Joint Factors And Rank Estimation For The Canonical Polyadic Decomposition Based On Convex Optimization

Ouafae Karmouda, Jérémie Boulanger, Remy Boyer

IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Digital Object Identifier

Joint Factors And Rank Estimation For The Canonical Polyadic Decomposition Based On Convex Optimization

OUAFAE KARMOUDA[1], (Student Member, IEEE), JEREMIE BOULANGER[2], AND REMY BOYER.[3], (Senior Member, IEEE)

[1]University of Lille, CNRS, CRIStAL, 59655 Lille, France (e-mail: ouafae.karmouda@univ-lille.fr)
[2]University of Lille, CNRS, CRIStAL, 59655 Lille, France (e-mail: jeremie.boulanger@univ-lille.fr)
[3]University of Lille, CNRS, CRIStAL, 59655 Lille, France (e-mail: remy.boyer@univ-lille.fr)

ABSTRACT Estimating the minimal number of rank-1 tensors in the Canonical Polyadic Decomposition (CPD), known as the canonical rank, is a challenging area of research. To address this problem, we propose a method based on convex optimization to jointly estimate the CP factors and the canonical rank called FARAC for joint FActors and RAnk canonical estimation for the PolyadiC Decomposition. We formulate the FARAC method as a convex optimization problem in which a sparse promoting constraint is added to the superdiagonal of the core tensor of the CPD, whereas the Frobenius norm of the offdiagonal terms is constrained to be bounded. We propose an alternated minimization strategy for the Lagrangien to solve the optimization problem. The FARAC method has been validated on synthetic data with varying levels of noise, as well as on three real data sets. Compared to state-of-the-art methods, FARAC exhibits very good performance in terms of rank estimation accuracy for a large range of SNR values. Additionally, FARAC can handle the case in which the canonical rank exceeds one of the dimensions of the input tensor.

INDEX TERMS Rank and factors estimation, Canonical Polyadic Decomposition, convex optimization

I. INTRODUCTION

Large amounts of data are usually generated by sensor networks, massive experiment, simulations, *etc*. In a very wide range of applications, data need more than two dimensions for efficient description [2], [10], [28], [41]. To represent such data, multidimensionnal arrays (*a.k.a.* tensors) are suitable in order to capture complex interactions among input features of data. Tensors can be seen as a generalization of vectors (1st order tensors) and matrices (2nd order tensors). The order of a tensor is its number of dimensions.

Various tensor decompositions exist to mitigate the curse of dimensionality *i.e.* to avoid the exponential growth of storage cost [3], [12], [18], [22], [29].

The most popular tensor decomposition is the Canonical Polyadic Decomposition (CPD) [37]. The CPD is widespreadly used in differents fields such as chemometrics, telecommunications, blind source separation [16], [24], [26], [27], [39]. CPD is particularly attractive owing to its uniqueness property [1], [14]. However, the CPD requires knowledge of the rank. Unfortunately, the problem of determining the canonical rank is NP-hard [4].

In the present study, we propose estimating both the canonical rank and the CP factors from noisy observations. We give a formulation of the problem of interest as a convex optimization problem. The reminder of the paper is organised as follows:

First, we introduce some notations and preliminaries in multilinear algebra in section II. We then review some existing works for the estimation of the canonical rank III. In section IV, we describe our proposed approach and our algorithm for solving the problem. As a final step, we do a number of numerical experiments in section VI to evaluate and compare our proposed approach with CORCONDIA (CORe CONsistency DIAgnostic) method.

II. NOTATIONS AND ALGEBRAIC BACKGROUND

In this section, we recall some algebraic definitions on tensor algebra from [37]:

***Definition** 1:* (Inner product): The inner product of two N-order tensors $\mathcal{X}, \mathcal{Y} \in \mathbb{R}^{I_1 \times \cdots \times I_N}$ is defined as:

$$\langle \mathcal{X}, \mathcal{Y} \rangle = \sum_{i_1=1}^{I_1} \sum_{i_2=1}^{I_2} \cdots \sum_{i_N=1}^{I_N} \mathcal{X}(i_1, \ldots, i_N) \mathcal{Y}(i_1, \ldots, i_N).$$

***Definition** 2:* (Tensor Frobenius Norm): The norm of a tensor $\mathcal{X}$ is defined as:

$$||\mathcal{X}||_F = \sqrt{\sum_{i_1=1}^{I_1}\sum_{i_2=1}^{I_2}\cdots\sum_{i_N=1}^{I_N}\mathcal{X}^2(i_1,\ldots,i_N)}.$$

***Definition** 3:* (n-mode multiplication) The n-mode product of a tensor $\mathcal{X} \in \mathbb{R}^{I_1\times\cdots\times I_N}$ with a matrix $U \in \mathbb{R}^{J\times I_n}$ is a tensor of order N and size $I_1 \times \ldots I_{n-1} \times J \times \cdots \times I_N$ as shown by:

$$(\mathcal{X}\times_n U)(i_1,\ldots,i_{n-1},i,i_{n+1},\ldots,i_N) = \sum_{i_n=1}^{I_n}\mathcal{X}(i_1,\ldots,i_N)U(i,i_n).$$

***Definition** 4:* (Diagonal tensor) A tensor $\mathcal{X} \in \mathbb{R}^{I_1\times\cdots\times I_N}$ is diagonal if all of its entries are zero except thoes in its superdiagonal, that is: $\mathcal{X}(i_1,\ldots,i_N) \neq 0$ only if $i_1 = i_2 = \cdots = i_N$.

***Definition** 5:* (Rank-one tensor) A tensor $\mathcal{X} \in \mathbb{R}^{I_1\times\cdots\times I_N}$ of order N is rank one if it can be written as the outer product of N vectors:

$$\mathcal{X} = u_1 \circ \cdots \circ u_N := \bigcirc_{n=1}^{N} u_n, \tag{1}$$

where $u_n \in \mathbb{R}^{I_n}$ with $1 \leq n \leq N$.

***Definition** 6:* (Unfolding) The n-mode unfolding of a tensor $\mathcal{X}$ is a matrix denoted by $X_{(n)}$, whose columns are the n-mode fibers of $\mathcal{X}$.

***Definition** 7:* (Tucker model) The Tucker decomposition transforms a tensor $\mathcal{X} \in \mathbb{R}^{I_1\times\cdots\times I_N}$ of multilinear ranks $(R_1,\ldots,R_N)$ using N factors $U_n \in \mathbb{R}^{I_n\times R_n}$ by a core tensor $\mathcal{G} \in \mathbb{R}^{R_1\times\cdots\times R_N}$ as follows:

$$\mathcal{X} = \mathcal{G}\times_1 U_1\cdots\times_N U_N := \mathcal{G}\times_{n=1}^{N} U_n$$
$$= \sum_{r_1}^{R_1}\cdots\sum_{i_N}^{R_N}\mathcal{G}(r_1,\ldots,r_N)u_{1,r_1}\circ\cdots\circ u_{N,r_N},$$

where $u_{n,r} \in \mathbb{R}^{I_n}$ is the r-th column of the n-th factor matrix U_n with $1 \leq n \leq N$.

***Definition** 8:* (CANDECOMP Decomposition (CPD)) A tensor of order N and canonical rank R follows a CPD if it admits a factorization as a sum of R rank-one tensors. The CPD of a tensor $\mathcal{X} \in \mathbb{R}^{I_1\times\cdots\times I_N}$ is given by:

$$\mathcal{X} = \sum_{r=1}^{R} u_{1,r}\circ\cdots\circ u_{N,r} = \sum_{r=1}^{R}\bigcirc_{n=1}^{N} u_{n,r}, \tag{2}$$

CPD is a special case of the Tucker model where the core tensor is diagonal and the multilinear ranks are all equal to R. Hence, CPD in eq. (2) can be written in the Tucker format as follows:

$$\mathcal{X} = \mathcal{I}\times_{n=1}^{N} U_n.$$

***Definition** 9:* (Tensor rank) The rank R of a tensor is defined as the smallest number of components in an exact CPD.

