

# Tensor Deflation for CANDECOMP/PARAFAC— Part I: Alternating Subspace Update Algorithm

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

**Abstract**—CANDECOMP/PARAFAC (CP) approximates multiway data by sum of rank-1 tensors. Unlike matrix decomposition, the procedure which estimates the best rank- $R$  tensor approximation through  $R$  sequential best rank-1 approximations does not work for tensors, because the deflation does not always reduce the tensor rank. In this paper, we propose a novel deflation method for the problem. When one factor matrix of a rank- $R$  CP decomposition is of full column rank, the decomposition can be performed through  $(R - 1)$  rank-1 reductions. At each deflation stage, the residue tensor is constrained to have a reduced multilinear rank. For decomposition of order-3 tensors of size  $R \times R \times R$  and rank- $R$ , estimation of one rank-1 tensor has a computational cost 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. The method can be extended to tracking one or a few rank-one tensors of slow changes, or inspect variations of common patterns in individual datasets.

**Index Terms**—Canonical polyadic decomposition, complex-valued tensor decomposition, PARAFAC, tensor deflation, tensor tracking.

## I. INTRODUCTION

CANDECOMP/PARAFAC (CP) has found numerous applications in wide variety of areas such as in chemometrics, telecommunication [1], data mining, neuroscience [2]–[4], separated representations, blind source separation [5]. It consists in decomposition of a given tensor (a multiway array of real or complex numbers with three or more indices) as a sum of the lowest possible number of rank-1 tensors (see illustration in Fig. 1(b)). Here, a rank-one tensor is an outer product of vectors. This decomposition became popular due to Carroll and Chang [6] and Harshman [7].

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, Wak 351-0198, Japan (e-mail: phan@brain.riken.jp).

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

A. Cichocki 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.2015.2458785

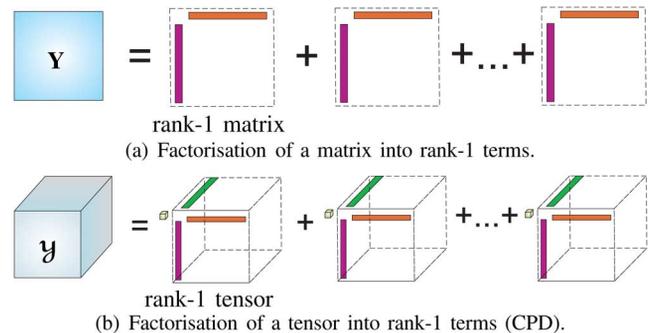


Fig. 1. (a) Illustration of factorization of a matrix into  $R$  rank-1 matrices  $\mathbf{Y} = \sum_{r=1}^R \mathbf{a}_r \mathbf{b}_r^T$ , (b) CANDECOMP/PARAFAC factorizes a rank- $R$  tensor into rank-1 tensors  $\mathcal{Y} = \sum_{r=1}^R \lambda_r \mathbf{a}_r \circ \mathbf{b}_r \circ \mathbf{c}_r$ . In matrix factorisation, rank-1 matrices can be sequentially estimated and deflated out of the matrix  $\mathbf{Y}$ . A similar sequential estimation of rank-1 tensors in general cannot be applied to tensors.

Together with the Tucker tensor decomposition, this CP decomposition is the most successful extension of the matrix decompositional approach to multiway data. In existing CP decomposition algorithms, all rank-one tensor components are estimated jointly. In this paper, the main objective is to estimate the rank-one components one by one, or only a few ones. Algorithms of this type are called deflation algorithms.

One of important properties in matrix factorisation methods like eigenvalue decomposition, singular value decomposition, is that rank-1 matrix components can be sequentially estimated via deflation method. An illustration of matrix factorization is shown in Fig. 1(a). For example, the power method of the eigendecomposition of a matrix [8], [9] falls in this class. The power method is especially useful when a few leading eigenvectors are of interest, or when estimating eigenvectors while one or more eigenpairs are known. Similar deflation methods are applied to extract independent components in blind source separation one by one [10], or in sparse principal components [11].

The matrix deflation procedure is possible because subtracting the best rank-1 term from a matrix reduces the matrix rank [12]. Unfortunately, this property, in general, does not hold for multiway arrays. Subtraction of the best fitting rank-one component works well only for symmetric decomposition of symmetric tensors [13]. However, for general tensors, we cannot guarantee to obtain a good rank- $R$  tensor approximation through  $R$  sequential rank-1 estimations [14]. The authors in [15] confirmed that subtracting the best rank-1 tensor from a tensor may increase its rank. Nevertheless, the standard deflation or sequential extraction of a rank-1 tensor has been used in practice in  $N$ -way Partial Least Squares Regression (PLS)

[16], [17]. The decomposition seeks a common rank-1 tensor between two multiway dataset, and deflates it out of both data to find further rank-1 tensors.

In this paper, we present a novel method for the rank-reduction problem in the CP decomposition. We show that for low-rank CP decomposition, we can sequentially extract rank-1 tensors from a rank- $R$  tensor if the residue tensor is constrained to be of multilinear rank- $(R-1)$ . It means that all matricizations of the residue tensor have rank at most  $(R-1)$ . The tensor deflation can be seen as a Tucker decomposition with one scalar and one block along the diagonal of the core tensor. In this sense, tensor deflation is more closely related to (block) tensor diagonalization (TEDIA) [18] which seeks factor matrices in the sense that the tensor  $\mathcal{Y}$  can be transformed into a block diagonal tensor.

The tensor deflation has many applications in extracting one or a few rank-1 tensors from a high rank tensor. For CPD, this advantage allows to simultaneously extract rank-1 tensors in a parallel system when using different initial points generated by different initialization methods. It may easily occur that the estimated rank-1 components are estimates of distinct components (or belong to distinct components). It could happen that some initial points lead to the same solutions, i.e., rank-1 one tensors, and we cannot extract all rank-1 tensors in one simultaneous run. However, the extraction process should be executed further on the residual tensor after the first run.

The tensor deflation can be used to track components of rank-1 tensors of interest in a system for online data receiving. When new data is coming, one only needs to inspect the change of specific rank-one tensors in the entire data or only a partial new data. A simple approach is to decompose the whole data to extract all components, but this is not an efficient method for big data and for online analysis. Alternatively, one can track all components using a partial old and new data as proposed in [19]. However, so far, the existing methods do not allow to track specific components. Note that in practice, there may be only a few components that are actually relevant while most of them are less significant. This can be achieved using the tensor deflation proposed in this paper.

Using the similar technique, we can inspect common loadings, e.g., biomarkers, in the entire data of multiway samples, and their individual variations. This is useful to extract features for new test data associated with some components selected from the training data.

In the conference paper [20], we presented a deflation algorithm for the CP decomposition for the first time. In this paper we extend the method, and the main contributions are summarized below

- Conditions for the tensor deflation are formulated to guarantee success of the rank-1 tensor extraction. In general, a rank-1 tensor can be “pulled” out of the data if its components do not lie in the column spaces of components of other rank-1 tensors. In most cases, the rank-1 tensor can even be extracted when having at least two components which cannot be expressed by linear combination of the other components.
- We propose new algorithms for real- and complex-valued tensor deflations. Unlike the algorithm in [20], the novel

algorithms have a lower number of parameters to estimate, basically two vectors and one scalar per dimension. This is possible because determining the block part of the decomposition is ambiguous. Complexity of the rank-1 deflation is  $\mathcal{O}(R^N)$  which is lower than the cost of the CPD-ALS algorithm to estimate  $R$  rank-1 tensors from a tensor of size  $R \times R \times \dots \times R$ .

- We also briefly introduce new applications for tensor decomposition involving tensor tracking whose major aim is to track components of one or a few rank-1 tensors slowly changing in an online system. The rank-1 tensor deflation is also useful to inspect common components in individual data entries.
- Furthermore, we illustrate an ability to extract rank-1 tensors in CPDs through both sequential and parallel processes.

Efficient initialization methods and the Cramér-Rao Bound for the rank-1 tensor deflation are presented in Part 2 of this work [21], whereas rank splitting for CPD, an extension of the tensor deflation, is described in Part 3 [22], [23].

The paper is organized as follows. A tensor decomposition for rank-1 tensor deflation with necessary conditions is discussed in Section II. Novel algorithms for real- and complex-valued tensor deflations based on subspace updating are presented in Sections III and V, respectively. Section IV discusses the tensor deflation with rank exceeding the true rank of the data. Simulations in Section VI will compare performance of algorithms using different initialization methods. The simulation results verify validity and performance of the proposed algorithms for real-world data and in application for tracking received signals in a direct sequence code division multiple access (DS-CDMA) system. Section VII concludes the paper.

Throughout the paper, we shall denote tensors by bold calligraphic letters, e.g.,  $\mathcal{A} \in \mathbb{R}^{I_1 \times I_2 \times \dots \times I_N}$ , matrices by bold capital letters, e.g.,  $\mathbf{A} = [\mathbf{a}_1, \mathbf{a}_2, \dots, \mathbf{a}_R] \in \mathbb{R}^{I \times R}$ , and vectors by bold italic letters, e.g.,  $\mathbf{a}_j$ . An  $\mathbf{i} = [i_1, i_2, \dots, i_N]$ -th entry  $y_{\mathbf{i}} = \mathcal{Y}(i_1, i_2, \dots, i_N)$  with  $1 \leq i_n \leq I_n$ ,  $n = 1, 2, \dots, N$ , is alternatively denoted by  $y_i$  with the linear index  $i$ . The Kronecker and Hadarmard products are denoted by  $\otimes$  and  $\circledast$ , respectively. Inner product of two tensors is denoted by  $\langle \mathcal{X}, \mathcal{Y} \rangle = \text{vec}(\mathcal{X})^T \text{vec}(\mathcal{Y})$ . Contraction between two tensors along modes- $\mathbf{m}$ , where  $\mathbf{m} = [m_1, \dots, m_K]$ , is denoted by  $\langle \mathcal{X}, \mathcal{Y} \rangle_{\mathbf{m}}$ , whereas  $\langle \mathcal{X}, \mathcal{Y} \rangle_{-n}$  represents contraction along all modes but mode- $n$ .

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

$$\mathcal{Y} \times \left\{ \mathbf{U}^{(n)} \right\} \triangleq \mathcal{Y} \times_1 \mathbf{U}^{(1)} \times_2 \mathbf{U}^{(2)} \dots \times_N \mathbf{U}^{(N)}.$$

We say that a tensor  $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \dots \times I_N}$  is in Kruskal form if

$$\mathcal{X} = \sum_{r=1}^R \lambda_r \mathbf{a}_r^{(1)} \circ \mathbf{a}_r^{(2)} \circ \dots \circ \mathbf{a}_r^{(N)}, \quad (1)$$

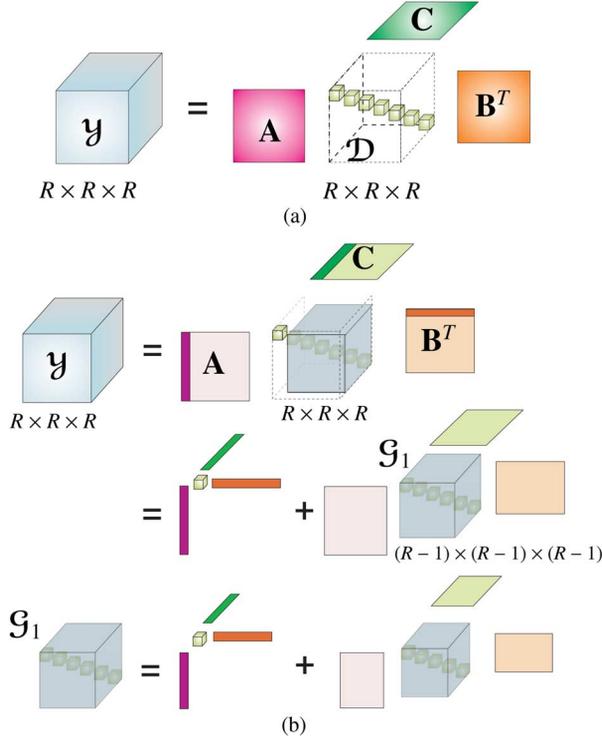


Fig. 2. Illustration of rank-1 deflation for the CP decomposition of a rank- $R$  tensor  $\mathcal{Y}$ . The proposed method estimates a rank-1 tensor and returns a tensor  $\mathcal{G}_1$  of multilinear rank- $(R-1, \dots, R-1)$  and of smaller size than  $\mathcal{Y}$ , that is compression. The deflation then can be applied to  $\mathcal{G}_1$  to extract the second rank-1 tensor, and receive a multilinear rank- $(R-2, \dots, R-2)$  tensor  $\mathcal{G}_2$ , and so on. The CP decomposition of  $\mathcal{Y}$  can be fully achieved after  $(R-1)$  rank-1 deflations. (a) CPD of a rank- $R$  tensor. (b) Rank-1 deflation for CPD.

where “ $\circ$ ” denotes the outer product,  $\mathbf{A}^{(n)} = [\mathbf{a}_1^{(n)}, \mathbf{a}_2^{(n)}, \dots, \mathbf{a}_R^{(n)}] \in \mathbb{R}^{I_n \times R}$  are factor matrices,  $\mathbf{a}_r^{(n)T} \mathbf{a}_r^{(n)} = 1$ , for  $r = 1, \dots, R$  and  $n = 1, \dots, N$ , and  $\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_R > 0$ .

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

$$\mathcal{X} = \sum_{r_1=1}^{R_1} \sum_{r_2=1}^{R_2} \dots \sum_{r_N=1}^{R_N} g_{r_1 r_2 \dots r_N} \mathbf{a}_{r_1}^{(1)} \circ \mathbf{a}_{r_2}^{(2)} \circ \dots \circ \mathbf{a}_{r_N}^{(N)}, \quad (2)$$

where  $\mathcal{G} = [g_{r_1 r_2 \dots r_N}]$ , and  $\mathbf{A}^{(n)}$  are of full column rank. For compact expression,  $\llbracket \boldsymbol{\lambda}; \{\mathbf{A}^{(n)}\} \rrbracket$  denotes a Kruskal tensor, where  $\llbracket \mathcal{G}; \{\mathbf{A}^{(n)}\} \rrbracket$  represents a Tucker tensor [24].

*Definition 1. (CANDECOMP/PARAFAC (CP) [6], [7]):* Approximation of an order- $N$  data tensor  $\mathcal{Y} \in \mathbb{R}^{I_1 \times I_2 \times \dots \times I_N}$  by a rank- $R$  tensor in the Kruskal form means  $\mathcal{Y} = \hat{\mathcal{Y}} + \mathcal{E}$ , where  $\hat{\mathcal{Y}} = \llbracket \boldsymbol{\lambda}; \mathbf{A}^{(1)}, \mathbf{A}^{(2)}, \dots, \mathbf{A}^{(N)} \rrbracket$ , so that  $\|\mathcal{Y} - \hat{\mathcal{Y}}\|_F^2$  is minimized.

Fig. 2(a) illustrates a CPD for order-3 tensor. It is worth noting that exact CP decomposition may not exist [25]. For various aspects of multiway analysis and its applications in signal processing, we refer to recent papers [26], [27].

