Efficient Computation of the PARAFAC2 Decomposition via Generalized Tensor Contractions

Kristina Naskovska [*], *Yao Cheng* [*], *André L. F. de Almeida* [§], and *Martin Haardt* [*]
[*]Communications Research Laboratory, Ilmenau University of Technology, P. O. Box 100565, D-98684 Ilmenau, Germany
[§]Department of Teleinformatics Engineering, Federal University of Ceará (UFC), Fortaleza, Brazil

***Abstract*—The PARAFAC2 decomposition is a generalization of the PARAFAC/CP (Canonical Polyadic) tensor decomposition. This tensor decomposition represents a set of coupled matrix decompositions with one mode in common, i.e., one of the components varies along the set of matrices (tensor slices), whereas the second component stays constant. It has many applications including the analysis of chromatographic data, data clustering, and biomedical signal processing. Typically, the PARAFAC2 decomposition is described using a slice-wise notation, where each of the slices corresponds to one of the matrix decompositions. In this paper, we first present a generalization of the slice-wise multiplication based on the tensor contraction operator, where the generalized contraction operator represents an inner product between two tensors of compatible dimensions. Next, we express the PARAFAC2 decomposition in terms of this new tensor model resulting in a constrained CP model. Moreover, we show that this new description opens an efficient single loop way to compute the PARAFAC2 decomposition based on the generalized tensor contraction.**

I. Introduction

The PARAFAC2 tensor decomposition initially proposed in [1] has many applications in multidimensional data analytics. For instance, it can be used in biomedical application when analyzing time-shifted signals. In [2] and [3] the PARAFAC2 decomposition is used for the identification of the dominant signal components in EEG signals resulting from visual-evoked potentials for each of the different time-shifted channels. Moreover, in [4] PARAFAC2 is used for the analysis of somatosensory evoked magnetic fields and somatosensory evoked electrical potentials. Furthermore, an algorithm for the computation of the coupled PARAFAC2 decomposition is proposed in [5] for the joint analysis of somatosensory evoked magnetic fields and electrical potentials.

Usually the PARAFAC2 decomposition is defined using a slice-wise (matrix) representation. This slice-wise description does not reveal the tensor structure explicitly. As an alternative, we propose to represent the slice-wise multiplication by a generalized contraction of two tensors to derive an explicit tensor description (constrained CP (Canonical Polyadic) decomposition) of the data tensor. The tensor contraction defines an inner product between two tensors with compatible dimensions [6]. Moreover, in [7] and [8] we show that the tensor contraction can be efficiently computed using the concept of the generalized unfoldings. The resulting constrained CP representation leads to a new tensor model that does not depend on a chosen unfolding. In this paper we exploit this representation to reveal the tensor structure of the PARAFAC2 model.

The computation of the PARAFAC2 decomposition can be performed based on indirect fitting algorithms and direct fitting algorithms. The indirect fitting approach originally proposed in [9] tries to fit the cross product that is equivalent to a symmetric PARATUCK2 decomposition [10]. On the other hand, in [11] a direct fitting algorithm is proposed consisting of two loops for the computation of the PARAFAC2 decomposition. In this paper, we propose a single loop direct fitting algorithm for PARAFAC2 that has been derived via generalized tensor contractions. The proposed single loop algorithm requires fewer iterations as compared to algorithms with two loops.

The rest of the paper is organized as follows. In Section II we introduce the notation and the contraction properties for element-wise and slice-wise multiplications. The novel formulation of the PARAFAC2 model as constrained CP is presented in Section III. The single loop, direct fitting algorithm based on this model is presented in Section IV. The performance of the proposed algorithm is evaluated in Section V based on simulation results. Finally, in Section VI we conclude this paper.

II. Contraction Properties for Element-wise and Slice-wise Multiplication

We use the following notation. Scalars are denoted either as capital or lower-case italic letters, A, a. Column vectors and matrices are denoted as bold-faced lower-case and capital letters, $\boldsymbol{a}, \boldsymbol{A}$, respectively. Tensors are represented by bold-faced calligraphic letters $\boldsymbol{\mathcal{A}}$. The following superscripts, ^{T}, ^{H}, $^{-1}$, and $^+$ denote transposition, Hermitian transposition, matrix inversion, and Moore-Penrose pseudo inversion, respectively. The outer product, Kronecker product, and Khatri-Rao product are denoted as $\circ$, $\otimes$, and $\diamond$, respectively. The operators $\|.\|_{\mathrm{F}}$ and $\|.\|_{\mathrm{H}}$ denote the Frobenius norm and the higher order norm, respectively. Moreover, the n-mode product between a tensor $\boldsymbol{\mathcal{A}} \in \mathbb{C}^{I_1 \times I_2 \ldots \times I_N}$ and a matrix $\boldsymbol{B} \in \mathbb{C}^{J \times I_n}$ is defined as $\boldsymbol{\mathcal{A}} \times_n \boldsymbol{B}$, for $n = 1, 2, \ldots N$. A super-diagonal or identity N-way tensor of dimension $R \times R \ldots \times R$ is denoted as $\boldsymbol{\mathcal{I}}_{N,R}$. Similarly, an identity matrix of dimension $R \times R$ is denoted as $\boldsymbol{I}_R$ and we denote a column vector of ones of length R as $\boldsymbol{1}_R$. The n-th 3-mode slice of a tensor $\boldsymbol{\mathcal{A}} \in \mathbb{C}^{I \times J \times N}$ is denoted as $\boldsymbol{\mathcal{A}}_{(:,:,n)}$ and accordingly one element of this tensor is denoted as $\boldsymbol{\mathcal{A}}_{(i,j,n)}$. The first mode vectors of the tensor $\boldsymbol{\mathcal{A}}$ are denoted by $\boldsymbol{\mathcal{A}}_{(:,j,n)}$. Furthermore, a matricization (which means transforming a tensor into a matrix) is also called an unfolding or flattening. For instance, the set of modes $(1, 2, ..., N)$ of a N-way tensor $\boldsymbol{\mathcal{A}}$ can be divided into two non-overlapping, P and $N - P$ dimensional subsets, $\alpha^{(1)} = [\alpha_1 \ldots \alpha_P]$ and $\alpha^{(2)} = [\alpha_{P+1} \ldots \alpha_N]$, respectively. This leads to the generalized unfolding $[\boldsymbol{\mathcal{A}}]_{(\alpha^{(1)},\alpha^{(2)})}$, where the indices contained in $\alpha^{(1)}$ vary along the rows and the

978-1-5386-9218-9/18/$31.00 ©2018 IEEE

indices contained in $\alpha^{(2)}$ vary along the columns [12]. The operator diag(.) transforms a vector into a diagonal matrix and the operator vec(.) transforms a matrix into a vector.