We have the following upper bound on the maximum rank (the largest attainable rank) for a 3-order tensor $\mathcal{X} \in \mathbb{R}^{I_1\times I_2\times I_3}$ [14]:

$$R \leq \min\{I_1I_2, I_1I_3, I_2I_3\}.$$

III. RELATED WORKS

Existing approaches for the canonical rank detection include:

- CORe CONsistency DIAgnostic (CORCONDIA) [33] is a heuristic method for detecting the number of components of the CP model. It measures the similarity between the estimated core tensor from the ideal identity core (called core consitency) for different CP models. By analyzing the gap between the core consistency of different CP models, it determines the correct number of components. Taking the last model (starting with the one-component model), whose core array is still similar to the ideal diagonal tensor, gives the proper number of components to use. CORCONDIA is intuitive and has a simple approach. However, its computation is prohibitive even for small tensors [8]. In addition, the CORCONDIA method can not handle the case when the canonical rank exceeds one of the dimensions of the input tensor.
- A fast version of CORCONDIA was presented in [8]. It suggests an efficient way to compute the CORCONDIA diagnostic that takes advantage of sparse data and works well as the tensor size grows. In cases where either the tensor or the factors or both are sparse [8], their algorithm significantly outperforms the state-of-the-art baselines and scales well when the tensor size increases. In the fully dense scenario, their proposed algorithm is as good as the state of the art (The CORCONDIA method) for rank estimation.
- Automatic Relevance Determination (ARD) [23] is a Bayesian approach applied to the Tucker and CP models. In ARD, entries of CP factors are assigned a Gaussian prior. The objective is to find the rank and the CP factors by solving an l_2-regularized CP decomposition. By assigning priors to model hyperparameters and learning the hyperparameters of these priors, ARD reduces the excess of components and simplifies the core structure.
- Detection of the number of components in CANDECOMP models via minimum description length (N-D MDL) [17] is an extension of the minimum description length (MDL) to the multilinear case for detecting the number of components in a CP model using the generalized unfolding of the observation tensor. Under the condition that the tensor rank is smaller than the size of the most squared unfolded matrix, the generalized N-D MDL criterion estimates the number of components of the CPD. The drawback of the multilinear MDL algorithm is that it fails to work when the tensor rank is larger than the size of the most squared unfolded matrix [17].

- Using a Bayesian approach, [20] integrates the tensor rank determination into the CPD. A scalable algorithm is developed that processes mini batch data at a time by incorporating stochastic optimization into the probalistic tensor CPD.
- Automatic tensor rank estimation for nonnegative tensor CPD is proposed in [19]. By interpreting the nonnegative CPD using probability density functions, the problem is formulated as a statistical inference.
- The Gaussian-gamma prior was introduced in [21] within a Bayesian framework in the context of probabilistic CPD modeling for automatic rank determination.
- The GSL-CP method proposed in [40] estimates the rank as well as the loading matrices of the decomposed tensor. Without knowing the expected rank, the CPD is computed by promoting group sparsity of the loading matrix in orthonormal subspace.
- The canonical rank of a tensor is estimated in [6] using an improved version of MDL (iMDL).
- Tensor learning models for regression are proposed in [38]. In their regularization process, they employ the group-sparsity norm, which promotes a low-rank decomposition of the CP core as well as automatic selection of the rank during the learning procedure.
- Using orthogonal CPD, [30] views the superdiagonal of the CP core as analogous to the vector of singular values. To determine the rank of an incomplete tensor, they integrate a regularization with the CP-based tensor nuclear norm.
- The method proposed in [25] is based on convolutional neural networks (CNN) with a pre-decomposition using CPD providing rank-one components to the CNN.

In short, state-of-the-art methods use either Bayesian approaches, propose rank estimation within a learning framework, or use too restricting constraints like factor orthogonality. In contrast to existing methods, the proposed method belongs to the family of deterministic parameter estimators.

IV. PROPOSED METHOD (FARAC)

The purpose of this section is to present the method FARAC for estimating the canonical rank and the CP factors simultaneously. This is accomplished by minimizing the superdiagonal of the core tensor using the l_0 norm as an objective function, as well as adding a constraint on the reconstruction error and an another one on the offdiagonal terms by allowing them to be non-zero but bounded. Our goal is to find a CP core tensor structured as shown in Fig. 1. The canonical rank is the number of strictly positive and ordered values of the tensors' superdiagonal [1].

In the following, $\mathcal{G}$ denotes the CP core tensor, λ its superdiagonal and $\tilde{\mathcal{G}}$ is defined according to:

$$\mathcal{G} = \text{diag}(\lambda) + \tilde{\mathcal{G}},$$

[1]The sign of the superdiagonal entries can always be absorbed into the factors.

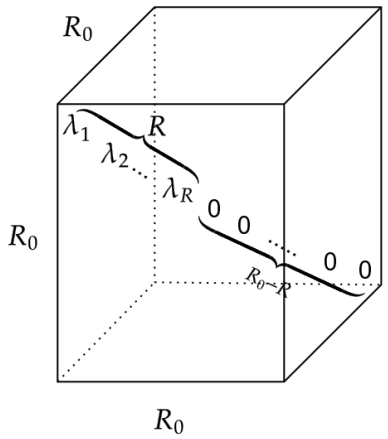

FIGURE 1. CP core tensor $\mathcal{G}$ of a rank-R tensor of order 3 and size R_0 with $\lambda_1 > \cdots > \lambda_R > 0$.

where diag(λ) is a diagonal tensor whose superdiagonal is λ. Our optimization problem can then be expressed mathematicaly as follows:

$$\begin{aligned} &\underset{\mathcal{G},(U_n)_n}{\text{minimize}}\ ||\lambda||_0, \\ &\text{subject to}\ \frac{1}{2}\left|\left|\mathcal{X} - \mathcal{G}\overset{N}{\underset{n=1}{\times_n}} U_n\right|\right|_F^2 \leq \epsilon_1, \\ &\qquad\qquad \frac{1}{2}\left|\left|\tilde{\mathcal{G}}\right|\right|_F^2 \leq \epsilon_2, \end{aligned} \tag{3}$$

where ϵ_1 and ϵ_2 are small positive constants. Unlike existing methods that estimate the canonical rank, we do not constrain the CP factors to be orthogonal [11], [13]. However, since minimizing the l_0 norm is NP-hard [5], we minimize the l_1 norm of λ. Eq. (3) becomes:

$$\begin{aligned} &\underset{\mathcal{G},(U_n)_n}{\text{minimize}}\ \gamma_1||\lambda||_1, \\ &\text{subject to}\ \frac{1}{2}\left|\left|\mathcal{X} - \mathcal{G}\overset{N}{\underset{n=1}{\times_n}} U_n\right|\right|_F^2 \leq \epsilon_1, \\ &\qquad\qquad \frac{1}{2}\left|\left|\tilde{\mathcal{G}}\right|\right|_F^2 \leq \epsilon_2, \end{aligned} \tag{4}$$

where γ_1 is a strictly positive hyper-parameter.

A. DERIVATION OF OUR APPROACH (FARAC)

In this section, we present the proposed approach for solving the optimization problem with constraints in eq. (4). Recall that the Lagrangien function is the augmented objective function by the constraint equations using the Lagrangien multipliers. Following this, the Lagrangien function of the problem in eq. (4) is given by:

$$\begin{aligned} &\mathcal{L}_{\mathcal{G},\{U_n\}} = \\ &\gamma_1||\lambda||_1 + \frac{\gamma_2}{2}\left(\left|\left|\mathcal{X} - \mathcal{G}\overset{N}{\underset{n=1}{\times_n}} U_n\right|\right|_F^2 - \epsilon_1\right) + \frac{\gamma_3}{2}\left(\left|\left|\tilde{\mathcal{G}}\right|\right|_F^2 - \epsilon_2\right), \end{aligned} \tag{5}$$

where γ_2 and γ_3 are two strictly positive Lagrange multipliers. According to the Lagrangien method [35], $\mathcal{L}_{\mathcal{G},\{U_n\}}$ is

minimized with respect to $\{U_n\}_n$, $\tilde{\mathcal{G}}$ and λ. To do that, we will proceed iteratively. We first minimize $\mathcal{L}_{\mathcal{G},\{U_n\}}$ w.r.t the n-th CP factor U_n at each iteration, assuming the remaining factors and $\mathcal{G}$ are known. This is a classical linear regression problem. Following that, we minimize $\mathcal{L}_{\mathcal{G},\{U_n\}}$ w.r.t $\tilde{\mathcal{G}}$ using the factors $\{U_n\}_n$ from the previous step. The convexity w.r.t $\tilde{\mathcal{G}}$ is demonstrated in Appendix B. Then, we update $\tilde{\mathcal{G}}$ using the Adam optimizer [7]. Finally, we minimize $\mathcal{L}_{\mathcal{G},\{U_n\}}$ w.r.t λ, which is also a convex optimization problem as shown in Appendix B. Its solution is given by the soft thresholding operator [34] using the current updates of $\tilde{\mathcal{G}}$ and $\{U_n\}_n$. The final expressions of the exact solutions of CP factors, the gradient of the Lagrangien w.r.t the $\tilde{\mathcal{G}}$ and the formula to update the superdiagonal of $\mathcal{G}$ are given below. Details of the computations can be found in Appendix A.