## II. A TENSOR DECOMPOSITION FOR RANK-1 TENSOR EXTRACTION

In this paper we consider an order- $N$  tensor  $\mathcal{Y}$  of size  $I_1 \times I_2 \times \dots \times I_N$  which admits the CP decomposition (CPD) of rank  $R$  with  $R \leq I_n$  for all  $n$ . Tensor  $\mathcal{Y}$  can be expressed as a

summation of a rank-1 tensor  $\mathcal{Y}_1$  and a rank- $(R-1)$  tensor  $\mathcal{Y}_2$ , that is

$$\begin{aligned} \mathcal{Y} &\approx \lambda_1 \mathbf{a}_1^{(1)} \circ \mathbf{a}_1^{(2)} \circ \dots \circ \mathbf{a}_1^{(N)} + \sum_{r=2}^R \lambda_r \mathbf{a}_r^{(1)} \circ \mathbf{a}_r^{(2)} \circ \dots \circ \mathbf{a}_r^{(N)} \\ &= \mathcal{Y}_1 + \mathcal{Y}_2. \end{aligned} \quad (3)$$

Instead of considering  $\mathcal{Y}_2$  as a rank- $(R-1)$  tensor, we constrain it as a residual of multi-linear rank- $(R-1, \dots, R-1)$  whose factor matrices form subspaces of  $\mathbf{A}_{2:R}^{(n)}$ . This motivates a procedure for factorizing  $\mathcal{Y}$  into two tensor blocks of rank-1, and multilinear rank- $(R-1, \dots, R-1)$

$$\mathcal{Y} \approx \llbracket \boldsymbol{\lambda}; \mathbf{a}^{(1)}, \mathbf{a}^{(2)}, \dots, \mathbf{a}^{(N)} \rrbracket + \llbracket \mathcal{G}; \mathbf{U}^{(1)}, \mathbf{U}^{(2)}, \dots, \mathbf{U}^{(N)} \rrbracket, \quad (4)$$

where  $\mathbf{U}^{(n)}$  are of size  $I_n \times (R-1)$ . The decomposition is a particular case of the block component decomposition (BCD) for order-3 tensors [28], [29]. Further rank-1 tensors can be extracted from the core tensor  $\mathcal{G}$  of size  $(R-1) \times (R-1) \times \dots \times (R-1)$  rather than the original data  $\mathcal{Y}$  as illustrated in Fig. 2(b). The process is sequentially applied  $(R-1)$  times and briefly summarized in Steps 3–4 in Algorithm 1. The rank-1 tensors extracted from the core tensors  $\mathcal{G}_1, \dots, \mathcal{G}_{r-1}$  of smaller sizes need to project back to the size of the data tensor  $\mathcal{Y}$  as in Step 4. More specifically, the projection from the second stage of deflation to the first stage is expressed as

$$\begin{aligned} \mathcal{Y} &= \llbracket \lambda_1; \{\mathbf{a}_1^{(n)}\} \rrbracket + \mathcal{G}_1 \times \{\mathbf{U}_1^{(n)}\} \\ &= \llbracket \lambda_1; \{\mathbf{a}_1^{(n)}\} \rrbracket + \left( \llbracket \lambda_2; \{\mathbf{a}_2^{(n)}\} \rrbracket + \mathcal{G}_2 \times \{\mathbf{U}_2^{(n)}\} \right) \\ &\quad \times \{\mathbf{U}_1^{(n)}\} \\ &= \llbracket \lambda_1; \{\mathbf{a}_1^{(n)}\} \rrbracket + \llbracket \lambda_2; \{\mathbf{U}_1^{(n)} \mathbf{a}_2^{(n)}\} \rrbracket + \mathcal{G}_2 \\ &\quad \times \{\mathbf{U}_1^{(n)} \mathbf{U}_2^{(n)}\}. \end{aligned}$$

In the last stage, the compressed tensor has size of  $2 \times 2 \times \dots \times 2$ . The final solution is a Kruskal tensor whose factor matrices comprise components of rank-1 Kruskal tensors  $\llbracket \lambda_r; \{\mathbf{a}_r^{(n)}\} \rrbracket$ ,  $r = 1, \dots, R$ . The method is primarily applicable if the tensor rank does not exceed the tensor dimensions, but when the rank may slightly exceed the tensor dimension, the proposed method still gives good results.

The main technique to extract a rank-1 tensor from a rank- $R$  tensor  $\mathcal{Y}$  is to perform a rank-1 plus multilinear rank- $(R-1)$  BCD. This raises a question whether the rank-1 tensor in (4) is identical to one of rank-1 tensors in (3). In order to address this concern, we introduce the following normalization and some additional conditions.

*Lemma 1 (Orthogonal Normalization):* Given an approximation of  $\mathcal{Y}$  by a sum of two tensors  $\llbracket \boldsymbol{\lambda}; \mathbf{a}^{(1)}, \mathbf{a}^{(2)}, \dots, \mathbf{a}^{(N)} \rrbracket$  and  $\llbracket \mathcal{G}; \mathbf{U}^{(1)}, \mathbf{U}^{(2)}, \dots, \mathbf{U}^{(N)} \rrbracket$ , where  $\mathbf{U}^{(n)} \in \mathbb{R}^{I_n \times (R-1)}$ ,  $R \leq I_n$ , one can construct an equivalent decomposition, denoted by tildas, which has the same approximation error,

- Matrices  $\tilde{\mathbf{U}}^{(n)}$  are with orthonormal columns for all  $n$ , i.e.,  $(\tilde{\mathbf{U}}^{(n)})^T \tilde{\mathbf{U}}^{(n)} = \mathbf{I}_{R-1}$ .
- The last  $(R-2)$  columns of  $\tilde{\mathbf{U}}^{(n)}$ , denoted by  $\tilde{\mathbf{u}}_2^{(n)}, \dots, \tilde{\mathbf{u}}_{R-1}^{(n)}$  form an arbitrary orthonormal basis for

orthogonal complement to  $[\mathbf{a}^{(n)}, \mathbf{u}_1^{(n)}]$  in the column space of  $\mathbf{U}^{(n)}$ ,  $(\tilde{\mathbf{u}}_1^{(n)})^T \mathbf{a}^{(n)} \geq 0$  and  $(\tilde{\mathbf{u}}_k^{(n)})^T \mathbf{a}^{(n)} = 0$  for  $k = 2, \dots, R-1$ .

*Lemma 2:* Let (4) be a decomposition of a tensor  $\mathcal{Y}$  into rank one and multilinear rank  $(R_1, \dots, R_N)$  tensors. Then,

- $\mathbf{a}^{(n)}$  does not lie in the column space of  $\mathbf{U}^{(n)}$ , otherwise  $\mathbf{a}^{(n)} = \mathbf{u}_1^{(n)}$ .
- The coefficient  $\rho = \prod_{n=1}^N (\mathbf{u}_1^{(n)T} \mathbf{a}^{(n)})$  is neither equal to  $-1$  nor  $1$ .
- At least two components  $\mathbf{a}^{(n)}$ , for  $n = 1, 2, \dots, N$  must not lie within the column spaces of the corresponding  $\mathbf{U}^{(n)}$ . Otherwise, the rank-1 tensor  $[[\mathbf{a}^{(1)}, \mathbf{a}^{(2)}, \dots, \mathbf{a}^{(N)}]]$  is not uniquely identified.

The proofs are given in Appendices A and B. Hereinafter, we assume that such normalization (and rotation with respect to  $\mathbf{a}^{(n)}$ ) is applied to the factor matrices  $\mathbf{U}^{(n)}$ .

*Theorem 1 (Rank-1 Deflation):* Let a rank- $R$  tensor  $\mathcal{Y} = [[\boldsymbol{\beta}; \mathbf{B}^{(1)}, \dots, \mathbf{B}^{(N)}]]$  having

- at least one factor matrix  $\mathbf{B}^{(n)} \in \mathbb{R}^{I_n \times R}$  of full column rank,
- and an exact decomposition  $\mathcal{Y} = [[\lambda; \mathbf{a}^{(1)}, \dots, \mathbf{a}^{(N)}]] + [[\mathcal{G}; \mathbf{U}^{(1)}, \dots, \mathbf{U}^{(N)}]]$ , where  $\mathbf{U}^{(n)} \in \mathbb{R}^{I_n \times R_n}$  and  $\mathcal{G}$  of size  $R_1 \times R_2 \times \dots \times R_N$ ,  $R_n < R$ , which satisfies condition (c) in Lemma 2.

Then vectors  $\mathbf{a}^{(1)}, \dots, \mathbf{a}^{(N)}$  are components  $\mathbf{b}_r^{(1)}, \dots, \mathbf{b}_r^{(N)}$  for a certain  $1 \leq r \leq R$ , and  $\mathcal{G}$  is of rank  $(R-1)$ .

*Proof:* Without loss of generality, we assume  $\mathbf{B}^{(N)}$  is of full-column rank. According to Lemma 1, the factor matrices  $\mathbf{U}^{(n)}$  can be rotated so that  $\mathbf{U}^{(n)T} \mathbf{U}^{(n)} = \mathbf{I}_{R-1}$  and  $\mathbf{U}^{(n)T} \mathbf{a}^{(n)} = \rho_n [1, 0, \dots, 0]^T$ , where  $\rho_n = \mathbf{u}_1^{(n)T} \mathbf{a}^{(n)} \geq 0$ . Assuming that  $0 \leq \rho_1, \rho_2 < 1$ . Let  $\tilde{\mathbf{u}}_1 = \mathbf{a}^{(1)} - \rho_1 \mathbf{u}_1^{(1)}$ , then  $\tilde{\mathbf{u}}_1^T \mathbf{U}^{(1)} = 0$ . By multiplying  $\mathcal{Y}$  with  $\tilde{\mathbf{u}}_1^T$  along mode-1 we have

$$\mathcal{Y} \times_1 \tilde{\mathbf{u}}_1^T = \left[ (1 - \rho_1^2) \lambda; \mathbf{a}^{(2)}, \dots, \mathbf{a}^{(N)} \right] \quad (5)$$

and

$$\begin{aligned} \mathcal{Y} \times_1 \tilde{\mathbf{u}}_1^T &= \left[ \text{diag} \left( \tilde{\mathbf{u}}_1^T \mathbf{B}^{(1)} \right) \boldsymbol{\beta}; \mathbf{B}^{(2)}, \dots, \mathbf{B}^{(N)} \right] \\ &= \left[ \boldsymbol{\gamma}_{\mathcal{R}} \circledast \boldsymbol{\beta}_{\mathcal{R}}; \mathbf{B}_{\mathcal{R}}^{(2)}, \dots, \mathbf{B}_{\mathcal{R}}^{(N)} \right], \end{aligned} \quad (6)$$

where  $\boldsymbol{\gamma} = \tilde{\mathbf{u}}_1^T \mathbf{B}^{(1)}$ ,  $\mathcal{R}$  denotes the set of indices such that  $\gamma_{\mathcal{R}} \neq 0$ , and  $\mathbf{B}_{\mathcal{R}}^{(n)} = \mathbf{B}^{(n)}(:, \mathcal{R})$  are taken columns of  $\mathbf{B}^{(n)}$ .

If  $\boldsymbol{\gamma}$  has at least two non-zero entries, then  $\mathbf{B}_{\mathcal{R}}^{(N)}$  is still a full column rank matrix, hence  $[[\boldsymbol{\gamma}_{\mathcal{R}} \circledast \boldsymbol{\beta}_{\mathcal{R}}; \mathbf{B}_{\mathcal{R}}^{(2)}, \dots, \mathbf{B}_{\mathcal{R}}^{(N)}]]$  has a unique CPD according to uniqueness condition in [30] and references therein. It follows that  $\mathcal{Y} \times_1 \tilde{\mathbf{u}}_1^T$  cannot be a rank-1 tensor as in (5). This implies that  $\boldsymbol{\gamma}$  has only one non-zero entry, say  $\gamma_r \neq 0$ . Thereby, we have

$$\left[ (1 - \rho_1^2) \lambda; \mathbf{a}^{(2)}, \dots, \mathbf{a}^{(N)} \right] = \left[ \gamma_r \beta_r; \mathbf{b}_r^{(2)}, \dots, \mathbf{b}_r^{(N)} \right], \quad (7)$$

which yields  $\mathbf{a}^{(n)} = \mathbf{b}_r^{(n)}$  for  $n = 2, \dots, N$ . Similarly, from the tensor-vector product  $\mathcal{Y} \times_2 \tilde{\mathbf{u}}_2^T$ , we obtain  $\mathbf{a}^{(1)} = \mathbf{b}_r^{(1)}$ .

Finally, because  $\gamma_r = \tilde{\mathbf{u}}_1^T \mathbf{b}_r^{(1)} = 1 - \rho_1^2$ , from (7) we obtain  $\lambda = \beta_r$ . The results also implies  $\mathcal{G}$  is a rank- $(R-1)$  tensor after eliminating the rank-1 tensor  $[[\gamma_r \beta_r; \mathbf{b}_r^{(2)}, \dots, \mathbf{b}_r^{(N)}]]$  from  $\mathcal{Y}$ . This completes the proof. ■

A similar result also holds for the decomposition  $\mathcal{Y} = [[\lambda; \mathbf{A}^{(1)}, \dots, \mathbf{A}^{(N)}]] + [[\mathcal{G}; \mathbf{U}^{(1)}, \dots, \mathbf{U}^{(N)}]]$ , which allows to deflate a rank- $K$  tensor from  $\mathcal{Y}$  instead of one rank-1 tensor, or sequential  $K$  rank-1 tensors.

*Theorem 2 (Rank- $K$  Deflation):* A rank- $R$  tensor  $\mathcal{Y} = [[\boldsymbol{\beta}; \mathbf{B}^{(1)}, \dots, \mathbf{B}^{(N)}]]$ , in which at least one factor matrix  $\mathbf{B}^{(n)}$  has a full column rank, has a decomposition  $\mathcal{Y} = [[\lambda; \mathbf{A}^{(1)}, \dots, \mathbf{A}^{(N)}]] + [[\mathcal{G}; \mathbf{U}^{(1)}, \dots, \mathbf{U}^{(N)}]]$ , where  $\mathbf{A}^{(n)} \in \mathbb{R}^{I_n \times K}$  and  $\mathbf{U}^{(n)} \in \mathbb{R}^{I_n \times (R-K)}$ . Then matrices  $\mathbf{A}^{(n)}$  comprise  $K$  components  $\mathbf{B}^{(n)}$ , and the tensor  $\mathcal{G}$  is of rank  $(R-K)$ .

Preliminary conditions for deflation of a rank-1 tensor from a high tensor are shown in Lemma 2 and Theorem 1. In general, a rank-1 tensor whose components do not lie in the column spaces of the other components can be considered an isolated rank-1 tensor, and thus can be extracted from the tensor. Note that the rank-1 tensor can still be extracted, when it interacts with other rank-1 tensors through no more than  $(N-2)$  modes.

### III. ALGORITHMS FOR RANK-1 TENSOR EXTRACTION

As mentioned earlier, the decomposition for rank-1 tensor deflation can be considered a rank- $(L_r, M_r, N_r)$  BCD which comprises only two blocks of rank-1 and multilinear rank- $(R-1, R-1, R-1)$ , respectively. Thereby, for three way tensors, one can apply the alternating least squares (ALS) algorithm for rank- $(L_r, M_r, N_r)$  BCD [28]. Alternatively, one can use the deflation algorithm in [20] derived for such particular tensor decomposition which sequentially updates  $\mathbf{a}^{(n)}$ ,  $\mathbf{U}^{(n)}$ ,  $\lambda$  and the core tensor  $\mathcal{G}$  through their closed-forms. This alternating minimisation algorithm [20] is simple to implement. However, it may encounter numerical issues involved in matrix inverse in updating  $\mathbf{U}^{(n)}$ . Moreover, computation of the cost function is relatively expensive, and the number of parameters is relatively large, mainly dominated by the core tensor  $\mathcal{G}$ . For a rank- $R$  tensor of size  $R \times R \times \dots \times R$ , the deflation algorithms based on the BCD need to estimate  $NR^2 + (R-1)^N$  parameters, while the purpose of tensor deflation is to extract only one rank-1 tensor requiring  $NR$  parameters.

In this section, we derive a new algorithm for the tensor deflation which acts as standard cyclic minimization algorithm, and has a lower complexity than existing algorithms for rank-1 plus multilinear rank- $(R-1, R-1, R-1)$  BCD, and algorithms for CPD. The main idea of the proposed algorithm is that we use the simplified least squares cost function after replacing the core tensor  $\mathcal{G}$  and the weight coefficient  $\lambda$  by their maximum likelihood estimates.

