A CLOSED-FORM SOLUTION FOR PARALLEL FACTOR (PARAFAC) ANALYSIS

Florian Roemer and Martin Haardt

Ilmenau University of Technology, Communications Research Laboratory
P.O. Box 100565, D-98684 Ilmenau, Germany, http://tu-ilmenau.de/crl
florian.roemer@ieee.org, haardt@ieee.org

Abstract **— Parallel Factor Analysis (PARAFAC) is a branch of multi-way signal processing that has received increased attention recently. This is due to the large class of applications as well as the milestone identifiability results demonstrating the superiority to matrix (two-way) analysis approaches. A significant amount of research was dedicated to iterative methods to estimate the factors from noisy data. In many situations these require many iterations and are not guaranteed to converge to the global optimum. Therefore, suboptimal closed-form solutions were proposed as initializations.**

In this contribution we derive a closed-form solution to completely replace the iterative approach by transforming PARAFAC into several joint diagonalization problems. Thereby, we obtain several estimates for each of the factors and present a new "best matching" scheme to select the best estimate for each factor.

In contrast to the techniques known from the literature, our closed-form solution can efficiently exploit symmetric as well as Hermitian symmetric models and solve the underdetermined case, if there are at least two modes that are non-degenerate and full rank. This closed-form solution achieves approximately the same performance as previously proposed iterative solutions and even outperforms them in critical scenarios.

Index Terms— Multidimensional signal processing, Parameter estimation, Array signal processing, Direction of arrival estimation

1. INTRODUCTION

In this contribution we focus on the three-way PARAFAC model, which is also known as trilinear model or canonical decomposition. It has originated as data analysis tool in psychometrics (the underlying idea first appeared in [2]) and was later adopted in various fields, e.g., spectroscopy, pattern recognition, explorative data analysis, but also several fields of signal processing [12, 8]. The simplicity and versatility of the model as well as the milestone results on its identifiability [7] render it an attractive approach to solve a large variety of tasks in different fields. The wide range of applications includes blind equalization [12], blind source separation [14], RADAR [8], psychometrics [2] any many more.

For this reason, a significant amount of research was dedicated to findind fast and yet robust methods to compute the factors from noisy data. The estimates are often obtained via iterative techniques such as alternating least squares (ALS) [1] that may require many iterations and are not guaranteed to converge to the global optimum [9]. Therefore, approximate closed-form solutions were proposed as initializations [3, 11].

In this contribution we introduce a new closed-form solution[1] to find the factors of the PARAFAC model that completely replaces the iterative approach. We demonstrate how PARAFAC can be transformed into the well-studied task of jointly diagonalizing several sets of matrices, for which numerous efficient solutions exist (e.g., [6]). Due to the structure of the problem we find several different estimates for each of the factors. In a second step we present a new "best matching" scheme to select the best estimate for each factor, which enhances the performance even further.

[1]The use of the term "closed-form" in the literature is conflicting. We consider simultaneous matrix diagonalization to be closed-form and therefore term our approach closed-form solution.

2. TENSOR AND MATRIX NOTATION

In order to facilitate the distinction between scalars, matrices, and tensors, the following representations are used: Scalars are denoted as italic letters $(a, b, \ldots, A, B, \ldots, \alpha, \beta, \ldots)$, vectors as lower-case bold-face letters $(\boldsymbol{a}, \boldsymbol{b}, \ldots)$, matrices as bold-face capitals $(\boldsymbol{A}, \boldsymbol{B}, \ldots)$, and tensors are written as bold-face calligraphic letters $(\boldsymbol{\mathcal{A}}, \boldsymbol{\mathcal{B}}, \ldots)$.We use the superscripts ${}^{\mathrm{T}}, {}^{\mathrm{H}}, {}^{-1}, {}^{+}$ for transposition, Hermitian transposition, matrix inversion, and the Moore-Penrose pseudo inverse of matrices and * for complex conjugation, respectively. The transpose (Hermitian transpose) of an inverse may be written as ${}^{-\mathrm{T}}$ (${}^{-\mathrm{H}}$). Moreover the Kronecker product between two matrices $\boldsymbol{A}$ and $\boldsymbol{B}$ is denoted by $\boldsymbol{A} \otimes \boldsymbol{B}$ and the Khatri-Rao (column-wise Kronecker) product between $\boldsymbol{A}$ and $\boldsymbol{B}$ by $\boldsymbol{A} \diamond \boldsymbol{B}$. Also, the notation $\operatorname{diag}\{\boldsymbol{A}(k,:)\}$ represents a diagonal matrix that is constructed from the elements in the k-th row of the matrix $\boldsymbol{A}$.

The tensor operations we use are consistent with [5]: **The higher-order norm** of a tensor $\boldsymbol{\mathcal{A}}$ is symbolized by $||\boldsymbol{\mathcal{A}}||_{\mathrm{H}}$ and defined as the square-root of the sum of the squared magnitude of all elements in $\boldsymbol{\mathcal{A}}$. The n-space of a tensor is defined as the space spanned by the n-mode vectors, which are the vectors obtained by varying the n-th index within its range $(1, 2, \ldots, I_n)$ and keeping all the other indices fixed. Moreover, the notation $[\boldsymbol{\mathcal{A}}]_{i_n=k}$ represents the tensor obtained by keeping the n-th index fixed to the value k and varying all other indices in their corresponding ranges. A **matrix unfolding** of the tensor $\boldsymbol{\mathcal{A}}$ along the n-th mode is symbolized by $[\boldsymbol{\mathcal{A}}]_{(n)} \in \mathbb{C}^{I_n \times I_{n+1} \cdot \ldots \cdot I_N \cdot I_1 \cdot \ldots \cdot I_{n-1}}$ and contains all the n-mode vectors of the tensor $\boldsymbol{\mathcal{A}}$. The order of the columns is chosen in accordance with [5].