- CP factors U_n:

$$U_n = X_{(n)} \left(\bigotimes_{\substack{l=N \\ l\neq n}}^{l=1} U_l \right) G_{(n)}^T \left(\bigotimes_{\substack{l=N \\ l\neq n}}^{l=1} \left(U_l^T U_l\right)^{-1} \right). \quad (6)$$

- Gradient of $\mathcal{L}_{\mathcal{G},\{U_n\}}$ w.r.t the offdiagonals of $\mathcal{G}$ *i.e* $\mathcal{G}(r_1,\ldots,r_N)$ such that $r_1 \neq r_2$ or $\ldots$ or $r_{N-1} \neq r_N$:

$$\begin{aligned}[\nabla_{\tilde{\mathcal{G}}}(\mathcal{L}_{\mathcal{G},\{U_n\}})](r_1,\ldots,r_N) = \\ -\gamma_2 \sum_{i_1=1}^{I_1} \cdots \sum_{i_N}^{I_N} \mathcal{A}(i_1,\ldots,i_N) \\ \left[\prod_{n=1}^{N} U_n(i_n,r_n)\right] + \gamma_3 \tilde{\mathcal{G}}(r_1,\ldots,r_N), \end{aligned} \quad (7)$$

where $\mathcal{A} = \mathcal{X} - \mathcal{G} \overset{N}{\underset{n=1}{\times}}_n U_n$.

- Formula for updating the superdiagonal of $\mathcal{G}$:

$$\begin{aligned}\lambda_l = S_{\frac{\gamma_1}{\gamma_2}}\Big(-\Big\langle \bigcirc_{n=1}^{N} u_{n,l}, \sum_{\substack{r=1 \\ r\neq l}}^{R_0} \lambda_r \bigcirc_{n=1}^{N} u_{n,l} \Big\rangle + \\ \Big\langle \mathcal{X} - \tilde{\mathcal{G}} \overset{N}{\underset{n=1}{\times}}_n U_n, \bigcirc_{n=1}^{N} u_{n,l} \Big\rangle \Big), \end{aligned} \quad (8)$$

where $S_{\frac{\gamma_1}{\gamma_2}}$ is the soft thresholding operator [34] defined as follows:

$$S_\mu(x) = \begin{cases} x-\mu \text{ if } x > \mu \\ 0 \text{ if } |x| \leq \mu \\ x+\mu \text{ if } x < -\mu \end{cases} \quad (9)$$

The whole algorithm of our derived approach is described in Algorithm 1.

V. COMPLEXITY ANALYSIS

- We evaluate the complexity of Algorithm 1 by taking into consideration the SVDs in the initialization and the main parts of the algorithm, as the computations of

Algorithm 1: Joint **FA**ctors and **RA**nk **C**anonical estimation for the Polyadic decomposition (FARAC)

Input: $\mathcal{X} \in \mathbb{R}^{I_1 \times \ldots I_N}$: CP Tensor,
R_0: Upper bound of the rank of $\mathcal{X}$.
Require: α: Stepsize;
$\beta_1, \beta_2 \in [0,1]$: Exponential decay rates;
T: Maximum number of iterations
$\epsilon = 10^{-8}$: Parameter to avoid numerical instability.
Initialize: For $1 \leq n \leq N$:

$$U_n^{(0)} = \begin{cases} svd(X_{(n)}, R_0) \text{ if } I_n > R_0, \\ \text{conc}\big(svd(X_{(n)}, R_0), \\ (R_0 - I_n)\text{random uniform vectors}\big) \text{ else.} \end{cases}$$

- $\mathcal{G}^{(0)} \sim \mathcal{U}(0,1)$ of order N and size $(R_0,\ldots,R_0)$.
- $m_{\tilde{\mathcal{G}}}^{(0)} = v_{\tilde{\mathcal{G}}}^{(0)} = 0$ (Initialize the first and the second moment estimates).

for $t = 1,\ldots,T$:

1: Compute ${U_n}^{(t)}$ from eq. (6).
2: Compute the gradients of $\mathcal{L}_{\mathcal{G},\{U_n\}}$ w.r.t $\tilde{\mathcal{G}}$ using (7).
3: Update biased first moment estimate of the offdiags:

$$m_{\tilde{\mathcal{G}}}^{(t)} \leftarrow \beta_1 m_{\tilde{\mathcal{G}}}^{(t-1)} + (1-\beta_1)\nabla_{\tilde{\mathcal{G}}}^{(t)}.$$

4: Update biased second raw moment estimate of the offdiags:

$$v_{\tilde{\mathcal{G}}}^{(t)} \leftarrow \beta_2 v_{\tilde{\mathcal{G}}}^{(t-1)} + (1-\beta_2){\nabla^2}_{\tilde{\mathcal{G}}}^{(t)},$$

5: Compute bias-corrected first and second moment estimates of the offdiags:

$$\hat{m}_{\tilde{\mathcal{G}}}^{(t)} \leftarrow \frac{m_{\tilde{\mathcal{G}}}^{(t)}}{1-\beta_1^t}; \quad \hat{v}_{\tilde{\mathcal{G}}}^{(t)} \leftarrow \frac{v_{\tilde{\mathcal{G}}}^{(t)}}{1-\beta_2^t},$$

6: Update $\tilde{\mathcal{G}}$:

$$\tilde{\mathcal{G}}^{(t)} \leftarrow \tilde{\mathcal{G}}^{(t-1)} - \alpha \frac{\hat{m}_{\tilde{\mathcal{G}}}^{(t)}}{\sqrt{\hat{v}_{\tilde{\mathcal{G}}}^{(t)}} + \epsilon}.$$

7: Update the superdiagonal of $\mathcal{G}$ using eq. (8):

$$\begin{aligned}\lambda_l^{(t)} = S_{\frac{\gamma_1}{\gamma_2}}\Bigg(-\Big\langle \bigcirc_{n=1}^{N} u_n^{(t)}, \sum_{\substack{r=1 \\ r\neq l}}^{R_0} \lambda_r^{(t)} \bigcirc_{n=1}^{N} u_n^{(t)} \Big\rangle + \\ \Big\langle \mathcal{X} - \tilde{\mathcal{G}} \overset{N}{\underset{n=1}{\times}}_n U_n^{(t)}, \bigcirc_{n=1}^{N} u_n^{(t)} \Big\rangle \Bigg), \end{aligned}$$

where $S_{\frac{\gamma_1}{\gamma_2}}$ is defined in eq. (9).

end for
CP-rank: Number of non-zero values of the superdiagonal of $\mathcal{G}$.
CP factors: Columns of $U_n^{(T)}$ with indices of non-zero values of the superdiagonal of $\mathcal{G}$
Returns: [CP-rank, CP factors]

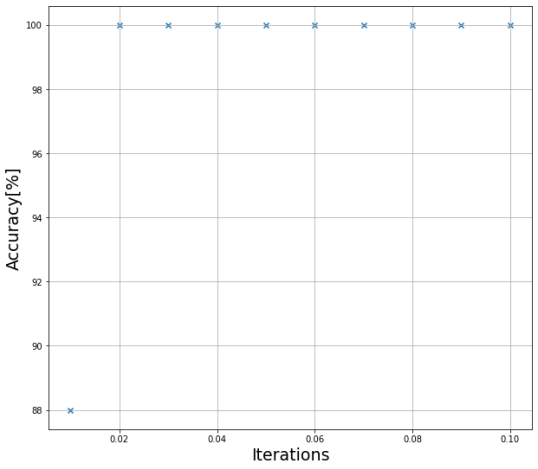

FIGURE 2. Accuracy of rank estimation of a tensor of size $5 \times 5 \times 5$ w.r.t the threshhold parameter. The true rank is $R = 2$ and $R_0 = 5$.

