A moment-based approach for guaranteed tensor decomposition

Arthur Marmin, Marc Castella, Jean-Christophe Pesquet

A MOMENT-BASED APPROACH FOR GUARANTEED TENSOR DECOMPOSITION

Arthur Marmin[†] *Marc Castella*[⋆] *Jean-Christophe Pesquet*[†]

[†]Université Paris-Saclay, CentraleSupélec, INRIA, Center for Visual Computing, Gif-sur-Yvette, France
[⋆]SAMOVAR, CNRS, Télécom SudParis, Institut Polytechnique de Paris, Evry, France

ABSTRACT

This paper presents a new scheme to perform the canonical polyadic decomposition (CPD) of a symmetric tensor. We first formulate the CPD problem as a truncated moment problem, where a measure has to be recovered knowing some of its moments. The support of the measure is discrete and encodes the CPD. The support is then retrieved by solving a polynomial system. Using algebraic results, our method resorts only to classical linear algebra operations (eigenvalue method and Schur reordered factorization). This new viewpoint offers theoretical guarantees on the retrieved decomposition. Finally experimental results show the validity of our method and a better reconstruction accuracy compared to classic CPD algorithms.

***Index Terms*—** tensors, canonical polyadic decomposition, moment problem

1. INTRODUCTION

Tensors techniques have spread into many scientific fields nowadays (see e.g. [1, 2]). However, due to their high dimensionality, tensors remain difficult to visualize, to handle, and to interpret. In this context, factorizing tensors has become a crucial step. The goal is to decompose an intricate tensor into simpler atoms that can be more easily interpreted and can lead to faster computations. Famous tensor decompositions include canonical polyadic (CPD) and Tucker decompositions [3] as well as tensor-train [4] or block-term [5] decompositions. We focus here on CPD. Popular algorithms to perform CPD include unconstrained nonlinear optimization [5] (OPT), alternating least square (ALS) [3], nonlinear least square (NLS) [3], and generalized eigenvalue decomposition [6] (GEVD). The objective of this paper is to present an alternative approach based on moments.

The starting point of our method is the link between CPD and the moment problem, which has been unveiled recently [7]. In the latter work, a method to detect the symmetric rank of the tensor based on the rank of certain moment matrices has been proposed. We here extend this work by proposing a novel approach for the extraction of the generating vectors of the CPD. More precisely, our approach is grounded on methods for solving polynomial systems based on algebraic results, which allow us to solve a moment problem. Theoretical results guarantee that the retrieved CPD is exact.

Links between CPD and moment problem have already been mentioned [8–10] but used in different ways. In [10], the authors reformulate the minimum rank CPD problem as a generalized problem of moments. It is then relaxed into a hierarchy of convex semi-definite programming problems, the dual problem of which provides certificates for the correctness of the CPD. Experiments are restricted to small dimensional tensors, which, from our experience, is related to the heavy computational load of this method. In contrast, our proposed method scales well for medium to high dimensional tensors. The algebraic methods in [8, 9] provide a completely different perspective based on a sum of given powers of linear forms that requires a heavy theoretical background. Although similar in spirit, the simplicity of our method may provide further insight and accessibility.

Our paper is organized as follows: Section 2 introduces the CPD for a symmetric tensor. Section 3 summarizes the connection between CPD and moment problem as well as some tools used in our method. Section 4 explains how to solve the moment problem and retrieve the generating vectors of the CPD. Section 5 shows simulation results and comparisons of our method with CPD algorithms from common tensor toolboxes, and Section 6 concludes our work.

We use the following notation: $\binom{n}{p}$ is the binomial coefficient "among n choose p". Upper case calligraphic letters denote tensors ($\mathcal{T}$) and fraktur letter ($\mathfrak{T}$) their values after reindexing (see Section 3). Bold upper case letters ($\mathbf{M}$) denote matrices, bold lower case letters ($\mathbf{v}$) denote vectors, and lower case letters (s) denote scalars. For a multi-index $\tilde{\boldsymbol{\alpha}} = (\alpha_0, \cdots, \alpha_n)$ of length $n+1$, we define its absolute value $|\tilde{\boldsymbol{\alpha}}| = \alpha_0 + \cdots + \alpha_n$ and we denote by $\boldsymbol{\alpha} = (\alpha_1, \cdots, \alpha_n)$ its sub-index of length n where the first index has been dropped. We define the lexicographic ordering for two multi-indices $\tilde{\boldsymbol{\alpha}}$ and $\tilde{\boldsymbol{\beta}}$ of same absolute value, as follows: $\tilde{\boldsymbol{\alpha}}$ comes before $\tilde{\boldsymbol{\beta}}$ if the leftmost non-zero entry of $\tilde{\boldsymbol{\alpha}} - \tilde{\boldsymbol{\beta}}$ is positive. $\mathbb{N}_k^n$ is the set of multi-indices of n elements whose absolute value is smaller than or equal to k.

