



HAL
open science

A COUPLED JOINT EIGENVALUE DECOMPOSITION ALGORITHM FOR CANONICAL POLYADIC DECOMPOSITION OF TENSORS

Rémi André, Xavier Luciani, Eric Moreau

► **To cite this version:**

Rémi André, Xavier Luciani, Eric Moreau. A COUPLED JOINT EIGENVALUE DECOMPOSITION ALGORITHM FOR CANONICAL POLYADIC DECOMPOSITION OF TENSORS. IEEE SAM 2016, Jul 2016, Rio de Janeiro, France. hal-01862247

HAL Id: hal-01862247

<https://hal.science/hal-01862247>

Submitted on 27 Aug 2018

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

A COUPLED JOINT EIGENVALUE DECOMPOSITION ALGORITHM FOR CANONICAL POLYADIC DECOMPOSITION OF TENSORS

Rémi André, Xavier Luciani, and Eric Moreau,

Aix Marseille Université, CNRS, ENSAM, LSIS, UMR 7296, 13397 Marseille, France.
Université de Toulon, CNRS, LSIS, UMR 7296, 83957 La Garde, France.

ABSTRACT

In this paper we propose a novel algorithm to compute the joint eigenvalue decomposition of a set of squares matrices. This problem is at the heart of recent direct canonical polyadic decomposition algorithms. Contrary to the existing approaches the proposed algorithm can deal equally with real or complex-valued matrices without any modifications. The algorithm is based on the algebraic polar decomposition which allows to make the optimization step directly with complex parameters. Furthermore, both factorization matrices are estimated jointly. This “coupled” approach allows us to limit the numerical complexity of the algorithm. We then show with the help of numerical simulations that this approach is suitable for tensors canonical polyadic decomposition.

1. INTRODUCTION

Many iterative algorithms have been proposed to compute the Canonical Polyadic Decomposition (CPD) of tensors [1–4]. However these approaches can suffer from convergence problems (local minima, slow convergence or high computational cost) and are very sensible to over-factoring and correlated factors [2,5]. In the meantime, a number of authors has shown how the CPD can be rewritten into a Joint EigenValue Decomposition problem (JEVD) [5–7].

JEVD consists in finding the eigenvector matrix \mathbf{A} that jointly diagonalizes a given set of K non-defective matrices $\mathbf{M}^{(k)}$ in the following way:

$$\mathbf{M}^{(k)} = \mathbf{A}\mathbf{D}^{(k)}\mathbf{A}^{-1}, \quad \forall k = 1, \dots, K. \quad (1)$$

It is worth mentioning that in practice if the JEVD have a solution then it is unique up to a permutation and a scaling of the columns of \mathbf{A} , that is assumed here. We refer to [6] for a sufficient and necessary condition of JEVD uniqueness.

The link between CPD and JEVD has given birth to efficient CPD algorithms such as CFS [7], DIAG [5] or SECSI [8]. Indeed, although the JEVD is computed iteratively throughout a Jacobi-like procedure, this procedure involves small matrices and it usually requires very few iterations to converge, we thus speak of “direct” CPD algorithms. In comparison with iterative algorithms, direct algorithms are fast and less

sensible to convergence, over-factoring and correlated factors problems [5].

Several JEVD algorithms has been proposed in the literature. Most of them are based on Jacobi-like updates (sweeping procedure) and look for a factorized form of the updating matrix. In [11] Fu and Gao introduced a first algorithm called sh-rt and based on the polar decomposition. The same factorization is also at the heart of JUST [12] and JD TM [5] algorithms while JET-U and JET-O resort to the LU factorisation [10]. Finally in [13] we proposed an alternative approach, called JDTE that does not resort to any sweeping procedure. These algorithms have been extensively compared in [5], [10] and [13]. Most of them require significant modifications to deal with complex-valued matrices that make the optimization step more difficult during the sweeping procedure and somehow decrease the performances [10]. This is because complex versions of these algorithms need to estimate the real and the imaginary part or the modulus and the angle of the parameters separately. Therefore, in this paper we propose a versatile JEVD algorithm that can deal equally with real or complex-valued matrices, without any modifications. The method resorts to the algebraic polar decomposition for the factorization of the updating matrix. This allows us to make the optimization step directly upon the complex parameters. The second originality of the approach in the JEVD context is that both factorization matrices are estimated jointly. This “coupled” strategy limits the numerical complexity of the algorithm. It was originally introduced to deal with other Joint Diagonalization (JD) problems [14–16].

