{\mathbf{U}^{(n)}\} \stackrel{\triangle}{=} \boldsymbol{\mathcal{Y}} \times_1 \mathbf{U}^{(1)} \times_2 \mathbf{U}^{(2)} \cdots \times_N \mathbf{U}^{(N)}$.

A tensor $\boldsymbol{\mathcal{X}} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$ is said in Kruskal form if $\boldsymbol{\mathcal{X}} = \sum_{r=1}^{R} \lambda_r \, \boldsymbol{a}_r^{(1)} \circ \boldsymbol{a}_r^{(2)} \circ \cdots \circ \boldsymbol{a}_r^{(N)}$, where "$\circ$" denotes the outer product, $\mathbf{A}^{(n)} = [\boldsymbol{a}_1^{(n)}, \boldsymbol{a}_2^{(n)}, \ldots, \boldsymbol{a}_R^{(n)}] \in \mathbb{R}^{I_n \times R}$ are factor matrices, $\boldsymbol{a}_r^{(n)T} \boldsymbol{a}_r^{(n)} = 1$, for $r = 1, \ldots, R$ and $n = 1, \ldots, N$, and $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_R > 0$.

A tensor $\boldsymbol{\mathcal{X}} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$ has multilinear rank-$(R_1, R_2, \ldots, R_N)$ if $\mathit{rank}(\mathbf{X}_{(n)}) = R_n \leq I_n$ for $n = 1, \ldots, N$, and can be expressed in the Tucker form as

$$\boldsymbol{\mathcal{X}} = \sum_{r_1=1}^{R_1} \sum_{r_2=1}^{R_2} \cdots \sum_{r_n=1}^{R_N} g_{r_1 r_2 \ldots r_N} \, \boldsymbol{a}_{r_1}^{(1)} \circ \boldsymbol{a}_{r_2}^{(2)} \circ \cdots \circ \boldsymbol{a}_{r_N}^{(N)}, \tag{1}$$

where $\boldsymbol{\mathcal{G}} = [g_{r_1 r_2 \ldots r_N}]$, and $\mathbf{A}^{(n)}$ are of full column rank. For compact expression, $[\![\boldsymbol{\lambda}; \{\mathbf{A}^{(n)}\}]\!]$ denotes a Kruskal tensor, where $[\![\boldsymbol{\mathcal{G}}; \{\mathbf{A}^{(n)}\}]\!]$ represents a Tucker tensor.

The main focus of this paper is a block deflation which splits a rank-R CPD into two sub rank-K and rank-$(R-K)$ CPDs. This tensor decomposition is a particular case of the block tensor decomposition [5] but with only two blocks

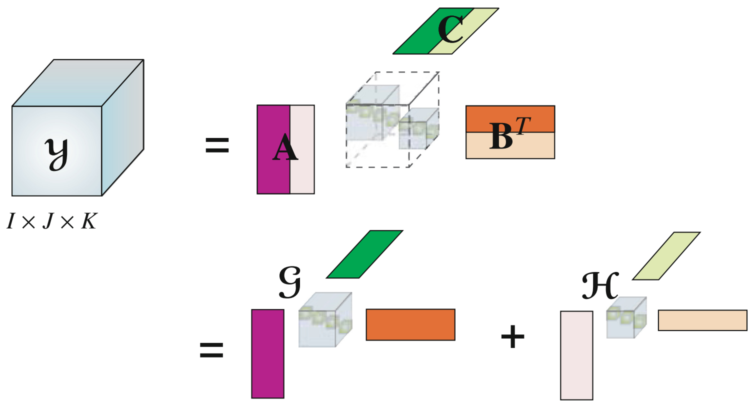

Fig. 1. Rank splitting for the CP decomposition of a rank-R tensor into two multilinear rank-$(K, \dots, K)$ and rank-$(R-K, \dots, R-K)$ tensors $\mathcal{G}$ and $\mathcal{H}$.

of multilinear rank-(K, K, K) and rank-$(R-K, R-K, R-K)$ as illustrated in Fig. 1. That is

$$\mathcal{Y} \approx [\![\mathcal{G}; \mathbf{U}^{(1)}, \mathbf{U}^{(2)}, \dots, \mathbf{U}^{(N)}]\!] + [\![\mathcal{H}; \mathbf{V}^{(1)}, \mathbf{V}^{(2)}, \dots, \mathbf{V}^{(N)}]\!] + \mathcal{E} \tag{2}$$

where $\mathbf{U}^{(n)}$ and $\mathbf{V}^{(n)}$ are matrices of size $I_n \times K$ and $I_n \times (R-K)$, respectively. Following this tensor decomposition, decomposition of a rank-R tensor can proceed simultaneously through decompositions of sub-tensors with smaller ranks.

