Enhanced Direct Fitting Algorithms for PARAFAC2 With Algebraic Ingredients

Yao Cheng, *Member, IEEE*, and Martin Haardt, *Fellow, IEEE*

***Abstract*—The PARAFAC2 decomposition has attracted growing research interest. To compute it, the direct fitting (DF) algorithm is usually employed. In this contribution, we propose to incorporate semi-algebraic approaches for the computation of PARAFAC in the DF algorithm and explore how it should be adapted to largely exploit the resulting diversity in the estimates of factor matrices to enhance the robustness and efficiency. Three versions have been tailored and recommended for different parameter settings according to, e.g., the number of components or tensor dimensions. Extensive numerical simulations have been conducted with both synthetic and measured biomedical as well as wine data to validate the improved performance of the proposed DF algorithms. In addition, it has been shown that applying the geometric search on top of the proposed DF algorithms further boosts their performance.**

***Index Terms*—PARAFAC2, direct fitting algorithm.**

I. Introduction

WIDELY used in a variety of scientific fields, the PARAFAC2 decomposition has become a crucial multilinear processing tool. It finds applications not only in the food industry for the processing of gas chromatography-mass spectrometry (GC-MS) data [1]–[5] but also in multi-way component analysis of biomedical signals, e.g., measured visual evoked potentials [6], somatosensory evoked magnetic fields (SEFs) and somatosensory evoked electrical potentials (SEPs) [7]–[9]. In addition to the aforementioned areas, PARAFAC2 plays a significant role in medical imaging [10] as well as classification of audio and images [11], [12].

The most commonly used approach to compute PARAFAC2 is the direct fitting (DF) algorithm [13]. Similar to many other alternating least squares (ALS) schemes, it is prone to a low convergence especially when the collinearity in data is strong [14]. This issue therefore motivates advanced efficient and robust computation algorithms for PARAFAC2. In the literature, line search approaches [15], [16] and a very recently proposed powerful alternative, geometric search (GS) [14], are employed to speed up the DF scheme. Only in [6] it has been pointed out that a semi-algebraic approach [17] with simultaneous matrix diagonalizations (SMDs) [18] efficiently computing the PARAFAC decomposition inherent in the DF algorithm can be incorporated to fit the PARAFAC2 model. However, it remains unclear how exactly to realize it, and the expected advantages have never been validated.

Indeed, adding algebraic ingredients to the DF algorithm is an intriguing idea but a non-trivial task, far more than some straightforward incorporation. It is then the aim of this letter to present the first efforts towards accomplishing it. We investigate the impact of the diverse PARAFAC factor estimates obtained via SMDs [18]–[20] on the DF algorithm and accordingly propose to adapt it to exploit this diversity while taking the complexity aspect into account. A special two-component case is discussed, where only the eigenvalue decompositions (EVDs) are needed instead of the SMDs. Moreover, the applicability of the GS [14] approach on the resulting enhanced DF algorithms is also addressed.

Throughout this letter, scalars are represented by italic letters, vectors by lower-case bold-faced letters, matrices by upper-case bold-faced letters, and tensors as bold-faced calligraphic letters. We use the superscript $^{\mathrm{T}}$ and $^{+}$ for transpose and the Moore-Penrose pseudo inverse of a matrix, respectively. The i-th row and the j-th column of a matrix $\boldsymbol{A} \in \mathbb{C}^{I \times J}$ is symbolized by $\boldsymbol{A}(i,:) \in \mathbb{C}^{J}$ and $\boldsymbol{A}(:,j) \in \mathbb{C}^{I}$, respectively, where $i = 1, \ldots, I$ and $j = 1, \ldots, J$. For an R-way tensor $\boldsymbol{\mathcal{A}} \in \mathbb{C}^{I_1 \times I_2 \times \ldots \times I_R}$, its r-mode vectors are obtained by varying the r-th index from 1 to I_r and keeping all other indices fixed. Aligning all r-mode vectors as the columns of a matrix yields the r-mode unfolding of $\boldsymbol{\mathcal{A}}$ which is denoted by $[\boldsymbol{\mathcal{A}}]_{(r)} \in \mathbb{C}^{I_r \times I_{r+1} \cdot \ldots \cdot I_R \cdot I_1 \cdot \ldots \cdot I_{r-1}}$. Here the reverse cyclical ordering of the columns [21] is used. The r-mode product between a tensor $\boldsymbol{\mathcal{A}}$ and a matrix $\boldsymbol{U}$ is written as $\boldsymbol{\mathcal{A}} \times_r \boldsymbol{U}$, computed by multiplying all r-mode vectors of $\boldsymbol{\mathcal{A}}$ with $\boldsymbol{U}$, i.e., $[\boldsymbol{\mathcal{A}} \times_r \boldsymbol{U}]_{(r)} = \boldsymbol{U} \cdot [\boldsymbol{\mathcal{A}}]_{(r)}$. For a three-dimensional tensor $\boldsymbol{\mathcal{B}} \in \mathbb{C}^{I_1 \times I_2 \times I_3}$, its k-th one-, two-, and three-mode slice is represented by $[\boldsymbol{\mathcal{B}}]_{(k,:,:)} \in \mathbb{C}^{I_2 \times I_3}$, $(k = 1, \ldots, I_1)$, $[\boldsymbol{\mathcal{B}}]_{(:,k,:)} \in \mathbb{C}^{I_1 \times I_3}$, $(k = 1, \ldots, I_2)$, and $[\boldsymbol{\mathcal{B}}]_{(:,:,k)} \in \mathbb{C}^{I_1 \times I_2}$, $(k = 1, \ldots, I_3)$, respectively. In addition, we denote the higher-order norm of a tensor $\boldsymbol{\mathcal{A}}$ by $\|\boldsymbol{\mathcal{A}}\|_{\mathrm{H}}$. It is defined as the square root of the sum of the squared magnitude of all elements in $\boldsymbol{\mathcal{A}}$. Moreover, the Khatri-Rao (column-wise Kronecker) product between two matrices is expressed as $\boldsymbol{A} \diamond \boldsymbol{B}$.