The contraction $\boldsymbol{\mathcal{A}} \bullet_p^m \boldsymbol{\mathcal{C}}$ between two tensors $\boldsymbol{\mathcal{A}} \in \mathbb{C}^{I_1 \times I_2 \ldots \times I_N}$ and $\boldsymbol{\mathcal{C}} \in \mathbb{C}^{J_1 \times J_2 \ldots \times J_N}$ represents an inner product of the n-th mode of $\boldsymbol{\mathcal{A}}$ with the m-th mode of $\boldsymbol{\mathcal{C}}$, provided that $I_n = J_m$ [6]. Contraction along several modes of compatible dimensions is also possible, which is called generalized contraction. Accordingly, the generalized contraction along two modes is denoted as $\boldsymbol{\mathcal{A}} \bullet_{n,k}^{m,l} \boldsymbol{\mathcal{C}}$. In particular, the contraction along two modes between the tensors $\boldsymbol{\mathcal{A}} \in \mathbb{C}^{I \times J \times M \times N}$ and $\boldsymbol{\mathcal{C}} \in \mathbb{C}^{M \times N \times K}$ is defined as [6],

$$(\boldsymbol{\mathcal{A}} \bullet_{3,4}^{1,2} \boldsymbol{\mathcal{C}})_{(i,j,k)} \triangleq \sum_{n=1}^{N} \sum_{m=1}^{M} \boldsymbol{\mathcal{A}}_{(i,j,m,n)} \cdot \boldsymbol{\mathcal{C}}_{(m,n,k)} = \boldsymbol{\mathcal{T}}_{(i,j,k)}.$$

This example represents a contraction of the 3-rd and 4-th mode of $\boldsymbol{\mathcal{A}}$ with the 1-st and 2-nd mode of $\boldsymbol{\mathcal{C}}$, respectively.

Using the concept of the generalized unfoldings [13], [7], it can be shown that the tensor contraction satisfies

$$[\boldsymbol{\mathcal{A}} \bullet_{3,4}^{1,2} \boldsymbol{\mathcal{C}}]_{([1,2],3)} = [\boldsymbol{\mathcal{A}}]_{([1,2],[3,4])} \cdot [\boldsymbol{\mathcal{C}}]_{([1,2],3)} =$$
$$[\boldsymbol{\mathcal{A}} \bullet_{4,3}^{2,1} \boldsymbol{\mathcal{C}}]_{([1,2],3)} = [\boldsymbol{\mathcal{A}}]_{([1,2],[4,3])} \cdot [\boldsymbol{\mathcal{C}}]_{([2,1],3)}.$$

In the generalized unfolding $[\boldsymbol{\mathcal{A}}]_{([1,2],[3,4])}$ the 1-st and the 2-nd mode vary along the rows and the 3-rd and the 4-th mode vary along the columns of the resulting matrix.

First, let us consider a Hadamard product (element-wise multiplication) between two vectors $\boldsymbol{a} \in \mathbb{C}^{M \times 1}$ and $\boldsymbol{b} \in \mathbb{C}^{M \times 1}$, $\boldsymbol{c} \in \mathbb{C}^{M \times 1}$ with elements $\boldsymbol{c}_{(m)} = \boldsymbol{a}_{(m)} \boldsymbol{b}_{(m)}$, $\forall m = 1, \ldots, M$. The Hadamard product can be expressed via the multiplication of a diagonal matrix and a vector, $\boldsymbol{a} \odot \boldsymbol{b} = \text{diag}(\boldsymbol{a})\boldsymbol{b} = \text{diag}(\boldsymbol{b})\boldsymbol{a}$. The matrix multiplication is equivalent to the contraction $\bullet_2^1$. Therefore, we get

$$\boldsymbol{c} = \boldsymbol{a} \odot \boldsymbol{b} = \text{diag}(\boldsymbol{a}) \bullet_2^1 \boldsymbol{b} = \text{diag}(\boldsymbol{b}) \bullet_2^1 \boldsymbol{a}.$$

Likewise, for a Hadamard product between two row vectors, we get $\boldsymbol{a}^{\mathrm{T}} \odot \boldsymbol{b}^{\mathrm{T}} = \boldsymbol{a}^{\mathrm{T}} \bullet_2^1 \text{diag}\left(\boldsymbol{b}^{\mathrm{T}}\right) = \boldsymbol{b}^{\mathrm{T}} \bullet_2^1 \text{diag}\left(\boldsymbol{a}^{\mathrm{T}}\right)$.

Next, for the Hadamard product between two matrices $\boldsymbol{A} \in \mathbb{C}^{M \times N}$ and $\boldsymbol{B} \in \mathbb{C}^{M \times N}$, $\boldsymbol{C}_{(m,n)} = \boldsymbol{A}_{(m,n)} \boldsymbol{B}_{(m,n)}$, $\forall m = 1, \ldots, M$ and $n = 1, \ldots, N$, we can show that $\boldsymbol{C} = \boldsymbol{A} \odot \boldsymbol{B} = \boldsymbol{\mathcal{D}}_A \bullet_{2,4}^{1,2} \boldsymbol{B} = \boldsymbol{\mathcal{D}}_B \bullet_{2,4}^{1,2} \boldsymbol{A}$. Here $\boldsymbol{\mathcal{D}}_A \in \mathbb{C}^{M \times M \times N \times N}$ and $\boldsymbol{\mathcal{D}}_B \in \mathbb{C}^{M \times M \times N \times N}$ are diagonal tensors where the non-zero elements $\boldsymbol{\mathcal{D}}_{A(m,m,n,n)} = \boldsymbol{A}_{(m,n)}$ and $\boldsymbol{\mathcal{D}}_{B(m,m,n,n)} = \boldsymbol{B}_{(m,n)}$, respectively. As an alternative, we also have

