DEFLATION METHOD FOR CANDECOMP/PARAFAC TENSOR DECOMPOSITION

Anh-Huy Phan[‡], Petr Tichavský[•], Andrzej Cichocki[‡†],*

[‡]Brain Science Institute, RIKEN, Wakoshi, Japan
[•]Institute of Information Theory and Automation, Prague, Czech Republic

ABSTRACT

CANDECOMP/PARAFAC tensor decomposition (CPD) approximates multiway data by 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 in which rank R does not exceed the tensor dimensions. A rank-R CPD can be performed through $(R-1)$ rank-1 reductions. At each deflation stage, the residue tensor is constrained to have a reduced multilinear rank.

Index Terms— tensor decomposition, CANDECOMP/PARAFAC, deflation, rank-1 reduction

1. INTRODUCTION

CANDECOMP/PARAFAC (CP) has found numerous applications in wide variety of areas such as in chemometrics [1], telecommunication [2], data mining [3, 4], neuroscience [5], separated representations [6]. The CP decomposition has been commonly used over the years because of its simplicity.

For a matrix of rank-R, one can subtract the best rank-1 representation from a matrix to reduce its rank [7]. It has advantage in sequential extraction such as in SVD. Unfortunately, this property in general does not hold for multiway arrays. We can only successively perform supersymmetric rank-1 decomposition of supersymmetric tensors [8]. However, for general tensors, we cannot obtain a good rank-R tensor approximation through R sequential rank-1 estimations [9]. The authors in [10] confirmed that subtracting rank-1 tensor from a tensor may increase its rank.

In this paper we present a novel method for rank-reduction problem in 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-$(R-1)$. The method is verified by an algorithm called rank reduction algorithm for CPD (summarized in Algorithm 2).

Throughout the paper, tensors are denoted by bold calligraphic letters, e.g., $\mathcal{A} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$, matrices by bold capital letters, e.g., $\mathbf{A} = [\boldsymbol{a}_1, \boldsymbol{a}_2, \ldots, \boldsymbol{a}_R] \in \mathbb{R}^{I \times R}$, and vectors by bold italic letters, e.g., $\boldsymbol{a}_j$. The mode-n matricization of tensor $\mathcal{Y}$ is a matrix $\mathbf{Y}_{(n)}$ of size $I_n \times (\prod_{k \neq n} I_k)$ whose j-th row is vectorization of sub-tensor of $\mathcal{Y}$ with mode-n index $i_n = j$ [19].

Definition 1. (Kruskal form (tensor)) *A tensor $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$ is in Kruskal form if*

$$\begin{aligned} \mathcal{X} &= \sum_{r=1}^{R} \lambda_r \, \boldsymbol{a}_r^{(1)} \circ \boldsymbol{a}_r^{(2)} \circ \cdots \circ \boldsymbol{a}_r^{(N)}, \qquad (1) \\ &\triangleq [\![\boldsymbol{\lambda}; \mathbf{A}^{(1)}, \mathbf{A}^{(2)}, \ldots, \mathbf{A}^{(N)}]\!], \quad \boldsymbol{\lambda} = [\lambda_1, \lambda_2, \ldots, \lambda_R]. \end{aligned}$$

where "$\circ$" denotes the outer product, $\mathbf{A}^{(n)} = [\boldsymbol{a}_1^{(n)}, \boldsymbol{a}_2^{(n)}, \ldots, \boldsymbol{a}_R^{(n)}] \in \mathbb{R}^{I_n \times R}$, $(n = 1, 2, \ldots, N)$ are factor matrices, $\|\boldsymbol{a}_r^{(n)}\|_2 = 1$, for all r and n, and $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_R > 0$.

Definition 2. (Multilinear rank-$(R_1, R_2, \ldots, R_N)$ (Tucker form)) *A tensor $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$ has multilinear rank-$(R_1, R_2, \ldots, R_N)$ if $\operatorname{rank}(\mathbf{X}_{(n)}) = R_n \leq I_n$ for $n = 1, \ldots, N$. $\mathcal{X}$ can be expressed in the Tucker form as*

$$\begin{aligned} \mathcal{X} &= \sum_{r_1=1}^{R_1} \sum_{r_2=1}^{R_2} \cdots \sum_{r_n=1}^{R_N} g_{r_1 r_2 \ldots r_N} \, \boldsymbol{a}_{r_1}^{(1)} \circ \boldsymbol{a}_{r_2}^{(2)} \circ \cdots \circ \boldsymbol{a}_{r_N}^{(N)}, \quad (2) \\ &\triangleq [\![\mathcal{G}; \mathbf{A}^{(1)}, \mathbf{A}^{(2)}, \ldots, \mathbf{A}^{(N)}]\!], \qquad (3) \end{aligned}$$

where $\mathcal{G} = [g_{r_1 r_2 \ldots r_N}] \in \mathbb{R}^{R_1 \times R_2 \times \cdots \times R_N}$, and $\mathbf{A}^{(n)}$ are of full column rank.

