CANDECOMP/PARAFAC Decomposition of High-Order Tensors Through Tensor Reshaping

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

***Abstract*—In general, algorithms for order-3 CANDECOMP/PARAFAC (CP), also coined canonical polyadic decomposition (CPD), are easy to implement and can be extended to higher order CPD. Unfortunately, the algorithms become computationally demanding, and they are often not applicable to higher order and relatively large scale tensors. In this paper, by exploiting the uniqueness of CPD and the relation of a tensor in Kruskal form and its unfolded tensor, we propose a fast approach to deal with this problem. Instead of directly factorizing the high order data tensor, the method decomposes an unfolded tensor with lower order, e.g., order-3 tensor. On the basis of the order-3 estimated tensor, a structured Kruskal tensor, of the same dimension as the data tensor, is then generated, and decomposed to find the final solution using fast algorithms for the structured CPD. In addition, strategies to unfold tensors are suggested and practically verified in the paper.**

***Index Terms*—Tensor factorization, canonical decomposition, PARAFAC, ALS, structured CPD, tensor unfolding, Cramér-Rao induced bound (CRIB), Cramér-Rao lower bound (CRLB).**

I. Introduction

CANDECOMP/PARAFAC [1], [2], also known as Canonical polyadic decomposition (CPD), is a common tensor factorization which has found applications such as in chemometrics [3], [4], telecommunication [5]–[8], time-varying EEG spectrum [9], [10], data mining [11], separated representations for generic functions involved in quantum mechanics or kinetic theory descriptions of materials [12]. Although the original decomposition and applications were developed for three-way data, the model was later widely extended to higher order tensors. For example, Constantine *et al.* [13] modeled the pressure measurements along the combustion chamber as order-6 tensors corresponding to the flight conditions—Mach number, altitude and angle of attack, and the wall temperatures in the combustor and the turbulence mode. Luciani *et al.* [14] factorized higher order tensors generated from characteristic function in blind identification of underdetermined mixtures. The work is extended to higher order structured CDP in [7]. In neuroscience, Mørup *et al.* [9] analyzed order-4 data constructed from EEG signals in the time-frequency domain. Order-5 tensors consisting of dictionaries × timeframes × frequency bins × channels × trials-subjects [15] built up from EEG signals were shown to give high performance in BCI based on EEG motor imagery. In object recognition (digits, faces, natural images), CPD was used to extract features from order-5 Gabor tensors including height × width × orientation × scale × images [15]. More applications and properties of CPD are discussed in [16].

In general, many CP algorithms for order-3 tensor can be straightforwardly extended to decompose higher order tensors. For example, there are numerous algorithms for CPD including the alternating least squares (ALS) algorithm [2], [1] with line search extrapolation methods [1], [17], [4], [18], [19], rotation [20] and compression [21], or all-at-once algorithms such as the OPT algorithm [22], the conjugate gradient algorithm for nonnegative CP [23], the PMF3, damped Gauss-Newton (dGN) algorithms [24], [4] and fast dGN [25], [26], or algorithms based on joint diagonalization problem [27], [28]. The fact is that the algorithms become more complicated, computationally demanding, and often not applicable to relatively large scale tensors. For example, complexity of gradients of the cost function with respect to factors grows linearly with the tensor order N. It has a computational cost of order $\mathcal{O}\left(NR\prod_{n=1}^{N} I_n\right)$ for a tensor of size $I_1 \times I_2 \times \cdots \times I_N$. More tensor unfoldings $\mathbf{Y}_{(n)}$ $(n = 2, 3, \ldots, N-1)$ means more time consuming due to accessing non-contiguous blocks of data entries and shuffling their orders in a computer. In addition, line search extrapolation methods [1], [29], [17], [4], [18] become more complicated, and demand high computational cost to build up and solve $(2N-1)$-order polynomials. The rotation method [20] needs to estimate N rotation matrices of size $R \times R$ with a whole complexity per iteration of order $\mathcal{O}(N^3R^6)$. Moreover, in practice, CPD algorithms may take a lot of iterations, and are very slow, when some factor matrices become nearly rank deficient [21]. In such cases, the optimal CP solution might not always exist [30]–[32].

By exploiting the uniqueness of CPD under mild conditions (see discussion on uniqueness conditions for order-3 CPD in [33], [34], [27] and for higher order CPD in [35]–[37]), and the relation of a tensor in the Kruskal form [38] and its unfolded tensor, we develop a fast approach for high order and relatively large-scale CPD. Instead of directly factorizing the high order data tensor, the approach decomposes an unfolded tensor in lower order, e.g., order-3 tensor.

Manuscript received November 15, 2012; revised March 21, 2013 and June 05, 2013; accepted June 06, 2013. Date of publication June 17, 2013; date of current version September 03, 2013. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Joseph Tabrikian. The work of P. Tichavský was supported by the Czech Science Foundation through project 102/09/1278.

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

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

A. Cichocki is with the Lab for Advanced Brain Signal Processing, Brain Science Institute, RIKEN, Wakoshi 351–0198, Japan and also with the Systems Research Institute, Polish Academy of Sciences, Warsaw 01–447, Poland (e-mail: cia@brain.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.2013.2269046

The idea of using a tensor reshape has been introduced for a fast decomposition of certain symmetric tensors. For example, in blind underdetermined mixture identification, Karfoul *et al.* [39] decomposed complex-valued order-$2N$ cumulant arrays based on decomposition of an order-3 tensor in connection to the Procrustes problem, and rank-one tensor approximation of several order-N tensors. For CPD with a columnwise orthonormal factor matrix, Sørensen *et al.* [40] proposed an iterative algorithm which estimates columns of nonorthogonal factor matrices from the best rank-1 approximations of order-$(N-1)$ tensors.

This paper presents the method in more general form together with a specific guide for selecting a folding strategy. We show that rank-one tensor approximation is sensitive to unfolding, and can cause a significant loss of accuracy when applying an inappropriate unfolding, or when noise level is high. In our propose method, after decomposition of an unfolded tensor, a structured Kruskal tensor of the same dimension of the data tensor is then generated, and decomposed to find the desired factor matrices using a fast ALS algorithm.

In addition, the method is supported by recently analytically computed Cramér-Rao Induced Bounds (CRIB) on attainable squared angular error of factors in the CP decomposition which has been proposed in [41]. The bound is valid under the assumption that the decomposed tensor is corrupted by additive Gaussian noise which is independently added to each tensor element. In this paper we use the results of [41] to design the tensor unfolding strategy which ensures as little deterioration of accuracy as possible. This strategy is then verified in the simulations.

The paper is organized as follows. Notation and the CANDECOMP/PARAFAC are briefly reviewed in Section II. The simplified version of the proposed algorithm is presented in Section III. Loss of accuracy is investigated in Section III, and an efficient strategy for tensor unfolding is summarized in Section III-E. For difficult scenario decomposition, we proposed a new algorithm in Section IV. Simulations are performed on random tensors and real-world dataset in Section VI. Section VII concludes the paper.

II. CANDECOMP/PARAFAC (CP) Decomposition

Throughout the paper, we shall denote tensors by bold calligraphic letters, e.g., $\boldsymbol{\mathcal{A}} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$, matrices by bold capital letters, e.g., $\mathbf{A} = [\boldsymbol{a}_1, \boldsymbol{a}_2, \ldots, \boldsymbol{a}_R] \in \mathbb{R}^{I \times R}$, and vectors by bold italic letters, e.g., $\boldsymbol{a}_j$ or $\boldsymbol{I} = [I_1, I_2, \ldots, I_N]$. A vector of integer numbers is denoted by colon notation such as $\boldsymbol{k} = i : j = [i, i+1, \ldots, j-1, j]$. For example, we denote $1 : n = [1, 2, \ldots, n]$. The Kronecker product, the Khatri-Rao (column-wise Kronecker) product, and the (element-wise) Hadamard product are denoted respectively by $\otimes, \odot, \circledast$ [38], [42].

Definition 2.1: (Kruskal Form (Tensor) [38]): A tensor $\boldsymbol{\mathcal{X}} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$ is in Kruskal form if

$$\boldsymbol{\mathcal{X}} = \sum_{r=1}^{R} \lambda_r \, \boldsymbol{a}_r^{(1)} \circ \boldsymbol{a}_r^{(2)} \circ \cdots \circ \boldsymbol{a}_r^{(N)}, \tag{1}$$

$$\triangleq \left[\left[\boldsymbol{\lambda}; \mathbf{A}^{(1)}, \mathbf{A}^{(2)}, \ldots, \mathbf{A}^{(N)}\right]\right], \quad \boldsymbol{\lambda} = [\lambda_1, \lambda_2, \ldots, \lambda_R] \tag{2}$$

where symbol "$\circ$" 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, $\boldsymbol{a}_r^{(n)T} \boldsymbol{a}_r^{(n)} = 1$, for all r and n, and $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_R > 0$.

Definition 2.2: (CANDECOMP/PARAFAC (CP) [1], [2]): Approximation of an order-N data tensor $\boldsymbol{\mathcal{Y}} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$ by a rank-R tensor in the Kruskal form means

$$\boldsymbol{\mathcal{Y}} = \widehat{\boldsymbol{\mathcal{Y}}} + \boldsymbol{\mathcal{E}}, \tag{3}$$

where $\widehat{\boldsymbol{\mathcal{Y}}} = [[\boldsymbol{\lambda}; \mathbf{A}^{(1)}, \mathbf{A}^{(2)}, \ldots, \mathbf{A}^{(N)}]]$, so that $\|\boldsymbol{\mathcal{Y}} - \widehat{\boldsymbol{\mathcal{Y}}}\|_F^2$ is minimized.

It is worth noting that, once again, CP solution may not exist [30], [31], [43], [8]. CPD uniqueness results hold only for an exact CPD, which has perfect fit. The latent components are only well estimated if the noise level is low. There are numerous algorithms for CPD including alternating least squares (ALS) or all-at-once optimization algorithms, or based on joint diagonalization. In general, most CP algorithms which factorize order-N tensor often face high computational cost due to computing gradients and (approximate) Hessian, line search and rotation. Table I summarizes complexities of major computations in popular CPD algorithms. Details on numerical complexity of ALS, ELS and Levenberg-Marquardt (LM) algorithm using QR factorization can also be found in [21]. Complexity per iteration of a CP algorithm can be roughly computed based on Table I. For example, the ALS algorithm with line search has a complexity of order $\mathcal{O}(NRJ + 2^N RJ + NR^3) = \mathcal{O}(2^N RJ + NR^3)$.

TABLE I
Complexities Per Iteration of Major Computations in CPD Algorithms. $J = \prod_{n=1}^{N} I_n$, $T = \sum_{n=1}^{N} I_n$

Computing Process	Complexity
Gradient[2], [1], [21]	$\mathcal{O}(NRJ)$
Fast gradient[47]	$\mathcal{O}(RJ)$
(Approximate) Hessian and its inverse[24], [4]	$\mathcal{O}\left(R^3T^3\right)$
Levenberg-Marquardt [21], [48]	$\mathcal{O}\left(JT^2R^2\right)$
Fast (approximate) Hessian and its inverse[26], [41], [49]	$\mathcal{O}\left(R^2T + N^3R^6\right)$ or $\mathcal{O}\left(R^2T + NR^6\right)$
Exact line search[1], [4], [18]	$\mathcal{O}\left(2^N RJ\right)$
Rotation[20]	$\mathcal{O}\left(N^3R^6\right)$

For easy reference we introduce here Tucker compression. This operation is sometimes used as a preprocessing step prior to the CPD, in order to reduce dimensionality of the problem and computational complexity [44], [21]. Tucker decomposition with orthonormal factor matrices $\mathbf{U}^{(n)}$ can be efficiently found using the HOSVD or HOOI algorithm [45].

Definition 2.3: (Tucker Compression or Decomposition (TD) [46]): Approximation of an order-N tensor $\boldsymbol{\mathcal{Y}}$ by a multilinear rank-$(R_1, R_2, \ldots, R_N)$ tensor $\widehat{\boldsymbol{\mathcal{Y}}}$ in the form

$$\widehat{\boldsymbol{\mathcal{Y}}} = \sum_{r_1=1}^{R_1} \sum_{r_2=1}^{R_2} \cdots \sum_{r_N=1}^{R_N} g_{r_1 r_2 \ldots r_N} \, \boldsymbol{u}_{r_1}^{(1)} \circ \boldsymbol{u}_{r_2}^{(2)} \circ \cdots \circ \boldsymbol{u}_{r_N}^{(N)} \tag{4}$$

$$= \left[\left[\boldsymbol{\mathcal{G}}\,; \mathbf{U}^{(1)}, \mathbf{U}^{(2)}, \ldots, \mathbf{U}^{(N)}\right]\right], \quad R_n \leq I_n, \text{ for all } n, \tag{5}$$

so that $\|\mathcal{Y} - \widehat{\mathcal{Y}}\|_F^2$ is minimized, where $\mathbf{U}^{(n)} = [\boldsymbol{u}_1^{(n)}, \boldsymbol{u}_2^{(n)}, \ldots, \boldsymbol{u}_{R_n}^{(n)}]$ are orthonormal matrices, and $\mathcal{G} = [g_{r_1 r_2 \ldots r_N}]$ is a core tensor of size $R_1 \times R_2 \times \cdots \times R_N$.

III. CPD of Unfolded Tensors

In order to deal with existing problems for high order and relatively large scale CPD, the following process is proposed:

1) Reduce order of the tensor $\mathcal{Y}$ to a lower order (e.g., order-3) through tensor unfolding $\mathcal{Y}_{[[\boldsymbol{l}]]}$ which is defined later in this section.
2) Approximate the unfolded tensor $\mathcal{Y}_{[[\boldsymbol{l}]]}$ by an order-3 tensor $\widehat{\mathcal{Y}}_{[[\boldsymbol{l}]]}$ in the Kruskal form. Dimensions of $\mathcal{Y}_{[[\boldsymbol{l}]]}$, which are relatively larger than rank R, can be reduced to R by the Tucker compression [21], [44] prior to CPD although it is not a lossless compression. In such case, we only need to decompose an $R \times R \times R$ dimensional tensor. Alternatively, the lossless compression using QR factorization can be applied to compress an $I \times J \times K$ dimensional tensor to be the size of $JK \times J \times K$ with $I > JK$ [50]. More simply, one can apply the Higher Order SVD in every mode [51], which is generally good enough in practice [21].
3) Estimate the desired components of the original tensor $\mathcal{Y}$ on basis of the tensor $\widehat{\mathcal{Y}}_{[[\boldsymbol{l}]]}$ in the Kruskal form.