$$\boldsymbol{C} = \boldsymbol{A} \odot \boldsymbol{B} = \boldsymbol{\mathcal{D}}^{(A)} \bullet_{2,3}^{1,3} \boldsymbol{\mathcal{D}}^{(B)}$$

where the diagonal tensors have the following non-zero elements $\boldsymbol{\mathcal{D}}^{(A)}{}_{(m,m,n)} = \boldsymbol{A}_{(m,n)}$ and $\boldsymbol{\mathcal{D}}^{(B)}{}_{(m,n,n)} = \boldsymbol{B}_{(m,n)}$. Moreover, these diagonal tensors can be defined using tensor notation, i.e., $\boldsymbol{\mathcal{D}}^{(A)} = \boldsymbol{\mathcal{I}}_{3,M} \times_3 \boldsymbol{A}^{\mathrm{T}}$ and $\boldsymbol{\mathcal{D}}^{(B)} = \boldsymbol{\mathcal{I}}_{3,N} \times_1 \boldsymbol{B}$.

Furthermore, a slice-wise multiplication of two tensors $\boldsymbol{\mathcal{A}} \in \mathbb{C}^{M \times N \times K}$ and $\boldsymbol{\mathcal{B}} \in \mathbb{C}^{N \times J \times K}$ is defined as $\boldsymbol{\mathcal{T}}_{1(:,:,k)} = \boldsymbol{\mathcal{A}}_{(:,:,k)} \boldsymbol{\mathcal{B}}_{(:,:,k)}$, $\forall k = 1, \ldots, K$. To express this slice-wise multiplication we can diagonalize $\boldsymbol{\mathcal{B}}$ to obtain

$$\boldsymbol{\mathcal{T}}_1 = \boldsymbol{\mathcal{A}} \bullet_{2,3}^{1,4} \boldsymbol{\mathcal{D}}_B \in \mathbb{C}^{M \times J \times K}, \tag{1}$$

where $\boldsymbol{\mathcal{D}}_B \in \mathbb{R}^{N \times J \times K \times K}$ and $\boldsymbol{\mathcal{D}}_{B(n,j,:,:)} = \text{diag}\left(\boldsymbol{\mathcal{B}}_{(n,j,:)}\right)$ with $n = 1, \ldots N$ and $j = 1, \ldots J$. Further combinations are also possible that lead to the same result, for instance, $\boldsymbol{\mathcal{T}}_2 = \boldsymbol{\mathcal{D}}_B \bullet_{1,4}^{2,3} \boldsymbol{\mathcal{A}} \in \mathbb{C}^{J \times K \times M}$ or $\boldsymbol{\mathcal{T}}_3 = \boldsymbol{\mathcal{D}}_A \bullet_{2,4}^{1,3} \boldsymbol{\mathcal{B}} \in \mathbb{C}^{M \times K \times J}$ with $\boldsymbol{\mathcal{D}}_{A(m,n,k,k)} = \boldsymbol{\mathcal{A}}_{(m,n,k)}$ as diagonal elements (non-zero elements of $\boldsymbol{\mathcal{D}}_A$). Note that the tensors $\boldsymbol{\mathcal{T}}_1$, $\boldsymbol{\mathcal{T}}_2$, and $\boldsymbol{\mathcal{T}}_3$ computed in this way have permuted dimensions. However, the permuted ordering of the dimensions is not relevant.

An explicit expression of the diagonalized tensor can be obtained by expressing its generalized unfolding in terms of a Kharti-Rao product as illustrated in Table I.

TABLE I: Link between diagonalized tensor structures and their generalized unfoldings.

non-zero elements	generalized unfoldings
$\boldsymbol{\mathcal{D}}_{(m,m)} = \boldsymbol{a}_{(m)}$	$\boldsymbol{D} = \boldsymbol{I}_M \diamond \boldsymbol{a}^{\mathrm{T}}$
$\boldsymbol{\mathcal{D}}_{(m,n,n)} = \boldsymbol{A}_{(m,n)}$	$[\boldsymbol{\mathcal{D}}]_{([1,3],[2])} = \boldsymbol{I}_N \diamond \boldsymbol{A}$
$\boldsymbol{\mathcal{D}}_{(m,m,n,n)} = \boldsymbol{A}_{(m,n)}$	$[\boldsymbol{\mathcal{D}}]_{([1,3],[2,4])} = \boldsymbol{I}_M \diamond \text{vec}(\boldsymbol{A})^T$
$\boldsymbol{\mathcal{D}}_{(m,n,k,k)} = \boldsymbol{A}_{(m,n,k)}$	$[\boldsymbol{\mathcal{D}}]_{([1,2,4],[3])} = \boldsymbol{I}_K \diamond [\boldsymbol{\mathcal{A}}]_{([1,2],[3])}$

Based on these examples we can conclude that the element-wise or slice-wise multiplication between two arrays (vectors/matrices/tensors) of the same order can be written in terms of a contraction if the corresponding mode vectors are transformed into a diagonal matrix (by adding an additional array dimension). The diagonalization can be performed using the Khatri-Rao product as shown in Table I.