II. PARAFAC2 and Direct Fitting Algorithm

PARAFAC2 decomposes the tensor $\boldsymbol{\mathcal{X}} \in \mathbb{R}^{M_1 \times M_2 \times M_3}$ such that the k-th frontal slice of $\boldsymbol{\mathcal{X}}$ $(k = 1, \ldots, M_3)$ is written as[1]

$$[\boldsymbol{\mathcal{X}}]_{(:,:,k)} = \boldsymbol{A} \cdot \mathrm{diag}\left\{\boldsymbol{C}(k,:)\right\} \cdot \boldsymbol{B}_k^{\mathrm{T}} + [\boldsymbol{\mathcal{N}}]_{(:,:,k)} \in \mathbb{R}^{M_1 \times M_2},$$

Manuscript received November 28, 2018; revised January 28, 2019; accepted January 28, 2019. Date of publication February 4, 2019; date of current version February 21, 2019. This work was partially supported by the Carl-Zeiss-Foundation (http://carl-zeiss-stiftung.de/). The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Joao Paulo Paulo Papa. *(Corresponding author: Yao Cheng.)*

The authors are with the Communications Research Laboratory, Ilmenau University of Technology, Ilmenau D-98684, Germany (e-mail: y.cheng@tu-ilmenau.de).

Digital Object Identifier 10.1109/LSP.2019.2897244

[1]PARAFAC2 can be applied to a set of M_3 matrices of size $M_1 \times M_{2_k}$ $(k = 1, \ldots, M_3)$. For notation simplicity, we assume $M_{2_k} = M_2$ for $k = 1, \ldots, M_3$, while the proposed algorithms are also applicable to the general PARAFAC2 model.

where $\boldsymbol{A} \in \mathbb{R}^{M_1 \times d}$, $\boldsymbol{C} \in \mathbb{R}^{M_3 \times d}$, $\boldsymbol{B}_k \in \mathbb{R}^{M_2 \times d}$ for $k = 1, \ldots, M_3$, and d represents the model order. The residuals are collected in $\boldsymbol{\mathcal{N}} \in \mathbb{R}^{M_1 \times M_2 \times M_3}$. We use $\mathrm{diag}\{\cdot\}$ to denote the operation of constructing a diagonal matrix whose diagonal elements are the entries of the input vector.

Note that the Harshman constraint [22] given by $\boldsymbol{B}_k^{\mathrm{T}} \cdot \boldsymbol{B}_k = \boldsymbol{H} \in \mathbb{R}^{d \times d}$ $(k = 1, \ldots, M_3)$ can be reformulated [13] such that $\boldsymbol{B}_k^{\mathrm{T}} = \boldsymbol{F}^{\mathrm{T}} \cdot \boldsymbol{V}_k$, where $\boldsymbol{F} \in \mathbb{R}^{d \times d}$, and $\boldsymbol{V}_k \in \mathbb{R}^{d \times M_2}$ for $k = 1, \ldots, M_3$, satisfying $\boldsymbol{V}_k \cdot \boldsymbol{V}_k^{\mathrm{T}} = \boldsymbol{I}_d$. Now we introduce a tensor $\widetilde{\boldsymbol{\mathcal{X}}}_0 \in \mathbb{R}^{M_1 \times d \times M_3}$ where the k-th frontal slice of $\widetilde{\boldsymbol{\mathcal{X}}}_0$ is written as

$$\left[\widetilde{\boldsymbol{\mathcal{X}}}_0\right]_{(:,:,k)} = \boldsymbol{A} \cdot \mathrm{diag}\left\{\boldsymbol{C}(k,:)\right\} \cdot \boldsymbol{F}^{\mathrm{T}}, \tag{1}$$

indicating that $\boldsymbol{A}$, $\boldsymbol{C}$, and $\boldsymbol{F}$ are the factor matrices of the PARAFAC decomposition of $\widetilde{\boldsymbol{\mathcal{X}}}_0$. In a zero-residual case, the following equality holds

$$\left[\widetilde{\boldsymbol{\mathcal{X}}}_0\right]_{(:,:,k)} = [\boldsymbol{\mathcal{X}}]_{(:,:,k)} \cdot \boldsymbol{V}_k^{\mathrm{T}}. \tag{2}$$

These facts of PARAFAC2 have led to the DF algorithm [13] summarized as follows:

- **Step 1**: Initialize $\boldsymbol{A}$, $\boldsymbol{C}$, and $\boldsymbol{F}$.
- **Step 2**: Reconstruct $\widetilde{\boldsymbol{\mathcal{X}}}_0$ with $\boldsymbol{A}, \boldsymbol{C}$, and $\boldsymbol{F}$ according to (1), and update $\boldsymbol{V}_k$ $(k = 1, \ldots, M_3)$ with the generalized solution of the Orthogonal Procrustes Problem (OPP) [23] owing to (2).
- **Step 3**: Update $\widetilde{\boldsymbol{\mathcal{X}}}_0$ according to (2) with $\boldsymbol{V}_k$ $(k = 1, \ldots, M_3)$ computed in **Step 2**. Then compute its PARAFAC decomposition to update $\boldsymbol{A}$, $\boldsymbol{C}$, and $\boldsymbol{F}$.
- **Step 4**: Compute the reconstructed tensor $\hat{\boldsymbol{\mathcal{X}}}$ and then calculate the residual defined via $E_{\mathrm{R}} = (\|\hat{\boldsymbol{\mathcal{X}}} - \boldsymbol{\mathcal{X}}\|_{\mathrm{H}}^2)/\|\boldsymbol{\mathcal{X}}\|_{\mathrm{H}}^2$. Repeat **Step 2 – Step 4** until the change of the residual quantified by $\Delta E_{\mathrm{R}} = (E_{\mathrm{R}}^{\mathrm{old}} - E_{\mathrm{R}})/E_{\mathrm{R}}^{\mathrm{old}}$ is smaller than a predefined threshold, implying the convergence of the algorithm, where $E_{\mathrm{R}}^{\mathrm{old}}$ represents the residual in the previous iteration.

III. Enhanced Direct Fitting Algorithms

As proposed in [13], in each iteration of the DF algorithm, an ALS scheme is used to compute the PARAFAC decomposition of $\widetilde{\boldsymbol{\mathcal{X}}}_0$ so that $\boldsymbol{A}$, $\boldsymbol{C}$, and $\boldsymbol{F}$ are updated (cf. **Step 3**). This inspires the enhancements that will be introduced in the sequel. On the one hand, the proposed methods should still benefit from the long-standing and well-constructed structure of the DF algorithm. On the other hand, the advancements in the semi-algebraic schemes for tensor decompositions should be exploited for updating the factor matrices in PARAFAC2.

