Tensor Deflation for CANDECOMP/PARAFAC—Part II: Initialization and Error Analysis

Anh-Huy Phan, *Member, IEEE*, Petr Tichavský, *Senior Member, IEEE*, and Andrzej Cichocki, *Fellow, IEEE*

***Abstract*—In Part I of the study of the tensor deflation for CANDECOMP/PARAFAC, we have shown that the rank-1 tensor deflation is applicable under some conditions. Part II of the study presents several initialization algorithms suitable for the algorithm proposed in Part I. In addition, Part II contains an algorithm for the case when one or more factor matrices in the estimated model is constrained to be orthogonal. Finally, Part II provides an error analysis of the tensor deflation algorithm, which shows that there is a marginal loss of accuracy of the deflation algorithm compared to the ordinary CP decomposition.**

***Index Terms*—CANDECOMP/PARAFAC, canonical polyadic decomposition (CPD), Cramér-Rao lower bound, tensor deflation.**

I. Introduction

TENSOR deflation in this paper and the papers [1]–[3] is developed for tensors of rank-R or approximate rank-R where R does not exceed the tensor dimensions, or tensors comprising rank-1 tensors and multilinear rank tensors in their block forms. The major aim of tensor deflation is to extract one rank-1 tensor from a high rank tensor instead of estimating all components in the CANDECOMP/PARAFAC (CP) decomposition. The tensor deflation can be used in a sequential process to extract several rank-1 tensors. This kind of tensor decomposition has applications in extraction of one or a few rank-1 tensors from tensors of high rank, or tracking markers (components) of interest in an online learning system.

In the Part I of this work, we have presented the Alternating Subspace Update (ASU) algorithm for rank-1 tensor deflation, and a sequential process to extract a few rank-1 tensors. The tensor deflation is related to the rank-1 plus multilinear rank-$(R-1,\ldots,R-1)$ block term decomposition (BTD) [4]. Practical results show that this kind of block term decomposition and the generalized rank-(L_r, M_r, N_r) BTD often get stuck in false local minima [4], [5]. In order to improve reliability of algorithms, one can perform the decomposition with different initial points, then chooses the result achieving the lowest approximation error. So far, multiple random initialization is most commonly used although this method is not much efficient. In order to achieve good performance with fast convergence, besides good optimization criteria, good initialization is always prerequisite. This second part will introduce new initialization methods for tensor deflation, which can find solutions for the noise-free case. In addition, the paper simplifies the model to have one or several orthogonal factor matrices. The orthogonality constraint in factor matrices improves uniqueness of the CANDECOMP/PARAFAC decomposition (CPD), and avoids degeneracy which often occurs in CPD [6]. Last but not least, we provide Cramér-Rao Lower Bound (CRB) for the tensor deflation. The CRB will serve as an accuracy indicator of the estimation of rank-1 tensor. In comparison with the CRB for CPD [7], we found that there is no significant loss in accuracy of tensor deflation using our approach.

Notation used in this paper is similar to that in Part I [1], [8]. For example, 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 $\otimes$. 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

$$\boldsymbol{\mathcal{Z}} = \boldsymbol{\mathcal{Y}} \times_n \mathbf{U} \in \mathbb{R}^{I_1\times\cdots\times I_{n-1}\times R\times I_{n+1}\times\cdots\times I_N}.$$

The rest of the paper is organized as follows. Some properties of the exact rank-1 tensor extraction are presented in Section II. Section III presents initialization methods based on the singular value decomposition (SVD), the joint eigenvalue decomposition. In Section IV, a variant of the ASU algorithm is developed for the case when one factor matrix is orthogonal. Cramér-Rao lower bound for the tensor deflation is derived in Section V. Simulations in Section VI verify initialization algorithms and analyze the loss of accuracy of the tensor deflation compared to the ordinary CP decomposition. Section VII concludes the paper.

II. Exact Rank-1 Tensor Extraction

As in Part I [1], for the rank-1 tensor deflation, we consider an order-N tensor $\boldsymbol{\mathcal{Y}}$ of size $R\times R\times\cdots\times R$, and a tensor decomposition of this tensor $\boldsymbol{\mathcal{Y}}$ into two tensor blocks of rank-1 and multilinear rank-$(R-1,\ldots,R-1)$

$$\boldsymbol{\mathcal{Y}} \approx [\![\lambda; \boldsymbol{a}^{(1)}, \boldsymbol{a}^{(2)}, \ldots, \boldsymbol{a}^{(N)}]\!] + [\![\boldsymbol{\mathcal{G}}; \mathbf{U}^{(1)}, \mathbf{U}^{(2)}, \ldots, \mathbf{U}^{(N)}]\!], \quad (1)$$

Manuscript received September 25, 2014; revised March 26, 2015 and June 16, 2015; accepted July 04, 2015. Date of publication July 20, 2015; date of current version October 06, 2015. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Adel Belouchrani. The work of P. Tichavský was supported by the Czech Science Foundation through project No. 14–13713S.

A.-H. Phan is with the Lab for Advanced Brain Signal Processing, Brain Science Institute, RIKEN, Wako 351-0198, Japan (e-mail: phan@brain.riken.jp).

P. Tichavský is with the Institute of Information Theory and Automation of the Czech Academy of Sciences, Prague, 182 08, Czech Republic (e-mail: tichavsk@utia.cas.cz)

A. Cichocki is with the is with the Lab for Advanced Brain Signal Processing, RIKEN BSI, Wako 351-0198, Japan, and also with the Systems Research Institute, PAS, Warsaw 00-447, Poland, and SKOLTECH, Moscow 143026, Russia (e-mail: a.cichocki@riken.jp).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TSP.2015.2458789

where $\mathbf{U}^{(n)}$ are of size $R \times (R-1)$, and the core tensor $\boldsymbol{\mathcal{G}}$ is of size $(R-1) \times (R-1) \times \cdots \times (R-1)$. If the tensor has sizes $I_n > R$, it should be compressed to the above size using the Tucker decomposition [9]. The CP decomposition and the rank-1 tensor extraction procedure are illustrated in [1, Fig. 2 of Part I].

According to the orthogonal normalization and rotation (Lemma 1 in [1]), the components $\boldsymbol{a}^{(n)}$ and factor matrices $\mathbf{U}^{(n)}$ can be assumed to be orthogonal, i.e., $(\mathbf{U}^{(n)})^T \mathbf{U}^{(n)} = \mathbf{I}_{R-1}$, $\rho_n = (\boldsymbol{u}_1^{(n)})^T \boldsymbol{a}^{(n)} \geq 0$ and $(\boldsymbol{u}_r^{(n)})^T \boldsymbol{a}^{(n)} = 0$ for $r = 2, \ldots, R-1$.

In this section, we present relation between components and factor matrices for tensors which admit the model in (1). The properties will be exploited to derive efficient initialization algorithms in Section III.

As similar to deriving the ASU algorithm in Part I [1], we express the components $\boldsymbol{a}^{(n)}$ through vectors $\boldsymbol{w}_n$ and the first columns $\boldsymbol{u}_n = \boldsymbol{u}_1^{(n)}$ of $\mathbf{U}^{(n)}$, that is

$$\boldsymbol{a}^{(n)} = \sqrt{1-\rho_n^2}\, \boldsymbol{w}_n + \rho_n\, \boldsymbol{u}_n, \tag{2}$$

where $\mathbf{V}_n = [\boldsymbol{w}_n, \mathbf{U}^{(n)}]$ are orthonormal matrices, i.e., $\mathbf{V}_n^T \mathbf{V}_n = \mathbf{I}_R$. Let

$$\mathbf{C}_n = \mathbf{Y}_{(n)}\, \mathbf{Y}_{(n)}^T \quad \in \mathbb{R}^{R \times R}, \tag{3}$$

$$\boldsymbol{s}_n = \mathbf{Y}_{(n)} \left(\bigotimes_{k \neq n} \boldsymbol{a}^{(k)} \right), \quad n = 1, \ldots, N. \tag{4}$$

We will show that parameters $\boldsymbol{w}_n$, $\mathbf{U}_n$ and ρ_n can be found in closed-form through the components $\boldsymbol{a}^{(n)}$ for the exact decomposition.

Lemma 1 (Scaling Coefficient of Rank-1 Tensor): The weight λ associated with rank-1 tensor $[\![\{\boldsymbol{a}^{(n)}\}]\!]$ is given by

$$\lambda = \frac{1}{\boldsymbol{s}_n^T\, \mathbf{C}_n^{-1} \boldsymbol{a}^{(n)}}, \quad n = 1, 2, \ldots, N. \tag{5}$$

Lemma 2 (Closed-Form Expressions for $\boldsymbol{w}_n$, ρ_n and $\boldsymbol{u}_n$):

$$\boldsymbol{w}_n = \frac{\mathbf{C}_n^{-1}\, \boldsymbol{s}_n}{\sqrt{\boldsymbol{s}_n^T\, \mathbf{C}_n^{-2}\, \boldsymbol{s}_n}} \tag{6}$$

$$\rho_n = \sqrt{1 - \frac{1}{\lambda^2\, \boldsymbol{s}_n^T\, \mathbf{C}_n^{-2}\, \boldsymbol{s}_n}} \tag{7}$$

$$\boldsymbol{u}_n = \frac{\boldsymbol{a}^{(n)} - \sqrt{1-\rho_n^2}\, \boldsymbol{w}_n}{\rho_n}. \tag{8}$$

Proofs of Lemma 1 and Lemma 2 are in Appendices A and B, respectively. The Lemmas give recipes, for a given rank-1 tensor $[\![\{\boldsymbol{a}^{(n)}\}]\!]$, how to compute the other parameters λ, $\boldsymbol{w}_n$ and $\boldsymbol{u}_n$. The last $(R-2)$ columns of $\mathbf{U}^{(n)}$ are arbitrarily orthonormal basis of orthogonal complement to $[\boldsymbol{w}_n, \boldsymbol{u}_n]$.

III. Initialization for Rank-1 Tensor Extraction

A. Initialization Using Leading Singular Values

1) Pre-Selection of Components: The simplest initialization method is that all parameters are initialized by random values. Unfortunately, this type of initialization method often makes algorithms getting stuck in false local minima, and thus requires a large number of trials. For tensors which admit the CP decomposition, the vectors $\boldsymbol{a}^{(n)}$ and factor matrices $\mathbf{U}^{(n)}$ can be initialized using the Direct Trilinear Decomposition (DTLD) [10] or the generalized rank annihilation method for (higher order) CPD [11], or simply but less efficiently using leading singular vectors of the tensor along modes, i.e., using the Higher Order SVD (HOSVD) or Higher Order Orthogonal Iteration (HOOI) algorithm [9].

Algorithm 1: Simple Initialization

Input: Data tensor $\boldsymbol{\mathcal{Y}}$: $(R \times R \times \cdots \times R)$, rank R

Output: A rank-1 tensor $[\![\boldsymbol{a}^{(1)}, \ldots, \boldsymbol{a}^{(N)}]\!]$ and $\{\mathbf{U}^{(1)}, \ldots, \mathbf{U}^{(N)}\}$

begin

1 Initialize $\mathbf{V}_n = [\boldsymbol{v}_{n1}, \boldsymbol{v}_{n2}, \ldots, \boldsymbol{v}_{nR}]$ by leading left singular vectors of $\mathbf{Y}_{(n)}$

for $r = 1, 2, \ldots, R$ **do**

for $n = 1, 2 \ldots, N$ **do**

2 $\boldsymbol{a}^{(n)} = \boldsymbol{v}_{n,r}$,

3 $\mathbf{U}^{(n)} = [\boldsymbol{v}_{n,1}, \ldots, \boldsymbol{v}_{n,r-1}, \boldsymbol{v}_{n,r+1}, \ldots, \boldsymbol{v}_{n,R}]$

% Run ASU with a small number of iterations

4 $[\{\boldsymbol{a}^{(n)}\}, \{\mathbf{U}^{(n)}\}, \varepsilon_r] = \texttt{asu}(\boldsymbol{\mathcal{Y}}, \{\boldsymbol{a}^{(n)}\}, \{\mathbf{U}^{(n)}\})$

5 Choose the initial point yielding the smallest approximation error $\varepsilon_{r^\star} = \min\{\varepsilon_1, \varepsilon_2, \ldots, \varepsilon_R\}$

For deflation of tensors which admit the block component model, we initialize components $\boldsymbol{a}^{(n)}$ of the rank-1 tensor, and factor matrices $\mathbf{U}^{(n)}$ of the multilinear rank-$(R-1)$ tensor, but do not need to initialize the weight coefficient λ of the rank-1 tensor and the core tensor $\boldsymbol{\mathcal{G}}$ of the second block.

If one block is considered to be fixed, then the other block can be obtained as the best rank-1 or multilinear rank-$(R-1)$ tensor of a certain residue tensor. One can also initialize components using leading singular eigenvectors of mode-n matricizations of $\boldsymbol{\mathcal{Y}}$ as in CPD or other matrix/tensor decompositions. However, simply choosing the first $(R-1)$ leading singular vectors for the factor matrices $\mathbf{U}^{(n)}$ and one for $\boldsymbol{a}^{(n)}$ may not give a good initial point. Instead, one can choose the components $\boldsymbol{a}^{(n)}$ as one of R leading singular eigenvectors of $\mathbf{Y}_{(n)}$, and the rest $(R-1)$ leading singular eigenvectors for $\mathbf{U}^{(n)}$ so that the ASU algorithm obtains the smallest approximation error after a small number of iterations (say 10). There are all R^N possible combinations of leading singular eigenvectors in all modes, but we only consider R combinations of singular eigenvectors in the same order. The procedure is summarized in Algorithm 1. Basically, complexity of the preselection is the same as that of the ASU algorithm, i.e., $\mathcal{O}(R^N)$ [1]. Together with the selection one among R rank-1 tensors, the computational cost is at most $O(R^{N+1})$. We note that since ASU in each run in Step 3 executes only a few iterations, the preselection, in general, is still fast. Experiment results show that the procedure improves the initial points, and thereby improves the obtained solution.