The n-mode product: The product of a tensor $\boldsymbol{\mathcal{A}} \in \mathbb{C}^{I_1 \times I_2 \times \ldots \times I_N}$ and a matrix $\boldsymbol{U} \in \mathbb{C}^{J_n \times I_n}$ along the n-th mode is denoted as $\boldsymbol{\mathcal{A}} \times_n \boldsymbol{U} \in \mathbb{C}^{I_1 \times I_2 \ldots \times J_n \ldots \times I_N}$. It is obtained by multiplying all n-mode vectors of $\boldsymbol{\mathcal{A}}$ from the left-hand side by the matrix $\boldsymbol{U}$.

To simplify the notation, we additionally define an identity tensor $\boldsymbol{\mathcal{I}}_d = \sum_{n=1}^{d} \boldsymbol{e}_{n,d} \circ \boldsymbol{e}_{n,d} \circ \boldsymbol{e}_{n,d} \in \mathbb{R}^{d \times d \times d}$, where $\boldsymbol{e}_{n,N}$ represents the n-th column of a $N \times N$ identity matrix (also termed the n-th pinning vector of size N).

3. DATA MODEL AND PROBLEM STATEMENT

The three-way PARAFAC model for a third order tensor $\mathcal{X}_0 \in \mathbb{C}^{M_1 \times M_2 \times M_3}$ can be expressed in the following fashion

$$\mathcal{X}_0 = \sum_{n=1}^{d} \boldsymbol{a}_n \circ \boldsymbol{b}_n \circ \boldsymbol{c}_n, \quad \text{where} \tag{1}$$

$\boldsymbol{a}_n \in \mathbb{C}^{M_1}$, $\boldsymbol{b}_n \in \mathbb{C}^{M_2}$, and $\boldsymbol{c}_n \in \mathbb{C}^{M_3}$. In other words, we decompose the tensor into a sum of d rank-1 tensors, where d is the rank of the tensor.[2] Under mild conditions, this decomposition is unique up to a scaling of each of the vectors $\boldsymbol{a}_n$, $\boldsymbol{b}_n$, and $\boldsymbol{c}_n$ and a permutation of the terms in the sum, see [7, 12] for details.

Let us rewrite equation (1) by introducing the matrices $\boldsymbol{A} \in \mathbb{C}^{M_1 \times d}$, $\boldsymbol{B} \in \mathbb{C}^{M_2 \times d}$, and $\boldsymbol{C} \in \mathbb{C}^{M_3 \times d}$ such that the n-th column of $\boldsymbol{A}$ is equal to $\boldsymbol{a}_n$, the n-th column of $\boldsymbol{B}$ is $\boldsymbol{b}_n$, and the n-th column of $\boldsymbol{C}$ is $\boldsymbol{c}_n$, where $n = 1, 2, \ldots, d$. We can then write

$$[\mathcal{X}_0]_{(3)} = \boldsymbol{C} \cdot (\boldsymbol{A} \diamond \boldsymbol{B})^{\mathrm{T}} \quad \text{or alternatively} \tag{2}$$

$$[\mathcal{X}_0]_{i_3=k} = \boldsymbol{A} \cdot \operatorname{diag}\{\boldsymbol{C}(k,:)\} \cdot \boldsymbol{B}^{\mathrm{T}} \tag{3}$$

The introduction of the identity tensor (cf. Section 2) facilitates an alternative notation of the PARAFAC model given by

$$\mathcal{X}_0 = \mathcal{I}_d \times_1 \boldsymbol{A} \times_2 \boldsymbol{B} \times_3 \boldsymbol{C}. \tag{4}$$

In practice, the data is usually contaminated by additive noise. We therefore extend (1) by incorporating a noise component in the following fashion: $\mathcal{X} = \mathcal{X}_0 + \mathcal{N}$, where $\mathcal{N}$ is the noise tensor which has the same size as $\mathcal{X}_0$. In the simulations we assume that $\mathcal{N}$ contains mutually independent zero mean circularly symmetric complex Gaussian random variables with variance equal to σ^2.

Consequently, the problem that we will solve can be formulated in the following manner: Given a noisy tensor $\mathcal{X}$ and the model order d find a PARAFAC model $\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{C}$, such that $||\mathcal{X}_0 - (\mathcal{I}_d \times_1 \boldsymbol{A} \times_2 \boldsymbol{B} \times_3 \boldsymbol{C})||_{\mathrm{H}}$ is minimized.

4. CLOSED-FORM SOLUTION

4.1. Transformation into joint diagonalization problem

The closed-form solution is based on the higher-order SVD (HOSVD) decomposition of $\mathcal{X}$ which is given by

$$\mathcal{X} = \mathcal{S} \times_1 \boldsymbol{U}_1 \times_2 \boldsymbol{U}_2 \times_3 \boldsymbol{U}_3, \tag{5}$$

where $\mathcal{S} \in \mathbb{C}^{M_1 \times M_2 \times M_3}$, $\boldsymbol{U}_r \in \mathbb{C}^{M_r \times M_r}$, $r = 1, 2, 3$ and which can easily be obtained from the singular value decomposition of the matrix unfoldings of $\mathcal{X}$ [5]. The HOSVD can be viewed as a Tucker3 model [13], which has a long history in tensor analysis.

We will first consider the "non-degenerate" case, where $d \leq \min\{M_1, M_2, M_3\}$. Also assume that the factors $\boldsymbol{A}$, $\boldsymbol{B}$, and $\boldsymbol{C}$ have full column-rank d. The degenerate and rank-deficient cases are discussed in Section 4.4. In the non-degenerate case, a low-rank approximation of $\mathcal{X}$ is given by

$$\mathcal{X} \approx \mathcal{S}^{[\mathrm{s}]} \times_1 \boldsymbol{U}_1^{[\mathrm{s}]} \times_2 \boldsymbol{U}_2^{[\mathrm{s}]} \times_3 \boldsymbol{U}_3^{[\mathrm{s}]}, \tag{6}$$

where $\mathcal{S}^{[\mathrm{s}]} \in \mathbb{C}^{d \times d \times d}$, $\boldsymbol{U}_r^{[\mathrm{s}]} \in \mathbb{C}^{M_r \times d}$, $r = 1, 2, 3$. Note that (6) holds exactly in the absence of noise and if d is the true rank of the tensor $\mathcal{X}$. For the following derivations we assume that this is true and hence write equalities. In the presence of noise, all the following relations still hold approximately.