To achieve these goals, we take computing the PARAFAC decomposition of $\widetilde{\boldsymbol{\mathcal{X}}}_0$ via SMDs as the starting point. In this work, the SECSI framework [19] is considered as an example to construct the SMD target matrices, while other semi-algebraic approaches [20], [24] for the computation of PARAFAC can also be used to initiate such an algebraic enhancement of the DF algorithm.

Assume that the truncated higher-order SVD (HOSVD) of $\widetilde{\boldsymbol{\mathcal{X}}}_0$ is given by

$$\widetilde{\boldsymbol{\mathcal{X}}}_0 = \boldsymbol{\mathcal{S}}^{[\mathrm{s}]} \times_1 \boldsymbol{U}_1^{[\mathrm{s}]} \times_2 \boldsymbol{U}_2^{[\mathrm{s}]} \times_3 \boldsymbol{U}_3^{[\mathrm{s}]}, \tag{3}$$

where $\boldsymbol{\mathcal{S}}^{[\mathrm{s}]} \in \mathbb{R}^{d \times d \times d}$, $\boldsymbol{U}_1^{[\mathrm{s}]} \in \mathbb{R}^{M_1 \times d}$, $\boldsymbol{U}_2^{[\mathrm{s}]} \in \mathbb{R}^{d \times d}$, and $\boldsymbol{U}_3^{[\mathrm{s}]} \in \mathbb{R}^{M_3 \times d}$. Since the columns of $\boldsymbol{U}_1^{[\mathrm{s}]}$, $\boldsymbol{U}_2^{[\mathrm{s}]}$, and $\boldsymbol{U}_3^{[\mathrm{s}]}$ span the same subspace as $\boldsymbol{A}$, $\boldsymbol{F}$, and $\boldsymbol{C}$, respectively, there exist non-singular transform matrices $\boldsymbol{T}_1, \boldsymbol{T}_2$, and $\boldsymbol{T}_3 \in \mathbb{R}^{d \times d}$, such that $\boldsymbol{A} = \boldsymbol{U}_1^{[\mathrm{s}]} \cdot \boldsymbol{T}_1$, $\boldsymbol{F} = \boldsymbol{U}_2^{[\mathrm{s}]} \cdot \boldsymbol{T}_2$, and $\boldsymbol{C} = \boldsymbol{U}_3^{[\mathrm{s}]} \cdot \boldsymbol{T}_3$ [19]. Consequently, the task of computing the factor matrices of the PARAFAC decomposition boils down to the estimation of the transform matrices $\boldsymbol{T}_1, \boldsymbol{T}_2$, and $\boldsymbol{T}_3$ [19]. Based on this concept, let us look at the two-component case, i.e., $d = 2$, and the case where $d > 2$, separately. In the former, a closed-form solution of $\boldsymbol{A}$, $\boldsymbol{F}$, and $\boldsymbol{C}$ is available, whereas in the latter, multiple estimates of these factor matrices can be obtained by solving multiple SMDs. This gives rise to the need of a more significant adaptation of the DF algorithm.

For $d = 2$, we observe that $\widetilde{\boldsymbol{\mathcal{X}}}_0$, of which the PARAFAC decomposition is computed to update $\boldsymbol{A}$, $\boldsymbol{C}$, and $\boldsymbol{F}$, is of size $M_1 \times 2 \times M_3$. Therefore, we propose to reshape $\widetilde{\boldsymbol{\mathcal{X}}}_0$ into a two-slab tensor denoted by $\widetilde{\boldsymbol{\mathcal{X}}}_0'$ of size $M_1 \times M_3 \times 2$. The two three-mode slices of $\widetilde{\boldsymbol{\mathcal{X}}}_0'$ are written as

$$\left[\widetilde{\boldsymbol{\mathcal{X}}}_0'\right]_{(:,:,\ell)} = \boldsymbol{A} \cdot \mathrm{diag}\left\{\boldsymbol{F}(\ell,:)\right\} \cdot \boldsymbol{C}^{\mathrm{T}}, \ell = 1, 2. \tag{4}$$

Multiplying $\widetilde{\boldsymbol{\mathcal{X}}}_0'$ with $\boldsymbol{U}_1^{[\mathrm{s}]^{\mathrm{H}}}$ and $\boldsymbol{U}_3^{[\mathrm{s}]^{\mathrm{H}}}$ from the truncated HOSVD (3) in the one- and two-mode yields

$$\begin{aligned}\boldsymbol{S}_\ell &= \boldsymbol{U}_1^{[\mathrm{s}]^{\mathrm{H}}} \cdot \left[\widetilde{\boldsymbol{\mathcal{X}}}_0'\right]_{(:,:,\ell)} \cdot \boldsymbol{U}_3^{[\mathrm{s}]^*} \\ &= \boldsymbol{T}_1 \cdot \mathrm{diag}\left\{\boldsymbol{F}(\ell,:)\right\} \cdot \boldsymbol{T}_3^{\mathrm{T}}, \ell = 1, 2.\end{aligned}$$

Then the columns of $\boldsymbol{T}_1$ and $\boldsymbol{T}_3$ can be directly computed as the eigenvectors of $\boldsymbol{S}_1 \cdot \boldsymbol{S}_2^{-1}$ and $(\boldsymbol{S}_1^{-1} \cdot \boldsymbol{S}_2)^{\mathrm{T}}$, respectively [19]. With $\boldsymbol{A}$ and $\boldsymbol{C}$ obtained as $\boldsymbol{A} = \boldsymbol{U}_1^{[\mathrm{s}]} \cdot \boldsymbol{T}_1$ and $\boldsymbol{C} = \boldsymbol{U}_3^{[\mathrm{s}]} \cdot \boldsymbol{T}_3$, respectively, the two rows of $\boldsymbol{F}$ can be calculated following (4).