For this kind of tensor decomposition and block tensor deflation, we can use the ALS algorithm [5] or the non-linear least squares (NLS) algorithm [6] developed for the multilinear rank-(L_r, M_r, N_r) block tensor decomposition with two blocks. However, these existing algorithms are expensive due to a large number of parameters of the two core tensors $\mathcal{G}$ and $\mathcal{H}$. The proposed algorithm will estimate only four vectors of length R per dimension whereas the core tensors $\mathcal{G}$ and $\mathcal{H}$ need not be estimated.

We will first introduce an orthogonal normalisation for the block tensor deflation, then state the correctness of the proposed deflation scheme.

Lemma 1 (Orthogonal Normalization for Rank Splitting). *Given a decomposition of* $\mathcal{Y}$ *as* $\mathcal{Y} \approx [\![\mathcal{G}; \mathbf{U}^{(1)}, \mathbf{U}^{(2)}, \dots, \mathbf{U}^{(N)}]\!] + [\![\mathcal{H}; \mathbf{V}^{(1)}, \mathbf{V}^{(2)}, \dots, \mathbf{V}^{(N)}]\!]$, *where* $\mathbf{U}^{(n)} \in \mathbb{R}^{I_n \times (K)}$ *and* $\mathbf{V}^{(n)} \in \mathbb{R}^{I_n \times (R-K)}$, $K \le R-K$, *one can construct an equivalent decomposition, denoted by tildas, which has the same approximation error, such that*

- $[\![\mathcal{G}; \{\mathbf{U}^{(n)}\}]\!] = [\![\widetilde{\mathcal{G}}; \{\widetilde{\mathbf{U}}^{(n)}\}]\!]$, $[\![\mathcal{H}; \{\mathbf{V}^{(n)}\}]\!] = [\![\widetilde{\mathcal{H}}; \{\widetilde{\mathbf{V}}^{(n)}\}]\!]$
- $\widetilde{\mathbf{U}}^{(n)}$ *and* $\widetilde{\mathbf{V}}^{(n)}$ *are orthogonal, i.e.,* $(\widetilde{\mathbf{U}}^{(n)})^T \widetilde{\mathbf{U}}^{(n)} = \mathbf{I}_K$ *and* $(\widetilde{\mathbf{V}}^{(n)})^T \widetilde{\mathbf{V}}^{(n)} = \mathbf{I}_{R-K}$.
- *and obey conditions* $(\widetilde{\mathbf{U}}^{(n)})^T \widetilde{\mathbf{V}}^{(n)} = [\operatorname{diag}\{\boldsymbol{\sigma}_n\}, \mathbf{0}_{R-2K}]$ *where* $\boldsymbol{\sigma}_n = [\sigma_{n,1}, \dots, \sigma_{n,K}] \in \mathbb{R}^K$ *and* $0 \le \sigma_{n,r} < 1$.

Theorem 1 (Rank Splitting). *A rank-R tensor* $\mathcal{Y} = [\![\boldsymbol{\beta}; \{\mathbf{B}^{(n)}\}]\!]$ *has an exact decomposition* $\mathcal{Y} = [\![\mathcal{G}; \mathbf{U}^{(1)}, \ldots, \mathbf{U}^{(N)}]\!] + [\![\mathcal{H}; \mathbf{V}^{(1)}, \ldots, \mathbf{V}^{(N)}]\!]$ *as in (2) where* $\mathbf{U}^{(n)} \in \mathbb{R}^{I_n \times K}$ *and* $\mathbf{V}^{(n)} \in \mathbb{R}^{I_n \times (R-K)}$, $K \leq R - K$ *and*

- *at least two factor matrices* $\mathbf{B}^{(n)} \in \mathbb{R}^{I_n \times R}$ *are of full column rank,*
- $\mathcal{G}$ *has multilinear rank-*$(K, \ldots, K)$*.*

Then $\mathcal{G}$ *is a tensor of rank-K and* $\mathcal{H}$ *of rank* $(R - K)$*.*

Proofs of Lemma 1 and Theorem 1 are provided in the full version of this paper [7].

3 Alternating Subspace Update Algorithm

In this section, we consider order-3 tensors of size $R \times R \times R$. Tensors of larger and unequal sizes should be compressed to this size using the Tucker decomposition [8–10]. We will develop an algorithm for the block tensor deflation which reduces the rank by $K = 2$. For this particular case, the core tensor $\mathcal{G}$ is size of $2 \times 2 \times 2$, and the core tensor $\mathcal{H}$ of size $(R-2) \times (R - 2 \times (R-2)$. The factor matrices $\mathbf{U}^{(n)}$ and $\mathbf{V}^{(n)}$ are of size $R \times 2$ and $R \times (R-2)$, respectively. The rank-2 block deflation has an advantage over the rank-1 tensor deflation when factor matrices have two nearly collinear components.