Notations Scalars are denoted by a lower case (a), vectors by a boldface lower case (\mathbf{a}), matrices by a boldface upper case (\mathbf{A}) and tensors by an upper case boldface calligraphic (\mathcal{A}). a_i is the i -th element of vector \mathbf{a} , $A_{i,j}$ is the (i, j) -th element of matrix \mathbf{A} and so on for tensors. Determinant and Frobenius norm of \mathbf{A} are denoted $\det\{\mathbf{A}\}$ and $\|\mathbf{A}\|$ respectively. $|z|$ is the complex modulus of z .

2. MATRIX JEVD FOR TENSOR CPD

We recall here how the canonical polyadic decomposition can be rewritten into a JEVD problem. This algorithm called

DIAG has been introduced in [17] (under the name SALT) and deeply investigated in [5]. Another connection between both problems had been proposed in [7]. The rank- N CPD of a tensor \mathcal{T} of order Q , size $I_1 \times \dots \times I_Q$ is given by:

$$\mathcal{T}_{i_1, \dots, i_Q} = \sum_{n=1}^N X_{i_1, n}^{(1)} \dots X_{i_Q, n}^{(Q)}, \quad (2)$$

where $\mathbf{X}^{(q)}$ defines the q -th factor matrix of size $I_q \times N$. Entries of \mathcal{T} can be stored in an unfolding matrix \mathbf{T} of size $\prod_{p=1}^P I_p \times \prod_{q=P+1}^Q I_q$ by regrouping, in an appropriate way, the first P modes on the rows of \mathbf{T} and the $Q - P$ other modes on its columns. The CPD can then be rewritten in a matrix form:

$$\mathbf{T} = \mathbf{Y}_{\mathbf{X}}^{(P,1)} \left(\mathbf{Y}_{\mathbf{X}}^{(Q,P+1)} \right)^\top, \quad (3)$$

with: $\mathbf{Y}_{\mathbf{X}}^{(b,a)} = \mathbf{X}^{(b)} \odot \mathbf{X}^{(b-1)} \odot \dots \odot \mathbf{X}^{(a)}$, ($b > a$) and where \odot is the Khatri-Rao product. Note that one can easily deduce $\mathbf{X}^{(a)}, \dots, \mathbf{X}^{(b)}$ from $\mathbf{Y}_{\mathbf{X}}^{(b,a)}$ as explained in [5] so we only recall here how to compute $\mathbf{Y}_{\mathbf{X}}^{(P,1)}$ and $\mathbf{Y}_{\mathbf{X}}^{(Q,P+1)}$. Let $\mathbf{U}\mathbf{S}\mathbf{V}^H$ be the singular value decomposition of \mathbf{T} , truncated at order N , there exists a non singular square matrix \mathbf{A} of size $N \times N$ such that:

$$\mathbf{Y}_{\mathbf{X}}^{(P,1)} = \mathbf{U}\mathbf{A} \text{ and } \left(\mathbf{Y}_{\mathbf{X}}^{(Q,P+1)} \right)^\top = \mathbf{A}^{-1} \mathbf{S}\mathbf{V}^H. \quad (4)$$

$\left(\mathbf{Y}_{\mathbf{X}}^{(Q,P+1)} \right)^\top$ can be seen as an horizontal block matrix:
 $\left(\mathbf{Y}_{\mathbf{X}}^{(Q,P+1)} \right)^\top = \left[\phi^{(1)} \left(\mathbf{Y}_{\mathbf{X}}^{(Q-1,P+1)} \right)^\top, \dots, \phi^{(I_Q)} \left(\mathbf{Y}_{\mathbf{X}}^{(Q-1,P+1)} \right)^\top \right],$ (5)
 where $\phi^{(1)}, \dots, \phi^{(I_Q)}$ are the I_Q diagonal matrices built from the I_Q rows of matrix $\mathbf{X}^{(Q)}$. Then, (4) and (5) yield:

$$\mathbf{S}\mathbf{V}^H = \left[\Gamma^{(1)\top}, \dots, \Gamma^{(I_Q)\top} \right], \quad (6)$$

where $\Gamma^{(k)} = \mathbf{Y}_{\mathbf{X}}^{(Q-1,P+1)} \phi^{(k)} \mathbf{A}^\top$ for $k = 1, \dots, I_Q$. Assuming that matrices $\Gamma^{(k)}$ and matrix $\mathbf{Y}_{\mathbf{X}}^{(Q-1,P+1)}$ are full column rank, then they all admit a Moore-Penrose matrix inverse denoted by \sharp . Thereby, we can define for any couple (k_1, k_2) with $k_1 = 1, \dots, I_Q - 1$ and $k_2 = k_1 + 1, \dots, I_Q$:

$$\mathbf{M}^{(k_1, k_2)} \stackrel{\text{def}}{=} \left(\Gamma^{(k_1)\sharp} \Gamma^{(k_2)} \right)^\top, \quad (7)$$

$$= \mathbf{A} \mathbf{D}^{(k_1, k_2)} \mathbf{A}^{-1}, \quad (8)$$

where $\mathbf{D}^{(k_1, k_2)} = \phi^{(k_2)} \phi^{(k_1)\sharp}$ are diagonal matrices. As a result, \mathbf{A} performs the JEVD of the set of matrices \mathbf{M} and is estimated using a JEVD algorithm. Matrices $\mathbf{Y}_{\mathbf{X}}^{(P,1)}$ and $\mathbf{Y}_{\mathbf{X}}^{(Q,P+1)}$ are then immediately deduced from \mathbf{A} using (4).

3. A COUPLED JEVD ALGORITHM

We now describe the proposed coupled JEVD algorithm, that works equally in the real and complex case *i.e* matrices \mathbf{A} , $\mathbf{D}^{(k)}$ and $\mathbf{M}^{(k)}$ can equally belong to \mathbb{R} or \mathbb{C} . Let us first recall the general scheme of JD algorithms. We want to build an estimate $\widehat{\mathbf{A}}$ of matrix \mathbf{A} (up to the scale and permutation indeterminacy). At each iteration (also called sweep),

the algorithm computes a new set of $N(N - 1)/2$ nonsingular matrices, denoted $\mathbf{B}^{(i,j)}$, with $i = 1, \dots, N - 1$ and $j = i + 1, \dots, N$ and does the following updates of $\widehat{\mathbf{A}}$ and $\mathbf{M}^{(k)}$ (for each k):

$$\begin{cases} \widehat{\mathbf{A}} & \leftarrow \widehat{\mathbf{A}} \left(\prod_{i=1}^{N-1} \prod_{j=i+1}^N \mathbf{B}^{(i,j)} \right)^{-1} \\ \mathbf{M}^{(k)} & \leftarrow \prod_{i=1}^{N-1} \prod_{j=i+1}^N \mathbf{B}^{(i,j)} \mathbf{M}^{(k)} \left(\prod_{i=1}^{N-1} \prod_{j=i+1}^N \mathbf{B}^{(i,j)} \right)^{-1}. \end{cases}$$

The process is repeated so that at the end all matrices $\mathbf{M}^{(k)}$ are diagonals and $\widehat{\mathbf{A}} = \mathbf{A}\mathbf{\Lambda}\mathbf{\Pi}$, where $\mathbf{\Lambda}$ and $\mathbf{\Pi}$ are diagonal and permutation matrices respectively. $\widehat{\mathbf{A}}$ can be initialized using the identity matrix, single EVD of one matrix $\mathbf{M}^{(k)}$ or generalized EVD. In practice, the updates in (9) consists in $N(N - 1)/2$ successive (i, j) -updates of $\widehat{\mathbf{A}}$ and $\mathbf{M}^{(k)}$, defined as:

$$\begin{cases} \widehat{\mathbf{A}} & \leftarrow \widehat{\mathbf{A}} \left(\mathbf{B}^{(i,j)} \right)^{-1} \\ \mathbf{M}^{(k)} & \leftarrow \mathbf{B}^{(i,j)} \mathbf{M}^{(k)} \left(\mathbf{B}^{(i,j)} \right)^{-1}. \end{cases} \quad (10)$$

The underlying idea being that each (i, j) -update brings matrices $\mathbf{M}^{(k)}$ closer to diagonal matrices by optimizing a ‘‘diagonality’’ criterion.

We propose here a new way to build the (i, j) -updating matrices $\mathbf{B}^{(i,j)}$. First, we define the (i, j) -updated matrices $\mathbf{N}^{(k)}$ as:

$$\mathbf{N}^{(k)} = \mathbf{B}^{(i,j)} \mathbf{M}^{(k)} \left(\mathbf{B}^{(i,j)} \right)^{-1}. \quad (11)$$

We then look for the matrix $\mathbf{B}^{(i,j)}$ that minimizes the cost function \mathcal{C} [5, 10, 18], given by:

$$\mathcal{C}(\mathbf{B}^{(i,j)}) = \sum_{k=1}^K \left(\left| N_{i,j}^{(k)} \right|^2 + \left| N_{j,i}^{(k)} \right|^2 \right). \quad (12)$$

