Under–Determined Tensor Diagonalization for Decomposition of Difficult Tensors

Petr Tichavský[1], Anh-Huy Phan[2], and Andrzej Cichocki[2,3]
Institute of Information Theory and Automation of the CAS
P.O.Box 18, 182 08 Prague 8, Czech Republic
p.tichavsky@ieee.org
[2]*Brain Science Institute, RIKEN, Wakoshi, Japan*
phan@brain.riken.jp, cia@brain.riken.jp
[3]*Skolkovo Institute of Science and Technology (Skoltech), Moscow 143026, Russia*

***Abstract*—Analysis of multidimensional arrays, usually called tensors, often becomes difficult in cases when the tensor rank (a minimum number of rank-one components) exceeds all the tensor dimensions. Traditional methods of canonical polyadic decomposition of such tensors, namely the alternating least squares, can be used, but a presence of a large number of false local minima can make the problem hard. Usually, multiple random initializations are advised in such cases, but the question is how many such random initializations are sufficient to get a good chance of finding the right solution. It appears that the number of the initializations can be very large. We propose a novel approach to the problem. The given tensor is augmented by some unknown parameters to the shape that admits ordinary tensor diagonalization, i.e., transforming the augmented tensor into an exact or nearly diagonal form through multiplying the tensor by non-orthogonal invertible matrices. Three possible constraints are proposed to make the optimization problem well defined. The method can be modified for an under-determined block-term decomposition.**

I. Introduction

The paper deals with the canonical polyadic decomposition of some tensors that are hard to be decomposed by traditional methods. For example, matrix multiplication can be represented by certain tensors containing zeros and ones only, ranks of these tensors represent the minimum possible number of scalar multiplications needed to execute the matrix products. Then, CP decomposition of these tensors represents algorithms to compute the products. In particular, tensor of matrix multiplication for two matrices of the size 3×3 has the size $9 \times 9 \times 9$, and despite a tremendous effort of many mathematicians, we only know so far that the rank of the tensor lies in the interval $[19, 23]$ [1], [2]. In other words, we know that product of two matrices of the size 3×3 can be performed through 23 scalar multiplications [3], but it is not certain if this number can be lower. A large number of algorithms for small–size matrix multiplications is presented in [4].

CP decomposition of this kind of tensors, as well as other tensors whose rank exceeds the tensor dimension, is a challenging problem. The term "under-determined" used in this context is borrowed from the area of blind source separation. Here, an under-determined mixture of signals means that the number of the sources in the mixture exceeds the number of the mixtures. Certain blind source separation algorithms separate under-determined mixtures by means of CP decomposition of "under-determined" tensors computed from the received data [7], [18].

If at least one dimension of the tensor exceeds the tensor rank, then the tensor can be decomposed by an algebraic method using joint diagonalization of a set of matrices, described in [8]. Assume that the tensor has order three and size $I \times J \times K$ with $I \geq J \geq K \geq 2$ and rank R. The method can be applied if $R(R-1) \leq J(J-1)K(K-1)/2$. It follows that R can obey $I \geq R \geq J \geq K$. This is the key component of the algorithm SOBIUM (Second Order Blind Identification of Under–determined Mixtures) [7].

The focus of this paper, however, is on the case $R > \min\{I, J, K\}$. In [9], Domanov and De Lathauwer developed an algebraic method to solve this case to some extent: R can exceed all I, J, K, but must obey the condition $R \leq (I+J+K-2)/2 + (I - \sqrt{(J-K)^2 + 4I})/2$. For example, for $I = J = K = 9$ we could have $R \leq 14$. The method seems to work perfectly if there is no noise. Unfortunately, the rank $R = 14$ is not sufficient for decomposition of the matrix multiplication tensor of this size. For this tensor we need a decomposition of rank $R = 23$.