Definition 3. (CANDECOMP/PARAFAC (CP) [11, 12]) *Approximation of an order-N data tensor $\mathcal{Y} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$ by a rank-R tensor in the Kruskal form means $\mathcal{Y} = \widehat{\mathcal{Y}} + \mathcal{E}$, where $\widehat{\mathcal{Y}} = [\![\boldsymbol{\lambda}; \{\mathbf{A}^{(n)}\}]\!]$, so that $\|\mathcal{Y} - \widehat{\mathcal{Y}}\|_F^2$ is minimized.*

For compact expression, $[\![\boldsymbol{\lambda}; \{\mathbf{A}^{(n)}\}]\!]$ denotes a Kruskal tensor, where $[\![\mathbf{G}; \{\mathbf{A}^{(n)}\}]\!]$ representes a Tucker tensor. It is worth noting that CP solution may not exist [13–15].

2. SEQUENTIAL RANK-1 TENSOR EXTRACTION AND COMPRESSION

In this paper we consider order-N tensor $\mathcal{Y}$ of size $I_1 \times I_2 \times \cdots \times I_N$ 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

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

*The work of P. Tichavsky was supported by the Czech Science Foundation through project No. 14–13713S.

†Also affiliated with Systems Research Institute, Polish Academy of Science, Poland.

978-1-4799-2893-4/14/$31.00 ©2014 IEEE

Note that tensor $\mathcal{Y}_2$ has rank $(R-1)$, and therefore its multilinear rank is at most $(R-1, ..., R-1)$. We can write

$$\mathcal{Y} \approx [\![\lambda; \boldsymbol{a}^{(1)}, \boldsymbol{a}^{(2)}, \ldots, \boldsymbol{a}^{(N)}]\!] + [\![\mathcal{G}; \mathbf{U}^{(1)}, \mathbf{U}^{(2)}, \ldots, \mathbf{U}^{(N)}]\!] \quad (4)$$

where $\mathbf{U}^{(n)}$ are of size $I_n \times (R-1)$ and $\mathcal{G}$ is of size $(R-1) \times \cdots \times (R-1)$. The decomposition is a particular case of the block term decomposition (BTD) for order-3 tensors [16, 17]. In general, in the second stage of the rank-reduction, we only need to decompose the core tensor $\mathcal{G}$ of size $(R-1) \times (R-1) \times \cdots \times (R-1)$ into a rank-1 tensor and a multilinear tensor of size $(R-2) \times (R-2) \times \cdots \times (R-2)$. Hence, if $R \ll I_n$, after the first stage, dimensions of the tensor to be decomposed are significantly reduced due to compression by $\mathbf{U}^{(n)}$, and we no longer process the original data.

The procedure is sequentially applied $(R-1)$ times as described in Algorithm 1. In the last stage, the compressed tensor has size of $2 \times 2 \times \cdots \times 2$, and can be quickly decomposed by some traditional methods or by (un)folding CP algorithm [18]. The final solution is a Kruskal tensor whose factor matrices comprise components of rank-1 Kruskal tensors $[\![\lambda_r; \{\boldsymbol{a}_r^{(n)}\}]\!]$, $r = 1, \ldots, R$.

Lemma 1 (Orthogonal normalization)**.** *Given a best decomposition of $\mathcal{Y}$ into two tensors $[\![\lambda; \boldsymbol{a}^{(1)}, \boldsymbol{a}^{(2)}, \ldots, \boldsymbol{a}^{(N)}]\!]$ and $[\![\mathcal{G}; \mathbf{U}^{(1)}, \mathbf{U}^{(2)}, \ldots, \mathbf{U}^{(N)}]\!]$, where $\mathbf{U}^{(n)} \in \mathbb{R}^{I_n \times J_n}$, one can construct an equivalent decomposition, denoted by tildas, which has the same approximation error,*

- *$\widetilde{\mathbf{U}}^{(n)} = [\widetilde{\boldsymbol{u}}_r^{(n)}]$ are orthogonal, i.e., $(\widetilde{\mathbf{U}}^{(n)})^T \widetilde{\mathbf{U}}^{(n)} = \mathbf{I}_{J_n}$,*
- *$\boldsymbol{a}^{(n)}$ and $(J_n - 1)$ column vectors $\widetilde{\mathbf{U}}_{2:J_n}^{(n)}$ are mutually orthogonal, i.e., $(\widetilde{\mathbf{U}}_{2:J_n}^{(n)})^T \boldsymbol{a}^{(n)} = \mathbf{0}_{J_n - 1}$.*

Proof of Lemma 1 is given in Appendix. Hereinafter, we assume that such orthogonal normalization (and rotation with respect to $\boldsymbol{a}^{(n)}$) is applied to factor matrices $\mathbf{U}^{(n)}$.