[2] The n-rank of a tensor is defined as the rank of the space spanned by the n-mode vectors. In contrast to this, the rank of a tensor is equal to r if the tensor can be decomposed into a sum of r, but not less than r, rank-1 tensors. An N-th order tensor is rank-1 if and only if it can be written as the outer product of N non-zero vectors.

Consider the 1-mode unfolding of $\mathcal{X}$. It can be expressed in terms of the PARAFAC model (4) and in terms of the HOSVD (6). Therefore

$$[\mathcal{X}]_{(1)} = \boldsymbol{U}_1^{[\mathrm{s}]} \cdot \left(\left[\mathcal{S}^{[\mathrm{s}]}\right]_{(1)} \cdot \left[\boldsymbol{U}_2^{[\mathrm{s}]} \otimes \boldsymbol{U}_3^{[\mathrm{s}]}\right]^{\mathrm{T}} \right) \tag{7}$$

$$= \boldsymbol{A} \cdot \left([\mathcal{I}_d]_{(1)} \cdot [\boldsymbol{B} \otimes \boldsymbol{C}]^{\mathrm{T}} \right). \tag{8}$$

Comparing (7) and (8) it is easy to see that $\boldsymbol{A}$ and $\boldsymbol{U}_1^{[\mathrm{s}]}$ span the same column space. Thus, there is a non-singular transform matrix $\boldsymbol{T}_1 \in \mathbb{C}^{d \times d}$, such that $\boldsymbol{A} = \boldsymbol{U}_1^{[\mathrm{s}]} \cdot \boldsymbol{T}_1$. A similar analysis for the two- and three-mode unfoldings of $\mathcal{X}$ shows that $\exists\, \boldsymbol{T}_2, \boldsymbol{T}_3 \in \mathbb{C}^{d \times d}$, such that $\boldsymbol{B} = \boldsymbol{U}_2^{[\mathrm{s}]} \cdot \boldsymbol{T}_2$ and $\boldsymbol{C} = \boldsymbol{U}_3^{[\mathrm{s}]} \cdot \boldsymbol{T}_3$.

Inserting these relations into equation (6) and comparing with the model from equation (4) yields

$$\mathcal{S}^{[\mathrm{s}]} \times_1 \boldsymbol{T}_1^{-1} \times_2 \boldsymbol{T}_2^{-1} \times_3 \boldsymbol{T}_3^{-1} = \mathcal{I}_d. \tag{9}$$

This identity shows that we are searching for matrices that diagonalize the core tensor by transforming it into the identity tensor. Next, we show how this problem can be reduced to a joint diagonalization problem of several matrices.

Replacing $\boldsymbol{A}$ and $\boldsymbol{B}$ in (4) and comparing the result with (6) we find that

$$\left(\mathcal{S}^{[\mathrm{s}]} \times_3 \boldsymbol{U}_3^{[\mathrm{s}]}\right) \times_1 \boldsymbol{T}_1^{-1} \times_2 \boldsymbol{T}_2^{-1} = \mathcal{I}_d \times_3 \boldsymbol{C} \tag{10}$$

Consider the k-th three-mode slice (i.e., the third index is fixed, the others vary) of the tensor equation (10). The right-hand side of this equation is a diagonal matrix constructed from the k-th row of $\boldsymbol{C}$. The slicing operation can be accomplished by computing the three-mode product between (10) and the transpose of the k-th pinning vector $\boldsymbol{e}_{k,M_3}$ of size M_3. Then we have

$$\left(\left(\mathcal{S}^{[\mathrm{s}]} \times_3 \boldsymbol{U}_3^{[\mathrm{s}]}\right) \times_1 \boldsymbol{T}_1^{-1} \times_2 \boldsymbol{T}_2^{-1} \right) \times_3 \boldsymbol{e}_{k,M_3}^{\mathrm{T}} = (\mathcal{I}_d \times_3 \boldsymbol{C}) \times_3 \boldsymbol{e}_{k,M_3}^{\mathrm{T}}$$

$$\underbrace{\left(\mathcal{S}^{[\mathrm{s}]} \times_3 \boldsymbol{U}_3^{[\mathrm{s}]}\right) \times_3 \boldsymbol{e}_{k,M_3}^{\mathrm{T}}}_{\boldsymbol{S}_{3,k}} \times_1 \boldsymbol{T}_1^{-1} \times_2 \boldsymbol{T}_2^{-1} = \operatorname{diag}\{\boldsymbol{C}(k,:)\}$$

$$\boldsymbol{T}_1^{-1} \cdot \boldsymbol{S}_{3,k} \cdot \boldsymbol{T}_2^{-\mathrm{T}} = \operatorname{diag}\{\boldsymbol{C}(k,:)\} \tag{11}$$

using some easily checked identities of n-mode products. We have defined the matrix $\boldsymbol{S}_{3,k}$ which represents the k-th three-mode slice of the tensor $\mathcal{S}^{[\mathrm{s}]} \times_3 \boldsymbol{U}_3^{[\mathrm{s}]}$. Obviously the tensor equation has been transformed into a matrix equation (more precisely, a tensor of size $d \times d \times 1$). We can proceed in the same manner for $k = 1, 2, \ldots, M_3$. Combining all the equations we have

$$\begin{aligned} \boldsymbol{S}_{3,1} &= \boldsymbol{T}_1 \cdot \operatorname{diag}\{\boldsymbol{C}(1,:)\} \cdot \boldsymbol{T}_2^{\mathrm{T}} \\ \boldsymbol{S}_{3,2} &= \boldsymbol{T}_1 \cdot \operatorname{diag}\{\boldsymbol{C}(2,:)\} \cdot \boldsymbol{T}_2^{\mathrm{T}} \\ &\vdots \\ \boldsymbol{S}_{3,M_3} &= \boldsymbol{T}_1 \cdot \operatorname{diag}\{\boldsymbol{C}(M_3,:)\} \cdot \boldsymbol{T}_2^{\mathrm{T}}. \end{aligned} \tag{12}$$