2) SVD-Based Initialization With Estimation of the Parameters ρ_n: According to the orthogonal normalization, we can initialize $\boldsymbol{a}^{(n)}$ and $\mathbf{U}^{(n)}$ by orthogonal vectors which are naturally the leading left singular vectors of mode-n matricizations $\mathbf{Y}_{(n)}$. For tensors which admit the CP decomposition, the method works well. However, when the tensors do not fully admit CPD, the method may be less efficient. Instead, we initialize $\boldsymbol{w}_n$ with a singular vector and $\mathbf{U}^{(n)}$ by the rest $(R-1)$

Algorithm 2: SVD Initialization with Estimation of Parameters ρ_n

Input: Data tensor $\mathcal{Y}$: $(R \times R \times \cdots \times R)$, rank R

Output: A rank-1 tensor $[\![\boldsymbol{a}^{(1)}, \ldots, \boldsymbol{a}^{(N)}]\!]$ and $\{\mathbf{U}^{(1)}, \ldots, \mathbf{U}^{(N)}\}$

begin

1 Initialize $[\boldsymbol{w}_n, \mathbf{U}^{(n)}]$ by singular eigenvectors of $\mathbf{Y}_{(n)}$

2 Initialize $0 \le \rho_n < 1$ $\quad / * \xi_n = \sqrt{1-\rho_n^2} * /$

3 $\mathcal{X} = \mathcal{Y} \times \{[\boldsymbol{w}_n, \boldsymbol{u}_n]^T\}$ $\quad / * \boldsymbol{u}_n = \mathbf{U}^{(n)}(:,1) * /$

repeat

for $n = 1, 2, \ldots, N$ **do**

4 $[x_{n1}, x_{n2}]^T = \mathcal{X} \times_{k \ne n} \{[\xi_n, \rho_n]^T\}$ $-[0, \rho_{-n}\mathcal{X}(2, \ldots, 2)]^T$

5 $\rho_n = \frac{|x_{n2}|}{\sqrt{x_{n1}^2(1-\rho_{-n}^2)^2 + x_{n2}^2}}$ $\quad / * \rho_{-n} = \prod_{k \ne n} \rho_k * /$

until *a stopping criterion is met*

6 **for** $n = 1, \ldots, N$ **do** $\boldsymbol{a}^{(n)} = \sqrt{1-\rho_n^2}\, \boldsymbol{w}_n + \rho_n \boldsymbol{u}_n$

ones where $\boldsymbol{a}^{(n)} = \xi_n \boldsymbol{w}_n + \rho_n \boldsymbol{u}_n$, $\xi_n = \sqrt{1-\rho_n^2}$, $\boldsymbol{u}_n$ are the first column vectors of $\mathbf{U}^{(n)}$ for $n = 1, \ldots, N$. Then we suggest to seek parameters $\rho_1, \rho_2, \ldots, \rho_N$ which can minimise the approximation error

$$D = \frac{1}{2}\|\mathcal{Y} - [\![\lambda; \{\boldsymbol{a}^{(n)}\}]\!] - [\![\mathcal{G}; \{\mathbf{U}^{(n)}\}]\!]\|_F^2$$
$$= \frac{1}{2}\left\|\mathcal{Y} - [\![\mathcal{Y}; \{\mathbf{U}^{(n)}\mathbf{U}^{(n)T}\}]\!] - \lambda\left([\![\{\boldsymbol{a}^{(n)}\}]\!] - [\![\rho; \{\boldsymbol{u}_n\}]\!]\right)\right\|_F^2,$$

where $\mathcal{G}$ is substituted by its optimum counterpart $\mathcal{G}_{opt} = \mathcal{T} - \lambda\rho t_1 \mathcal{E}_1$ [1], $\rho = \prod_n \rho_n$, $\mathcal{T} = \mathcal{Y} \times \{\mathbf{U}^{(n)T}\}$, and $\mathcal{E}_1$ is tensor of the same size as $\mathcal{G}$ but has only one entry $\mathcal{E}_1(1) = 1$, and zeros elsewhere. As similarly to deriving in the ASU algorithm [1], the optimum weight λ_{opt} is given in closed-form as

$$\lambda_{opt} = \frac{\mathcal{Y} \times \{\boldsymbol{a}^{(n)T}\} - \rho\, t_1}{1 - \rho^2}, \tag{9}$$

where $t_1 = \mathcal{Y} \times_1 \boldsymbol{u}_1^T \cdots \times_N \boldsymbol{u}_N^T$ is the first entry of $\mathcal{T}$. Parameters ρ_n for $n = 1, \ldots, N$ are sequentially updated as

$$\rho_n = \frac{|x_{n2}|}{\sqrt{x_{n1}^2(1-\rho_{-n}^2)^2 + x_{n2}^2}}, \tag{10}$$

where $\rho_{-n} = \prod_{k \ne n} \rho_k$, and

$$\begin{bmatrix} x_{n1} \\ x_{n2} \end{bmatrix} = \left(\mathcal{X} \times_{k \ne n} \left\{\begin{bmatrix} \xi_k \\ \rho_k \end{bmatrix}\right\}\right) - \begin{bmatrix} 0 \\ \rho_{-n} t_1 \end{bmatrix} \tag{11}$$

with $\mathcal{X} = \mathcal{Y} \times \{[\boldsymbol{w}_n, \boldsymbol{u}_n]^T\}$. Note that $t_1 = \mathcal{X}(2, \ldots, 2)$.

Pseudo-code of the SVD-based initialization algorithm is summarized in Algorithm 2. The initialization in Step 1 computes the left singular vectors of $\mathbf{Y}_{(n)}$ with complexity of $\mathcal{O}(R^3)$. The tensor $\mathcal{X}$ computed only once in Step 3 is of size $2 \times 2 \times \cdots \times 2$ with a cost of $\mathcal{O}(R^N)$. Small loop (Steps 4 and 5) runs a few iterations, usually less than 4, with a low cost per iteration of $O(2^N)$. Therefore, Algorithm 2 has complexity of $\mathcal{O}(R^N)$ which is dominated by computation of the tensor $\mathcal{X}$.

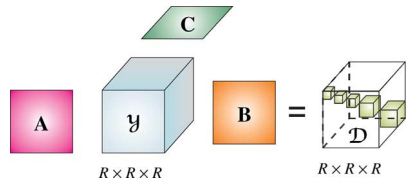

Fig. 1. Illustration of the tensor diagonalization as tool for initialization of the tensor deflation.

B. Initialization Using the Tensor Diagonalization (TEDIA)

An alternative initialization for the ASU algorithm consists in utilizing the tensor diagonalization TEDIA [5]. TEDIA was developed so far only for order-3 tensors. The method seeks factor matrices $\widetilde{\mathbf{A}}$, $\widetilde{\mathbf{B}}$ and $\widetilde{\mathbf{C}}$ as shown in Fig. 1 which transform the tensor $\mathcal{Y}$ into a block diagonal tensor by minimizing the sum of squared off-diagonal elements of the tensor $\mathcal{T} = \mathcal{Y} \times_1 \widetilde{\mathbf{A}} \times_2 \widetilde{\mathbf{B}} \times_3 \widetilde{\mathbf{C}}$. By inspecting the diagonal $\mathcal{T}(r,r,r)$ and corresponding off-diagonal coefficients, we can find suitable components $\mathbf{A}(:,r)$, $\mathbf{B}(:,r)$ and $\mathbf{C}(:,r)$ for the rank-1 tensor where $\mathbf{A} = \widetilde{\mathbf{A}}^{-1}$, and the rest components of $\mathbf{A}$, $\mathbf{B}$ and $\mathbf{C}$ for the second block. Each iteration of the TEDIA algorithm costs $\mathcal{O}(R^4)$ as the same fast ALS algorithm for CPD. In simulations (see Section IV), the initialization by TEDIA alone looks costly, but since it typically provides a better initial solution than the other algorithms, TEDIA + ASU has the complexity comparable to ASU with other initialization methods.

C. Initialization Based on Joint Eigenvalue Decomposition

In this section, we first introduce rank-1 deflation of two matrices based on the generalized eigenvalue decomposition. The method is then applied to tensors comprising only two slices, compressed from the data tensor. For simplicity, the methods will be derived for order-3 tensors, then extended to higher order tensors. In addition, the components $\boldsymbol{a}$, $\boldsymbol{b}$ and $\boldsymbol{c}$ are used in place of $\boldsymbol{a}^{(1)}$, $\boldsymbol{a}^{(2)}$, $\boldsymbol{a}^{(3)}$.

1) Rank-1 Deflation of Two Matrices:

Theorem 1: Two real-valued full-rank matrices $\mathbf{X}_1$ and $\mathbf{X}_2$ of size $R \times R$ can be simultaneously decomposed into

$$\mathbf{X}_1 = \alpha_1\, \boldsymbol{a}\, \boldsymbol{b}^T + \mathbf{U}\, \mathbf{G}_1\, \mathbf{V}^T, \tag{12}$$
$$\mathbf{X}_2 = \alpha_2\, \boldsymbol{a}\, \boldsymbol{b}^T + \mathbf{U}\, \mathbf{G}_2\, \mathbf{V}^T, \tag{13}$$

where $\boldsymbol{a}$ and $\boldsymbol{b}$ are eigenvectors associated with the same eigenvalue λ for matrix pencils $(\mathbf{X}_2^{-1}, \mathbf{X}_1^{-1})$ and $(\mathbf{X}_2^{-1^T}, \mathbf{X}_1^{-1^T})$, $\mathbf{G}_1$ and $\mathbf{G}_2$ are matrices of size $(R-1) \times (R-1)$ and have rank $(R-1)$.

Proof of Theorem 1 is in Appendix C.

2) Multiple Rank-1 Deflations of Two Matrices: For deflation of a tensor $\mathcal{Y}$, one can use the Tucker decomposition to compress the tensor $\mathcal{Y}$ to order-3 tensors $\mathcal{X}$ consisting of only two slices, e.g., $R \times R \times 2$, or $R \times 2 \times R$, and apply Theorem 1 to the two slices to obtain estimates of the components $\boldsymbol{a}^{(n)}$ and matrices $\mathbf{U}^{(n)}$. The tensor $\mathcal{X}$ can be of lower order than $\mathcal{Y}$. For simplicity, we consider an order-3 tensor $\mathcal{Y}$, and compress this tensor along mode-3 to yield two frontal slices $\mathbf{X}_1^{(1,2)}$ and $\mathbf{X}_2^{(1,2)}$ of size $R \times R$ which share a common rank-1 matrix

$$\mathbf{X}_1^{(1,2)} \approx \alpha_1^{(1,2)}\, \boldsymbol{a}\, \boldsymbol{b}^T + \mathbf{U}_1\, \breve{\mathbf{G}}_1\, \mathbf{U}_2^T,$$
$$\mathbf{X}_2^{(1,2)} \approx \alpha_2^{(1,2)}\, \boldsymbol{a}\, \boldsymbol{b}^T + \mathbf{U}_1\, \breve{\mathbf{G}}_2\, \mathbf{U}_2^T,$$

where $\breve{\mathbf{G}}_1$ and $\breve{\mathbf{G}}_2$ are of size $(R-1)\times(R-1)$, and $\boldsymbol{a}$, $\boldsymbol{b}$ can be approximated by generalized eigenvectors $\boldsymbol{a}_{1,2}$, $\boldsymbol{b}_{1,2}$ of matrix pencils $(\mathbf{X}_1^{(1,2)}, \mathbf{X}_2^{(1,2)})$ and $(\mathbf{X}_1^{(1,2)^T}, \mathbf{X}_2^{(1,2)^T})$. For real-valued tensors, there may not be any real-valued eigenvectors. One can convert the complex eigenvalues to real 2-by-2 blocks on the diagonal using Matlab's routine "cdf2rdf". Even when real eigenvalues exist, there may have multiple pairs of eigenvectors $(\boldsymbol{a}_r^{(1,2)}, \boldsymbol{b}_r^{(1,2)})$ which are associated with the same real eigenvalues. In order to select a good estimate to $\boldsymbol{a}$ and $\boldsymbol{b}$, we perform a further compression of $\boldsymbol{\mathcal{Y}}$ along mode-2 (or mode-3) and obtain estimates to $\boldsymbol{a}$ and $\boldsymbol{c}$ as eigenvectors $(\boldsymbol{a}_s^{(1,3)}, \boldsymbol{c}_s^{(1,3)})$. Loading components $(\boldsymbol{a}_r^{(1,2)}, \boldsymbol{b}_r^{(1,2)}, \boldsymbol{c}_s^{(1,3)})$ are chosen such that $\boldsymbol{a}_r^{(1,2)}$ is most correlated with $\boldsymbol{a}_s^{(1,3)}$, that is $\max\,(\boldsymbol{a}_r^{(1,2)})^T\,\boldsymbol{a}_s^{(1,3)}$.

Alternatively, one can perform three compressions to obtain pairs of eigenvectors $(\boldsymbol{a}_r^{(1,2)}, \boldsymbol{b}_r^{(1,2)})$, $(\boldsymbol{a}_s^{(1,3)}, \boldsymbol{c}_s^{(1,3)})$ and $(\boldsymbol{b}_t^{(2,3)}, \boldsymbol{c}_t^{(2,3)})$, and select a good estimate to $(\boldsymbol{a}, \boldsymbol{b}, \boldsymbol{c})$ such that

$$\max_{r,s,t} \quad ((\boldsymbol{a}_r^{(1,2)})^T\,\boldsymbol{a}_s^{(1,3)})\,((\boldsymbol{b}_r^{(1,2)})^T\,\boldsymbol{b}_t^{(2,3)})\,((\boldsymbol{c}_s^{(1,3)})^T\,\boldsymbol{c}_t^{(2,3)})\,. \tag{14}$$

Matrices $\mathbf{U}_1$, $\mathbf{U}_2$ and $\mathbf{U}_3$ can be computed as in proof of Theorem 1, or using the method in Section II. Computational cost of this initialization method is dominated by the compression of the tensor of size $R\times R\times R$ to ones of two slices. This requires a cost of order $\mathcal{O}(R^3)$.

3) Joint Eigenvalue Decomposition-Based Initialization (JEVD): The following initialization works with three compressions of the original tensor $\boldsymbol{\mathcal{Y}}$, including $\boldsymbol{\mathcal{X}}^{(1,2)}$ of size $R\times R\times 2$, $\boldsymbol{\mathcal{X}}^{(1,3)}$ of size $R\times 2\times R$ and $\boldsymbol{\mathcal{X}}^{(2,3)}$ of size $2\times R\times R$. We define matrices $\mathbf{F}_{i,j}$, for $1\le i\ne j\le 3$ as

$$\begin{aligned}
\mathbf{F}_{12} &= \mathbf{X}_1^{(1,2)}\left(\mathbf{X}_2^{(1,2)}\right)^{-1}, \quad \mathbf{F}_{21} = \left(\mathbf{X}_1^{(1,2)}\right)^T\left(\mathbf{X}_2^{(1,2)T}\right)^{-1},\\
\mathbf{F}_{13} &= \mathbf{X}_1^{(1,3)}\left(\mathbf{X}_2^{(1,3)}\right)^{-1}, \quad \mathbf{F}_{31} = \left(\mathbf{X}_1^{(1,3)}\right)^T\left(\mathbf{X}_2^{(1,3)T}\right)^{-1},\\
\mathbf{F}_{23} &= \mathbf{X}_1^{(2,3)}\left(\mathbf{X}_2^{(2,3)}\right)^{-1}, \quad \mathbf{F}_{32} = \left(\mathbf{X}_1^{(2,3)}\right)^T\left(\mathbf{X}_2^{(2,3)T}\right)^{-1},
\end{aligned}$$

where $\mathbf{X}_1^{(n,m)}$ and $\mathbf{X}_2^{(n,m)}$ are two slices of the tensor $\boldsymbol{\mathcal{X}}^{(n,m)}$. As stated in Theorem 1, $\boldsymbol{a}$, $\boldsymbol{b}$ and $\boldsymbol{c}$ can be considered joint eigenvectors of matrices $\mathbf{F}_{i,j}$ and $\mathbf{F}_{j,i}$, that is,