We denote matrices $\bar{\mathbf{V}}^{(n)} = [\boldsymbol{v}_1^{(n)}, \boldsymbol{v}_2^{(n)}]$ which comprise the first two columns of $\mathbf{V}^{(n)}$, and perform reparameterization of $\mathbf{U}^{(n)}$ as

$$\mathbf{U}^{(n)} = \mathbf{W}^{(n)} \operatorname{diag}(\boldsymbol{\xi}_n) + \bar{\mathbf{V}}^{(n)} \operatorname{diag}(\boldsymbol{\sigma}_n), \tag{3}$$

where $\boldsymbol{\xi}_n = [\xi_{n1}, \xi_{n2}]^T$, $\xi_{nr} = \sqrt{1 - \sigma_{nr}^2}$, and $\mathbf{W}^{(n)} = [\boldsymbol{w}_1^{(n)}, \boldsymbol{w}_2^{(n)}]$ of size $R \times 2$. $[\mathbf{W}^{(n)}, \mathbf{V}^{(n)}]$ are orthonormal matrices of size $R \times R$, i.e., $[\mathbf{W}^{(n)}, \mathbf{V}^{(n)}]^T [\mathbf{W}^{(n)}, \mathbf{V}^{(n)}] = \mathbf{I}_R$.

Consider the following criterion to be minimized,

$$D = \frac{1}{2} \|\mathcal{Y} - \mathcal{G} \times \{\mathbf{U}^{(n)}\} - \mathcal{H} \times \{\mathbf{V}^{(n)}\}\|_F^2. \tag{4}$$

We will later simplify the objective function in (4) by replacing the core tensors by their closed-form expressions and applying the above reparameterization. The objective function will finally depend only on $\mathbf{W}^{(n)}$, $\bar{\mathbf{V}}^{(n)}$ and $\boldsymbol{\sigma}_n$ for $n = 1, 2, 3$.

3.1 Closed-Form Expressions for the Core Tensors

From the model (2) and the cost function D in (4), we can derive closed-form expressions for $\mathcal{H}$ and $\mathcal{G}$ as

$$\mathcal{H} = \mathcal{Y} \times \{\mathbf{V}^{(n)T}\} - \mathcal{G} \times \left\{ \begin{bmatrix} \operatorname{diag}(\boldsymbol{\sigma}_n) \\ \mathbf{0}_{(R-2) \times 2} \end{bmatrix} \right\}, \tag{5}$$

$$\mathcal{G} = \left(\mathcal{Y} \times \{\mathbf{U}^{(n)T}\} - \left(\mathcal{Y} \times \{\bar{\mathbf{V}}^{(n)T}\}\right) \circledast \mathcal{S}\right) \oslash (1 - \mathcal{S} \circledast \mathcal{S}), \tag{6}$$

where $\mathcal{S} = \boldsymbol{\sigma}_1 \circ \boldsymbol{\sigma}_2 \circ \boldsymbol{\sigma}_3$ is a rank-1 tensor of size $2 \times 2 \times 2$, $\circledast$ and $\oslash$ represent the Hadamard (element-wise) product and division, respectively.

We replace $\mathcal{H}$ in the cost function (4) by its closed-form in (5), and rewrite D as

$$D = \frac{1}{2}\|\mathcal{Y} - \mathcal{Y} \times \left\{\mathbf{V}^{(n)}\mathbf{V}^{(n)T}\right\} - \mathcal{G} \times \{\mathbf{U}^{(n)}\} + \mathcal{G} \times \{\bar{\mathbf{V}}^{(n)} \operatorname{diag}(\boldsymbol{\sigma}_n)\}\|_F^2$$
$$= \frac{1}{2}\left(\|\mathcal{Y}\|_F^2 - \|\mathcal{Y} \times \left\{\mathbf{V}^{(n)}\mathbf{V}^{(n)T}\right\}\|_F^2 - \langle \mathcal{G} \circledast (1 - \mathcal{S} \circledast \mathcal{S}), \mathcal{G}\rangle\right). \quad (7)$$

For an index $n \in \{1, 2, 3\}$, define n_1 and n_2 with $n_1 < n_2$ as its complement in $\{1, 2, 3\}$, i.e., $\{n, n_1, n_2\} = \{1, 2, 3\}$. Put $\boldsymbol{t}_{r,s}^{(n)} = \mathcal{Y} \times_{n_1} \boldsymbol{u}_r^{(n_1)T} \times_{n_2} \boldsymbol{u}_s^{(n_2)T}$, $\boldsymbol{z}_{r,s}^{(n)} = \mathcal{Y} \times_{n_1} \boldsymbol{v}_r^{(n_1)T} \times_{n_2} \boldsymbol{v}_s^{(n_2)T}$, and $\boldsymbol{d}_{r,s}^{(n)} = \boldsymbol{t}_{r,s}^{(n)} - \boldsymbol{z}_{r,s}^{(n)}\, \sigma_{n_1,r}\, \sigma_{n_2,s}$. The objective function in (7) can be expressed as