#### A. Closed-Form of $\lambda$ and Core Tensor $\mathcal{G}$

Let  $\mathcal{Y}_a = [[\lambda; \mathbf{a}^{(1)}, \dots, \mathbf{a}^{(N)}]]$  and  $\mathcal{Y}_u = [[\mathcal{G}; \mathbf{U}^{(1)}, \dots, \mathbf{U}^{(N)}]]$ . From (4), we consider the following cost function which minimizes the Frobenius norm between  $\mathcal{Y}$  and its estimate

$$\begin{aligned} D &= \frac{1}{2} \|\mathcal{Y} - \mathcal{Y}_a - \mathcal{Y}_u\|_F^2 \\ &= \frac{1}{2} \left\| \text{vec}(\mathcal{Y}) - \lambda \left( \bigotimes_n \mathbf{a}^{(n)} \right) - \left( \bigotimes_n \mathbf{U}^{(n)} \right) \text{vec}(\mathcal{G}) \right\|_F^2. \end{aligned} \quad (8)$$

In order to estimate  $\lambda$  and  $\mathcal{G}$ , we compute derivatives of  $D$  with respect to  $\lambda$  and  $g_{\mathbf{r}}$ ,  $\mathbf{r} = [r_1, r_2, \dots, r_N]$ ,  $1 \leq r_n \leq R$  as

$$\begin{aligned} \frac{\partial D}{\partial \lambda} &= - \left( \bigotimes_n \mathbf{a}^{(n)} \right)^T \text{vec}(\mathcal{Y}) + \left( \bigotimes_n \mathbf{a}^{(n)} \right)^T \left( \bigotimes_n \mathbf{a}^{(n)} \right) \lambda \\ &\quad + \left( \bigotimes_n \mathbf{a}^{(n)} \right)^T \left( \bigotimes_n \mathbf{U}^{(n)} \right) \text{vec}(\mathcal{G}) \\ &= -s + \lambda + \rho g_1, \end{aligned} \quad (9)$$

$$\begin{aligned} \frac{\partial D}{\partial g_{\mathbf{r}}} &= - \left( \bigotimes_n \mathbf{u}_{r_n}^{(n)} \right)^T \text{vec}(\mathcal{Y}) + \left( \bigotimes_n \mathbf{u}_{r_n}^{(n)} \right)^T \left( \bigotimes_n \mathbf{a}^{(n)} \right) \lambda + g_{\mathbf{r}} \\ &= -t_{\mathbf{r}} + \delta_{r,1} \lambda \rho + g_{\mathbf{r}} = 0, \end{aligned} \quad (10)$$

where  $r$  denotes the linear index of the sub-index  $\mathbf{r} = [r_1, r_2, \dots, r_N]$ ,  $r = 1, \dots, R_1 R_2 \dots R_N$ ,  $g_{\mathbf{r}}$  and  $t_{\mathbf{r}}$  are entries of the tensors  $\mathcal{G}$  and  $\mathcal{T}$  respectively,  $\delta_{r,1}$  is the Kronecker delta. Setting derivatives to zero will yield the optimum  $\lambda$  and  $\mathcal{G}$  as follows

$$\lambda = \frac{s - \rho t_1}{1 - \rho^2}, \quad \mathcal{G} = \mathcal{T} - \lambda \rho \mathcal{E}_1 \quad (11)$$

where  $\mathcal{T} = \mathcal{Y} \times \{\mathbf{U}^{(n)T}\}$ ,  $\mathcal{E}_1$  is a tensor of the same size as  $\mathcal{T}$  filled with zeros up to the first element, which is 1,  $\rho = \rho_1 \rho_2 \dots \rho_N$  and

$$s = \left( \bigotimes_n \mathbf{a}^{(n)} \right)^T \text{vec}(\mathcal{Y}). \quad (12)$$

By replacing the optimum  $\lambda$  and  $\mathcal{G}$ , the optimum value of the cost function can be expressed as

$$\begin{aligned} D &= \frac{1}{2} \left( \|\mathcal{Y}\|_F^2 + \|\mathcal{Y}_a\|_F^2 + \|\mathcal{Y}_u\|_F^2 - 2\langle \mathcal{Y}, \mathcal{Y}_a \rangle \right. \\ &\quad \left. - 2\langle \mathcal{Y}, \mathcal{Y}_u \rangle + 2\langle \mathcal{Y}_a, \mathcal{Y}_u \rangle \right) \\ &= \frac{1}{2} \left( \|\mathcal{Y}\|_F^2 + \lambda^2 + \|\mathcal{G}\|_F^2 - 2\lambda s - 2\langle \mathcal{T}, \mathcal{G} \rangle + 2\lambda \rho g_1 \right) \\ &= \frac{1}{2} \left( \|\mathcal{Y}\|_F^2 - \|\mathcal{T}\|_F^2 - \frac{(s - \rho t_1)^2}{1 - \rho^2} \right). \end{aligned} \quad (13)$$

### B. The Alternating Subspace Update Algorithm

In this subsection we consider the rank-one deflation for tensors of size  $R \times R \times \dots \times R$ . Tensors of larger and unequal sizes should be compressed to this size using the Tucker decomposition [31]–[33]. In this case we show that it is not necessary to estimate all  $R$  components  $\mathbf{a}^{(n)}$  and  $\mathbf{U}^{(n)}$  for each mode, but only update two components and one scalar parameter which is cosine of the angle between  $\mathbf{a}^{(n)}$  and  $\mathbf{u}_1^{(n)}$ . All parameters are found in closed-form through the eigenvalue decomposition. As the result, the cost function of the algorithm can be assessed with low complexity.

For notational simplicity, we put  $\mathbf{u}_n = \mathbf{u}_1^{(n)}$ , and define unit-norm vectors  $\mathbf{w}_n$  as

$$\mathbf{w}_n = \frac{\mathbf{a}^{(n)} - \rho_n \mathbf{u}_n}{\sqrt{1 - \rho_n^2}}, \quad (14)$$

provided that  $\rho_n = \mathbf{u}_n^T \mathbf{a}^{(n)} \neq 1$ . We perform a change of parameters so that instead of estimating  $\mathbf{a}^{(n)}$ ,  $\mathbf{U}^{(n)}$  and  $\lambda$ ,  $\mathcal{G}$ , we

derive update rules for  $\mathbf{w}_n$ ,  $\mathbf{u}_n$  and  $\rho_n$  for  $n = 1, \dots, N$ . The updates are then cyclically iterated in the form summarized as Alternating Subspace Update (ASU)—Algorithm 2.

Once  $\mathbf{w}_n$ ,  $\mathbf{u}_n$  are estimated, the remaining  $(R - 2)$  columns of  $\mathbf{U}^{(n)}$  can be set by an arbitrary orthogonal complement to  $[\mathbf{w}_n, \mathbf{u}_n]$ . Often we even need not compute the factor matrices  $\mathbf{U}^{(n)}$  explicitly except its first component  $\mathbf{u}_n$ , because  $\mathbf{U}^{(n)}$  appear only in products  $\mathbf{U}^{(n)} \mathbf{U}^{(n)T}$  which can be computed as  $\mathbf{U}^{(n)} \mathbf{U}^{(n)T} = \mathbf{I}_R - \mathbf{w}_n \mathbf{w}_n^T$ . See more discussion in Section III-C.

Finally, when  $\mathbf{w}_n$ ,  $\mathbf{u}_n$  and  $\rho_n$  are given, the components of the rank-1 tensor are computed as

$$\mathbf{a}^{(n)} = \rho_n \mathbf{u}_n + \sqrt{1 - \rho_n^2} \mathbf{w}_n. \quad (15)$$

1) *Closed-Form Update for  $\rho_n$* : Assuming that  $\rho_{-n} \neq 1$ , put

$$\mathbf{d}_n = \frac{\mathbf{s}_n - \rho_{-n} \mathbf{z}_n}{\sqrt{1 - \rho_{-n}^2}}, \quad (16)$$

where

$$\mathbf{s}_n = \mathbf{Y}_{(n)} \left( \bigotimes_{k \neq n} \mathbf{a}^{(k)} \right), \quad \mathbf{z}_n = \mathcal{Y} \times_{k \neq n} \{\mathbf{u}_k^T\}. \quad (17)$$

Taking into account that  $\rho = \rho_n \rho_{-n}$ , where  $\rho_{-n} = \prod_{k \neq n} \rho_k$  and

$$s = \mathbf{a}^{(n)T} \mathbf{s}_n = \sqrt{1 - \rho_n^2} \mathbf{w}_n^T \mathbf{s}_n + \rho_n \mathbf{u}_n^T \mathbf{s}_n \quad (18)$$

$$t_1 = \mathbf{u}_n^T \mathbf{z}_n, \quad (19)$$

we obtain the following Cauchy-Schwarz inequality<sup>1</sup>

$$\begin{aligned} (s - \rho t_1)^2 &= \left( \sqrt{1 - \rho_n^2} (\mathbf{w}_n^T \mathbf{s}_n) + \left( \rho_n \sqrt{1 - \rho_{-n}^2} \right) (\mathbf{u}_n^T \mathbf{d}_n) \right)^2 \\ &\leq \left( 1 - \rho_n^2 + \left( \rho_n \sqrt{1 - \rho_{-n}^2} \right)^2 \right) \\ &\quad \times \left( (\mathbf{w}_n^T \mathbf{s}_n)^2 + (\mathbf{u}_n^T \mathbf{d}_n)^2 \right) \\ &= (1 - \rho^2) \left( (\mathbf{w}_n^T \mathbf{s}_n)^2 + (\mathbf{u}_n^T \mathbf{d}_n)^2 \right). \end{aligned} \quad (20)$$

Thereby, the cost function satisfies the following inequality

$$\begin{aligned} D &= \frac{1}{2} \left( \|\mathcal{Y}\|_F^2 - \|\mathcal{T}\|_F^2 - \frac{(s - \rho t_1)^2}{1 - \rho^2} \right) \\ &\geq \frac{1}{2} \left( \|\mathcal{Y}\|_F^2 - \|\mathcal{T}\|_F^2 - (\mathbf{w}_n^T \mathbf{s}_n)^2 - (\mathbf{u}_n^T \mathbf{d}_n)^2 \right). \end{aligned} \quad (21)$$

The equality holds when

$$\sqrt{1 - \rho_n^2} : \rho_n \sqrt{1 - \rho_{-n}^2} = \mathbf{w}_n^T \mathbf{s}_n : \mathbf{u}_n^T \mathbf{d}_n \quad (22)$$

indicating that the optimum  $\rho_n$  when keeping other parameters fixed is given in closed-form solution as

$$\rho_n = \frac{|\mathbf{u}_n^T \mathbf{d}_n|}{\sqrt{(\mathbf{w}_n^T \mathbf{s}_n)^2 (1 - \rho_{-n}^2) + (\mathbf{u}_n^T \mathbf{d}_n)^2}}. \quad (23)$$

<sup>1</sup> $(ac + bd)^2 \leq (a^2 + b^2)(c^2 + d^2)$

2) *Closed-Form Expressions for  $\mathbf{w}_n$  and  $\mathbf{u}_n$* : Let

$$\mathbf{Z}_n = \mathbf{Y}_{(n)} \left( \bigotimes_{k \neq n} \mathbf{U}^{(k)} \right). \quad (24)$$

Note that  $\mathbf{z}_n$  defined in (17) is the first column of  $\mathbf{Z}_n$ . Since we assumed that the tensor  $\mathcal{Y}$  has the size of  $R \times R \times \dots \times R$ , it follows that  $\mathbf{U}^{(n)}$  are of size  $R \times (R-1)$  for  $n = 1, \dots, N$ , and

$$\mathbf{U}^{(n)} \mathbf{U}^{(n)T} = \mathbf{I}_R - \mathbf{w}_n \mathbf{w}_n^T, \quad (25)$$

$$\begin{aligned} \|\mathcal{T}\|_F^2 &= \left\| \mathbf{U}^{(n)T} \mathbf{Z}_n \right\|_F^2 = \text{tr} \left( \mathbf{Z}_n^T (\mathbf{I} - \mathbf{w}_n \mathbf{w}_n^T) \mathbf{Z}_n \right) \\ &= \text{tr}(\Phi_n) - \mathbf{w}_n^T \Phi_n \mathbf{w}_n. \end{aligned} \quad (26)$$

With the optimum  $\rho_n$  in (23), the cost function (13), i.e. the right-hand side of (21), is now rewritten as

$$\begin{aligned} D &= \frac{1}{2} \left( \|\mathcal{Y}\|_F^2 - \text{tr}(\Phi_n) + \mathbf{w}_n^T \Phi_n \mathbf{w}_n - (\mathbf{w}_n^T \mathbf{s}_n)^2 - (\mathbf{u}_n^T \mathbf{d}_n)^2 \right) \\ &= \frac{1}{2} \left( \|\mathcal{Y}\|_F^2 - \text{tr}(\Phi_n) - \mathbf{d}_n^T \mathbf{d}_n + \mathbf{w}_n^T (\Phi_n - \mathbf{s}_n \mathbf{s}_n^T + \mathbf{d}_n \mathbf{d}_n^T) \mathbf{w}_n \right. \\ &\quad \left. + \mathbf{d}_n^T (\mathbf{I}_R - \mathbf{w}_n \mathbf{w}_n^T - \mathbf{u}_n \mathbf{u}_n^T) \mathbf{d}_n \right). \end{aligned} \quad (27)$$

The two vectors  $\mathbf{w}_n$  and  $\mathbf{u}_n$  only involve in two quadratic terms  $\mathbf{w}_n^T (\Phi_n - \mathbf{s}_n \mathbf{s}_n^T + \mathbf{d}_n \mathbf{d}_n^T) \mathbf{w}_n$  and  $\mathbf{d}_n^T (\mathbf{I}_R - \mathbf{w}_n \mathbf{w}_n^T - \mathbf{u}_n \mathbf{u}_n^T) \mathbf{d}_n$ . Since the matrix  $(\mathbf{I}_R - \mathbf{w}_n \mathbf{w}_n^T - \mathbf{u}_n \mathbf{u}_n^T)$  is positive-semidefinite<sup>2</sup>

$$\mathbf{d}_n^T (\mathbf{I}_R - \mathbf{w}_n \mathbf{w}_n^T - \mathbf{u}_n \mathbf{u}_n^T) \mathbf{d}_n \geq 0, \quad (28)$$

from (27), the following inequality for the cost function holds

$$D \geq \frac{1}{2} \left( \|\mathcal{Y}\|_F^2 - \text{tr}(\Phi_n) - \mathbf{d}_n^T \mathbf{d}_n + \sigma_n \right), \quad (29)$$

where  $\sigma_n$  is the smallest eigenvalue of  $(\Phi_n - \mathbf{s}_n \mathbf{s}_n^T + \mathbf{d}_n \mathbf{d}_n^T)$ . The equality in (29) holds when

- 1)  $\mathbf{w}_n$  is the eigenvector associated with the smallest eigenvalue  $\sigma_n$  of the matrix  $(\Phi_n - \mathbf{s}_n \mathbf{s}_n^T + \mathbf{d}_n \mathbf{d}_n^T)$ .
- 2)  $\mathbf{d}_n^T (\mathbf{I}_R - \mathbf{w}_n \mathbf{w}_n^T - \mathbf{u}_n \mathbf{u}_n^T) \mathbf{d}_n = 0$ .  
That is  $(\mathbf{d}_n^T \mathbf{u}_n)^2 = \mathbf{d}_n^T \mathbf{d}_n - (\mathbf{d}_n^T \mathbf{w}_n)^2$  while  $\mathbf{u}_n^T \mathbf{w}_n = 0$ . It turns out that  $\mathbf{u}_n$  takes the following form

$$\mathbf{u}_n = \frac{\mathbf{d}_n - \mathbf{w}_n (\mathbf{w}_n^T \mathbf{d}_n)}{\sqrt{\mathbf{d}_n^T \mathbf{d}_n - (\mathbf{w}_n^T \mathbf{d}_n)^2}}. \quad (30)$$

Finally, while fixing parameters  $\mathbf{w}_k$ ,  $\mathbf{u}_k$  and  $\rho_k$  for  $k \neq n$ ,  $\mathbf{w}_n$ ,  $\mathbf{u}_n$  and  $\rho_n$  are updated in closed-form. The updates (23) and (30) form the core of the proposed Alternating Subspace Update (ASU) algorithm, which is summarized as Algorithm 2. The core tensor  $\mathcal{G}$  and weight  $\lambda$  need to be computed only once. The ASU acts as a standard cyclic minimization algorithm [34], [35], which partitions the variables into groups and minimizes the problem with respect to a group of parameters while keeping the other parameters fixed. More specifically, the ASU algorithm estimates parameters including two vectors  $\mathbf{w}_n$ ,  $\mathbf{u}_n$  and a scalar  $\rho_n$  for each mode  $n$ , whereas minimization of sub-problems is done in closed-form. As a result, this cyclic minimizer generates a monotonically nonincreasing sequence of the cost values bounded by 0. Such cyclic minimization algorithm must converge at least to local minima as discussed in [36].