2.1. Update factor matrices

From the decomposition in (4), we consider a cost function which minimizes the Frobenius norm between $\mathcal{Y}$ and its approximation

$$D = \frac{1}{2} \left\| \mathcal{Y} - [\![\lambda; \boldsymbol{a}^{(1)}, \ldots, \boldsymbol{a}^{(N)}]\!] - [\![\mathcal{G}; \mathbf{U}^{(1)}, \ldots, \mathbf{U}^{(N)}]\!] \right\|_F^2, \quad (5)$$

and derive the alternating least-squares algorithm which sequentially estimates $\boldsymbol{a}^{(n)}$ and $\mathbf{U}^{(n)}$ while fixing other parameters. The cost function is written as

$$D = \frac{1}{2} \left\| \mathbf{Y}_{(n)} - \lambda \, \boldsymbol{a}^{(n)} \left(\bigotimes_{k \neq n} \boldsymbol{a}^{(k)} \right)^T - \mathbf{U}^{(n)} \, \mathbf{G}_{(n)} \left(\bigotimes_{k \neq n} \mathbf{U}^{(n)} \right)^T \right\|_F^2, \quad (6)$$

where $\otimes$ denotes the Kronecker product, $\bigotimes_{k \neq n} \mathbf{U}^{(k)} = \mathbf{U}^{(N)} \otimes \cdots \otimes \mathbf{U}^{(n+1)} \otimes \mathbf{U}^{(n-1)} \otimes \cdots \otimes \mathbf{U}^{(1)}$, $\mathbf{Y}_{(n)} \in \mathbb{R}^{I_n \times \prod_{k \neq n} I_k}$ and $\mathbf{G}_{(n)} \in \mathbb{R}^{R_n \times \prod_{k \neq n} R_k}$ are mode-n matricizations of $\mathcal{Y}$ and $\mathcal{G}$, respectively. Due to the orthogonal normalization in Lemma 1, $\left(\bigotimes_{k \neq n} \mathbf{U}^{(k)} \right)^T \left(\bigotimes_{k \neq n} \boldsymbol{a}^{(k)} \right) = \bigotimes_{k \neq n} \left(\mathbf{U}^{(k)T} \boldsymbol{a}^{(k)} \right) = [\rho_n, 0, \ldots, 0]^T$, where $\rho_n = \prod_{k \neq n}^{N} (\boldsymbol{u}_1^{(k)T} \boldsymbol{a}^{(k)})$. Gradients of the cost function D in (6) w.r.t $\boldsymbol{a}^{(n)}$ and $\mathbf{U}^{(n)}$ take the forms

$$\begin{aligned}
\frac{\partial D}{\partial \boldsymbol{a}^{(n)}} &= -\lambda \, \mathbf{Y}_{(n)} \left(\bigotimes_{k \neq n} \boldsymbol{a}^{(k)} \right) + \lambda^2 \, \boldsymbol{a}^{(n)} \left(\bigotimes_{k \neq n} \boldsymbol{a}^{(k)} \right)^T \left(\bigotimes_{k \neq n} \boldsymbol{a}^{(k)} \right) \\
&\quad + \lambda \, \mathbf{U}^{(n)} \mathbf{G}_{(n)} \left(\bigotimes_{k \neq n} \mathbf{U}^{(k)} \right)^T \left(\bigotimes_{k \neq n} \boldsymbol{a}^{(k)} \right) \\
&= -\lambda \, \boldsymbol{s}_n + \lambda^2 \, \boldsymbol{a}^{(n)} + \lambda \, \rho_n \, \mathbf{U}^{(n)} \, \boldsymbol{g}_n \quad (7) \\
\frac{\partial D}{\partial \mathbf{U}^{(n)}} &= -\mathbf{Y}_{(n)} \left(\bigotimes_{k \neq n} \mathbf{U}^{(k)} \right) \mathbf{G}_{(n)}^T + \lambda \boldsymbol{a}^{(n)} \left(\bigotimes_{k \neq n} \left(\boldsymbol{a}^{(k)T} \, \mathbf{U}^{(k)} \right) \right) \mathbf{G}_{(n)}^T \\
&\quad + \mathbf{U}^{(n)} \mathbf{G}_{(n)} \left(\bigotimes_{k \neq n} \left(\mathbf{U}^{(k)T} \, \mathbf{U}^{(k)} \right) \right) \mathbf{G}_{(n)}^T \\
&= -\mathbf{T}_n \, \mathbf{G}_{(n)}^T + \lambda \rho_n \, \boldsymbol{a}^{(n)} \, \boldsymbol{g}_n^T + \mathbf{U}^{(n)} \, \mathbf{G}_{(n)} \, \mathbf{G}_{(n)}^T, \quad (8)
\end{aligned}$$

where $\boldsymbol{g}_n = \mathbf{G}_{(n)}(:, 1)$ is the first column of $\mathbf{G}_{(n)}$, $\boldsymbol{s}_n = \mathbf{Y}_{(n)} \left(\bigotimes_{k \neq n} \boldsymbol{a}^{(k)} \right)$ and $\mathbf{T}_n = \mathbf{Y}_{(n)} \left(\bigotimes_{k \neq n} \mathbf{U}^{(k)} \right)$. By setting the gradients (7) and (8) to zeros, we derive the update rules

$$\begin{aligned}
\mathbf{U}^{(n)} &\leftarrow \left(\mathbf{T}_{(n)} \, \mathbf{G}_{(n)}^T - \rho_n \, \boldsymbol{s}_n \, \boldsymbol{g}_n^T \right) \left(\mathbf{G}_{(n)} \, \mathbf{G}_{(n)}^T - \rho_n^2 \, \boldsymbol{g}_n \, \boldsymbol{g}_n^T \right)^{-1}, \\
\boldsymbol{a}^{(n)} &\leftarrow \frac{1}{\lambda} \left(\boldsymbol{s}_n - \rho_n \, \mathbf{U}^{(n)} \, \boldsymbol{g}_n \right).
\end{aligned}$$