$$D = \frac{1}{2}\left(\|\mathcal{Y}\|_F^2 - \|\mathcal{Y} \times \left\{\mathbf{V}^{(n)}\mathbf{V}^{(n)T}\right\}\|_F^2 - \sum_{r_1,r_2,r_3=1}^{2} \frac{(\xi_{1,r_1} \boldsymbol{w}_{r_1}^{(1)T} \boldsymbol{t}_{r_2,r_3}^{(1)} + \sigma_{1,r_1} \boldsymbol{v}_{r_1}^{(1)T} \boldsymbol{d}_{r_2,r_3}^{(1)})^2}{1 - \sigma_{1,r_1}^2 \sigma_{2,r_2}^2 \sigma_{3,r_3}^2}\right). \quad (8)$$

3.2 Estimation of $\boldsymbol{\sigma}_n$

We begin with deriving update rules for $\boldsymbol{\sigma}_1 = [\sigma_{1,1}, \sigma_{1,2}]$. As shown in the cost function in (8), the parameters $\boldsymbol{\sigma}_1$ involve only the third term. In order to estimate $\boldsymbol{\sigma}_1$, we keep other parameters fixed. Then minimization of the cost function (8) leads to maximization of the following function of $\boldsymbol{\sigma}_1$

$$\max_{\sigma_{1,1},\sigma_{1,2}} \quad \sum_{r_1=1}^{2} \sum_{r_2=1}^{2} \sum_{r_3=1}^{2} \frac{(\xi_{1,r_1} \boldsymbol{w}_{r_1}^{(1)T} \boldsymbol{t}_{r_2,r_3}^{(1)} + \sigma_{1,r_1} \boldsymbol{v}_{r_1}^{(1)T} \boldsymbol{d}_{r_2,r_3}^{(1)})^2}{1 - \sigma_{1,r_1}^2 \sigma_{2,r_2}^2 \sigma_{3,r_3}^2}. \quad (9)$$

Each σ_{1,r_1} is found as $\sigma_{1,r_1} = 1/\sqrt{1 + x_{r_1}^2}$ where x_{r_1} is solution to the problem

$$x_{r_1} = \arg\max_{x} \sum_{r_2=1}^{2} \sum_{r_3=1}^{2} \frac{(\alpha_{r_2,r_3}\, x + \beta_{r_2,r_3})^2}{x^2 + 1 - \sigma_{2,r_2}^2 \sigma_{3,r_3}^2} \quad (10)$$

$\alpha_{r_2,r_3} = \boldsymbol{w}_{r_1}^{(1)T} \boldsymbol{t}_{r_2,r_3}^{(1)}$ and $\beta_{r_2,r_3} = \boldsymbol{v}_{r_1}^{(1)T} \boldsymbol{d}_{r_2,r_3}^{(1)}$. The optimal x_{r_1} is a root of a polynomial of degree-8. The others $\sigma_{n,r}$ can be estimated similarly.

3.3 Estimation of Orthogonal Components $\mathbf{W}^{(n)}$ and $\mathbf{V}^{(n)}$

This section presents update rules which preserve orthogonality constrains on $\mathbf{W}^{(n)}$ and $\mathbf{V}^{(n)}$. Indeed we only need to update $\mathbf{W}^{(n)}$ and the first two column vectors $\bar{\mathbf{V}}^{(n)} = [\boldsymbol{v}_1^{(n)}, \boldsymbol{v}_2^{(n)}]$, whereas the last $(R-4)$ columns $[\boldsymbol{v}_3^{(n)}, \ldots, \boldsymbol{v}_{R-2}^{(n)}]$ are chosen as arbitrary orthogonal complement to $[\mathbf{W}^{(n)}, \bar{\mathbf{V}}^{(n)}]$.

Since $\mathbf{V}^{(n)}\mathbf{V}^{(n)T} = \mathbf{I}_R - \mathbf{W}^{(n)}\mathbf{W}^{(n)T}$, we have

$$\|\mathcal{Y} \times \left\{\mathbf{V}^{(n)}\mathbf{V}^{(n)T}\right\}\|_F^2 = \operatorname{tr}(\boldsymbol{\Phi}_n) - \operatorname{tr}(\mathbf{W}^{(n)T} \boldsymbol{\Phi}_n \mathbf{W}^{(n)}) \quad (11)$$

where $\boldsymbol{\Phi}_n = \mathbf{Y}_{(n)} \left(\bigotimes_{k \neq n} \mathbf{V}^{(n)} \mathbf{V}^{(k)T} \right) \mathbf{Y}_{(n)}^T$ are matrices of size $R \times R$. The cost function in (8) is rewritten as