The method is based on an observation that unfolding of a Kruskal tensor also yields a Kruskal tensor. Moreover due to uniqueness of CPD under "mild" conditions, the estimated components along the unfolded modes are often good approximates to components for the full tensor. In the sequel, we introduce basic concepts that will be used in the rest of this paper. Loss of accuracy in decomposition of the unfolded tensors is analyzed theoretically based on the CRIB.

Definition 3.1 (Reshaping): The reshape operator for a tensor $\mathcal{Y} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$ to a size specified by a vector $\boldsymbol{L} = [L_1, L_2, \ldots, L_M]$ with $\prod_{m=1}^{M} L_m = \prod_{n=1}^{N} I_n$ returns an order-M tensor $\mathcal{X}$, such that $\text{vec}(\mathcal{Y}) = \text{vec}(\mathcal{X})$, and is expressed as $\mathcal{X} = \texttt{reshape}(\mathcal{Y}, \boldsymbol{L}) \in \mathbb{R}^{L_1 \times L_2 \times \cdots \times L_M}$.

Definition 3.2 (Tensor Transposition [52]): If $\mathcal{A} \in \mathbb{R}^{I_1 \times \cdots \times I_N}$ and $\boldsymbol{p}$ is a permutation of $[1, 2, \ldots, N]$, then $\boldsymbol{p}$-transpose of $\mathcal{A}$ is defined by $\mathcal{A}^{\langle \boldsymbol{p} \rangle}(i_{p_1}, \ldots, i_{p_N}) = \mathcal{A}(i_1, \ldots, i_N)$, $\boldsymbol{1} \le \boldsymbol{i} \le \boldsymbol{I} = [I_1, I_2, \ldots, I_N]$.

Definition 3.3 (Generalized Tensor Unfolding): Reshaping a $\boldsymbol{p}$-transpose $\mathcal{Y}^{\langle \boldsymbol{p} \rangle}$ to an order-M tensor of size $\boldsymbol{L} = [L_1, L_2, \ldots, L_M]$ with $L_m = \prod_{k \in \boldsymbol{l}_m} I_k$, where $[\boldsymbol{l}_1, \boldsymbol{l}_2, \ldots, \boldsymbol{l}_M] \equiv [p_1, p_2, \ldots, p_N]$, $\boldsymbol{l}_m = [\boldsymbol{l}_m(1), \ldots, \boldsymbol{l}_m(L_m)]$

$$\mathcal{Y}_{[[\boldsymbol{l}]]} \triangleq \texttt{reshape}(\mathcal{Y}^{\langle \boldsymbol{p} \rangle}, \boldsymbol{L}), \qquad \boldsymbol{l} = [\boldsymbol{l}_1, \boldsymbol{l}_2, \ldots, \boldsymbol{l}_M]. \tag{6}$$

Remark 3.1:

- If $\boldsymbol{l} = [n, (1 : n-1, n+1 : N)]$, then $\mathcal{Y}_{[[\boldsymbol{l}]]} = \mathbf{Y}_{(n)}$ is mode-n unfolding.
- If $\mathcal{Y}$ is an order-4 tensor, then $\mathcal{Y}_{[[1,2,(3,4)]]}$ is an order-3 tensor of size $I_1 \times I_2 \times I_3 I_4$.
- If $\mathcal{Y}$ is an order-6 tensor, then $\mathcal{Y}_{[[(1,4),(2,5),(3,6)]]}$ is an order-3 tensor of dimension $I_1 I_4 \times I_2 I_5 \times I_3 I_6$.

We denote Khatri-Rao product of a set of matrices $\mathbf{U}^{(n)}$, $n = 1, 2, \ldots, N$, as $\bigodot_{n=1}^{N} \mathbf{U}^{(n)} \triangleq \mathbf{U}^{(N)} \odot \mathbf{U}^{(N-1)} \odot \cdots \odot \mathbf{U}^{(1)}$.

Algorithm 1: `rank-one FCP (R1FCP)`

Input: Data tensor $\mathcal{Y}$: $(I_1 \times I_2 \times \cdots \times I_N)$, rank R,
Unfolding $\boldsymbol{l} = [\boldsymbol{l}_1, \boldsymbol{l}_2, \ldots, \boldsymbol{l}_M]$ where $\boldsymbol{l}_m = [l_m(1), \ldots, l_m(K_m)]$, $M \ge 3$
Output: $\boldsymbol{\lambda} \in \mathbb{R}^N$, N matrices $\mathbf{A}^{(n)} \in \mathbb{R}^{I_n \times R}$
begin

```
% Stage 1: Optional compression of Y_[[l]] --------------
[[G, U^(1), ..., U^(M)]] = TD(Y_[[l]], min(I, R))
% Stage 2: CPD of G and back projection-----------------
[[λ; B^(1), ..., B^(M)]] = CPD(G, R)
for m = 1, 2, ..., M do  B^(m) ← U^(m) B^(m)
% Stage 3: Rank-one tensor approximations------------
for m = 1, 2, ..., M do
    for r = 1, 2, ..., R do
        B_r = reshape(b_r^(m), [I_{l_m(1)}, ..., I_{l_m(K_m)}])
        [[g; a_r^(l_m(1)), ..., a_r^(l_m(K_m))]] = TD(B_r, 1)
        λ_r ← λ_r g
```

TD($\mathcal{Y}$,$\boldsymbol{R}$): rank-$\boldsymbol{R}$ Tucker decomposition of order-N tensor $\mathcal{Y}$ where $\boldsymbol{R} = [R_1, R_2, \ldots, R_N]$.
$\widehat{\mathcal{Y}}$ = CPD($\mathcal{Y}$, R, $\mathcal{Y}_{init}$): approximates an order-N tensor or a tensor in the Kruskal form $\mathcal{Y}$ by a rank-R Kruskal tensor $\widehat{\mathcal{Y}}$ using initial values $\mathcal{Y}_{init}$.

Lemma 3.1: Unfolding of a rank-R tensor in the Kruskal form $\mathcal{Y} = [[\boldsymbol{\lambda}; \mathbf{A}^{(1)}, \mathbf{A}^{(2)}, \ldots, \mathbf{A}^{(N)}]]$ returns an order-M rank-R Kruskal tensor $\mathcal{Y}_{[[\boldsymbol{l}]]}$, $\boldsymbol{l} = [\boldsymbol{l}_1, \boldsymbol{l}_2, \ldots, \boldsymbol{l}_M]$, given by

$$\mathcal{Y}_{[[\boldsymbol{l}]]} = \left[\left[\boldsymbol{\lambda}; \mathbf{B}^{(1)}, \mathbf{B}^{(2)}, \ldots, \mathbf{B}^{(M)}\right]\right], \tag{7}$$

where $\mathbf{B}^{(m)} = \bigodot_{k \in \boldsymbol{l}_m} \mathbf{A}^{(k)} \in \mathbb{R}^{(\prod_{k \in \boldsymbol{l}_m} I_k) \times R}$ for $m = 1, 2, \ldots, M$.

Remark 3.2:

1) If $\boldsymbol{l} = [n, (1 : n-1, n+1 : N)]$, then $\mathcal{Y}_{[[\boldsymbol{l}]]} = \mathbf{Y}_{(n)} = \mathbf{A}^{(n)} \,\text{diag}(\boldsymbol{\lambda}) \left(\bigodot_{k \ne n} \mathbf{A}^{(k)}\right)^T$.
2) If $\boldsymbol{l} = [(1 : n), (n+1 : N)]$, then $\mathcal{Y}_{[[\boldsymbol{l}]]} = \left(\bigodot_{k=1}^{n} \mathbf{A}^{(k)}\right) \text{diag}(\boldsymbol{\lambda}) \left(\bigodot_{k=n+1}^{N} \mathbf{A}^{(k)}\right)^T$.
3) For an order-4 Kruskal tensor $\mathcal{Y}$, $\mathcal{Y}_{[[1,2,(3,4)]]} = [[\boldsymbol{\lambda}; \mathbf{A}^{(1)}, \mathbf{A}^{(2)}, \mathbf{A}^{(4)} \odot \mathbf{A}^{(3)}]]$.

Corollary 3.1: An order-K tensor $\boldsymbol{\mathcal{B}}_{m_r}$ of size $I_{l_{m1}} \times I_{l_{m2}} \times \cdots \times I_{l_{mK}}$, $\boldsymbol{l}_m = [l_{m1} \ldots l_{mK}]$ folded from the r-th column vector $\boldsymbol{b}_r^{(m)}$ of $\mathbf{B}^{(m)}$, i.e., $\text{vec}(\boldsymbol{\mathcal{B}}_{m_r}) = \boldsymbol{b}_r^{(m)}$ is a rank-1 tensor

$$\boldsymbol{\mathcal{B}}_{m_r} = \boldsymbol{a}_r^{(l_{m1})} \circ \boldsymbol{a}_r^{(l_{m2})} \circ \cdots \circ \boldsymbol{a}_r^{(l_{mK})}. \tag{8}$$

In practice for real data, folded tensors $\boldsymbol{\mathcal{B}}_{m_r}$ are not exact rank-1 tensors, but can be approximated by rank-1 tensors composed from components corresponding modes in $\boldsymbol{l}_m$. In other words, computing the leading-left singular vector of the mode-k unfolding $[\boldsymbol{\mathcal{B}}_{m_r}]_{(k)}$ is the simplest approach to recover $\boldsymbol{a}_r^{(l_{mk})}$ from $\boldsymbol{b}_r^{(m)}$ for $k = 1, 2, \ldots, K$. Pseudo-code of this simple algorithm for unFolding CPD (FCP) is described in Algorithm 1. TD denotes the multilinear rank-$(R_1, R_2, \ldots, R_N)$ Tucker decomposition (compression) with $R_n \le I_n$, and the min is taken elementwise. We don't need to estimate factors $\mathbf{A}^{(n)}$ if $I_n = R_n$. The more complex and efficient algorithm is discussed later in Section IV.

A. CRIB Analysis

For (noiseless) tensors which have exact rank-R CP decompositions without (nearly) collinear components, factors computed from folded tensors can be close to the true solutions.

However, for real data tensor, there exists loss of accuracy when using the rank-one approximation approach. The loss can be induced by the unfolding, or by the choice of R, especially when R is smaller than the true rank of the data tensor.

This section analyzes such loss based on comparing CRIBs on the first component $\boldsymbol{a}_1^{(1)}$ of CPDs of the full tensor and its unfolded version. We use $\boldsymbol{a}_1$ as a shorthand notation for $\boldsymbol{a}_1^{(1)}$. The results of this section give us an insight into how to unfold a tensor without or possibly minimal loss of accuracy.

Let α_1 denote the mutual angle between the true factor $\boldsymbol{a}_1$ and its estimate $\hat{\boldsymbol{a}}_1$

$$\alpha_1 = \operatorname{acos} \frac{\boldsymbol{a}_1^T \hat{\boldsymbol{a}}_1}{\|\boldsymbol{a}_1\| \, \|\hat{\boldsymbol{a}}_1\|}. \tag{9}$$

When the noise in the model has a zero mean Gaussian distribution with variance σ^2, and is independently added to each element of the tensor, the Cramér-Rao induced bound of a least squares estimator on the squared angular error α_1^2 [radians^2], denoted by $\mathrm{CRIB}(\boldsymbol{a}_1)$ [53], [54], [41], can be computed from the Cramér-Rao lower bound for CPD [55]

$$\mathrm{CRIB}(\boldsymbol{a}_1) = \frac{\mathrm{tr}[\Pi_{\boldsymbol{a}_1}^{\perp} \mathrm{CRLB}(\boldsymbol{a}_1)]}{\|\boldsymbol{a}_1\|^2}, \tag{10}$$

where $\Pi_{\boldsymbol{a}_1}^{\perp} = \mathbf{I}_{I_1} - \boldsymbol{a}_1 \boldsymbol{a}_1^T / \|\boldsymbol{a}_1\|^2$. Fast computation of CRLB and CRIB for general order-N tensors has been recently developed in [41]. $\mathrm{CRIB}(\boldsymbol{a}_1)$ is a scalar measure, and serves a gauge of achievable accuracy of estimation/CP decomposition. $\mathrm{CRIB}(\boldsymbol{a}_1)$ in decibels (dB) is defined as $-10\log_{10}[\mathrm{CRIB}(\boldsymbol{a}_1)]$. A CRIB of 50 dB means that the standard angular deviation (square root of mean square angular error) of the factor is cca. 0.18 degrees; a CRIB of 20 dB corresponds to the standard deviation of cca 5.7 degrees. CRIB is also a bound on the achievable Distortion-to-Signal Ratio [41].

In this section, the accuracy loss in decomposition of unfolded tensor is defined as the loss of CRIB on components of the unfolded tensor through the unfolding rule $\boldsymbol{l}$ compared with CRIB on components of the original tensor. For simplicity, we consider tensors in the Kruskal form (1)–(2) of rank 2, and illustrate the loss of accuracy for higher ranks. The analytic CRIB for order-N rank-2 tensor is derived in [41], and provided in Theorem B.1 in Appendix.