The algorithm may face numerical instability when $\rho_n = 1$, which implies that $\boldsymbol{a}^{(k)}$ and $\boldsymbol{u}_1^{(k)}$ are identical for all $k \neq n$. For such a case, the decomposition should be restarted with lower multilinear rank for the second block. We can see that optimization with respect to $\mathbf{U}^{(n)}$ and $\boldsymbol{a}^{(n)}$ for one specific n is performed in closed form. The factor matrices are then normalized as in Lemma 1. The algorithm proceeds by cyclically repeating the above procedure for all $n = 1, ..., N$, as shown in Algorithm 2. Parameters $\boldsymbol{a}^{(n)}$ and $\mathbf{U}^{(n)}$ can be initialized, for example, using the generalized rank annihilation method for (higher order) CPD [18], or by the best rank-1 approximation of $\mathcal{Y}$ and the best multilinear rank-$(R-1, \ldots, R-1)$ tensor approximation of the residue $\mathcal{Y} - [\![\lambda; \boldsymbol{a}^{(1)}, \ldots, \boldsymbol{a}^{(N)}]\!]$ [20]. A more efficient initialization is based on a recently developed tensor diagonalization tool [25].

2.2. Update core tensor

In practice, we may not need to update $\mathcal{G}$ and λ explicitly because they are modified due to normalization and rotation of factors $\boldsymbol{a}^{(n)}$ and $\mathbf{U}^{(n)}$ at each iteration. However, for completeness, we present the update rules for $\mathcal{G}$ and λ. Denote $s = \left(\bigotimes_n \boldsymbol{a}^{(n)} \right)^T \text{vec}(\mathcal{Y})$, $\boldsymbol{t} = \left(\bigotimes_n \mathbf{U}^{(n)} \right)^T \text{vec}(\mathcal{Y})$ and $\rho = \prod_{n=1}^{N} (\boldsymbol{u}_1^{(n)T} \boldsymbol{a}^{(n)})$. Setting the gradients of the cost function w.r.t λ and $\mathcal{G}$ to zero

$$\begin{aligned}
\frac{\partial D}{\partial \lambda} &= -s + \lambda + \rho \, g_1 = 0, \\
\frac{\partial D}{\partial g_i} &= -t_i + \lambda \rho \, \delta_{i1} + g_i = 0, \quad i = 1, \ldots, R_1 R_2 \cdots R_N,
\end{aligned}$$

Algorithm 1: Sequential Rank-1 Reduction

Input: Data tensor $\mathcal{Y}$: $(I_1 \times I_2 \times \cdots \times I_N)$, rank R
Output: A rank-R Kruskal tensor $\mathcal{X}$
begin
$\mathbf{V}^{(n)} = \mathbf{I}_{I_n}$ for $n = 1, 2, \ldots, N$
for $r = 1, 2, \ldots, R-1$ **do**
$[[\lambda_r; \{\boldsymbol{a}_r^{(n)}\}], [\mathcal{G}_r; \{\mathbf{U}_r^{(n)}\}]] =$ Rank-1-Reduction$(\mathcal{G}_{r-1})$
/* $\mathcal{G}_0 = \mathcal{Y}$ */
for $n = 1, \ldots, N$ **do**
$\boldsymbol{a}_r^{(n)} \leftarrow \mathbf{V}^{(n)} \boldsymbol{a}_r^{(n)}$, $\mathbf{V}^{(n)} \leftarrow \mathbf{V}^{(n)} \mathbf{U}^{(n)}$
$\mathcal{X} = [\![\lambda; \{\mathbf{A}^{(n)}\}]\!]$ /* $\lambda = [\lambda_1, \ldots, \lambda_R], \mathbf{A}^{(n)} = [\boldsymbol{a}_1^{(n)}, \ldots, \boldsymbol{a}_R^{(n)}]$, $\lambda_R \triangleq \mathcal{G}_{R-1}$, $\boldsymbol{a}_R^{(n)} \triangleq \mathbf{U}_{R-1}^{(n)}$ */