CP decomposition of these tensors of high rank is a challenging problem. In [17], it was proposed to perform the CP decomposition of such tensors by solving the following optimization task:

$$\begin{aligned} &\min \|\mathcal{T} - [[\mathbf{A}, \mathbf{B}, \mathbf{C}]]\|_F^2 \\ &\text{subject to} \quad \|\mathbf{A}\|_F^2 + \|\mathbf{B}\|_F^2 + \|\mathbf{C}\|_F^2 \leq c \end{aligned} \tag{1}$$

where $\mathcal{T}$ is the tensor to be decomposed, $[[\mathbf{A}, \mathbf{B}, \mathbf{C}]]$ is the Kruskal form of the tensor approximation [5] with factor matrices $\mathbf{A}, \mathbf{B}, \mathbf{C}$, $\|\cdot\|_F$ denotes the Frobenius norm (either for a tensor or a matrix), and c is a suitable constant. The optimization can be performed by means of the Levenberg-Marquardt algorithm [6].

The optimization problem in (1) is well defined, because the criterion function is continuous and differentiable, and is minimized in a compact set. The function is not convex, however, false local minima may exist, and multiple random initializations are inevitable. The method is very efficient in decomposing the tensor for 3×3 matrix multiplication, i.e.,

the tensor has the size $9 \times 9 \times 9$ and rank 23. No noise is considered, and the constant c should be in the range of 150 and 200 to get the best performance. Unfortunately, for the method is not able to decompose multiplication tensors of bigger size, say $16 \times 16 \times 16$ for 4×4 matrix multiplication. Neither the currently best existing method can do that [4].

Recently, it has been shown that the CP decomposition of (over) determined tensors (with rank smaller or equal to the tensor dimension) can be done through a tensor diagonalization. The advantage of this approach is that in some difficult scenarios, a cost function of the tensor diagonalization seems to have a lower number of false local minima than other optimization-based methods. Since, however, good algebraic tensor decomposition methods exist in this case, the space for improvements is very narrow if any. Therefore the tensor diagonalization seems to be suitable as a tool for tensor block decomposition rather than for the CP decomposition [12].

Tensor diagonalization means transforming a given tensor into an exact or nearly diagonal form through multiplying the tensor by non-orthogonal invertible matrices along selected dimensions of the tensor. However, the tensor rank has to be smaller or equal to the tensor dimensions. It is a generalization of approximate joint diagonalization (AJD) of a set of matrices [15]. The concept of tensor diagonalization has been introduced by P. Comon and his co-workers [10], [11]. It works for order–three tensors of a cubic shape. The tensor diagonalization in those papers seeks orthogonal matrices that transforms the given tensor into a diagonal one.

In [12], a generalized (non-symmetric) tensor diagonalization was proposed. Here, symmetric and nonsymmetric two-sided tensor diagonalizations of order-three tensors, and three-sided diagonalizations of order-3 or order-4 tensors are considered. Note that in the case of nonsymmetric two-sided diagonalization of order-3 tensors, the method is equivalent to the SECSI method of CP tensor decomposition [13], [14].

This paper presents an extension of the tensor diagonalization method for the under-determined tensors.

The paper is organized as follows: Section 2 presents principles of the ordinary tensor diagonalization and under-determined tensor diagonalization. In Section 3, we explain details of a possible efficient implementation of the method. Section 4 presents some numerical examples, and Section 5 concludes the paper.

II. Tensor diagonalization principle

A. Ordinary (Determined) Tensor Diagonalization

The main idea of the ordinary (determined) tensor diagonalization is to find so-called demixing matrices that transform the given tensor into another tensor that is diagonally dominant. For simplicity, in this section, we only discuss the two-sided tensor diagonalization of order-3 tensors.

