This article has been accepted for publication in IEEE Access. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2022.3189793

IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Digital Object Identifier

Joint Factors And Rank Canonical Estimation For The 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 while the frobenius norm of the offdiagonal terms is constrained to be bounded. We propose an alternated minimization strategy of the Lagrangien to solve the proposed optimization problem. Compared to the state-of-the-art method, FARAC shows very good performance in terms of accuracy of the rank estimation for large range of values of SNRs. In addition, FARAC can handle the case where the canonical rank exceeds one of the dimensions of the input tensor. Several numerical experiments are conducted in the presence of various levels of noise.

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 to be efficiently described [2], [7], [21], [29]. 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], [8], [14], [15], [22].
The most popular tensor decomposition is the Canonical Polyadic Decomposition (CPD) [13]. The CPD is widespreadly used in differents fields such as as chemometrics, telecommunications, blind source separation [11], [17]–[19], [27]. The CPD is particularly attractive due to its uniqueness property [1], [10]. However, the CPD requires the knowledge of the rank. Unfortunately, the problem of determining the canonical rank is NP-hard [5].
In the present study, we propose estimating both of 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 present our proposed approach and describe our algorithm for solving it. As a final step, we perform several numerical experiments in section VI to evaluate and compare our proposed approach with the CORCONDIA method.

II. NOTATIONS AND ALGEBRAIC BACKGROUND

In this section, we recall some algebraic definitions on tensor algebra from [13]:

This article has been accepted for publication in IEEE Access. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2022.3189793

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 $I_1 \times \ldots I_{n-1} \times J \times \cdots \times I_N$ and is given for instance 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 the ones in its superdiagonal *i.e.*: $\mathcal{X}(i_1, \ldots, i_N) \neq 0$ 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 $X_{(n)}$ where its 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:

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

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. More formally, for a tensor $\mathcal{X} \in \mathbb{R}^{I_1 \times \cdots \times I_N}$, the CPD 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:

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

Definition 9: (Tensor rank) The rank R of a tensor is defined as the smallest number of components in an exact CPD.
For a 3-order tensor $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times I_3}$, we have the following upper bound of its maximum rank (the largest attainable rank) [10]:

$$R \leq \min\{I_1 I_2, I_1 I_3, I_2 I_3\}.$$

III. RELATED WORKS

Existing approaches for the canonical rank detection include:

- The CORe CONsistency DIAgnostic (CORCONDIA) [23] is a heuristic method to detect 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 a simple approach. However, its computation is prohibitive even for small tensors [20]. 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 is presented in [20]. It proposes an efficient algorithm for computing the CORCONDIA diagnostic which exploits data sparsity and scales well as the tensor size increases. In cases where either the tensor or the factors or both are sparse [20], 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) [16] is a bayesian approach applied for the Tucker and CP model. 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 turns excess of components and simplify the core structure.
- Detection of number of components in CANDECOMP models via minimum description length (N-D MDL) [12] 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

This article has been accepted for publication in IEEE Access. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2022.3189793

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 [12].

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 achieved by minimizing the core tensors' superdiagonal using the l_0 norm as an objective function, adding a constraint on the reconstruction error and an additional 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:

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

where $\mathrm{diag}(\lambda)$ is a diagonal tensor with λ in its superdiagonal. 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 [4], [9]. However, since minimizing the l_0 norm is NP-hard [28], 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:

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

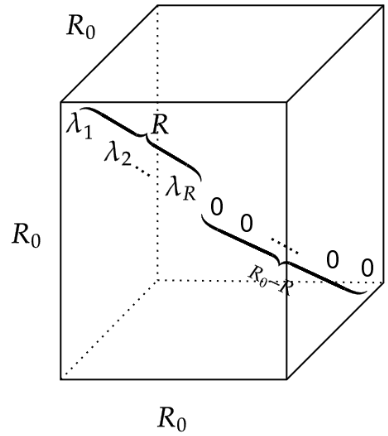

FIGURE 1. CP core tensor $\mathcal{G}$ of a rank-R tensor of order 3 and size R_0 with $\lambda_1 > \cdots > \lambda_R > 0$.

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

where γ_2 and γ_3 are two strictly positive Lagrange multipliers. Following the Lagrangien method [24], $\mathcal{L}_{\mathcal{G},\{U_n\}}$ is minimized w.r.t $\{U_n\}_n$, $\tilde{\mathcal{G}}$ and λ. To do that, we will proceed iteratively. At each iteration, we first minimize $\mathcal{L}_{\mathcal{G},\{U_n\}}$ w.r.t the n-th CP factor U_n assuming the remaining factors and $\mathcal{G}$ are known. This is a classical linear regression problem. Next, 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 [6]. As a final step, 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 [26] using the current updates of $\tilde{\mathcal{G}}$ and $\{U_n\}_n$. We provide in the following, 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 $\mathcal{G}$. Details of the computations can be found in Appendix A.