The inner products $c_n = \boldsymbol{a}_1^{(n)T} \boldsymbol{a}_2^{(n)}$ are called degree of collinearity. The bound is known to be largely independent of c_1 unless c_1 is close to ± 1 [41]. In the case of $c_1 = 0$, the bound can be rewritten in particularly simple form,

$$\mathrm{CRIB}(\boldsymbol{a}_1) = \frac{\sigma^2}{\lambda_1^2} \left(\frac{I_1 - 2}{1 - h^2} + \frac{1}{1 + (N - \zeta - 2)h^2} \right) \tag{11}$$

where $h = c_2 c_3 \cdots c_N$, and $\zeta = \sum_{k=2}^{N} \frac{1}{c_k^2}$. Note that it can be proved by mathematical induction that $1 + (N - \zeta - 2)h^2 \geq 0$ for all N.

If two modes, say j-th and k-th, are folded together, the colinearity coefficient corresponding to the folded mode is equal to product $c_j c_k$. The same formula, (11), allows to compute the CRIB for the folded tensor.

B. Unfolding Order-4 Tensors

For simplicity, consider order-4 tensors first. Assuming again $c_1 = 0$, it holds

$$\begin{aligned}
&\mathrm{CRIB}(\boldsymbol{a}_1) \\
&\quad = \frac{\sigma^2}{\lambda_1^2} \left(\frac{I_1 - 2}{1 - h^2} + \frac{1}{1 + 2h^2 - c_2^2 c_3^2 - c_2^2 c_4^2 - c_3^2 c_4^2} \right), \\
&\mathrm{CRIB}_{[[1,2,(3,4)]]}(\boldsymbol{a}_1) \\
&\quad = \frac{\sigma^2}{\lambda_1^2} \left(\frac{I_1 - 2}{1 - h^2} + \frac{1}{1 + h^2 - c_2^2 - c_3^2 c_4^2} \right), \qquad (12) \\
&\mathrm{CRIB}_{[[1,(2,3),4)]]}(\boldsymbol{a}_1) \\
&\quad = \frac{\sigma^2}{\lambda_1^2} \left(\frac{I_1 - 2}{1 - h^2} + \frac{1}{1 + h^2 - c_2^2 c_3^2 - c_4^2} \right). \qquad (13)
\end{aligned}$$

CRIBs in (12) and (13) still hold when $c_1 \neq 0$ (see (33)). In general, $\mathrm{CRIB}(\boldsymbol{a}_1) \leq \mathrm{CRIB}_{[[1,2,(3,4)]]}(\boldsymbol{a}_1)$. The equality is achieved for $c_2 = 0$.

$$\begin{aligned}
\mathrm{CRIB}(\boldsymbol{a}_1)_{c_2=0} &= \mathrm{CRIB}_{[[1,2,(3,4)]]}(\boldsymbol{a}_1)_{c_2=0} \\
&= \frac{\sigma^2}{\lambda_1^2} \left(I_1 - 2 + \frac{1}{1 - c_3^2 c_4^2} \right).
\end{aligned}$$

It means that if modes 1 and 2 comprise (nearly) orthogonal components, the tensor unfolding $[1, 2, (3, 4)]$ does not affect the accuracy of the decomposition.

From (12) and (13), it is obvious that $\mathrm{CRIB}_{[[1,2,(3,4)]]}(\boldsymbol{a}_1) \leq \mathrm{CRIB}_{[[1,(2,3),4)]]}(\boldsymbol{a}_1)$ if $c_2^2 \leq c_4^2$. This indicates that ***collinearities of modes to be unfolded should be higher than those of other modes in order to reduce the loss of accuracy in estimating*** $\boldsymbol{a}_1$. Note that the new factor matrices yielded through tensor unfolding have lower collinearity than those of original ones. Moreover, tensors with high collinear components are always more difficult to decompose than ones with lower collinearity [21], [24], [56], [57]. Hence, it is natural to unfold modes with highest collinearity so that the CPD becomes easier.

A similar CRIB expression can be derived for higher rank R as well using the general CRIB expression in [41], assuming that all pairs of factors have the same colinearity in each mode. We skip the details here to save space, but confirm the above rule in a numerical example.

Example 1: We illustrate the similar behavior of CRIB over unfolding but for higher-order ranks. We decomposed $\boldsymbol{\mathcal{Y}}_{[[1,2,(3,4)]]}$ unfolded from rank-R tensors $\boldsymbol{\mathcal{Y}}$ of size $R \times R \times R \times R$ with $R = 3, 5, \ldots, 30$, composed from factor matrices which have identical $c_n = \boldsymbol{a}_r^{(n)T} \boldsymbol{a}_s^{(n)}$, for $n = 1, 2, \ldots, N$, $r \neq s$, and $\boldsymbol{a}_r^{(n)T} \boldsymbol{a}_r^{(n)} = 1$, $-1 \leq c_n \leq 1$. The tensors were corrupted with additive Gaussian noise of 10 dB signal-to-noise ratio SNR (dB) $= -10 \log_{10} \frac{\|\boldsymbol{\mathcal{Y}}\|_F^2}{\sigma^2 \prod I_n}$, where σ^2 denotes the noise variance, and $\|\boldsymbol{\mathcal{Y}}\|_F$ is the Frobenius norm of $\boldsymbol{\mathcal{Y}}$. The squared angular errors (SAE) between the original and estimated components, i.e., α_1^2, are computed and compared with their CRIBs. As seen in Figs. 1(a)–(c), there was not any significant loss in factors when modes 1 and 2 comprised low-collinear components despite of collinearity in modes 3 and 4. For all the other cases of (c_1, c_2, c_3, c_4), there were always significant losses, especially when all the factors

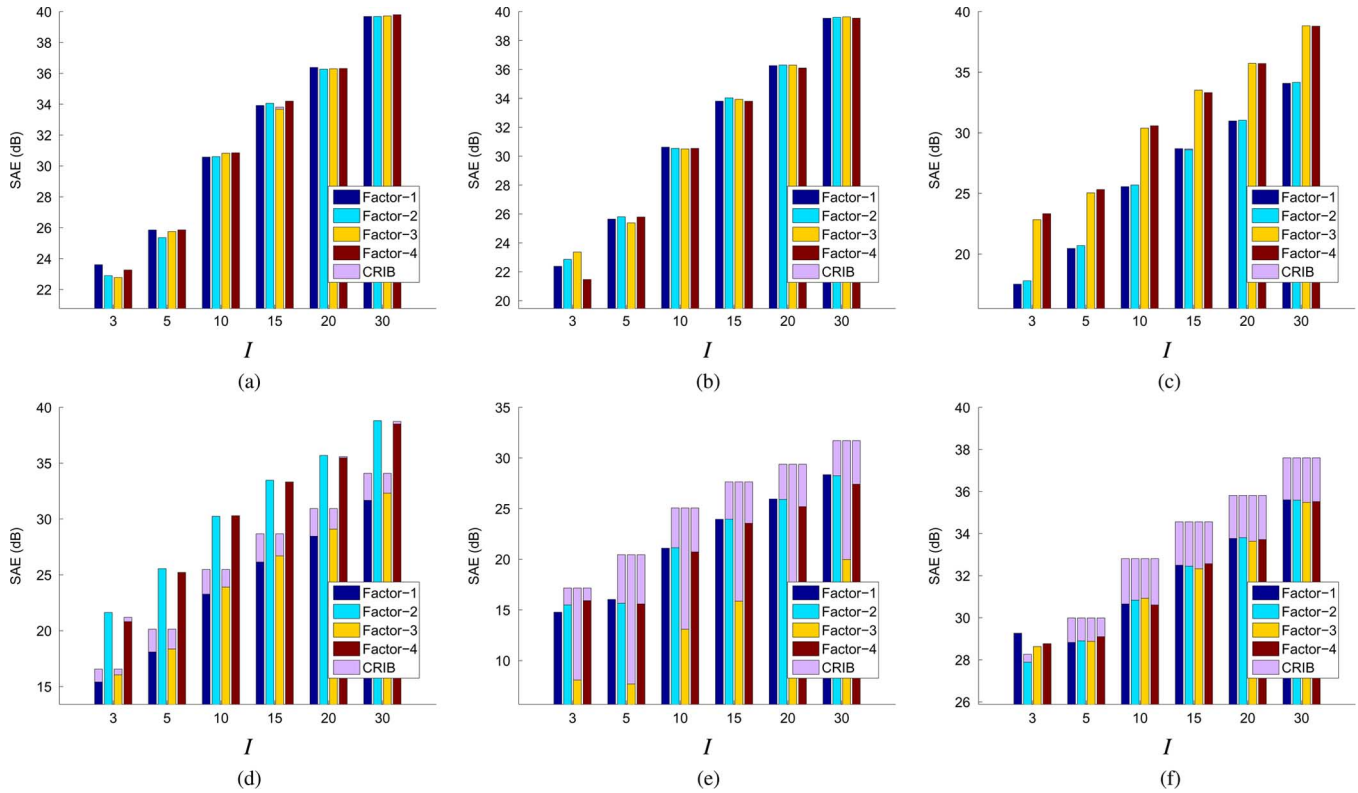

Fig. 1. Median squared angular error (SAE) of all components for factors over 30 Monte Carlo runs versus CRIB in decomposition of order-4 tensors of size $I_n = R = 3, 5, 10, \ldots, 30$, for all n through the unfolding rule $\boldsymbol{l} = [1, 2, (3, 4)]$. Collinearity coefficients c_n have been chosen from the set $\in \{0.1, 0.9\}$, for all n. The signal to white Gaussian noise power ratio (SNR) was at 10 dB or 30 dB. (a) $(c_1, c_2, c_3, c_4) = (0.1, 0.1, 0.1, 0.1)$. (b) $(c_1, c_2, c_3, c_4) = (0.1, 0.1, 0.9, 0.1)$. (c) $(c_1, c_2, c_3, c_4) = (0.1, 0.1, 0.9, 0.9)$. (d) $(c_1, c_2, c_3, c_4) = (0.1, 0.9, 0.1, 0.9)$. (e) $(c_1, c_2, c_3, c_4) = (0.1, 0.9, 0.9, 0.9)$. (f) $(c_1, c_2, c_3, c_4) = (0.9, 0.9, 0.9, 0.9)$, SNR = 30 dB.

comprised highly collinear components (i.e., c_n is close to ± 1) as seen in Figs. 1(d)–(f).

C. A Case When Two Factor Matrices Have Orthogonal Columns

In the previous subsection, there is an interesting case when there is no accuracy loss when $c_1 = c_2 = 0$. This is a special case of a more general result of [41] that the same property holds for arbitrary order-N rank-R tensors which have orthogonal components on two modes. In such case, the analytical CRIB is given by

Theorem 3.1 ([41]): When $\mathbf{A}^{(1)}$ and $\mathbf{A}^{(2)}$ have mutually orthogonal columns, it holds

$$\mathrm{CRIB}(\boldsymbol{a}_1) = \frac{\sigma^2}{\lambda_1^2}\left(I_1 - R + \sum_{r=2}^{R} \frac{1}{1-\gamma_r^2}\right), \qquad (14)$$

where $\gamma_r = \prod_{n=3}^{N} \left(\boldsymbol{a}_1^{(n)T} \boldsymbol{a}_r^{(n)}\right)$, $r = 2, 3, \ldots, R$.
It is obvious that $\mathrm{CRIB}_{[[1,2,(3:N)]]}(\boldsymbol{a}_1) = \mathrm{CRIB}(\boldsymbol{a}_1)$. Hence, estimation of $\mathbf{A}^{(1)}$ and $\mathbf{A}^{(2)}$ through unfolding is lossless in terms of accuracy. Note, however, that if the orthogonality is added in the optimization problem as a constraint, the CRIB in Theorem 3.1 is no longer valid bound on the estimation accuracy. In particular, more accurate estimation of the decomposition is possible, which can be attained by algorithms dedicated to this problem [58], [59], [40].

D. Unfolding Tensors of the Order 5 and Higher

Assume that the tensor to be decomposed has rank 2 and is transposed in the way that its collinearity degrees are sorted in nondecreasing way, and assume that $|c_1| \le |c_2| \le |c_3| \le \cdots \le |c_N|$. Assume for simplicity that $c_1 = 0$ and $|c_2| > 0$.

It follows from (11) that among all simple unfoldings that combining two modes in one, $\boldsymbol{l}_{N-1} = [1, \ldots, N-2, (N-1, N)]$ has the lowest CRIB in estimating $\boldsymbol{a}_1$.

Optimum unfoldings of the tensor to order $(N-2)$ and lower order can be found recursively. In the second step, the tensor has order-$(N-1)$, and its colinearity degrees are $c_1, \ldots, c_{N-2}$ and $\tilde{c}_{N-1} = c_N c_{N-1}$. The colinearity degrees are sorted again, and the optimum second unfolding collects again the modes with the highest colinearities. It is either $(1, 2, \ldots, N-4, (N-3, N-2), (N-1, N))$ or $(1, 2, \ldots, N-3, (N-2, N-1, N))$, in dependence if $|c_{N-1} c_N| < |c_{N-2}|$ or vice versa. Deeper unfolding rules can be generated recursively until the unfolded tensor has order 3. (Indeed, the unfolding needs not to go down to order-3, but can stop earlier.)