The system of equation (12) represents an asymmetric joint diagonalization problem, since the transform matrices $\boldsymbol{T}_1$ and $\boldsymbol{T}_2$ can in general be different. However, (12) can be transformed into a symmetric joint diagonalization problem by multiplying all the slices $\boldsymbol{S}_{3,k}$ by the inverse of one particular[3] slice $\boldsymbol{S}_{3,p}$ from either the left-hand side (lhs) or the right-hand side (rhs), where p is an arbitrary number between 1 and M_3.

$$\begin{aligned} \boldsymbol{S}_{3,k}^{\mathrm{rhs}} &= \boldsymbol{S}_{3,k} \cdot \boldsymbol{S}_{3,p}^{-1} \\ &= \boldsymbol{T}_1 \cdot \operatorname{diag}\{\boldsymbol{C}(k,:)\} \cdot \boldsymbol{T}_2^{\mathrm{T}} \cdot \boldsymbol{T}_2^{-\mathrm{T}} \cdot \operatorname{diag}\{\boldsymbol{C}(p,:)\}^{-1} \cdot \boldsymbol{T}_1^{-1} \\ &= \boldsymbol{T}_1 \cdot \underbrace{\operatorname{diag}\{\boldsymbol{C}(k,:)\} \cdot \operatorname{diag}\{\boldsymbol{C}(p,:)\}^{-1}}_{\boldsymbol{C}_k^{\mathrm{D},p}} \cdot \boldsymbol{T}_1^{-1} \end{aligned} \tag{13}$$

[3] The choice of p is still arbitrary and therefore we can use it to optimize the performance even further. Since we have to form the inverse of $\boldsymbol{S}_{3,p}$ we can choose a slice that is well conditioned, i.e., $p = \arg\min_k \operatorname{cond}\{\boldsymbol{S}_{3,k}\}$. Here, $\operatorname{cond}\{\boldsymbol{A}\}$ is a function that indicates the conditioning of the matrix $\boldsymbol{A}$, i.e., its value should be small for well conditioned matrices and large otherwise. Note that this process is similar to selecting pivots when solving linear sets of equations.

$$\begin{aligned}\boldsymbol{S}_{3,k}^{\mathrm{lhs}} &= \left(\boldsymbol{S}_{3,p}^{-1} \cdot \boldsymbol{S}_{3,k}\right)^{\mathrm{T}} = \boldsymbol{S}_{3,k}^{\mathrm{T}} \cdot \boldsymbol{S}_{3,p}^{-\mathrm{T}} \\ &= \boldsymbol{T}_2 \cdot \operatorname{diag}\left\{\boldsymbol{C}(k,:)\right\} \cdot \boldsymbol{T}_1^{\mathrm{T}} \cdot \boldsymbol{T}_1^{-\mathrm{T}} \cdot \operatorname{diag}\left\{\boldsymbol{C}(p,:)\right\}^{-1} \cdot \boldsymbol{T}_2^{-1} \\ &= \boldsymbol{T}_2 \cdot \boldsymbol{C}_k^{\mathrm{D},p} \cdot \boldsymbol{T}_2^{-1}. \end{aligned} \tag{14}$$

Note that the additional transpose operator in the definition of $\boldsymbol{S}_{3,k}^{\mathrm{lhs}}$ is introduced due to symmetry reasons in the resulting diagonalization problems. The matrix $\boldsymbol{C}_k^{\mathrm{D},p}$ is a diagonal matrix of size $d \times d$, the n-th element on its diagonal equals $c_{k,n}/c_{p,n}$. The set of matrices $\boldsymbol{C}_k^{\mathrm{D},p}$ for $k = 1, 2, \ldots, M_3$ therefore describes $\boldsymbol{C}$ up to an arbitrary scaling of each of the columns. Since this ambiguity is inherent in the PARAFAC model, this set of matrices contains all the information about $\boldsymbol{C}$ we can extract. We now have transformed the original problem onto two joint diagonalization problems

$$\begin{array}{rclcrcl} \boldsymbol{S}_{3,1}^{\mathrm{rhs}} &=& \boldsymbol{T}_1 \cdot \boldsymbol{C}_1^{\mathrm{D},p} \cdot \boldsymbol{T}_1^{-1} && \boldsymbol{S}_{3,1}^{\mathrm{lhs}} &=& \boldsymbol{T}_2 \cdot \boldsymbol{C}_1^{\mathrm{D},p} \cdot \boldsymbol{T}_2^{-1} \\ \boldsymbol{S}_{3,2}^{\mathrm{rhs}} &=& \boldsymbol{T}_1 \cdot \boldsymbol{C}_2^{\mathrm{D},p} \cdot \boldsymbol{T}_1^{-1} && \boldsymbol{S}_{3,2}^{\mathrm{lhs}} &=& \boldsymbol{T}_2 \cdot \boldsymbol{C}_2^{\mathrm{D},p} \cdot \boldsymbol{T}_2^{-1} \\ &\vdots& && &\vdots& \\ \boldsymbol{S}_{3,M_3}^{\mathrm{rhs}} &=& \boldsymbol{T}_1 \cdot \boldsymbol{C}_{M_3}^{\mathrm{D},p} \cdot \boldsymbol{T}_1^{-1} && \boldsymbol{S}_{3,M_3}^{\mathrm{lhs}} &=& \boldsymbol{T}_2 \cdot \boldsymbol{C}_{M_3}^{\mathrm{D},p} \cdot \boldsymbol{T}_2^{-1} \end{array}$$