- CP factors U_n:

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

- Gradient of $\mathcal{L}_{\mathcal{G},\{U_n\}}$ w.r.t the offdiagonals of $\mathcal{G}$ *i.e*

This article has been accepted for publication in IEEE Access. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2022.3189793

$\mathcal{G}(r_1,\ldots,r_N)$ such that $r_1 \neq r_2$ or $\ldots$ or $r_{N-1} \neq r_N$:

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

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

- Formula to update the superdiagonal of $\mathcal{G}$:

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

where $S_{\frac{\gamma_1}{\gamma_2}}$ is the soft thresholding operator [26] 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 analysis of the Algorithm 1 by taking into consideration the svds in the initialization and the main parts of the algorithm as the computations of the different gradients and the shrinkage operator. The complexity is evaluated as follows:

$$\mathcal{O}\Bigg(\left(\sum_{n=1}^{N} I_n\right) R_0^2 + T\Bigg[N + \left(\sum_{n=1}^{N} I_n\right) R_0 + \left(\prod_{n=1}^{N} I_n\right)\left(N + R_0 + NR_0^{N-1}\right)\Bigg]\Bigg), \quad (10)$$

where T is the number of iterations.

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

$$\mathcal{O}\left(IR_0^2 + T\left(N + IR_0 + I(N + R_0 + NR_0^{N-1})\right)\right).$$

VI. NUMERICAL EXPERIMENTS

A. PROTOCOL EXPERIMENT

Using the CP model, we create a synthetic rank-R real valued tensor $\mathcal{X}$ of order N. The CP factors are derived from a single normal standard distribution. $\mathcal{B}$ is a zero mean, unit variance and white noise added so that we use different values of SNR (Signal to Noise Ratio) ranging from 0db to 40db. Our noisy data are then obtained by:

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

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):

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

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]

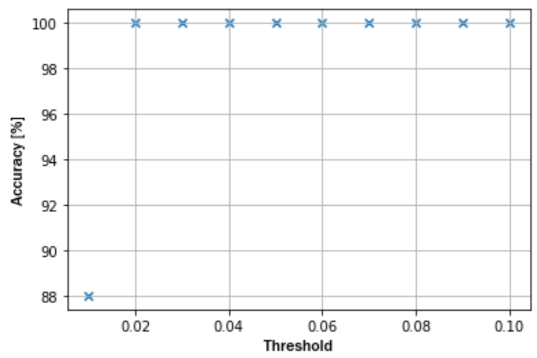

FIGURE 2. Accuracy of rank estimation of a tensor of size $5 \times 5 \times 5$ w.r.t the threshhold parameter. The true rank is $R = 2$ and $R_0 = 5$.

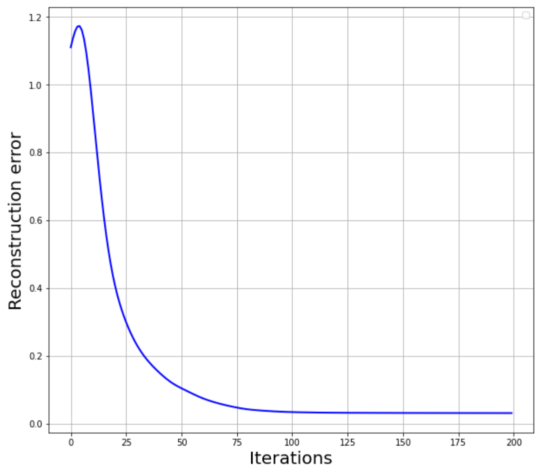

FIGURE 3. This figure shows the convergence curve of the mean reconstruction error using the Relative Square Error (RSE) along iterations using FARAC. We have used a noisy tensor $\mathcal{X}_{\text{noise}}$ of size $5 \times 5 \times 5$ with SNR = $25db$. The threshold paramter is equal to 0.02.

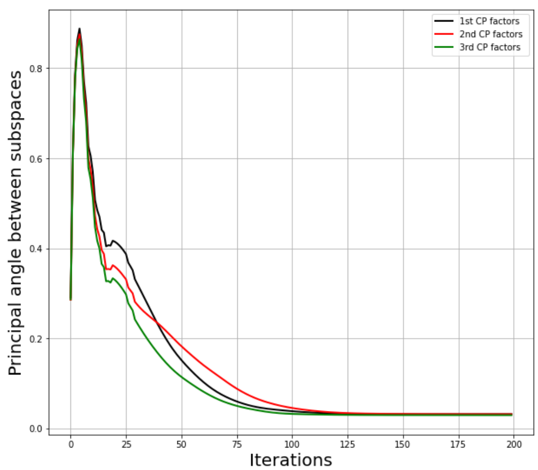

FIGURE 4. This figure shows the mean principal angle between the three subspaces spanned by CP factors along iterations using our method. We have used a noisy tensor $\mathcal{X}_{\text{noise}}$ of size $5 \times 5 \times 5$ with SNR = $25db$. The threshold parameter is equal to 0.02.