$$\begin{aligned}
\mathbf{F}_{12}\,\boldsymbol{a} &= \boldsymbol{a}\,\lambda_{12}\,, & \mathbf{F}_{13}\,\boldsymbol{a} &= \boldsymbol{a}\,\lambda_{13}\\
\mathbf{F}_{21}\,\boldsymbol{b} &= \boldsymbol{b}\,\lambda_{21}\,, & \mathbf{F}_{23}\,\boldsymbol{b} &= \boldsymbol{b}\,\lambda_{23}\\
\mathbf{F}_{31}\,\boldsymbol{c} &= \boldsymbol{c}\,\lambda_{31}\,, & \mathbf{F}_{32}\,\boldsymbol{c} &= \boldsymbol{c}\,\lambda_{32}
\end{aligned}$$

with $\lambda_{12}=\lambda_{21}$, $\lambda_{13}=\lambda_{31}$ and $\lambda_{23}=\lambda_{31}$, $\boldsymbol{a}^T\boldsymbol{a}=1$, $\boldsymbol{b}^T\boldsymbol{b}=1$ and $\boldsymbol{c}^T\boldsymbol{c}=1$. It follows that we can find unit-length vectors $\boldsymbol{a}$, $\boldsymbol{b}$ and $\boldsymbol{c}$ as solution to the following minimisation problem

$$\begin{aligned}
\min_{\boldsymbol{a},\boldsymbol{b},\boldsymbol{c}} &\|\mathbf{F}_{12}\,\boldsymbol{a}-(\boldsymbol{a}^T\,\mathbf{F}_{12}\,\boldsymbol{a})\,\boldsymbol{a}\|^2+\|\mathbf{F}_{13}\,\boldsymbol{a}-(\boldsymbol{a}^T\,\mathbf{F}_{13}\,\boldsymbol{a})\,\boldsymbol{a}\|^2\\
&+\|\mathbf{F}_{21}\,\boldsymbol{b}-(\boldsymbol{b}^T\,\mathbf{F}_{21}\,\boldsymbol{b})\,\boldsymbol{b}\|^2+\|\mathbf{F}_{23}\,\boldsymbol{b}-(\boldsymbol{b}^T\,\mathbf{F}_{23}\,\boldsymbol{b})\,\boldsymbol{b}\|^2\\
&+\|\mathbf{F}_{31}\,\boldsymbol{c}-(\boldsymbol{c}^T\,\mathbf{F}_{31}\,\boldsymbol{c})\,\boldsymbol{c}\|^2+\|\mathbf{F}_{32}\,\boldsymbol{c}-(\boldsymbol{c}^T\,\mathbf{F}_{32}\,\boldsymbol{c})\,\boldsymbol{c}\|^2\\
&+\|\boldsymbol{a}^T\,\mathbf{F}_{12}\,\boldsymbol{a}-\boldsymbol{b}^T\,\mathbf{F}_{21}\,\boldsymbol{b}\|^2+\|\boldsymbol{a}^T\,\mathbf{F}_{13}\,\boldsymbol{a}-\boldsymbol{c}^T\,\mathbf{F}_{31}\,\boldsymbol{c}\|^2\\
&+\|\boldsymbol{b}^T\,\mathbf{F}_{23}\,\boldsymbol{b}-\boldsymbol{c}^T\,\mathbf{F}_{32}\,\boldsymbol{c}\|^2\,,
\end{aligned}$$

Algorithm 3: Joint EVD-based Initialization

Input: Data tensor $\boldsymbol{\mathcal{Y}}$: $(R\times R\times\ldots\times R)$
Output: $\{\boldsymbol{a}^{(1)},\ldots,\boldsymbol{a}^{(N)}\}$ and $\{\boldsymbol{u}_1,\ldots,\boldsymbol{u}_N\}$
begin
% **Stage 1:** Estimate components $\boldsymbol{a}^{(n)}$----------
Initialize $[\![\{\boldsymbol{a}^{(n)}\}]\!]$
Construct matrices $\mathbf{F}_{n,m}$ for $1\le n\ne m\le N$
repeat
  for $n=1,2,\ldots,N$ **do**
    $[\boldsymbol{a}^{(n)},\lambda_n]=$ eigs$(\mathbf{Q}_n,1,$'SA'$)$ /* $\mathbf{Q}_n$ in (16) */
until *a stopping criterion is met*
% **Stage 2:** Estimate λ of rank-1 tensor $[\![\{\boldsymbol{a}^{(n)}\}]\!]$-----
$\lambda=\frac{1}{N}\sum_n \frac{1}{\boldsymbol{s}_n^T\mathbf{C}_n^{-1}\boldsymbol{a}^{(n)}}$
% **Stage 3:** Estimate $\boldsymbol{w}_n$, ρ_n and $\boldsymbol{u}_n$----------
$\boldsymbol{w}_n=\frac{\mathbf{C}_n^{-1}\boldsymbol{s}_n}{\sqrt{\boldsymbol{s}_n^T\mathbf{C}_n^{-2}\boldsymbol{s}_n}}$, $\rho_n=\sqrt{1-\frac{1}{\lambda^2\,\boldsymbol{s}_n^T\mathbf{C}_n^{-2}\boldsymbol{s}_n}}$ and $\boldsymbol{u}_n=\frac{\boldsymbol{a}^{(n)}-\sqrt{1-\rho_n^2}\,\boldsymbol{w}_n}{\rho_n}$

eigs$(\mathbf{C},1,$'SA'$)$: compute eigenvector associated with the smallest eigenvalue of $\mathbf{C}$.

which can be simplified into

$$\begin{aligned}
\min_{\boldsymbol{a},\boldsymbol{b},\boldsymbol{c}}\ &\boldsymbol{a}^T\mathbf{C}_a\boldsymbol{a}+\boldsymbol{b}^T\mathbf{C}_b\boldsymbol{b}+\boldsymbol{c}^T\mathbf{C}_c\boldsymbol{c}-2\,(\boldsymbol{a}^T\,\mathbf{F}_{12}\,\boldsymbol{a})\,(\boldsymbol{b}^T\,\mathbf{F}_{21}\,\boldsymbol{b})\\
&-2\,(\boldsymbol{a}^T\,\mathbf{F}_{13}\,\boldsymbol{a})\,(\boldsymbol{c}^T\,\mathbf{F}_{31}\,\boldsymbol{c})-2\,(\boldsymbol{b}^T\,\mathbf{F}_{23}\,\boldsymbol{b})\,(\boldsymbol{c}^T\,\mathbf{F}_{32}\,\boldsymbol{c})
\end{aligned}$$

where $\mathbf{C}_a=\mathbf{F}_{12}^T\,\mathbf{F}_{12}+\mathbf{F}_{13}^T\,\mathbf{F}_{13}$, $\mathbf{C}_b=\mathbf{F}_{21}^T\,\mathbf{F}_{21}+\mathbf{F}_{23}^T\,\mathbf{F}_{23}$ and $\mathbf{C}_c=\mathbf{F}_{31}^T\,\mathbf{F}_{31}+\mathbf{F}_{32}^T\,\mathbf{F}_{32}$.

While fixing $\boldsymbol{b}$ and $\boldsymbol{c}$, $\boldsymbol{a}$ is solution to the eigenvalue problem

$$\arg\min_{\boldsymbol{a}}\ \boldsymbol{a}^T\,\mathbf{Q}_a\,\boldsymbol{a}\,, \tag{15}$$

where $\mathbf{Q}_a=\mathbf{C}_a-(\boldsymbol{b}^T\mathbf{F}_{21}\boldsymbol{b})(\mathbf{F}_{12}+\mathbf{F}_{12}^T)-(\boldsymbol{c}^T\mathbf{F}_{31}\boldsymbol{c})(\mathbf{F}_{13}+\mathbf{F}_{13}^T)$. That is $\boldsymbol{a}$ is the eigenvector associated with the smallest eigenvalue of the symmetric matrix $\mathbf{Q}_a$. Similarly, we can iteratively estimate $\boldsymbol{b}$ and $\boldsymbol{c}$ as eigenvectors of the matrices $\mathbf{Q}_b$ and $\mathbf{Q}_c$

$$\begin{aligned}
\mathbf{Q}_b &= \mathbf{C}_b-(\boldsymbol{a}^T\,\mathbf{F}_{12}\,\boldsymbol{a})\,(\mathbf{F}_{21}+\mathbf{F}_{21}^T)-(\boldsymbol{c}^T\,\mathbf{F}_{32}\,\boldsymbol{c})\,(\mathbf{F}_{23}+\mathbf{F}_{23}^T),\\
\mathbf{Q}_c &= \mathbf{C}_c-(\boldsymbol{a}^T\,\mathbf{F}_{13}\,\boldsymbol{a})\,(\mathbf{F}_{31}+\mathbf{F}_{31}^T)-(\boldsymbol{b}^T\,\mathbf{F}_{23}\,\boldsymbol{b})\,(\mathbf{F}_{32}+\mathbf{F}_{32}^T).
\end{aligned}$$

For higher order tensors, $\boldsymbol{a}^{(n)}$ are solved similarly as eigenvectors of the matrices

$$\mathbf{Q}_n=\sum_{m\ne n}\mathbf{F}_{n,m}^T\,\mathbf{F}_{n,m}-(\boldsymbol{a}^{(m)T}\,\mathbf{F}_{m,n}\,\boldsymbol{a}^{(m)})\,(\mathbf{F}_{n,m}+\mathbf{F}_{n,m}^T)\,, \tag{16}$$

where $\mathbf{F}_{n,m}=\mathbf{X}_1^{(n,m)}\left(\mathbf{X}_2^{(n,m)}\right)^{-1}$ and $\mathbf{F}_{m,n}=\left(\mathbf{X}_1^{(n,m)}\right)^T\left(\mathbf{X}_2^{(n,m)T}\right)^{-1}$ for $1\le n<m\le N$.

The algorithm is summarized as Algorithm 3. Components $\boldsymbol{a}^{(n)}$ and matrices $\mathbf{U}^{(n)}$ can be initialized by eigenvectors in Section III-C2. The cost to construct the matrices $\mathbf{F}_{n,m}$ of size $R\times R$ is substantially due to compressions of the tensor $\boldsymbol{\mathcal{Y}}$ to tensors of two slices $\boldsymbol{\mathcal{X}}^{(n,m)}$ for $1\le n<m\le N$, which are of order $\mathcal{O}(N(N-1)/2\,R^N)=\mathcal{O}(N^2R^N)$. For order-3 tensors, we only need to construct three tensors $\boldsymbol{\mathcal{X}}^{(1,2)}$, $\boldsymbol{\mathcal{X}}^{(1,3)}$ and $\boldsymbol{\mathcal{X}}^{(2,3)}$. Moreover, it is worth noting that the matrices $\mathbf{F}_{n,m}$ and

the terms $\sum_{m\neq n} \mathbf{F}_{n,m}^T \mathbf{F}_{n,m}$ are computed once before the iterations in Step 2. Since matrices $\mathbf{Q}_n$ are of size $R \times R$, each iteration to estimate $\boldsymbol{a}^{(n)}$ in Step 3 requires an operation of $\mathcal{O}(R^3)$. Therefore, the total cost of JEVD is at most $\mathcal{O}(N^2 R^N)$.

IV. Deflation of Tensor With Orthogonal Factor Matrices

In this section, we present a special case of the rank-1 deflation for tensor decompositions with orthogonality constraints in one or several factor matrices. In the CP tensor decompositions, the orthogonality constraints can be imposed onto components in order to improve the stability of the decomposition, and avoid diverging components [6], [12]. In decomposition of received signals in a direct sequence code division multiple access (DS-CDMA) system [13], spreading codes can be orthogonal. The constraint has found applications for analyzing EEG signals [14], [15].

For simplicity, assuming that the decomposition seeks one orthonormal factor matrix, which is $\mathbf{A}^{(1)}$, i.e., $\mathbf{A}^{(1)T}\mathbf{A}^{(1)} = \mathbf{I}_R$. From the orthogonal normalization, we have that $[\boldsymbol{a}^{(1)}, \mathbf{U}^{(1)}]$ is an orthonormal matrix. That means $\rho_1 = \boldsymbol{u}_1^T \boldsymbol{a}^{(1)} = 0$, and thus, $\rho = 0$. The weight of the rank-1 tensor λ and the core tensor $\boldsymbol{\mathcal{G}}$ are then given by

$$\lambda = \boldsymbol{\mathcal{Y}} \times \{\boldsymbol{a}^{(n)T}\}, \qquad \boldsymbol{\mathcal{G}} = \boldsymbol{\mathcal{Y}} \times \{\mathbf{U}^{(n)T}\}. \tag{17}$$

A. Estimation of the Orthogonal Components $\boldsymbol{a}^{(1)}$ and $\mathbf{U}^{(1)}$

In order to estimate $\boldsymbol{a}^{(1)}$ and $\mathbf{U}^{(1)}$, the cost function which minimizes the Frobenius norm between the tensor $\boldsymbol{\mathcal{Y}}$ and its approximate

$$D = \frac{1}{2}\|\boldsymbol{\mathcal{Y}} - [\![\lambda; \boldsymbol{a}^{(1)}, \boldsymbol{a}^{(2)}, \ldots, \boldsymbol{a}^{(N)}]\!] - [\![\boldsymbol{\mathcal{G}}; \mathbf{U}^{(1)}, \mathbf{U}^{(2)}, \ldots, \mathbf{U}^{(N)}]\!]\|_F^2$$

can be rewritten as

$$\begin{aligned} \min \quad D &= \frac{1}{2}\left(\|\boldsymbol{\mathcal{Y}}\|_F^2 - \|\boldsymbol{\mathcal{G}}\|_F^2 - \lambda^2\right) \\ &= \frac{1}{2}\left(\|\boldsymbol{\mathcal{Y}}\|_F^2 - \|\mathbf{U}^{(1)}\mathbf{Z}_1\|_F^2 - (\boldsymbol{w}_1^T \boldsymbol{s}_1)^2\right) \\ &= \frac{1}{2}\left(\|\boldsymbol{\mathcal{Y}}\|_F^2 - \|\mathbf{Z}_1\|_F^2 + \boldsymbol{w}_1^T\left(\boldsymbol{\Phi}_1 - \boldsymbol{s}_1\boldsymbol{s}_1^T\right)\boldsymbol{w}_1\right), \end{aligned} \tag{18}$$

where $\boldsymbol{\Phi}_n = \mathbf{Z}_n\mathbf{Z}_n^T$, where $\mathbf{Z}_n = \mathbf{Y}_{(n)}\left(\bigotimes_{k\neq n}\mathbf{U}^{(k)}\right)$. It follows that $\boldsymbol{a}^{(1)} = \boldsymbol{w}_1$ is the eigenvector associated with the smallest eigenvalue of $(\boldsymbol{\Phi}_1 - \boldsymbol{s}_1\boldsymbol{s}_1^T)$, and $\mathbf{U}^{(1)}$ is orthogonal complement to $\boldsymbol{w}_1$.