Algorithm 2: Rank-1 Reduction

Input: Data tensor $\mathcal{Y}$: $(I_1 \times I_2 \times \cdots \times I_N)$, rank R
Output: A rank-1 tensor $[\![\lambda; \{\boldsymbol{a}^{(n)}\}]\!]$ and multilinear rank tensor $[\![\mathcal{G}; \{\mathbf{U}^{(n)}\}]\!]$
begin
Initialize $[\![\lambda; \{\boldsymbol{a}^{(n)}\}]\!]$ and $[\![\mathcal{G}; \{\mathbf{U}^{(n)}\}]\!]$
repeat
for $n = 1, 2, \ldots, N$ **do**
$\mathbf{U}^{(n)} \leftarrow \left(\mathbf{T}_{(n)}\mathbf{G}_{(n)}^T - \rho_n \boldsymbol{s}_n \boldsymbol{g}_n^T\right)\left(\mathbf{G}_{(n)}\,\mathbf{G}_{(n)}^T - \rho_n^2\, \boldsymbol{g}_n\, \boldsymbol{g}_n^T\right)^{-1}$
$\boldsymbol{a}^{(n)} \leftarrow \left(\boldsymbol{s}_n - \rho_n \mathbf{U}^{(n)}\, \boldsymbol{g}_n\right)$
$\lambda \leftarrow \|\boldsymbol{a}^{(n)}\|$, $\boldsymbol{a}^{(n)} \leftarrow \frac{\boldsymbol{a}^{(n)}}{\|\boldsymbol{a}^{(n)}\}}$
QR-decomposition: $\mathbf{U}^{(n)} = \mathbf{Q}_n \mathbf{Z}_n$
Construct $\mathbf{\Gamma}_n$ from $\mathbf{Q}_n{}^T \boldsymbol{a}^{(n)}$ as in Appendix
$\mathbf{U}^{(n)} \leftarrow \mathbf{Q}_n \mathbf{\Gamma}_n$, $\mathcal{G} \leftarrow \mathcal{G} \times_n \mathbf{\Gamma}_n^T \mathbf{Z}_n$
until *a stopping criterion is met*

where g_i is i-th entry of $\text{vec}(\mathcal{G})$ with the linear index i, and $\delta_{i,1}$ is the Kronecker delta, we get update rules

$$\lambda \leftarrow \frac{s - \rho\, t_1}{1 - \rho^2}, \qquad g_i \leftarrow t_i - \lambda \rho\, \delta_{i,1}. \tag{9}$$

3. SIMULATIONS

3.1. An Example for Rank-3 $3 \times 3 \times 3$ Tensor

In this first example, we decompose a simple tensor of size $3 \times 3 \times 3$ whose mode-1 unfolding is given by

$$\mathbf{Y}_{(1)} = \left[\begin{array}{ccc|ccc|ccc} 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1 & 0 & 0 & 1 & 0 \end{array}\right].$$

This tensor has rank-3 CPD with factor matrices given by

$$\mathbf{A} = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 1 \\ 1 & -1 & 0 \end{bmatrix}, \quad \mathbf{B} = \begin{bmatrix} 1 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad \mathbf{C} = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 0 & 0 \\ 1 & 1 & 0 \end{bmatrix}$$

and $\lambda_1 = \lambda_2 = \lambda_3 = 1$.

Let $\alpha = \dfrac{3\pi}{8}$, $\boldsymbol{a} = [\cos(\alpha), \sin(\alpha), 0]^T$, $\boldsymbol{b} = [1, 0, 1]^T$, $\boldsymbol{c} = [\sin(\alpha), 0, \cos(\alpha)]^T$, and $\lambda = \dfrac{\sqrt{2}}{2}(1 - \cos(2\alpha) + \sin(2\alpha))$. The rank-1 tensor $[\![\lambda; \boldsymbol{a}, \boldsymbol{b}, \boldsymbol{c}]\!]$ has an approximation error $\|\mathcal{Y} - [\![\lambda; \boldsymbol{a}, \boldsymbol{b}, \boldsymbol{c}]\!]\|_F^2 = \frac{13}{2} - \sqrt{2} < \min_r \|\mathcal{Y} - [\![\boldsymbol{a}_r, \boldsymbol{b}_r, \boldsymbol{c}_r]\!]\|_F^2 = \min\{6, 8, 6\}$, where $\boldsymbol{a}_r$, $\boldsymbol{b}_r$, $\boldsymbol{c}_r$, $r = 1, 2, 3$ are columns of $\mathbf{A}$, $\mathbf{B}$ and $\mathbf{C}$, respectively. This implies that extraction of the best rank-1 tensor approximation to $\mathcal{Y}$ could not return any rank-1 tensor among $[\![\boldsymbol{a}_r, \boldsymbol{b}_r, \boldsymbol{c}_r]\!]$ for $r = 1, 2, 3$. Notice $[\![\lambda; \boldsymbol{a}, \boldsymbol{b}, \boldsymbol{c}]\!]$ is not the best rank-1 approximation to $\mathcal{Y}$.

When applying the rank-1 reduction algorithm to $\mathcal{Y}$, after the first run, this algorithm extracts accurately the rank-1 tensor $[\![\boldsymbol{a}_3, \boldsymbol{b}_3, \boldsymbol{c}_3]\!]$, and returns a residue tensor

$$\mathbf{X}_{(1)} = \left[\begin{array}{ccc|ccc|ccc} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1 & 0 & 0 & 1 & 0 \end{array}\right].$$

After two runs, we obtained the full factor matrices of $\mathcal{Y}$.