SNR [db]	15	20	25	30	35	40
Accuracy [%]	72	77	87	91	96	100

TABLE 1. Accuracy of FARAC w.r.t SNR for a tensor of a noisy tensor $\mathcal{X}_{\text{noise}}$ of size $5 \times 5 \times 5$. The true rank is $R = 6$ and $R_0 = 7$.

the different gradients and the shrinkage operator. The complexity is evaluated as follows:

$$\mathcal{O}\left(\left(\sum_{n=1}^{N} I_n\right) R_0^2 + T\left[N + \left(\sum_{n=1}^{N} I_n\right) R_0 + \left(\prod_{n=1}^{N} I_n\right)\left(N + R_0 + N R_0^{N-1}\right)\right]\right), \quad (10)$$

where T is the number of iterations.

- If $I = \max(I_1, \ldots, I_N)$, then the complexity in eq. (10) becomes:

$$\mathcal{O}\left(I R_0^2 + T\left(N + I R_0 + I(N + R_0 + N R_0^{N-1})\right)\right).$$

VI. NUMERICAL EXPERIMENTS

All the experiments are conducted on a computer with an Intel Core i7 9th generation 2.6 GHz processor and 32 Go RAM memory running Windows 10.

It should be noted that the gradients' computations are automatically done on the Tensorflow framework [9]. The learning rate and the exponential decay rates used in the Adam optimizer are all equal to their default values ($\alpha = 0.001$, $\beta_1 = 0.09, \beta_2 = 0.0999$).

A. NUMERICAL SIMULATIONS

1) Synthetic data

We create a synthetic rank-R real valued tensor $\mathcal{X}$ of order N using the CP model. The CP factors are derived from a single realization of the normal standard distribution. $\mathcal{B}$ is a zero

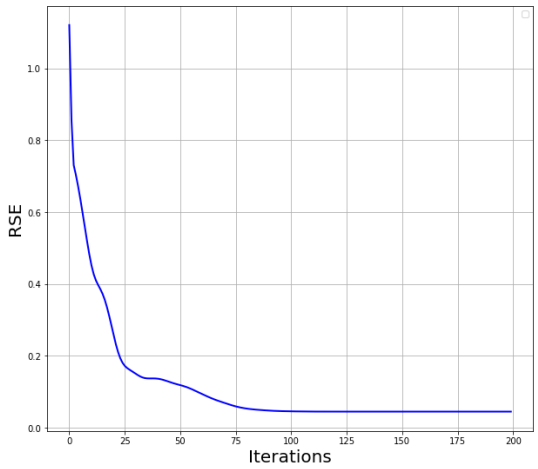

FIGURE 3. Convergence curve of the mean reconstruction error using the Relative Square Error (RSE) along iterations using FARAC. We used a noisy tensor $\mathcal{X}_{\text{noise}}$ with a size of $5 \times 5 \times 5$ and an SNR of $25db$. The threshold parameter is equal to 0.02.

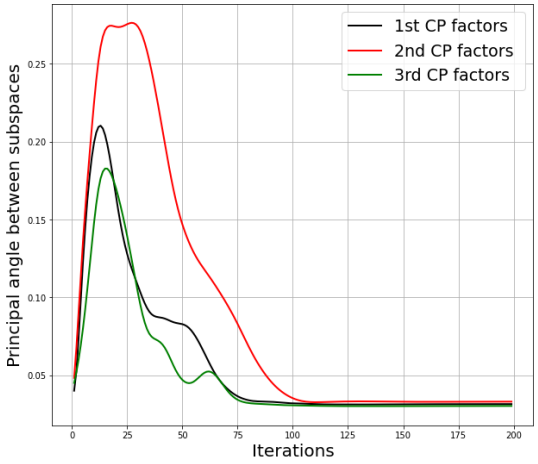

FIGURE 4. The mean principal angle between the three subspaces spanned by CP factors along iterations using our method. We used a noisy tensor $\mathcal{X}_{\text{noise}}$ with a size of size $5 \times 5 \times 5$ and an SNR of $25db$. The threshold parameter is equal to 0.02.

mean, unit variance and white noise. Our noisy data tensor is given by:

$$\mathcal{X}_{\text{noise}} = \mathcal{X}' + \sigma \mathcal{B}', \quad (11)$$

where $\mathcal{X}' = \frac{\mathcal{X}}{||\mathcal{X}||_F}$ and $\mathcal{B}' = \frac{\mathcal{B}}{||\mathcal{B}||_F}$. Hence, the SNR will be calculated using the following formula:

$$SNR = -10 \log_{10} \sigma^2 \in [0\,db, 40\,db].$$

FARAC has been runed on a rank-2 tensor $\mathcal{X}$ with size $I \times I \times I$ where $R_0 = I = 5$. Experiments are conducted on a tensor with these parameters until other settings are indicated. Similar results are obtained for tensors with other orders and sizes. By using $\mathcal{X}$ and eq. (11), we generate 100 noisy realisations of the inpout tensor. Accuracy is defined as the proportion of realizations where the estimated rank is accurate.

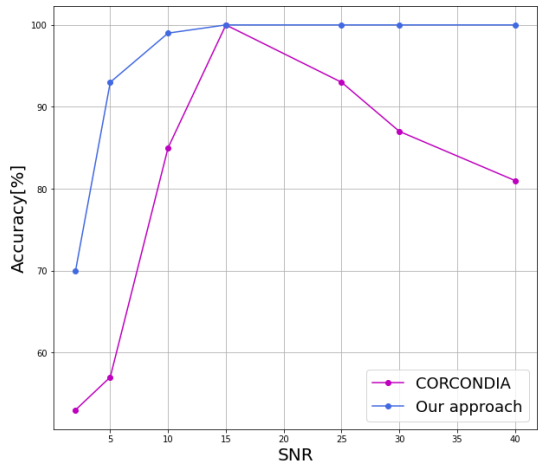

FIGURE 5. FARAC Vs. CORCONDIA accuracy w.r.t SNR for a tensor of a noisy tensor $\mathcal{X}_{\text{noise}}$ with SNR values ranging from 0db to 40db. $\mathcal{X}_{\text{noise}}$ is of size $5 \times 5 \times 5$. Different CP models with a rank ranging from 1 to 5 are used to fit CORCONDIA. $R = 2$ is the true rank.

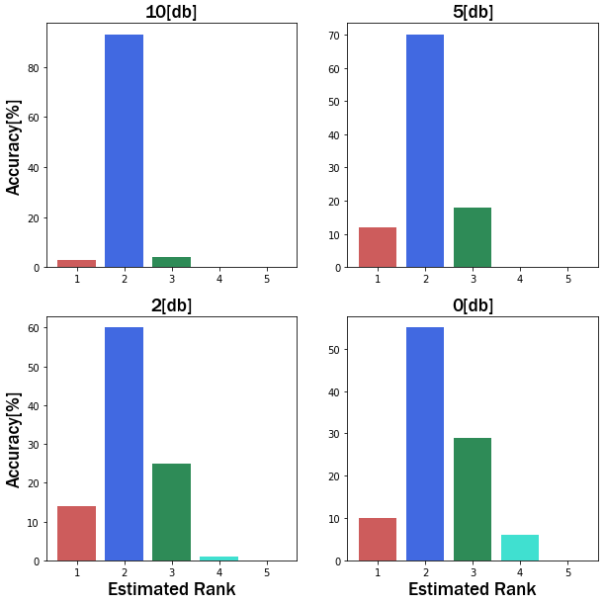

FIGURE 6. Accuracy of rank estimation of a noisy tensor $\mathcal{X}_{\text{noise}}$ with different low SNR values ranging from 0db to 10db using FARAC. The size of $\mathcal{X}_{\text{noise}}$ is $5 \times 5 \times 5$. The used large bound of rank used is $R_0 = 5$, while the true rank is $R = 2$.