Hence, in the two-component case, $\boldsymbol{A}$, $\boldsymbol{C}$, and $\boldsymbol{F}$ are updated very efficiently in **Step 3** via the aforementioned algebraic procedures such that the iterative ALS scheme is avoided, while allowing the remaining steps of the DF algorithm to stay unchanged. We call the resulting scheme DF-EVD, as for $d = 2$ the SMDs reduce to two EVDs.

For $d > 2$, $\boldsymbol{T}_1, \boldsymbol{T}_2$, and $\boldsymbol{T}_3$ are obtained via SMDs [19]. Let us define

$$\boldsymbol{\mathcal{S}}_r = \boldsymbol{\mathcal{S}}^{[\mathrm{s}]} \times_r \boldsymbol{U}_r^{[\mathrm{s}]}, \quad r = 1, 2, 3 \tag{5}$$

using the outcome of the truncated HOSVD in (3), consequently $\boldsymbol{S}_{1,\ell} = [\boldsymbol{\mathcal{S}}_1]_{(\ell,:,:)}$ $(\ell = 1, \ldots, M_1)$, $\boldsymbol{S}_{2,\ell} = [\boldsymbol{\mathcal{S}}_2]_{(:,\ell,:)}$ $(\ell = 1, \ldots, d)$, and $\boldsymbol{S}_{3,\ell} = [\boldsymbol{\mathcal{S}}_3]_{(:,:,\ell)}$ $(\ell = 1, \ldots, M_3)$, all of size $d \times d$. Taking $\boldsymbol{S}_{3,\ell}$ as an example, they represent the M_3 three-mode slices of $\boldsymbol{\mathcal{S}}_3$. Further computing $\boldsymbol{S}_{3,\ell}^{[\mathrm{R}]} = \boldsymbol{S}_{3,\ell} \cdot \boldsymbol{S}_{3,p}^{-1}$ and $\boldsymbol{S}_{3,\ell}^{[\mathrm{L}]} = (\boldsymbol{S}_{3,p}^{-1} \cdot \boldsymbol{S}_{3,\ell})^{\mathrm{T}}$ leads to two sets of M_3 d-by-d matrices, where $\boldsymbol{S}_{3,p}$ is the slice with the minimum condition number [19].

Via the simultaneous diagonalization of $\boldsymbol{S}_{3,\ell}^{[\mathrm{R}]}$, we obtain $\boldsymbol{T}_1$ and consequently one estimate of $\boldsymbol{A}$, denoted by $\boldsymbol{A}^{(\mathrm{I})}$, thanks to the relation $\boldsymbol{A}^{(\mathrm{I})} = \boldsymbol{U}_1^{[\mathrm{s}]} \cdot \boldsymbol{T}_1$. Collecting the diagonal elements of $\boldsymbol{T}_1^{-1} \cdot \boldsymbol{S}_{3,\ell}^{[\mathrm{R}]} \cdot \boldsymbol{T}_1 \in \mathbb{R}^{d \times d}$ provides us with of the ℓ-th row of $\boldsymbol{C}^{(\mathrm{I})}$, an estimate of $\boldsymbol{C}$ $(\ell = 1, \ldots, M_3)$ up to a scaling ambiguity. Finally, the two-mode factor matrix $\boldsymbol{F}$ can be estimated via a least squares fit, i.e., $\boldsymbol{F}^{(\mathrm{I})} = \left[\widetilde{\boldsymbol{\mathcal{X}}}_0\right]_{(2)} \cdot \left[(\boldsymbol{C}^{(\mathrm{I})} \diamond \boldsymbol{A}^{(\mathrm{I})})^{+}\right]^{\mathrm{T}}$ [19].

TABLE I
DF-SMD FOR $d > 3$

Version 1:	**DF-SMD parallel**
Step 1	Initialize $\boldsymbol{A}$, $\boldsymbol{C}$, and $\boldsymbol{F}$ and construct $\widetilde{\boldsymbol{\mathcal{X}}}_0$ Update $\boldsymbol{V}_k$ $(k = 1, \ldots, M_3)$ and $\widetilde{\boldsymbol{\mathcal{X}}}_0$ accordingly Compute $\boldsymbol{A}^{(j)}$, $\boldsymbol{C}^{(j)}$, and $\boldsymbol{F}^{(j)}$ $(j = \mathrm{I}, \ldots, \mathrm{VI})$ **for** $j = \mathrm{I}, \ldots, \mathrm{VI}$, **respectively**
Step 2	Reconstruct $\widetilde{\boldsymbol{\mathcal{X}}}_0^{(j)}$ and update $\boldsymbol{V}_k^{(j)}$ $(k = 1, \ldots, M_3)$
Step 3	Update $\widetilde{\boldsymbol{\mathcal{X}}}_0^{(j)}$ with $\boldsymbol{V}_k^{(j)}$ $(k = 1, \ldots, M_3)$ and compute $\boldsymbol{A}^{(j)}$, $\boldsymbol{C}^{(j)}$, and $\boldsymbol{F}^{(j)}$
Step 4	Compute the reconstructed tensor $\hat{\boldsymbol{\mathcal{X}}}^{(j)}$ and calculate the residual $E_{\mathrm{R}}^{(j)}$ Repeat **Step 2 – Step 4** until $\Delta E_{\mathrm{R}}^{(q)}$, where $q = \underset{j=\mathrm{I},\ldots,\mathrm{VI}}{\arg\min}\ E_{\mathrm{R}}^{(j)}$ is smaller than a predefined threshold, and obtain $\boldsymbol{A} = \boldsymbol{A}^{(q)}$, $\boldsymbol{C} = \boldsymbol{C}^{(q)}$, $\boldsymbol{F} = \boldsymbol{F}^{(q)}$, and $\boldsymbol{V}_k = \boldsymbol{V}_k^{(q)}$ $(k = 1, \ldots, M_3)$
Version 2:	**DF-SMD one batch**
Step 1	Initialize $\boldsymbol{A}$, $\boldsymbol{C}$, and $\boldsymbol{F}$ and construct $\widetilde{\boldsymbol{\mathcal{X}}}_0$ Update $\boldsymbol{V}_k$ $(k = 1, \ldots, M_3)$
Step 2	Update $\widetilde{\boldsymbol{\mathcal{X}}}_0$ with $\boldsymbol{V}_k$ $(k = 1, \ldots, M_3)$ Compute $\boldsymbol{A}^{(j)}$, $\boldsymbol{C}^{(j)}$, and $\boldsymbol{F}^{(j)}$ $(j = \mathrm{I}, \ldots, \mathrm{VI})$
Step 3	Reconstruct $\widetilde{\boldsymbol{\mathcal{X}}}_0^{(j)}$ and update $\boldsymbol{V}_k^{(j)}$ $(k = 1, \ldots, M_3)$ for $j = \mathrm{I}, \ldots, \mathrm{VI}$, respectively
Step 4	Compute the reconstructed tensor $\hat{\boldsymbol{\mathcal{X}}}^{(j)}$ and calculate the residual $E_{\mathrm{R}}^{(j)}$ for $j = \mathrm{I}, \ldots, \mathrm{VI}$, respectively Select $\boldsymbol{V}_k = \boldsymbol{V}_k^{(q)}$ $(k = 1, \ldots, M_3)$, where $q = \underset{j=\mathrm{I},\ldots,\mathrm{VI}}{\arg\min}\ E_{\mathrm{R}}^{(j)}$, and accordingly $E_{\mathrm{R}} = E_{\mathrm{R}}^{(q)}$ Repeat **Step 2 – Step 4** until ΔE_{R} is smaller than a predefined threshold, and obtain $\boldsymbol{A} = \boldsymbol{A}^{(q)}$, $\boldsymbol{C} = \boldsymbol{C}^{(q)}$, and $\boldsymbol{F} = \boldsymbol{F}^{(q)}$