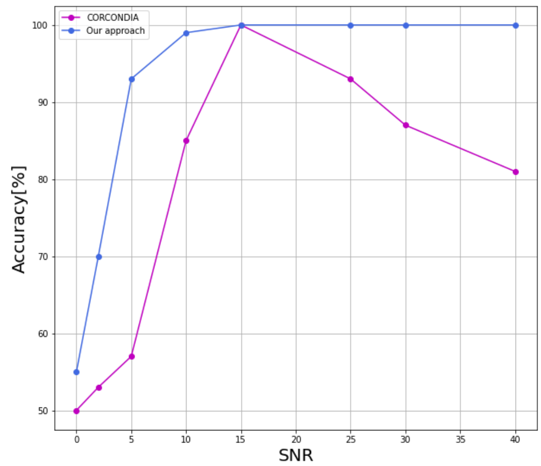

FIGURE 5. Accuracy of FARAC Vs. CORCONDIA w.r.t SNR for a tensor of a noisy tensor $\mathcal{X}_{\text{noise}}$ with SNR values ranging from 0db to 40db. $\mathcal{X}_{\text{noise}}$ is of size $5 \times 5 \times 5$. Different CP models with ranks ranging from 1 to 5 are used to fit CORCONDIA. The true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

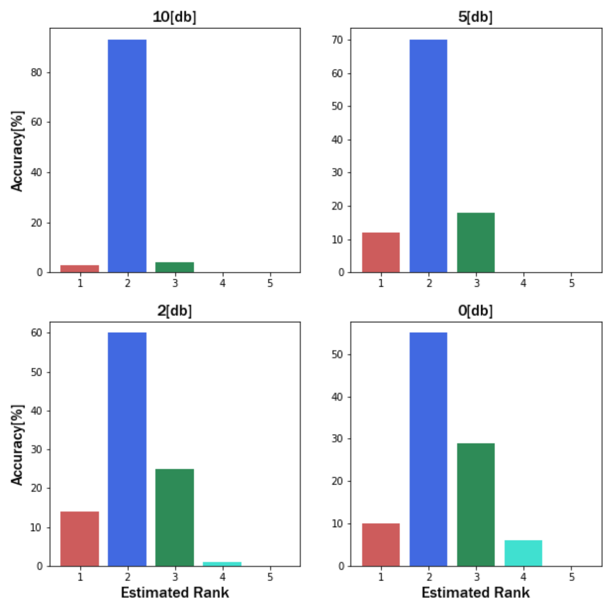

FIGURE 6. Accuracy of rank estimation using FARAC of a noisy tensor $\mathcal{X}_{\text{noise}}$ with different low SNR values ranging from 0db to 10db. $\mathcal{X}_{\text{noise}}$ is of size $5 \times 5 \times 5$. The large bound of rank used is $R_0 = 5$ and the true rank is $R = 2$.

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 $R_0 = 7$.

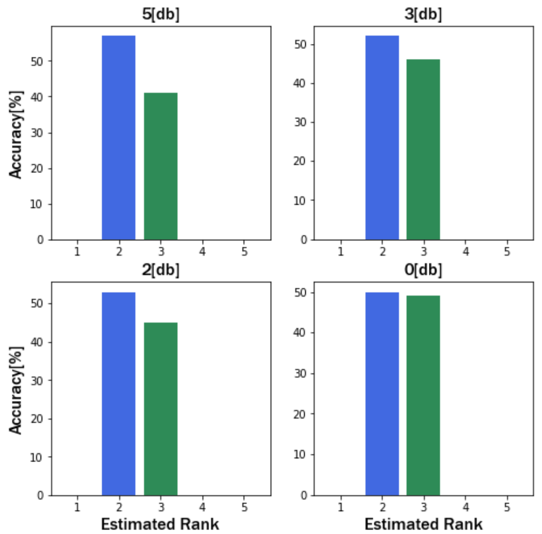

FIGURE 7. Accuracy of the CORCONDIA approach for a noisy tensor $\mathcal{X}_{\text{noise}}$ with SNR values ranging from 0db to 5db. $\mathcal{X}_{\text{noise}}$ is of size $5 \times 5 \times 5$. Different CP models with ranks ranging from 1 to 5 are used to fit CORCONDIA. The true rank is $R = 2$.