2. CANONICAL POLYADIC DECOMPOSITION

Let $\mathcal{T}$ denote a tensor of order d on $\mathbb{R}^{n+1}$ with $d \geq 4$ an even integer. In this paper, we deal with symmetric tensors only, i.e. tensors whose entries $(\mathcal{T}_{i_1,\ldots,i_d})_{0\leq i_1,\ldots,i_d\leq n}$ are unchanged by any permutation of the indices. A tensor is said to be symmetric rank-1 if it can be expressed as

$$\mathbf{v}^{\otimes d} = \underbrace{\mathbf{v} \otimes \cdots \otimes \mathbf{v}}_{d \text{ times}}$$

for a vector $\mathbf{v} = (v_i)_{i\in[\![0,n]\!]}$ of $\mathbb{R}^{n+1}$, that is $\left[\mathbf{v}^{\otimes d}\right]_{i_1,\ldots,i_d} = v_{i_1}\ldots v_{i_d}$. The CPD problem that we consider consists in finding a decomposition of $\mathcal{T}$ into a sum of rank-1 tensors, $\mathcal{T} = \sum_{r=1}^{R} \mathbf{v}(r)^{\otimes d}$, or equivalently

$$\mathcal{T}_{i_1,\ldots,i_d} = \sum_{r=1}^{R} v_{i_1}(r)\ldots v_{i_d}(r)\,. \tag{1}$$

The minimum value of R is called the (symmetric) rank and is here assumed to be known. This is a common assumption in CPD problems and many algorihms such as OPT, ALS and NLS require the prior knowledge of the rank value, which can be determined or estimated first [7, 11, 12]. Hence, we want here to determine the vectors $(\mathbf{v}(r))_{r\in[\![1,R]\!]}$.

Under the assumption that there is an index l in $[\![0, n]\!]$ such that $v_l(r) \neq 0$ for every r in $[\![1, R]\!]$, Decomposition (1) can be expressed in a dehomogenized form by normalizing each $\mathbf{v}(r)$ with its l^{th} coordinate

$$\mathcal{T} = \sum_{r=1}^{R} \lambda_r \left(\frac{\mathbf{v}(r)}{v_l(r)}\right)^{\otimes d} = \sum_{r=1}^{R} \lambda_r \mathbf{u}(r)^{\otimes d}\,, \tag{2}$$

where $\mathbf{u}(r) = (v_1(r)/v_l(r),\ldots,v_n(r)/v_l(r))$ and $\lambda_r = v_l(r)^d$ is positive. With no loss of generality, we take $l = 0$ in the following. All the following results still hold for any other index l after an adequate permutation of coordinates.

3. CP DECOMPOSITION AS A MOMENT PROBLEM

We recall the connection between the CPD and the moment problem, which was introduced in [7] and gives our strong theoretical guarantees on the CPD.

Due to the symmetry assumption, the order of the indices in $\mathbf{i} = (i_1,\ldots,i_d)$ has no influence on the value of the tensor element $\mathcal{T}_{i_1,\ldots,i_d}$, which is uniquely defined by specifying the number of times each index value appears in $\mathbf{i}$. More precisely, to any d-tuple $\mathbf{i} = (i_1,\ldots,i_d)$, we associate an $(n+1)$-tuple $\tilde{\boldsymbol{\alpha}}(\mathbf{i}) = (\alpha_0(\mathbf{i}),\ldots,\alpha_n(\mathbf{i}))$, where for each l in $[\![0, n]\!]$, $\alpha_l(\mathbf{i})$ is the number of times the index value l appears in $\mathbf{i}$. We therefore define the tensor values $\mathcal{T}_{\mathbf{i}} = \mathfrak{T}_{\tilde{\boldsymbol{\alpha}}(\mathbf{i})}$, where $\mathfrak{T}_{\tilde{\boldsymbol{\alpha}}}$ is indexed by $(n+1)$-tuples $\tilde{\boldsymbol{\alpha}}$ satisfying $|\tilde{\boldsymbol{\alpha}}| = d$.