Thus we can impose the following structure for $\mathbf{B}^{(i,j)}$:

$$\begin{cases} \forall (p, q) \notin \{(i, i), (i, j), (j, i), (j, j)\}, & B_{p,q}^{(i,j)} = \delta_{p,q}, \\ B_{i,i}^{(i,j)} = b_1^{(i,j)}, & B_{i,j}^{(i,j)} = b_2^{(i,j)}, \\ B_{j,i}^{(i,j)} = b_3^{(i,j)}, & B_{j,j}^{(i,j)} = b_4^{(i,j)}. \end{cases} \quad (13)$$

where $b_1^{(i,j)}, b_2^{(i,j)}, b_3^{(i,j)}, b_4^{(i,j)}$ are unknown parameters and δ is the Kronecker delta. Thanks to the scaling indeterminacy of the JEVD problem, we can also impose $\det\{\mathbf{B}^{(i,j)}\} = 1$. Now, defining matrices:

$$\widetilde{\mathbf{M}}^{(k)} = \begin{pmatrix} M_{i,i}^{(k)} & M_{i,j}^{(k)} \\ M_{j,i}^{(k)} & M_{j,j}^{(k)} \end{pmatrix}, \quad \widetilde{\mathbf{N}}^{(k)} = \begin{pmatrix} N_{i,i}^{(k)} & N_{i,j}^{(k)} \\ N_{j,i}^{(k)} & N_{j,j}^{(k)} \end{pmatrix},$$

$$\widetilde{\mathbf{B}}^{(i,j)} = \begin{pmatrix} b_1^{(i,j)} & b_2^{(i,j)} \\ b_3^{(i,j)} & b_4^{(i,j)} \end{pmatrix},$$

equation (11) becomes:

$$\widetilde{\mathbf{N}}^{(k)} = \widetilde{\mathbf{B}}^{(i,j)} \widetilde{\mathbf{M}}^{(k)} \left(\widetilde{\mathbf{B}}^{(i,j)} \right)^{-1} \quad (14)$$

and the cost function becomes:

$$\mathcal{C}(\widetilde{\mathbf{B}}^{(i,j)}) = \sum_{k=1}^K \left(\left| \widetilde{N}_{1,2}^{(k)} \right|^2 + \left| \widetilde{N}_{2,1}^{(k)} \right|^2 \right). \quad (15)$$

In order to simplify the notations, in the following we will refer to $\mathbf{B}^{(i,j)}$, $\tilde{\mathbf{B}}^{(i,j)}$, $b_1^{(i,j)}$, $b_2^{(i,j)}$, $b_3^{(i,j)}$ and $b_4^{(i,j)}$ as \mathbf{B} , $\tilde{\mathbf{B}}$, b_1 , b_2 , b_3 and b_4 respectively.

We can immediately deduce from $\det\{\mathbf{B}\} = 1$ that:

$$\tilde{\mathbf{B}}^{-1} = \begin{pmatrix} b_4 & -b_2 \\ -b_3 & b_1 \end{pmatrix} \quad (16)$$

and (14) yields:

$$\begin{cases} \tilde{N}_{1,2}^{(k)} &= b_1 b_2 (M_{j,j}^{(k)} - M_{i,i}^{(k)}) + b_1^2 M_{i,j}^{(k)} - b_2^2 M_{j,i}^{(k)}, \\ \tilde{N}_{2,1}^{(k)} &= -b_3 b_4 (M_{j,j}^{(k)} - M_{i,i}^{(k)}) - b_2^2 M_{i,j}^{(k)} + b_4^2 M_{j,i}^{(k)}. \end{cases} \quad (17)$$

At this stage, one should note that during the sweeping procedure it is expected that the updating matrix \mathbf{B} gets closer and closer to the identity matrix. Thus, if we assume that we are close enough to the solution we can reasonably assume that $|b_1| \gg |b_2|$ and $|b_4| \gg |b_3|$ and the previous system can be simplified:

$$\begin{cases} \tilde{N}_{1,2}^{(k)} &\simeq b_1 b_2 (M_{j,j}^{(k)} - M_{i,i}^{(k)}) + b_1^2 M_{i,j}^{(k)}, \\ \tilde{N}_{2,1}^{(k)} &\simeq -b_3 b_4 (M_{j,j}^{(k)} - M_{i,i}^{(k)}) + b_4^2 M_{j,i}^{(k)}. \end{cases} \quad (18)$$