<sup>2</sup>The  $R \times R$  matrix  $\mathbf{I}_R - \mathbf{w}_n \mathbf{w}_n^T - \mathbf{u}_n \mathbf{u}_n^T = \mathbf{U}_{2:R-1}^{(n)} \mathbf{U}_{2:R-1}^{(n)T}$  has  $(R-2)$  eigenvalues of 1, and 2 eigenvalues of zero.

---

### Algorithm 1: Sequential Rank-1 Reduction

---

**Input:** Data tensor  $\mathcal{Y}$ :  $(I_1 \times I_2 \times \dots \times I_N)$ , rank  $R$   
**Output:** A rank- $R$  Kruskal tensor  $\mathcal{X}$   
**begin**  
 % **Stage 1:** Optional Tucker compression of  $\mathcal{Y}$ -----  
 1  $\llbracket \mathcal{G}_0; \{\mathbf{B}^{(n)}\} \rrbracket = \text{Tuckercompression}(\mathcal{Y}, [R, R, \dots, R])$   
 % **Stage 2:** Sequential rank-1 tensor extraction---  
 2  $\mathbf{V}^{(n)} = \mathbf{I}_R$  for  $n = 1, 2, \dots, N$   
**for**  $r = 1, 2, \dots, R-1$  **do**  
 3  $\llbracket \{\lambda_r; \{\mathbf{a}_r^{(n)}\}\rrbracket, \llbracket \mathcal{G}_r; \{\mathbf{U}_r^{(n)}\} \rrbracket = \text{ASU}(\mathcal{G}_{r-1})$   
 4 **for**  $n = 1, \dots, N$  **do**  $\mathbf{a}_r^{(n)} \leftarrow \mathbf{V}^{(n)} \mathbf{a}_r^{(n)}$ ,  $\mathbf{V}^{(n)} \leftarrow \mathbf{V}^{(n)} \mathbf{U}^{(n)}$   
 5  $\mathcal{X} = \llbracket \lambda; \{\mathbf{B}^{(n)} \mathbf{A}^{(n)}\} \rrbracket$   


---

 $\lambda = [\lambda_1, \dots, \lambda_R]$ ,  $\mathbf{A}^{(n)} = [\mathbf{a}_1^{(n)}, \dots, \mathbf{a}_R^{(n)}]$ ,  $\lambda_R \triangleq \mathcal{G}_{R-1}$ ,  $\mathbf{a}_R^{(n)} \triangleq \mathbf{U}_{R-1}^{(n)}$

---



---

### Algorithm 2: Alternating Subspace Update (ASU) for Rank-1 Tensor Deflation

---

**Input:** Data tensor  $\mathcal{Y}$ :  $(R \times R \times \dots \times R)$ , rank  $R$   
**Output:** A rank-1 tensor  $\llbracket \lambda; \{\mathbf{a}^{(n)}\} \rrbracket$  and multilinear rank- $(R-1)$  tensor  $\llbracket \mathcal{G}; \{\mathbf{U}^{(n)}\} \rrbracket$   
**begin**  
 1 Initialise components  $\mathbf{a}^{(n)}$  and  $\mathbf{U}^{(n)}$   
**repeat**  
 2 **for**  $n = 1, 2, \dots, N$  **do**  
 3  $[\mathbf{w}_n, \sigma] = \text{eigs}(\Phi_n - \mathbf{s}_n \mathbf{s}_n^T + \mathbf{d}_n \mathbf{d}_n^T, 1, \text{'SA'})$   
 4  $\mathbf{u}_n = \frac{\mathbf{d}_n - \mathbf{w}_n (\mathbf{w}_n^T \mathbf{d}_n)}{\sqrt{\mathbf{d}_n^T \mathbf{d}_n - (\mathbf{w}_n^T \mathbf{d}_n)^2}}$   
 5  $\rho_n = \frac{|\mathbf{u}_n^T \mathbf{d}_n|}{\sqrt{(\mathbf{w}_n^T \mathbf{s}_n)^2 (1 - \rho_{-n}^2) + (\mathbf{u}_n^T \mathbf{d}_n)^2}}$  as in (23)  
 6  $\mathbf{a}^{(n)} \leftarrow \mathbf{w}_n \sqrt{1 - \rho_{-n}^2} + \mathbf{u}_n \rho_n$   
**until** a stopping criterion is met  
 7 **for**  $n = 1, \dots, N$  **do**  
 8 Select  $\mathbf{U}_{2:R-1}^{(n)}$  as an orthogonal complement of  $[\mathbf{w}_n, \mathbf{u}_n]$   
 9  $\lambda = \frac{\mathcal{Y} \times \{\mathbf{a}^{(n)T}\} - \rho \mathcal{Y} \times \{\mathbf{u}^{(n)T}\}}{1 - \rho^2}$ ,  $\mathcal{G} = \mathcal{Y} \times \{\mathbf{U}^{(n)T}\} - \lambda \rho \mathcal{E}_1$

---

eigs(C, 1, 'SA'): compute eigenvector associated with the smallest eigenvalue of C.  
 $\mathbf{s}_n$ ,  $\mathbf{d}_n$  and  $\mathcal{E}_1$  are defined in (17), (16) and (11), respectively.  
 $\Phi_n$  are computed as in (31) and (32).

---

3) *The Case When  $\rho_{-n} = 1$* : When  $\rho_{-n} = 1$ , it implies that  $\rho_k = 1$ , i.e.,  $\mathbf{a}^{(k)} = \mathbf{u}_k$  for all  $k \neq n$ , and  $\mathbf{s}_n = \mathbf{z}_n$ . Thereby,

$$\begin{aligned} s - \rho t_1 &= \sqrt{1 - \rho_n^2} \mathbf{w}_n^T \mathbf{s}_n + \rho_n \mathbf{u}_n^T \mathbf{z}_n - \rho_n \rho_{-n} \mathbf{u}_n^T \mathbf{z}_n \\ &= \sqrt{1 - \rho_n^2} \mathbf{w}_n^T \mathbf{s}_n, \end{aligned}$$

and the cost function in (13) becomes independent of the choice of the vector  $\mathbf{u}_n$  and parameter  $\rho_n$

$$D = \frac{1}{2} \left( \|\mathcal{Y}\|_F^2 - \text{tr}(\Phi_n) + \mathbf{w}_n^T \Phi_n \mathbf{w}_n - (\mathbf{w}_n^T \mathbf{s}_n)^2 \right).$$

For such case, the tensor may violate the condition c in Lemma 2.

### C. Complexity of the ASU Algorithm

We will show that the ASU algorithm has a lower complexity than the fastest existing algorithm for CPD. According to Algorithm 2, for order-3 tensor, computation of  $\mathbf{s}_n$ ,  $\mathbf{z}_n$  and eigenvalue decomposition of  $\Phi_n - \mathbf{s}_n \mathbf{s}_n^T + \mathbf{d}_n \mathbf{d}_n^T$  to update  $\mathbf{w}_n$  is of complexity  $\mathcal{O}(R^3)$ . The most expensive step in ASU is to compute  $\Phi_n = \mathbf{Z}_n \mathbf{Z}_n^T$ .

Let  $\mathbf{k} = [k_1, k_2, \dots, k_d]$  for  $1 \leq d \leq N$ ,  $\mathbf{w}_{\mathbf{k}} = [\mathbf{w}_{k_1}, \mathbf{w}_{k_2}, \dots, \mathbf{w}_{k_d}]$  and  $\mathcal{T}_{\mathbf{k}} = \mathcal{Y} \times_{\mathbf{k}} \{\mathbf{w}_{\mathbf{k}}^T\}$ , the matrix  $\Phi_1 = \mathbf{Y}_{(1)} (\bigotimes_{n \neq 1} \mathbf{U}^{(n)} \mathbf{U}^{(n)T}) \mathbf{Y}_{(1)}^T$  can be computed as

$$\Phi_1 = \mathbf{Y}_{(1)} \mathbf{Y}_{(1)}^T + \sum_{d=1}^{N-1} (-1)^d \sum_{\substack{\mathbf{k}=[k_1 < \dots < k_d] \\ k_j \neq 1}} \langle \mathcal{T}_{\mathbf{k}}, \mathcal{T}_{\mathbf{k}} \rangle_{-1}. \quad (31)$$

For example, for  $N = 3$ , we compute  $\Phi_1$  as

$$\begin{aligned} \Phi_1 &= \mathbf{Y}_{(1)} \mathbf{Y}_{(1)}^T + (\mathcal{Y} \times_2 \mathbf{w}_2^T \times_3 \mathbf{w}_3^T) (\mathcal{Y} \times_2 \mathbf{w}_2^T \times_3 \mathbf{w}_3^T)^T \\ &\quad - (\mathcal{Y} \times_2 \mathbf{w}_2^T) (\mathcal{Y} \times_2 \mathbf{w}_2^T)^T - (\mathcal{Y} \times_3 \mathbf{w}_3^T) (\mathcal{Y} \times_3 \mathbf{w}_3^T)^T. \end{aligned} \quad (32)$$

The tensor product  $\mathcal{Y} \times_2 \mathbf{w}_2^T \times_3 \mathbf{w}_3^T$  returns a vector of size  $R$ , whereas products  $\mathcal{Y} \times_2 \mathbf{w}_2^T$  or  $\mathcal{Y} \times_3 \mathbf{w}_3^T$  yield matrices of size  $R \times R$ . Similarly, the other matrices  $\Phi_n$  can be computed, without evaluating  $\mathbf{U}^{(n)}$ , with complexity of  $\mathcal{O}(R^3)$ , while  $\mathbf{Y}_{(n)} \mathbf{Y}_{(n)}^T$  is computed once. Finally, the total expense of ASU per iteration to update  $\mathbf{w}_n$ ,  $\mathbf{u}_n$  and  $\mathbf{a}^{(n)}$  is approximately  $\mathcal{O}(R^3)$ . It is worth noting that the fastest algorithm for CPD has a complexity per iteration of  $\mathcal{O}(RI_1I_2I_3) = \mathcal{O}(R^4)$  [37] for estimation of  $R$  rank-1 tensors. Hence, extraction of a rank-1 tensor requires a cost of  $\mathcal{O}(R^3)$ .

For higher order tensors, construction of  $\Phi_n = \mathbf{Z}_n \mathbf{Z}_n^T$  is still the most expensive step. According to (31), the computational cost is dominated by tensor products  $\mathcal{Y} \times_{\mathbf{k}} \mathbf{w}_{\mathbf{k}}^T$  for  $\mathbf{k} \neq n$ . Taking account that we only need to compute one tensor-vector product for each  $n$ , whereas  $(N-2)$  other products are taken from the previous construction of  $\Phi_{n-1}$  or  $\Phi_N$ . Therefore, the computational cost of ASU is of  $\mathcal{O}(R^N)$ .

#### IV. RANK IN TENSOR DEFLATION

Deflation of a rank-1 tensor out of a rank- $R$  tensor requires the tensor rank  $R$  as prior information. This section will discuss how to apply tensor deflation when the rank is not given. A simple way is that one can apply the deflation with sufficient high rank (overestimated), which is higher than the expected true rank of the data, but does not exceed the tensor dimensions. For example, in practice, the initial rank can be set to its dimension, i.e.,  $R = \min\{I_1, I_2, \dots, I_N\}$ , or multilinear rank of the data, i.e.,  $R = \min\{\text{rank}(\mathbf{Y}_{(1)}), \dots, \text{rank}(\mathbf{Y}_{(N)})\}$ .

*Lemma 3 (Deflation With Overestimated Rank):* Given a data tensor  $\mathcal{Y}$  whose rank  $R$  does not exceed its dimensions, a decomposition of this tensor into a rank-1 plus multilinear rank- $(L)$  tensors with  $L \geq R$  yields a non-unique solution with  $\mathbf{a}^{(n)T} \mathbf{u}_n = 1$ .

*Proof:* The proof can be seen from condition c in Lemma 2. The multilinear rank- $(L)$  tensor can explain the data tensor with a perfect fit, and the first rank-1 tensor can be chosen arbitrarily. In addition, since components  $\mathbf{a}^{(n)}$  lie in subspaces of mode- $n$  matriculations spanned by factor matrices  $\mathbf{U}^{(n)}$ , one has  $\mathbf{a}^{(n)T} \mathbf{u}_n = 1$ . ■

In practice, tensor may not be of exact rank- $R$  but can be approximated by a rank- $R$  Kruskal tensor. A deflation with rank higher than  $R$  will yield at least  $(N-1)$  rotational parameters close to one, i.e.,  $\rho_n \approx 1$ . Based on this observation, we can verify whether any rank-1 tensor in the CP decomposition or tensor deflation violates the uniqueness condition, and reduce rank of the further decomposition. More specifically, for

CPD, we check  $\rho_{n,r} = \|\mathbf{V}_r^{(n)T} \mathbf{a}_r^{(n)}\|_2$ , where  $\mathbf{V}_r^{(n)}$  are orthonormal basis of matrices  $\mathbf{A}^{(n)}$  excluding the column  $\mathbf{a}_r^{(n)}$  for  $n = 1, \dots, N$  and  $r = 1, \dots, R$ .

Another observation is that components of the rank-1 tensor (the first block) in the deflation with overestimated rank may not explain the data. The fit explained by the decomposition is mainly due to the second block. Hence, together with verifying  $\rho_{n,r}$ , identification of rank-1 tensors which do not explain the tensor will help us to reduce the deflation rank to be close to the tensor rank, especially for noise-free data or data with relatively low noise, e.g.,  $\text{SNR} \geq 30$  dB. The procedure will be illustrated in Example 4 in the Simulation section.

Finally, when the rank used in the deflation procedure is close to the true rank, practical experiments show that the tensor deflation can extract desired rank-1 tensors with high accuracy. This behavior of the ASU algorithm and tensor deflation procedure is illustrated in Examples 4 and 5 in the Simulation section. When the noise is high or the data tensor is not well approximated by a rank- $R$  tensor, a deflation with slightly higher rank is recommended.