$$D = \frac{1}{2} \left(\|\mathcal{Y}\|_F^2 - \operatorname{tr}(\boldsymbol{\Phi}_n) + \sum_{r=1}^{2} \boldsymbol{w}_r^{(n)T} \mathbf{Q}_{n,r} \boldsymbol{w}_r^{(n)} - \boldsymbol{v}_r^{(n)T} \mathbf{F}_{n,r} \boldsymbol{v}_r^{(n)} - 2\boldsymbol{w}_r^{(n)T} \mathbf{K}_{n,r} \boldsymbol{v}_r^{(n)} \right) \quad (12)$$

where

$$\mathbf{Q}_{n,r} = \boldsymbol{\Phi}_n - \xi_{n,r}^2 \sum_{k,l} \frac{\boldsymbol{t}_{k,l}^{(n)} \, \boldsymbol{t}_{k,l}^{(n)T}}{1 - \sigma_{n,r}^2 \sigma_{n_1,k}^2 \sigma_{n_2,l}^2}, \qquad \mathbf{F}_{n,r} = \sigma_{n,r}^2 \sum_{k,l} \frac{\boldsymbol{d}_{k,l}^{(n)} \, \boldsymbol{d}_{k,l}^{(n)T}}{1 - \sigma_{n,r}^2 \sigma_{n_1,k}^2 \sigma_{n_2,l}^2}, \quad (13)$$

$$\mathbf{K}_{n,r} = \xi_{n,r} \sigma_{n,r} \sum_{k,l} \frac{\boldsymbol{t}_{k,l}^{(n)} \, \boldsymbol{d}_{k,l}^{(n)T}}{1 - \sigma_{n,r}^2 \sigma_{n_1,k}^2 \sigma_{n_2,l}^2}. \quad (14)$$

It follows that $\mathbf{W}^{(n)}$ and $\bar{\mathbf{V}}^{(n)}$ are solutions to the following quadratic optimisation

$$\min f(\mathbf{W}^{(n)}, \bar{\mathbf{V}}^{(n)}) = \frac{1}{2} \left(\sum_{r=1}^{2} \boldsymbol{w}_r^{(n)T} \mathbf{Q}_{n,r} \boldsymbol{w}_r^{(n)} - \boldsymbol{v}_r^{(n)T} \mathbf{F}_{n,r} \boldsymbol{v}_r^{(n)} - 2\boldsymbol{w}_r^{(n)T} \mathbf{K}_{n,r} \boldsymbol{v}_r^{(n)} \right) \quad (15)$$
$$\text{subject to} \quad [\mathbf{W}^{(n)} \, \bar{\mathbf{V}}^{(n)}]^T [\mathbf{W}^{(n)} \, \bar{\mathbf{V}}^{(n)}] = \mathbf{I}_4.$$

Following the Crank-Nicholson-like scheme [11], we can update the orthogonal matrices $\mathbf{X}_n = [\mathbf{W}^{(n)}, \bar{\mathbf{V}}^{(n)}]$ with $\mathbf{X}_n^T \mathbf{X}_n = \mathbf{I}_4$ using the following rules

$$\mathbf{X}_n \leftarrow \mathbf{X}_n - 2\tau [\mathbf{G}_f, \mathbf{X}_n] \left(\mathbf{I}_8 + \tau \begin{bmatrix} \mathbf{X}_n^T \mathbf{G}_f & \mathbf{I}_4 \\ -\mathbf{G}_f^T \mathbf{G}_f & -\mathbf{G}_f^T \mathbf{X}_n \end{bmatrix} \right)^{-1} \begin{bmatrix} \mathbf{I}_4 \\ -\mathbf{G}_f^T \mathbf{X}_n \end{bmatrix}, \quad (16)$$

where $\mathbf{G}_f = [\boldsymbol{g}_{f,\boldsymbol{w}_1^{(n)}}, \boldsymbol{g}_{f,\boldsymbol{w}_2^{(n)}}, \boldsymbol{g}_{f,\boldsymbol{v}_1^{(n)}}, \boldsymbol{g}_{f,\boldsymbol{w}_2^{(n)}}]$ of size $R \times 4$ are the first order derivatives of the function $f(\mathbf{W}^{(n)}, \bar{\mathbf{V}}^{(n)})$ with respect to $[\mathbf{W}^{(n)}, \bar{\mathbf{V}}^{(n)}]$

$$\boldsymbol{g}_{f,\boldsymbol{w}_r^{(n)}} = \frac{\partial f}{\partial \boldsymbol{w}_r^{(n)}} = \mathbf{Q}_{n,r} \boldsymbol{w}_r^{(n)} - \mathbf{K}_{n,r} \boldsymbol{v}_r^{(n)}, \quad \boldsymbol{g}_{f,\boldsymbol{v}_r^{(n)}} = \frac{\partial f}{\partial \boldsymbol{v}_r^{(n)}} = -\mathbf{F}_{n,r} \boldsymbol{v}_r^{(n)} - \mathbf{K}_{n,r}^T \boldsymbol{w}_r^{(n)}, \quad (17)$$

and $\boldsymbol{\Gamma}_n = \mathbf{X}_n^T \mathbf{G}_f$ and $\tau > 0$ is a step size chosen using the Barzilai-Borwein method [12].