Many efficient solutions for finding these factors exist, e.g., [6]. From the joint diagonalization of the matrices $\boldsymbol{S}_{3,k}^{\mathrm{rhs}}$, $k = 1, 2, \ldots, M_3$ we obtain an estimate for $\boldsymbol{C}$, which we will term $\hat{\boldsymbol{C}}_{\mathrm{I}}$ and another estimate for $\boldsymbol{C}$ from the diagonalization of $\boldsymbol{S}_{3,k}^{\mathrm{lhs}}$ which is denoted as $\hat{\boldsymbol{C}}_{\mathrm{II}}$. Additionally, the matrices that diagonalize the sets yield estimates for $\boldsymbol{T}_1$ and $\boldsymbol{T}_2$ which we can use to compute estimates for $\boldsymbol{A}$ and $\boldsymbol{B}$, i.e., $\hat{\boldsymbol{A}}_{\mathrm{IV}} = \boldsymbol{U}_1^{[\mathrm{s}]} \cdot \boldsymbol{T}_1$ and $\hat{\boldsymbol{B}}_{\mathrm{IV}} = \boldsymbol{U}_2^{[\mathrm{s}]} \cdot \boldsymbol{T}_2$.

Due to the symmetry of the problem we can generate similar diagonalization problems for the second mode

$$\begin{array}{rclcrcl} \boldsymbol{S}_{2,1}^{\mathrm{rhs}} &=& \boldsymbol{T}_1 \cdot \boldsymbol{B}_1^{\mathrm{D},p} \cdot \boldsymbol{T}_1^{-1} && \boldsymbol{S}_{2,1}^{\mathrm{lhs}} &=& \boldsymbol{T}_3 \cdot \boldsymbol{B}_1^{\mathrm{D},p} \cdot \boldsymbol{T}_3^{-1} \\ \boldsymbol{S}_{2,2}^{\mathrm{rhs}} &=& \boldsymbol{T}_1 \cdot \boldsymbol{B}_2^{\mathrm{D},p} \cdot \boldsymbol{T}_1^{-1} && \boldsymbol{S}_{2,2}^{\mathrm{lhs}} &=& \boldsymbol{T}_3 \cdot \boldsymbol{B}_2^{\mathrm{D},p} \cdot \boldsymbol{T}_3^{-1} \\ &\vdots& && &\vdots& \\ \boldsymbol{S}_{2,M_2}^{\mathrm{rhs}} &=& \boldsymbol{T}_1 \cdot \boldsymbol{B}_{M_2}^{\mathrm{D},p} \cdot \boldsymbol{T}_1^{-1} && \boldsymbol{S}_{2,M_2}^{\mathrm{lhs}} &=& \boldsymbol{T}_3 \cdot \boldsymbol{B}_{M_2}^{\mathrm{D},p} \cdot \boldsymbol{T}_3^{-1} \end{array}$$

leading to the estimates $\hat{\boldsymbol{B}}_{\mathrm{I}}$ and $\hat{\boldsymbol{B}}_{\mathrm{II}}$ from the diagonalized matrices and $\hat{\boldsymbol{A}}_{\mathrm{III}} = \boldsymbol{U}_1^{[\mathrm{s}]} \cdot \boldsymbol{T}_1$ and $\hat{\boldsymbol{C}}_{\mathrm{IV}} = \boldsymbol{U}_3^{[\mathrm{s}]} \cdot \boldsymbol{T}_3$. Here, $\boldsymbol{B}_k^{\mathrm{D},p} = \operatorname{diag}\left\{\boldsymbol{B}(k,:)\right\} \cdot \operatorname{diag}\left\{\boldsymbol{B}(p,:)\right\}^{-1}$, $\boldsymbol{S}_{2,k}^{\mathrm{rhs}} = \boldsymbol{S}_{2,k} \cdot \boldsymbol{S}_{2,p}^{-1}$, $\boldsymbol{S}_{2,k}^{\mathrm{lhs}} = \left(\boldsymbol{S}_{2,p}^{-1} \cdot \boldsymbol{S}_{2,k}\right)^{\mathrm{T}}$, and $\boldsymbol{S}_{2,k} = \left[\left(\boldsymbol{\mathcal{S}}^{[\mathrm{s}]} \times_2 \boldsymbol{U}_2^{[\mathrm{s}]}\right) \times_2 \boldsymbol{e}_{k,M_2}^{\mathrm{T}}\right]_{(1)}$. Note that the unfolding operator in the definition of $\boldsymbol{S}_{2,k}$ merely serves to transform the $d \times 1 \times d$ tensor into a $d \times d$ matrix, as the `squeeze` operator would do in Matlab.

Finally, for the first mode we obtain

$$\begin{array}{rclcrcl} \boldsymbol{S}_{1,1}^{\mathrm{rhs}} &=& \boldsymbol{T}_2 \cdot \boldsymbol{A}_1^{\mathrm{D},p} \cdot \boldsymbol{T}_2^{-1} && \boldsymbol{S}_{1,1}^{\mathrm{lhs}} &=& \boldsymbol{T}_3 \cdot \boldsymbol{A}_1^{\mathrm{D},p} \cdot \boldsymbol{T}_3^{-1} \\ \boldsymbol{S}_{1,2}^{\mathrm{rhs}} &=& \boldsymbol{T}_2 \cdot \boldsymbol{A}_2^{\mathrm{D},p} \cdot \boldsymbol{T}_2^{-1} && \boldsymbol{S}_{1,2}^{\mathrm{lhs}} &=& \boldsymbol{T}_3 \cdot \boldsymbol{A}_2^{\mathrm{D},p} \cdot \boldsymbol{T}_3^{-1} \\ &\vdots& && &\vdots& \\ \boldsymbol{S}_{1,M_1}^{\mathrm{rhs}} &=& \boldsymbol{T}_2 \cdot \boldsymbol{A}_{M_2}^{\mathrm{D},p} \cdot \boldsymbol{T}_2^{-1} && \boldsymbol{S}_{1,M_1}^{\mathrm{lhs}} &=& \boldsymbol{T}_3 \cdot \boldsymbol{A}_{M_2}^{\mathrm{D},p} \cdot \boldsymbol{T}_3^{-1} \end{array}$$