#### V. DECOMPOSITION OF A COMPLEX-VALUED TENSOR

This section derives an ASU algorithm for complex-valued tensor deflation. The model is similar to (4), but  $\lambda$  is a real-valued coefficient, while the other parameters are complex-valued, i.e.,  $\mathbf{a}^{(n)} \in \mathbb{C}^R$ ,  $\mathbf{U}^{(n)} \in \mathbb{C}^{R \times (R-1)}$  and  $\mathcal{G}$ . Similarly to deflation of a real-valued tensor,  $\mathbf{a}^{(n)}$  and  $\mathbf{U}^{(n)}$  can also be normalized to be orthogonal, i.e.,  $(\mathbf{U}^{(n)})^H \mathbf{U}^{(n)} = \mathbf{I}_{R-1}$ ,  $(\mathbf{U}_{2:(R-1)}^{(n)})^H \mathbf{a}^{(n)} = \mathbf{0}_{R-2}$ , and  $0 \leq \mathbf{u}_n^H \mathbf{a}^{(n)} \leq 1$ . The cost function for the complex-valued tensor deflation is given by

$$\begin{aligned} D &= \frac{1}{2} \left\| \mathcal{Y} - \lambda \left[ \left\{ \mathbf{a}^{(n)} \right\} \right] - \left[ \mathcal{G}; \left\{ \mathbf{U}^{(n)} \right\} \right] \right\|_F^2 \\ &= \frac{1}{2} \left\| \text{vec}(\mathcal{Y}) - \left( \bigotimes \left\{ \mathbf{a}^{(n)} \right\} \right) \lambda - \left( \bigotimes_n \mathbf{U}^{(n)} \right) \text{vec}(\mathcal{G}) \right\|_F^2. \end{aligned} \quad (33)$$

Again, instead of estimating  $\mathbf{a}^{(n)}$ ,  $\mathbf{U}^{(n)}$ ,  $\lambda$  and  $\mathcal{G}$  explicitly, we perform a reparameterization with  $\mathbf{a}^{(n)} = \sqrt{1 - \rho_n^2} \mathbf{w}_n + \rho_n \mathbf{u}_n$  where  $\rho_n = \mathbf{u}_n^H \mathbf{a}^{(n)}$ , and  $[\mathbf{w}_n, \mathbf{U}^{(n)}]$  are  $R \times R$  unitary matrices. The vectors of  $[\mathbf{w}_n, \mathbf{u}_n]$  and positive parameters  $\rho_n$  can be estimated in closed-form, whereas  $(R-2)$  columns  $[\mathbf{u}_2^{(n)}, \dots, \mathbf{u}_{R-1}^{(n)}]$  can take arbitrary orthonormal basis for the orthogonal complement to  $[\mathbf{w}_n, \mathbf{u}_n]$ . To this end, we will first derive closed-form expressions for  $\lambda$  and the core tensor  $\mathcal{G}$ , then rewrite the cost function (33) in term of  $\mathbf{w}_n$ ,  $\mathbf{u}_n$  and  $\rho_n$ .

##### A. Closed-Form Expressions for $\lambda$ and $\mathcal{G}$

Let  $\rho = \prod_n \rho_n$ , then gradients of the cost function with respect to  $\lambda$  and the complex conjugate  $\mathcal{G}^*$  can be expressed as

$$\begin{aligned} \frac{\partial D}{\partial \lambda} &= -\text{Re} \left\{ \left( \bigotimes \left\{ \mathbf{a}^{(n)} \right\} \right)^H \text{vec}(\mathcal{Y}) \right\} + \rho \text{Re}\{g_1\} + \lambda \\ &= -\text{Re}\{s\} + \rho \text{Re}\{g_1\} + \lambda, \end{aligned} \quad (34)$$

$$\begin{aligned} \frac{\partial D}{\partial \mathcal{G}^*} &= \frac{1}{2} \left( \frac{\partial D}{\partial \text{Re}\{\mathcal{G}\}} + i \frac{\partial D}{\partial \text{Im}\{\mathcal{G}\}} \right) \\ &= \frac{1}{2} \left( -\mathcal{Y} \times \left\{ \mathbf{U}^{(n)H} \right\} + \lambda \rho \mathbf{E}_1 + \mathcal{G} \right) \end{aligned} \quad (35)$$

where  $s = \mathcal{Y} \times \{\mathbf{a}^{(n)H}\}$ , and  $\mathcal{E}_1$  is a tensor whose first entry is 1 and others are zeros. By setting the gradients to zeros we obtain closed-form expressions for  $\lambda$  and  $\mathcal{G}$

$$\lambda = \frac{\text{Re}\{s\} - \rho \text{Re}\{t_1\}}{1 - \rho^2}, \quad \mathcal{G} = \mathcal{T} - \lambda \rho \mathcal{E}_1 \quad (36)$$

where  $\mathcal{T} = \mathcal{Y} \times \{\mathbf{U}^{(n)H}\}$  and  $t_1$  is the first entry of  $\mathcal{T}$ .

### B. The ASU Algorithm

Before we derive the closed-form updates for  $\mathbf{w}_n$ ,  $\mathbf{u}_n$  and  $\rho_n$ , we introduce additional notations

$$\tilde{\mathbf{x}} = \begin{bmatrix} \text{Re}\{\mathbf{x}\} \\ \text{Im}\{\mathbf{x}\} \end{bmatrix}, \quad \check{\mathbf{x}} = \begin{bmatrix} -\text{Im}\{\mathbf{x}\} \\ \text{Re}\{\mathbf{x}\} \end{bmatrix}. \quad (37)$$

The orthogonality constraints on  $\mathbf{w}_n$  and  $\mathbf{u}_n$  lead to three real-valued vectors  $\tilde{\mathbf{u}}_n$ ,  $\tilde{\mathbf{w}}_n$ ,  $\check{\mathbf{w}}_n$  that are orthonormal, i.e.,  $[\tilde{\mathbf{u}}_n, \tilde{\mathbf{w}}_n, \check{\mathbf{w}}_n]^T [\tilde{\mathbf{u}}_n, \tilde{\mathbf{w}}_n, \check{\mathbf{w}}_n] = \mathbf{I}_3$ . As in (16), we define  $\mathbf{d}_n = (\mathbf{s}_n - \rho_{-n} \mathbf{z}_n) / \sqrt{1 - \rho_{-n}^2}$ , where  $\mathbf{s}_n = \mathcal{Y} \times_{-n} \{\mathbf{a}^H\}$ ,  $\mathbf{z}_n = \mathcal{Y} \times_{-n} \{\mathbf{u}^H\}$ .

Using the above expressions for  $\lambda$  and  $\mathcal{G}$ , and taking into account that

$$\begin{aligned} \|\mathcal{G}\|_F^2 &= \|\mathcal{T}\|_F^2 + \lambda^2 \rho^2 - 2\lambda \rho \text{Re}\{t_1\}, \\ \|\mathcal{T}\|_F^2 &= \|\mathbf{U}_n \mathbf{Z}_n\|_F^2 = \text{tr}(\Phi_n) - \mathbf{w}_n^H \Phi_n \mathbf{w}_n \end{aligned}$$

where  $\Phi_n = \mathbf{Z}_n \mathbf{Z}_n^H$ , we can rewrite the cost function (33) as a function of parameters  $\mathbf{w}_n$ ,  $\mathbf{U}^{(n)}$  and  $\rho_n$

$$\begin{aligned} D &= \frac{1}{2} \left( \|\mathcal{Y}\|_F^2 + \|\mathcal{Y}_a\|_F^2 + \|\mathcal{Y}_u\|_F^2 - 2\text{Re}\{\langle \mathcal{Y}, \mathcal{Y}_a \rangle\} \right. \\ &\quad \left. - 2\text{Re}\{\langle \mathcal{Y}, \mathcal{Y}_u \rangle\} + 2\text{Re}\{\langle \mathcal{Y}_a, \mathcal{Y}_u \rangle\} \right) \\ &= \frac{1}{2} \left( \|\mathcal{Y}\|_F^2 - \|\mathcal{T}\|_F^2 + \lambda^2 (1 - \rho^2) 2\lambda \text{Re}\{s\} + 2\lambda \rho \text{Re}\{t_1\} \right) \\ &= \frac{1}{2} \left( \|\mathcal{Y}\|_F^2 - \text{tr}(\Phi_n) + \mathbf{w}_n^H \Phi_n \mathbf{w}_n - \frac{(\text{Re}\{s - \rho t_1\})^2}{1 - \rho^2} \right). \end{aligned} \quad (38)$$

By the Cauchy-Schwarz inequality, the following inequality holds

$$\begin{aligned} (\text{Re}\{s - \rho t_1\})^2 &= \left( \sqrt{1 - \rho^2} \text{Re}\{\mathbf{w}_n^H \mathbf{s}_n\} \right. \\ &\quad \left. + \rho_n \sqrt{1 - \rho_{-n}^2} \text{Re}\{\mathbf{u}_n^H \mathbf{d}_n\} \right)^2 \\ &\leq (1 - \rho^2) \left( \text{Re}\{\mathbf{w}_n^H \mathbf{s}_n\}^2 + \text{Re}\{\mathbf{u}_n^H \mathbf{d}_n\}^2 \right) \\ &= (1 - \rho^2) \left( \left( \tilde{\mathbf{w}}_n^T \tilde{\mathbf{s}}_n \right)^2 + \left( \tilde{\mathbf{u}}_n^T \tilde{\mathbf{d}}_n \right)^2 \right). \end{aligned} \quad (39)$$

This leads to the following inequality

$$\begin{aligned} D &\geq \frac{1}{2} \left( \|\mathcal{Y}\|_F^2 - \text{tr}(\Phi_n) + \mathbf{w}_n^H \Phi_n \mathbf{w}_n - \left( \tilde{\mathbf{w}}_n^T \tilde{\mathbf{s}}_n \right)^2 - \left( \tilde{\mathbf{u}}_n^T \tilde{\mathbf{d}}_n \right)^2 \right) \\ &= \frac{1}{2} \left( \|\mathcal{Y}\|_F^2 - \text{tr}(\Phi_n) + \tilde{\mathbf{w}}_n^T \left( \Phi_n - \tilde{\mathbf{s}}_n \tilde{\mathbf{s}}_n^T + \tilde{\mathbf{d}}_n \tilde{\mathbf{d}}_n^T + \check{\mathbf{d}}_n \check{\mathbf{d}}_n^T \right) \tilde{\mathbf{w}}_n \right. \\ &\quad \left. + \tilde{\mathbf{d}}_n^T \left( \mathbf{I}_{2R} - \tilde{\mathbf{w}}_n \tilde{\mathbf{w}}_n^T - \check{\mathbf{w}}_n \check{\mathbf{w}}_n^T - \tilde{\mathbf{u}}_n \tilde{\mathbf{u}}_n^T \right) \tilde{\mathbf{d}}_n - \tilde{\mathbf{d}}_n^T \check{\mathbf{d}}_n \right) \\ &\geq \frac{1}{2} \left( \|\mathcal{Y}\|_F^2 - \text{tr}(\Phi_n) + \sigma_n - \mathbf{d}_n^H \mathbf{d}_n \right) \end{aligned} \quad (40)$$

where  $\tilde{\Phi}_n = \begin{bmatrix} \text{Re}\{\Phi_n\} & -\text{Im}\{\Phi_n\} \\ \text{Im}\{\Phi_n\} & \text{Re}\{\Phi_n\} \end{bmatrix}$  and  $\sigma_n$  is the smallest eigenvalue of the matrix  $(\tilde{\Phi}_n - \tilde{\mathbf{s}}_n \tilde{\mathbf{s}}_n^T + \tilde{\mathbf{d}}_n \tilde{\mathbf{d}}_n^T + \check{\mathbf{d}}_n \check{\mathbf{d}}_n^T)$ . The equality in (40) holds when

$$1) \sqrt{1 - \rho_n^2} : \rho_n = \text{Re}\{\mathbf{w}_n^H \mathbf{s}_n\} : \text{Re}\{\mathbf{u}_n^H \mathbf{d}_n\}. \text{ That is} \quad (41)$$

$$\rho_n = \frac{|\text{Re}\{\mathbf{u}_n^H \mathbf{d}_n\}|}{\sqrt{\text{Re}\{\mathbf{w}_n^H \mathbf{s}_n\}^2 (1 - \rho_{-n}^2) + \text{Re}\{\mathbf{u}_n^H \mathbf{d}_n\}^2}}$$

2)  $\tilde{\mathbf{w}}_n$  is eigenvector associated with the smallest eigenvalue  $\sigma_n$  of the matrix  $(\tilde{\Phi}_n - \tilde{\mathbf{s}}_n \tilde{\mathbf{s}}_n^T + \tilde{\mathbf{d}}_n \tilde{\mathbf{d}}_n^T + \check{\mathbf{d}}_n \check{\mathbf{d}}_n^T)$ .

3)  $\tilde{\mathbf{d}}_n^T (\mathbf{I}_{2R} - \tilde{\mathbf{w}}_n \tilde{\mathbf{w}}_n^T - \check{\mathbf{w}}_n \check{\mathbf{w}}_n^T - \tilde{\mathbf{u}}_n \tilde{\mathbf{u}}_n^T) \tilde{\mathbf{d}}_n = 0$ .  
It turns out that

$$\begin{aligned} \mathbf{d}_n^H \mathbf{u}_n &= \tilde{\mathbf{d}}_n^T \tilde{\mathbf{u}}_n = \tilde{\mathbf{d}}_n^T \tilde{\mathbf{d}}_n - \left( \tilde{\mathbf{d}}_n^T \tilde{\mathbf{w}}_n \right)^2 - \left( \tilde{\mathbf{d}}_n^T \check{\mathbf{w}}_n \right)^2 \\ &= \mathbf{d}_n^H \mathbf{d}_n - |\mathbf{w}_n^H \mathbf{d}_n|^2. \end{aligned} \quad (42)$$

Since  $\mathbf{u}_n^H \mathbf{w}_n = 0$ ,  $\mathbf{u}_n$  has closed-form solution as

$$\mathbf{u}_n = \frac{\mathbf{d}_n - \mathbf{w}_n (\mathbf{w}_n^H \mathbf{d}_n)}{\sqrt{\|\mathbf{d}_n\|^2 - |\mathbf{w}_n^H \mathbf{d}_n|^2}}. \quad (43)$$

Finally, we have derived closed-form updates for  $\rho_n$ ,  $\mathbf{w}_n$  and  $\mathbf{u}_n$  while keeping other parameters fixed. The ASU algorithm acts similarly to that in Algorithm 2 for real-valued tensors.

## VI. SIMULATIONS

*Example 1 Comparison of Algorithms for Block Tensor Decompositions:* The first example aims to compare performance of algorithms for the rank-1 tensor extraction, including the deflation algorithm [20], the ASU in Algorithm 2, and two algorithms for rank- $(L_r, M_r, N_r)$  BCD which are the ALS algorithm<sup>3</sup> [29] and the non-linear least squares (NLS) algorithm<sup>4</sup> [38]. We briefly discuss efficiency of different initialization methods including the random initialization, the SVD-based method with pre-selection, and the joint eigenvalue decomposition (JEVD), while a detailed comparison can be found in Part 2 [21]. All algorithms were initialized by the same values, while their parameters are left at default values. For example, the NLS algorithm used the Gauss-Newton algorithm with dogleg trust region. For each decomposition using the random initialization method, we randomly generated 10 sets of initial points, and selected the one achieving the smallest approximation error after passing through the ASU algorithm with a small number of iterations (typically, less than 10).

The data tensors  $\mathcal{Y}$  considered in this example are of size  $R \times R \times R$ , randomly generated as

$$\mathcal{Y} = \left[ \left[ \lambda; \mathbf{a}^{(1)}, \mathbf{a}^{(2)}, \mathbf{a}^{(3)} \right] \right] + \left[ \left[ \mathcal{G}; \mathbf{U}^{(1)}, \mathbf{U}^{(2)}, \mathbf{U}^{(3)} \right] \right] + \sigma \mathcal{E}, \quad (44)$$

where  $\lambda = 1$ ,  $\mathbf{U}^{(n)}$  are matrices of size  $R \times (R - 1)$  with orthonormal columns,  $\mathcal{E}$  represents additive Gaussian noise with zero-mean, unit-variance, and  $\sigma^2$  denotes the noise variance for specific signal-to-noise ratios SNR (dB). The parameters

<sup>3</sup>The ALS algorithm [29] is provided from website of the authors.

<sup>4</sup>The NLS algorithm is available in the Tensorlab toolbox at [www.tensorlab.net](http://www.tensorlab.net).

$\rho_n = \mathbf{a}^{(n)T} \mathbf{u}_n$  was in range  $[0, 0.5]$ . The tensors  $\mathcal{G}$  were random tensors with  $\|\mathcal{G}\|_F^2 = 1$ . Algorithms were set to run until differences between consecutive approximation errors were small enough, i.e.,  $|\varepsilon_k - \varepsilon_{k+1}| < 10^{-6} \varepsilon_k$  where  $\varepsilon = \|\mathcal{Y} - \hat{\mathcal{Y}}\|_F^2$ , or when the number of iterations exceeds 1000. There were 100 independent runs for each setting of rank and SNR, where  $R = 10, 20$  and  $30$  and SNR = 10, 20, 30 dB. Since the ALS algorithm [29] become significant time and memory consuming when  $R \geq 20$ , this algorithm only involved in decompositions of tensors of size  $10 \times 10 \times 10$ . Simulations were run on a computer consisted of Intel Xeon 2 processors clocked at 3.33 GHz, 64 GB of main memory.