III. PARAFAC2 via Generalized Tensor Contractions

According to [1] the PARAFAC2 decomposition of a tensor $\boldsymbol{\mathcal{X}} \in \mathbb{C}^{I \times J \times K}$ is defined in the following slice-wise fashion.

$$\boldsymbol{X}_k = \boldsymbol{A} \cdot \text{diag}\left(\boldsymbol{C}_{(k,:)}\right) \cdot \boldsymbol{B}_k^{\mathrm{T}}, \quad \forall k = 1, \ldots, K \tag{2}$$

It can be interpreted as a coupled matrix decomposition of K matrices $\boldsymbol{X}_k \in \mathbb{R}^{I \times J}$, where the matrix $\boldsymbol{A} \in \mathbb{R}^{I \times R}$ is the coupled mode. The rows of the matrix $\boldsymbol{C} \in \mathbb{R}^{K \times R}$ contain the weights corresponding to the R underlying components. The second mode is not coupled and therefore each $\boldsymbol{X}_k$ matrix has a different loading factor $\boldsymbol{B}_k^{\mathrm{T}} \in \mathbb{R}^{R \times J}$. In general, the PARAFAC2 decomposition is not unique. However, in [1] it has been shown that it is essentially unique under mild conditions if $\boldsymbol{B}_k^{\mathrm{T}} \cdot \boldsymbol{B}_k = \boldsymbol{F}^{\mathrm{T}} \cdot \boldsymbol{F}$ such that $\boldsymbol{B}_k^{\mathrm{T}} = \boldsymbol{F}^{\mathrm{T}} \cdot \boldsymbol{V}_k$, $\boldsymbol{V}_k \cdot \boldsymbol{V}_k^{\mathrm{T}} = \boldsymbol{I}_R$, and K is large enough. This is known as the Harshman constraint [1].

The slice-wise description of PARAFAC2 is visualized in Fig. 1. It can be regarded as a slice-wise multiplications of two tensor, which can be expressed in terms of the generalized contraction as proposed in Section II. The resulting tensor structure enables a simultaneous view of all dimensions, leading to an efficient computation of the PARAFAC2 decomposition.

First, let us define $\boldsymbol{\mathcal{X}}_{(:,:,k)} = \boldsymbol{A} \cdot \boldsymbol{\mathcal{C}}_{(:,:,k)} \cdot \boldsymbol{\mathcal{B}}_{(:,:,k)}$ with $\boldsymbol{\mathcal{X}}_{(:,:,k)} = \boldsymbol{X}_k$ ($\boldsymbol{\mathcal{X}} \in \mathbb{R}^{I \times J \times K}$), $\boldsymbol{\mathcal{C}} = \boldsymbol{\mathcal{I}}_{3,R} \times_3 \boldsymbol{C} \in \mathbb{R}^{R \times R \times K}$, and $\boldsymbol{\mathcal{B}}_{(:,:,k)} = \boldsymbol{B}_k^{\mathrm{T}}$ ($\boldsymbol{\mathcal{B}} \in \mathbb{R}^{R \times J \times K}$). Using the Harshman constraint and defining $\boldsymbol{\mathcal{V}}_{(:,:,k)} = \boldsymbol{V}_k$ ($\boldsymbol{\mathcal{V}} \in \mathbb{R}^{R \times J \times K}$) , we get

$$\boldsymbol{\mathcal{B}} = \boldsymbol{\mathcal{V}} \times_1 \boldsymbol{F}^{\mathrm{T}}, \quad \boldsymbol{\mathcal{B}} \bullet_{2,3}^{2,3} \boldsymbol{\mathcal{B}} = K \boldsymbol{F}^{\mathrm{T}} \boldsymbol{F} \in \mathbb{R}^{R \times R}, \text{and}$$
$$\boldsymbol{\mathcal{V}} \bullet_{2,3}^{2,3} \boldsymbol{\mathcal{V}} = K \boldsymbol{I}_R \in \mathbb{R}^{R \times R}. \tag{3}$$

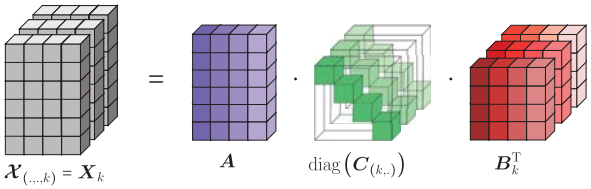

Fig. 1: Slice-wise description of PARAFAC2.

Using the generalized tensor contraction, we can rewrite the PARAFAC2 decomposition of the tensor $\mathcal{X}$ as

$$\mathcal{X} = (\mathcal{D}_C \times_1 \boldsymbol{A}) \bullet_{2,4}^{1,3} (\mathcal{V} \times_1 \boldsymbol{F}^{\mathrm{T}}) \in \mathbb{R}^{I \times K \times J}, \tag{4}$$

where $[\mathcal{D}_{\mathcal{C}}]_{([1,2,4],[3])} = \boldsymbol{I}_K \diamond [\mathcal{C}]_{([1,2],[3])}$. Note the permuted dimensions of the tensor $\mathcal{X}$ as compared to (2) due to the definition of the contraction operator. The diagonalized tensor $\mathcal{D}_C \in \mathbb{C}^{R \times R \times K \times K}$ has also a BTD (Block Term Decomposition) [14] structure given by

$$\mathcal{D}_C = \sum_{k=1}^{K} (\mathcal{I}_{4,1} \otimes \mathcal{I}_{3,R}) \times_3 \left((\boldsymbol{e}_k \otimes \boldsymbol{C}_{(k,:)})\right) \times_4 \boldsymbol{e}_k,$$

where the pinning vector $\boldsymbol{e}_k \in \mathbb{R}^K$ is the k-th column of an identity matrix of size $K \times K$. Replacing the sum by a block diagonal tensor, we get

$$\mathcal{D}_C = (\mathcal{I}_{4,K} \otimes \mathcal{I}_{3,R}) \times_1 \left(\mathbf{1}_K^{\mathrm{T}} \otimes \boldsymbol{I}_R\right) \times_2 \left(\mathbf{1}_K^{\mathrm{T}} \otimes \boldsymbol{I}_R\right) \times_3 \left((\boldsymbol{I}_K \otimes \mathbf{1}_R^{\mathrm{T}}) \diamond \mathrm{vec}\left(\boldsymbol{C}^{\mathrm{T}}\right)^{\mathrm{T}}\right).$$