leading to the estimates $\hat{\boldsymbol{A}}_{\mathrm{I}}$ and $\hat{\boldsymbol{A}}_{\mathrm{II}}$ from the diagonalized matrices and $\hat{\boldsymbol{B}}_{\mathrm{III}} = \boldsymbol{U}_2^{[\mathrm{s}]} \cdot \boldsymbol{T}_2$ and $\hat{\boldsymbol{C}}_{\mathrm{III}} = \boldsymbol{U}_3^{[\mathrm{s}]} \cdot \boldsymbol{T}_3$. As before, $\boldsymbol{A}_k^{\mathrm{D},p} = \operatorname{diag}\left\{\boldsymbol{A}(k,:)\right\} \cdot \operatorname{diag}\left\{\boldsymbol{A}(p,:)\right\}^{-1}$, $\boldsymbol{S}_{1,k}^{\mathrm{rhs}} = \boldsymbol{S}_{1,k} \cdot \boldsymbol{S}_{1,p}^{-1}$, $\boldsymbol{S}_{1,k}^{\mathrm{lhs}} = \left(\boldsymbol{S}_{1,p}^{-1} \cdot \boldsymbol{S}_{1,k}\right)^{\mathrm{T}}$, and $\boldsymbol{S}_{1,k} = \left[\left(\boldsymbol{\mathcal{S}}^{[\mathrm{s}]} \times_1 \boldsymbol{U}_1^{[\mathrm{s}]}\right) \times_1 \boldsymbol{e}_{k,M_1}^{\mathrm{T}}\right]_{(2)}$.

4.2. Best matching

As we have demonstrated in the previous subsection, the structure of the problem enables us to compute four estimate for each of the factors $\boldsymbol{A}$, $\boldsymbol{B}$, and $\boldsymbol{C}$. In order to select the final estimates we propose the following scheme

$$\hat{\boldsymbol{A}} = \hat{\boldsymbol{A}}_{e_1}, \quad \hat{\boldsymbol{B}} = \hat{\boldsymbol{B}}_{e_1}, \quad \hat{\boldsymbol{C}} = \hat{\boldsymbol{C}}_{e_1}, \qquad \text{where}$$
$$e_1, e_2, e_3 = \underset{i_1, i_2, i_3 \in \{\mathrm{I},\mathrm{II},\mathrm{III},\mathrm{IV}\}}{\arg\min} \left\| \boldsymbol{\mathcal{X}} - \left(\boldsymbol{\mathcal{I}}_d \times_1 \hat{\boldsymbol{A}}_{i_1} \times_2 \hat{\boldsymbol{B}}_{i_2} \times_3 \hat{\boldsymbol{C}}_{i_3}\right) \right\|_{\mathrm{H}}$$

In other words, we reconstruct the tensor with all possible combinations of estimates and select the triple that best matches the *noisy* tensor $\boldsymbol{\mathcal{X}}$. If all combinations are tested exhaustively, there are $4^3 = 64$ combinations to be tested.

4.3. Combined joint diagonalization problems

The structure of the problem can be exploited even further to generate more estimates of the factors $\boldsymbol{A}$, $\boldsymbol{B}$, and $\boldsymbol{C}$. Each of the transform matrices $\boldsymbol{T}_1$, $\boldsymbol{T}_2$, and $\boldsymbol{T}_3$ appears in two of the six diagonalization problems we considered so far. We can therefore combine the corresponding sets of matrices to solve three bigger joint diagonalization problems. As a result we obtain three more estimates (V, VI, VII) for each of the factors (one from the combined estimate of the transform matrix, two from the diagonalized matrices). In total we therefore have seven estimates for $\boldsymbol{A}$, for $\boldsymbol{B}$, and for $\boldsymbol{C}$. For the best matching we can then test up to $7^3 = 343$ possible combinations. However, simulations have shown that the additional performance enhancement is very small. To save computational complexity it is therefore sufficient to compute the four estimates as described in Section 4.1.

4.4. Degenerate case

In the degenerate case, the number of components d satisfies $d > \min\{M_1, M_2, M_3\}$. We show that our closed-form solution can still be applied to the case where d exceeds the size of the array in *one of the* modes but is still less or equal than the size of the array in all other modes. Without loss of generality we assume that the "degenerate mode" is the first mode, i.e., $d > M_1$, $d \leq M_2, M_3$.

In this case the matrix $\boldsymbol{U}_1^{[\mathrm{s}]}$ is of size $M_1 \times M_1$ whereas $\boldsymbol{A}$ is of size $M_1 \times d$. Since $d > M_1$ we cannot recover $\boldsymbol{A}$ from $\boldsymbol{U}_1$ by multiplying with a transform matrix $\boldsymbol{T}_1$. However, since we assumed non-degeneracy in the remaining modes, $\boldsymbol{B}$ and $\boldsymbol{C}$, we can still use the transform matrices $\boldsymbol{T}_2$ and $\boldsymbol{T}_3$. Consequently, the "asymmetric" joint diagonalization problem (as in (12)) cannot be expressed for the third or the second mode, but still for the first mode, i.e., the diagonalization of the slices of $\boldsymbol{S}_{1,k} = \left[\left(\boldsymbol{\mathcal{S}}^{[\mathrm{s}]} \times_1 \boldsymbol{U}_1^{[\mathrm{s}]}\right) \times_1 \boldsymbol{e}_{k,M_1}^{\mathrm{T}}\right]_{(2)}$. We can also multiply by the inverse of the p-th slice to render the problem symmetric. This results in the two diagonalization problems for $\boldsymbol{S}_{1,k}^{\mathrm{rhs}}$ and $\boldsymbol{S}_{1,k}^{\mathrm{lhs}}$, $k = 1, 2, \ldots, M_1$. Therefore, in the degenerate case, we obtain two estimates for $\boldsymbol{A}$ (from the diagonalized matrices), one for $\boldsymbol{B}$, and one for $\boldsymbol{C}$ (from the corresponding transform matrices). For the best matching we only need to test two alternatives.