Performances of algorithms for the tensor deflation can be assessed through approximation error of the components  $\mathbf{a}^{(n)}$  and the factor matrices  $\mathbf{U}^{(n)}$ . However, as shown in Appendices C, the approximation error between  $\mathbf{U}^{(n)}$  and its approximate can be deduced from that of  $\mathbf{w}_n$ . We will evaluate only the performance of estimation of  $\mathbf{a}^{(n)}$ . Results reported in Table I compare the relative errors  $\frac{1}{N} \sum_{n=1}^N \frac{\|\mathbf{a}^{(n)} - \hat{\mathbf{a}}^{(n)}\|_F^2}{\|\mathbf{a}^{(n)}\|_F^2}$  in dB obtained by these algorithms using multiple random initial points and the JEVD initialization method [21]. The results imply that algorithms purely initialized by random values often yielded poor performance or demanded a large number of iterations. When using JEVD, the algorithms achieved small approximation errors which attain the average CRB on this error for  $\mathbf{a}^{(n)}$  [21]. Results using the SVD-based method are similar to those using JEVD, and are not shown in Table I. When using a good initialization, the algorithms ALS, ASU and NLS converged to similar results. This is because these algorithms minimise the same optimization criterion. The difference between them is mainly the computational cost as measured by execution time and compared in Table II. The ASU algorithm was the fastest algorithm in our simulations. When using the JEVD initialization, this algorithm converged quickly after less than 0.02 seconds, whereas the NLS algorithm stopped after on average 10 seconds when  $R = 30$  and SNR = 10 dB. On average, ASU was 15 times faster than the algorithm [20], 222 times faster than NLS, and 312 times faster than ALS [29]. This example confirms that the ASU algorithm worked well as other algorithms for the same model, but it was much faster than the competitors.

*Example 2 Tensor Deflation in Hard Scenarios:* In this example, we analyze performance of the rank-1 plus multilinear rank- $(R-1, \dots, R-1)$  block tensor decomposition when the angular parameter between components of the rank-1 tensor and factor matrices of the block tensor, i.e., the parameter  $\rho_n$ , was varied from 0 to 0.99. The data tensors  $\mathcal{Y}$  were of size  $10 \times 10 \times 10$ , and their factor matrices were randomly generated with  $\rho_1 = \rho_2 = \rho_3 = c$  taking values of 0, 0.1, ..., 0.9, 0.95 and 0.99. The core tensor  $\mathcal{G}$  was randomly generated and normalized to have unit norm, i.e.,  $\|\mathcal{G}\|_F = 1$ , while we set  $\lambda = 1$ . Gaussian noise was added into the tensor with SNR=20 and 30 dB. The squared angular errors  $\text{SAE}(\mathbf{a}, \hat{\mathbf{a}}) = -10 \log_{10} \arccos \left( \frac{\mathbf{a}^T \hat{\mathbf{a}}}{\|\mathbf{a}\|_2 \|\hat{\mathbf{a}}\|_2} \right)^2$  (dB) between the true and estimated components of the rank-1 tensor were assessed, and compared with the Cramér-Rao Induced bound (CRIB) on this error derived in Part 2 [21]. The results illustrated in Fig. 3 are reported for 400 independent runs, indicate

TABLE I  
COMPARISON OF RELATIVE ERRORS (IN dB) IN ESTIMATION OF COMPONENTS OF THE RANK-1 TENSORS BY THE ALS, ASU AND NLS ALGORITHMS IN EXAMPLE 1. THE RESULTS ARE REPORTED OVER 100 RUNS FOR RANKS  $R = 10, 20$  AND  $30$ , SNR = 10, 20, 30 dB. ALGORITHMS WERE INITIALIZED BY RANDOM VALUES OR USING THE JEVD METHOD

Algorithm	SNR					
	10 dB		20 dB		30 dB	
	Random	JEVD	Random	JEVD	Random	JEVD
$R = 10$						
CRB [21]	37.3		47.3		57.3	
ASU	7.4	37.3	7.4	47.2	7.3	57.3
Alg. [20]	7.4	37.3	7.4	47.2	7.3	57.3
NLS	7.8	37.3	8.1	47.2	7.9	57.3
ALS [29]	7.4	37.3	7.4	47.2	7.3	57.3
$R = 20$						
CRB [21]	46.2		56.2		66.2	
ASU	10.3	46.2	10.3	56.2	10.2	66.2
Alg. [20]	10.3	46.2	10.3	56.2	10.2	66.2
NLS	10	46.1	9.6	56.2	9.2	66.2
$R = 30$						
CRB [21]	51.5		61.5		71.5	
ASU	12.0	51.5	11.9	61.4	11.9	71.5
Alg. [20]	12.0	51.5	11.9	61.4	11.9	71.5
NLS	10.5	51.5	10.3	61.4	10.7	71.5

TABLE II  
COMPARISON OF EXECUTION TIME (SECONDS) OF ALGORITHMS IN EXAMPLE 1. VALUES INSIDE BRACKET ARE SPEEDUP RATIOS OF EXECUTION TIME OF ALGORITHMS TO THAT OF ASU

Algorithm	SNR					
	10 dB		20 dB		30 dB	
	SVD	JEVD	SVD	JEVD	SVD	JEVD
$R = 10$						
ASU	0.010	0.010	0.011	0.009	0.011	0.009
Alg. [20]	0.24 (24)	0.22 (21)	0.20 (18)	0.13 (15)	0.21 (19)	0.13 (14)
NLS	2.7 (269)	0.79 (76)	1.3 (119)	0.28 (32)	0.79 (72)	0.27 (30)
ALS [29]	4.3 (426)	4 (384)	3.3 (303)	2.1 (239)	3.2 (287)	2.1 (231)
$R = 20$						
ASU	0.010	0.022	0.009	0.007	0.009	0.008
Alg. [20]	0.17 (17)	0.37 (17)	0.13 (15)	0.082 (11)	0.13 (14)	0.085 (11)
NLS	5 (511)	6 (290)	2.5 (276)	0.51 (70)	1.5 (170)	0.53 (70)
$R = 30$						
ASU	0.014	0.023	0.014	0.011	0.014	0.011
Alg. [20]	0.21 (15)	0.34 (15)	0.17 (12)	0.11 (10)	0.17 (12)	0.11 (10)
NLS	10 (722)	11 (488)	5 (375)	1.1 (98)	3.2 (237)	1.1 (96)

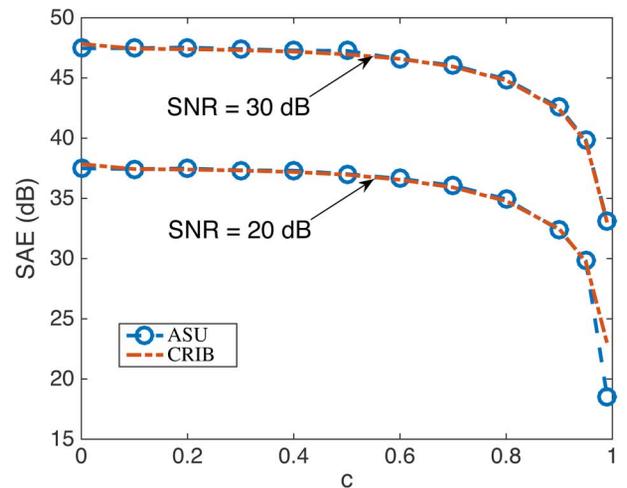


Fig. 3. Illustration of squared angular errors (SAE) as function of the parameter  $\rho_1 = \rho_2 = \rho_3 = c$  varying in the range  $[0, 0.99]$ . The performance is compared with the Cramér-Rao Induced bound on the squared angular error [21] for two different noise levels. Tensors in this example are of size  $10 \times 10 \times 10$ .

that the error bounds are small for small  $c$ , especially when components  $\mathbf{a}^{(n)}$  and  $\mathbf{U}^{(n)}$  are mutually orthogonal,  $c = 0$ . The tensor deflation procedure becomes more difficult when  $c$  approaches 1. For most test cases, ASU achieved good performance with its SAEs attaining the error bound, except for the difficult case when  $\rho_1 = \rho_2 = \rho_3 = 0.99$ , that is,  $\mathbf{a}^{(n)}$  are highly correlated with  $\mathbf{u}_n$ , for  $n = 1, 2, 3$ . We note that tensors in a such difficult case can be considered not to satisfy the condition  $c$  in Lemma 2.

*Example 3 Deflation of Tensors Obeying the CP Model:* We have shown that by using the rank-1 tensor deflation, e.g., the ASU or ALS algorithm, one can sequentially perform CPD of a rank- $R$  tensor through  $R$  rank-1 tensor deflations. In this example, we will show that the extraction process of  $R$  rank-1 tensors can be done in parallel. The main idea of this method is to provide a good initial value to the ASU algorithm which can be obtained using the tensor diagonalization (TEDIA) [18] or the direct trilinear decomposition (DTLD) [39] algorithm.

We illustrate rank-1 tensor extraction through decompositions of synthetic data of size  $100 \times 100 \times 100$  with rank  $R = 5, 10, 20, 30, 40$ . Factor matrices  $\mathbf{A}^{(n)}$  were randomly generated such that their collinearity coefficients  $\frac{\mathbf{a}_r^{(n)T} \mathbf{a}_s^{(n)}}{\|\mathbf{a}_r^{(n)}\| \|\mathbf{a}_s^{(n)}\|}$  were in the ranges of  $[0.2, 0.8]$ ,  $[0.4, 0.8]$ ,  $[0.4, 0.9]$ , see, e.g., [40], [41]. The tensors were corrupted with additive Gaussian noise of signal-to-noise ratio SNR = 10, 20, 30 and 40 dB. We generated 100 Kruskal tensors with  $\lambda_1 = \dots = \lambda_R = 1$  for rank  $R$ , and added an i.i.d Gaussian noise to achieve the specified SNR level. Therefore, there were in total 2000 noisy tensors to be decomposed.

The performance of the decomposition was evaluated through the squared angular errors, and compared with the Cramér-Rao induced bound for CPD CRIB( $\mathbf{a}$ ) (in dB) [42], [43]. The mean SAE (MSAE) was averaged over 300  $R$  SAEs of all components in all factor matrices. Fig. 4 compares the MSAEs achieved by FastALS [37], Seq-ASU which sequentially extracts rank-1 tensors, and Par-ASU which simultaneously extracts rank-1 tensors in a parallel loop. Algorithms stop when differences between successive relative errors  $\varepsilon = \frac{\|\mathcal{Y} - \hat{\mathcal{Y}}\|_F}{\|\mathcal{Y}\|_F}$  were lower than  $10^{-8}$ , or until the maximum number of iterations (2000) was achieved. For most of the tests, both Seq-ASU and Par-ASU attained the CRIB, such as when  $R = 5, 10, 20, 30$ . Only when the cases of high rank  $R = 40$  and low SNR=10 dB, the two methods for rank-1 tensor extraction achieved slightly lower MSAE than the FastALS algorithm for CPD. It is worth noting that CRIB for such decomposition was 21.34 dB, implying that the standard angular deviation (square root of mean square angular error) of the factor is cca. 5.72 degrees.

*Example 4 Tensor Deflation With Overestimated Rank:* Tensor deflation developed in this paper requires tensor rank to be known. This example will show that when the tensor rank is not given, the ASU algorithm is still able to extract the desired factor matrices (components). We generated rank- $R$  tensors of size  $I \times I \times I$  where the tensor rank  $R$  did not exceed dimension  $I$ . In this example,  $I = 3R$ . The rank- $R$  tensors were then degraded by additive Gaussian noise with SNR = 10, 20, 30, 40 dB. The ASU algorithm was applied to sequentially extract

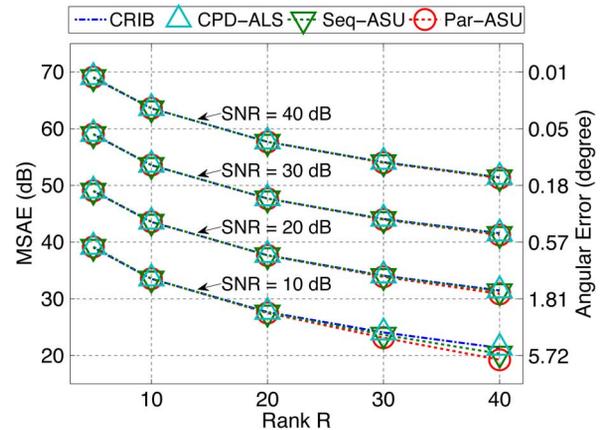


Fig. 4. Comparison of SAE (dB) of CPD algorithms with CRIB for decomposition of noisy tensors.

TABLE III

COMPARISON OF THE MEAN SQUARED ANGULAR ERRORS (dB) IN ESTIMATION OF COMPONENTS FROM RANK- $R$  TENSORS OF SIZE  $3R \times 3R \times 3R$  DEGRADED BY GAUSSIAN NOISE AT SNR = 10, 20, 30, 40 dB. FOR EACH RANK  $R$  AND SNR, WE SHOW PERFORMANCES OF THE ASU ALGORITHM WITH EXACT RANK AND WHEN THE RANK WAS OVERESTIMATED

Rank	SNR (dB)							
	10		20		30		40	
5	21.51	17.62	31.57	30.01	41.55	40.94	51.53	51.37
		21.30		31.48		41.40		51.50
10	22.17	18.52	32.24	30.09	42.22	41.43	52.17	51.89
		21.82		31.98		42.07		52.15
20	22.55	19.93	32.55	30.32	42.54	41.53	52.54	52.37
		21.98		32.42		42.40		52.49
30	22.70	20.67	32.70	30.11	42.69	41.65	52.69	52.59
		21.94		32.57		42.59		52.63

CRIB	ASU with over-estimate rank
	ASU with exact rank

rank-1 tensors of the data with a sufficient high initial rank, say  $R = I$ . The tensors were re-factorized with lower ranks when there was any  $\rho_n = \mathbf{u}_n^T \mathbf{a}^{(n)} \approx 1$ . The mean squared angular errors (MSAE) in estimation of components were assessed over 100 independent runs, and compared with the Cramér-Rao induced bound (CRIB) [43] in Table III. When the noise was relatively low, e.g., SNR = 30 or 40 dB, factorisation using ASU with inexact initial rank still achieved high performance which attained CRIB. The loss of SAE was slightly higher and approximately 1–2 dB with SNR = 10 and 20 dB.