Let $\mathcal{T}$ be a tensor of size $R \times R \times M$. The outcome of the diagonalization is the tensor

$$\mathcal{E} = \mathcal{T} \times_1 \mathbf{A} \times_2 \mathbf{B} \tag{2}$$

where $\times_i$ denotes the tensor-matrix multiplication along the dimension i, $i = 1, 2$. The diagonalization is *exact* if all frontal slices of the tensor $\mathcal{E}$ are diagonal matrices. If this happens and the matrices $\mathbf{A}$ and $\mathbf{B}$ are invertible, then $\mathbf{A}^{-1}$ and $\mathbf{B}^{-1}$ are factor matrices of a CP decomposition of $\mathcal{T}$. In particular,

$$\mathcal{T} = [[\mathbf{A}^{-1}, \mathbf{B}^{-1}, \mathbf{C}]]$$

where $\mathbf{C}$ is an $M \times R$ matrix composed of diagonals of the M frontal slices of tensor $\mathcal{E}$, i.e., $\mathbf{C}_{mr} = \mathcal{E}_{rrm}$ for $r = 1, \ldots, R$ and $m = 1, \ldots, M$.

As in [12], consider an operator off_2 which nullifies diagonals in all frontal slices of a tensor. If tensor $\mathcal{E}$ has elements $\mathcal{E}_{ijm}$, $i, j = 1, \ldots, r$, $m = 1, \ldots, M$, then $\text{off}_2(\mathcal{E})$ has elements $(1 - \delta_{ij})\mathcal{E}_{ijm}$, where δ_{ij} is the Kronecker delta. The exact diagonalization means that $\text{off}_2(\mathcal{E}) = \mathbf{0}$ or $\|\text{off}_2(\mathcal{E})\|_F = 0$.

The algorithm TEDIA proposed in [12] does not minimize the Frobenius norm of $\text{off}_2(\mathcal{E})$ directly, but it terminates at invertible $\mathbf{A}$ and $\mathbf{B}$. The algorithm halts when the residual tensor $\mathcal{E}$ cannot be diagonalized any further by any infinitely small rotations $\delta\mathbf{A}$, $\delta\mathbf{B}$, where diagonals of $\delta\mathbf{A}$ and $\delta\mathbf{B}$ are filled with ones, i.e., when it holds $\|\text{off}_2(\mathcal{E})\|_F \leq \|\text{off}_2(\mathcal{E} \times_1 \delta\mathbf{A} \times_2 \delta\mathbf{B})\|_F$ for all such $\delta\mathbf{A}$, $\delta\mathbf{B}$. See [12] for more details.

B. Under–Determined Tensor Diagonalization

Consider a tensor $\mathcal{T}_0$ of size $I \times J \times M$ has rank $R > I \geq J \geq M$. The idea of the under-determined tensor diagonalization is to augment the tensor to a bigger size $R \times R \times M$ which has the same rank and is fully diagonalizable.

Let $\mathcal{T}$ be an augmented tensor of $\mathcal{T}_0$ of size $R \times R \times M$, which has the same elements as $\mathcal{T}_0$, say, $\mathcal{T}_{ijm}$ for $i = 1, \ldots, I$, $j = 1, \ldots, J$, $m = 1, \ldots, M$, and the remaining elements are arbitrary. Let $\text{boff}(\mathcal{T})$ be a tensor of the same size, $R \times R \times M$, with the same elements outside $\mathcal{T}_0$, while the place of $\mathcal{T}_0$ is filled with nulls. Since every element of $\mathcal{T}$ lies either in $\mathcal{T}_0$ or in $\text{boff}(\mathcal{T})$, it holds

$$\|\mathcal{T}\|_F^2 = \|\mathcal{T}_0\|_F^2 + \|\text{boff}(\mathcal{T})\|_F^2 \ .$$

Under-determined Tensor Diagonalization (U-TEDIA) of $\mathcal{T}_0$ consists, in general, in finding demixing matrices $\mathbf{A}$, $\mathbf{B}$ and elements of $\text{boff}(\mathcal{T})$ such that the tensor $\mathcal{E}$ in (2) is as diagonal to the largest possible extent. The diagonalization is illustrated in Fig. 1.