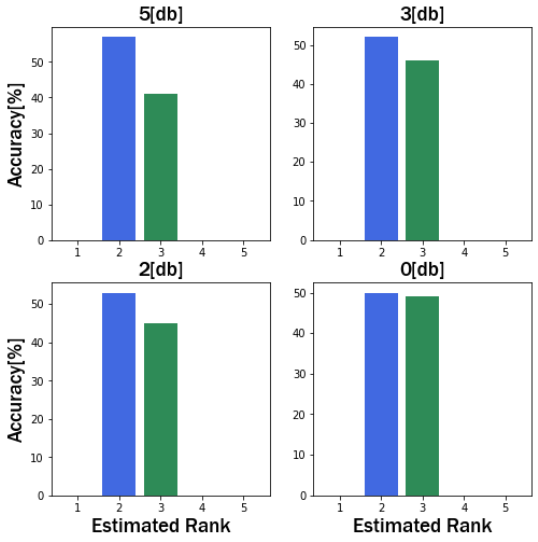

FIGURE 7. The CORCONDIA approach's accuracy for a noisy tensor $\mathcal{X}_{\text{noise}}$. SNR values range from 0db to 5db. The size of $\mathcal{X}_{\text{noise}}$ is $5 \times 5 \times 5$. Different CP models with a rank ranging from 1 to 5 are used to fit CORCONDIA. $R = 2$ is the true rank.

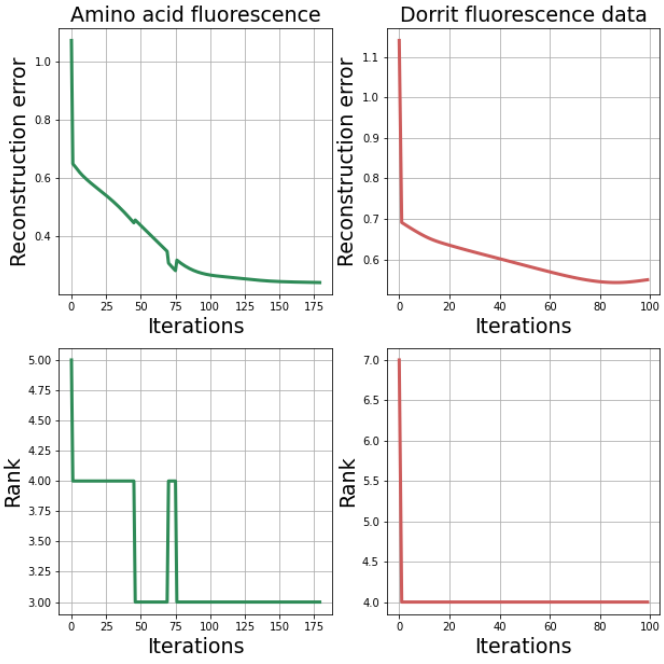

FIGURE 8. Convergence losses of FARAC on the Amino acid and the Dorrit fluorescence datasets (top) with rank estimation over iterations (bottom).

2) Real datatsets

- Amino acid fluorescence: As described in [31], the data set includes the excitation and emission spectra of five samples of different concentrations of tyrosine, tryptophane and phenylalanine, forming a tensor of 5 (samples) $\times$ 51 (excitation) $\times$ 201 (emission). This dataset can be described by a rank-3 CP model.
- Sugar process data [32]: The dataset contains 265 samples that can be arranged in an IJK three-order tensor of size $265 \times 571 \times 7$. The first mode relates to samples, the second to emission wavelengths, and the third to excitation wavelengths. The (ijk)-th element of this tensor, $\mathcal{X}(i, j, k)$, represents the measured emission intensity from sample i, excited at wavelength k, and measured at wavelength j. This dataset is modeled by a rank-4 CP model.
- Dorrit fluorescence data [15]: Fluorescence spectrometer was used to measure 27 synthetic samples containing different concentrations of four analytes (hydroquinone, tryptophan, phenylalanine and dopa). Each fluorescence landscape corresponds to 233 emission wavelengths and

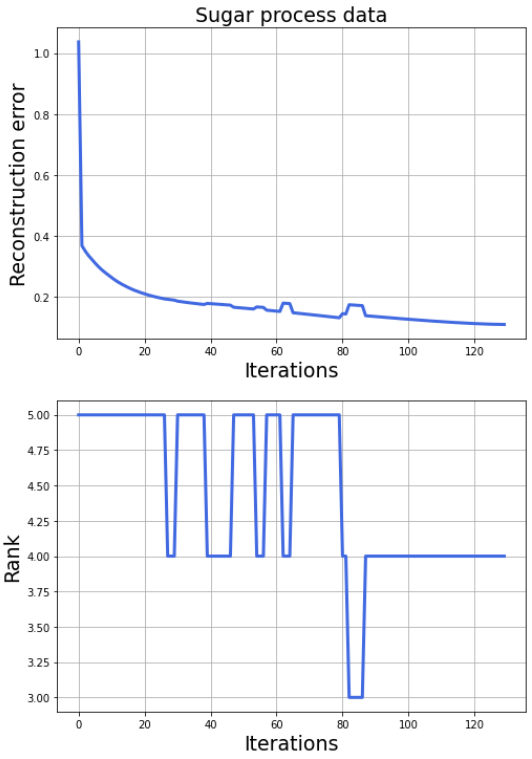

FIGURE 9. Convergence loss of FARAC (top) and rank estimation (bottom) on the sugar process dataset over iterations .

24 excitation wavelengths. A rank-4 CP model is used to model this dataset, which is represented as a tensor of size $27 \times 233 \times 24$.

B. RESULTS

1) Synthetic data

- Accuracy of estimated ranks using the proposed approach is illustrated in Fig. 2. We can clearly see that the FARAC approach can estimate the exact canonical rank while being robust to the choise of the threshold parameter.
- Fig. 3 depicts the convergence curves of the proposed method in terms of the Reconstruction Square Error (RSE) at each iteration t, which is given by:

$$\frac{||\mathcal{X}_{\text{noise}} - \mathcal{X}^{(t)}||_F}{||\mathcal{X}_{\text{noise}}||_F}.$$

  We can see that the reconstruction error quickly decreases to zero w.r.t iterations.
- The recovery of the CP factors is shown in Fig. 4 by checking the subspace error. This can be done by computing the principal angle [36] between the true subspaces and the estimated ones at each iteration t:

$$\theta^{(t)} = \arcsin\left(\sqrt{2}||U_n U_n^{\dagger} - \hat{U}_n^{(t)}\hat{U}_n^{(t)\dagger}||_F\right),$$

  where U_n and $\hat{U}_n^{(t)}$ are the exact and the estimated factors at iteration t, respectively. We see that the principal angle between the subspaces of the three CP factors converges to a low value with respect to iterations.
- In Fig. 5, we compare the FARAC method with CORCONDIA with respect to SNR values. We can see that FARAC clearly outperforms the state-of-the-art method for large range values of SNRs.
- In Table 1, we show the accuracy of the rank estimation using FARAC when the true rank exceeds one of the dimensions of the input tensor. As we can see, FARAC can handle this difficult scenario for a large range of SNR values. The CORCONDIA approach, on the other hand, is ineffective in that situation [17].
- According to Fig. 6, at very low SNR, FARAC tends to overestimate the true rank. In contrast, rank estimation using the CORCONDIA method becomes a very difficult task at low SNR according to Fig. 7.

Compared with the state-of-the-art method, FARAC shows very good performance in terms of the accuracy of rank estimation for large range of values of SNRs while being robust to the choise of the threshold parameter. FARAC also handles the difficult case where the rank is bigger than one of the dimensions of the input tensor.

2) Real datasets

- In Fig. 8 and 9, we present the convergence loss of FARAC, as well as the rank estimation over iterations on the amino acid fluorescence, the dorrit fluorescence and the sugar process datasets. As shown in these figures, the rank is well estimated for the three real datasets. The threshold parameter is selected so that the curve of the reconstruction loss exhibits good convergence properties. Based on the noise level, one can use a grid search over the range of values $[0.1, 0.01, 0.001]$. It is important to note that the FARAC method is robust to the choice of the threshold as shown in Fig. 2. For the amino and dorrit fluorescence datasets, the threshold parameter used is equal to 0.01; for the sugar process dataset, it is equal to 0.001.

VII. CONCLUSION

We have addressed in this work the challenging problem of the canonical rank estimation by defining it as a convex optimization problem. The proposed method, called FARAC jointly estimates the canonical rank and the CP factors. We have compared FARAC to the well-known CORCONDIA method and found that FARAC is much more accurate. We have also shown that FARAC shows strong robustness to the choice of the threshold parameter and can handle the difficult case when the rank exceeds one of the dimensions of the tensor, unlike the CORCONDIA method. Last but not least, we illustrate the efficiency of the FARAC method on real datasets.