Using equation (4), the property $[\mathcal{X}]_{([1,2],3)} = [\mathcal{D}_C \times_1 \boldsymbol{A}]_{([1,3],[2,4])} [\mathcal{B}]_{([1,3],2)}$, and the structure of the tensor $\mathcal{D}_C$, we get

$$[\mathcal{X}]_{([1,2],3)} = \left(\left((\boldsymbol{I}_K \otimes \mathbf{1}_R^{\mathrm{T}}) \diamond \mathrm{vec}\left(\boldsymbol{C}^{\mathrm{T}}\right)^{\mathrm{T}}\right) \otimes \boldsymbol{A}\left(\mathbf{1}_K^{\mathrm{T}} \otimes \boldsymbol{I}_R\right)\right) \cdot [\mathcal{I}_{4,K} \otimes \mathcal{I}_{3,R}]_{([1,3],[2,4])} \cdot \left(\boldsymbol{I}_K \otimes \mathbf{1}_K^{\mathrm{T}} \otimes \boldsymbol{I}_R\right)^{\mathrm{T}} \cdot \left(\boldsymbol{I}_K \otimes \boldsymbol{F}^{\mathrm{T}}\right) \cdot [\mathcal{V}]_{([1,3],2)}.$$

Note that the matrix $[\mathcal{I}_{4,K} \otimes \mathcal{I}_{3,R}]_{([1,3],[2,4])} \cdot \left(\boldsymbol{I}_K \otimes \mathbf{1}_K^{\mathrm{T}} \otimes \boldsymbol{I}_R\right)^{\mathrm{T}} = \boldsymbol{I}_{RK} \diamond \boldsymbol{I}_{RK}$ is a selection matrix that converts the Kronecker product into a Khatri-Rao product. Using this property and the unfolding of a CP decomposition, we have

$$[\mathcal{X}]_{([1,2],3)} = \left(\left((\boldsymbol{I}_K \otimes \mathbf{1}_R^{\mathrm{T}}) \diamond \mathrm{vec}\left(\boldsymbol{C}^{\mathrm{T}}\right)^{\mathrm{T}}\right) \diamond \boldsymbol{A}\left(\mathbf{1}_K^{\mathrm{T}} \otimes \boldsymbol{I}_R\right)\right) \cdot \left(\boldsymbol{I}_K \otimes \boldsymbol{F}^{\mathrm{T}}\right) \cdot [\mathcal{V}]_{([1,3],2)}.$$

Hence, the tensor $\mathcal{X} \in \mathbb{R}^{I \times K \times T}$ can be expressed in the following CP format:

$$\mathcal{X} = \mathcal{I}_{3,RK} \times_1 \bar{\boldsymbol{A}} \times_2 \bar{\boldsymbol{C}} \times_3 \bar{\boldsymbol{B}}, \tag{5}$$

where $\bar{\boldsymbol{A}} = \boldsymbol{A}\left(\mathbf{1}_K^{\mathrm{T}} \otimes \boldsymbol{I}_R\right) \in \mathbb{R}^{I \times RK}$, $\bar{\boldsymbol{C}} = (\boldsymbol{I}_K \otimes \mathbf{1}_R^{\mathrm{T}}) \diamond \mathrm{vec}\left(\boldsymbol{C}^{\mathrm{T}}\right)^{\mathrm{T}} \in \mathbb{R}^{K \times RK}$, and $\bar{\boldsymbol{B}} = [\mathcal{B}]_{(2,[1,3])} = \left(\left(\boldsymbol{I}_K \otimes \boldsymbol{F}^{\mathrm{T}}\right) \cdot [\mathcal{V}]_{([1,3],2)}\right)^{\mathrm{T}} \in \mathbb{R}^{J \times RK}$. Equation (5) shows that PARAFAC2 is equivalent to a constrained CP decomposition [15], which is degenerate in all three modes. This decomposition is also referred to in the literature as the constrained factor (CONFAC) decomposition [16], which is a special case of the PARALIND model [17], [18]. As shown in [18], CONFAC/PARALIND models enjoy uniqueness (or partial uniqueness) under mild conditions, depending on their linear dependence structure.

IV. Computation of the PARAFAC2

Assume that the matrices $\boldsymbol{A}$, $\boldsymbol{C}$, and $\boldsymbol{F}$ are known. From the PARAFAC2 tensor model in equation (5) we can estimate $\mathcal{V}$, in the LS (Least Squares) sense, as

$$[\mathcal{V}]_{([1,3],2)} = \left((\bar{\boldsymbol{C}} \diamond \bar{\boldsymbol{A}}) \cdot (\boldsymbol{I}_K \otimes \boldsymbol{F}^{\mathrm{T}})\right)^{+} \cdot [\mathcal{X}]_{([1,2],3)}.$$

This LS estimate does not take into account the orthogonality constraints of the tensor $\mathcal{V}$ defined before equation (3). Therefore, it is not applicable in this case. However, it is applicable if the Harshman constraint is not considered. Alternatively, we can estimate the unfolding of the tensor $\mathcal{B}$

$$\bar{\boldsymbol{B}} = [\mathcal{B}]_{(2,[1,3])} = [\mathcal{X}]_{(3,[1,2])} \cdot \left((\bar{\boldsymbol{C}} \diamond \bar{\boldsymbol{A}})^{\mathrm{T}}\right)^{+},$$

followed by an estimate of $\mathcal{V}$ via a solution of the OPP (Orthogonal Procrustes Problem) [19] using

$$[\mathcal{B}]_{(1,[2,3])} = \boldsymbol{F}^{\mathrm{T}} \cdot [\mathcal{V}]_{(1,[2,3])}.$$

Simulation results, though, have shown that this approach is less accurate in a noisy case than solving directly the OPP. Therefore, we propose to estimate the tensor $\mathcal{V}$ via OPP using directly equation (4). Using the orthogonality properties of $\mathcal{V}$ it can be shown that

$$\tilde{\mathcal{X}} = \mathcal{X} \bullet_{2,3}^{2,3} \mathcal{D}_{\mathcal{V}} = \mathcal{I}_{3,R} \times_1 \boldsymbol{A} \times_2 \boldsymbol{F}^{\mathrm{T}} \times_3 \boldsymbol{C} \tag{6}$$

has a CP structure, where $\mathcal{D}_{\mathcal{V}} \in \mathbb{R}^{R \times J \times K \times K}$, $[\mathcal{D}_{\mathcal{V}}]_{([1,2,4],[3])} = \boldsymbol{I}_K \diamond [\mathcal{V}]_{([1,2],[3])}$ according to Table I. As shown in [19] the best orthogonal estimate of $\mathcal{V}$ is provided from $\mathcal{Q} = \mathcal{X} \bullet_{1,3}^{1,3} \mathcal{D}_{\tilde{\mathcal{X}}}$, where $\left[\mathcal{D}_{\tilde{\mathcal{X}}}\right]_{([1,2,4],[3])} = \boldsymbol{I}_K \diamond \left[\tilde{\mathcal{X}}\right]_{([1,2],[3])}$. Then, $\mathcal{V}$ is obtained from

$$\mathcal{V}_{(:,:,k)} = \left(\mathcal{Q}_{(:,:,k)}^{\mathrm{T}} \cdot \mathcal{Q}_{(:,:,k)}\right)^{-\frac{1}{2}} \mathcal{Q}_{(:,:,k)}^{\mathrm{T}}, \quad \forall k = 1, \ldots, K.$$