The most expensive step in the ASU algorithm is computation of the matrices $\boldsymbol{\Phi}_n = \mathbf{Y}_{(n)} \left(\bigotimes_{k \neq n} \mathbf{V}^{(n)} \mathbf{V}^{(k)T} \right) \mathbf{Y}_{(n)}^T$. A naive computation method might cost $\mathcal{O}(R^4)$. We present a more efficient computation which requires a cost of order $\mathcal{O}(R^3)$

$$\begin{aligned} \boldsymbol{\Phi}_n &= \mathbf{Y}_{(n)} \left((\mathbf{I} - \mathbf{W}^{(n_2)} \mathbf{W}^{(n_2)T}) \otimes (\mathbf{I} - \mathbf{W}^{(n_1)} \mathbf{W}^{(n_1)T}) \right) \mathbf{Y}_{(n)}^T \\ &= \mathbf{Y}_{(n)} \mathbf{Y}_{(n)}^T - \langle \mathcal{Y} \times_{n_1} \mathbf{W}^{(n_1)}, \mathcal{Y} \times_{n_1} \mathbf{W}^{(n_1)} \rangle_{n_1,n_2} - \langle \mathcal{Y} \times_{n_2} \mathbf{W}^{(n_2)}, \mathcal{Y} \times_{n_2} \mathbf{W}^{(n_2)} \rangle_{n_1,n_2} \\ &\quad - \langle \mathcal{Y} \times_{n_1} \mathbf{W}^{(n_1)} \times_{n_2} \mathbf{W}^{(n_2)}, \mathcal{Y} \times_{n_1} \mathbf{W}^{(n_1)} \times_{n_2} \mathbf{W}^{(n_2)} \rangle_{n_1,n_2}, \end{aligned} \quad (18)$$

where $\{n_1 < n_2\} = \{1, 2, 3\} \setminus \{n\}$.

The proposed Alternating Subspace Update (ASU) algorithm is summarized in Algorithm 1.

Algorithm 1. `Alternating Subspace Update (ASU)`

Input: Data tensor $\boldsymbol{\mathcal{Y}}$: $(R \times R \times R)$ of rank R
Output: A rank-(2,2,2) tensor $[\![\boldsymbol{\mathcal{G}}; \{\mathbf{U}^{(n)}\}]\!]$ and rank-$(R-2, R-2, R-2)$ tensor $[\![\boldsymbol{\mathcal{H}}; \{\mathbf{V}^{(n)}\}]\!]$
begin
- Initialise components $\mathbf{U}^{(n)}$ and $\mathbf{V}^{(n)}$
- Orthogonal normalization to $\mathbf{U}^{(n)}$ and $\mathbf{V}^{(n)}$ and compute $\boldsymbol{\sigma}_n = [\sigma_{n,1}, \sigma_{n,2}]^T$ and $\mathbf{W}^{(n)}$
- **repeat**
  - **for** $n = 1, 2, 3$ **do**
    - **for** $r = 1, 2$ **do** Update $\sigma_{n,r} = \frac{1}{\sqrt{1+x^2}}$ where x is solved as in (10)
    - Compute $\mathbf{G}_f$ as in (17), $\boldsymbol{\Gamma}_n = \mathbf{X}_n^T \mathbf{G}_f$ where $\mathbf{X}_n = [\mathbf{W}^{(n)}, \bar{\mathbf{V}}^{(n)}]$
    - Update $\mathbf{X}_n = [\mathbf{W}^{(n)}, \bar{\mathbf{V}}^{(n)}]$ as in (16)
    - $\mathbf{U}^{(n)} \leftarrow \mathbf{W}^{(n)} \operatorname{diag}(\boldsymbol{\xi}_n) + \bar{\mathbf{V}}^{(n)} \operatorname{diag}(\boldsymbol{\sigma}_n)$
- **until** *a stopping criterion is met*
- **for** $n = 1, \ldots, N$ **do** Select $\mathbf{V}^{(n)}_{3:R-2}$ as an orthogonal complement of $[\mathbf{W}^{(n)}, \bar{\mathbf{V}}^{(n)}]$
- Compute output $\boldsymbol{\mathcal{G}}$ and $\boldsymbol{\mathcal{H}}$ as in (6) and (5)