B. Estimation of Nonorthogonal Components $\boldsymbol{a}^{(n)}$ and $\mathbf{U}^{(n)}$

For other factors $\boldsymbol{a}^{(n)}$ and $\mathbf{U}^{(n)}$ with $n \neq 1$, since $\lambda = \boldsymbol{a}^{(n)}\boldsymbol{s}_n$ the cost function (18) is simplified into

$$\min\ D = \frac{1}{2}\big(\|\boldsymbol{\mathcal{Y}}\|_F^2 - \|\mathbf{Z}_n\|_F^2 + \boldsymbol{w}_n^T\boldsymbol{\Phi}_n\boldsymbol{w}_n - (\xi_n\boldsymbol{w}_n^T\boldsymbol{s}_n + \rho_n\boldsymbol{u}_n^T\boldsymbol{s}_n)^2\big),$$

where $\xi_n = \sqrt{1-\rho_n^2}$. This yields the optimal parameter

$$\rho_n^{opt} = \frac{\boldsymbol{w}_n^T\boldsymbol{s}_n}{\sqrt{(\boldsymbol{w}_n^T\boldsymbol{s}_n)^2 + (\boldsymbol{u}_n^T\boldsymbol{s}_n)^2}}. \tag{19}$$

Algorithm 4: ASU with Orthogonality in Mode-1

Input: Data tensor $\boldsymbol{\mathcal{Y}}$: $(R \times R \times \cdots \times R)$, rank R

Output: A rank-1 tensor $[\![\lambda; \{\boldsymbol{a}^{(n)}\}]\!]$ and multilinear rank-$(R-1)$ tensor $[\![\boldsymbol{\mathcal{G}}; \{\mathbf{U}^{(n)}\}]\!]$

begin

1 Initialize components $\boldsymbol{a}^{(n)}$ and $\mathbf{U}^{(n)}$

repeat

2 $[\boldsymbol{a}_1, \sigma] = \texttt{eigs}\left(\boldsymbol{\Phi}_1 - \boldsymbol{s}_1\boldsymbol{s}_1^T, 1, \text{`SA'}\right)$

for $n = 2, 3, \ldots, N$ **do**

3 $[\boldsymbol{w}_n, \sigma] = \texttt{eigs}\left(\boldsymbol{\Phi}_n, 1, \text{`SA'}\right)$

4 $\boldsymbol{u}_n = \frac{\boldsymbol{s}_n - \boldsymbol{w}_n(\boldsymbol{w}_n^T\boldsymbol{s}_n)}{\sqrt{\boldsymbol{s}_n^T\boldsymbol{s}_n - (\boldsymbol{w}_n^T\boldsymbol{s}_n)^2}}$, $\rho_n = \frac{\boldsymbol{w}_n^T\boldsymbol{s}_n}{\sqrt{(\boldsymbol{w}_n^T\boldsymbol{s}_n)^2 + (\boldsymbol{u}_n^T\boldsymbol{s}_n)^2}}$

5 $\boldsymbol{a}^{(n)} \leftarrow \boldsymbol{w}_n\sqrt{1-\rho_n^2} + \boldsymbol{u}_n\rho_n$

until *a stopping criterion is met*

6 Select $\mathbf{U}^{(1)}$ as an orthogonal complement to $\boldsymbol{a}_1$

for $n = 2, \ldots, N$ **do**

7 Select $\mathbf{U}_{2:R-1}^{(n)}$ as an orthogonal complement to $[\boldsymbol{w}_n, \boldsymbol{u}_n]$

8 $\lambda = \boldsymbol{\mathcal{Y}} \times \{\boldsymbol{a}^{(n)T}\}$, $\boldsymbol{\mathcal{G}} = \boldsymbol{\mathcal{Y}} \times \left\{\mathbf{U}^{(n)^T}\right\}$

The results can also be seen from the ASU algorithm after some manipulations. Once again, we come to that $\boldsymbol{w}_n$ and $\boldsymbol{u}_n$ are solution to the following optimization

$$\begin{aligned} \min \quad & \frac{1}{2}\left(\boldsymbol{w}_n^T(\boldsymbol{\Phi}_n - \boldsymbol{s}_n\boldsymbol{s}_n^T)\boldsymbol{w}_n - (\boldsymbol{u}_n^T\boldsymbol{s}_n)^2\right) \\ \text{subject to} \quad & [\boldsymbol{w}_n, \boldsymbol{u}_n]^T[\boldsymbol{w}_n, \boldsymbol{u}_n] = \mathbf{I}_2. \end{aligned} \tag{20}$$

Closed-forms for $\boldsymbol{u}_n$ can be simplified

$$\boldsymbol{u}_n = \frac{\boldsymbol{s}_n - \boldsymbol{w}_n(\boldsymbol{w}_n^T\boldsymbol{s}_n)}{\sqrt{\boldsymbol{s}_n^T\boldsymbol{s}_n - (\boldsymbol{w}_n^T\boldsymbol{s}_n)^2}} \tag{21}$$

and $\boldsymbol{w}_n$ is the eigenvector of the smallest eigenvalue of the matrix $\boldsymbol{\Phi}_n = \mathbf{Z}_n\mathbf{Z}_n^T$.

The whole ASU algorithm for decomposition with one orthogonal factor matrix is summarized in Algorithm 4. Similarly to the ASU algorithm [1], the matrices $\mathbf{U}^{(n)}$ need not be constructed except the first components $\boldsymbol{u}_n$ for $n = 2, \ldots, N$.

V. Cramér-Rao Lower Bound for Tensor Deflation

In this section, we derive the Cramér-Rao Lower Bound (CRB) for the rank-1 plus multilinear rank-$(R-1, \ldots, R-1)$ block tensor decomposition. We will assess the loss in accuracy of the tensor deflation by comparing its CRB with the similar bound obtained for the ordinary CP decomposition.

We consider the data model

$$\boldsymbol{\mathcal{Y}} = \hat{\boldsymbol{\mathcal{Y}}} + \boldsymbol{\mathcal{E}} \tag{22}$$

where the deterministic part is

$$\hat{\boldsymbol{\mathcal{Y}}} = \lambda[\![\{\boldsymbol{a}^{(n)}\}]\!] + [\![\boldsymbol{\mathcal{G}}; \mathbf{U}^{(1)}, \mathbf{U}^{(2)}, \ldots, \mathbf{U}^{(N)}]\!] \tag{23}$$

and $\boldsymbol{\mathcal{E}}$ represents an error tensor which has i.i.d. normally distributed elements with zero mean and variance σ^2. In the CP decomposition, $\boldsymbol{\mathcal{G}}$ is spatially diagonal and $\mathbf{U}^{(n)}$ are arbitrary; in the tensor deflation $\boldsymbol{\mathcal{G}}$ does not have any structure but $\mathbf{U}^{(n)}$ are orthogonal.

For now, the number of the estimated parameters is too high and the decomposition is not unique. Therefore, as in deriving the ASU algorithm, we perform reparameterization $\boldsymbol{a}^{(n)} = \boldsymbol{w}_n\,\xi_n + \boldsymbol{u}_n\,\rho_n$, where $\boldsymbol{\rho}_n = [\xi_n, \rho_n]^T$ are unit-length vectors, and $[\boldsymbol{w}_n, \mathbf{U}_n]$ are orthonormal matrices for $n = 1, \ldots, N$. The tensor $\boldsymbol{\mathcal{G}}$ will be replaced by its maximum likelihood estimate

$$\widehat{\boldsymbol{\mathcal{G}}} = \boldsymbol{\mathcal{Y}} \times \{\mathbf{U}^{(n)^T}\} - \lambda\,\rho\,\boldsymbol{\mathcal{E}}_1\,. \tag{24}$$

We follow the approach of [16] to derive the CRB from the so-called concentrated log-likelihood function. Inserting (24) into (23) we get

$$\hat{\boldsymbol{\mathcal{Y}}} = \lambda[\![\{\boldsymbol{a}^{(n)}\}]\!] - \lambda\rho[\![\{\boldsymbol{u}_n\}]\!] + \boldsymbol{\mathcal{Y}} \times \{\mathbf{I}_R - \boldsymbol{w}_n\boldsymbol{w}_n^T\}\,. \tag{25}$$

Considering that $\boldsymbol{a}^{(n)} = \boldsymbol{w}_n\,\xi_n + \boldsymbol{u}_n\,\rho_n$ and $\rho = \rho_1\rho_2\ldots,\rho_N$, the total number of parameters is reduced to $2N(R+1)+1$. The parameters are listed as elements of a vector parameter

$$\boldsymbol{\alpha} = [\boldsymbol{w}_1^T, \boldsymbol{u}_1^T, \ldots, \boldsymbol{w}_N^T, \boldsymbol{u}_N^T, \boldsymbol{\rho}_1^T, \ldots, \boldsymbol{\rho}_N^T, \lambda]^T\,. \tag{26}$$

The parameter $\boldsymbol{\alpha}$ obeys the constraint

$$\boldsymbol{f}(\boldsymbol{\alpha}) = \begin{bmatrix} (\boldsymbol{w}_1^T\,\boldsymbol{w}_1 - 1)/2 \\ (\boldsymbol{u}_1^T\,\boldsymbol{u}_1 - 1)/2 \\ (\boldsymbol{w}_1^T\,\boldsymbol{u}_1)/\sqrt{2} \\ \vdots \\ (\boldsymbol{w}_N^T\,\boldsymbol{w}_N - 1)/2 \\ (\boldsymbol{u}_N^T\,\boldsymbol{u}_N - 1)/2 \\ (\boldsymbol{w}_N^T\,\boldsymbol{u}_N)/\sqrt{2} \\ (\xi_1^2 + \rho_1^2 - 1)/2 \\ \vdots \\ (\xi_N^2 + \rho_N^2 - 1)/2 \end{bmatrix} = \mathbf{0}. \tag{27}$$

The constrained Cramér-Rao bound on $\boldsymbol{\alpha}$ [17], [18] is given by

$$\mathrm{CRB}(\tilde{\boldsymbol{\alpha}}) = \mathbf{F}_{\boldsymbol{f}}(\boldsymbol{\alpha})\,\left(\mathbf{F}_{\boldsymbol{f}}(\boldsymbol{\alpha})^T\,\mathbf{I}(\boldsymbol{\alpha})\mathbf{F}_{\boldsymbol{f}}(\boldsymbol{\alpha})\right)^{-1}\,\mathbf{F}_{\boldsymbol{f}}(\tilde{\boldsymbol{\alpha}})^T \tag{28}$$

where $\mathbf{F}_{\boldsymbol{f}}(\boldsymbol{\alpha})$ is an orthonormal complement matrix whose vectors span the null space of the Jacobian matrix of the constraint functions with respect to $\boldsymbol{\alpha}$, and $\mathbf{I}(\boldsymbol{\alpha})$ is the Fisher information matrix

$$\mathbf{I}(\boldsymbol{\alpha}) = \frac{1}{\sigma^2}\mathbf{H} \tag{29}$$

where $\mathbf{H}$ is the Hessian matrix of the error function $\frac{1}{2}\|\hat{\boldsymbol{\mathcal{Y}}}(\boldsymbol{\alpha}) - \boldsymbol{\mathcal{Y}}\|_F^2$ with respect to $\boldsymbol{\alpha}$. Expressions of the matrix elements of $\mathbf{H}$ are given in Appendix D, whereas derivation of the orthogonal matrix $\mathbf{F}_{\boldsymbol{f}}(\boldsymbol{\alpha})$ appearing in (28) is summarized in Appendix E.

Now, CRB on $\boldsymbol{a}^{(n)} = \boldsymbol{a}^{(n)}(\boldsymbol{\alpha})$ and the Cramér-Rao Induced Bound (CRIB) on squared angular errors (SAE)[1] of $\boldsymbol{a}^{(n)}$ can be computed as [7]

$$\mathrm{CRB}(\{\boldsymbol{a}^{(n)}\}) = \mathbf{J}_a(\boldsymbol{\alpha})\,\mathrm{CRB}(\boldsymbol{\alpha})\,\mathbf{J}_a(\boldsymbol{\alpha})^T\,, \tag{30}$$

$$\mathrm{CRIB}(\boldsymbol{a}^{(n)}) = \mathrm{tr}(\mathrm{CRB}(\boldsymbol{a}^{(n)})) - \boldsymbol{a}^{(n)T}\,\mathrm{CRB}(\boldsymbol{a}^{(n)})\,\boldsymbol{a}^{(n)}\,, \tag{31}$$

[1]Squared Angular Error between two vectors $\boldsymbol{a}$ and $\hat{\boldsymbol{a}}$ is defined as $\mathrm{SAE}(\boldsymbol{a}, \hat{\boldsymbol{a}}) \triangleq \arccos\left(\frac{\boldsymbol{a}^T\hat{\boldsymbol{a}}}{\|\boldsymbol{a}\|_2\|\hat{\boldsymbol{a}}\|_2}\right)^2$, and $-10\log_{10}(\mathrm{SAE})$ in dB.

where $\mathbf{J}_a(\boldsymbol{\alpha})$ represents the Jacobian matrix of the function $\boldsymbol{a}^{(n)} = \boldsymbol{a}^{(n)}(\boldsymbol{\alpha}) = [\boldsymbol{w}_n, \boldsymbol{u}_n]\,\boldsymbol{\rho}_n$

$$\mathbf{J}_a(\boldsymbol{\alpha}) = \begin{bmatrix} \xi_1\mathbf{I}_R & \rho_1\mathbf{I}_R & & & \boldsymbol{w}_1 & \boldsymbol{u}_1 & & \\ & \ddots & \ddots & & & \ddots & \ddots & \\ & & \xi_N\mathbf{I}_R & \rho_N\mathbf{I}_R & & & \boldsymbol{w}_N & \boldsymbol{u}_N \;\; 0 \end{bmatrix}. \tag{32}$$

The CRBs in (28), (30) and (31) derived in this paper not only serve for assessing performance of the block tensor decomposition, but also provide error bound for extraction of components of a rank-1 tensor in the CP decomposition. For the case when components $\boldsymbol{a}^{(n)}$ are orthogonal to $\mathbf{U}^{(n)}$, i.e., $\rho_n = 0$, the CRB should be derived with simplified constraint functions. Comparison between the CRBs derived for tensor deflation and CPD using the method developed in [7] shows that there is no significant loss in accuracy using our tensor deflation approach.

VI. Simulations

Example 1 Loss of Accuracy in Tensor Deflation Compared With CPD: This example compares CRB on extracting a rank-1 tensor in the tensor deflation with the CRB of this rank-1 tensor in CPD. The loss of accuracy in estimation of components will be shown as the difference between two similar CRBs for tensor deflation and the ordinary CPD.

The tensors in this analysis are of size $R \times R \times R$ and of rank R where $R = 10, 20$. The weight coefficients λ_r are set to 1, whereas collinearity degrees between components $\boldsymbol{a}_r^{(n)}$ and $\boldsymbol{a}_s^{(n)}$ for all $r \neq s$ are identical to a specific value c, which is varied in the range $[0, 1)$, $\boldsymbol{a}_r^{(n)T}\boldsymbol{a}_s^{(n)} = c$ and $\boldsymbol{a}_r^{(n)T}\boldsymbol{a}_r^{(n)} = 1$ for all n. Appendix F presents a method to generate the factor matrices $\mathbf{A}^{(n)}$.

The parameter ρ_n between a component $\boldsymbol{a}_r^{(n)}$ and the rest $(R-1)$ columns of $\mathbf{A}^{(n)}$ is a function of c and R defined as (see proof in Appendix G)

$$\rho_n = c\sqrt{\frac{R-1}{(R-2)c+1}}\,. \tag{33}$$

In Fig. 2(a), we compare the Cramér-Rao Induced bound (CRIB) [7] on the squared angular error (SAE) in dB, i.e., $-10\log_{10}\mathrm{SAE}$, of components $\boldsymbol{a}^{(n)}$, and the CRIB in (31). The CRBs are shown when the Gaussian noise is at $\mathrm{SNR} = 30$ dB.