A. DETAIL COMPUTATIONS FOR THE DERIVATION OF FARAC

Let us recall the Lagrangien of the problem (4) that we want to minimize w.r.t $(U_n)_n$, $\tilde{\mathcal{G}}$ and λ:

$$\mathcal{L}_{\mathcal{G},\{U_n\}} = \gamma_1||\lambda||_1+\frac{\gamma_2}{2}\left(\left|\left|\mathcal{X}-\mathcal{G}\overset{N}{\underset{n=1}{\times_n}}U_n\right|\right|_F^2-\epsilon_1\right)+\frac{\gamma_3}{2}\left(\left|\left|\tilde{\mathcal{G}}\right|\right|_F^2-\epsilon_2\right),$$

- We denote by $\mathcal{L}_{U_n}$, the part of $\mathcal{L}_{\mathcal{G},\{U_n\}}$ which depends only on U_n.

$$\mathcal{L}_{U_n} = \frac{\gamma_2}{2}\left|\left|\mathcal{X}-\mathcal{G}\overset{N}{\underset{n=1}{\times_n}}U_n\right|\right|_F^2 \tag{12}$$

The matricized form of eq. (12) is given by:

$$\mathcal{L}_{U_n} = \left|\left|X_{(n)}-U_nG_{(n)}\left(\bigotimes_{\substack{l=N\\l\neq n}}^{l=1}U_l^T\right)\right|\right|_F^2,$$

In contrast to the derivation of a classical CPD, $G_{(n)}$ is not the identity matrix.
The gradient of $\mathcal{L}_{U_n}$ w.r.t U_n can be calculated as in a traditional matrix linear regression problem:

$$\nabla_{U_n}(\mathcal{L}_{\mathcal{G},\{U_n\}}) = \gamma_2\left(-X_{(n)}\left(\bigotimes_{\substack{l=N\\l\neq n}}^{l=1}U_l\right)G_{(n)}^T+U_n\left(\bigotimes_{\substack{l=N\\l\neq n}}^{l=1}U_l^TU_l\right)\right).$$

We set $\nabla_{U_n}(\mathcal{L}_{\mathcal{G},\{U_n\}})$ to 0 and get the exact formula of U_n:

$$U_n = X_{(n)}\left(\bigotimes_{\substack{l=N\\l\neq n}}^{l=1}U_l\right)G_{(n)}^T\left(\bigotimes_{\substack{l=N\\l\neq n}}^{l=1}\left(U_l^TU_l\right)^{-1}\right).$$

- The part of $\mathcal{L}_{\mathcal{G},\{U_n\}}$ that depends only on $\tilde{\mathcal{G}}$ is denoted by $\mathcal{L}_{\tilde{\mathcal{G}}}$:

$$\mathcal{L}_{\tilde{\mathcal{G}}} = \frac{\gamma_2}{2}\sum_{i_1,\ldots,i_N}\underbrace{\left(\mathcal{X}(i_1,\ldots,i_N)-\left(\mathcal{G}\overset{N}{\underset{n=1}{\times_n}}U_n\right)(i_1,\ldots,i_N)\right)^2}_{\mathcal{A}^2(i_1,\ldots,i_N)}+\frac{\gamma_3}{2}\sum_{r_1,\ldots,r_N}\tilde{\mathcal{G}}^2(r_1,\ldots,r_N)$$

Let us derive $\mathcal{L}_{\tilde{\mathcal{G}}}$ with respect to $\tilde{\mathcal{G}}(r_1,\ldots,r_N)$ such that $r_1\neq r_2$ or $\ldots$ or $r_{N-1}\neq r_N$ (diagonal elements are excluded since they are equal to 0):

$$[\nabla_{\tilde{\mathcal{G}}}\mathcal{L}_{\tilde{\mathcal{G}}}](r_1,\ldots,r_N) = -\gamma_2\sum_{i_1,\ldots,i_N}\mathcal{A}(i_1,\ldots,i_N)\left(\prod_{n=1}^{N}U_n(i_n,r_n)\right)+\gamma_3\tilde{\mathcal{G}}(r_1,\ldots,r_N). \tag{13}$$

- We want to derive $\mathcal{L}_{\mathcal{G},\{U_n\}}$ w.r.t λ. Let us first rewrite it as follows:

$$\begin{aligned}\mathcal{L}_{\mathcal{G},\{U_n\}} &= \gamma_1||\lambda||_1+\frac{\gamma_2}{2}\left|\left|\mathcal{X}-\left(\tilde{\mathcal{G}}+diag(\lambda)\right)\overset{N}{\underset{n=1}{\times_n}}U_n\right|\right|_F^2+\frac{\gamma_3}{2}\left|\left|\tilde{\mathcal{G}}\right|\right|_F^2,\\ &= \gamma_1||\lambda||_1+\frac{\gamma_2}{2}\left|\left|\left(\mathcal{X}-\tilde{\mathcal{G}}\overset{N}{\underset{n=1}{\times_n}}U_n\right)-\sum_{r=1}^{R_0}\lambda_r\overset{N}{\underset{n=1}{\bigcirc}}u_{n,r}\right|\right|_F^2+\frac{\gamma_3}{2}\left|\left|\tilde{\mathcal{G}}\right|\right|_F^2\end{aligned}$$

- We denote by $\mathcal{L}_\lambda$, the part of $\mathcal{L}_{\mathcal{G},\{U_n\}}$ which depends only on λ. $\mathcal{L}_\lambda$ is given as follows:

$$\begin{aligned}\mathcal{L}_\lambda &= \gamma_1||\lambda||_1+\frac{\gamma_2}{2}\left(\left|\left|\sum_{r=1}^{R_0}\lambda_r\overset{N}{\underset{n=1}{\bigcirc}}u_{n,r}\right|\right|_F^2-2\left\langle\mathcal{X}-\tilde{\mathcal{G}}\overset{N}{\underset{n=1}{\times_n}}U_n,\sum_{r=1}^{R_0}\lambda_r\overset{N}{\underset{n=1}{\bigcirc}}u_{n,r}\right\rangle+\left|\left|\mathcal{X}-\tilde{\mathcal{G}}\overset{N}{\underset{n=1}{\times_n}}U_n\right|\right|_F^2\right)\\ &= \gamma_1\left(|\lambda_l|+\sum_{r\neq l}|\lambda_r|\right)+\frac{\gamma_2}{2}\left(\left|\left|\sum_{\substack{r=1\\r\neq l}}^{R_0}\lambda_r\overset{N}{\underset{n=1}{\bigcirc}}u_{n,r}\right|\right|_F^2+\lambda_l^2\left|\left|\overset{N}{\underset{n=1}{\bigcirc}}u_{n,l}\right|\right|_F^2+2\left\langle\lambda_l\overset{N}{\underset{n=1}{\bigcirc}}u_{n,l},\sum_{\substack{r=1\\r\neq l}}^{R_0}\lambda_r\overset{N}{\underset{n=1}{\bigcirc}}u_{n,r}\right\rangle\right.\\ &\quad-2\left\langle\mathcal{X}-\tilde{\mathcal{G}}\overset{N}{\underset{n=1}{\times_n}}U_n,\sum_{\substack{r=1\\r\neq l}}^{R_0}\lambda_r\overset{N}{\underset{n=1}{\bigcirc}}u_{n,r}\right\rangle-2\left\langle\mathcal{X}-\tilde{\mathcal{G}}\overset{N}{\underset{n=1}{\times_n}}U_n,\lambda_l\overset{N}{\underset{n=1}{\bigcirc}}u_{n,l}\right\rangle+\left.\left|\left|\mathcal{X}-\tilde{\mathcal{G}}\overset{N}{\underset{n=1}{\times_n}}U_n\right|\right|_F^2\right).\end{aligned} \tag{14}$$

Let us derive $\mathcal{L}_\lambda$ with respect to λ_l and set it to 0.