Now defining vectors $\mathbf{b}_{12} = [b_1 b_2 \ b_1^2]^\top$, $\mathbf{b}_{34} = [b_3 b_4 \ b_4^2]^\top$ and matrices:

$$\mathbf{M}_{12} = \begin{pmatrix} M_{j,j}^{(1)} - M_{i,i}^{(1)} & M_{i,j}^{(1)} \\ \vdots & \vdots \\ M_{j,j}^{(K)} - M_{i,i}^{(K)} & M_{i,j}^{(K)} \end{pmatrix} \quad \mathbf{M}_{21} = \begin{pmatrix} M_{i,i}^{(1)} - M_{j,j}^{(1)} & M_{j,i}^{(1)} \\ \vdots & \vdots \\ M_{i,i}^{(K)} - M_{j,j}^{(K)} & M_{j,i}^{(K)} \end{pmatrix}$$

yields:

$$\begin{cases} \sum_{k=1}^K |\tilde{N}_{1,2}^{(k)}|^2 &= \mathbf{b}_{12}^\mathbf{H} \mathbf{M}_{12}^\mathbf{H} \mathbf{M}_{12} \mathbf{b}_{12}, \\ \sum_{k=1}^K |\tilde{N}_{2,1}^{(k)}|^2 &= \mathbf{b}_{34}^\mathbf{H} \mathbf{M}_{21}^\mathbf{H} \mathbf{M}_{21} \mathbf{b}_{34}. \end{cases} \quad (19)$$

It then clearly appears that the two contributions to the cost function can be minimized separately and thus both parameters vectors \mathbf{b}_{12} and \mathbf{b}_{34} can be computed independently. In practice, we can first look for the 1-dimensional subspaces of \mathbf{b}_{12} and \mathbf{b}_{34} . In this purpose it is well known that the norm-1 vector that minimize $f(\mathbf{x}) = \mathbf{x}^\mathbf{H} \mathbf{M}_{12}^\mathbf{H} \mathbf{M}_{12} \mathbf{x}$ is the normalized eigenvector associated to the eigenvalue of $\mathbf{M}_{12}^\mathbf{H} \mathbf{M}_{12}$ of smallest modulus. In the same way the norm-1 vector that minimize $g(\mathbf{x}) = \mathbf{x}^\mathbf{H} \mathbf{M}_{21}^\mathbf{H} \mathbf{M}_{21} \mathbf{x}$ is the normalized eigenvector associated to the eigenvalue of $\mathbf{M}_{21}^\mathbf{H} \mathbf{M}_{21}$ of smallest modulus. We denote \mathbf{e} and \mathbf{f} these two vectors and we have:

$$\begin{cases} \mathbf{e} &= \alpha \mathbf{b}_{12}, \\ \mathbf{f} &= \beta \mathbf{b}_{34}, \end{cases} \quad (20)$$

where α and β are two unknown values. We now explain how we deduce $\tilde{\mathbf{B}}$ from \mathbf{e} and \mathbf{f} by using the algebraic polar decomposition of $\tilde{\mathbf{B}}$.

Matrix polar decomposition states that any square real (complex) valued matrix \mathbf{X} can be factorized as a product of an orthogonal (unitary) matrix and a positive-definite symmetric (hermitian) matrix. Actually, in the complex case we have

an alternative result stating that any square matrix \mathbf{X} can be factorized as a product of a complex orthogonal matrix and a complex symmetric matrix if and only if $\mathbf{X}^\top \mathbf{X}$ is similar to $\mathbf{X} \mathbf{X}^\top$ [19]. This factorization is called the algebraic polar decomposition of \mathbf{X} [20, 21]. A complex matrix \mathbf{Q} is said complex orthogonal if $\mathbf{Q}^\top \mathbf{Q} = \mathbf{I}$ and a complex matrix \mathbf{S} is said complex symmetric if $\mathbf{S}^\top = \mathbf{S}$.

Therefore, let \mathbf{Q} and \mathbf{S} be the (complex) orthogonal and symmetric matrices of the (algebraic) polar decomposition of $\tilde{\mathbf{B}}$ i.e. $\tilde{\mathbf{B}} = \mathbf{Q} \mathbf{S}$. Note that since $\tilde{\mathbf{B}}$ is nonsingular, $\tilde{\mathbf{B}}^\top \tilde{\mathbf{B}}$ is similar to $\tilde{\mathbf{B}} \tilde{\mathbf{B}}^\top$ and so the existence of the decomposition is guaranteed. Because we have imposed $\det\{\tilde{\mathbf{B}}\} = 1$ and thanks to the scaling indeterminacy of the JEVD problem we can still impose $\det\{\mathbf{Q}\} = \det\{\mathbf{S}\} = 1$ and in these conditions there exists a couple of parameters (t, t_h) so that:

$$\mathbf{Q} = \frac{1}{\sqrt{1+t^2}} \begin{pmatrix} 1 & -t \\ t & 1 \end{pmatrix}; \quad \mathbf{S} = \frac{1}{\sqrt{1-t_h^2}} \begin{pmatrix} 1 & t_h \\ t_h & 1 \end{pmatrix}. \quad (21)$$