The two CRB curves are almost overlapped for all c, and both imply that decomposition becomes difficult, when ρ_n is close to 1, i.e., $\boldsymbol{a}^{(n)}$ is highly correlated to the other components, and easier when ρ_n approaches 0, i.e., components are mutually orthogonal.

The loss of accuracy in estimation of components $\boldsymbol{a}_r^{(n)}$ using the tensor deflation is shown in Fig. 2(b) for various ranks R. The loss is small when $c \leq 0.5$, but it tends to be high when c is closed to 1, and the rank R is small. Nevertheless, the curves reveal that the differences between two similar CRIBs were less than 0.1 dB and insignificant. Note that the loss is independent of the noise level.

Similar curves which compare the squared angular error of the ASU algorithm and Cramér-Rao bound on this error are shown in Part I [1].

Example 2 Loss of Accuracy in Tensor Deflation When Components are of Different Magnitudes: This example shows another comparison between CRIBs for tensor deflation and CPD

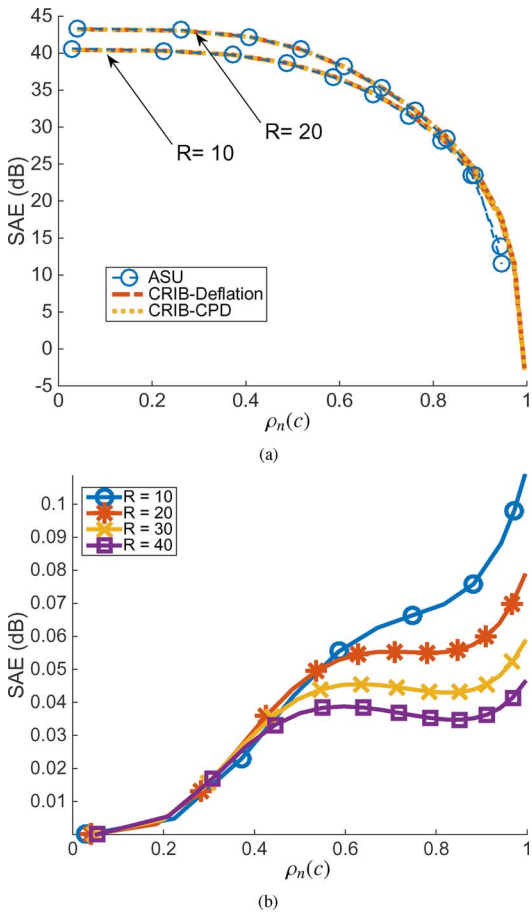

Fig. 2. Comparison of CRBs for CPD and tensor deflation on squared angular errors of components $\boldsymbol{a}^{(n)}$. (a) CRBs are illustrated as function of collinearity degree $c = \boldsymbol{a}_r^{(n)T}\boldsymbol{a}_s^{(n)}$ varying in the range $[0,\ 1)$. Tensor size $R = 10$ and 20. (b) Loss of the squared angular error in (dB) as the difference between CRIB for tensor deflation in (31) and the similar bound for CPD in [7]. (a) SAE vs c. (b) Loss of SAE $= -10\log 10(\text{CRIB-CPD [7]}/\text{CRIB in (31)})$.

[7]. Similarly to the previous example, components $\boldsymbol{a}_r^{(n)}$ have identical collinearity degree $c = 0.01$ and 0.6. However, their magnitudes λ_r are different and take values as $\lambda_r = r$ for $r = 1, \ldots, R$.

The two CRIBs for CPD and tensor deflation are shown in Fig. 3 when SNR $= 30$ dB. Again the bounds are almost identical. Components of rank-1 tensors with large λ_r are easier to extract than ones with smaller weights. The loss of squared angular error using the tensor deflation does not exceed 0.07 dB.

Example 3 Comparison of the Initialization Algorithms: This example aims to verify efficiency of various initialization methods, including the random initialization, the SVD-based method with seeking parameters ρ_n and preselection in Algorithm 2, the tensor diagonalization tool (TEDIA) [5], and the JEVD method in Algorithm 3. For the random initialization, for each decomposition, we randomly generate 10 sets of initial points, and select the one achieving the lowest approximation error after passing through the ASU algorithm with a small number of iterations (say 10). Simulations were run on a computer consisted of Intel Xeon 2 processors clocked at 3.33 GHz, 64 GB of main memory.

The simulations are similar to those in Example 1 in Part I [1]. The tensors $\boldsymbol{\mathcal{Y}}$ of size $R \times R \times R$ were randomly generated,

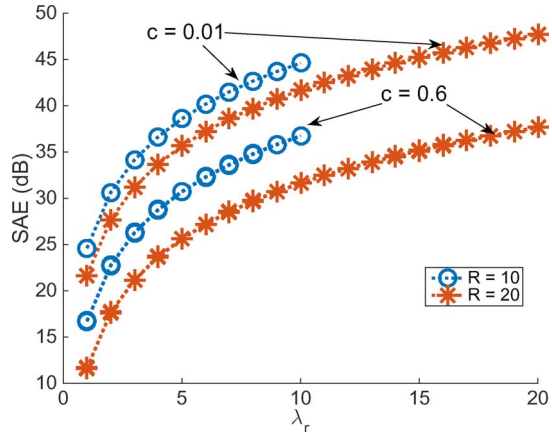

Fig. 3. Comparison of CRBs for CPD and tensor deflation on squared angular errors of components $\boldsymbol{a}^{(n)}$ whose associated weights are $\lambda_r = 1, 2, \ldots, R$.

and corrupted by additive Gaussian noise $\boldsymbol{\mathcal{E}}$ at specific signal-to-noise ratios SNR (dB)

$$\boldsymbol{\mathcal{Y}} = [\![\boldsymbol{a}^{(1)}, \boldsymbol{a}^{(2)}, \boldsymbol{a}^{(3)}]\!] + [\![\boldsymbol{\mathcal{G}}\,; \mathbf{U}^{(1)}, \mathbf{U}^{(2)}, \mathbf{U}^{(3)}]\!] + \sigma\boldsymbol{\mathcal{E}}, \quad (34)$$

where $\mathbf{U}^{(n)}$ are matrices of size $R \times (R-1)$ with orthonormal columns, $\boldsymbol{\mathcal{E}}$ is a zero-mean, unit-variance normal measurement error, and σ^2 denotes the noise variance. The tensors were normalized to have $\|\boldsymbol{\mathcal{G}}\|_F = 1$.

For this example, we used the ASU algorithm [1]. Comparison of performance of algorithms is shown in Part I [1]. The ASU algorithm ran until differences between consecutive approximation errors were small enough, $|\varepsilon_k - \varepsilon_{k+1}| < 10^{-6}\,\varepsilon_k$ where $\varepsilon = \|\boldsymbol{\mathcal{Y}} - \hat{\boldsymbol{\mathcal{Y}}}\|_F^2$, or when the number of iterations exceeded 1000. Performances were assessed through SAE in estimation of components $\boldsymbol{a}^{(n)}$. There were 100 independent runs for each setting of rank and SNR, $R = 10,\ 20,\ 30$ and SNR $= 10,\ 20$ and 30 dB.

Fig. 4 illustrates box-plots of the squared angular errors in dB of the estimated components $\boldsymbol{a}^{(n)}$. Each box represents the median (central mark), 25th and 75th percentiles, and outliers. The average Cramér-Rao induced bounds on the angular errors over independent runs are shown as horizontal lines in Fig. 4.

The results indicate that when purely initialized by multiple random points, ASU failed in most test cases even for low noise levels, e.g., SNR $= 30$ dB. Similar observation was also reported for the block component decompositions using random initialization in [4]. Performances of algorithms were much improved when the components were initialized using SVD-based algorithm, JEVD and TEDIA. Except for the case using the random method, the SAEs achieved the average Cramér-Rao bound on the angular error.

Regarding running times, computing leading singular vectors was very fast, which took less than 0.02 seconds. In companion with the estimation of parameters ρ_n and pre-selection using ASU, the SVD-based initialization method in Algorithm 2 on average took 0.22, 0.33 and 0.67 seconds when $R = 10, 20$ and 30, respectively. The JEVD method was slightly faster. For example, this initialization took an average 0.53 seconds when $R = 30$. A full graphical comparison of running times between initialization methods can be seen in Fig. 5. The random initialization was always simple and fast, but did not generate good

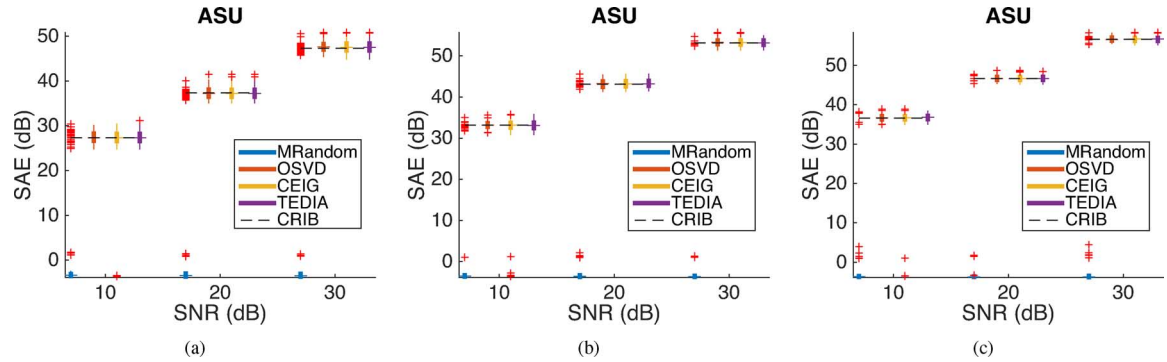

Fig. 4. Box-plots of the squared angular error achieved by algorithms using four different initialization methods: 1 – multiple random initial points, 2- leading singular vectors for $[\boldsymbol{w}_n, \boldsymbol{u}_n]$ plus estimation of parameters ρ_n, 3-JEVD, 4- TEDIA. Tensors are of size $R \times R \times R$ where $R = 10$, 20 and 30. CRIB represents the Cramér-Rao induced bounds on the squared angular errors (SAE), which were averaged over 100 independent runs. Each box represents the median (central mark), 25th and 75th percentiles, whereas the outliers are marked with red crosses "+". (a) $R = 10$. (b) $R = 20$. (c) $R = 30$.

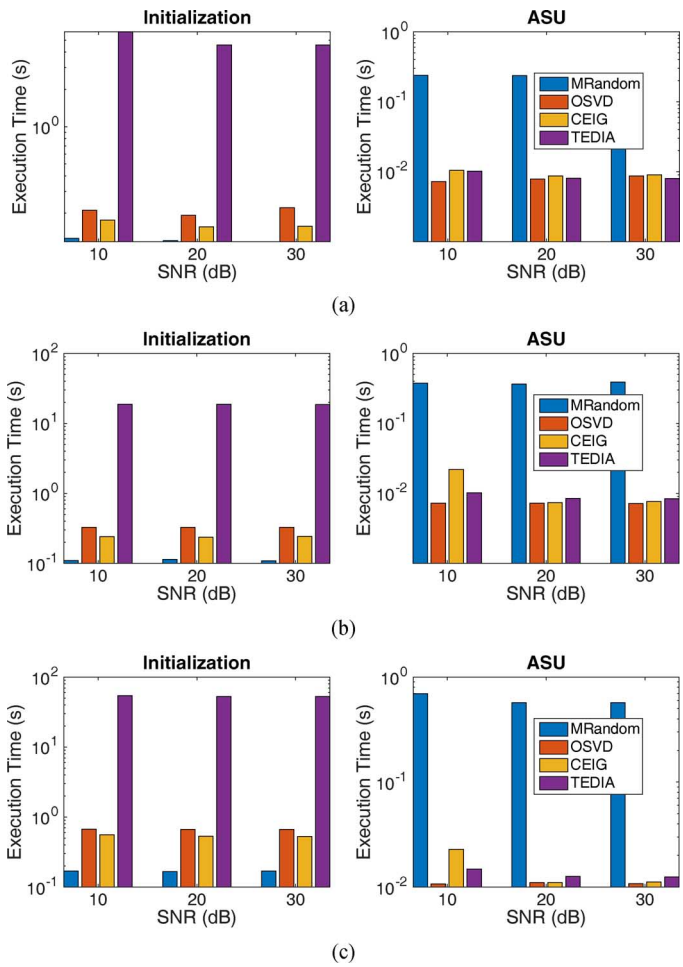

Fig. 5. Comparison of execution times of the ASU algorithm using different initialization methods for Example 3. Initialization methods are numbered as: 1- multiple random initial points, 2- leading singular vectors for $[\boldsymbol{w}_n, \boldsymbol{u}_n]$ plus estimation of parameters ρ_n, 3-JEVD, 4- TEDIA. (a) $R = 10$. (b) $R = 20$. (c) $R = 30$.

initial points. Using this initialization, ASU as well as other algorithms often did not converge to the desired solution with in the prescribed maximum number of iterations (1000). However, this algorithm converged quickly after approximately 0.02 seconds with other initialization methods.

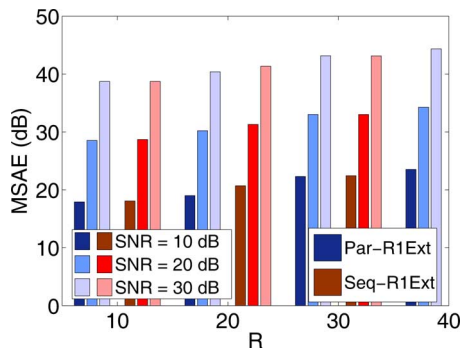

Fig. 6. Performance of the parallel rank-1 extraction (Par-R1Ext) and sequential rank-1 extraction (Seq-R1Ext) with orthogonality constrains in one mode in Example 4. Results for Par-R1Ext are shown for $R = 10,\ 20,\ 30,$ and 40, respectively, whereas results for Seq-R1Ext for $R = 10,\ 20,$ and 30, respectively. The color intensity indicates the SNR level.

1) Example 4 Rank-One Extraction With Orthogonality Constrain in One Mode: In this example, we verify the algorithm in Section IV for deflation of tensors whose one factor matrix is column-wise orthogonal. Three way tensors of size $R \times R \times R$ were generated from random factor matrices $\mathbf{A}^{(n)}$ of size $R \times R$, in which $\mathbf{A}^{(1)}$ was an orthonormal matrix. The tensors were then corrupted with additive Gaussian noise of signal-to-noise ratio SNR $= -10 \log_{10} \frac{\|\mathcal{Y}\|_F^2}{\sigma^2 \prod I_n}$, where σ^2 denotes the noise variance, and $\|\mathcal{Y}\|_F$ is the Frobenius norm of $\mathcal{Y}$. The decompositions were executed over 100 independent runs for each tensor rank $R = 10$, 20, 30, 40 and each noise level SNR $=$ 10, 20 and 30 dB. Parameters of the deflation were initialized using the Direct Trilinear Decomposition (DTLD) [10] without preselection. That is, we did not choose the best rank-1 tensor among R rank-1 tensors generated by DTLD, which yields the smallest approximation error after a small number of iterations. Instead, each rank-1 tensor was assigned to the first block. Thereby, the deflation was proceeded with R different initial values. In Fig. 6, the method is labeled by "Par-R1Ext", while Seq-R1Ext denotes the sequential rank-1 deflation. The mean SAEs (in dB) $= (-10 \log_{10}(SAE))$ on estimation of components were compared in Fig. 6. For most cases, the MSAEs (dB) achieved with Par-R1Ext were only slightly worse than those by Seq-R1Ext. The results indicate that we can accurately retrieve all components of the tensors $\mathcal{Y}$ using different initial points.