*Example 5 Decomposition of the Fluorescence Data Set:* In this section, we illustrate advantage of the tensor deflation using ASU over CPD algorithms in factorisation of a fluorescence data set consisting of five simple samples of fluorescence excitation-emission of size 5 samples  $\times$  201 emission wavelengths  $\times$  61 excitation wavelengths [16]. According to [16], a CP model with  $R = 3$  is appropriate to this fluorescence data. Using the ASU algorithm over two deflation stages, one can extract all three components with high accuracy compared with those extracted by using CPD-ALS. However, the major aim of this example is to show performance of ASU when the decomposition rank  $R$  exceeds the “true” rank 3. It is known that for such cases of decomposition, CPD algorithms often cannot preserve waveforms of the “true” components, which are estimated with appropriate rank, and tends to generate diverging components. For

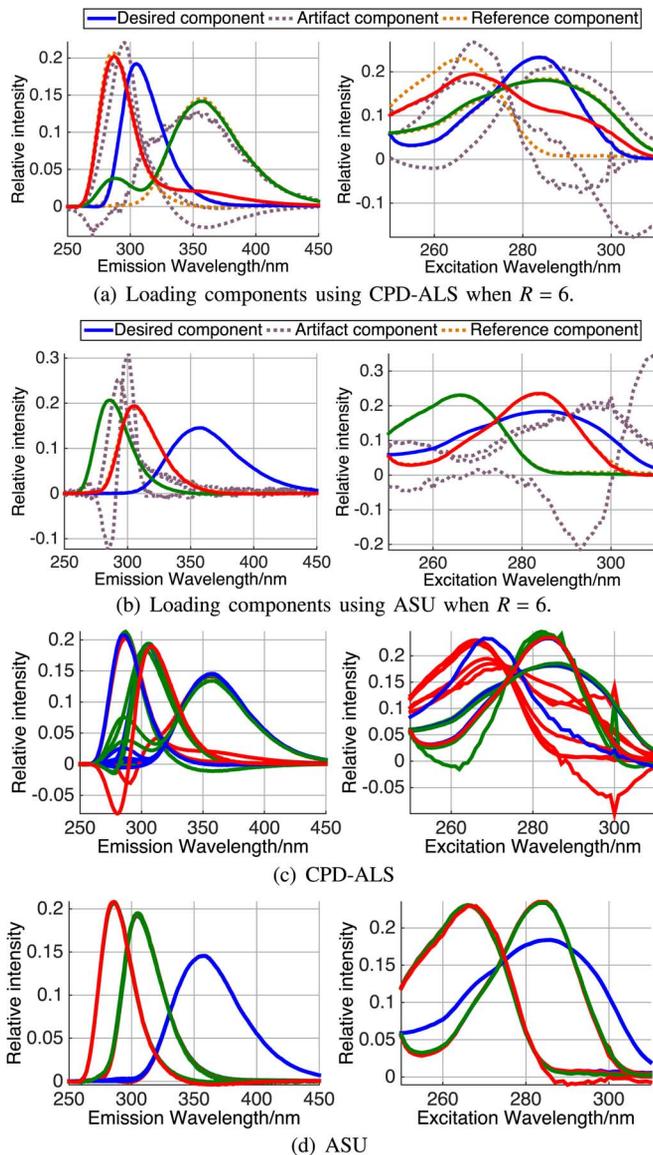


Fig. 5. (a)–(b) Illustration of components extracted from the fluorescence data set [16] using CPD-ALS and ASU when rank  $R = 6$ , (c)–(d) components matched with the reference components obtained by two algorithms when  $R = 3, 4, \dots, 10$ .

example, in Fig. 5(a), among six components extracted using CPD-ALS, there was only one rank-1 tensor whose components (blue solid curves) matched with the reference components, i.e., ones estimated with the “true rank”  $R = 3$  using CPD-ALS. There were also two rank-one components whose components (red and green solid curves) were degraded from the true ones. With the same decomposition rank  $R = 6$ , as seen solid curves in Fig. 5(b), ASU returned the desired components which were slightly different from the true components by less than  $0.53^\circ$ . We note that since rank  $R = 6$  exceeds the first dimension, in order to apply ASU, the fluorescence tensor was expanded along the mode-1 by itself with a small Gaussian noise.

Further results are reported in Figs. 5(c), 5(d) and Fig. 6 for the decompositions with various ranks  $R = 4, 5, \dots, 10$ . In Fig. 5(c), we show the components estimated by CPD-ALS which are highly matched with the “reference” components, whereas those obtained by ASU are illustrated in Fig. 5(d).

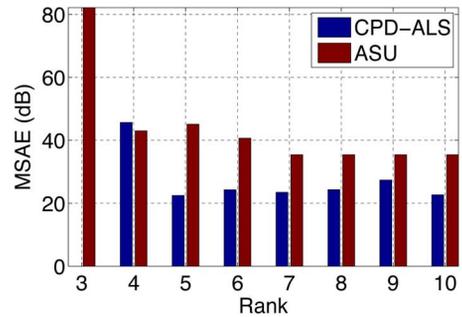


Fig. 6. Mean squared angular errors of components extracted using ASU and CPD-ALS which match with the reference components in Example for factorization of the fluorescence data set [16].

The figures indicate that the desired components were reliably extracted by the ASU algorithm even when the decomposition rank was higher than the “true” rank of data. In Fig. 6, the observation is also confirmed through the squared angular error between the “reference” components, and the components estimated with higher ranks. Since the reference components are estimated by CPD-ALS with rank  $R = 3$ , there is only one bar showing MSAE for the ASU algorithm. When  $R = 4$ , CPD-ALS was still able to extract the desired components. However, when the rank was higher,  $R = 5, 6, \dots, 10$ , the components estimated by CPD were significantly different from the “true” ones with an MSAE  $\approx 24.1$  dB, i.e., by an angle of  $3.6$  degree. For this problem with overestimated rank, one can apply the all-at-once optimization (OPT) algorithm for CPD [44] which is reported to give stable results.

*Example 6 Tracking Received Signals in a Direct Sequence Code Division Multiple Access (DS-CDMA) System:* This example illustrates an application of tensor deflation in tracking one or a few components of rank-1 tensors of interest in a DS-CDMA system. Consider a DS-CDMA system of  $R$  users and  $K$  antennas over a flat Rayleigh fading, each information sequence of user  $r$   $\mathbf{s}_r \in \mathbb{C}^P$  is spread using a code  $\mathbf{c}_m \in \mathbb{C}^Q$  before transmission over fading channels. At the receiver side, an array of  $K$  antennas is employed to receive and decode the signals. Sidiropoulos *et al.* [45] established the model of wireless transmission as a three-way diversity tensor  $\mathcal{Y} \in \mathbb{C}^{K \times P \times Q}$  whose an entry  $y_{k,p,q}$  denotes the baseband output of the  $k$ -th antenna, for symbol  $p$  and chip  $q$

$$y_{k,p,q} = \sum_{r=1}^R a_{k,r} s_{p,r} c_{q,r} + e_{k,p,q}, \quad (45)$$

where  $a_{k,r}$  fading/gain between user  $r$  and antenna element  $k$ . This model can be expressed as

$$\mathcal{Y} = \llbracket \mathbf{A}, \mathbf{S}, \mathbf{C} \rrbracket + \mathcal{E}, \quad (46)$$

where  $\mathcal{E} = [e_{k,p,q}]$  is tensor of additive Gaussian noise,  $\mathbf{A} \in \mathbb{C}^{K \times R}$  denotes the compound flat fading/array response pattern,  $\mathbf{S} \in \mathbb{C}^{P \times R}$  is the information bearing signal matrix, and  $\mathbf{C} \in \mathbb{C}^{Q \times R}$  is the spreading code matrix [45]. Approximation of the output of antennas returns the signal matrix  $\hat{\mathbf{S}}$ . Then with an appropriate demodulation for each column  $\hat{\mathbf{s}}_r$ , the user information sequences can be retrieved.

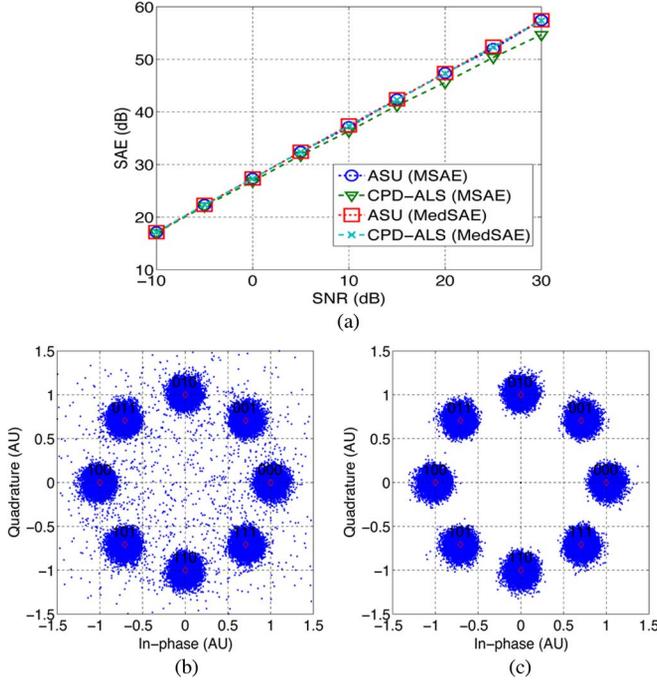


Fig. 7. Illustration of SAEs on tracking components of user-1, and constellation of the received signals at SNR = -5 dB by CPD-ALS and ASU algorithms. (a) MSAE on estimation of  $\mathbf{a}_1^{(k)}$ ,  $\mathbf{b}_1^{(k)}$  and  $\mathbf{c}_1^{(k)}$ . (b) CPD-ALS (c) ASU.

In our experiments,  $K = 50$  receivers (antennas) and  $R = 10$  users,  $P = 50$ ,  $Q = 64$  spreading gain, and user sequences were modulated by M-DPSK ( $M = 8$ ). There were all  $L = 20$  tensor blocks  $\mathcal{Y}_l$  of size  $50 \times 50 \times 64$ ,  $l = 1, \dots, L$  recorded at different time points. The data tensors  $\mathcal{Y}_l$  can be obtained sequentially in an online system, and have the same number of users, i.e., rank  $R$ . However, we assume that only user-1 joins in all  $L = 20$  tensors  $\mathcal{Y}_l$  with the same or slowly changing array response  $\mathbf{a}_1$  and spreading code  $\mathbf{c}_1$ , while the  $(R - 1)$  other users in different blocks  $\mathcal{Y}_l$  are different, that is components  $\mathbf{a}_r^{(l)}$ ,  $\mathbf{b}_r^{(l)}$  and  $\mathbf{c}_r^{(l)}$  are completely changed, for  $r = 2, \dots, R$ . In our simulations, components of the user-1  $\mathbf{a}_1^{(l)}$  and  $\mathbf{c}_1^{(l)}$  for  $l = 1, \dots, L$  were degraded from  $\mathbf{a}_1$  and  $\mathbf{c}_1$  by Gaussian noise at SNR = 20 dB. The task was to track rank-1 tensors corresponding to this user in tensors  $\mathcal{Y}_2, \dots, \mathcal{Y}_L$ . Note that for such formulated problem, concatenation of two rank- $R$  tensors  $\mathcal{Y}_k$  and  $\mathcal{Y}_{k+1}$  does not yield a tensor of rank  $R$  or approximate rank- $R$ . Thereby, we cannot track all the components  $\mathbf{a}_r^{(l)}$ ,  $\mathbf{b}_r^{(l)}$  and  $\mathbf{c}_r^{(l)}$  using the algorithm proposed in [19].

In our experiment, we first decomposed  $\mathcal{Y}$  using the fLM algorithm [46], and selected rank-1 tensor associated with this user. When a new data  $\mathcal{Y}_l$  of size  $50 \times 50 \times 64$  was available, we compressed this tensor to the size of  $R \times R \times R$ , and performed a rank-1 tensor extraction from  $\mathcal{Y}_l$  using the ASU algorithm with components initialized by the rank-1 tensor of interest in  $\mathcal{Y}_1$  or in the previous observed tensors  $\mathcal{Y}_{k-1}$ . The mean SAE (dB) on estimation of the components of user-1 was averaged over 100 independent runs for each SNR = -10, -5,  $\dots$ , 30 dB. Results are compared with those obtained by full CP decompositions of  $\mathcal{Y}_k$  using FastALS [37]. We set algorithms to run within 1000 iterations, and stop when the relative approximation error was lower than  $10^{-6}$ . As shown in Fig. 7, although both ASU and

FastALS achieved competitive median SAEs (dB), the MSAEs of FastALS were slightly lower than those of ASU when SNR is high, e.g., 20, 30 dB. Fig. 7 also shows constellation of the received signals at SNR = -5 dB.

## VII. CONCLUSIONS

Tensor deflation approach and associated algorithms have been proposed for both real- and complex-valued tensors of rank- $R$  and tensors comprising rank-1 tensors and block tensors. The algorithms update parameters in closed-form. Together with new efficient initialization methods presented in Part 2 [21], the proposed algorithms have been confirmed to be superior over existing algorithms which are applicable for the same or similar tensor decomposition. The tensor deflation can find some potential applications related to tracking or inspection of individual time variation of components. The tensor deflation is also able to extract rank-1 tensors in CPD in parallel way. The rank-1 tensor deflation can be extended to extract dominant rank-1 tensor with the largest weight  $\lambda$ . A more general tensor deflation is rank splitting which splits a high rank CPD into decompositions of two blocks of smaller ranks. For this extension, we refer to Part 3 of this study [22]. Finally, the deflation algorithms 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 1

*Proof:* Let  $\mathbf{Q}_n$  be a matrix comprising an orthonormal basis of column space of  $\mathbf{U}^{(n)}$ ,  $\boldsymbol{\gamma}_n = \mathbf{Q}_n^T \mathbf{a}^{(n)} \in \mathbb{R}^{R-1}$  and let  $\boldsymbol{\Gamma}_n$  be an  $(R - 1) \times (R - 1)$  orthonormal matrix, having the first normalized column  $\boldsymbol{\gamma}_n / \|\boldsymbol{\gamma}_n\|$ , and the remaining columns form an orthonormal basis for the orthogonal complement of  $\boldsymbol{\gamma}_n$ . Note that  $\mathbf{Q}_n$  and  $\boldsymbol{\Gamma}_n$  can be obtained by using QR decomposition of  $\mathbf{U}^{(n)}$  and  $\boldsymbol{\gamma}_n$ , respectively, or alternatively the singular value decomposition of these matrices. Then,  $[\boldsymbol{\mathcal{G}}; \mathbf{U}^{(1)}, \mathbf{U}^{(2)}, \dots, \mathbf{U}^{(N)}] = [\tilde{\boldsymbol{\mathcal{G}}}; \tilde{\mathbf{U}}^{(1)}, \tilde{\mathbf{U}}^{(2)}, \dots, \tilde{\mathbf{U}}^{(N)}]$  with  $\mathbf{U}^{(n)} = \mathbf{Q}_n \boldsymbol{\Gamma}_n$ ,  $n = 1, \dots, N$ ,  $\tilde{\boldsymbol{\mathcal{G}}} = \boldsymbol{\mathcal{G}} \times_1 (\boldsymbol{\Gamma}_1^T \mathbf{Q}_1^T \mathbf{U}^{(1)}) \dots \times_N (\boldsymbol{\Gamma}_N^T \mathbf{Q}_N^T \mathbf{U}^{(N)})$ . It can be verified that  $\tilde{\mathbf{U}}^{(n)}$  are orthogonal and  $(\tilde{\mathbf{U}}^{(n)})^T \mathbf{a}^{(n)} = \boldsymbol{\Gamma}_n^T \mathbf{Q}_n^T \mathbf{a}^{(n)} = \boldsymbol{\Gamma}_n^T \boldsymbol{\gamma}_n = \|\boldsymbol{\gamma}_n\|_2 \mathbf{e}_1$ , where  $\mathbf{e}_1 = [1, 0, \dots, 0]^T \in \mathbb{R}^{R-1}$ . It means that the new decomposition obeys the required orthogonality condition. ■

## APPENDIX B

### PROOF OF LEMMA 2

*Proof:*

- If  $\mathbf{a}^{(n)}$  lies in the column space of  $\mathbf{U}^{(n)}$ , it can be expressed as  $\mathbf{a}^{(n)} = \mathbf{U}^{(n)} \mathbf{v}$ . According to the normalization, we have  $\mathbf{v} = \mathbf{U}^{(n)T} \mathbf{a}^{(n)} = \rho_n [1, 0, \dots, 0]^T$  and  $\mathbf{v}^T \mathbf{v} = \mathbf{a}^{(n)T} \mathbf{a}^{(n)} = 1$ , indicating that  $\rho_n = 1$ , or  $\mathbf{a}^{(n)} = \mathbf{u}_1^{(n)}$ .