Similarly, jointly diagonalizing $\boldsymbol{S}_{3,\ell}^{[\mathrm{L}]}$ gives $\boldsymbol{T}_2$ and subsequently another set of factor matrix estimates, represented by $\boldsymbol{A}^{(\mathrm{II})}$, $\boldsymbol{C}^{(\mathrm{II})}$, and $\boldsymbol{F}^{(\mathrm{II})}$. From now on, we index the estimates of factor matrices obtained through the SMD of $\boldsymbol{S}_{2,\ell}^{[\mathrm{R}]}$ and $\boldsymbol{S}_{2,\ell}^{[\mathrm{L}]}$ as $^{(\mathrm{III})}$ and $^{(\mathrm{IV})}$ in the superscript, respectively, whereas $^{(\mathrm{V})}$ and $^{(\mathrm{VI})}$ in the case of $\boldsymbol{S}_{1,\ell}^{[\mathrm{R}]}$ and $\boldsymbol{S}_{1,\ell}^{[\mathrm{L}]}$, respectively.

Suppose we straightforwardly apply the aforementioned procedures to the DF algorithm to update $\boldsymbol{A}$, $\boldsymbol{C}$, and $\boldsymbol{F}$. Then $\boldsymbol{A}^{(j)}$, $\boldsymbol{C}^{(j)}$, and $\boldsymbol{F}^{(j)}$ $(j = \mathrm{I}, \ldots, \mathrm{VI})$ obtained in the first iteration would be used to reconstruct $\widetilde{\boldsymbol{\mathcal{X}}}_0^{(j)}$. In the second iteration, the resulting six $\widetilde{\boldsymbol{\mathcal{X}}}_0^{(j)}$ would be updated, and the PARAFAC decomposition of each would lead to six sets of factor matrix estimates, i.e., 36 sets in total. Thus, the number of candidate factor matrices would increase exponentially (up to 6^n in the n-th iteration), giving rise to a prohibitive computational complexity.

On the other hand, it is observed that simply resorting to a selection criterion proposed for SECSI [19] to directly select one set of factor matrices estimates of PARAFAC, i.e., $\boldsymbol{A}$, $\boldsymbol{C}$, and $\boldsymbol{F}$, in each iteration significantly slows down or in some occasions even hampers the convergence of the DF algorithm. This observation implies that the PARAFAC decomposition in **Step 3** of each iteration should not be taken as an isolated procedure, as all factor matrices of PARAFAC2 computed via the DF algorithm are related and have impact on each other.

Hence, two enhanced versions of the DF algorithms are developed to achieve a flexible control of the performance-complexity trade-off and to merge the SMD-based computation of the PARAFAC factor matrices into the context of PARAFAC2.

In the first version, referred to as "DF-SMD parallel", $\boldsymbol{A}^{(j)}$, $\boldsymbol{C}^{(j)}$, and $\boldsymbol{F}^{(j)}$ $(j = \mathrm{I}, \ldots, \mathrm{VI})$ are obtained in an initialization stage. Then in each of the subsequent iterations, for $j = \mathrm{I}, \ldots, \mathrm{VI}$, $\widetilde{\boldsymbol{\mathcal{X}}}_0^{(j)}$ are constructed, updated, and used to compute $\boldsymbol{A}^{(j)}$, $\boldsymbol{C}^{(j)}$, and $\boldsymbol{F}^{(j)}$, respectively. For instance, in the case of $\widetilde{\boldsymbol{\mathcal{X}}}_0^{(\mathrm{I})}$, only $\boldsymbol{\mathcal{S}}_3$ is computed with the outcome of its truncated HOSVD according to (5) and only $\boldsymbol{S}_{3,\ell}^{[\mathrm{R}]}$ constructed with the three-mode slices of $\boldsymbol{\mathcal{S}}_3$ are simultaneously diagonalized, finally yielding an update of $\boldsymbol{A}^{(\mathrm{I})}$, $\boldsymbol{C}^{(\mathrm{I})}$, and $\boldsymbol{F}^{(\mathrm{I})}$. As the above-mentioned procedures can be conducted independently for $j = \mathrm{I}, \ldots, \mathrm{VI}$, it can be easily deduced that the proposed DF-SMD parallel scheme allows a parallel computation, which is an intriguing feature.