To be more specific, we studied three types of diagonalization, denoted U-TEDIA-1, U-TEDIA-2, and U-TEDIA-3.

Type 1: Minimize $\|\text{off}_2(\mathcal{E})\|_F$ subject to

$$\|\text{boff}(\mathcal{T})\|_F^2 + \|\mathbf{A}^{-1}\|_F^2 + \|\mathbf{B}^{-1}\|_F^2 \leq c \tag{3}$$

where c is a suitable constant.

Type 2: Minimize $\|\text{off}_2(\mathcal{E})\|_F$ under the constraint (3) and an additional constraint

$$\sum_{m=1}^{M} \mathcal{E}_{mmr}^2 = 1 \tag{4}$$

for all $r = 1, \ldots, R$. The condition (4) in the type-2 diagonalization means that columns of the estimated factor matrix $\mathbf{C}$ have all unit Euclidean norm.

Type 3: Minimize $\|\text{off}_2(\mathcal{E})\|_F$ subject to

$$\|\text{boff}(\mathcal{T})\|_F^2 + \|\mathbf{A}\|_F^2 + \|\mathbf{B}\|_F^2 + \|\mathbf{A}^{-1}\|_F^2 + \|\mathbf{B}^{-1}\|_F^2 \leq c \quad (5)$$

where c is a suitable constant.

Note that $\text{off}_2(\mathcal{E})$ is a trilinear function of $\mathbf{A}$, $\mathbf{B}$ and $\text{boff}(\mathcal{T})$, i.e., it is linear individually in $\mathbf{A}$, $\mathbf{B}$ and $\text{boff}(\mathcal{T})$, but not jointly. We could apply, for example, a kind of constrained Alternating Least Squares (ALS) minimization algorithm similar to an ALS algorithm for CPD, but in this paper, we propose a Levenberg–Marquardt algorithm.

All three types of diagonalization guarantee that both $\mathbf{A}$ and $\mathbf{B}$ would be invertible thanks to limiting the Frobenius norms of $\mathbf{A}^{-1}$ and $\mathbf{B}^{-1}$ in (3) and (5). It appears that solutions produced by U-TEDIA-1 are sometimes not well balanced in the sense that the rank-one components differ dramatically. U-TEDIA-2 was proposed to overcome this problem. Its convergence is slightly slower, but it is guaranteed that the decomposition would be well-balanced.

In some cases, these methods achieve exact fit solutions, but it occurs rather rarely (more frequently for U-TEDIA-1). In typical cases, when the final diagonalization error was nonzero, we observe that additional iterations further decrease the cost function, but the norm of the Frobenius norm of $\mathbf{A}$ and $\mathbf{B}$ increases. We hypothesize that if the number of iterations goes to infinity, the Frobenius norm converges to infinity as well. In order to avoid such singular solutions, we can apply U-TEDIA-3. This method has typical behavior of damped Gauss-Newton algorithms. It requires a few dozens of iterations to converge, unlike U-TEDIA-1 and U-TEDIA-2, which never completely converge, unless an exact fit solution is found.

All three variants of TEDIA produce decompositions with low diagonalization error. If the considered tensors are noisy, then the results can be satisfactory. However, we observed that despite the diagonalization errors are small, the approximation errors may not be small. The latter error is defined as

$$E(\mathcal{T}, \mathbf{A}, \mathbf{B}) = \|\mathcal{T}_0 - [[\mathbf{A}_1, \mathbf{B}_1, \mathbf{C}_1]]\|_F^2$$

where $\mathbf{A}_1$ and $\mathbf{B}_1$ are submatrices of $\mathbf{A}^{-1}$ and $\mathbf{B}^{-1}$, respectively, and $\mathbf{C}_1$ is obtained from the diagonal of $\mathcal{T} \times_1 \mathbf{A} \times_2 \mathbf{B}$.