4 Simulations

Example 1 *[Decomposition of Small Tensors Admitting the CP Model].* In this first example, we illustrate the block deflation of tensor of size $R \times R \times R$ and of rank R where $R = 10, 20, 30$. The weight coefficients λ_r were set to 1, whereas collinearity degrees between components $\boldsymbol{a}_r^{(n)}$ and $\boldsymbol{a}_s^{(n)}$ for all $r \neq s$ were set to c in the range [0, .9], $\boldsymbol{a}_r^{(n)T} \boldsymbol{a}_s^{(n)} = c$ and $\boldsymbol{a}_r^{(n)T} \boldsymbol{a}_r^{(n)} = 1$ for all n. We compare the ASU algorithm with the ALS algorithm [5] for the multilinear rank-(L_r, M_r, N_r) block tensor decomposition with two blocks. For this problem, one can use the non-linear least squares (NLS) algorithm [6]. However, as similar to the ALS algorithm [5], the NLS algorithm needs to estimate two core tensors and full factor matrices. Hence this algorithm is much more expensive than the ASU algorithm. Simulations were run on a Macbook-air laptop having 4 GB memory and a 1.8 GHz core i7. Due to space and time consuming, the ALS [5] was only ran in simulations for $R = 10$.

The algorithms were initialised by the same values generated using the Direct Trilinear Decomposition (DTLD) [13]. The algorithms ran until differences between consecutive approximation errors were small enough, $|\varepsilon_k - \varepsilon_{k+1}| \leq 10^{-6}\, \varepsilon_k$ where $\varepsilon = \|\boldsymbol{\mathcal{Y}} - \hat{\boldsymbol{\mathcal{Y}}}\|_F^2$, or when the number of iterations exceeded 1000. Performances were assessed through the squared angular error SAE in estimation of components $\boldsymbol{a}^{(n)}$ SAE $= \arccos\left(\frac{\boldsymbol{a}^T \hat{\boldsymbol{a}}}{\|\boldsymbol{a}\|_2 \|\hat{\boldsymbol{a}}\|_2}\right)^2$. There were 100 independent

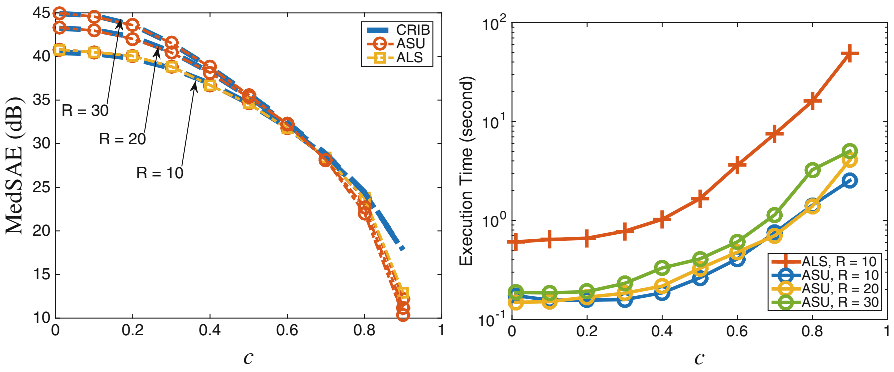

Fig. 2. Comparison of median SAEs and execution times of the ASU and ALS algorithms [5] in decomposition of tensors of size $R \times R \times R$ and rank R where $R = 10$, 20 and 30 for Example 1.

Table 1. Comparison of execution times of the ASU algorithm to extract two components from high rank-R tensors, and those of the CP-FastALS algorithm for Example 2.

	Execution time (second)					
	$c = 0.1$	0.2	0.3	0.4	0.5	0.6
$R = 300$						
ASU	3.81	3.66	3.76	3.82	3.89	3.77
CP-FastALS	530.6	543.5	537.6	537.6	541.9	539.2
$R = 500$						
ASU	38.4	16.7	16.5	16.9	16.8	17.1
CP-FastALS	3658	3672	3679	3693	3678	3669

runs for each rank $R = 10, 20$ and 30. The Gaussian noise was added into the tensor with signal-noise-ratio SNR = 30 dB.

Figure 2 shows median SAE (MedSAE) in dB ($-10 \log_{10} SAE$) obtained by ASU and ALS [5] compared with the Cramér-Rao Induced bound (CRIB) [14] on the squared angular error. The algorithms succeeded in most cases, but failed only when $c = 0.9$. For such difficult scenarios, CRIB on SAE was about 17.8 dB, indicating an angular error of 7.4 degrees between the original and estimated components. We note that in practice, it is hard to estimate a component with CRIB less than 20 dB, i.e., angular error of 5.7 degrees [15].

In Fig. 2, we compare execution times (in second) of algorithms for different ranks. Since the decomposition became more difficult when c was close to 1, running times of algorithms increased as shown in Fig. 2 (right). The ASU algorithm was on average 8 times faster than ALS [5] when $R = 10$.