TABLE II
COMPUTATION OF $\boldsymbol{A}^{(j)}$, $\boldsymbol{C}^{(j)}$, AND $\boldsymbol{F}^{(j)}$ $(j = \mathrm{I}, \ldots, \mathrm{VI})$ IN DF-SMD

tensor to decompose		target matrices for SMD	result of SMD	factor matrices
Ver. 2	Ver. 1			
$\widetilde{\boldsymbol{\mathcal{X}}}_0$	$\widetilde{\boldsymbol{\mathcal{X}}}_0^{(\mathrm{I})}$	$\boldsymbol{S}_{3,\ell}^{[\mathrm{R}]}$	$\boldsymbol{T}_1$	$\boldsymbol{A}^{(\mathrm{I})}, \boldsymbol{C}^{(\mathrm{I})}, \boldsymbol{F}^{(\mathrm{I})}$
	$\widetilde{\boldsymbol{\mathcal{X}}}_0^{(\mathrm{II})}$	$\boldsymbol{S}_{3,\ell}^{[\mathrm{L}]}$	$\boldsymbol{T}_2$	$\boldsymbol{A}^{(\mathrm{II})}, \boldsymbol{C}^{(\mathrm{II})}, \boldsymbol{F}^{(\mathrm{II})}$
	$\widetilde{\boldsymbol{\mathcal{X}}}_0^{(\mathrm{III})}$	$\boldsymbol{S}_{2,\ell}^{[\mathrm{R}]}$	$\boldsymbol{T}_1$	$\boldsymbol{A}^{(\mathrm{III})}, \boldsymbol{C}^{(\mathrm{III})}, \boldsymbol{F}^{(\mathrm{III})}$
	$\widetilde{\boldsymbol{\mathcal{X}}}_0^{(\mathrm{IV})}$	$\boldsymbol{S}_{2,\ell}^{[\mathrm{L}]}$	$\boldsymbol{T}_3$	$\boldsymbol{A}^{(\mathrm{IV})}, \boldsymbol{C}^{(\mathrm{IV})}, \boldsymbol{F}^{(\mathrm{IV})}$
	$\widetilde{\boldsymbol{\mathcal{X}}}_0^{(\mathrm{V})}$	$\boldsymbol{S}_{1,\ell}^{[\mathrm{R}]}$	$\boldsymbol{T}_2$	$\boldsymbol{A}^{(\mathrm{V})}, \boldsymbol{C}^{(\mathrm{V})}, \boldsymbol{F}^{(\mathrm{V})}$
	$\widetilde{\boldsymbol{\mathcal{X}}}_0^{(\mathrm{VI})}$	$\boldsymbol{S}_{1,\ell}^{[\mathrm{L}]}$	$\boldsymbol{T}_3$	$\boldsymbol{A}^{(\mathrm{VI})}, \boldsymbol{C}^{(\mathrm{VI})}, \boldsymbol{F}^{(\mathrm{VI})}$

In the DF-SMD parallel scheme, six batches of all factor matrices of PARAFAC2 are computed in a parallel manner for each iteration and are all used for the following iteration. Instead, we can select the set of candidate factor matrices $\boldsymbol{A}$, $\boldsymbol{C}$, $\boldsymbol{F}$, and $\boldsymbol{V}_k$ $(k = 1, \ldots, M_3)$ leading to the minimum residual of PARAFAC2 at the end of each iteration. Consequently, a single $\widetilde{\boldsymbol{\mathcal{X}}}_0$ is reconstructed in the follow-up iteration. We, therefore, name this version "DF-SMD one batch". Note that this is equivalent to applying a minimum residual-based selection criterion on the DF algorithm, i.e., the q-th batch of factor matrix estimates are selected where $q = \underset{j=\mathrm{I},\ldots,\mathrm{VI}}{\arg\min}\ E_{\mathrm{R}}^{(j)}$.

Table I presents an outline of the two schemes introduced above. In addition, we use Table II to summarize how $\boldsymbol{A}^{(j)}$, $\boldsymbol{C}^{(j)}$, and $\boldsymbol{F}^{(j)}$ $(j = \mathrm{I}, \ldots, \mathrm{VI})$ are computed in each iteration of the two methods, respectively, to further clarify the distinction.

IV. NUMERICAL RESULTS AND DISCUSSIONS

Now we assess the performance of the proposed DF-EVD and DF-SMD algorithms with both synthetic and real-world measurement data. All simulations have been conducted on an Intel Core i7-7700K 4.20 GHz machine running 64-bit Windows 10 and MATLAB R2016b 64-bit. First, we look at a two-component Scenario 1 with $M_1 = M_2 = M_3 = 5$ and a three-component Scenario 2 with $M_1 = 20$, $M_2 = 30$, and $M_3 = 10$. The factor matrices are randomly generated according to the PARAFAC2 model, while $\boldsymbol{\mathcal{N}}$ contains zero-mean uncorrelated Gaussian entries with variance σ_n^2. Accordingly, we define $\mathrm{SNR} = 1/\sigma_n^2$.

A comparison of different versions of the DF algorithm in terms of the complementary cumulative distribution function (CCDF) of the residual is shown in Fig. 1(a)–Fig. 1(e) considering various settings that differ in SNR and correlation (as indicated in the caption). We use DF-ALS to refer to the original DF scheme in [13] and include "GS" in the name of an algorithm whenever GS [14] is employed. If not otherwise stated, an SVD-based initialization [13] is employed in all DF algorithms. The threshold for convergence is set to 10^{-6}.

It is observed that DF-EVD and DF-SMD significantly outperform DF-ALS especially in critical scenarios, e.g., with bad-conditioned three-mode factor matrices (cf. Fig. 1(b) and 1(d)), indicating a stronger robustness of DF-EVD and DF-SMD. All settings shown here imply that GS is applicable to DF-EVD and DF-SMD and contributes to an even better performance. Moreover, it is worth noting that the initialization appears very important for DF algorithms (e.g., as observed in Fig. 1(a),