VII. Conclusions

We have proposed several initialization methods for the rank-1 tensor deflation, and verified their efficiency and performance. The SVD-based initialization method works efficiently when accompanied by the preselection. Although the computational cost of this procedure is $\mathcal{O}(R^{N+1})$, this initialization is still fast because the preselection is performed by running the ASU within a small number of iterations (e.g., 10). The JEVD initialization works well, and has relatively low complexity of order $\mathcal{O}(N^2R^N)$ which is dominated by the tensor compressions, while the core implementation requires a cost only $\mathcal{O}(R^3)$. TEDIA is an alternative to the SVD-based and JEVD initializations.

Together with good initialization algorithms, we have shown that there is no significant loss of accuracy in estimating rank-tensors using our tensor deflation methods compared with the ordinary CP decomposition. The loss in squared angular error was at most 0.1 dB for difficult scenarios when components were highly collinear despite the noise ratio.

Finally, the initialization methods, CRLB as well as the ASU algorithm for tensor deflation are implemented in the Matlab package TENSORBOX which is available online at: http://www.bsp.brain.riken.jp/ phan/tensorbox.php.

Appendix A
Proof of Lemma 3

In order to derive expression for λ in Lemma 1, we introduce the following results.

Lemma 3: Given a tensor $\boldsymbol{\mathcal{Y}}$ which has an exact CPD or rank-1 plus multilinear rank-$(R-1,\ldots,R-1)$ BTD in (1). If a rank-one tensor $[\![\boldsymbol{a}^{(1)},\boldsymbol{a}^{(2)},\ldots,\boldsymbol{a}^{(n)}]\!]$ of $\boldsymbol{\mathcal{Y}}$ is uniquely identified then for arbitrary orthogonal matrices $\mathbf{F}_n$ that span the row space of $\mathbf{Y}_{(n)}$

$$\left(\bigotimes_{k\neq N}\boldsymbol{a}^{(k)}\right)^T \mathbf{F}_n\,\mathbf{F}_n^T\left(\bigotimes_{k\neq N}\boldsymbol{a}^{(k)}\right)=1, \tag{35}$$

where $\bigotimes_{k\neq N}\boldsymbol{a}^{(k)}=\boldsymbol{a}^{(N)}\otimes\cdots\otimes\boldsymbol{a}^{(n+1)}\otimes\boldsymbol{a}^{(n-1)}\otimes\cdots\otimes\boldsymbol{a}^{(1)}$.

Proof: For simplicity, the proof is given for order-3 tensors in block term decomposition. Mode-1 matricization for the exact decomposition for order-3 tensors in (1) shows that

$$\mathbf{Y}_{(1)}=[\boldsymbol{a}^{(1)},\,\mathbf{U}^{(1)}]\begin{bmatrix}\lambda\left(\boldsymbol{a}^{(3)}\otimes\boldsymbol{a}^{(2)}\right)^T\\ \mathbf{G}_{(1)}\left(\mathbf{U}^{(3)}\otimes\mathbf{U}^{(2)}\right)^T\end{bmatrix}, \tag{36}$$

where $[\boldsymbol{a}^{(1)},\mathbf{U}^{(1)}]$ is a full-rank matrix. Since we have assumed that $\boldsymbol{a}^{(1)}$, $\boldsymbol{a}^{(2)}$ and $\boldsymbol{a}^{(3)}$ are uniquely identified, and $\mathbf{F}_1$ spans the row space of $\mathbf{Y}_{(1)}$, there exists a vector $\boldsymbol{v}$ such that

$$\mathbf{F}_1\,\boldsymbol{v}=\boldsymbol{a}^{(3)}\otimes\boldsymbol{a}^{(2)}\,. \tag{37}$$

Since ${\boldsymbol{a}^{(2)}}^T\boldsymbol{a}^{(2)}={\boldsymbol{a}^{(3)}}^T\boldsymbol{a}^{(3)}=1$, we have $\boldsymbol{v}=\mathbf{F}_1^T\,(\boldsymbol{a}^{(3)}\otimes\boldsymbol{a}^{(2)})$ and $\boldsymbol{v}^T\boldsymbol{v}=1$. Thereby,

$$\left(\boldsymbol{a}^{(3)}\otimes\boldsymbol{a}^{(2)}\right)^T\mathbf{F}_1\,\mathbf{F}_1^T\left(\boldsymbol{a}^{(3)}\otimes\boldsymbol{a}^{(2)}\right)=1\,. \tag{38}$$

■

Corollary 1: Assuming that $\mathbf{Y}_{(n)}^T$ has a full column rank, then

$$\boldsymbol{s}_n^T\mathbf{C}_n^{-1}\boldsymbol{s}_n=1\,. \tag{39}$$

Proof: Since $\mathbf{Y}_{(n)}^T\left(\mathbf{Y}_{(n)}\,\mathbf{Y}_{(n)}^T\right)^{-1}\mathbf{Y}_{(n)}$ is the projection matrix onto the subspace spanned by the rows of $\mathbf{Y}_{(n)}$, we can replace $\mathbf{F}_n\mathbf{F}_n^T$ in (35) by $\mathbf{Y}_{(n)}^T\left(\mathbf{Y}_{(n)}\,\mathbf{Y}_{(n)}^T\right)^{-1}\mathbf{Y}_{(n)}$, and rewrite the expression using the new notion $\boldsymbol{s}_n$ to obtain (39). ■

Proof of Lemma 1: When the rank-1 tensor $[\![\{\boldsymbol{a}^{(n)}\}]\!]$ is given, $\mathbf{U}^{(n)}$ are orthonormal basis of column spaces of mode-n matricizations of the tensor $\boldsymbol{\mathcal{Y}}-[\![\lambda;\{\boldsymbol{a}^{(n)}\}]\!]$ or column spaces of matrices

$$\begin{aligned}\boldsymbol{\Psi}_n=&\left(\mathbf{Y}_{(n)}-\lambda\boldsymbol{a}^{(n)}\left(\bigotimes_{k\neq n}\boldsymbol{a}^{(k)}\right)^T\right)\\&\cdot\left(\mathbf{Y}_{(n)}-\lambda\boldsymbol{a}^{(n)}\left(\bigotimes_{k\neq n}\boldsymbol{a}^{(k)}\right)^T\right)^T\\=&\,\mathbf{Y}_{(n)}\mathbf{Y}_{(n)}^T+\mathbf{T}_n\begin{bmatrix}\lambda^2&-\lambda\\-\lambda&0\end{bmatrix}\mathbf{T}_n^T,\end{aligned} \tag{40}$$

with $\mathbf{T}_n=\left[\boldsymbol{a}^{(n)},\boldsymbol{s}_n\right]$. For the exact case, λ is selected such that ranks of the matrices $\left(\mathbf{Y}_{(n)}-\lambda\boldsymbol{a}^{(n)}\left(\bigotimes_{k\neq n}\boldsymbol{a}^{(k)}\right)^T\right)$ or $\boldsymbol{\Psi}_n$ are reduced by 1. This implies that λ is a solution to equations

$$\det(\boldsymbol{\Psi}_n)=0\,,\quad\text{for } n=1,\ldots,N,$$

where $\det(\boldsymbol{\Psi}_n)$ are quadratic polynomials given by

$$\begin{aligned}\det(\boldsymbol{\Psi}_n)&=\beta_n\,\det\left(\mathbf{I}_2+\mathbf{K}_n\begin{bmatrix}\lambda^2&-\lambda\\-\lambda&0\end{bmatrix}\right)\\&=\beta_n\,\left(\alpha_{n2}\,\lambda^2+\alpha_{n1}\,\lambda+1\right),\end{aligned} \tag{41}$$

where $\mathbf{K}_n=\mathbf{T}_n^T\left(\mathbf{Y}_{(n)}\mathbf{Y}_{(n)}^T\right)^{-1}\mathbf{T}_n$ are matrices of size 2×2, $\beta_n=\det(\mathbf{Y}_{(n)}\mathbf{Y}_{(n)}^T)$, $\alpha_{n2}=\mathbf{K}_n(1,1)-\det(\mathbf{K}_n)$ and $\alpha_{n1}=-2\,\mathbf{K}_n(1,2)$.

According to (39), we have

$$\mathbf{K}_n(2,2)=\boldsymbol{s}_n^T\,\mathbf{C}_n^{-1}\,\boldsymbol{s}_n=1.$$

Thereby, $\alpha_{n2}=\mathbf{K}_n(1,2)^2$, or

$$\det(\boldsymbol{\Psi}_n)=\beta_n\,(\mathbf{K}_n(1,2)\,\lambda-1)^2, \tag{42}$$

giving us closed-form expression in (5). ■

Appendix B
Proof of Lemma 2

Proof: Multiplying along mode-1 both sides of $\boldsymbol{\mathcal{Y}}=[\![\lambda;\{\boldsymbol{a}^{(n)}\}]\!]+[\![\boldsymbol{\mathcal{G}};\{\mathbf{U}^{(n)}\}]\!]$ with

$$\widetilde{\boldsymbol{w}}_1=\frac{\boldsymbol{w}_1}{\sqrt{1-\rho_1^2}}=\frac{\boldsymbol{a}^{(1)}-\rho_1\boldsymbol{u}_1}{1-\rho_1^2}\,, \tag{43}$$

we get

$$\boldsymbol{\mathcal{Y}}\times_1\widetilde{\boldsymbol{w}}_1^T=\lambda\,[\![\boldsymbol{a}^{(2)},\ldots,\boldsymbol{a}^{(N)}]\!]\,, \tag{44}$$

since $\widetilde{\boldsymbol{w}}_1^T\boldsymbol{a}^{(1)}=1$. Consider the Frobenius norm of the difference between the two sides of (44) which achieves zero for the exact decomposition

$$\begin{aligned}D(\widetilde{\boldsymbol{w}}_1)&=\|\boldsymbol{\mathcal{Y}}\times_1\widetilde{\boldsymbol{w}}_1^T-[\![\boldsymbol{a}^{(2)},\ldots,\boldsymbol{a}^{(N)}]\!]\|_F^2\\&=\widetilde{\boldsymbol{w}}_1^T\mathbf{C}_1\widetilde{\boldsymbol{w}}_1+\lambda^2-2\lambda\,\widetilde{\boldsymbol{w}}_1^T\boldsymbol{s}_1.\end{aligned}$$

It is straightforward to see that $\lambda \mathbf{C}_1^{-1}\boldsymbol{s}_1$ is root of the above function since $\lambda\,\boldsymbol{a}^{(1)T}\,\mathbf{C}_1^{-1}\,\boldsymbol{s}_1 = 1$ and

$$\begin{aligned} D(\lambda \mathbf{C}_1^{-1}\,\boldsymbol{s}_1) &= \lambda^2\,\boldsymbol{s}_1^T\mathbf{C}_1^{-1}\mathbf{C}_1\,\mathbf{C}_1^{-1}\boldsymbol{s}_1 + \lambda^2 - 2\,\lambda^2\,\boldsymbol{s}_1^T\mathbf{C}_1^{-1}\boldsymbol{s}_1 \\ &= \lambda^2 + \lambda^2 - 2\lambda^2 = 0\,. \end{aligned}$$

From (43), we obtain $\boldsymbol{w}_1 = \sqrt{1-\rho_1^2}\lambda\,\mathbf{C}_1^{-1}\,\boldsymbol{s}_1$. Since $\boldsymbol{w}_1$ is a unit-length vector, this vector takes the form

$$\boldsymbol{w}_1 = \frac{\mathbf{C}_1^{-1}\,\boldsymbol{s}_1}{\sqrt{\boldsymbol{s}_1^T\mathbf{C}_1^{-2}\,\boldsymbol{s}_1}} \tag{45}$$

and ρ_1 can be derived from

$$(1-\rho_1^2)\lambda^2 = \frac{1}{\boldsymbol{s}_1^T\mathbf{C}_1^{-2}\,\boldsymbol{s}_1}\,, \tag{46}$$

which yields the optimal ρ_1 as in (7). Similarly, the expressions (6) and (7) hold for other $n = 2,\ldots,N$. When given $\boldsymbol{w}_n$ and ρ_n, the vectors $\boldsymbol{u}_n$ are computed from (2). ■

Appendix C
Proof of Theorem 1

Proof: Since $\mathbf{X}_1\,\mathbf{X}_2^{-1} = \mathbf{X}_2(\mathbf{X}_2^{-1}\,\mathbf{X}_1)\,\mathbf{X}_2^{-1}$, $\mathbf{X}_1\,\mathbf{X}_2^{-1}$ and $\mathbf{X}_2^{-1}\,\mathbf{X}_1$ are similar, and have the same eigenvalues λ

$$\begin{aligned} \mathbf{X}_1\,\mathbf{X}_2^{-1}\,\boldsymbol{a} &= \boldsymbol{a}\,\lambda \\ \boldsymbol{b}^T\,\mathbf{X}_2{}^{-1}\mathbf{X}_1 &= \boldsymbol{b}^T\,\lambda. \end{aligned}$$

Hence, $\boldsymbol{a}$ and $\widetilde{\boldsymbol{b}} = \frac{(\mathbf{X}_2{}^{-1})^T\boldsymbol{b}}{\boldsymbol{b}^T\,\mathbf{X}_2{}^{-1}\,\boldsymbol{a}}$ with $\widetilde{\boldsymbol{b}}^T\,\boldsymbol{a} = 1$, are left and right eigenvectors of the matrix $\mathbf{X}_1\,\mathbf{X}_2^{-1}$. Let $\mathbf{B}$ be an orthogonal complement to $\widetilde{\boldsymbol{b}}$, we define a matrix $\mathbf{Q}$ of size $R\times R$

$$\mathbf{Q} = [\boldsymbol{a},\,\mathbf{B}]\,,$$

whose inverse is given by

$$\mathbf{Q}^{-1} = \begin{bmatrix} \widetilde{\boldsymbol{b}}^T \\ \mathbf{B}^T\,\left(\mathbf{I}_R - \boldsymbol{a}\widetilde{\boldsymbol{b}}^T\right) \end{bmatrix}.$$

It is obvious that

$$\mathbf{Q}^{-1}\,\mathbf{X}_1\,\mathbf{X}_2^{-1}\,\mathbf{Q} = \begin{bmatrix} \lambda & \\ & \mathbf{C} \end{bmatrix},$$

where $\mathbf{C} = \mathbf{B}^T\,\left(\mathbf{I}_R - \boldsymbol{a}\widetilde{\boldsymbol{b}}^T\right)\,\mathbf{X}_1\,\mathbf{X}_2^{-1}\,\mathbf{B}$ is of size $(R-1)\times(R-1)$, and has rank-$(R-1)$. It follows that

$$\begin{aligned} \mathbf{X}_1 &= \lambda\,\boldsymbol{a}\widetilde{\boldsymbol{b}}^T\,\mathbf{X}_2 + \mathbf{B}\,\mathbf{C}\,\mathbf{B}^T\,\left(\mathbf{I}_R - \boldsymbol{a}\widetilde{\boldsymbol{b}}^T\right)\,\mathbf{X}_2 \\ &= \frac{1}{\boldsymbol{b}^T\,\mathbf{X}_1^{-1}\,\boldsymbol{a}}\,\boldsymbol{a}\,\boldsymbol{b}^T + \mathbf{B}\,\mathbf{C}\,\mathbf{B}^T\,\left(\mathbf{I}_R - \boldsymbol{a}\widetilde{\boldsymbol{b}}^T\right)\,\mathbf{X}_2, \end{aligned} \tag{47}$$

which indicates that $\mathbf{X}_1$ can be split into a rank-1 matrix composed by generalized eigenvectors $\frac{1}{\boldsymbol{b}^T\,\mathbf{X}_1^{-1}\,\boldsymbol{a}}\,\boldsymbol{a}\,\boldsymbol{b}^T$, and a rank-$(R-1)$ term.