The case where one of the factors $\boldsymbol{A}$, $\boldsymbol{B}$, and $\boldsymbol{C}$ does not have full column rank can be treated in a similar fashion. In this case, the corresponding transform matrix $\boldsymbol{T}_n$ is rank-deficient and therefore none of the slices $\boldsymbol{S}_{n,p}$ is invertible for $p = 1, 2, \ldots, M_n$. If the factors in the other two modes are non-degenerate and full-rank we can still solve the joint diagonalization problem where the n-th mode appears on the diagonal. Of course, the identifiability of the PARAFAC model has to be fulfilled [12].

4.5. Symmetric case

In the symmetric case, two factors in the PARAFAC model are equal, e.g., $\boldsymbol{A} = \boldsymbol{B}$. In this case we can define the HOSVD of $\boldsymbol{\mathcal{X}}$ in such a way[4] that $\boldsymbol{U}_1^{[\mathrm{s}]} = \boldsymbol{U}_2^{[\mathrm{s}]}$ and consequently $\boldsymbol{T}_1 = \boldsymbol{T}_2$. Applying the closed-form solution as described above, it is easy to see that it will automatically be guaranteed that $\hat{\boldsymbol{A}}_e = \hat{\boldsymbol{B}}_e$ for $e \in \{\mathrm{I}, \mathrm{II}, \mathrm{III}, \mathrm{IV}, \mathrm{V}, \mathrm{VI}, \mathrm{VII}\}$, i.e., the estimates are automatically symmetric.

[4] There is an ambiguity in the HOSVD since every singular vector can be multiplied by a factor of -1 if the corresponding elements in the core tensor are appropriately scaled. Choosing the factors appropriately, we can render the HOSVD symmetric.

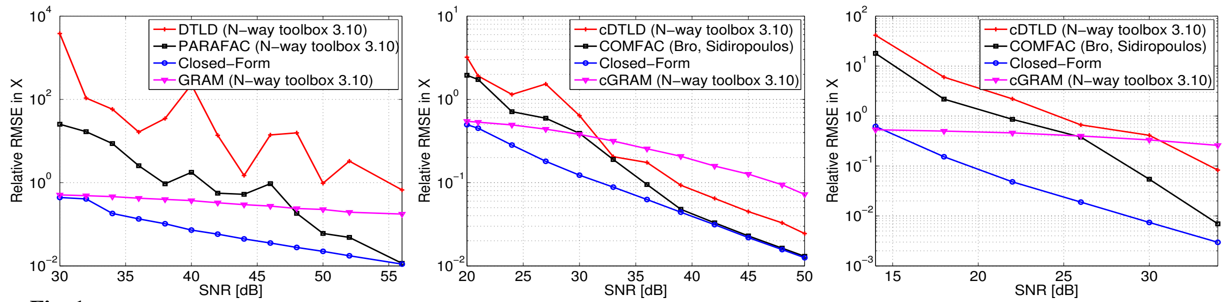

Fig. 1: Relative RMSE vs. SNR. Critical scenario. Left: real-valued, middle: complex-valued, Right: complex-valued, Hermitian symmetric.

The symmetry can be exploited to enhance the estimate further. Consider the asymmetric joint diagonalization problem in equation (12). We note that for $\boldsymbol{T}_1 = \boldsymbol{T}_2$ it has a special structure which is known in the literature as "congruence transform". Direct solutions of joint diagonalization problems of this form exist (e.g., the AC-DC algorithm [14]). We therefore do not need to multiply by the inverse of the pivot slice which is numerically advantageous, especially if the slices are badly conditioned.

Another frequently encountered form of symmetry in the PARAFAC model is the conjugate symmetry, where, e.g., $\boldsymbol{A} = \boldsymbol{B}^*$ [10]. In this case, the HOSVD of $\boldsymbol{\mathcal{X}}$ can be defined in such a way that $\boldsymbol{U}_1 = \boldsymbol{U}_2^*$ and consequently $\boldsymbol{T}_1 = \boldsymbol{T}_2^*$. As for the "transpose" symmetry, the estimates resulting from our proposed method are automatically symmetric, i.e., $\hat{\boldsymbol{A}}_e = \hat{\boldsymbol{B}}_e^*$ for $e \in \{\mathrm{I}, \mathrm{II}, \mathrm{III}, \mathrm{IV}, \mathrm{V}, \mathrm{VI}, \mathrm{VII}\}$. Also, (12) is in the form of a congruence transform and can be solved directly (e.g., via the AC-DC algorithm [14] in its "Hermitian version").