- b. The rest of the proof is by contradiction. Assume that  $\rho^2 = 1$ . Since the vectors  $\mathbf{a}^{(n)}$  and  $\mathbf{u}_1^{(n)}$  have both unit norm, they are identical up to sign for all  $n$ . It follows that the approximation in (13),  $\mathcal{Y}$  has multilinear rank  $(R_1 - 1, \dots, R_N - 1)$  as well (not only the latter term). Now, we can construct new  $\mathbf{a}^{(n)}$  based on rank one approximation of the residual,  $\mathcal{Y} - \llbracket \mathcal{G}; \{\mathbf{U}^{(n)}\} \rrbracket$ . The new decomposition will have a lower approximation error than the former one—which contradicts the assumption that the original decomposition was optimal.
- c. Assume that  $(N - 1)$  components  $\mathbf{a}^{(n)}$  lie in the column spaces of  $\mathbf{U}^{(n)}$ , that is,  $\mathbf{a}^{(n)} = \mathbf{u}_1^{(n)}$  for  $n = 2, \dots, N$  and  $0 \leq \rho_1 = \mathbf{a}^{(1)T} \mathbf{u}_1^{(1)} < 1$ . Let  $\bar{\mathbf{u}}_1 = \frac{\mathbf{a}^{(1)} - \rho_1 \mathbf{u}_1^{(1)}}{\xi_1}$ , where  $\xi_1 = \sqrt{1 - \rho_1^2}$ , then  $\bar{\mathbf{u}}_1^T \bar{\mathbf{u}}_1 = 1$ ,  $\bar{\mathbf{u}}_1^T \mathbf{U}^{(1)} = 0$ , and  $\mathbf{a}^{(1)} = \bar{\mathbf{u}}_1 \xi_1 + \mathbf{u}_1^{(1)} \rho_1$ . From (4), we have

$$s = \mathcal{Y} \times_1 \bar{\mathbf{u}}_1^T \times_2 \mathbf{a}^{(2)T} \dots \times_N \mathbf{a}^{(N)T} = \lambda \xi_1$$

$$\mathcal{T} = \mathcal{Y} \times_1 \mathbf{U}^{(1)T} \times_2 \mathbf{U}^{(2)T} \dots \times_N \mathbf{U}^{(N)T} = \mathcal{G} + \lambda \rho_1 \mathcal{E}_1,$$

where  $\mathcal{E}_1$  has one at its first entry and zeros elsewhere. It yields

$$\begin{aligned} & \llbracket \mathcal{G}; \mathbf{U}^{(1)}, \dots, \mathbf{U}^{(N)} \rrbracket + \llbracket \lambda; \mathbf{a}^{(1)}, \dots, \mathbf{a}^{(N)} \rrbracket \\ &= \llbracket \mathcal{T}; \mathbf{U}^{(1)}, \dots, \mathbf{U}^{(N)} \rrbracket - \llbracket \lambda \rho_1; \mathbf{u}_1^{(1)}, \mathbf{a}^{(2)}, \dots, \mathbf{a}^{(N)} \rrbracket \\ &+ \llbracket \lambda; \xi_1 \bar{\mathbf{u}}_1 + \rho_1 \mathbf{u}_1^{(1)}, \mathbf{a}^{(2)}, \dots, \mathbf{a}^{(N)} \rrbracket \\ &= \llbracket \mathcal{Y}; \mathbf{U}^{(1)} \mathbf{U}^{(1)T}, \dots, \mathbf{U}^{(N)} \mathbf{U}^{(N)T} \rrbracket \\ &+ \llbracket s; \bar{\mathbf{u}}_1, \mathbf{a}^{(2)}, \dots, \mathbf{a}^{(N)} \rrbracket. \end{aligned}$$

For any real-valued  $0 < \alpha < \pi$ , let  $\mathcal{G}^* = \mathcal{T} - s \cot(\alpha) \mathcal{E}_1$ , and  $\mathbf{a}^{(1)*} = \sin(\alpha) \bar{\mathbf{u}}_1 + \cos(\alpha) \mathbf{u}_1^{(1)}$ . We show that the following decomposition also achieves the same approximation error

$$\begin{aligned} \mathcal{Y} &\approx \llbracket \mathcal{G}^*; \mathbf{U}^{(1)}, \dots, \mathbf{U}^{(N)} \rrbracket \\ &+ \llbracket \frac{s}{\sin(\alpha)}; \mathbf{a}^{(1)*}, \mathbf{a}^{(2)}, \dots, \mathbf{a}^{(N)} \rrbracket \\ &= \llbracket \mathcal{Y}; \mathbf{U}^{(1)} \mathbf{U}^{(1)T}, \dots, \mathbf{U}^{(N)} \mathbf{U}^{(N)T} \rrbracket \\ &+ \llbracket s; \bar{\mathbf{u}}_1, \mathbf{a}^{(2)}, \dots, \mathbf{a}^{(N)} \rrbracket. \end{aligned}$$

This implies that  $\mathbf{a}^{(1)}$  cannot be uniquely identified. ■

## APPENDIX C

### RELATION BETWEEN APPROXIMATION ERRORS AND ANGULAR ERROR OF $\mathbf{U}^{(n)}$ AND $\mathbf{w}_n$

In order to assess the approximation error between  $\mathbf{U}^{(n)}$  and its estimate  $\hat{\mathbf{U}}^{(n)}$ , as suggested in [29] in the tensor deflation,

one can rotate  $\hat{\mathbf{U}}^{(n)}$  by an orthonormal matrix  $\mathbf{Q}$  of size  $(R - 1) \times (R - 1)$  such that its approximation error is minimized

$$\text{err}(\mathbf{U}^{(n)}, \hat{\mathbf{U}}^{(n)}) = \min_{\mathbf{Q}} \left\| \mathbf{U}^{(n)} - \hat{\mathbf{U}}^{(n)} \mathbf{Q} \right\|_F^2. \quad (47)$$

It is obvious that  $\mathbf{Q} = \hat{\mathbf{U}}^{(n)T} \mathbf{U}^{(n)}$ , and the error is given by

$$\begin{aligned} \text{err}(\mathbf{U}^{(n)}, \hat{\mathbf{U}}^{(n)}) &= \left\| \left( \mathbf{I} - \hat{\mathbf{U}}^{(n)} \hat{\mathbf{U}}^{(n)T} \right) \mathbf{U}^{(n)} \right\|_F^2 \\ &= \left\| \hat{\mathbf{w}}_n \hat{\mathbf{w}}_n^T \mathbf{U}^{(n)} \right\|_F^2 = \hat{\mathbf{w}}_n^T \mathbf{U}^{(n)} \mathbf{U}^{(n)T} \hat{\mathbf{w}}_n \\ &= 1 - \left( \hat{\mathbf{w}}_n^T \mathbf{w}_n \right)^2 \\ &= \sin^2 \left( \angle \left( \mathbf{w}^{(n)}, \hat{\mathbf{w}}^{(n)} \right) \right) \\ &= \sin^2 \left( \angle \left( \mathbf{U}^{(n)}, \hat{\mathbf{U}}^{(n)} \right) \right) \end{aligned} \quad (48)$$

where  $\mathbf{w}_n$  and  $\hat{\mathbf{w}}_n$  are vectors of orthogonal complement to  $\mathbf{U}^{(n)}$  and  $\hat{\mathbf{U}}^{(n)}$ , respectively, and notation  $\angle(\mathbf{A}, \mathbf{B})$  represents angle between two subspaces, which can be computed using in the Matlab subroutine “subspace”. It indicates that the approximation error between  $\mathbf{U}^{(n)}$  and  $\hat{\mathbf{U}}^{(n)}$  can be assessed through the angular error of  $\mathbf{w}_n$  and  $\hat{\mathbf{w}}_n$ , which is also equivalent to angular error between  $\mathbf{U}^{(n)}$  and  $\hat{\mathbf{U}}^{(n)}$ .

## 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] N. D. Sidiropoulos and R. Bro, “PARAFAC techniques for signal separation,” in *Signal Processing Advances in Communications*, P. Stoica, G. Giannakis, Y. Hua, and L. Tong, Eds. Upper Saddle River, NJ, USA: Prentice-Hall, 2000, vol. 2, ch. 4.
- [2] H. Becker, P. Comon, L. Albera, M. Haardt, and I. Merlet, “Multi-way space-time-wave-vector analysis for EEG source separation,” *Signal Process.*, vol. 92, no. 4, pp. 1021–1031, 2012.
- [3] M. Mørup, L. K. Hansen, C. S. Herrmann, J. Parnas, and S. M. Arnfred, “Parallel factor analysis as an exploratory tool for wavelet transformed event-related EEG,” *NeuroImage*, vol. 29, no. 3, pp. 938–947, 2006.
- [4] E. Acar, C. A. Bingol, H. Bingol, R. Bro, and B. Yener, “Multiway analysis of epilepsy tensors,” *Bioinformatics*, vol. 23, pp. 10–18, 2007.
- [5] 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.
- [6] J. D. Carroll and J. J. Chang, “Analysis of individual differences in multidimensional scaling via an  $n$ -way generalization of Eckart-Young decomposition,” *Psychometrika*, vol. 35, no. 3, pp. 283–319, 1970.
- [7] R. A. Harshman, “Foundations of the PARAFAC procedure: Models and conditions for an explanatory multimodal factor analysis,” *UCLA Working Papers in Phonetics*, vol. 16, pp. 1–84, 1970.

- [8] P. White, "The computation of eigenvalues and eigenvectors of a matrix," *J. Soc. Ind. Appl. Math.*, vol. 6, no. 4, pp. 393–437, 1958.
- [9] E. E. Osborne, "On acceleration and matrix deflation processes used with the power method," *J. Soc. Ind. Appl. Math.*, vol. 6, no. 3, pp. 279–287, 1958.
- [10] N. Delfosse and P. Loubaton, "Adaptive blind separation of independent sources: A deflation approach," *Signal Process.*, vol. 45, no. 1, pp. 59–83, 1995.
- [11] L. Mackey, "Deflation methods for sparse PCA," in *Proc. NIPS*, D. Koller, D. Schuurmans, Y. Bengio, and L. Bottou, Eds., 2008, pp. 1017–1024.
- [12] M. T. Chu, R. E. Funderlic, and G. H. Golub, "A rank-one reduction formula and its applications to matrix factorizations," *SIAM Rev.*, vol. 37, no. 4, pp. 512–530, 1995.
- [13] Y. Wang and L. Qi, "On the successive supersymmetric rank-1 decomposition of higher-order supersymmetric tensors," *Numerical Linear Algebra Appl.*, vol. 14, no. 6, pp. 503–519, 2007.
- [14] E. Kofidis, Phillip, and A. Regalia, "On the best rank-1 approximation of higher-order supersymmetric tensors," *SIAM J. Matrix Anal. Appl.*, vol. 23, pp. 863–884, 2002.
- [15] A. Stegeman and P. Comon, "Subtracting a best rank-1 approximation may increase tensor rank," *J. Linear Algebra Appl.*, vol. 433, no. 7, pp. 1276–1300, Dec. 2010.
- [16] R. Bro, "Multi-way analysis in the food industry—Models, algorithms, and applications," Ph.D. dissertation, Univ. of Amsterdam, Amsterdam, The Netherlands, 1998.
- [17] J. A. Westerhuis and A. K. Smilde, "Deflation in multiblock PLS," *J. Chemometr.*, vol. 15, no. 5, pp. 485–493, 2001.
- [18] 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>
- [19] D. Nion and N. D. Sidiropoulos, "Adaptive algorithms to track the PARAFAC decomposition of a third-order tensor," *IEEE Trans. Signal Process.*, vol. 57, no. 6, pp. 2299–2310, June 2009.
- [20] 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.
- [21] A.-H. Phan, P. Tichavský, and A. Cichocki, "Tensor deflation for CANDECOMP/PARAFAC—Part II: Initialization and Error analysis," *IEEE Trans. Signal Process.*, vol. 63, no. 22, pp. 5939–5950, 2015.
- [22] 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>
- [23] A.-H. Phan, P. Tichavský, and A. Cichocki, "Rank splitting for CANDECOMP/PARAFAC," in *Latent Variable Analysis and Signal Separation: 12th International Conference, LVA/ICA 2015, Liberec, Czech Republic, August 25–28, 2015 Proceedings*. New York, NY, USA: Springer, 2015, LNCS 9237.
- [24] T. G. Kolda and B. W. Bader, "Tensor decompositions and applications," *SIAM Rev.*, vol. 51, no. 3, pp. 455–500, Sep. 2009.
- [25] L.-H. Lim and P. Comon, "Blind multilinear identification," 2012, abs/1212.6663 [Online]. Available: <http://arxiv.org/abs/1212.6663>, preprint
- [26] P. Comon, "Tensors: A brief introduction," *IEEE Signal Process. Mag.*, vol. 31, no. 1, pp. 44–53, 2014.
- [27] A. Cichocki, D. P. Mandic, A.-H. Phan, C. Caifa, G. Zhou, and Q. Zhao *et al.*, "Tensor decompositions for signal processing applications. from two-way to multiway component analysis," *IEEE Signal Process. Mag.*, vol. 32, no. 2, pp. 145–163, 2015.
- [28] L. De Lathauwer, "Decompositions of a higher-order tensor in block terms—Part I: Lemmas for partitioned matrices," *SIAM J. Matrix Anal. Appl., Special Issue on Tensor Decompositions and Applications*, vol. 30, no. 3, pp. 1022–1032, 2008.
- [29] 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., Special Issue Tensor Decompositions and Applications*, vol. 30, no. 3, pp. 1067–1083, 2008.
- [30] A. Stegeman, "On uniqueness conditions for Candecom/Parafac and Indscal with full column rank in one mode," *Linear Algebra Appl.*, vol. 431, no. 1–2, pp. 211–227, 2009.
- [31] 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.
- [32] P. Comon, X. Luciani, and A. L. F. de Almeida, "Tensor decompositions, alternating least squares and other tales," *J. Chemometr.*, vol. 23, no. 7–8, pp. 393–405, 2009.
- [33] A.-H. Phan, A. Cichocki, and P. Tichavský, "On fast algorithms for orthogonal Tucker decomposition," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process. (ICASSP)*, May 2014, pp. 6766–6770.
- [34] W. I. Zangwill, *Nonlinear Programming: A Unified Approach*. Englewood Cliffs, NJ, USA: Prentice-Hall, 1967.
- [35] P. Stoica and Y. Selén, "Model-order selection: A review of information criterion rules," *IEEE Signal Process. Mag.*, vol. 21, no. 4, pp. 36–47, July 2004.
- [36] E. G Larsson and P. Stoica, *Space-time Block Coding for Wireless Communications*. Cambridge, U.K.: Cambridge Univ. Press, 2008.
- [37] A.-H. Phan, P. Tichavský, and A. Cichocki, "Fast alternating LS algorithms for high order CANDECOMP/PARAFAC tensor factorizations," *IEEE Trans. Signal Process.*, vol. 61, no. 19, pp. 4834–4846, 2013.
- [38] L. Sorber, M. Van Barel, and L. De Lathauwer, "Structured data fusion," *IEEE J. Sel. Topics Signal Process.*, vol. 9, no. 4, pp. 586–600, Jun. 2015.
- [39] E. Sanchez and B. R. Kowalski, "Tensorial resolution: A direct trilinear decomposition," *J. Chemometr.*, vol. 4, pp. 29–45, 1990.
- [40] K. Numpacharoen and A. Atsawarungruangkit, "Generating correlation matrices based on the boundaries of their coefficients," *PLoS ONE*, vol. 7, pp. 489–502, Nov. 2012.
- [41] 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.
- [42] P. Tichavský and Z. Koldovský, "Weight adjusted tensor method for blind separation of underdetermined mixtures of nonstationary sources," *IEEE Trans. Signal Process.*, vol. 59, no. 3, pp. 1037–1047, 2011.
- [43] 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.
- [44] E. Acar, D. M. Dunlavy, and T. G. Kolda, "A scalable optimization approach for fitting canonical tensor decompositions," *J. Chemometr.*, vol. 25, no. 2, pp. 67–86, February 2011.
- [45] 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.
- [46] A.-H. Phan, P. Tichavský, and A. Cichocki, "Low complexity damped Gauss-Newton algorithms for CANDECOMP/PARAFAC," *SIAM J. Matrix Anal. Appl.*, vol. 34, no. 1, pp. 126–147, 2013.



**Anh-Huy Phan** (M'15) received the master 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 CYBERNETICS, 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.