Example 2 Unfolding Tensors With the Same Collinearity in All Modes: As an example, we can consider unfolding of an order-6 tensor where the collinearity degrees are $c_1 = 0$, $c_2 = \cdots = c_6 = c > 0$. The above described procedure suggests order-5 unfolding rule $(1, 2, 3, 4, (5, 6))$, order-4 unfolding rule $(1, 2, (3, 4), (5, 6))$ and order-3 unfolding rule $(1, (3, 4), (2, 5, 6))$. The other possible folding rules, order-4 rule $(1, 2, 3, (4, 5, 6))$ and order-3 rule $(1, 2, (3, 4, 5, 6))$

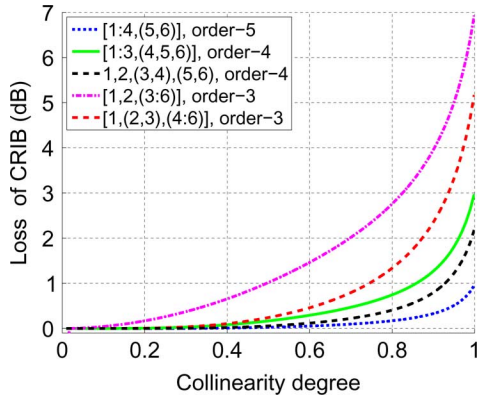

Fig. 2. The CRIB loss in decomposition of order-6 rank-20 tensors with size $I_n = 20$ and identical collinearity coefficients following five unfolding rules in Example 2. The CRIB loss is significant when components are collinear, i.e., $c \to 1$. Unfolding $\boldsymbol{l} = [1:4,(5,6)]$ causes a lesser CRIB loss than other rules. The unfolding $\boldsymbol{l} = [1,2,(3,4),(5,6)]$ is more efficient than multimode unfolding.

are less good, leading to a higher loss in accuracy. Fig. 2 shows the CRIB loss $-10\log_{10}\left(\frac{\text{CRIB}(\boldsymbol{a}_1)}{\text{CRIB}_{\boldsymbol{l}}(\boldsymbol{a}_1)}\right)$ (dB). The loss by $\boldsymbol{l} = [1,2,3,(4,5,6)]$ was higher than that by $\boldsymbol{l} = [1,2,(3,4),(5,6)]$.

E. Unfolding Without Collinearity Information

It might happen that no prior information about collinearity in the tensor modes is available. Then, a bad choice of the folding strategy may result in poor accuracy of the decomposition, as will be shown in Example 3 and examples in the simulation section. One remedy to this problem is proposed in the next section in the form of another variant of the FCP algorithm (Algorithm 2). The algorithm is shown to be much less vulnerable to bad choice of the folding strategy.

Another option is a combined strategy, trying sequentially several unfolding rules and accept the decomposition which gives the best fit between the Kruskal form approximation and the original tensor. One tentative folding rule can give a guidance for a better folding rule in the next step.

It is worth noting that CRIB also depends on the length of factor matrix. For example, when combining many factors into one, length of the new factor matrix significantly increase, while its collinearity coefficients tend to be much smaller. From CRIB for rank-2 order-3 tensor given in (33), one can see that CRIB of unfolded components is significant. It indicates that an unfolding which combines many modes can reduce loss in estimation of a component, but can cause a significant loss in estimating components in folding modes.

In the first run, one should not combine many modes into one, but can try an unfolding which balances lengths of unfolding factors. Collinearity coefficients of the resulted factor matrices are used to verify whether all factor with low collinearity are unfolded. An important observation is that the loss of accuracy is significant when combining two or several modes with lowest collinearity degrees in the unfolding $\boldsymbol{l} = [p_1, p_2, (p_3, \ldots, p_N)]$, where $\boldsymbol{p}$ is a permutation of $[1:N]$, and $p_1 < p_2$, $p_3 < \cdots < p_N$. On the basis of the order-3 unfoldings, we can determine modes with lowest collinearity degrees using not more than $\binom{N}{2}$ such unfoldings, and a more suitable unfolding will be chosen further.

Algorithm 2: FCP

Input: Data tensor $\mathcal{Y}$: $(I_1 \times I_2 \times \cdots \times I_N)$, rank R, threshold $\tau(\geq 0.98)$
Unfolding rule $\boldsymbol{l} = [l_1, l_2, \ldots, l_M]$, $l_m = [l_m(1), l_m(2), \ldots, l_m(K_m)]$
Output: Rank-R Kruskal tensor $\mathcal{Y}_R = [\![\boldsymbol{\lambda}; \mathbf{A}^{(1)}, \ldots, \mathbf{A}^{(N)}]\!]$
begin
% **Stage 1:** Tensor unfolding and compression----------
1 $[\![\mathcal{G}, \mathbf{U}^{(1)}, \ldots, \mathbf{U}^{(M)}]\!] = \text{TD}(\mathcal{Y}_{[\boldsymbol{l}]}, \min(\boldsymbol{I}, R))$
% **Stage 2:** CPD of $\mathcal{G}$ and back projection ----------
2 $[\![\boldsymbol{\lambda}; \mathbf{B}^{(1)}, \ldots, \mathbf{B}^{(M)}]\!] = \text{CPD}(\mathcal{G}, R)$
3 $\mathcal{Y}_R = [\![\boldsymbol{\lambda}; \mathbf{U}^{(1)}\mathbf{B}^{(1)}, \ldots, \mathbf{U}^{(M)}\mathbf{B}^{(M)}]\!]$ % rank-R K-tensor
% **Stage 3:** Sequential estimation ----------
foreach *group* l_m *with* $K_m \geq 2$ **do**
4 $n = K_1 + \cdots + K_{m-1}$
for $k = 1, 2, \ldots, K_m$ **do**
% **Stage 3a:** Construction of rank-J K-tensor---
$[\mathcal{Y}_J, \mathcal{Y}_R] = \text{twomodeFCP}(\mathcal{Y}_R, n+k, I_{l_m(k)})$
% **Stage 3b:** Rank-J to rank-R K-tensor---------
5 **if** $J > R$ **then** $\mathcal{Y}_R = \text{structuredCPD}(\mathcal{Y}_J, R, \mathcal{Y}_R)$
% **Stage 4:** Refinement if needed ----------
6 $\mathcal{Y}_R = \text{CPD}(\mathcal{Y}, R, \mathcal{Y}_R)$

function $[\mathcal{Y}_J, \mathcal{Y}_R] = \text{twomodeFCP}(\mathcal{Y}_R, n, T)$
Input: Order-N rank-R Kruskal tensor $\mathcal{Y}_R = [\![\boldsymbol{\lambda}; \mathbf{A}^{(1)}, \ldots, \mathbf{A}^{(N)}]\!]$, with $\mathbf{A}^{(n)} \in \mathbb{R}^{I_n \times R}$ to be reconstructed
Output: Two order-$(N+1)$ rank-J and rank-R K-tensors $\mathcal{Y}_J, \mathcal{Y}_R$
for $r = 1, 2, \ldots, R$ **do**
7 $\mathbf{F}_r = \text{reshape}(\boldsymbol{a}_r^{(n)}, [T, I_n/T])$
8 $\mathbf{F}_r \approx \mathbf{U}_r \,\text{diag}(\boldsymbol{\sigma}_r)\,\mathbf{V}_r^T$ % rank-J_r truncated SVD, $\|\boldsymbol{\sigma}_r\|_2^2 \geq \tau$
% Construct a rank-J K-tensor, $J = J_1 + J_2 + \cdots + J_R$ -------
9 $\mathbf{M} = \text{blkdiag}(\mathbf{1}_{1\times J_1}, \ldots, \mathbf{1}_{1\times J_R})$
10 $\widetilde{\boldsymbol{\lambda}} = [\lambda_1\boldsymbol{\sigma}_1, \lambda_2\boldsymbol{\sigma}_2, \ldots, \lambda_R\boldsymbol{\sigma}_R]$, $\mathbf{G} = [\mathbf{U}_1, \ldots, \mathbf{U}_R]$, $\mathbf{H} = [\mathbf{V}_1, \ldots, \mathbf{V}_R]$
11 $\mathcal{Y}_J = [\![\widetilde{\boldsymbol{\lambda}}; \mathbf{A}^{(1)}\mathbf{M}, \ldots, \mathbf{A}^{(n-1)}\mathbf{M}, \mathbf{G}, \mathbf{H}, \mathbf{A}^{(n+1)}\mathbf{M}, \ldots, \mathbf{A}^{(N)}\mathbf{M}]\!]$
% Construct a rank-R Kruskal tensor ----------
12 $\lambda \leftarrow [\lambda_1\sigma_{11}, \lambda_2\sigma_{21}, \ldots, \lambda_R\sigma_{R1}]$, $\mathbf{G} = [\boldsymbol{u}_{11}, \ldots, \boldsymbol{u}_{R1}]$, $\mathbf{H} = [\boldsymbol{v}_{11}, \ldots, \boldsymbol{v}_{R1}]$
13 $\mathcal{Y}_R = [\![\boldsymbol{\lambda}; \mathbf{A}^{(1)}, \ldots, \mathbf{A}^{(n-1)}, \mathbf{G}, \mathbf{H}, \mathbf{A}^{(n+1)}, \ldots, \mathbf{A}^{(N)}]\!]$

K-tensor means tensor in Kruskal form.

There are many options, and this paper does not have the ambition to investigate them all. We shall mainly study performance of the proposed Algorithm 1 and Algorithm 2 and show that namely performance of the latter Algorithm is very good in general.

Example 3: We decomposed order-5 tensors with size $I_n = R = 15$ and additive Gaussian noise of 0 dB SNR. Factors matrices $\mathbf{A}^{(n)}$ were randomly generated such that their collinearity coefficients were in given ranges [0, 0.45], [0.2, 0.65], [0.5, 0.99], [0.95, 0.99] and [0.9, 0.99] [60] as shown in Fig. 3(a), respectively. Columns of the fourth and fifth factor matrices are highly collinear, whereas those of the first factor matrix are less collinear. The MSAEs (dB) were computed over 100 runs for all possible unfolding rules (see Fig. 3(c)). Performance of rank-one FCP (R1FCP) was highly affected by the choice of unfolding rules. For example, R1FCP completely failed when using unfoldings $[3,4,(1,2,5)]$, $[3,5,(1,2,4)]$ and $[(1,2,3),4,5]$ with very low fits $(=1 - \frac{\|\mathcal{Y}-\widehat{\mathcal{Y}}\|_F}{\|\mathcal{Y}\|_F})$, because the first and second factor matrices comprised low collinear components. Fits of R1FCP using "good" unfoldings varied in a narrow range. Some "good" unfoldings for R1FCP were $[1,(2,4),(3,5)]$, $[1,(2,5),(3,4)]$, $[2,(1,4),(3,5)]$, $[2,(1,5),(3,4)]$, $[3,(1,4),(2,5)]$ and $[3,(1,5),(2,4)]$. The unfolding $[1,2,[3,4,5]$ is good to estimate $\mathbf{A}^{(1)}$, but it is not the best unfolding rule in this example. Fig. 3(b) shows

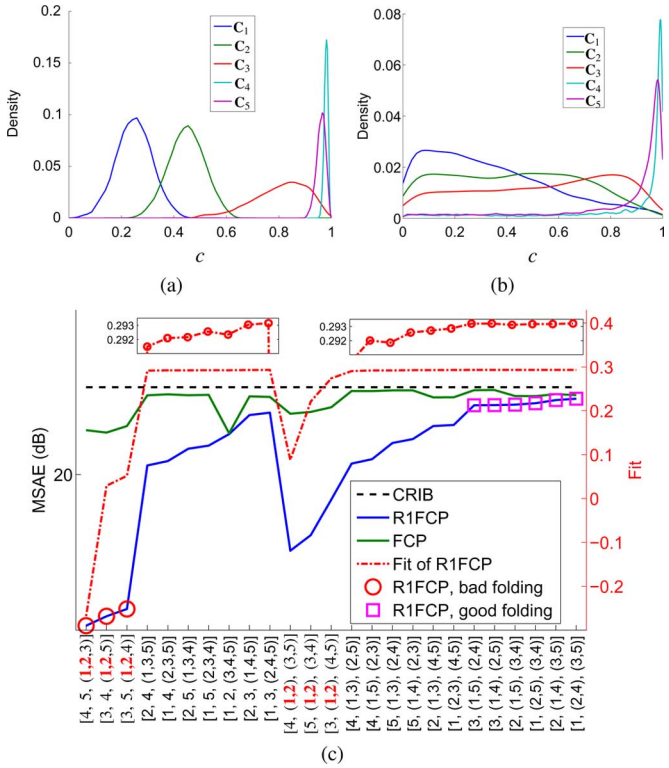

Fig. 3. Illustration of loss of accuracy of R1FCP in Example 3. Collinearity coefficients of factor matrices are distributed as shown in (a). (b) Collinearity coefficients of the estimated factors when using the bad unfolding $[(1,2,3),4,5]$. (c) MSAEs were averaged over 100 runs for all possible unfolding rules.

distributions of collinearity coefficients of estimated factors for the case using a "bad" unfolding $[(1,2,3),4,5]$. It indicates that factors 1 and 2 comprise the lowest collinear components. Note that the unfolding $[3,(1,4),(2,5)]$, not the unfolding $[3,(\mathbf{1},\mathbf{2}),(4,5)]$, is good to estimate $\mathbf{A}^{(3)}$. The green solid line in Fig. 3 shows results of the FCP algorithm, which is discussed later in Section IV as Algorithm 2.

In addition, one can estimate the average collinearity degrees, and uses them to design further unfoldings. For more examples, see decomposition of the ITPC tensor in Example 6 when $R = 8$.

IV. Fast Approximation for High Order and Difficult Scenario CPD

In difficult scenarios, at presence of highly colinear factors in several modes and/or high level additive noise, it happens that the folded tensor can be approximated by Kruskal tensor of a lower rank than the rank of the original tensor. Consequently, the factors of the folded dimensions cannot be well approximated by rank one approximations as in Algorithm 1.

Below we propose an extension of Algorithm 1, which utilizes intermediate approximations of the original tensor of a rank higher than the desired one to get a better approximation of the original tensor. The algorithm is first derived for unfolding two modes, and extended to multimode unfolding.