5. SIMULATION RESULTS

In this section we evaluate the performance of our proposed closed-form solution through numerical computer simulations. For all the simulations we set $M_1 = 4$, $M_2 = 8$, $M_3 = 7$ and $d = 3$. The first factor is fixed to

$$\boldsymbol{A} = \begin{bmatrix} 1.00 & 1.00 & 1.00 \\ 1.00 & 0.95 & 0.95 \\ 1.00 & 0.95 & 1.00 \\ 1.00 & 1.00 & 0.95 \end{bmatrix} \tag{15}$$

We observe that the columns of $\boldsymbol{A}$ are almost colinear ($\mathrm{cond}\{\boldsymbol{A}\} \approx 116$), i.e., we consider a critical scenario. The elements of $\boldsymbol{B}$ and $\boldsymbol{C}$ are independently drawn from a zero mean Gaussian distribution in the first simulation and from a zero mean circularly symmetric complex Gaussian distribution in the second and third simulations. In the third simulation $\boldsymbol{B}$ is set to $\boldsymbol{C}^*$ and $M_2 = M_3$ to 6 to create a Hermitian symmetric problem. We display a Monte-Carlo estimate of the relative root mean square reconstruction error defined as

$$\mathrm{rRMSE} = \sqrt{\mathrm{E}\left\{ \frac{\left|\left| \boldsymbol{\mathcal{I}}_d \times_1 \hat{\boldsymbol{A}} \times_2 \hat{\boldsymbol{B}} \times_3 \hat{\boldsymbol{C}} - \boldsymbol{\mathcal{X}}_0 \right|\right|_{\mathrm{H}}^2}{||\boldsymbol{\mathcal{X}}_0||_{\mathrm{H}}^2} \right\}}. \tag{16}$$

For comparison we also show the performance of an iterative implementation of PARAFAC, the DTLD and the GRAM algorithms in the real-valued case and for the second simulation their complex-valued counterparts, the COMFAC, the cDTLD and the cGRAM algorithms.[5]

From the simulation results depicted in Fig. 1 we can conclude that the proposed closed-form scheme outperforms DTLD and GRAM and at low SNRs even the iterative PARAFAC and COMFAC schemes (which required around 75 iterations on the average, in some cases even up to several thousands).

[5]For DTLD, GRAM, and PARAFAC, the N-way toolbox v3.10 from http://www.models.life.ku.dk/source/nwaytoolbox/index.asp is used, which includes the enhanced line search [9]. An implementation of COMFAC is taken from http://www.ece.umn.edu/users/nikos/public_html/3SPICE/code.html.

6. CONCLUSIONS

In this contribution, a closed-form solution to PARAFAC is introduced. It is demonstrated that the task of finding the factors can be reduced to the well-studied problem of a joint diagonalization of several matrices. Thereby we extend the ideas of [3, 4] by rendering the diagonalization problems symmetric. We also show that the structure of the problem allows to compute up to seven estimates for each of the factors $\boldsymbol{A}$, $\boldsymbol{B}$, and $\boldsymbol{C}$. To find the best combination, a selection scheme based on the noisy tensor is proposed.

Moreover, we discuss the degenerate case and show that the closed-form solution can still be applied if one of the factors does not have full column rank. For the symmetric and the Hermitian symmetric cases, it is demonstrated that the estimates of the closed-form solution are automatically symmetric or Hermitian symmetric, and an improved estimation scheme based on congruence transforms is proposed. In simulations, the performance of the closed-form solution is comparable to iterative PARAFAC solutions that include many state of the art improvements (e.g., the enhanced line search [9]). In critical scenarios, the closed-form solution has a superior estimation accuracy.

REFERENCES

[1] R. Bro, N. Sidiropoulos, and G. B. Giannakis, "A fast least squares algorithm for separating trilinear mixtures", in *Proc. Int. Workshop on Independent Component Analysis for Blind Signal Separation (ICA 99)*, pp. 289–294, Jan. 1999.

[2] R. B. Cattell, "Parallel proportional profiles and other principles for determining the choice of factors by rotation", *Psychometrika*, vol. 9, Dec. 1944.

[3] L. de Lathauwer, "Parallel Factor Analysis by Means of Simultaneous Matrix Decompositions", in *Proc. First IEEE Int. Workshop on Computational Advances in Multi-Sensor Adaptive Processing (CAMSAP 2005)*, pp. 125–128, Dec. 2005.

[4] L. de Lathauwer, "A link between the canonical decomposition in multilinear algebra and simultaneous matrix diagonalization", *SIAM J. Matrix Anal. Appl.*, vol. 28, no. 3, pp. 642–666, Dec. 2006.

[5] L. de Lathauwer, B. de Moor, and J. Vanderwalle, "A multilinear singular value decomposition", *SIAM J. Matrix Anal. Appl.*, vol. 21, no. 4, 2000.

[6] T. Fu and X. Gao, "Simultaneous diagonalization with similarity transformation for non-defective matrices", in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 2006)*, vol. IV, pp. 1137–1140, Toulouse, France, May 2006.

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

[8] N. Petrochilos and P. Comon, "Link between the Joint Diagonalisation of symmetrical cubes and PARAFAC: an application to Secondary Surveillance Radar", in *Proc. of IEEE SAM 2006*, Waltham (MA), 12-14 July 2006.

[9] M. Rajih, *Blind Identification of Underdetermined Mixtures Based on the Characteristic Function*, PhD thesis, Université de Nice à Sophia Antipolis, Dec. 2006.

[10] Y. Rong, S. A. Vorobyov, A. B. Gershman, and N. D. Sidiropoulos, "Blind Spatial Signature Estimation via Time-Varying User Power Loading and Parallel Factor Analysis", *IEEE Trans. Signal Processing*, vol. 53, pp. 1697 – 1710, May 2005.

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

[12] N. D. Sidiropoulos, G. B. Giannakis, and R. Bro, "Blind PARAFAC receivers for DS-CDMA Systems", *IEEE Transactions Sig. Process.*, vol. 48, Mar. 2000.

[13] L. R. Tucker, "Some mathematical notes on three-mode factor analysis", *Psychometrika*, vol. 31, no. 3, pp. 279–311, Sept. 1966.

[14] A. Yeredor, "Non-orthogonal Joint Diagonalization in the Least-Squares Sense With Application in Blind Source Separation", *IEEE Transactions on Signal Processing*, vol. 50, pp. 1545–1553, July 2002.