Next, $\mathbf{U}$ and $\mathbf{V}$ can be chosen arbitrarily as orthonormal basis vectors of the column and row space of the second term in (47) or of the matrix $(\mathbf{X}_1 - \lambda\,\mathbf{X}_2)$. The matrices are then normalized to be orthogonal to $\boldsymbol{a}$ and $\boldsymbol{b}$, respectively, i.e., $\mathbf{U}^T\boldsymbol{a} = \rho_1\,[1,0,\ldots,0]^T$, $\mathbf{V}^T\boldsymbol{b} = \rho_2\,[1,0,\ldots,0]^T$ where $0\le\rho_1<1$ and $0\le\rho_2<1$. ■

Appendix D
The FIM $\mathbf{I}(\tilde{\alpha})$ in (29)

Consider the error function

$$\begin{aligned} &D(\boldsymbol{\mathcal{Y}}|\boldsymbol{\alpha}) \\ &= \frac{1}{2}\|\boldsymbol{\mathcal{Y}} - \lambda[\![\{\boldsymbol{a}^{(n)}\}]\!] + \lambda\rho[\![\{\boldsymbol{u}_n\}]\!] - \boldsymbol{\mathcal{Y}}\times\{\mathbf{I}_R - \boldsymbol{w}_n\boldsymbol{w}_n^T\}\|_F^2\,. \end{aligned} \tag{48}$$

The Hessian of the error function with respect to $\boldsymbol{\alpha}$ is evaluated at $\boldsymbol{\mathcal{Y}} = \hat{\boldsymbol{\mathcal{Y}}}$, i.e., when the tensor-valued function $\boldsymbol{\mathcal{E}}$ inside the Frobenius norm in (48) goes to zero. For such case, the Hessian is given by

$$\mathbf{H} = \mathbf{J}_D(\boldsymbol{\alpha})^T\,\mathbf{J}_D(\boldsymbol{\alpha}) \tag{49}$$

where $\mathbf{J}_D(\boldsymbol{\alpha})$ is the Jacobian of the tensor-valued function $\boldsymbol{\mathcal{E}}$ with respect to $\boldsymbol{\alpha}$ and evaluated when $\boldsymbol{\mathcal{Y}} = \hat{\boldsymbol{\mathcal{Y}}}$.

The Hessian $\mathbf{H}$ of size $(2N(R+1)+1)\times(2N(R+1)+1)$ is given in block form as

$$\mathbf{H} = \begin{bmatrix} \mathbf{H}_{\mathbf{V},\mathbf{V}} & \mathbf{H}_{\mathbf{V},\boldsymbol{\rho}} & \boldsymbol{h}_{\mathbf{V},\lambda} \\ \mathbf{H}_{\mathbf{V},\boldsymbol{\rho}}^T & \mathbf{H}_{\boldsymbol{\rho},\boldsymbol{\rho}} & \boldsymbol{h}_{\boldsymbol{\rho},\lambda} \\ \boldsymbol{h}_{\mathbf{V},\lambda}^T & \boldsymbol{h}_{\boldsymbol{\rho},\lambda}^T & 1-\rho^2 \end{bmatrix} \tag{50}$$

where $\mathbf{H}_{\mathbf{V},\mathbf{V}} = [\mathbf{H}_{\mathbf{V}_n,\mathbf{V}_m}]_{n,m=1}^N$ is an $N\times N$ partitioned matrix of submatrices $\mathbf{H}_{\mathbf{V}_n,\mathbf{V}_m}$ of size $2R\times 2R$, $\mathbf{H}_{\mathbf{V},\boldsymbol{\rho}} = [\mathbf{H}_{\mathbf{V}_n,\boldsymbol{\rho}_m}]_{n,m=1}^N$ is a partitioned matrix of $\mathbf{H}_{\mathbf{V}_n,\boldsymbol{\rho}_m}$ of size $2R\times 2$, $\mathbf{H}_{\boldsymbol{\rho},\boldsymbol{\rho}} = [\mathbf{H}_{\boldsymbol{\rho}_n,\boldsymbol{\rho}_m}]$, $\mathbf{H}_{\boldsymbol{\rho}_n,\boldsymbol{\rho}_m}$ are of size 2×2, $\boldsymbol{h}_{\mathbf{V},\lambda} = [\boldsymbol{h}_{\boldsymbol{w}_1,\lambda}^T,\boldsymbol{h}_{\boldsymbol{u}_1,\lambda}^T,\ldots,\boldsymbol{h}_{\boldsymbol{w}_N,\lambda}^T,\boldsymbol{h}_{\boldsymbol{u}_N,\lambda}^T]^T$ is of length $2NR$, and $\boldsymbol{h}_{\boldsymbol{\rho},\lambda} = [\boldsymbol{h}_{\boldsymbol{\rho}_1,\lambda}^T,\ldots,\boldsymbol{h}_{\boldsymbol{\rho}_N,\lambda}^T]^T$

$$\mathbf{H}_{\mathbf{V}_n,\mathbf{V}_m} = \begin{bmatrix} \mathbf{H}_{\boldsymbol{w}_n,\boldsymbol{w}_m} & \mathbf{H}_{\boldsymbol{w}_n,\boldsymbol{u}_m} \\ \mathbf{H}_{\boldsymbol{u}_n,\boldsymbol{w}_m} & \mathbf{H}_{\boldsymbol{u}_n,\boldsymbol{u}_m} \end{bmatrix}, \quad \mathbf{H}_{\mathbf{V}_n,\boldsymbol{\rho}_m} = \begin{bmatrix} \mathbf{H}_{\boldsymbol{w}_n,\boldsymbol{\rho}_m} \\ \mathbf{H}_{\boldsymbol{u}_n,\boldsymbol{\rho}_m} \end{bmatrix} \tag{51}$$

where $\mathbf{V}_n = [\boldsymbol{w}_n,\boldsymbol{u}_n]$.

The element sub-matrices to construct the Hessian $\mathbf{H}$, i.e., $\mathbf{H}_{\boldsymbol{w}_n,\boldsymbol{w}_m}, \mathbf{H}_{\boldsymbol{w}_n,\boldsymbol{u}_m}, \mathbf{H}_{\boldsymbol{u}_n,\boldsymbol{u}_m},\ldots$, are given by

$$\begin{aligned} \mathbf{H}_{\boldsymbol{w}_n,\boldsymbol{w}_n} &= \boldsymbol{\Phi}_n + \lambda^2\xi_n^2(1-\rho_{-n}^2)\mathbf{I}_R \\ \mathbf{H}_{\boldsymbol{u}_n,\boldsymbol{u}_n} &= \lambda^2\rho_n^2(1-\rho_{-n}^2)\mathbf{I}_R \\ \mathbf{H}_{\boldsymbol{w}_n,\boldsymbol{u}_n} &= \lambda^2\xi_n\rho_n(1-\rho_{-n}^2)\mathbf{I}_R \\ \mathbf{H}_{\boldsymbol{u}_n,\boldsymbol{u}_m} &= \lambda^2\rho_n\rho_m\boldsymbol{a}^{(n)}\,\boldsymbol{a}^{(m)T} + \lambda^2\rho^2\boldsymbol{u}_n\boldsymbol{u}_m^T \\ &\quad - \lambda^2\rho\rho_{-n}\boldsymbol{a}^{(n)}\boldsymbol{u}_m^T - \lambda^2\rho\rho_{-m}\boldsymbol{u}_n\boldsymbol{a}^{(m)T} \\ \mathbf{H}_{\boldsymbol{w}_n,\boldsymbol{w}_m} &= -2\lambda\xi_n\xi_m\rho_{-[n,m]}\mathbf{Z}_{nm} + \lambda^2\xi_n\xi_m\boldsymbol{a}^{(n)}\,\boldsymbol{a}^{(m)T} \\ &\quad + \lambda^2\xi_n\xi_m\rho_{-[n,m]}^2(2\xi_n\xi_m\boldsymbol{w}_n\boldsymbol{w}_m^T - \rho_n\rho_m\boldsymbol{u}_n\boldsymbol{u}_m^T) \\ \mathbf{H}_{\boldsymbol{w}_n,\boldsymbol{u}_m} &= -\lambda\xi_n\rho_{-[n,m]}\mathbf{Z}_{n,m} \\ &\quad + \lambda^2\xi_n\rho_m(1-\rho_{-[n,m]}^2)\boldsymbol{a}^{(n)}\,\boldsymbol{a}^{(m)T} \\ &\quad + \lambda^2\xi_n\xi_m\rho_m\rho_{-[n,m]}^2(2\boldsymbol{a}^{(n)} - \rho_n\boldsymbol{u}_n)\boldsymbol{w}_m^T \\ \mathbf{H}_{\boldsymbol{u}_n,\boldsymbol{\rho}_n} &= \lambda^2\rho_n\left(1-\rho_{-n}^2\right)[\boldsymbol{w}_n,\boldsymbol{u}_n] \\ \mathbf{H}_{\boldsymbol{w}_n,\boldsymbol{\rho}_n} &= \lambda^2\xi_n(1-\rho_{-n}^2)[\boldsymbol{w}_n,\boldsymbol{u}_n] - \lambda\rho_{-n}\boldsymbol{z}_n\,[1,0] \\ \mathbf{H}_{\boldsymbol{u}_n,\boldsymbol{\rho}_m} &= \lambda^2\rho_n\,\boldsymbol{a}^{(n)}\,[\xi_m,\rho_m(1-\rho_{-[n,m]}^2)] \\ \mathbf{H}_{\boldsymbol{w}_n,\boldsymbol{\rho}_m} &= \lambda^2\xi_n\boldsymbol{a}^{(n)}\,[\xi_m,\rho_m(1-\rho_{-[n,m]}^2)] \\ &\quad - \lambda\xi_n\rho_{-[n,m]}\boldsymbol{z}_n[0,1] \\ \mathbf{H}_{\boldsymbol{\rho}_n,\boldsymbol{\rho}_n} &= \lambda^2\,\mathrm{diag}(1,1-\rho_{-n}^2) \\ \mathbf{H}_{\boldsymbol{\rho}_n,\boldsymbol{\rho}_m} &= \lambda^2(\boldsymbol{\rho}_n\,\boldsymbol{\rho}_m^T - \mathrm{diag}(0,\rho_{-n}\rho_{-m})) \\ \boldsymbol{h}_{\boldsymbol{u}_n,\lambda} &= \lambda\rho_n\left(1-\rho_{-n}^2\right)\boldsymbol{a}^{(n)} \\ \boldsymbol{h}_{\boldsymbol{w}_n,\lambda} &= \lambda\xi_n(1-\rho_{-n}^2)\boldsymbol{a}^{(n)} - \xi_n\rho_{-n}\boldsymbol{z}_n \\ \boldsymbol{h}_{\boldsymbol{\rho}_n,\lambda} &= \lambda[\xi_n,\rho_n\left(1-\rho_{-n}^2\right)]^T \end{aligned}$$

where $\rho_{-n} = \frac{\rho}{\rho_n}$, $\rho_{-[n,m]} = \frac{\rho}{\rho_n \rho_m}$ and

$$\begin{aligned}
\boldsymbol{z}_n &= \hat{\boldsymbol{\mathcal{Y}}} \times_{k \neq n} \{\boldsymbol{u}_k^T\} = \lambda \rho_{-n} \boldsymbol{a}^{(n)} + \mathbf{U}^{(n)} \boldsymbol{g}_1^{(n)} \\
\mathbf{Z}_{n,m} &= \hat{\boldsymbol{\mathcal{Y}}} \times_{\substack{k \neq n, \\ k \neq m}} \{\boldsymbol{u}_k^T\} \\
&= \lambda \rho_{-[n,m]} \boldsymbol{a}^{(n)} \boldsymbol{a}^{(m)T} + \mathbf{U}^{(n)} \mathbf{G}_1^{(n,m)} \mathbf{U}^{(m)T} \\
\boldsymbol{\Phi}_n &= \hat{\mathbf{Y}}_{(n)} \left(\bigotimes_{k \neq n} (\mathbf{I} - \boldsymbol{w}_k \boldsymbol{w}_k^T) \right) \hat{\mathbf{Y}}_{(n)}^T \\
&= \lambda^2 \rho_{-n}^2 \boldsymbol{a}^{(n)} \boldsymbol{a}^{(n)T} + \mathbf{U}^{(n)} \left(\mathbf{G}_{(n)} \mathbf{G}_{(n)}^T \right) \mathbf{U}^{(n)T} \\
&\quad + \lambda \rho_{-n} \mathbf{U}^{(n)} \boldsymbol{g}_1^{(n)} \boldsymbol{a}^{(n)T} + \lambda \rho_{-n} \boldsymbol{a}^{(n)} \boldsymbol{g}_1^{(n)T} \mathbf{U}^{(n)T}
\end{aligned}$$

where $\boldsymbol{g}_1^{(n)}$ is the first column of $\mathbf{G}_{(n)}$, i.e., the fiber of the tensor $\boldsymbol{\mathcal{G}}$ along mode-n with all other indices $i_k = 1$, $k \neq n$, $\mathbf{G}_1^{(n,m)}$ is the slice of $\boldsymbol{\mathcal{G}}$ along modes n and m with all other indices $i_k = 1$, $k \neq n, m$.