Next, an estimate of the matrix $\boldsymbol{A}$ is obtained from the unfolding $[\mathcal{X}]_{(1,[2,3])}$.

$$\boldsymbol{A} = [\mathcal{X}]_{(1,[2,3])} \cdot \left((\mathbf{1}_K^{\mathrm{T}} \otimes \boldsymbol{I}_R) \cdot (\bar{\boldsymbol{B}} \diamond \bar{\boldsymbol{C}})^{\mathrm{T}}\right)^{+}$$

Utilizing the orthogonality of $\mathcal{V}$, an estimate of $\boldsymbol{F}$ follows from equation (6).

$$\boldsymbol{F} = \left[\tilde{\mathcal{X}}\right]_{(2,[1,3])} \cdot \left((\boldsymbol{C} \diamond \boldsymbol{A})^{\mathrm{T}}\right)^{+}$$

The tensor $\mathcal{B}$ is then computed by $\mathcal{B} = \mathcal{V} \times_1 \boldsymbol{F}^{\mathrm{T}}$.

Finally, an LS estimate of $\boldsymbol{C}$ is estimated based on the unfolding $[\mathcal{X}]_{(2,[3,1])}$ of equation (5).

$$[\mathcal{X}]_{(2,[3,1])} = \left((\boldsymbol{I}_K \otimes \mathbf{1}_R^{\mathrm{T}}) \diamond \mathrm{vec}\left(\boldsymbol{C}^{\mathrm{T}}\right)^{\mathrm{T}}\right) \cdot (\bar{\boldsymbol{A}} \diamond \bar{\boldsymbol{B}})^{\mathrm{T}}$$

This unfolding represents also an unfolding of a 4-D CP tensor. Hence

$$\mathcal{X} = \mathcal{I}_{3,RK} \times_1 \bar{\boldsymbol{A}} \times_2 (\boldsymbol{I}_K \otimes \mathbf{1}_R^{\mathrm{T}}) \times_3 \bar{\boldsymbol{B}} \times_4 \mathrm{vec}\left(\boldsymbol{C}^{\mathrm{T}}\right)^{\mathrm{T}},$$

which leads to

$$\mathrm{vec}\left(\boldsymbol{C}^{\mathrm{T}}\right) = \left(\bar{\boldsymbol{B}} \diamond \left(\boldsymbol{I}_K \otimes \mathbf{1}_R^{\mathrm{T}}\right) \diamond \bar{\boldsymbol{A}}\right)^{+} \cdot \mathrm{vec}\left(\boldsymbol{\mathcal{X}}\right).$$

Based on these estimates, we propose a single loop ALS based direct fitting algorithm for the computation of the PARAFAC2 decomposition. The proposed algorithm is initialized with a initial values of $\boldsymbol{A}$ chosen based on the SVD of $[\boldsymbol{\mathcal{X}}]_{(1,[2,3])}$, $\boldsymbol{C}$ is chosen randomly, and $\boldsymbol{F}$ as an identity matrix. The algorithm is stopped if it exceeds the predefined maximum number of iterations or reaches a predefined minimum of the cost function $e_{\mathrm{r}} = \|\hat{\boldsymbol{\mathcal{X}}} - \boldsymbol{\mathcal{X}}\|_{\mathrm{H}}^2 / \|\boldsymbol{\mathcal{X}}\|_{\mathrm{H}}^2$, where $\hat{\boldsymbol{\mathcal{X}}}$ is the reconstructed tensor after the decomposition.

V. Simulation Results

In this section we provide comparisons between the direct fitting approaches for the the computation of the PARAFAC2 decomposition based on simulation results. The single loop ALS based algorithm proposed in Section IV is denoted as "P2-ALS". The ALS algorithm [11] that consists of two ALS loops, an inner and an outer loop is denoted by "P2-ALS, two loops". Finally, "P2-SMD" denotes an algorithm that is similar to the "P2-ALS, two loops", but the second loop fits the CP decomposition using a SMD (Simultaneous Matrix Diagonalization) similar to [20]. A distinctive difference between the "P2-SMD" and the SECSI framework [20] is that the "P2-SMD" solves only one SMD instead of all possible SMDs.

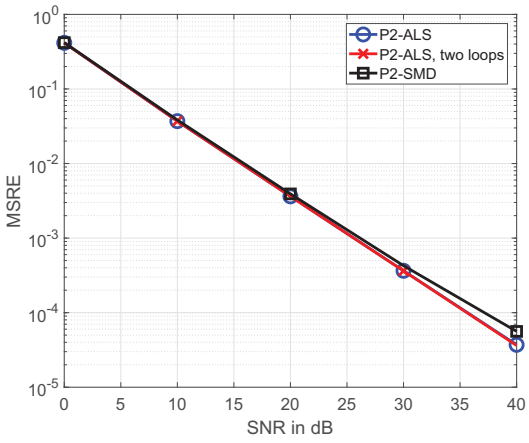

Fig. 2: MSRE as a function of the SNR= $0, \ldots, 40$ dB for real-valued tensors with dimensions $8 \times 10 \times 12$, and 3 components. The results are averaged over 2000 realizations. All algorithms are initialized with same initial matrices.