A. Unfolding Two Modes

We consider a rank-R CPD of $\mathcal{Y}$

$$\mathcal{Y} = \sum_{r=1}^{R} \lambda_r \, \boldsymbol{a}_r^{(1)} \circ \boldsymbol{a}_r^{(2)} \circ \cdots \circ \boldsymbol{a}_r^{(N)} + \mathcal{E}_1, \tag{15}$$

$$= \left[\left[\boldsymbol{\lambda}; \mathbf{A}^{(1)}, \mathbf{A}^{(2)}, \ldots, \mathbf{A}^{(N)}\right]\right] + \mathcal{E}_1 \tag{16}$$

and a simple unfolding $\boldsymbol{l} = [1, \ldots, N-2, (N-1, N)]$

$$\mathcal{Y}_{[[\boldsymbol{l}]]} = \sum_{r=1}^{R} \mu_r \, \boldsymbol{b}_r^{(1)} \circ \boldsymbol{b}_r^{(2)} \circ \cdots \circ \boldsymbol{b}_r^{(N-1)} + \mathcal{E}_2 \tag{17}$$

$$= \left[\left[\boldsymbol{\mu}; \mathbf{B}^{(1)}, \mathbf{B}^{(2)}, \ldots, \mathbf{B}^{(N-1)}\right]\right] + \mathcal{E}_2 \tag{18}$$

where $\boldsymbol{\mu} = [\mu_1, \ldots, \mu_R]$. If the noise variance is low, i.e., $\|\mathcal{E}_1\|_F \ll \|\boldsymbol{\lambda}\|$ and $\|\mathcal{E}_2\|_F \ll \|\mu\|$, and the factors of both tensors are ordered in accord, it holds

$$\boldsymbol{\lambda} \approx \boldsymbol{\mu}, \quad \mathbf{A}^{(k)} \approx \mathbf{B}^{(k)} \quad \text{for } k = 1, \ldots N-2,$$
$$\boldsymbol{b}_k^{(N-1)} \approx \boldsymbol{a}_k^{(N)} \otimes \boldsymbol{a}_k^{(N-1)}, \quad k = 1, \ldots, R,$$

where $\mathbf{B}^{(N-1)} = \left[\boldsymbol{b}_1^{(N-1)}, \boldsymbol{b}_2^{(N-1)}, \ldots, \boldsymbol{b}_R^{(N-1)}\right]$. Put $\mathbf{F}_r = \texttt{reshape}(\boldsymbol{b}_r^{(N-1)}, [I_{N-1}, I_N])$, and let $\mathbf{F}_r = \mathbf{U}_r \boldsymbol{\Sigma}_r \mathbf{V}_r^T$ be singular value decomposition of $\mathbf{F}_r$ with the left, right singular vectors, and the singular values, respectively, given by

$$\mathbf{U}_r = [\boldsymbol{u}_{r1}, \ldots, \boldsymbol{u}_{rT}], \quad \mathbf{V}_r = [\boldsymbol{v}_{r1}, \ldots, \boldsymbol{v}_{rT}]$$
$$\boldsymbol{\sigma}_r = [\sigma_{r1}, \sigma_{r2}, \ldots, \sigma_{rT}], \quad 1 \geq \sigma_{r1} \geq \sigma_{r2} \geq \cdots \geq \sigma_{rT} \geq 0.$$

Note that the sum of the squared singular values is 1, because $\|\boldsymbol{b}_k^{(N-1)}\| = \|\mathbf{F}_r\|_F = 1$ by definition.

Theoretically, $\mathbf{F}_r$ has rank one and only the first singular value σ_{r1} is significant, i.e., $\boldsymbol{u}_{r1} \approx \boldsymbol{a}_r^{(N-1)}$ and $\boldsymbol{v}_{r1} \approx \boldsymbol{a}_r^{(N)}$. The rank-$R$ rank-one-based approximation of the tensor $\mathcal{Y}$ is defined as

$$\mathcal{Y}_R = \left[\left[\boldsymbol{\mu}; \mathbf{B}^{(1)}, \mathbf{B}^{(2)}, \ldots, \tilde{\mathbf{B}}^{(N-1)}, \tilde{\mathbf{B}}^{(N)}\right]\right] \tag{19}$$

where $\tilde{\mathbf{B}}^{(N-1)} = [\boldsymbol{u}_{11}, \ldots, \boldsymbol{u}_{R1}]$ and $\tilde{\mathbf{B}}^{(N)} = [\boldsymbol{v}_{11}, \ldots, \boldsymbol{v}_{R1}]$. In general, a numerical rank of matrices $\mathbf{F}_r$ is $J_r \geq 1$, $r = 1, \ldots, R$, being defined as the minimum constant such that

$$\sum_{j=1}^{J_r} \sigma_{rj}^2 \geq \tau$$

where $0 < \tau < 1$ is a given constant (we use $\tau = 0.98$) for truncating the SVD. We can write then

$$\boldsymbol{b}_r^{(N-1)} \approx \sum_{j=1}^{J_r} \sigma_{rj} \, (\boldsymbol{v}_{rj} \otimes \boldsymbol{u}_{rj}). \tag{20}$$

In this way we get a rank-J approximation of the original tensor, where $J = \sum_{r=1}^{R} J_r$

$$\mathcal{Y}_J = \sum_{r=1}^{R} \mu_r \, \boldsymbol{b}_r^{(1)} \circ \boldsymbol{b}_r^{(2)} \circ \cdots \circ \boldsymbol{b}_r^{(N-2)} \circ \left(\sum_{j=1}^{J_r} \sigma_{rj} \, (\boldsymbol{u}_{rj} \circ \boldsymbol{v}_{rj}) \right). \tag{21}$$

Symbolically we can write $\mathcal{Y}_J$ as

$$\mathcal{Y}_J = \left[\left[\tilde{\boldsymbol{\lambda}}; \tilde{\mathbf{A}}^{(1)}, \tilde{\mathbf{A}}^{(2)}, \ldots, \tilde{\mathbf{A}}^{(N)}\right]\right] \tag{22}$$

where $\tilde{\mathbf{A}}^{(k)}$ have dimension $I_k \times J$. Columns of $\tilde{\mathbf{A}}^{(k)}$ for $k = 1, \ldots, N-2$ are replicated copies of columns of $\mathbf{B}^{(k)}$, $\tilde{\mathbf{A}}^{(N-1)}$ is formed of the left singular vectors $\boldsymbol{u}_{rj}$, and $\tilde{\mathbf{A}}^{(N)}$ is formed of the right singular vectors $\boldsymbol{v}_{rj}$. In particular,

$$\tilde{\mathbf{A}}^{(k)} = \mathbf{B}^{(k)}\mathbf{M} \quad \text{for} \quad k = 1, \ldots, N-2 \tag{23}$$

$$\tilde{\mathbf{A}}^{(N-1)} = [\mathbf{U}_1, \ldots, \mathbf{U}_R] \tag{24}$$

$$\tilde{\mathbf{A}}^{(N)} = [\mathbf{V}_1, \ldots, \mathbf{V}_R] \tag{25}$$

$$\mathbf{M} = \mathrm{bdiag}\left(\mathbf{1}_{1\times J_1}, \ldots, \mathbf{1}_{1\times J_R}\right) \tag{26}$$

$$\tilde{\boldsymbol{\lambda}} = \left[\mu_1\boldsymbol{\sigma}_{11}, \ldots, \mu_1\boldsymbol{\sigma}_{1J_1}, \ldots, \mu_R\boldsymbol{\sigma}_{R1}, \ldots, \mu_R\boldsymbol{\sigma}_{RJ_R}\right]. \tag{27}$$

Now, the desired rank-R approximation of $\mathcal{Y}$ can be obtained by applying a structured CPD algorithm to $\mathcal{Y}_J$, which is presented in Appendix A. The algorithm can be initialized by the Kruskal tensor $\mathcal{Y}_R$ in (19), and does not access and manipulate on the real data $\mathcal{Y}$. The mode-n CP-gradients which are the largest workload in CP algorithms such as ALS, OPT, dGN can be quickly computed as shown in Appendix A. In other words, estimation of factors $\mathbf{A}^{(n)}$ from the Kruskal tensor $\mathcal{Y}_J$ is relatively fast. Moreover, only few iterations are usually needed, because a good initial decomposition (19) is already available.

B. Multimode Unfolding/The Proposed Algorithm

The procedure described in the previous section can be easily extended to the more general case. For example, consider the unfolding rule $\boldsymbol{l} = [1, 2, \ldots, N-3, (N-2, N-1, N)]$. The unfolding can proceed in double execution of the above procedure. Starting with decomposition of $\mathcal{Y}_{[[\boldsymbol{l}]]}$, continue with finding a rank-J decomposition of the tensor $\mathcal{Y}_{[[1,2,\ldots,N-2,(N-1,N)]]}$, reducing the rank to R using a structured CPD. The algorithm proceeds by another rank-J decomposition of the original tensor and its rank reduction using another structured CPD algorithm.

Similarly, it is possible to do unfolding when there are more than one group of folded dimensions, e.g., $\boldsymbol{l} = [(1, 2), (3, 4, 5), (7, 8)]$. The algorithm in its full generality is summarized in Algorithm 2. The algorithm reduces the tensor order from N to M (e.g., 3) specified by the unfolding rule $\boldsymbol{l} = [\boldsymbol{l}_1, \boldsymbol{l}_2, \ldots, \boldsymbol{l}_M]$, where each group of modes $\boldsymbol{l}_m = [l_m(1), l_m(2), \ldots, l_m(K_m)]$, $K_m \geq 1$ and $\sum_{m=1}^{M} K_m = N$.

In stage 1, Tucker compression can be applied to the unfolded tensor $\mathcal{Y}_{[[\boldsymbol{l}]]}$ using the HOOI algorithm [45] with a few iterations. A rank-R order-M Kruskal tensor $\mathcal{Y}_R$ is obtained after stage 2. The reconstruction process is then sequentially proceeded through two loops over all groups $\boldsymbol{l}_m$ which have $K_m \geq 2$, and their modes $l_m(k), k = 1, \ldots, K_m$. In each run of stage 3a, the method for two-mode unfolding in Section IV-A is executed, and returns a rank-J Kruskal tensor $\mathcal{Y}_J$ and a rank-R Kruskal tensor $\mathcal{Y}_R$. $\mathcal{Y}_J$ is then approximated by a rank-R Kruskal tensor using $\mathcal{Y}_R$ to initialize.

The algorithm seems complex, but in practice it is very efficient, as is shown in the simulation section. Moreover, the simulations show that it is largely tolerant to a wrong selection of the unfolding rule at the beginning. Note, however, that the algorithm works better (namely faster), when the folding rule is appropriate, because the intermediate ranks J's are smaller in that case. An alternative approach to multimode-unfolding is based on multilinear low-rank tensor approximation [45].

In some situations, namely in dealing with difficult data, we found it useful to modify the Algorithm 2 in the sense that the rank reduction in step 5 from J to R is replaced with the reduction of the rank from J to slightly higher rank, say $R + 2$. Only in the terminating rank reducing step the rank is reduced directly to R. The algorithm can be completed by a few iterations of a CPD algorithm (e.g., ALS) of the original (unfolded) tensor. The full implementation of FCP is provided at http://www.bsp.brain.riken.jp/~phan/tensorbox.php.

V. Generalized Rank Annihilation Method for Higher Order CPD

It is known that there is closed-form solution for exact CPD [27]. For CPD of order-3 tensor $\mathcal{Y}$ of size $R \times R \times 2$, i.e., $\mathcal{Y}$ has only two frontal slices $\mathbf{Y}_1$ and $\mathbf{Y}_2$, solution can be found from a generalized eigenvalue problem of its two frontal slices, which is known as the generalized rank annihilation method (GRAM) [61]. For example, columns of $\mathbf{B} = \left(\mathbf{A}^{(2)T}\right)^{-1}$ are computed as generalized eigenvectors of the matrix pencil $(\mathbf{Y}_1, \mathbf{Y}_2)$.

Based on this result, Sanchez and Kowalski [62] developed the Direct Tri-Linear Decomposition (DTLD) for fitting the order-3 CPD, which first uses rank-$(R, R, 2)$ Tucker decomposition, then factorizes the core tensor using the GRAM algorithm. This algorithm is used as a useful initialization for third-order tensor factorizations [3], [17], [42].

This section presents the use of GRAM or DTLD to higher order CPD as an application of the FCP algorithm. The higher order tensor is first unfolded to be an order-3 tensor of size $I \times J \times K$, i.e., $M = 3$, so that the two largest dimensions, say I and J, are greater than or equal to rank R. DTLD [62] is simply applied to the order-3 unfolded tensor as a CP algorithm in stage 2 of Algorithm 2, which itself consists of two substages: compress the unfolded tensor using Tucker decomposition to yield a core tensor of size $R \times R \times 2$, find factor matrices from two slices of the core tensor using GRAM [61]. Finally, N factor matrices are estimated as in stage 3 of Algorithm 2. In the experimental section, we will show that this extension of the GRAM algorithm is a very efficient initialization technique for the ALS algorithm.

VI. Simulations

Throughout the simulations, the ALS algorithm factorized data tensors in 1000 iterations and stopped when $\varepsilon = \frac{\|\mathcal{Y} - \widehat{\mathcal{Y}}\|_F}{\|\mathcal{Y}\|_F} \leq 10^{-8}$. The FCP algorithm itself is understood as Algorithm 2 with low-rank approximation. Otherwise, the FCP algorithm with rank-one approximation is denoted by R1FCP. ALS was also utilized in FCP to decompose unfolded tensors. Execution times were measured using the stopwatch command: "tic" "toc" of Matlab release 2011a on a computer

which had 96 GB of RAM and two six-core Intel Xeon processors X5690@3.47 GHz and the Windows 7 operating system. The ALS algorithm was initialized by multi-initial points with ten iterations, including five random values, and one based on R leading singular vectors of mode-n unfoldings. The component matrices with the lowest approximation error are selected to continue further in the process. The ALS algorithm was also initialized using the FCP-GRAM in Section V.

Example 4 (Decomposition of Order-6 Tensors): We analyzed the mean SAEs (MSAE) of algorithms in decomposition of order-6 tensors with size $I_n = R = 20$ by varying the number of low collinear factors from 1 to 6 with $c_n = \boldsymbol{a}_r^{(n)T}\boldsymbol{a}_s^{(n)} \in \{0.1, 0.95\}$ for $r \neq s$, and $[c_1 \leq c_2 \leq \cdots \leq c_6]$. Collinearity coefficients were assumed to be identical for any components $\mathrm{r} \neq \mathrm{s}$. Tensors were corrupted with additive Gaussian noise of various noise levels $\mathrm{SNR} = -10$ dB, 0 dB, 10 dB, 20 dB and 30 dB. MSAE was averaged over 100 runs, whereas mean execution time was measured over 30 runs.