3.2. Decomposition of Rank-4 $3 \times 3 \times 3$ Tensor

In this example, we will show that the proposed deflation method allows to decompose tensors of a rank that slightly exceeds the tensor dimension. We considered a $3 \times 3 \times 3$ tensor $\mathcal{Y}$ whose mode-1 unfolding is given by

$$\mathbf{Y}_{(1)} = \left[\begin{array}{ccc|ccc|ccc} 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{array}\right].$$

The tensor has rank-4, whereas its border rank is 3. Numerical approximation of this tensor by CP decomposition with rank 3 would lead to diverging factors. Our rank-1 reduction, however, does not have such problems. The rank-1 reduction algorithm factorized the data with rank $R = 3$. After the first deflation stage, we obtain a rank-1 tensor $\boldsymbol{a}_1 \circ \boldsymbol{b}_1 \circ \boldsymbol{c}_1$ where $\boldsymbol{a}_1 = \boldsymbol{b}_1 = \boldsymbol{c}_1 = [0, 0, 1]^T$, and a rank-(2,2,2) tensor $[\![\mathcal{G}; \mathbf{U}, \mathbf{V}, \mathbf{W}]\!]$ where

$$\mathbf{G}_{(1)} = \left[\begin{array}{cc|cc} 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 \end{array}\right], \quad \mathbf{U} = \mathbf{V} = \mathbf{W} = \begin{bmatrix} 0 & 1 \\ 1 & 0 \\ 0 & 0 \end{bmatrix}.$$

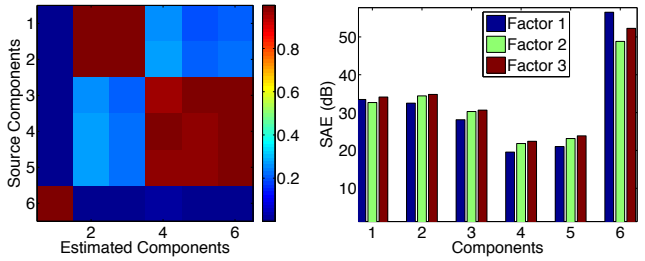
After two deflation stages, we obtain a rank-3 Kruskal tensor $[\![\boldsymbol{\lambda}; \mathbf{A}, \mathbf{B}, \mathbf{C}]\!]$ where $\boldsymbol{\lambda} = [1, 1, 1]^T$ and

$$\mathbf{A} = \begin{bmatrix} 0 & 1 & 1 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \end{bmatrix}, \mathbf{B} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix}, \mathbf{C} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix}.$$

The residue $\mathcal{E} = \mathcal{Y} - [\![\boldsymbol{\lambda}\ \mathbf{A}, \mathbf{B}, \mathbf{C}]\!]$ can be approximated by a rank-1 tensor $\mathcal{E} = \boldsymbol{a}_4 \circ \boldsymbol{b}_4 \circ \boldsymbol{c}_4$, where $\boldsymbol{a}_4 = [0, 1, 0]$, $\boldsymbol{b}_4 = [1, 0, 0]$, $\boldsymbol{c}_4 = [0, 1, 0]$. Finally, $\mathcal{Y}$ has a rank-4 CPD $\mathcal{Y} = [\![\mathbf{1}_4; [\mathbf{A}, \boldsymbol{a}_4], [\mathbf{B}, \boldsymbol{b}_4], [\mathbf{C}, \boldsymbol{c}_4]]\!]$.

3.3. Decomposition of Tensor with Diverging Components

In this section, we decomposed a $6 \times 6 \times 6$ tensor whose factor matrices are introduced in [21]. Factor matrices have



(a) Correlation degree. (b) Squared angular error.

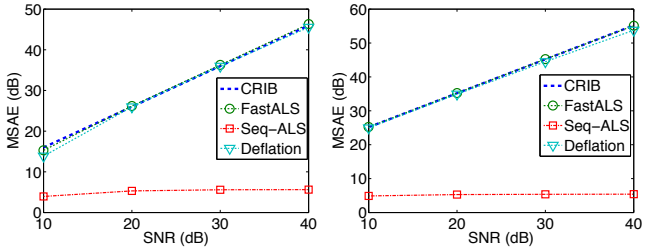
Fig. 1. Performance of the rank-one reduction in decomposition of a $6 \times 6 \times 6$ tensor.

one nondiverging component, and two groups of diverging components with their mutual angle less than 2.7^o: a group of the 1st and 2nd components, and a group of the 3rd, 4th and 5th components. The tensor was corrupted with additive Gaussian noise of signal-to-noise ratio SNR $= -10 \log_{10} \frac{\|\mathcal{Y}\|_F^2}{\sigma^2 \prod I_n} = 50$ dB, where σ^2 denotes the noise variance, and $\|\mathcal{Y}\|_F$ is the Frobenius norm of $\mathcal{Y}$. The estimated components were evaluated through the squared angular errors $SAE(\boldsymbol{a}, \hat{\boldsymbol{a}}) = -10 \log_{10} \left(\text{acos} \frac{\boldsymbol{a}^T \hat{\boldsymbol{a}}}{\|\boldsymbol{a}\| \|\hat{\boldsymbol{a}}\|}\right)^2$ (dB) [22–24] which are shown in Fig. 1(b). In addition, in Fig. 1(a), we show correlation between the original rank-one tensor $\boldsymbol{a}_r \circ \boldsymbol{b}_r \circ \boldsymbol{c}_r$ and the estimated one $\hat{\boldsymbol{a}}_s \circ \hat{\boldsymbol{b}}_s \circ \hat{\boldsymbol{c}}_s$. Fig. 1 indicates that 6-th components were first estimated with high accuracy $SAE > 50$ dB, then diverging components 1, 2, 3, 4, 5 with lower SAE. Thus, the proposed method helps to distinguish the stable components from the unstable ones.