For the purpose of comparing the algorithms, "P2-ALS", "P2-ALS, two loops", and "P2-SMD", we generate synthetic data according to equation (2). The elements of the matrices, $\boldsymbol{A}$, $\boldsymbol{C}$, and $\boldsymbol{F}$ are drawn randomly from a zero mean Gaussian distribution with variance one. The tensor $\boldsymbol{\mathcal{V}}$ is also randomly generated, such that each slice has orthogonal rows. Next, we add zero mean additive white Gaussian noise with variance σ_{N}^2 to the generated signal tensor, $\boldsymbol{\mathcal{X}} = \boldsymbol{\mathcal{X}}_0 + \boldsymbol{\mathcal{N}}$. The tensors $\boldsymbol{\mathcal{X}}_0$, $\boldsymbol{\mathcal{N}}$, and $\boldsymbol{\mathcal{X}}$ represent the noiseless signal tensor, the noise tensor, and the noisy tensor, respectively. Therefore, the instantaneous SNR (Signal to Noise Ratio) equals $10\log_{10}\left(\|\boldsymbol{\mathcal{X}}_0\|_{\mathrm{H}}^2 / \|\boldsymbol{\mathcal{N}}\|_{\mathrm{H}}^2\right)$.

As an accuracy measure, we use the SRE $= \|\hat{\boldsymbol{\mathcal{X}}} - \boldsymbol{\mathcal{X}}_0\|_{\mathrm{H}}^2 / \|\boldsymbol{\mathcal{X}}_0\|_{\mathrm{H}}^2$ (Squared Reconstruction Error) where $\hat{\boldsymbol{\mathcal{X}}}$ is the reconstructed tensor using the estimated factor matrices. All algorithms are initialized with the same initial factor matrices. The maximum number of iterations for all algorithms is set to 2000 iterations as discussed at the end of the last section. The algorithms are stopped if they reach the maximum number of iterations or if they reach the minimum error of the cost function equal to 10^{-7}.

In Fig. 2 we illustrate the MSRE (Mean Squared Reconstruction Error) as a function of the SNR. Here, the MSRE is the SRE averaged over 2000 realizations. A real-valued tensor with dimensions $8 \times 10 \times 12$ and $R = 3$ components is generated. The accuracy of the three algorithms is very similar with the exception of the "P2-SMD" for high SNRs, where its accuracy is slightly lower as compared to "P2-ALS" and "P2-ALS, two loops". The maximum number of iterations for the inner loop, of the algorithm "P2-ALS, two loops" and "P2-SMD" was set to 5 and 50 iterations, respectively. The average number of iterations for SNR= 20 dB is equal to 122, 137, and 157 for "P2-ALS", "P2-ALS, two loops" and "P2-SMD", respectively.

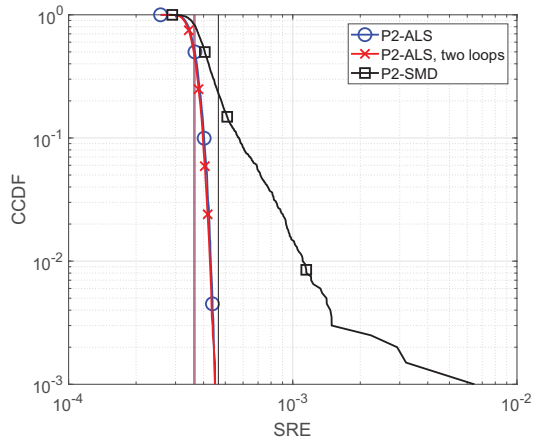

Fig. 3: CCDF of the SRE for real-valued tensors with dimensions $8 \times 10 \times 12$, 3 components, and SNR= 30 dB. The matrix $\boldsymbol{C}$ has correlated columns with a correlation coefficient of 0.8. All algorithms are initialized with same initial matrices. The vertical lines represent the mean value.

In Figs. 3 and 4 we depict the CCDF (Complementary Cumulative Distribution Function) of the SRE and the number of iterations, respectively, for a real-valued tensor with correlation. The decomposed tensor consists of $R = 3$ components and it has dimensions $8 \times 10 \times 12$. As previously mentioned, its factor matrices $\boldsymbol{A}$, $\boldsymbol{C}$, and $\boldsymbol{F}$ are drawn from a zero mean Gaussian distribution with variance one. However, the matrix $\boldsymbol{C}$ has correlated columns with a correlation coefficient of 0.8. The SREs presented in Fig. 3 correspond to an SNR of 30 dB and 2000 realizations. The vertical lines represent the mean values for each curve. In Fig. 4 we depict the CCDF

of the number of iterations for these 2000 realizations. The initializations are identical for all algorithms. For the "P2-ALS, two loops" and "P2-SMD" we only count the number of iterations of the outer loop, not of the inner one. The vertical lines represent the averaged number of iterations equal to 278, 466, and 95 for "P2-ALS", "P2-ALS, two loops", and "P2-SMD", respectively. The algorithms "P2-ALS" and "P2-ALS, two loops" have the same accuracy in terms of SRE, but the "P2-ALS" requires fewer iterations. The algorithm "P2-SMD" has different convergence properties with a lower accuracy. According to Fig. 4, the "P2-SMD" requires the lowest number of iterations for the outer loop, but the inner loop was set to a maximum number of 50 iterations.

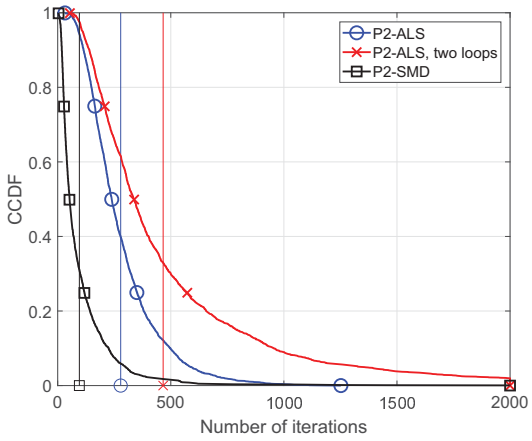

Fig. 4: CCDF of the number of iterations for real-valued tensors with dimensions $8 \times 10 \times 12$, 3 components, and SNR= 30 dB. The matrix $\boldsymbol{C}$ has correlated columns with a correlation coefficient of 0.8. All algorithms are initialized with same initial matrices. The vertical lines represent the averaged number of iterations equal to 278, 466, and 95 for "P2-ALS", "P2-ALS, two loops", and "P2-SMD", respectively.