ALS using random and SVD-based initializations was not efficient. Its MSAEs over all the estimated components were clearly lower than CRIB, when $\mathrm{SNR} = -10$ dB or when there were 5 collinear factors and $\mathrm{SNR} = 0$ dB (the first test case in Fig. 4(a)). Performance of ALS was better, and approached the CRIB for higher SNRs, when there were not more than four $c_n = 0.1$ as seen in Figs. 4(a)–(d). However, when there were five or six factors with low collinearity (Figs. 4(e)–(f)), ALS often got stuck in local minima, and was not comparable to FCP.

As seen in Fig. 4, using the FCP-GRAM-based initialization, performance of ALS was significantly improved for difficult test cases, e.g., $\mathrm{SNR} = -10$ and 0 dB. Moreover, ALS was sped-up by at least 200 seconds (see Fig. 4(g)).

The FCP method was executed with "good unfolding" $\boldsymbol{l} = [1, 2, (3, 4, 5, 6)]$ suggested by the strategy in Section III-D and "bad" ones which violated the unfolding strategy such as $[2, 3, (1, 4, 5, 6)]$, $[3, 4, (1, 2, 5, 6)]$, $[4, 5, (1, 2, 3, 6)]$, and $[(1, 2), (3, 4), (5, 6)]$. Performance of R1FCP (Algorithm 1) was strongly affected by the unfolding rules. For example, R1FCP with "bad unfoldings" completely failed when $\mathrm{SNR} = -10$ dB, $c_1 = 0.1$ and $c_2 = \cdots = c_6 = 0.95$, and lost an approximate MSAE of 9–12 dB for other test cases. For all the test cases, FCP with low-rank approximations obtained high performance even with "bad unfolding" rules. Finally, in this simulation, FCP was on average 77 to 37 times faster than ALS as seen in Figs. 4(g)–(h).

Example 5 (CPD With One Column-Wise Orthogonal Factor Matrix): We decomposed order-5 tensors of size and rank $I_n = R = 15$ composed from one orthonormal factor matrix $\mathbf{A}^{(1)}$, and four factor matrices whose collinearity coefficients c_n were randomly distributed in $[0.9, 0.95]$. One of promising algorithms resolving such problem is the recently proposed CPO-ALS2 [40], which iteratively estimates $\boldsymbol{a}_r^{(n)}$, for $n > 1$, from rank-one tensor approximation of order-4 tensors $\boldsymbol{\mathcal{Y}} \bar{\times}_1 \boldsymbol{a}_r^{(1)}$[1], and computes $\mathbf{A}^{(1)} = \mathbf{U}\mathbf{V}^T$ from singular vectors of $\mathbf{Y}_{(1)}\left(\bigodot_{n\neq 1}\mathbf{A}^{(n)}\right) = \mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^T$ as in the INDORT algorithm [59]. For the FCP algorithm, CPO-ALS2 was employed to decompose order-3 unfolded tensors using unfolding $\boldsymbol{l} = [1, (2, 3), (4, 5)]$ in stage 2 of Algorithm 2, also adapted for structured CPD in stage 3b. We skip the detailed derivation here because it is similar to that of ALS in Appendix A.

[1]"$\bar{\times}_1$" denotes mode-1 tensor-vector product [38].

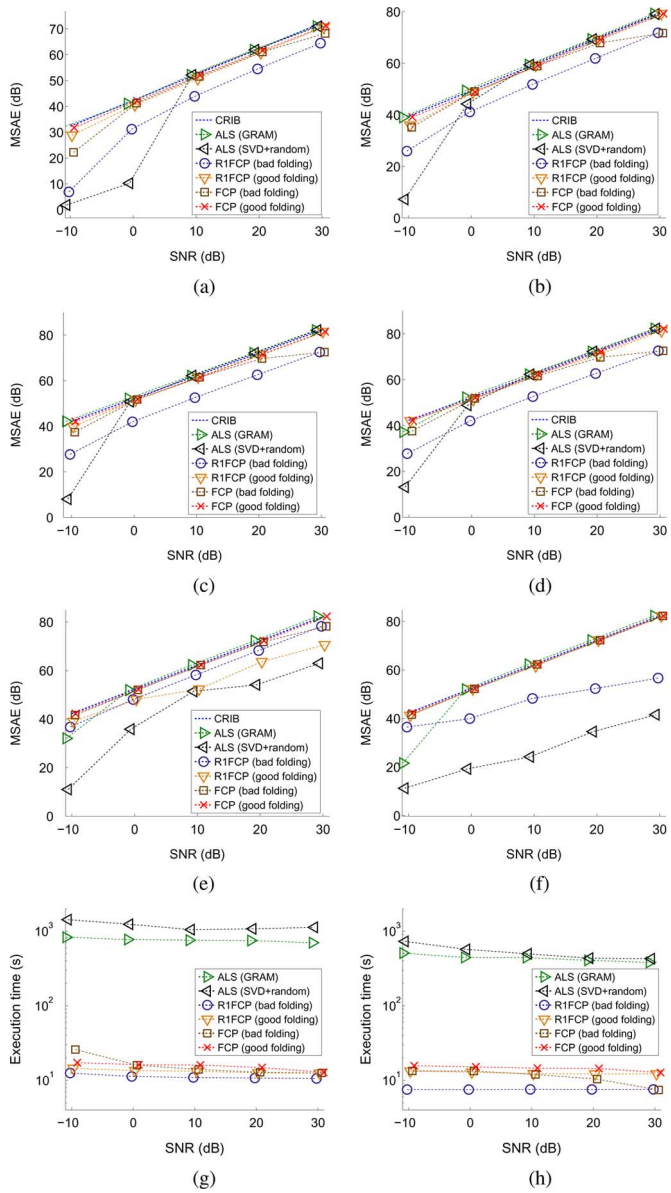

Fig. 4. (a)–(f) Illustration of MSAE loss averaged over 100 MC runs in decomposition of order-6 rank-20 tensors of size $I = R = 20$ corrupted with additive Gaussian noise in Example 4. (g)–(h) Comparison of average execution times of the ALS and FCP algorithms. (a) $c_1 = 0.1, c_2 = \cdots = c_6 = 0.95$. (b) $c_1 = c_2 = 0.1, c_3 = \cdots = c_6 = 0.95$. (c) $c_1 = c_2 = c_3 = 0.1, c_4 = c_5 = c_6 = 0.95$. (d) $c_1 = \cdots = c_4 = 0.1, c_5 = c_6 = 0.95$. (e) $c_1 = \cdots = c_5 = 0.1, c_6 = 0.95$. (f) $c_1 = \cdots = c_6 = 0.1$. (g) $c_1 = 0.1, c_2 = \cdots = c_6 = 0.95$. (h) $c_1 = \cdots = c_6 = 0.1$.

Fig. 5 compares execution time (seconds) and MSAE (dB) of CPO-ALS2 and FCP averaged over 100 runs at different noise levels. When $\mathrm{SNR} \geq 0$ dB, the MSAEs of two algorithms were not significantly different, and slightly higher than CRIBs computed without orthogonality constraints in the cost function. However, when $\mathrm{SNR} = -10$ dB, both algorithms failed to retrieve the hidden components. In this example, FCP factorized the tensors only in a few seconds, while CPO-ALS2 were approximately 7–18 times slower than FCP as SNR varied from

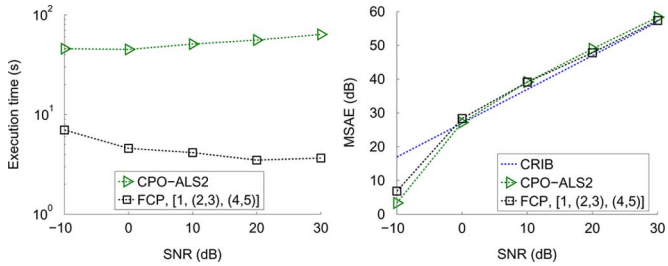

Fig. 5. Comparison of (left) execution time (seconds) and (right) mean SAEs (dB) of CPO-ALS2 and FCP in factorizations of order-5 tensors with one column-wise orthogonal factor matrix.

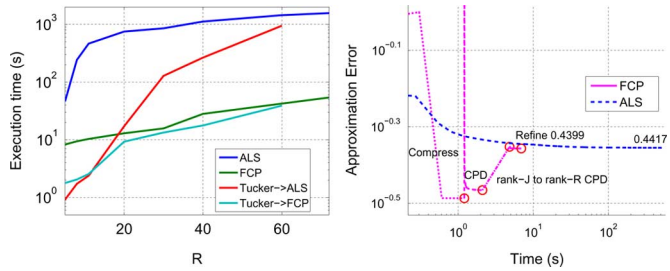

Fig. 6. (left) Comparison of execution times (seconds) of ALS and FCP for factorization of order-4 ITPC tensor with different rank R in Example 6; (right) illustration of approximation error as function of time when $R = 20$.

-10 dB to 30 dB. Note that CPO-ALS2 were more time consuming when SNR =30 dB than when SNR $= -10$ dB, because high collinearity of four factor matrices were deteriorated by noise.

Example 6 (Factorization of Event-Related EEG Time-Frequency Representation): This example illustrates an application of CPD for analysis of real-world EEG data [9], [63] which consisted of 28 inter-trial phase coherence (ITPC) measurements of EEG signals of 14 subjects during a proprioceptive pull of the left and right hands. The whole ITPC data set was organized as a 4-way tensor of 28 measurements $\times$ 61 frequency bins $\times$ 64 channels $\times$ 72 time frames. The first 14 measurements were associated to a group of the left hand stimuli, while the other ones were with the right hand stimuli. The order-4 ITPC tensor can be fitted by a multilinear CP model. Mørup *et al.* analyzed the dataset by nonnegative CP of three components and Tucker components and compared them with components extracted by NMF and ICA [63].

In this example, our aim was to compare the factorization time of ALS and FCP over various R in the range of $[5, 72]$ with and without a Tucker compression prior to the CP decompositions. The FCP method employed ALS to factorize the order-3 unfolded tensor, and the fast ALS for structured Kruskal tensors. Interpretation of the results can be found in [9], [63]. The low-rank FCP algorithm was applied with the unfolding rule $\boldsymbol{l} = [1, 2, (3, 4)]$.

Execution time for each algorithm was averaged over 10 Monte Carlo runs with different initial values and illustrated in Fig. 6 for various R. For relatively low rank R, a prior Tucker compression sped up ALS, and made it more efficient than FCP when $R \leq 11$. The reason is explained by compression time for unfolding tensor in FCP. However, this acceleration technique was less efficient as $R \rightarrow I_n$, and inapplicable to ALS for $R \geq I_n$. FCP significantly reduced the execution time of ALS by a factor of 5–60 times, and was slightly improved by the prior compression. Comparison of fits explained by algorithms in Table II indicates that while FCP (Algorithm 2) quickly factorized the data, its fit was equivalent to that of ALS.

For this data, R1FCP, unfortunately, did not work well. Fits of this algorithm are given in Table II. Performance of this algorithm with several unfolding rules including $[(1, 2), 3, 4]$, $[1, (2, 3), 4]$, $[1, 2, (3, 4)]$ and $[1, 3, (2, 4)]$ is compared in Table III. When $R = 8$ and using the rule $\boldsymbol{l} = [(1, 2), 3, 4]$, R1FCP showed the worst performance with a fit of 26.7% which was not competitive to a fit of 43.8% obtained by ALS.

TABLE II
COMPARISON OF FIT ($= \left(1 - \frac{\|\mathcal{Y} - \widehat{\mathcal{Y}}\|_F}{\|\mathcal{Y}\|_F}\right) 100\%$) VALUES IN FACTORIZATION OF THE ITPC TENSOR BY ALS AND FCP IN EXAMPLE 6. R1FCP COMPLETELY FAILED IN THIS EXAMPLE. STRIKETHROUGH VALUES MEAN THAT THE ALGORITHM DID NOT CONVERGE TO THE DESIRED SOLUTION

R	ALS	R1FCP [(1,2),3,4]	R1FCP [1,2,(3, 4)]	FCP	Tucker → ALS	Tucker → FCP
5	37.7 ± 0.17	32.7	36.0	37.7 ± 0.17	36.2 ±0.03	36.9 ±0.00
8	43.8 ± 0.13	26.7	42.1	43.8 ± 0.00	42.7 ±0.01	42.9 ±0.58
11	48.0 ± 0.04	19.5	32.5	47.9 ± 0.16	47.1 ±0.09	47.5 ±0.08
20	56.0 ± 0.12	~~-61.1~~	~~-10.8~~	55.9 ± 0.16	55.5 ±0.18	55.8 ±0.13
30	61.1 ± 0.09	~~-519~~	~~-301.0~~	61.1 ± 0.09	60.8 ±0.07	60.9 ±0.13
40	64.5 ± 0.08	~~-649~~	~~-319.1~~	64.4 ± 0.14	64.2 ±0.08	64.3 ±0.18
60	68.7 ± 0.02	~~-1295~~	~~-432.8~~	68.1 ± 0.05	68.6 ±0.09	68.6 ±0.10
72	70.4 ± 0.02	~~-7384~~	~~-535.0~~	69.9 ± 0.08		

The average collinearity degrees of the estimated components $c_n = \frac{\sum_{r,s} |c_{r,s}^{(n)}|}{R(R-1)} = [0.48, \mathbf{0.70}, 0.54, \mathbf{0.89}]$ indicates that we should not fold modes 1 and 2; in addition, folding modes 2 and 4 which had the largest collinear degrees is suggested, i.e., the unfolding rule $\boldsymbol{l} = [1, 3, (2, 4)]$. It is clear to see that the unfolding rule $\boldsymbol{l} = [1, 3, (2, 4)]$ significantly improved performance of R1FCP with a fit of 41.3%. Moreover, the unfolding rule $\boldsymbol{l} = [1, 3, (2, 4)]$ was also suggested according to the average collinear degrees $c_n = [0.37, \mathbf{0.50}, 0.40, \mathbf{0.83}]$ achieved when applying the unfolding rule $\boldsymbol{l} = [1, (2, 3), 4]$. This confirms the applicability of the suggested unfolding strategy. For this test case, the unfolding rule $\boldsymbol{l} = [1, 2, (3, 4)]$ allowed to achieve the best fit of 42.1%, although this rule was not suggested by the strategy. This can be understood due to the fact that the average collinear degrees of modes 2 and 3 were very similar (0.50 and 0.49, or 0.52 and 0.50, see in Table III).