Defining $\gamma = \frac{1}{\sqrt{(1+t^2)(1-t_h^2)}}$, $\tilde{\mathbf{B}} = \mathbf{Q} \mathbf{S}$ gives:

$$\mathbf{b}_{12} = \gamma \begin{pmatrix} (1-tt_h)(t_h-t) \\ (1-tt_h)^2 \end{pmatrix}; \quad \mathbf{b}_{34} = \gamma \begin{pmatrix} (t+t_h)(tt_h+1) \\ (tt_h+1)^2 \end{pmatrix}. \quad (22)$$

Now, (22) and (20) yield:

$$\frac{e_1}{e_2} = \frac{t_h - t}{1 - tt_h} \quad \text{and} \quad \frac{f_1}{f_2} = \frac{t + t_h}{tt_h + 1} \quad (23)$$

and finally we deduce:

$$\begin{cases} t_h^2 - 2 \frac{e_1 f_1 + e_2 f_2}{e_2 f_1 + e_1 f_2} t_h + 1 &= 0, \\ t &= \frac{e_1 - e_2 t_h}{t_h e_1 - e_2}. \end{cases} \quad (24)$$

Because t_h should be close to 0, it is chosen as the root of the binomial with the smallest modulus. From t_h we immediately deduce t then $\tilde{\mathbf{B}}$ and \mathbf{B} .

It is worth mentioning that: (i.) In the real case the algebraic polar decomposition is equivalent to the polar decomposition, therefore the proposed algorithm can be directly applied on real- or complex-valued matrices without any modification.

(ii.) By construction, $\det\{\mathbf{Q}\} = \det\{\mathbf{S}\} = \det\{\tilde{\mathbf{B}}\} = 1$ so that existence of $\tilde{\mathbf{B}}^{-1}$ is guaranteed throughout the algorithm.

Numerical complexity Numerical complexity of an iterative algorithm based on matrix computations can be approximated by the number of multiplications computed during one iteration of the algorithm. Numerical complexities of JEVD algorithms are dominated by the updates of the matrix set. When dealing with real-valued matrices, the dominant term for the proposed algorithm is $4KN^3$. In comparison dominant terms for JUST, JDTE, JET-O and JET-U algorithms are $11KN^3$, $8KN^3$, $8KN^3$, $2KN^3$, $2KN^3$ and KN^3 respectively. Numerical complexities of complex versions of the algorithms (when they exist) are still dominated by the updating step. Thus we can simply multiply the previous values by 4. In conclusion, one iteration of the proposed

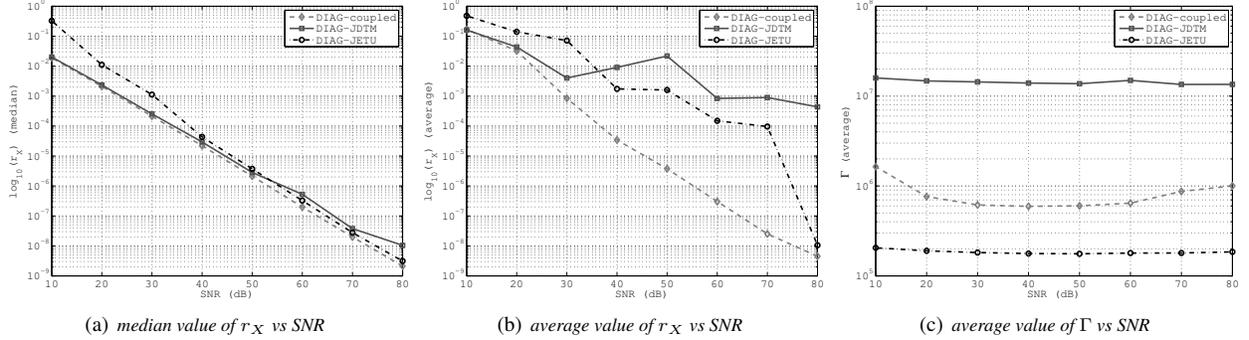


Fig. 1. Comparison of DIAG-JDTM, DIAG-JETU and DIAG-coupled for the rank 5 CPD of complex valued-tensors of size $10 \times 10 \times 10$.

algorithm is less costly than one iteration of other polar decomposition based algorithms but more costly than one iteration of JDTE or LU decomposition based algorithms. However, the global (and practical) computational cost of each algorithm is obtained by multiplying its numerical complexity by the number of computed iterations before convergence and this number varies a lot from one algorithm to another.