The results confirmed high speed and accuracy of the proposed ASU algorithm.

Example 2 *[Decomposition of Large-Scale Tensors with High Rank].* This example illustrates an advantage of ASU over existing algorithms for the ordinary CPD in decomposition of large-scale tensors with relatively high rank $R = 300$ and 500. We generated rank-R synthetic tensors of size $R \times R \times R$ as in the previous example. Components $\boldsymbol{a}_r^{(n)}$ and $\boldsymbol{a}_s^{(n)}$ for $r \neq s$ have identical collinearity degrees, i.e., $\boldsymbol{a}_r^{(n)T} \boldsymbol{a}_s^{(n)} = c$ where $c = 0.1, 0.2, \ldots, 0.6$. The Gaussian noise was at SNR = 30 dB. Simulations were run on a computer consisted of Intel Xeon 2 processors clocked at 3.33 GHz, 64 GB of main memory. Comparison of execution times of ASU and FastALS [16] is given in Table 1.

5 Conclusions

We have introduced a rank-splitting scheme for CPD, and developed an ASU algorithm for rank-2 block deflation. The algorithm needs to estimate only 4 vectors and two scalars per dimension, and has a computational cost of $\mathcal{O}(R^3)$ for a tensor of size $R \times R \times R$. Further detail and applications of the block tensor deflation are described in the full paper [7]. The algorithm can be extended to higher order tensors, and decomposition with additional constraints. Algorithms for the block tensor deflation are implemented in the Matlab package TENSORBOX which is available online at: http://www.bsp.brain.riken.jp/~phan/tensorbox.php.

References

1. Stegeman, A., Comon, P.: Subtracting a best rank-1 approximation may increase tensor rank. Linear Algebra Appl. **433**(7), 1276–1300 (2010)
2. Phan, A.H., Tichavský, P., Cichocki, A.: Deflation method for CANDECOMP/PARAFAC tensor decomposition. In: IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), May 2014, pp. 6736–6740 (2014)
3. Phan, A.H., Tichavský, P., Cichocki, A.: Tensor deflation for CANDECOMP/PARAFAC. Part 1: alternating subspace update algorithm. IEEE Trans. Sig. Process. (2015). (in print)
4. Cichocki, A., Zdunek, R., Phan, A.H., Amari, S.: Nonnegative Matrix and Tensor Factorizations: Applications to Exploratory Multi-way Data Analysis and Blind Source Separation. Wiley, Chichester (2009)
5. De Lathauwer, L., Nion, D.: 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). Special issue tensor decompositions and applications
6. Sorber, L., Van Barel, M., De Lathauwer, L.: Structured data fusion. Technical report, ESAT-SISTA, Internal report 13–177 (2013)
7. Phan, A.H., Tichavský, P., Cichocki, A.: Tensor deflation for CANDECOMP/PARAFAC. Part 3: rank splitting. arXiv, CoRR, vol. abs/1506.04971 (2015)
8. De Lathauwer, L., Moor, B.D., Vandewalle, J.: On the best rank-1 and rank-(R1, R2, RN) approximation of higher-order tensors. SIAM J. Matrix Anal. Appl. **21**(4), 1324–1342 (2000)

9. Comon, P., Luciani, X., de Almeida, A.L.F.: Tensor decompositions, alternating least squares and other tales. J. Chemometr. **23**, 393–405 (2009)
10. Phan, A.H., Cichocki, A., Tichavský, P.: On fast algorithms for orthogonal Tucker decomposition. In: IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), May 2014, pp. 6766–6770 (2014)
11. Wen, Z., Yin, W.: A feasible method for optimization with orthogonality constraints. Math. Program. **142**, 397–434 (2013)
12. Barzilai, J., Borwein, J.M.: Two-point step size gradient methods. IMA J. Numer. Anal. **8**(1), 141–148 (1988)
13. Sanchez, E., Kowalski, B.: Tensorial resolution: a direct trilinear decomposition. J. Chemometr. **4**, 29–45 (1990)
14. Tichavský, P., Phan, A.H., Koldovský, Z.: Cramér-Rao-induced bounds for CANDECOMP/PARAFAC tensor decomposition. IEEE Trans. Sig. Process. **61**(8), 1986–1997 (2013)
15. Phan, A.H., Tichavský, P., Cichocki, A.: Low complexity damped Gauss-Newton algorithms for CANDECOMP/PARAFAC. SIAM J. Matrix Anal. Appl. **34**(1), 126–147 (2013)
16. Phan, A.H., Tichavský, P., Cichocki, A.: Fast alternating LS algorithms for high order CANDECOMP/PARAFAC tensor factorizations. IEEE Trans. Sig. Process. **61**(19), 4834–4846 (2013)