For higher ranks, e.g., $R \geq 11$, R1FCP completely failed. The unfolding strategy did not help anymore (see fit values in Table II). In Fig. 7, we display leading singular values of reshaped matrices $\mathbf{F}_r$ $(r = 1, 2, \ldots, R)$ from the estimated components for $R = 8$ and 20. The results indicate that $\mathbf{F}_r$ were not rank-one matrices, especially the matrices received when using the rule $\boldsymbol{l} = [(1, 2), 3, 4]$. Note that R1FCP works if and only if all $\mathbf{F}_r$ are rank-one.

Fig. 6 illustrates the relative approximation errors $\varepsilon = \frac{\|\mathcal{Y} - \widehat{\mathcal{Y}}\|_F}{\|\mathcal{Y}\|_F}$ of ALS and FCP for $R = 20$ as functions of the execution time. ALS took 536.5 seconds to converge. FCP took 1.2 seconds for compression, 0.9 seconds for CPD of the order-3 unfolded tensor, 2.73 seconds for low-rank approximations, 2.1 seconds for the refinement stage. ALS and FCP

TABLE III
PERFORMANCE OF RANK-1 FCP WITH DIFFERENT UNFOLDING RULES IN DECOMPOSITION OF THE ITPC TENSOR IN EXAMPLE 6. STRIKETHROUGH VALUES MEAN THAT THE ALGORITHM DID NOT CONVERGE TO THE DESIRED SOLUTION

R	R1FCP Unfolding rules	R1FCP Fit (%)	R1FCP Time (s)	R1FCP Collinearity degree $[c_1, c_2, c_3, c_4]$	ALS Fit (%)	ALS Time (s)
8	[(1, 2), 3, 4]	26.7	2.15	[0.48, 0.70, 0.54, 0.89]	43.8	242
	[1, (2, 3), 4]	29.9	5.72	[0.37, 0.50, 0.40, 0.83]		
	[1, 2, (3, 4)]	42.1	5.59	[0.34, 0.50, 0.49, 0.72]		
	[1, 3, (2, 4)]	41.3	4.67	[0.49, 0.52, 0.50, 0.75]		
20	[(1, 2), 3, 4]	~~61.1~~	5.79	[0.43, 0.62, 0.50, 0.89]	56.0	752
	[1, (2, 3), 4]	28.3	12.12	[0.34, 0.45, 0.56, 0.90]		
	[1, 2, (3, 4)]	15.4	10.39	[0.33, 0.48, 0.46, 0.86]		
	[1, 3, (2, 4)]	~~22.6~~	9.17	[0.39, 0.57, 0.69, 0.92]		

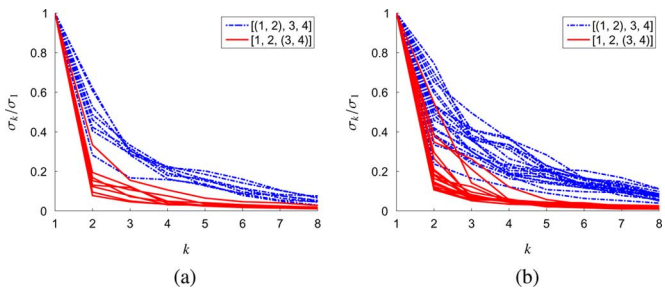

Fig. 7. Illustration of leading singular values $\boldsymbol{\sigma}_r$ of matrices $\mathbf{F}_r$ ($r = 1, 2, \ldots, R$) reshaped from components estimated from the ITPC tensor with different unfolding rules $[(1, 2), 3, 4]$ and $[1, 2, (3, 4)]$. The singular values are normalized by the largest values $\frac{\sigma_k}{\sigma_1}$. R1FCP failed in this experiment because this algorithm worked only if all $\mathbf{F}_r$ were rank-one matrices. (a) $R = 8$, (b) $R = 20$.

converged to the relative approximation errors $\varepsilon_{ALS} = 0.4417$, while $\varepsilon_{FCP} = 0.4399$, respectively.

Example 7 (Decomposition of Gabor Tensor of the ORL Face Database): This example illustrated classification of the ORL face database[2] consisting of 400 faces for 40 subjects. We constructed Gabor feature tensors for 8 different orientations at 4 scales which were then down-sampled to $16 \times 16 \times 8 \times 4 \times 400$ dimensional tensor $\boldsymbol{\mathcal{Y}}$. The unfolding $\boldsymbol{l} = [1, 2, (3, 4, 5)]$ was applied to unfold $\boldsymbol{\mathcal{Y}}$ to be an order-3 tensor. ALS [17] factorized both $\boldsymbol{\mathcal{Y}}$ and $\boldsymbol{\mathcal{Y}}_{[[\boldsymbol{l}]]}$ into $R = 30, 40, 60$ rank-1 tensors in 1000 iterations, and stopped when $|\varepsilon - \varepsilon_{old}| \leq 10^{-6}\varepsilon$. The R1FCP algorithm did not work for this data. For example, when $R = 10$ and applying the rule $\boldsymbol{l} = [1, 2, (3, 4, 5)]$, R1FCP explained the data with a fit of -31.2%, and yielded average collinearity degrees of $[0.60, 0.66, 0.64, 0.95, 0.64]$. Although a further decomposition with the unfolding rule $\boldsymbol{l} = [1, (2, 3), (4, 5)]$ achieved a better fit of 44.8%, this result was much lower than a fit of 54.5% obtained by ALS and FCP.

The factor $\mathbf{A}^{(5)} \in \mathbb{R}^{400 \times R}$ comprised compressed features which were used to cluster faces using the K-means algorithm. Table IV compares performance of two algorithms including execution time, fit, accuracy (ACC %) and normalized mutual information (NMI). For $R = 40$, ALS factorized $\boldsymbol{\mathcal{Y}}$ in 1599 seconds, while FCP completed the task in only 39 seconds with a slightly reduction of fit ($\approx$0.17%). For $R = 60$, ALS was extremely time consuming, required 16962 seconds, while FCP only took 105 seconds. Regarding the clustering accuracy, features extracted by FCP still achieved comparable performance as those obtained by ALS.

[2]http://www.cl.cam.ac.uk/research/dtg/attarchive/facedatabase.html

TABLE IV
COMPARISON BETWEEN ALS AND FCP IN FACTORIZATION OF ORDER-5 GABOR TENSOR CONSTRUCTED FROM THE ORL FACE DATASET

R	Algorithm	Fit (%)	Time (seconds)	Ratio $\frac{\text{ALS}}{\text{FCP}}$	ACC (%)	NMI (%)
30	FCP	60.59	24	**39**	85.00	92.91
	ALS	60.56	927		85.25	93.22
40	FCP	62.46	39	**41**	84.25	92.57
	ALS	62.63	1599		85.50	93.68
60	FCP	65.47	105	**162**	83.00	91.62
	ALS	65.64	16962		81.38	91.44

VII. DISCUSSION AND CONCLUSIONS

Through out analysis of CRIB and examples, we provided some guidelines on choosing unfolding. The most important recommendation is using the FCP algorithm to reduce affect due to inappropriate unfoldings. Moreover, a major rule is not to fold orthogonal or low collinear factor matrices if possible.

By decomposition of unfolded tensor with a prior lossy or lossless compression, the proposed FCP algorithm has been shown 40–160 times faster than ALS for decomposition of order-5 and -6 tensors. In addition, it also indicates that FCP is much less space consuming than other algorithms for higher order CPD, although this aspect has not yet been clearly discussed in the paper. FCP can also be applied to the CPD with one column-wise orthogonal factor matrix as seen in Example 5. For other constrained CPDs, one needs to employ suitable CP algorithms in stage 2, and derives an algorithm for structured CP used in stage 3b in Algorithm 2. Finally, an important application of FCP is that it can convey algorithms for order-3 CPD to higher order such as GRAM, DTLD.

APPENDIX A
ALGORITHM FOR STRUCTURED CPD

This section derives a fast ALS algorithm which approximates the structured tensor $\boldsymbol{\mathcal{Y}}_J$ given in (22) by a rank-R tensor $\boldsymbol{\mathcal{Y}}$. We first compute the CP gradients [21], [47]

$$\left(\mathbf{Y}_{J(n)} - \mathbf{Y}_{(n)}\right)\left(\bigodot_{k \neq n} \mathbf{A}^{(k)}\right) = \tilde{\mathbf{A}}^{(n)} \operatorname{diag}(\tilde{\boldsymbol{\lambda}})\, \mathbf{W}_n - \mathbf{A}^{(n)}\, \boldsymbol{\Gamma}_n, \quad (28)$$

for $n = 1, \ldots, N$, where $\mathbf{W}_n = \left(\circledast_{k \neq n}(\tilde{\mathbf{A}}^{(n)T}\, \mathbf{A}^{(n)})\right)$ and $\boldsymbol{\Gamma}_n = \left(\circledast_{k \neq n}(\mathbf{A}^{(k)T}\, \mathbf{A}^{(k)})\right)$. Taking into account that $(\mathbf{M}^T\mathbf{A}) \circledast (\mathbf{M}^T\mathbf{B}) = \mathbf{M}^T(\mathbf{A} \circledast \mathbf{B})$, $\mathbf{M}((\mathbf{M}^T\mathbf{A}) \circledast \mathbf{B}) = \mathbf{A} \circledast (\mathbf{MB})$, where $\mathbf{M}$ is defined in (26), $\mathbf{A}$ and $\mathbf{B}$ are matrices

of compatible dimensions}, the first term in (28) is further expressed by

$$\tilde{\mathbf{A}}^{(n)}\,\mathrm{diag}(\tilde{\boldsymbol{\lambda}})\,\mathbf{W}_n = \mathbf{B}^{(n)}\left(\left(\circledast_{\substack{k=1\\k\neq n}}^{N-2}\left(\mathbf{B}^{(k)T}\,\mathbf{A}^{(k)}\right)\right)\circledast\mathbf{K}\right),\ n=1,\ldots,N-2,$$

$$\tilde{\mathbf{A}}^{(N-1)}\,\mathrm{diag}(\tilde{\boldsymbol{\lambda}})\,\mathbf{W}_{N-1} = \tilde{\mathbf{A}}^{(N-1)}\begin{bmatrix}\lambda_1\tilde{\mathbf{V}}_1^T\,\mathbf{A}^{(N)}\,\mathrm{diag}(\boldsymbol{\omega}_1)\\ \vdots\\ \lambda_R\tilde{\mathbf{V}}_R^T\,\mathbf{A}^{(N)}\,\mathrm{diag}(\boldsymbol{\omega}_R)\end{bmatrix},$$

$$\tilde{\mathbf{A}}^{(N)}\,\mathrm{diag}(\tilde{\boldsymbol{\lambda}})\,\mathbf{W}_N = \tilde{\mathbf{A}}^{(N)}\begin{bmatrix}\lambda_1\mathbf{U}_1^T\,\mathbf{A}^{(N-1)}\,\mathrm{diag}(\boldsymbol{\omega}_1)\\ \vdots\\ \lambda_R\mathbf{U}_R^T\,\mathbf{A}^{(N-1)}\,\mathrm{diag}(\boldsymbol{\omega}_R)\end{bmatrix},$$

$$\mathbf{K} = \begin{bmatrix}\lambda_1\mathbf{1}_R^T\left(\left(\mathbf{U}_1^T\,\mathbf{A}^{(N-1)}\right)\circledast\left(\tilde{\mathbf{V}}_r^T\,\mathbf{A}^{(N)}\right)\right)\\ \vdots\\ \lambda_R\mathbf{1}_R^T\left(\left(\mathbf{U}_R^T\,\mathbf{A}^{(N-1)}\right)\circledast\left(\tilde{\mathbf{V}}_R^T\,\mathbf{A}^{(N)}\right)\right)\end{bmatrix},$$

where $\boldsymbol{\Omega} = [\boldsymbol{\omega}_r] = \circledast_{k=1}^{N-2}(\mathbf{A}^{(k)T}\,\mathbf{B}^{(k)})$. For each $n = 1,2,\ldots,N$, this computation has a low computational complexity of order $\mathcal{O}\left(R^2\left(N+\sum_{n=1}^{N-2}I_n\right)+JR(I_{N-1}+I_N)\right)$. Using the results, we can derive fast update rules to estimate $\boldsymbol{\mathcal{Y}}$. For example,

$$\mathbf{A}^{(n)} \leftarrow \mathbf{B}^{(n)}\left(\left(\circledast_{\substack{k=1\\k\neq n}}^{N-2}(\mathbf{B}^{(n)T}\mathbf{A}^{(n)})\right)\circledast\mathbf{K}\right)\boldsymbol{\Gamma}_n^{-1},\quad n=1,\ldots,N-2.$$