VI. Conclusion

This paper provides an application of a novel tensor representation derived using the generalized contraction between two tensors for a slice-wise (matrix) multiplication. This representation leads to new tensor models that reveal the complete tensor structure. In this paper we reveal the tensor structure of the PARAFAC2 tensor decomposition. PARAFAC2 is equivalent to a constrained, degenerate CP model. Unitizing this model, we derive a direct fitting, single loop ALS algorithm ("P2-ALS"). This "P2-ALS" algorithm has the same accuracy, but requires fewer iterations as compared to the state-of-art direct fitting ALS algorithm with two loops.

Acknowledgments

The authors gratefully acknowledge the partial support of the Internal Excellence Initiative of the TU Ilmenau, the Carl-Zeiss-Foundation, and the PROBRAL CAPES/DAAD project no. 88887.144009/2017-00.

References

[1] R. A. Harshman, "PARAFAC2: Mathematical and Technical Notes," vol. 22, pp. 30–47, 1972.

[2] M. Weis, D. Jannek, F. Roemer, T. Guenther, M. Haardt, and P. Husar, "Multi-dimensional PARAFAC2 component analysis of multi-channel EEG data including temporal tracking," in *Proc. 32-th Inter. Conf. of the IEEE Engineering in Medicine and Biology Society (EMBC)*, 2010.

[3] M. Weis, D. Jannek, T. Guenther, P. Husar, F. Roemer, and M. Haardt, "Temporally Resolved Multi-Way Component Analysis of Dynamic Sources in Event-Related EEG Data using PARAFAC2," in *Proc. 18-th European Signal Processing Conf. (EUSIPCO 2010)*, pp. 696–700.

[4] Y. Cheng, M. Haardt, T. Götz, and J. Haueisen, "Using PARAFAC2 for multi-way component analysis of somatosensory evoked magnetic fiels and somatosensory evoked potentials," in *10th IEEE Sensor Array and Multichannel Signal Processing Workshop (SAM))*, 2018.

[5] Y. Cheng, K. Naskovska, M. Haardt, T. Götz, and J. Haueisen, "A new coupled PARAFAC2 decomposition for joint processing of somatosensory evoked magnetic fields and somatosensory evoked electrical potentials," in *Proc. of 52nd Asilomar Conf. on Signals, Systems, and Computers*, 2018.

[6] A. Cichocki, "Era of big data processing: A new approach via tensor networks and tensor decompositions," *arXiv:1403.2048 [cs.ET]*, 2014.

[7] K. Naskovska, M. Haardt, and A. L. F. de Almeida, "Generalized tensor contraction with application to Khatri-Rao Coded MIMO OFDM systems," in *Proc. IEEE 7th Int. Workshop on Computational Advances in Multi-Sensor Adaptive Processing (CAMSAP)*, 2017, pp. 286–290.

[8] ——, "Generalized Tensor Contractions for an Improved Receiver design in MIMO-OFDM Systems," in *Proc. IEEE Int. Conf. on Acoustics, Speech, and Signal Processing (ICASSP)*, 2018, pp. 3186–3190.

[9] H. A. Kiers, "An alternating least squares algorithm for PARAFAC2 and three-way DEDICOM," *Computational Statistics & Data Analysis*, vol. 16, no. 1, pp. 103–118, 1993.

[10] R. A. Harshman and M. E. Lundy, "Uniqueness proof for a family of models sharing features of tuckers three-mode factor analysis and parafac/candecomp," *Psychometrika*, vol. 61, pp. 133–154, 1996.

[11] H. A. L. Kiers, J. M. F. Ten Berge, and R. Bro, "PARAFAC2 - Part I. A direct fitting algorithm for the PARAFAC2 model," *Journal of Chemometrics*, vol. 13, no. 3-4, pp. 275–294, 1999.

[12] X. Luciani and L. Albera, "Semi-algebraic canonical decomposition of multi-way arrays and joint eigenvalue decomposition," in *Proc. IEEE Int. Conf. on Acoustics, Speech, and Signal Processing (ICASSP)*, 2011, pp. 4104–4107.

[13] F. Roemer, C. Schroeter, and M. Haardt, "A semi-algebraic framework for approximate CP decompositions via joint matrix diagonalization and generalized unfoldings," *in Proc. of the 46th Asilomar Conf. on Signals, Systems, and Computers*, pp. 2023–2027, November 2012.

[14] L. D. Lathauwer, "Decompositions of a higher-order tensor in block terms-part II: Definitions and uniqueness," *SIAM J. Matrix Anal. Appl. (SIMAX)*, vol. 30, pp. 1033–1066, 2008.

[15] G. Favier and A. L. F. de Almeida, "Overview of constrained parafac models," *EURASIP Journal on Advances in Signal Processing*, no. 1, 2014.

[16] A. L. F. de Almeida, G. Favier, and J. C. M. Mota, "A Constrained Factor Decomposition With Application to MIMO Antenna Systems," *IEEE Trans. on Signal Processing*, vol. 56, no. 6, pp. 2429–2442, 2008.

[17] R. Bro, R. A. Harshman, N. D. Sidiropoulos, and M. E. Lundy, "Modeling multiway data with linearly dependent loadings," *Journal of Chemometrics*, vol. 23, no. 7, pp. 324–340, 2009.

[18] A. Stegeman and A. L. F. de Almeida, "Uniqueness conditions for constrained three-way factor decompositions wit linearly dependent loadings," *SIAM Journal on Matrix Analysis and Applications*, vol. 31, no. 3, pp. 1469–1490, 2010.

[19] P. H. Schönemann, "A generalized solution of the orthogonal procrustes problem," vol. 31, no. 1, pp. 1–10, 1999.

[20] F. Roemer and M. Haardt, "A SEmi-algebraic framework for approximate CP decomposition via SImultaneous matrix diagonalization (SECSI)," *Signal Processing*, vol. 93, pp. 2722–2738, September 2013.