4. NUMERICAL SIMULATIONS

Canonical polyadic decomposition of tensors is one of the main application of JEVD. Thereby, we chose to evaluate the performances of the proposed approach in this context by considering CPD of complex-valued tensors of order 3. In this purpose we compare 3 different implementations of the DIAG algorithm described in section 2; namely, DIAG-JDTM, DIAG-JETU and DIAG-coupled, respectively based on JDTE, JETU and the proposed algorithm. JDTE and JETU have been chosen as references because JDTE is very efficient to estimate the matrix of eigenvectors from noisy matrix sets while JETU has the lowest numerical complexity. The first comparison criterion, r_X , is an average value of the normalized mean squares error computed between actual and estimated factor matrices:
$$r_X = \frac{1}{3} \sum_{q=1}^3 \frac{\|\mathbf{X}^{(q)} - \widehat{\mathbf{X}}^{(q)}\|^2}{\|\mathbf{X}^{(q)}\|^2},$$

where $\widehat{\mathbf{X}}^{(q)}$ is the estimated factor matrix corrected from scaling and permutation indeterminacies. Algorithms performances are evaluated from 100 Monte-Carlo (MC) runs. For each run a new noise-free tensor is built using a new random draw of the 3 complex factor matrices. Then a Gaussian white noise is added in order to obtain the desired SNR value and the noisy tensor is decomposed using the three algorithms. When computing the DIAG algorithm, the JEVD is applied to the whole matrix set built from equation (7) (no matrix is removed). Matrix $\widehat{\mathbf{A}}$ is initialized as the identity matrix. During the JEVD process the squared off-diagonal components of the K matrices $\mathbf{M}^{(k)}$ are summed and the obtained value is compared to the value computed at the previous sweep. The JEVD process is stopped when the relative deviation

between two successive values is smaller than 10^{-6} or when a maximal number of 100 sweeps has been reached. Then the global computational cost of the JEVD step, Γ , is computed as defined previously. This gives us our second comparison criterion. Factor matrices used to build the tensors are of size 10×5 , so rank-5 CPD of $10 \times 10 \times 10$ tensors are performed and the JEVD problem is of size: $K = 45$, $N = 5$.

We present on figure 1 the evolution of the median and average values of r_X and of the average value of Γ with respect to the SNR value. Regarding the estimation precision of the factor matrices, the proposed approach is the more efficient whatever the SNR value. Indeed DIAG-JDTM can compete only at the lowest SNR values (10 and 20 dB) whereas DIAG-JETU is clearly surpassed for the more realistic SNR values (this is particularly true below 40 dB). Furthermore, the average value of r_X shows a clear gap between the three algorithms in favor of DIAG-coupled. Indeed during the JEVD process, JDTE and JETU have not always converged contrary to our coupled JEVD algorithm. Now regarding the global computational cost, figure 1(c) shows that the proposed algorithm is from 9.7 to 23.5 times less costly than JDTE (depending on the SNR value) and from 3.4 to 8 times more costly than JETU. To sum up, DIAG-coupled appears here as the best choice in terms of estimation precision while being much less costly than the second more accurate solution.

5. CONCLUSION

In this paper we have introduced a new method for joint eigenvalue decomposition. Contrary to classical approaches, this method works equally on real or complex-valued matrices by exploiting the algebraic polar decomposition and a coupled estimation of the factorization matrices. Numerical simulations demonstrate that it can be successfully used to compute the canonical polyadic decomposition of complex-valued tensors, rewritten as a JEVD problem. Indeed, in the representative scenario that we have considered: the rank-5 CPD of a noisy tensor of size $10 \times 10 \times 10$, the proposed algorithm allows a better estimation of the factors matrices than reference algorithms while keeping a low computational cost.