$$\begin{aligned}
&\nabla_{\lambda_l}(\mathcal{L}_\lambda) = 0 \\
&\Leftrightarrow \gamma_1 \frac{\partial|\lambda_l|}{\lambda_l} + \gamma_2 \Bigg(\lambda_l \left\| \bigcirc_{n=1}^{N} u_{n,l} \right\|_F^2 \\
&+ \left\langle \bigcirc_{n=1}^{N} u_{n,l}, \sum_{\substack{r=1 \\ r\neq l}}^{R_0} \lambda_r \bigcirc_{n=1}^{N} u_{n,r} \right\rangle \\
&- \left\langle \mathcal{X} - \tilde{\mathcal{G}} \times_{n=1}^{N} U_n, \bigcirc_{n=1}^{N} u_{n,l} \right\rangle \Bigg) = 0 \\
&\Leftrightarrow \lambda_l + \frac{\gamma_1}{\gamma_2 \| \bigcirc_{n=1}^{N} u_{n,l} \|^2} \frac{\partial|\lambda_l|}{\partial\lambda_l} = \\
&- \gamma_2 \left\langle \bigcirc_{n=1}^{N} u_{n,l}, \sum_{\substack{r=1 \\ r\neq l}}^{R_0} \lambda_r \bigcirc_{n=1}^{N} u_{n,r} \right\rangle \\
&+ \gamma_2 \left\langle \mathcal{X} - \tilde{\mathcal{G}} \times_{n=1}^{N} U_n, \bigcirc_{n=1}^{N} u_{n,l} \right\rangle . \quad (15)
\end{aligned}$$

Furthermore, we have the following:

$$\left\| \bigcirc_{n=1}^{N} u_{n,l} \right\|^2 = \prod_{n=1}^{N} \|u_{n,l}\|^2 .$$

Since $u_{n,l}$ are unit vectors, we have $\left\| \bigcirc_{n=1}^{N} u_{n,l} \right\|^2 = 1$. Hence we find the following:

$$\begin{aligned}
\lambda_l + \frac{\gamma_1}{\gamma_2} \frac{\partial|\lambda_l|}{\partial\lambda_l} = &- \left\langle \bigcirc_{n=1}^{N} u_{n,l}, \sum_{\substack{r=1, \\ r\neq l}}^{R_0} \lambda_r \bigcirc_{n=1}^{N} u_{n,r} \right\rangle + \\
&\left\langle \mathcal{X} - \tilde{\mathcal{G}} \times_{n=1}^{N} U_n, \bigcirc_{n=1}^{N} u_{n,l} \right\rangle
\end{aligned}$$

As a result, the value of λ_l can be computed using the soft threshholding operator $S_{\frac{\gamma_1}{\gamma_2}}$ [34]:

$$\begin{aligned}
\lambda_l = S_{\frac{\gamma_1}{\gamma_2}} \Bigg(&- \left\langle \bigcirc_{n=1}^{N} u_{n,l}, \sum_{\substack{r=1 \\ r\neq l}}^{R_0} \lambda_r \bigcirc_{n=1}^{N} u_{n,r} \right\rangle + \\
&\left\langle \mathcal{X} - \tilde{\mathcal{G}} \times_{n=1}^{N} U_n, \bigcirc_{n=1}^{N} u_{n,l} \right\rangle \Bigg)
\end{aligned}$$

B. CONVEXITY

In this section, we will demonstrate that the Lagrangien in eq. (14) is convex w.r.t to λ and $\tilde{\mathcal{G}}$ so that the global minimum will be reached. To do that, we will show that the hessian w.r.t λ and $\tilde{\mathcal{G}}$ are positive semi-definite matrices.

- Convexity w.r.t λ:
  Let H_λ be the hessian of $\mathcal{L}_{\mathcal{G},\{U_n\}}$ w.r.t λ. Using the gradient computed in eq. (15), we have:

$$H_\lambda(s,l) := \frac{\partial^2 \mathcal{L}_{\mathcal{G},\{U_n\}}}{\partial\lambda_s \partial\lambda_l} = \gamma_2 \left\langle \bigcirc_{n=1}^{N} u_{n,l}, \bigcirc_{n=1}^{N} u_{n,s} \right\rangle .$$

  Let $x \in \mathbb{R}^{R_0}$,

$$\begin{aligned}
x^T H_\lambda x &= \gamma_2 \sum_{l,s} x_l \, x_s \left\langle \bigcirc_{n=1}^{N} u_{n,l}, \bigcirc_{n=1}^{N} u_{n,s} \right\rangle \\
&= \gamma_2 \left\langle \sum_l x_l \bigcirc_{n=1}^{N} u_{n,l}, \sum_s x_s \bigcirc_{n=1}^{N} u_{n,s} \right\rangle \\
&= \gamma_2 \left\| \sum_l x_l \bigcirc_{n=1}^{N} u_{n,l} \right\|^2 \geq 0,
\end{aligned}$$

  since γ_2 is a positive Lagrange multiplier.
- Convexity w.r.t $\tilde{\mathcal{G}}$:
  We first place the elements $\tilde{\mathcal{G}}(r_1, \ldots, r_N)$ in a vector $x \in \mathbb{R}^{{R_0}^N - R_0}$ and use the same method as for λ.
  Let $H_{\tilde{\mathcal{G}}}$ be the hessian of $\mathcal{L}_{\mathcal{G},\{U_n\}}$ w.r.t $\tilde{\mathcal{G}}$. Using (13), we compute the second derivates of $\mathcal{L}_{\mathcal{G},\{U_n\}}$. The following derivatives are done only on the offdiagonals of $\tilde{\mathcal{G}}$ since its diagonal is zero.

$$\begin{aligned}
&H_{\tilde{\mathcal{G}}}(r_1 \ldots r_N, r'_1 \ldots r'_N) := \\
&\frac{\partial^2 \mathcal{L}_{\mathcal{G},\{U_n\}}}{\partial\tilde{\mathcal{G}}(r_1, \ldots, r_N) \partial\tilde{\mathcal{G}}(r'_1, \ldots, r'_N)} \\
&= \gamma_2 \sum_{i_1,\ldots,i_N} \left[\prod_{n=1}^{N} U_n(i_n, r'_n) \right] \left[\prod_{n=1}^{N} U_n(i_n, r_n) \right] + \gamma_3 .
\end{aligned}$$

  Let $x \in \mathbb{R}^{{R_0}^N - R_0}$,

$$\begin{aligned}
&x^T H_{\tilde{\mathcal{G}}} x = \\
&\gamma_2 \sum_{i_1,\ldots,i_N} \underbrace{\sum_{r_n} \left[\prod_{n=1}^{N} U_n(i_n, r_n) \right]}_{\mathcal{Z}(i_1,\ldots,i_N)} \underbrace{\sum_{r'_n} \left[\prod_{n=1}^{N} U_n(i_n, r'_n) \right]}_{\mathcal{Z}(i_1,\ldots,i_N)} \\
&\qquad + \gamma_3 x^T x \\
&= \gamma_2 ||\mathcal{Z}||^2 + \gamma_3 ||x||^2 \geq 0,
\end{aligned}$$

  since γ_2 and γ_3 are strictly positive values.

REFERENCES

[1] A.Bhaskara, M.Charikar, and A.Vijayaraghavan. Uniqueness of tensor decompositions with applications to polynomial identifiability. In Conference on Learning Theory, pages 742–778. PMLR, 2014.

[2] A.Cichocki, D.Mandic, L.De Lathauwer, G.Zhou, Q.Zhao, C.Caiafa, and H.Phan. Tensor decompositions for signal processing applications: From two-way to multiway component analysis. IEEE Signal Processing Magazine, 32(2):145–163, 2015.

[3] A.Cichocki, N.Lee, I.Oseledets, H.Phan, Q.Zhao, and D.Mandic. Tensor networks for dimensionality reduction and large-scale optimization: Part 1 low-rank tensor decompositions. Found. Trends Mach. Learn., 9(4–5):249–429, dec 2016.

[4] C.Hillar and L.Lim. Most tensor problems are np-hard. Journal of the ACM (JACM), 60(6):1–39, 2013.

[5] D.Geand, X.Jiang, and Y.Ye. A note on the complexity of l p minimization. Math. Program., 129(2):285–299, oct 2011.

[6] D.Guimaraes and R.De Souza. Simple and efficient algorithm for improving the mdl estimator of the number of sources. Sensors, 14(10):19477–19492, 2014.

[7] D.Kingma and J.Ba. Adam: A method for stochastic optimization. In Yoshua Bengio and Yann LeCun, editors, 3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings, 2015.

[8] E.Papalexakis and C.Faloutsos. Fast efficient and scalable core consistency diagnostic for the parafac decomposition for big sparse tensors. In 2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pages 5441–5445, 2015.

[9] M.Abadi et.al. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. Software available from tensorflow.org.

[10] F.Cong, Q.Lin, L.Kuang, X.Gong, P.Astikainen, and T.Ristaniemi. Tensor decomposition of eeg signals: a brief review. Journal of neuroscience methods, 248:59–69, 2015.