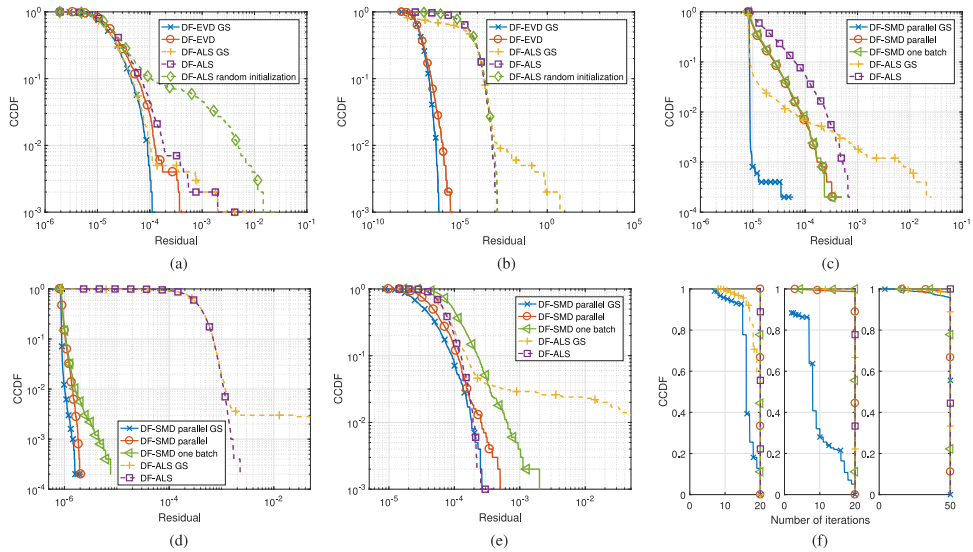

Fig. 1. CCDF of residuals for Scenario 1 with $M_1 = M_2 = M_3 = 5$, $d = 2$ (a - b), CCDF of residuals for Scenario 2 with $M_1 = 20, M_2 = 30, M_3 = 10$, $d = 3$ (c - e), and CCDF of number of iterations for Scenario 2 (f). (a) uncorrelated, SNR $= 40$ dB, maximally 50 iterations (b) bad conditioned three-mode factor matrix $\boldsymbol{C}$ with mutually correlated columns (correlation factor as 0.98), SNR $= 70$ dB, maximally 50 iterations (c) uncorrelated, SNR $= 50$ dB, maximally 20 iterations (d) bad conditioned three-mode factor matrix $\boldsymbol{C}$ with mutually correlated columns (correlation factor as 0.98), SNR $= 60$ dB, maximally 20 iterations (e) bad conditioned one-mode factor matrix $\boldsymbol{A}$ with mutually correlated columns (correlation factor as 0.98), SNR $= 50$ dB, maximally 50 iterations (f) CCDF of number of iterations for cases corresponding to (c) - (e), respectively.

TABLE III
NUMBER OF ITERATIONS/RUN TIME IN SECONDS OF DIFFERENT ALGORITHMS ON GC-MS DATA OF WINE AND SEPS/SEFS DATA

Algorithms	GC-MS data of wine					SEPs/SEFs				
	data set 1 44×14×200	data set 2 44×24×200	data set 3 44×11×200	data set 4 44×50×200	data set 5 44×9×200	data set 1 14×151×58	data set 2 14×151×59	data set 3 14×151×58	data set 4 14×151×57	data set 5 14×151×102
DF-EVD GS	9 / 0.44	4 / 0.21	6 / 0.21	7 / 0.37	17 / 0.77	17 / 0.26	518 / 4.28	6 / 0.05	261 / 2.35	5 / 0.08
DF-EVD	8 / 0.14	5 / 0.08	7 / 0.09	5 / 0.09	101 / 1.55	151 / 0.93	1810 / 11.2	7 / 0.04	512 / 3.76	6 / 0.06
DF-ALS GS	54 / 0.92	21 / 0.39	8 / 0.18	18 / 0.39	19 / 0.23	57 / 0.41	793 / 5.41	67 / 0.47	302 / 2.05	76 / 0.82
DF-ALS	58 / 0.99	54 / 1.06	14 / 0.23	49 / 1.10	135 / 2.01	169 / 1.14	3394 / 22	870 / 6.04	2412 / 16	294 / 3.79

TABLE IV
MEAN RUN TIME [MS] FOR DIFFERENT ALGORITHMS IN SCENARIO 1

Fig.	DF-EVD GS	DF-EVD	DF-ALS GS	DF-ALS	DF-ALS rand. ini.
1 (a)	23.16	49.18	27.47	70.56	50.30
1 (b)	6.22	19.95	58.02	44.58	29.18

where the SVD-based initialization is obviously superior to the random initialization).

On the other hand, the comparison between the two DF-SMD versions shows that when the tensor sizes are large enough (e.g., in Scenario 2) and the correlation in the data is not particularly strong, they perform similarly (cf. Fig. 1(c)). Note that DF-SMD one batch requires less memory compared to DF-SMD parallel, whereas by enabling a parallel processing, the latter can be faster than the former. Therefore, in such cases, DF-SMD is very flexible in fulfilling specific requirements with respect to memory and computation time.

Nevertheless, in critical settings with bad conditioned one-mode or three-mode factor matrices, DF-SMD one batch fails to offer a performance as good as DF-SMD parallel (cf. Fig. 1(d) and 1(e)) which can further benefit from an efficient practical implementation with parallel computing. Fig. 1(e) also shows that DF-ALS performs well enough in scenarios with a bad conditioned one-mode factor matrix.

We further compare the DF algorithms in terms of the mean run time for Scenario 1 and CCDF of the number of iterations for Scenario 2 in Table IV and Fig. 1(f), respectively.

It can be concluded that DF-EVD is very efficient, and incorporating GS [14] further enhances its efficiency. For instance, in the scenario of Fig. 1(b), the mean run time of DF-EVD is less than half of that for DF-ALS, and it decreases by about 69% thanks to the use of GS. In the three-component scenario, DF-SMD GS requires a much smaller number of iterations than the remaining schemes especially in the settings of Fig. 1(c) and Fig. 1(d). Note that the SMD scheme in [18] has been employed, whereas other SMD approaches, such as the JDTM algorithm [25], can also be used and may have impact on the performance.

Finally, Table IV shows the convergence time and the number of iterations when applying the DF algorithms on GC-MS data of wine [26], [14] and SEPs/SEFs data [27] that fit the PARAFAC2 model very well as mentioned in Section I. The threshold for convergence is set to 10^{-8} [14], and the maximum number of iterations is 10000. Five wine data tensors of model order two have been taken as representative examples, each corresponding to 44 wine samples (one-mode) with a mass spectrum from 5 m/z to 204 m/z (three-mode) measured at a certain number of elution time-points (two-mode) [14]. Moreover, we have analyzed four SEP tensors and one SEF tensor constructed with the time-frequency representation of signals in the low frequency range [9], thus having three modes with respect to frequency, time, and channels. DF-EVD and its GS enhanced version, DF-EVD GS, require a smaller number of iterations and in most occasions also less time for convergence compared to their ALS-based counterparts. In some cases, e.g., wine data set 1 and 4, DF-EVD turns out to be more efficient than DF-EVD GS.