6. REFERENCES

- [1] R. A. Harshman, "Foundation of PARAFAC procedure: Models and conditions for an 'explanatory' multi-mode factor analysis," *UCLA working papers in Phonetics*, no. 16, pp. 1–84, 1970.
- [2] P. Comon, X. Luciani, and A. L. F. de Almeida, "Tensor decompositions, alternating least squares and other tales," *Journal of Chemometrics*, vol. 23, april 2009.
- [3] E. Acar, T. G. Kolda, and D. M. Dunlavy, "An optimization approach for fitting canonical tensor decompositions," *Tech. Rep. SANDIA national laboratories*, February 2009.
- [4] T. G. Kolda and B. W. Bader, "Tensor decompositions and applications," *SIAM Review*, vol. 51, no. 3, pp. 455–500, 2009.
- [5] X. Luciani and L. Albera, "Canonical polyadic decomposition based on joint eigenvalue decomposition," *Chemometrics and Intelligent Laboratory Systems*, vol. 132, no. 0, pp. 152 – 167, 2014.
- [6] L. De Lathauwer, B. De Moor, and J. Vandewalle, "Computation of the canonical decomposition by means of a simultaneous Schur decomposition," *SIAM Journal on Matrix Analysis and Applications*, vol. 26, no. 2, pp. 295–327, 2004.
- [7] F. Roemer and M. Haardt, "A closed-form solution for multilinear parafac decompositions," in *SAM 08, Fifth IEEE Sensor Array and Multichannel Signal Processing Workshop*, july 2008, pp. 487–491.
- [8] —, "A semi-algebraic framework for approximate {CP} decompositions via simultaneous matrix diagonalizations (secsi)," *Signal Processing*, vol. 93, no. 9, pp. 2722 – 2738, 2013.
- [9] M. Haardt and J. Nosssek, "Simultaneous Schur decomposition of several nonsymmetric matrices to achieve automatic pairing in multidimensional harmonic retrieval problems," *IEEE Transactions on Signal Processing*, vol. 46, no. 1, pp. 161–169, January 1998.
- [10] X. Luciani and L. Albera, "Joint eigenvalue decomposition of non-defective matrices based on the lu factorization with application to ica," *IEEE Transactions on Signal Processing*, vol. 63, no. 17, pp. 4594–4608, Sept 2015.
- [11] T. Fu and X. Gao, "Simultaneous diagonalization with similarity transformation for non-defective matrices," in *ICASSP 2006, 2006 IEEE International Conference on Acoustics Speech and Signal Processing*, vol. 4, May 2006, pp. 1137–1140.
- [12] R. Iferroudjene, K. Abed-Meraim, and A. Belouchrani, "A new Jacobi-like method for joint diagonalization of arbitrary non-defective matrices," *Applied Mathematics and Computation*, vol. 211, no. 2, pp. 363–373, 2009.
- [13] R. Andre, T. Trainini, X. Luciani, and E. Moreau, "A fast algorithm for joint eigenvalue decomposition of real matrices," in *European Signal Processing Conference (EU-SIPCO'2015)*, Nice, France, 2015.
- [14] T. Trainini and E. Moreau, "A coordinate descent algorithm for complex joint diagonalization under Hermitian and transpose congruences," *IEEE Transactions on Signal Processing*, vol. 62, no. 19, pp. 4974–4983, Oct 2014.
- [15] V. Maurandi and E. Moreau, "A decoupled Jacobi-like algorithm for non-unitary joint diagonalization of complex-valued matrices," *IEEE Signal Processing Letters*, vol. 21, no. 12, pp. 1453–1456, Dec 2014.
- [16] V. Maurandi, E. Moreau, and C. D. Luigi, "Jacobi like algorithm for non-orthogonal joint diagonalization of Hermitian matrices," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP'2014)*, Florence, Italy, 2014.
- [17] X. Luciani and L. Albera, "Semi-algebraic canonical decomposition of multi-way arrays and joint eigenvalue decomposition," in *IEEE International Conference on Acoustics Speech and Signal Processing*, Prague, Czech Republic, May 22-27 2011, pp. 4104–4107.
- [18] A. Souloumiac, "Nonorthogonal joint diagonalization by combining Givens and hyperbolic rotations," *IEEE Transactions on Signal Processing*, vol. 57, no. 6, pp. 2222–2231, June 2009.
- [19] R. A. Horn and D. I. Merino, "Contragredient equivalence: A canonical form and some applications," *Linear Algebra and its Applications*, vol. 214, pp. 43–92, 1995.
- [20] D. Choudhury and R. Horn, "A complex orthogonal-symmetric analog of the polar decomposition," *SIAM J. Algebraic Discrete Methods*, vol. 8, pp. 219–225, 1987.
- [21] I. Kaplansky, "Algebraic polar decomposition," *SIAM Journal on Matrix Analysis and Applications*, vol. 11, no. 2, pp. 213–217, 1990.