3.4. Decomposition of Random Tensors

We illustrate the performance of the rank-reduction algorithm through decomposition of synthetic data of size $20 \times 20 \times 20$ and $50 \times 50 \times 50$. Factor matrices $\mathbf{A}^{(n)}$ had rank of $R = 10$, and were randomly generated such that their collinearity coefficients $c_{r,s} = \frac{\boldsymbol{a}_r^{(n)T} \boldsymbol{a}_s^{(n)}}{\|\boldsymbol{a}_r^{(n)}\| \|\boldsymbol{a}_s^{(n)}\|}$ were given in the ranges of [0.2, 0.8], [0.4, 0.8], [0.7, 0.995], see, e.g., [18, 26]. The tensors were corrupted with additive Gaussian noise of signal-to-noise ratio SNR = 10, 20, 30 and 40 dB. We generated 20 Kruskal tensors with $\lambda_1 = \cdots = \lambda_R = 1$, and added an i.i.d Gaussian noise to achieve the specified SNR level. Therefore, there were in total 1200 noisy tensors to be decomposed for each size.

The performance of the decomposition was evaluated through the squared angular errors, and compared with the Cramér-Rao induced bound CRIB($\boldsymbol{a}$) in decibels (dB) [22–24]. The mean SAE (MSAE) was averaged over ($1200 \times R$) SAEs of all components in all factor matrices. Fig. 2 illustrates the SAEs achieved by FastALS [27], SeqALS which sequentially extracts rank-1 tensor and the proposed algorithm. Algorithms stop when differences of successive relative errors $\varepsilon = \frac{\|\mathcal{Y} - \widehat{\mathcal{Y}}\|_F}{\|\mathcal{Y}\|_F}$ were lower than 10^{-8}, or until



(a) Tensor size $20 \times 20 \times 20$. (b) Tensor size $50 \times 50 \times 50$.

Fig. 2. Comparison of SAE (dB) of CPD algorithms with CRIB in decomposition of noisy tensors

the maximum number of iterations (1000) was achieved. In Fig. 2, SAE of the proposed algorithm attained the CRIB for low and high SNR levels, while SeqALS completely failed.

4. CONCLUSIONS

We proposed a novel deflation method for CPD, and confirmed its correctness through simulations. The method also plays key role as tool for data compression after the first deflation stage, since the data to be decomposed in next stages is of size $(R-1) \times (R-1) \times \cdots \times (R-1)$, which might be much smaller than the original data. The method may find application in tracking environment, when tensor is sequentially updated - hence, there is no need to track the whole decomposition but only one or a few factors. The deflation algorithm is implemented in the Matlab package TENSORBOX which is available online at: `http://www.bsp.brain.riken.jp/~phan/tensorbox.php`.

Appendix: Proof of Lemma 1

Proof. Let $\mathbf{Q}_n$ be a matrix comprising an orthogonal basis of columnspace of $\mathbf{U}^{(n)}$, $\boldsymbol{\gamma}_n = \mathbf{Q}_n{}^T \boldsymbol{a}^{(n)} \in \mathbb{R}^{R-1}$ and let $\boldsymbol{\Gamma}_n$ be an $(R-1) \times (R-1)$ orthogonal matrix, having the first column $\boldsymbol{\gamma}_n / \|\boldsymbol{\gamma}_n\|$, and the remaining columns form an orthogonal basis of the orthogonal complement of $\boldsymbol{\gamma}_n$. Note that $\mathbf{Q}_n$ and $\boldsymbol{\Gamma}_n$ can be obtained by QR decomposition of $\mathbf{U}^{(n)}$ and $\boldsymbol{\gamma}_n$, respectively, or by singular value decomposition of these matrices.

Then, the new decomposition of the latter term, $[\![\mathcal{G}; \mathbf{U}^{(1)}, \mathbf{U}^{(2)}, \ldots, \mathbf{U}^{(N)}]\!] = [\![\widetilde{\mathcal{G}}; \widetilde{\mathbf{U}}^{(1)}, \widetilde{\mathbf{U}}^{(2)}, \ldots, \widetilde{\mathbf{U}}^{(N)}]\!]$ can be defined through

$$\widetilde{\mathbf{U}}^{(n)} = \mathbf{Q}_n \boldsymbol{\Gamma}_n, \qquad n = 1, \ldots, N \tag{10}$$

$$\widetilde{\mathcal{G}} = \mathcal{G} \times_1 (\boldsymbol{\Gamma}_1 \mathbf{Q}_1^T \mathbf{U}^{(1)}) \cdots \times_N (\boldsymbol{\Gamma}_N \mathbf{Q}_N^T \mathbf{U}^{(N)}), \tag{11}$$

where $\times_n$ represents mode-n multiplication of a tensor and a matrix. It can be verified that $\widetilde{\mathbf{U}}^{(n)}$ are orthogonal and

$$(\widetilde{\mathbf{U}}^{(n)})^T \boldsymbol{a}^{(n)} = \boldsymbol{\Gamma}_n^T \mathbf{Q}_n^T \boldsymbol{a}^{(n)} = \boldsymbol{\Gamma}_n^T \boldsymbol{\gamma}_n = \|\boldsymbol{\gamma}_n\|_2 \boldsymbol{e}_1, \tag{12}$$

where $\boldsymbol{e}_1 = [1, 0, \ldots, 0]^T \in \mathbb{R}^{R-1}$. It means that the new decomposition obeys the required orthogonality condition. □