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:

$$SNR = -10 \log_{10} \sigma^2.$$

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

B. RESULTS

- Fig. 2 illustrates the accuracy of estimated ranks using the proposed approach. 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 shows the convergence curves of our method in the sense of the Reconstruction Square Error (RSE) at each iteration t 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.
- In Fig. 4, the recovery of the CP factors can be shown by checking the subspace error. This can be done by computing the principal angle [25] 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 respectively the exact and the estimated factors at iteration t. We see that the principal angle between the subspaces of the three CP factors converges to a low value w.r.t iterations.
- In Fig. 5, we compare the FARAC method to the CORCONDIA method w.r.t 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 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. On the other hand, the CORCONDIA approach cannot deal with that case [12].
- According to Fig. 6, we can say that 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. Moreover, FARAC deals with the difficult case where the rank exceeds one of the dimensions of the input tensor.

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 choise of the threshold parameter and can handle the difficult case when the rank exceeds one of the dimensions of the tensor unlike to the CORCONDIA method. Applications such as image denoising could be explored using the FARAC method.

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:

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

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

$$\mathcal{L}_{U_n} = \frac{\gamma_2}{2}\left|\left|\mathcal{X} - \mathcal{G} \times_{n=1}^{N}{}_n U_n\right|\right|_F^2 \quad (12)$$

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

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

As a difference from 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 derived as in a classical matrix linear regression problem:

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

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

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

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

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

Let us derive $\mathcal{L}_{\tilde{\mathcal{G}}}$ w.r.t $\tilde{\mathcal{G}}(r_1,\ldots,r_N)$ 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):

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

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

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

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

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

Let us derive $\mathcal{L}_\lambda$ w.r.t λ_l and set it to 0.

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

Furthermore, we have the following:

$$\left|\left| \bigcirc_{n=1}^{N} u_n^{(l)} \right|\right|^2 = \prod_{n=1}^{N} \left|\left| u_n^{(l)} \right|\right|^2.$$

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

This article has been accepted for publication in IEEE Access. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2022.3189793

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

Hence, using the soft threshholding operator $S_{\frac{\gamma_1}{\gamma_2}}$, we conclude the value of λ_l:

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

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 matrix.

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

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

  since γ_2 is a positive Lagrange multiplier.
- Convexity w.r.t $\tilde{\mathcal{G}}$:
  We first put 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 $x \in \mathbb{R}^{{R_0}^N - R_0}$,

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

  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.P.Mandic, L.De Lathauwer, G.Zhou, Q.Zhao, C.Caiafa, and H.A.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.V.Oseledets, H.A.Phan, Q.Zhao, and D.P.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] G. Bergqvist and Erik G. Larsson. The higher-order singular value decomposition: Theory and an application [lecture notes]. IEEE Signal Processing Magazine, 27(3):151–154, 2010.
[5] C.JHillar, , and L.H.Lim. Most tensor problems are np-hard. Journal of the ACM (JACM), 60(6):1–39, 2013.
[6] DP.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.
[7] F.Cong, Q.H.Lin, L.D.Kuang, X.F.Gong, P.Astikainen, and T.Ristaniemi. Tensor decomposition of eeg signals: a brief review. Journal of neuroscience methods, 248:59–69, 2015.
[8] I.V.Oseledets. Tensor-train decomposition. SIAM Journal on Scientific Computing, 33(5):2295–2317, 2011.
[9] J.B.Denis and T.Dhorne. Orthogonal tensor decomposition of 3-way tables. In Multiway Data Analysis, pages 31–37, 1989.
[10] J.B.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.
[11] 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.
[12] K.Liu, JPCL.Da.Costa, HC.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.
[13] T.G. Kolda and B.W. Bader. Tensor decompositions and applications. SIAM Review, 51(3):455–500, September 2009.
[14] L.D Lathauwer, B.De.Moor, and J.Vandewalle. A multilinear singular value decomposition. SIAM journal on Matrix Analysis and Applications, 21(4):1253–1278, 2000.
[15] 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.
[16] M.Mørup and L.K.Hansen. Automatic relevance determination for multiway models. Journal of Chemometrics, 23:352 – 363, 07 2009.
[17] L.De.Lathauwer M.Sørensen. Blind signal separation via tensor decomposition with vandermonde factor: Canonical polyadic decomposition. IEEE Transactions on Signal Processing, 61(22):5507–5519, 2013.
[18] N.D.Sidiropoulos, G.B Giannakis, and R.Bro. Blind parafac receivers for ds-cdma systems. IEEE Transactions on Signal Processing, 48(3):810–823, 2000.
[19] N.D.Sidiropoulos, R.Bro, and G.B.Giannakis. Parallel factor analysis in sensor array processing. IEEE Transactions on Signal Processing, 48(8):2377–2388, 2000.

[20] E.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.

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

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

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

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

[25] 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.

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

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

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

[29] 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 multidimensional 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.