Finally, we need to address a choice of parameter c in the constraint (3) or (5). Obviously, if a CP decomposition of $\mathcal{T}_0$ exists in the form $\mathcal{T}_0 = [[\mathbf{A}_0, \mathbf{B}_0, \mathbf{C}_0]]$, then c should be greater than $\|\mathbf{A}_0\|_F^2 + \|\mathbf{B}_0\|_F^2$. The problem is that we do not know the norms of $\mathbf{A}_0$ and $\mathbf{B}_0$ in advance. On the other hand, the parameter c should not be too high. A greater c means that the domain if the optimization increases, and the criterion may have more side local minima. As a result, achieving the exact diagonalization (exact fit) is harder, in general. A suitable parameter c can be selected by a trial and error method.

III. Implementation Details

We seek a vector of parameters

$$\theta = [\text{vec}(\mathbf{A})^T, \text{vec}(\mathbf{B})^T, \text{vec}(\text{boff}(\mathcal{T}))^T]^T$$

of the length $2R^2 + M(R^2 - IJ)$ that minimizes the cost function

$$\varphi(\theta) = \|\text{off}_2(\mathcal{T} \times_1 \mathbf{A} \times_2 \mathbf{B})\|_F^2$$

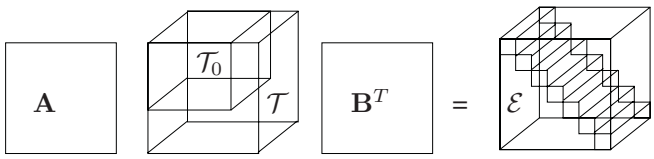

Fig. 1. Under-determined tensor diagonalization seeks demixing matrices $\mathbf{A}$, and $\mathbf{B}$, and elements of $\mathcal{T}$ outside $\mathcal{T}_0$ so that tensor $\mathcal{E}$ is diagonally dominant.

under the constraint (3) or the two constraints (3) and (4), respectively. The ordinary (unconstrained) Levenberg-Marquardt (LM) algorithm updates θ as

$$\theta \leftarrow \theta - (\mathbf{H} + \mu\mathbf{I})^{-1}\mathbf{g} \quad (6)$$

where

$$\mathbf{H} = \mathbf{J}^T\mathbf{J}, \qquad \mathbf{J} = \frac{\partial \mathbf{b}(\theta)}{\partial \theta}, \qquad \mathbf{g} = \mathbf{J}^T\mathbf{b}(\theta) \quad (7)$$

and

$$\mathbf{b}(\theta) = \text{vec}(\text{off}_2(\mathcal{E}(\theta))) \ . \quad (8)$$

μ is a damping parameter, which is sequentially updated according to a rule described in [6]. Let $\mathbf{E}_m$ and $\mathbf{T}_m$ be the $m-$th frontal slices of $\mathcal{E}$ and $\mathcal{T}$, respectively, for $m = 1, \ldots, M$. Then,

$$\mathbf{E}_m = \mathbf{A}\mathbf{T}_m\mathbf{B}^T \ . \quad (9)$$

Computation of the Jacobian $\mathbf{J}$ is straightforward. Note that right-lower part of the Hessian $\mathbf{H} = \mathbf{J}^T\mathbf{J}$ is block diagonal, i.e., easier-to-invert than a full matrix of the same size. This fact helps to reduce number of operations to compute $(\mathbf{H} + \mu\mathbf{I})^{-1}\mathbf{g}$, i.e., to solve the linear system $(\mathbf{H} + \mu\mathbf{I})\mathbf{x} = \mathbf{g}$.