Appendix B
Cramér-Rao Induced Bound for Angular Error

Theorem B.1: [41]: The Cramér-Rao induced bound (CRIB) on α_1^2 for rank-2 tensor is given by

$$\mathrm{CRIB}(\boldsymbol{a}_1) = \frac{\sigma^2}{\lambda_1^2}\left(\frac{I_1-1}{1-h_1^2}+\frac{(1-c_1^2)h_1^2}{1-h_1^2}\times\frac{y^2+z-h_1^2z(z+1)}{(1-c_1y-h_1^2(z+1))^2+h_1^2(y+c_1z)^2}\right) \quad (29)$$

where $c_n = a_1^{(n)T}a_2^{(n)}$, and

$$h_n = \prod_{2\le k\neq n}^{N} c_k \quad \text{for} \quad n=1,\ldots,N, \quad (30)$$

$$y = -c_1\sum_{n=2}^{N}\frac{h_n^2(1-c_n^2)}{c_n^2-h_n^2c_1^2}, \quad (31)$$

$$z = \sum_{n=2}^{N}\frac{1-c_n^2}{c_n^2-h_n^2c_1^2}. \quad (32)$$

For order-3 rank-2 tensors, $\mathrm{CRIB}(a_1)$ is given by

$$\mathrm{CRIB}(\boldsymbol{a}_1) = \frac{\sigma^2}{\lambda_1^2}\left(\frac{I_1-2}{1-h_1^2}+\frac{1}{1+h_1^2-c_2^2-c_3^2}\right). \quad (33)$$

Acknowledgment

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

References

[1] R. A. Harshman, "Foundations of the PARAFAC procedure: Models and conditions for an explanatory multimodal factor analysis," *UCLA Work. Papers Phonet.*, vol. 16, pp. 1–84, 1970.
[2] 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.
[3] A. Smilde, R. Bro, and P. Geladi, *Multi-Way Analysis: Applications in the Chemical Sciences*. New York, NY, USA: Wiley, 2004.
[4] G. Tomasi, "Practical and computational aspects in chemometric data analysis," Ph.D. dissertation, Food Sci., Quality and Technol. section, The Royal Veterinary and Agricultural Univ., Frederiksberg, Denmark, 2006.
[5] L. De Lathauwer and J. Castaing, "Tensor-based techniques for the blind separation of DS-CDMA signals," *Signal Process.*, vol. 87, no. 2, pp. 322–336, Feb. 2007.
[6] 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.
[7] A. L. F. de Almeida, X. Luciani, A. Stegeman, and P. Comon, "CONFAC decomposition approach to blind identification of underdetermined mixtures based on generating function derivatives," *IEEE Trans. Signal Process.*, vol. 60, no. 11, pp. 5698–5713, 2012.
[8] L.-H. Lim and P. Comon, "Blind multilinear identification," *IEEE Trans. Inf. Theory*, to be published.
[9] 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.
[10] 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.
[11] B. W. Bader, M. W. Berry, and M. Browne, "Discussion tracking in Enron email using PARAFAC," in *Survey of Text Mining II*, M. W. Berry and M. Castellanos, Eds. London, U.K.: Springer, 2008, pp. 147–163.
[12] D. Gonzlez, A. Ammar, F. Chinesta, and E. Cueto, "Recent advances on the use of separated representations," *Int. J. Numer. Methods Eng.*, vol. 81, no. 5, pp. 637–659, 2010.
[13] P. Constantine, A. Doostan, Q. Wang, and G. Iaccarino, "A surrogate accelerated Bayesian inverse analysis of the HyShot II supersonic combustion data," in *Proc. 13th AIAA Non-Deterministic Approaches Conf.*, Denver, CO, USA, 2010, vol. 6, pp. 4867–4875, AIAA-2011-2037.
[14] X. Luciani, A. L. F. de Almeida, and P. Comon, "Blind identification of underdetermined mixtures based on the characteristic function: The complex case," *IEEE Trans. Signal Process.*, vol. 59, no. 2, pp. 540–553, 2011.
[15] A.-H. Phan and A. Cichocki, "Tensor decompositions for feature extraction and classification of high dimensional datasets," *IEICE Nonlin. Theory Appl.*, vol. 1, pp. 37–68, 2010, invited paper.
[16] P. Comon, "Tensors, usefulness and unexpected properties," in *Proc. IEEE Workshop Statist. Signal Process. (SSP'09)*, 2009, pp. 781–788 [Online]. Available: http://hal.archives-ouvertes.fr/hal-00417258/
[17] C. A. Andersson and R. Bro, "The N-way toolbox for MATLAB," *Chemometr. Intell. Lab. Syst.* , vol. 52, no. 1, pp. 1–4, 2000.
[18] M. Rajih, P. Comon, and R. A. Harshman, "Enhanced line search: A novel method to accelerate PARAFAC," *SIAM J. Matrix Anal. Appl.*, vol. 30, no. 3, pp. 1128–1147, 2008.

[19] Y. Chen, D. Han, and L. Qi, "New ALS methods with extrapolating search directions and optimal step size for complex-valued tensor decompositions," *IEEE Trans. Signal Process.*, vol. 59, no. 12, pp. 5888–5898, 2011.

[20] P. Paatero, C. Navasca, and P. Hopke, "Fast rotationally enhanced alternating-least-squares," presented at the Proc. SIAM Workshop Tensor Decomposit. Appl., 2010.

[21] 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.

[22] 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, Feb. 2011.

[23] J.-P. Royer, N. T. Moreau, and P. Comon, "Computing the polyadic decomposition of nonnegative third order tensors," *Signal Process.*, vol. 91, no. 9, pp. 2159–2171, 2011.

[24] P. Paatero, "A weighted non-negative least squares algorithm for three-way PARAFAC factor analysis," *Chemometr. Intell. Lab. Syst.*, vol. 38, no. 2, pp. 223–242, 1997.

[25] P. Tichavský and Z. Koldovský, "Simultaneous search for all modes in multilinear models," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process. (ICASSP10)*, 2010, pp. 4114–4117.

[26] 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.

[27] L. De Lathauwer, "A link between the canonical decomposition in multilinear algebra and simultaneous matrix diagonalization," *SIAM J. Matrix Anal. Appl.*, vol. 28, pp. 642–666, 2006.

[28] L. De Lathauwer and J. Castaing, "Blind identification of underdetermined mixtures by simultaneous matrix diagonalization," *IEEE Trans. Signal Process.*, vol. 56, no. 3, pp. 1096–1105, 2008.

[29] A. Franc, "Etude Algebrique des Multitableaux: Apports de L'algebrique Tensorielle," Ph.D. Thesis, Université de Montpellier II—Science et Techniques du Languedoc, Montpellier, France, 1992.

[30] 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.

[31] 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, Sep. 2008.

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

[33] J. B. Kruskal, "Three-way arrays: Rank and uniqueness of trilinear decompositions, with application to arithmetic complexity and statistics," *Linear Algebra Appl.*, vol. 18, pp. 95–138, 1977.

[34] A. Stegeman and N. D. Sidiropoulos, "On Kruskal's uniqueness condition for the Candecomp/Parafac decomposition," *Linear Algebra Appl.*, vol. 420, no. 23, pp. 540–552, 2007.

[35] A. Stegeman, "On uniqueness of the nth order tensor decomposition into rank-1 terms with linear independence in one mode," *SIAM J. Matrix Anal. Appl.*, vol. 31, no. 5, pp. 2498–2516, 2010.

[36] A. Stegeman, "On uniqueness of the canonical tensor decomposition with some form of symmetry," *SIAM J. Matrix Anal. Appl.*, vol. 32, no. 2, pp. 561–583, 2011.

[37] N. D. Sidiropoulos and R. Bro, "On the uniqueness of multilinear decomposition of N-way arrays," *J. Chemometr.*, vol. 14, no. 3, pp. 229–239, 2000.

[38] T. G. Kolda and B. W. Bader, "Tensor decompositions and applications," *SIAM Rev.*, vol. 51, no. 3, pp. 455–500, Sept. 2009.

[39] A. Karfoul, L. Albera, and L. De Lathauwer, "Canonical decomposition of even higher order cumulant arrays for blind underdetermined mixture identification," in *Proc. Sens. Array Multichannel Signal Process. Workshop*, Jul. 2008, pp. 501–505.

[40] M. Sørensen, L. Lathauwer, P. Comon, S. Icart, and L. Deneire, "Canonical polyadic decomposition with a columnwise orthonormal factor matrix," *SIAM J. Matrix Anal. Appl.*, vol. 33, no. 4, pp. 1190–1213, 2012.

[41] 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.

[42] 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.

[43] P. Paatero, "Construction and analysis of degenerate PARAFAC models," *J. Chemometr.*, vol. 14, no. 3, pp. 285–299, 2000.

[44] C. A. Andersson and R. Bro, "Improving the speed of multi-way algorithms: Part I. Tucker3," *Chemometr. Intell. Lab. Syst.*, vol. 42, pp. 93–103, 1998.

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

[46] L. R. Tucker, "The extension of factor analysis to three-dimensional matrices," in *Contributions to Mathematical Psychology*, H. Gulliksen and N. Frederiksen, Eds. New York, NY, USA: Holt, Rinehart,Winston, 1964, pp. 110–127.

[47] 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.

[48] G. Tomasi, "Recent developments in fast algorithms for fitting the PARAFAC model," in *Proc. TRICAP*, Crete, Greece, 2006 [Online]. Available: http://www.telecom.tuc.gr/~nikos/TR-ICAP2006main/TomasiTRICAP2006.ppt

[49] P. Tichavský, A.-H. Phan, and A. Cichocki, "A further improvement of a fast damped GAUSS-NEWTON algorithm for CANDECOMP-PARAFAC tensor decomposition," in *Proc. of IEEE Int. Conf. Acoust., Speech, Signal Process. (ICASSP)*, 2013, pp. 5964–5968.

[50] H. A. L. Kiers and R. A. Harshman, "Relating two proposed methods for speedup of algorithms for fitting two- and three-way principal component and related multilinear models," *Chemometr. Intell. Lab. Syst.*, vol. 36, pp. 31–40, 1997.

[51] L. De Lathauwer, B. De Moor, and J. Vandewalle, "A multilinear singular value decomposition," *SIAM J. Matrix Anal. Appl.*, vol. 21, no. 4, pp. 1253–1278, 2000.

[52] S. Ragnarsson and C. F. Van Loan, "Block tensor unfoldings," *SIAM J. Matrix Anal. Appl.* , vol. 33, no. 1, pp. 149–169, 2012.

[53] P. Tichavský and Z. Koldovský, "Stability of CANDECOMP-PARAFAC tensor decomposition," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process. (ICASSP)*, 2011, pp. 4164–4167.

[54] 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.

[55] X. Q. Liu and N. D. Sidiropoulos, "Cramer-Rao lower bounds for low-rank decomposition of multidimensional arrays," *IEEE Trans. Signal Process.*, vol. 49, no. 9, pp. 2074–2086, 2001.

[56] P. K. Hopke, P. Paatero, H. Jia, R. T. Ross, and R. A. Harshman, "Three-way (PARAFAC) factor analysis: Examination and comparison of alternative computational methods as applied to ill-conditioned data," *Chemometr. Intell. Lab. Syst.*, vol. 43, no. 1–2, pp. 25–42, 1998.

[57] A. Stegeman, "Using the simultaneous generalized Schur decomposition as a CANDECOMP/PARAFAC algorithm for ill-conditioned data," *J. Chemometr.*, vol. 23, no. 7–8, pp. 385–392, 2009.

[58] H. A. L. Kiers and R. A. Harshman, "An efficient algorithm for Parafac with uncorrelated mode—A components applied to large $I \times J \times K$ data sets with $I \gg JK$," *J. Chemometr.*, vol. 23, no. 7–8, pp. 442–447, 2009.

[59] M. B. Dosse, J. M. F. Berge, and J. N. Tendeiro, "Some new results on orthogonally constrained Candecomp," *J. Classification*, vol. 28, pp. 144–155, 2011.

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

[61] K. S. Booksh and B. R. Kowalski, "Error analysis of the generalized rank annihilation method," *J. Chemometr.*, vol. 8, no. 1, pp. 45–63, 1994.

[62] E. Sanchez and B. R. Kowalski, "Tensorial resolution: A direct trilinear decomposition," *J. Chemometr.*, vol. 4, no. 1, pp. 29–45, 1990.

[63] M. Mørup, L. K. Hansen, and S. M. Arnfred, "ERPWAVELAB a toolbox for multi-channel analysis of time-frequency transformed event related potentials," *J. Neurosci. Methods*, vol. 161, no. 2, pp. 361–368, 2006.

Anh-Huy Phan (M'13) received the master's degree from Hochiminh University of Technology, Ho Chi Minh City, Vietnam, in 2005, and the Ph.D. degree from the Kyushu Institute of Technology, Japan, in 2011.

He worked as Deputy Head of the Research and Development Department, Broadcast Research and Application Center, Vietnam Television, and is currently a Research Scientist at the Laboratory for Advanced Brain Signal Processing, and a Visiting Research Scientist with the Toyota Collaboration Center, Brain Science Institute, RIKEN. His research interests include multilinear algebra, tensor computation, blind source separation, and brain computer interface.

Petr Tichavský (M'98–SM'04) received the M.S. degree in mathematics, in 1987, from the Czech Technical University, Prague, Czechoslovakia, and the Ph.D. degree in theoretical cybernetics from the Czechoslovak Academy of Sciences, in 1992.

Since that time he has been with the Institute of Information Theory and Automation, Academy of Sciences of the Czech Republic, Prague. In 1994, he received a*Fulbright grant* for a 10 month fellowship with the Department of Electrical Engineering, Yale University, New Haven, CT, USA. He is the 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.

Dr. Tichavský received the Otto Wichterle Award from the Academy of Sciences of the Czech Republic in 2002. He served as an 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 again from 2011 until the present. He also served as a General Co-Chair of the 36th IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) held in Prague in 2011.

Andrzej Cichocki (M'96–SM'06–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, 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, RIKEN Brain Science Institute, Japan, and Professor in IBS, PAN, Poland. His researches focus on tensor decompositions, brain machine interface, human robot interactions, EEG hyper-scanning, brain to brain interface, and their practical applications.

Dr. Cichocki has served as an Associate Editor of the IEEE TRANSACTIONS ON NEURAL NETWORKS, 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 four monographs in English (two of them translated to Chinese). His publications currently report over 19,000 citations according to Google Scholar, with an h-index of 59.