5. REFERENCES

[1] C.M. Andersson and R. Bro, "Practical aspects of PARAFAC modelling of fluorescence excitation-emission data," *Journal of Chemometrics*, vol. 17, pp. 200–215, 2003.

[2] 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., vol. 2, chapter 4. Prentice-Hall, Upper Saddle River, NJ, USA, 2000.

[3] A. Shashua and T. Hazan, "Non-negative tensor factorization with applications to statistics and computer vision," in *Proc. of the 22-th International Conference on Machine Learning (ICML)*, Bonn, Germany, 2005, pp. 792–799, ICML.

[4] A.-H. Phan and A. Cichocki, "Tensor decompositions for feature extraction and classification of high dimensional datasets," *Nonlinear Theory and Its Applications, IEICE*, vol. 1, pp. 37–68 (invited paper), 2010.

[5] H. Becker, P. Comon, L. Albera, M. Haardt, and I. Merlet, "Multi-way space-time-wave-vector analysis for EEG source separation," *Signal Processing*, vol. 92, no. 4, pp. 1021–1031, 2012.

[6] D. González, A. Ammar, F. Chinesta, and E. Cueto, "Recent advances on the use of separated representations," *International Journal for Numerical Methods in Engineering*, vol. 81, no. 5, pp. 637–659, 2010.

[7] M.T. Chu, R.E. Funderlic, and G.H. Golub, "A rank-one reduction formula and its applications to matrix factorizations," *SIAM review*, vol. 37, no. 4, pp. 512–530, 1995.

[8] Y. Wang and L. Qi, "On the successive supersymmetric rank-1 decomposition of higher-order supersymmetric tensors," *Numerical Linear Algebra with Applications*, vol. 14, no. 6, pp. 503–519, 2007.

[9] 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.

[10] A. Stegeman and P. Comon, "Subtracting a best rank-1 approximation may increase tensor rank," *Linear Algebra and its Applications*, vol. 433, no. 7, pp. 1276–1300, Dec. 2010.

[11] 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.

[12] 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.

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

[14] V. de Silva and L.-H. Lim, "Tensor rank and the ill-posedness of the best low-rank approximation problem," *SIAM J. Matrix Anal. Appl.*, vol. 30, no. 3, pp. 1084–1127, Sept. 2008.

[15] L.-H. Lim and P. Comon, "Blind multilinear identification," *IEEE Transactions on Information Theory*, vol. 60, no. 2, pp. 1260–1280, Sept. 2014.

[16] L. De Lathauwer, "Decompositions of a higher-order tensor in block terms – Part I: Lemmas for partitioned matrices," *SIAM Journal of Matrix Analysis and Applications*, vol. 30, no. 3, pp. 1022–1032, 2008, Special Issue on Tensor Decompositions and Applications.

[17] L. De Lathauwer and D. Nion, "Decompositions of a higher-order tensor in block terms – Part III: Alternating least squares algorithms," *SIAM Journal of Matrix Analysis and Applications*, vol. 30, no. 3, pp. 1067–1083, 2008, Special Issue Tensor Decompositions and Applications.

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

[19] 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*, Wiley, Chichester, 2009.

[20] 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 Journal of Matrix Analysis and Applications*, vol. 21, no. 4, pp. 1324–1342, 2000.

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

[22] P. Tichavský and Z. Koldovský, "Stability of CANDECOMP-PARAFAC tensor decomposition," in *ICASSP*, 2011, pp. 4164–4167.

[23] P. Tichavský and Z. Koldovský, "Weight adjusted tensor method for blind separation of underdetermined mixtures of nonstationary sources," *IEEE Transactions on Signal Processing*, vol. 59, no. 3, pp. 1037–1047, 2011.

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

[25] P. Tichavský, A.-H. Phan, and A. Cichocki, "Tensor diagonalization-a new tool for PARAFAC and block-term decomposition," *arXiv preprint*, arXiv:1402.1673, 2014.

[26] K. Numpacharoen and A. Atsawarungruangkit, "Generating correlation matrices based on the boundaries of their coefficients," *PLoS ONE*, vol. 7, pp. 489–502, Nov. 2012.

[27] A.-H. Phan, P. Tichavský, and A. Cichocki, "Fast alternating LS algorithms for high order CANDECOMP/PARAFAC tensor factorizations," *Signal Processing, IEEE Transactions on*, vol. 61, no. 19, pp. 4834–4846, 2013.

[28] R. Bro, *Multi-way Analysis in the Food Industry - Models, Algorithms, and Applications*, Ph.D. thesis, University of Amsterdam, Holland, 1998.