Optimization constrained by the condition (3) or (5) is performed by minimizing the cost function in the tangent plane to the corresponding variety first, and then the minimum point is projected into the variety. The tangent plane is described by one its point θ_0, which is the latest available approximation of the optimum θ, and a normal vector $\mathbf{g}_c$. It can be shown that for the condition (3) it holds

$$\mathbf{g}_c = \begin{bmatrix} \text{vec}(\mathbf{A}^{-T}\mathbf{A}^{-1}\mathbf{A}^{-T}) \\ \text{vec}(\mathbf{B}^{-T}\mathbf{B}^{-1}\mathbf{B}^{-T}) \\ \text{vec}(\text{boff}(\mathcal{T})) \end{bmatrix} \ . \quad (10)$$

The minimizing point in the variety θ_1' is obtained by minimizing the following second-order approximation of the cost function,

$$\varphi(\theta) \approx \varphi(\theta_0) + \mathbf{g}^T(\theta - \theta_0) + \frac{1}{2}(\theta - \theta_0)^T\mathbf{H}(\theta - \theta_0) \quad (11)$$

under the linear constraint $(\theta - \theta_0)^T\mathbf{g}_c = 0$. We use the method of Lagrange multiplier to get

$$\theta_1' = \theta_0 - \mathbf{H}^{-1}\mathbf{g} + \frac{\mathbf{g}_c^T\mathbf{H}^{-1}\mathbf{g}}{\|\mathbf{g}_c\|^2}\mathbf{H}^{-1}\mathbf{g}_c \ . \quad (12)$$

Instead of using (12) directly, we replace $\mathbf{H}^{-1}$ by $(\mathbf{H} + \mu\mathbf{I})^{-1}$ as in (6) in the LM method.

Once θ_1' is found, its projection to the constraint variety is obtained by an appropriate scale change.

Computation of the updates for type-2 and type-3 optimizations is similar, details are omitted for lack of space.

IV. Simulations

We studied CP decomposition of synthetic tensors of the size $5 \times 5 \times 5$ and rank 9. The tensors were constructed from their factor matrices $\mathbf{A}_0$, $\mathbf{B}_0$ and $\mathbf{C}_0$ of the size 5×9 that are taken at random with independent Gaussian distributions of zero mean and unit variance. We generated 100 such tensors and decomposed them using

1) a CP decomposition technique of tensorlab [19], namely the nonlinear-least squares cpd-nls,
2) the constrained Levenberg-Marquardt method denoted LM3c [17] with $c = 200$
3) U-TEDIA-1 with 400 iterations and $c = 300$, followed by refinement by LM3c
4) U-TEDIA-2 with 400 iterations and $c = 300$, followed by refinement by LM3c.
5) U-TEDIA-3 with 100 iterations and $c = 500$, followed by refinement by LM3c.

Since we are primarily interested in low approximation errors, we switch to an algorithm minimizing the approximation error, i.e., LM3c.

The success rate was assessed as the percentage of runs that an algorithm attained an exact decomposition, i.e. the fitting error was lower than 10^{-7}. The tensorlab nls method had a success rate lower than 1%. LM3c had a success rate of 12%, and U-TEDIA-x + LM3c, x=1,2,3, had at most the same success rate. LM3c was not outperformed by the new algorithms.

Another result was obtained for the CP decomposition of the tensor corresponding to the product of two matrices 3×3 to rank 23. Here, we used $c = 500$ for U-TEDIA-1 and U-TEDIA-2, $c = 800$ for U-TEDIA-3, and $c = 200$ for LM3c again. LM3c achieved the exact decomposition in 40% trials, U-TEDIA-1 with LM3c in 47% trials, and U-TEDIA-2 with LM3c in 54% trials, and U-TEDIA-3 with LM3c in 42% trials.

V. Conclusions

U-TEDIA extends the algorithm TEDIA to non-orthogonal diagonalization of tensors of arbitrary sizes. It is a novel approach to the CP tensor decomposition. We present three versions of the algorithm. Their application depends on the criterion of success. Our primary motivation was the challenging decomposition of the matrix multiplication tensors. In this area, we have not succeeded to outperform our earlier algorithm LM3c.

Results in this paper are rather preliminary, further modification of the methods are possible. Like in the case of the ordinary (determined) tensor diagonalization, we anticipate an extension of the method to the under-determined block-term decomposition.