[11] G.Bergqvist and E.Larsson. The higher-order singular value decomposition: Theory and an application [lecture notes]. IEEE Signal Processing Magazine, 27(3):151–154, 2010.

[12] I.Oseledets. Tensor-train decomposition. SIAM Journal on Scientific Computing, 33(5):2295–2317, 2011.

[13] J.Denis and T.Dhorne. Orthogonal Tensor Decomposition of 3-Way Tables, page 31–37. North-Holland Publishing Co., NLD, 1989.

[14] J.Kruskal. Three-way arrays: rank and uniqueness of trilinear decompositions, with application to arithmetic complexity and statistics. Linear Algebra and its Applications, 18(2):95 – 138, 1977.

[15] J.Riu and R.Bro. Jack-knife technique for outlier detection and estimation of standard errors in parafac models. Chemometrics and Intelligent Laboratory Systems, 65(1):35–49, 2003.

[16] J.Sadecka and J.Tothova. Fluorescence spectroscopy and chemometrics in the food classification - a review. Czech Journal of Food Sciences, 25:159–173, 01 2007.

[17] K.Liu, J.Da Costa, H.So, L.Huang, and J.Ye. Detection of number of components in candecomp/parafac models via minimum description length. Digital Signal Processing, 51:110–123, 2016.

[18] L.De Lathauwer, B.De Moor, and J.Vandewalle. A multilinear singular value decomposition. SIAM journal on Matrix Analysis and Applications, 21(4):1253–1278, 2000.

[19] L.Cheng, X.Tong, and Y.Wu. Distributed nonnegative tensor canonical polyadic decomposition with automatic rank determination. In 2020 IEEE 11th Sensor Array and Multichannel Signal Processing Workshop (SAM), pages 1–5. IEEE, 2020.

[20] L.Cheng, Y.Wu, and H.Poor. Scaling probabilistic tensor canonical polyadic decomposition to massive data. IEEE Transactions on Signal Processing, 66(21):5534–5548, 2018.

[21] L.Cheng, Z.Chen, Q.Shi, Y.Wu, and S.Theodoridis. Towards flexible sparsity-aware modeling: Automatic tensor rank learning using the generalized hyperbolic prior. IEEE Transactions on Signal Processing, 70:1834–1849, 2022.

[22] L.Yuan, C.Li, D.Mandic, J.Cao, and Q.Zhao. Tensor ring decomposition with rank minimization on latent space: An efficient approach for tensor completion. In Proceedings of the AAAI Conference on Artificial Intelligence, volume 33, pages 9151–9158, 2019.

[23] M.Mørup and L.Hansen. Automatic relevance determination for multi-way models. Journal of Chemometrics, 23:352 – 363, 07 2009.

[24] M.Sørensen and L.De Lathauwer. Blind signal separation via tensor decomposition with vandermonde factor: Canonical polyadic decomposition. IEEE Transactions on Signal Processing, 61(22):5507–5519, 2013.

[25] M.Zhou, Y.Liu, Z.Long, L.Chen, and C.Zhu. Tensor rank learning in cp decomposition via convolutional neural network. Signal Processing: Image Communication, 73:12–21, 2019.

[26] N.Sidiropoulos, G.Giannakis, and R.Bro. Blind parafac receivers for ds-cdma systems. IEEE Transactions on Signal Processing, 48(3):810–823, 2000.

[27] N.Sidiropoulos, R.Bro, and G.Giannakis. Parallel factor analysis in sensor array processing. Trans. Sig. Proc., 48(8):2377–2388, aug 2000.

[28] E. Papalexakis, C.Faloutsos, and N.Sidiropoulos. Tensors for data mining and data fusion: Models, applications, and scalable algorithms. ACM Trans. Intell. Syst. Technol., 8(2), October 2016.

[29] P.Comon. Tensor decompositions. Mathematics in signal processing V, pages 1–24, 2002.

[30] Q.Shi, H.Lu, and Y.Cheung. Tensor rank estimation and completion via cp-based nuclear norm. In Proceedings of the 2017 ACM on Conference on Information and Knowledge Management, pages 949–958, 2017.

[31] R.Bro. Parafac. tutorial and applications. Chemometrics and intelligent laboratory systems, 38(2):149–171, 1997.

[32] R.Bro. Exploratory study of sugar production using fluorescence spectroscopy and multi-way analysis. Chemometrics and Intelligent Laboratory Systems, 46(2):133–147, 1999.

[33] R.Rasmus and H.Kiers. A new efficient method for determining the number of components in parafac models. Journal of Chemometrics, 17(5):274–286, May 2003.

[34] R.Tibshirani. Regression shrinkage and selection via the lasso. Journal of the Royal Statistical Society. Series B (Methodological), 58(1):267–288, 1996.

[35] S.Boyd and L.Vandenberghe. Convex optimization. Cambridge university press, 2004.

[36] S.Jayasumana, R.Hartley, M.Salzmann, H.Li, and M.Harandi. Kernel methods on riemannian manifolds with gaussian rbf kernels. IEEE transactions on pattern analysis and machine intelligence, 37(12):2464–2477, 2015.

[37] T.Kolda and B.Bader. Tensor decompositions and applications. SIAM Review, 51(3):455–500, September 2009.

[38] W.Guo, I.Kotsia, and I.Patras. Tensor learning for regression. IEEE Transactions on Image Processing, 21(2):816–827, 2012.

[39] X.Gong, Q.Lin, F.Cong, and L.De Lathauwer. Double coupled canonical polyadic decomposition for joint blind source separation. IEEE Transactions on Signal Processing, 66(13):3475–3490, 2018.

[40] X.Han, L.Albera, A.Kachenoura, L.Senhadji, and H.Shu. Low rank canonical polyadic decomposition of tensors based on group sparsity. In 2017 25th European Signal Processing Conference (EUSIPCO), pages 668–672, 2017.

[41] Y.Ji, Q.Wang, X.Li, and J.Liu. A survey on tensor techniques and applications in machine learning. IEEE Access, 7:162950–162990, 2019.

Karmouda Ouafae receives a Master's degree in Applied Mathematics from the Faculty of Science and Technologies of Fes, Morocco in 2018. She receives a Master degree in Data Science from the Aix-Marseille University in 2019. Currently, she is a third year PhD student and is a student member IEEE. Her research interests focus on developing/improving Machine Learning Algorithms for multi-dimensional data (tensors). She is particularly interested in kernel methods and Deep Learning techniques for high-dimensional data.

Jeremie Boulanger received his PhD degree from the laboratory Gipsa (Grenoble, France) and the University of Melbourne (Melbourne, Australia) in 2013. From 2013 to 2015, he was a postdoctoral at the laboratory CETAPS (Rouen, France). He is currently an assistant professor at the University of Lille (Lille, France) at the laboratory Cristal (Lille, France).

Remy Boyer received the M.Sc. and Ph.D degrees from the Ecole Nationale Supérieure des Telecommunications (ENST-Paris or Télécom ParisTech) in 1999 and 2002, respectively, in statistical signal processing. From 2002 to 2003, he was a postdoctoral fellow during six months at Sherbrooke University (Canada). From 2003 to 2018, he is an associate professor at Université Paris-Sud - Laboratory of Signals and Systems (L2S). From 2011 to 2012, Remy Boyer was a visiting researcher at the SATIE Laboratory (Ecole Normale Supérieure de Cachan) and at the University of Aalborg (Danemark). Remy Boyer received an "Habilitation à Diriger des Recherches (HDR)" from the Université Paris-Sud in December 2012. He is currently a full professor with University of Lille (faculty of Sciences and Technologies) in computer science department and at laboratory CRIStAL. Remy Boyer conducts research in mathematical methods for signal processing (Bayesian signal processing, tensor-based signal processing and machine leaning). Remy Boyer is an IEEE Senior member and a member of the IEEE TC Sensor Array and Multichannel (SAM) in charge of the subcommittee SAP, EURASIP TAC SPMuS (Signal Processing for Multisensor Systems and EURASIP TAC TMTSP ("Theoretical and Methodological Trends in Signal Processing »). Remy Boyer is the elected secretary of the French chapter of the IEEE Signal Processing Society. He is an Associate Editor in ELSEVIER Signal Processing and Senior Area Editor for IEEE Trans. On Signal Processing. He has published over 180 peer reviewed articles.