Appendix E
Jacobian of the Functional Equality Constraints and its Orthonormal Complement

The Jacobian matrix of the constraint functions $\boldsymbol{f}(\boldsymbol{\alpha})$ in (27) is given in block form as

$$\mathbf{J}_{\boldsymbol{f}}(\boldsymbol{\alpha}) = \texttt{bdiag}(\mathbf{J}_1, \ldots, \mathbf{J}_N, \boldsymbol{\rho}_1^T, \ldots, \boldsymbol{\rho}_N^T, 0) \tag{52}$$

where bdiag constructs a block diagonal matrix, and

$$\mathbf{J}_n = \begin{bmatrix} \boldsymbol{w}_n^T & \\ & \boldsymbol{u}_n^T \\ \frac{1}{\sqrt{2}} \boldsymbol{u}_n^T & \frac{1}{\sqrt{2}} \boldsymbol{w}_n^T \end{bmatrix} \in \mathbb{R}^{3 \times 2R}, \quad n = 1, \ldots, N. \tag{53}$$

We define a matrix $\mathbf{F}_n$ of size $2R \times (2R-3)$ as

$$\mathbf{F}_n = \begin{bmatrix} \tilde{\mathbf{U}}^{(n)} & & \frac{1}{\sqrt{2}} \boldsymbol{u}_n \\ & \tilde{\mathbf{U}}^{(n)} & \frac{-1}{\sqrt{2}} \boldsymbol{w}_n \end{bmatrix} \tag{54}$$

where $\tilde{\mathbf{U}}^{(n)} = \mathbf{U}_{2:R-1}^{(n)}$ takes the last $(R-2)$ columns of $\mathbf{U}^{(n)}$. It is straightforward to see that $[\mathbf{J}_n^T, \mathbf{F}_n]$ forms an orthonormal matrix of size $2R \times 2R$

$$[\mathbf{J}_n^T, \mathbf{F}_n]^T [\mathbf{J}_n^T, \mathbf{F}_n] = \mathbf{I}_{2R}\,.$$

Therefore, we can chose the orthogonal basis matrix $\mathbf{F}_{\boldsymbol{f}}(\boldsymbol{\alpha})$ of size $(2N(R+1)+1) \times (2N(R-1)+1)$ as follows

$$\mathbf{F}_{\boldsymbol{f}}(\boldsymbol{\alpha}) = \texttt{bdiag}\left(\mathbf{F}_1, \ldots, \mathbf{F}_N, \begin{bmatrix} -\rho_1 \\ \xi_1 \end{bmatrix}, \ldots, \begin{bmatrix} -\rho_N \\ \xi_N \end{bmatrix}, 1 \right). \tag{55}$$

Appendix F
Generation of Matrix With Specific Collinearity Degree

This appendix presents a method to generate a random matrix $\mathbf{A}$ of size $I \times R$ with $I \geq R$ whose collinearity degree between columns are identical to a specific value c, that is

$$\mathbf{A}^T \mathbf{A} = (1-c)\mathbf{I}_R + c\mathbf{1}_{R \times R}. \tag{56}$$

We define a matrix $\mathbf{Q}$ of size $R \times R$ as

$$\mathbf{Q} = \begin{bmatrix} 1 & c_1 & c_1 & \ldots & c_1 \\ & b_1 & c_2 & \ldots & c_2 \\ & & \ddots & \ddots & \vdots \\ & & & b_{R-2} & c_{R-1} \\ & & & & b_{R-1} \end{bmatrix} \tag{57}$$

where $c_1 = c$, $c_r = \frac{1}{b_{r-1}} \left(c - \sum_{i=1}^{r-1} c_i^2 \right)$, $b_r = \sqrt{1 - \sum_{i=1}^{r} c_i^2}$ for $r = 1, 2, \ldots, R-1$. It is obvious to see that $\boldsymbol{\Omega} = \mathbf{Q}^T \mathbf{Q} = (1-c)\mathbf{I}_R + c\mathbf{1}_{R \times R}$, because

$$\begin{aligned}
\omega_{r,r} &= \boldsymbol{q}_r^T \boldsymbol{q}_r = \sum_{i=1}^{r-1} c_i^2 + b_{r-1}^2 = 1 \\
\omega_{r,s} &= \boldsymbol{q}_r^T \boldsymbol{q}_s = \sum_{i=1}^{r-1} c_i^2 + b_{r-1} c_r = c, \qquad 1 \leq r < s \leq R.
\end{aligned}$$

Now with an arbitrary orthogonal matrix $\mathbf{U}$ of size $I \times R$, $\mathbf{U}^T \mathbf{U} = \mathbf{I}_R$, the matrix $\mathbf{A} = \mathbf{U}\mathbf{Q}$ satisfies the condition in (56).

Appendix G
Relation of Collinearity Degree c and Parameter ρ_n in Example 1

We consider a matrix $\mathbf{A} = [\boldsymbol{a}, \mathbf{B}]$ of size $R \times R$ with $\boldsymbol{a}_r^T \boldsymbol{a}_r = 1$ and $\boldsymbol{a}_r^T \boldsymbol{a}_s = c$ for $r \neq s$. $\boldsymbol{a}$ is the first column, and $\mathbf{B}$ comprises the last $(R-1)$ columns of $\mathbf{A}$. Since

$$\begin{aligned}
\mathbf{B}^T \mathbf{B} &= (1-c)\mathbf{I}_{R-1} + c\,\mathbf{1}_{R-1} \mathbf{1}_{R-1}^T \\
&= \frac{(R-2)c+1}{R-1} \mathbf{1}\mathbf{1}^T + (1-c) \left(\mathbf{I}_{R-1} - \frac{1}{R-1} \mathbf{1}\mathbf{1}^T \right),
\end{aligned}$$

the first eigenpair $(\boldsymbol{v}_1, \sigma_1^2)$ of the matrix $\mathbf{B}^T \mathbf{B} = \mathbf{V} \boldsymbol{\Sigma}^2 \mathbf{V}^T$, $\boldsymbol{\Sigma} = \mathrm{diag}(\sigma_1, \ldots, \sigma_{R-1})$ is given by

$$\boldsymbol{v}_1 = \frac{1}{\sqrt{R-1}} \mathbf{1}_{R-1}, \quad \boldsymbol{\sigma}_1 = \sqrt{(R-2)c+1} \tag{58}$$

the rest $(R-2)$ eigenvalues of $\mathbf{B}^T \mathbf{B}$ are identical

$$\boldsymbol{\sigma}_r = \sqrt{1-c}, \qquad r = 2, \ldots, R-1, \tag{59}$$

and $\mathbf{V}_{2:R-1} = [\boldsymbol{v}_2, \ldots, \boldsymbol{v}_{R-1}]$ can take arbitrary orthonormal basis of orthogonal complement to $\boldsymbol{v}_1$, i.e., $\mathbf{V}_{2:R-1}^T \mathbf{1}_{R-1} = \mathbf{0}$. This gives us singular value decomposition of the matrix $\mathbf{B}$

$$\mathbf{B} = \mathbf{U}\,\boldsymbol{\Sigma}\,\mathbf{V}^T \tag{60}$$

where $\mathbf{U}$ of size $R \times (R-1)$. We denote by $\boldsymbol{w}$ a unit-length vector of orthogonal complement to $\mathbf{U}$, i.e., $[\mathbf{U}, \boldsymbol{w}]$ forms an orthonormal matrix of size $R \times R$. The vector $\boldsymbol{a}$ can be always expressed as linear combination of $[\mathbf{U}, \boldsymbol{w}]$

$$\boldsymbol{a} = \mathbf{U}\boldsymbol{\beta} + \xi\,\boldsymbol{w} \tag{61}$$

where $\boldsymbol{\beta}$ is a vector of length $R-1$. Since

$$c\,\mathbf{1}_{R-1} = \mathbf{B}^T \boldsymbol{a} = \mathbf{V}\,\boldsymbol{\Sigma}\,\boldsymbol{\beta}\,,$$

we obtain

$$\boldsymbol{\beta} = c\boldsymbol{\Sigma}^{-1}\mathbf{V}^T\mathbf{1}_{R-1} = c\boldsymbol{\Sigma}^{-1}\left[\sqrt{R-1}, \mathbf{0}_{R-2}^T\right]^T = c\sqrt{\frac{R-1}{(R-2)c+1}}\,\boldsymbol{e}_1\,, \tag{62}$$

indicating that

$$\boldsymbol{a} = c\sqrt{\frac{R-1}{(R-2)c+1}}\,\boldsymbol{u}_1 + \xi\,\boldsymbol{w}. \tag{63}$$

Therefore, for Example 1, the parameter ρ_n defined for $\boldsymbol{a}_r^{(n)}$ and the rest $(R-1)$ columns of $\mathbf{A}^{(n)}$ is given by

$$\rho_n = c\sqrt{\frac{R-1}{(R-2)c+1}}\,. \tag{64}$$

ACKNOWLEDGMENT

The authors would like to thank the associate editor and the referees for their time and efforts to review this paper, and for very constructive comments which helped to improve the quality and presentation of the paper.

REFERENCES

[1] A.-H. Phan, P. Tichavský, and A. Cichocki, "Tensor deflation for CANDECOMP/PARAFAC—Part I: Alternating subspace update algorithm," *IEEE Trans. Signal Process.*, vol. 63, no. 22, pp. 5924–5938, 2015.

[2] A.-H. Phan, P. Tichavský, and A. Cichocki, "Deflation method for CANDECOMP/PARAFAC tensor decomposition," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process. (ICASSP)*, May 2014, pp. 6736–6740.

[3] A.-H. Phan, P. Tichavský, and A. Cichocki, "Tensor deflation for CANDECOMP/PARAFAC. Part 3: Rank splitting," ArXiv e-prints 2015 [Online]. Available: http://arxiv.org/abs/1506.04971

[4] L. De Lathauwer and D. Nion, "Decompositions of a higher-order tensor in block terms—Part III: Alternating least squares algorithms," *SIAM J. Matrix Anal. Appl.*, vol. 30, no. 3, pp. 1067–1083, 2008.

[5] P. Tichavský, A.-H. Phan, and A. Cichocki, "Tensor diagonalization—A new tool for PARAFAC and block-term decomposition," 2014, arXiv:1402.1673 [Online]. Available: http://arxiv.org/abs/1402.1673v1, to be published

[6] W. P. Krijnen, T. K. Dijkstra, and A. Stegeman, "On the non-existence of optimal solutions and the occurrence of degeneracy in the CANDECOMP/PARAFAC model," *Psychometrika*, vol. 73, pp. 431–439, 2008.

[7] P. Tichavský, A.-H. Phan, and Z. Koldovský, "Cramér-Rao-induced bounds for CANDECOMP/PARAFAC tensor decomposition," *IEEE Trans. Signal Process.*, vol. 61, no. 8, pp. 1986–1997, 2013.

[8] A. Cichocki, R. Zdunek, A.-H. Phan, and S. Amari, *Nonnegative Matrix and Tensor Factorizations: Applications to Exploratory Multi-way Data Analysis and Blind Source Separation*. Chichester, U.K.: Wiley, 2009.

[9] L. De Lathauwer, B. De Moor, and J. Vandewalle, "On the best rank-1 and rank-(R1, R2, . . . , RN) approximation of higher-order tensors," *SIAM J. Matrix Anal. Appl.*, vol. 21, no. 4, pp. 1324–1342, 2000.

[10] E. Sanchez and B. R. Kowalski, "Tensorial resolution: A direct trilinear decomposition," *J. Chemometr.*, vol. 4, pp. 29–45, 1990.

[11] A.-H. Phan, P. Tichavský, and A. Cichocki, "CANDECOMP/PARAFAC decomposition of high-order tensors through tensor reshaping," *IEEE Trans. Signal Process.*, vol. 61, no. 19, pp. 4847–4860, 2013.

[12] A. Stegeman, "CANDECOMP/PARAFAC: From diverging components to a decomposition in block terms," *SIAM J. Matrix Anal. Appl.*, vol. 33, no. 2, pp. 291–316, 2012.

[13] N. Sidiropoulos, G. Giannakis, and R. Bro, "Blind PARAFAC receivers for DS-CDMA systems," *IEEE Trans. Signal Process.*, vol. 48, no. 3, pp. 810–823, 2000.

[14] A. S. Field and D. Graupe, "Topographic component (parallel factor) analysis of multichannel evoked potentials: Practical issues in trilinear spatiotemporal decomposition," *Brain Topograph.*, vol. 3, no. 4, pp. 407–423, 1991.

[15] K. Vanderperren, B. Mijovi, N. Novitskiy, B. Vanrumste, P. Stiers, and B. R. H. Van den Bergh *et al.*, "Single trial ERP reading based on parallel factor analysis," *Psychophysiol.*, vol. 50, no. 1, pp. 97–110, 2013.

[16] B. M. Hochwald and A. Nehorai, "Concentrated Cramér-Rao bound expressions," *IEEE Trans. Inf. Theory*, vol. 40, no. 2, pp. 363–371, 1994.

[17] P. Stoica and B. C. Ng, "On the Cramér-Rao bound under parametric constraints," *IEEE Signal Process. Lett.*, vol. 5, no. 7, pp. 177–179, Jul. 1998.

[18] T. J. Moore, "A Theory of Cramér-Rao bounds for constrained parametric models," Ph.D. dissertation, Univ. of Maryland, College Park, MD, USA, 2010.

Anh-Huy Phan (M'15) received the master's degree from Hochiminh University of Technology, Vietnam, in 2005, and the Ph.D. degree from the Kyushu Institute of Technology, Japan, in 2011. He is a research scientist at the Laboratory for Advanced Brain Signal Processing, RIKEN. He has served on the editorial board of the International Journal of Computational Mathematics. His research interests include multilinear algebra, tensor computation, blind source separation and brain computer interface.

Petr Tichavský (M'98–SM'04) received the Ph.D. degree in theoretical cybernetics from the Czechoslovak Academy of Sciences in 1992. Since that time he is with the Institute of Information Theory and Automation of the Czech Academy of Sciences in Prague. In 1994 he received the Fulbright grant for a 10 month fellowship at Yale University, Department of Electrical Engineering, in New Haven, U.S.A. He is author and co-author of research papers in the area of sinusoidal frequency/frequency-rate estimation, adaptive filtering and tracking of time varying signal parameters, algorithm-independent bounds on achievable performance, sensor array processing, independent component analysis and blind source separation, and tensor decompositions. Petr Tichavský served as associate editor of the IEEE SIGNAL PROCESSING LETTERS from 2002 to 2004, and as associate editor of the IEEE TRANSACTIONS ON SIGNAL PROCESSING from 2005 to 2009 and from 2011 to now. Petr Tichavský has also served as a general co-chair of the 36th IEEE International Conference on Acoustics, Speech and Signal Processing ICASSP 2011 in Prague, Czech Republic, and as general co-chair of the 12th International Conference on Latent Variable Analysis and Signal Separation LVA/ICA 2015 in Liberec, Czech Republic.

Andrzej Cichocki (M'96–SM'07–F'13) received the M.Sc. (with honors), Ph.D. and Dr.Sc. (Habilitation) degrees, all in electrical engineering from Warsaw University of Technology (Poland). He spent several years at University Erlangen (Germany) as an Alexander-von-Humboldt Research Fellow and Guest Professor. In 1995–1997 he was a team leader of the Laboratory for Artificial Brain Systems, at Frontier Research Program RIKEN (Japan), in the Brain Information Processing Group. He is currently a Senior Team Leader and Head of the laboratory for Advanced Brain Signal Processing, at RIKEN Brain Science Institute (Japan) and affiliated full professor in several universities. He has served as an Associated Editor of the IEEE TRANSACTIONS ON NEURAL NETWORKS, the IEEE TRANSACTIONS ON CYBERBETICS, the IEEE TRANSACTIONS ON SIGNALS PROCESSING, the *Journal of Neuroscience Methods* and as founding Editor in Chief for the *Journal Computational Intelligence and Neuroscience*. He is (co)author of more than 400 technical journal papers and 4 monographs in English (two of them translated to Chinese). His publications currently report over 25,000 citations according to Google Scholar, with an h-index of 70. His researches focus on tensor decompositions, brain machine interface, human robot interactions, and EEG hyper-scanning, and their practical applications.