Matlab code of the proposed technique will be posted on the web page of the first author.

ACKNOWLEDGEMENTS

The work of Petr Tichavský was supported by the Czech Science Foundation through Project No. 17-00902S.

References

[1] J. M. Landsberg, *Tensors: Geometry and Applications*, AMS 2012.

[2] M. Bläser, *Fast Matrix Multiplication*. No. 5 in Graduate Surveys. Theory of Computing Library, 2013.

[3] J.D. Laderman, "A noncommutative algorithm for multiplying 3×3 matrices using 23 multiplications", *Bul. Amer. Math. Soc.*, vol. 82, pp. 126–128, 1976.

[4] A. V. Smirnov, "The Bilinear Complexity and Practical Algorithms for Matrix Multiplication", *Computational Mathematics and Mathematical Physics*, vol. 53, no. 12, pp. 1781-1795, 2013.

[5] T.G. Kolda and B.W. Bader, "Tensor decompositions and applications", *SIAM Review*, vol. 51, pp. 455–500, 2009.

[6] K. Madsen, H. B. Nielsen, O. Tingleff, Methods for nonlinear least squares problems, second ed., Department of Mathematical Modelling, Technical University of Denmark, Lyngby, Denmark, 2004.

[7] L. De Lathauwer and J. Castaing, "Blind identification of underdetermined mixtures by simultaneous matrix diagonalization," *IEEE Trans. Signal Process.*, vol. 56, no. 3, pp. 1096-1105, Mar. 2008.

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

[9] I. Domanov and L. De Lathauwer, "Canonical polyadic decomposition of third-order tensors: Relaxed uniqueness conditions and algebraic algorithm," *Linear Algebra and its Applications.*, vol. 513, pp. 342-375, 2017.

[10] P. Comon, "Tensor diagonalization, a useful tool in signal processing," *IFAC-SYSID, 10th IFAC Symposium on System Identification*, M. Blanke and T. Soderstrom, ed., Copenhagen, Denmark, 1994, pp. 77–82.

[11] P. Comon, M. Sorensen, "Tensor diagonalization by orthogonal transforms", *Rapport de recherche ISRN I3S/RR-2007-06-FR*, Universite Nice, Sophia Antipolis.

[12] P. Tichavský, A.H. Phan, A. Cichocki, "Non-orthogonal tensor diagonalization", *Signal Processing* 138 (2017) pp. 313-320.

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

[14] K. Naskovska, M. Haardt, P. Tichavský, G. Chabriel, J. Barrere, "Extension of the semi-algebraic framework for approximate CP decompositions via non-symmetric simultaneous matrix diagonalization", *Proc. ICASSP 2016*, Shanghai, China, 2016, pp. 2971–2975.

[15] G. Chabriel, M. Kleinsteuber , E. Moreau, H. Shen, P. Tichavský, and A. Yeredor, "Joint matrices decompositions and blind source separation", *IEEE Signal Proc. Magazine*, vol. 31, pp. 34-43, 2014.

[16] P. Tichavský, A.H. Phan, A. Cichocki, "Two-sided diagonalization of order-three tensors", *Proc. EUSIPCO 2015*, Nice, pp. 998–1002.

[17] P. Tichavský, A.H. Phan, and A. Cichocki, "Numerical CP decomposition of some difficult tensors", *Journal of Computational and Applied Mathematics*, vol. 317, pp. 362-370, 2017.

[18] P. Tichavský and Z. Koldovský, "Weight adjusted tensor method for blind separation of underdetermined mixtures of nonstationary sources", *IEEE Trans. on Signal Processing*, vol. 59, no. 3, pp. 1037–1047, March 2011.

[19] N. Vervliet, O. Debals, L. Sorber, M. Van Barel and L. De Lathauwer, Tensorlab 3.0, available online, Mar. 2016. URL: http://www.tensorlab.net/