REFERENCES

[1] L. G. Johnsen, P. B. Skou, B. Khakimov, and R. Bro, "Gas chromatography – mass spectrometry data processing made easy," *J. Chromatography A*, vol. 1503, pp. 57–64, 2017.

[2] B. Khakimov, R. J. Mongi, K. M. Sørensen, B. K. Ndabikunze, B. E. Chove, and S. B. Engelsen, "A comprehensive and comparative GCMS metabolomics study of non-volatiles in Tanzanian grown mango, pineapple, jackfruit, baobab and tamarind fruits," *Food Chem.*, vol. 213, pp. 691–699, 2016.

[3] M. Vosough and A. Salemi, "Exploiting second-order advantage using PARAFAC2 for fast HPLC-DAD quantification of mixture of aflatoxins in pistachio nuts," *Food Chem.*, vol. 127, pp. 827–833, 2011.

[4] A. Herrero, C. Reguera, M. C. Ortiz, and L. A. Sarabia, "Determination of dichlobenil and its major metabolite (BAM) in onions by PTV-GC-MS using PARAFAC2 and experimental design methodology," *Chemometrics Intell. Lab. Syst.*, vol. 133, pp. 92–108, 2014.

[5] I. García, M. C. Ortiz, L. Sarabia, and J. M. Aldama, "Validation of an analytical method to determine sulfamides in kidney by HPLC-DAD and PARAFAC2 with first-order derivative chromatograms," *Analytica Chimica Acta*, vol. 587, pp. 222–234, 2007.

[6] 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. 18th Eur. Signal Process. Conf.*, Aug. 2010, pp. 696–700.

[7] J. Haueisen, B. Schack, T. Meier, G. Curio, and Y. Okada, "Multiplicity in the high-frequency signals during the short-latency somatosensory evoked cortical activity in humans," *Clin. Neurophysiology*, vol. 112, pp. 1316–1325, 2001.

[8] J. Haueisen, L. Leistritz, T. Süsse, G. Curio, and H. Witte, "Identifying mutual information transfer in the brain with differential-algebraic modeling: Evidence for fast oscillatory coupling between cortical somatosensory areas 3b and 1," *NeuroImage*, vol. 37, pp. 130–136, 2007.

[9] Y. Cheng, M. Haardt, T. Götz, and J. Haueisen, "Using PARAFAC2 for multi-way component analysis of somatosensory evoked magnetic fields and somatosensory evoked electrical potentials," in *Proc. 10th IEEE Sens. Array Multichannel Signal Process. Workshop*, Jul. 2018, pp. 385–389.

[10] K. H. Madsen, N. W. Churchill, and M. Mørup, "Quantifying functional connectivity in multi-subject fMRI data using component models," *Human Brain Mapping*, vol. 38, no. 2, pp. 882–889, 2016.

[11] Y. Panagakis and C. Kotropoulos, "Automatic music tagging via PARAFAC2," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, May 2011, pp. 481–484.

[12] E. Pantraki and C. Kotropoulos, "Automatic image tagging and recommendation via PARAFAC2," in *Proc. IEEE 25th Int. Workshop Mach. Learn. Signal Process.*, Sep. 2015, pp. 1–6.

[13] H. A. L. Kiers, J. M. F. T. Berge, and R. Bro, "PARAFAC2 Part I. A direct fitting algorithm for the PARAFAC2 model," *J. Chemometrics*, vol. 13, pp. 275–294, 1999.

[14] K. Tian, L. Wu, S. Min, and R. Bro, "Geometric search: A new approach for fitting PARAFAC2 models on GC-MS data," *Talanta*, vol. 185, pp. 378–386, 2018.

[15] R. Harshman, "Foundations of the PARAFAC procedure: Models and conditions for an 'explanatory' multimodal factor analysis," *UCLA Working Papers Phonetics*, vol. 16, pp. 1–10, 1970.

[16] J. Kiefer, "Sequential minimax search for a maximum," *Proc. Amer. Math. Soc.*, vol. 4, pp. 502–506, 1953.

[17] F. Roemer and M. Haardt, "A closed-form solution for parallel factor (PARAFAC) analysis," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, Apr. 2008, pp. 2365–2368.

[18] T. Fu and X. Gao, "Simultaneous diagonalization with similarity transformation for non-defective matrices," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, May 2006, pp. IV–IV.

[19] F. Roemer and M. Haardt, "A semi-algebraic framework for approximate CP decompositions via simultaneous matrix diagonalizations (SECSI)," *Elsevier Signal Process.*, vol. 93, pp. 2722–2738, Sep. 2013.

[20] K. Naskovska, M. Haardt, P. Tichavsky, G. Chabriel, and J. Barrere, "Extension of the semi-algebraic framework for approximate CP decompositions via non-symmetric simultaneous matrix diagonalization," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, Mar. 2016, pp. 2971–2975.

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

[22] R. A. Harshman, "PARAFAC2: Mathematical and technical notes," *UCLA Working Papers Phonetics*, vol. 22, pp. 30–44, 1972.

[23] P. H. Schoenemann, "A generalized solution of the orthogonal procrustes problem," *Psychometrika*, vol. 31, no. 1, pp. 1–10, 1966.

[24] X. Luciani and L. Albera, "Canonical polyadic decomposition based on joint eigenvalue decomposition," *Chemometrics Intell. Lab. Syst.*, vol. 132, pp. 152–167, 2014.

[25] X. Luciani and L. Albera, "Joint eigenvalue decomposition using polar matrix factorization," in *Proc. Int. Conf. Latent Variable Anal. Signal Separation*, 2010, pp. 555–562.

[26] J. M. Amigo, T. Skov, R. Bro, J. Coello, and S. Maspoch, "Solving GC-MS problems with PARAFAC2," *TrAC Trends Analytical Chem.*, vol. 27, no. 8, pp. 714–725, 2008.

[27] T. Götz, R. Huonker, O. W. Witte, and J. Haueisen, "Thalamocortical impulse propagation and information transfer in EEG and MEG," *J. Clin. Neurophysiology*, vol. 31, no. 3, pp. 253–260, 2014.