Note that dehomogenization pairs each $\tilde{\boldsymbol{\alpha}}$ with a unique $\boldsymbol{\alpha}$ since $\alpha_0 = d - |\boldsymbol{\alpha}|$. Following the dehomogenization and the above reindexing, Decomposition (1) can be reexpressed as

$$\begin{aligned}\mathfrak{T}_{\tilde{\boldsymbol{\alpha}}} &= \sum_{r=1}^{R} \lambda_r u_1(r)^{\alpha_1}\ldots u_n(r)^{\alpha_n} \\ &= \int x_1^{\alpha_1}\ldots x_n^{\alpha_n}\mu(\mathrm{d}\mathbf{x}) = \int \mathbf{x}^{\boldsymbol{\alpha}}\mu(\mathrm{d}\mathbf{x})\,,\end{aligned} \tag{3}$$

where μ is the R-atomic positive measure defined on n variables and supported on the points $(\mathbf{u}(r))_{r\in[\![1,R]\!]}$

$$\mu = \sum_{r=1}^{R} \lambda_r \delta_{\mathbf{u}(r)}\,.$$

The right hand side of (3) is the moment of order $\boldsymbol{\alpha}$ of the measure μ and its degree is $|\boldsymbol{\alpha}|$. Finding the vectors $(\mathbf{u}(r))_{r\in[\![1,R]\!]}$ and the coefficients $(\lambda_r)_{r\in[\![1,R]\!]}$ in a CPD is therefore equivalent to estimating an R-atomic measure μ from its moments of degree up to $2k$. The latter is a truncated moment problem which is a well known problem encountered in several scientific fields.

An important tool for solving our problem is the moment matrix of order $k = \frac{d}{2}$ defined by

$$\left(\forall(\tilde{\boldsymbol{\alpha}},\tilde{\boldsymbol{\beta}}) \in \left(\mathbb{N}^{n+1}\right)^2, |\tilde{\boldsymbol{\alpha}}| = \left|\tilde{\boldsymbol{\beta}}\right| = k\right)\ (\mathbf{M}_k)_{(\tilde{\boldsymbol{\alpha}},\tilde{\boldsymbol{\beta}})} = \mathfrak{T}_{\tilde{\boldsymbol{\alpha}}+\tilde{\boldsymbol{\beta}}}\,,$$

where the multi-indices $\tilde{\boldsymbol{\alpha}}$ and $\tilde{\boldsymbol{\beta}}$ are arranged with respect to the lexicographic order. The number of rows and columns in the matrix $\mathbf{M}_k$ is $N = \binom{n+k}{k}$. Note that odd order tensors can be handled by setting $k = \frac{d-1}{2}$ and defining the moment matrix elements $(\mathbf{M}_k)_{(\tilde{\boldsymbol{\alpha}},\tilde{\boldsymbol{\beta}})}$ as $\mathfrak{T}_{\tilde{\boldsymbol{\alpha}}+\tilde{\boldsymbol{\beta}}+(1,0,\ldots,0)}$.

4. EXTRACTING THE CPD VECTORS

This section deals with the recovery of the support of the measure μ, or equivalently the vectors in the CPD, from $\mathbf{M}_k$. Since we assume the rank R of $\mathcal{T}$ is known, the existence of an R-atomic measure μ and its corresponding decomposition (3) is guaranteed. Furthermore, results from [7] ensure that $\mathbf{M}_k$ has rank R.

4.1. Link between vectors $(\mathbf{u}(r))_{r\in[\![1,R]\!]}$ and matrix $\mathbf{M}_k$

The kernel of the moment matrix $\mathbf{M}_k$ is defined as

$$\operatorname{Ker}\mathbf{M}_k = \{\mathbf{p} \in \mathbb{R}^N \mid \mathbf{M}_k\mathbf{p} = 0\}\,.$$

Moreover, the moment matrix $\mathbf{M}_k$ is indexed by the pair of multi-indices $(\tilde{\boldsymbol{\alpha}},\tilde{\boldsymbol{\beta}})$ whose absolute values are equal to k. Equivalently, we can use the pair of multi-indices $(\boldsymbol{\alpha},\boldsymbol{\beta})$, each one belonging to $\mathbb{N}_k^n$. Since to each multi-index $\boldsymbol{\alpha}$ corresponds a monic monomial $\mathbf{x}^{\boldsymbol{\alpha}}$, we can associate to each vector $\mathbf{p}$ in the kernel of the moment matrix $\mathbf{M}_k$ a polynomial p such that

$$(\forall\mathbf{x} \in \mathbb{R}^n)\quad p(\mathbf{x}) = \sum_{\boldsymbol{\alpha}\in\mathbb{N}_k^n} p_{\boldsymbol{\alpha}}\mathbf{x}^{\boldsymbol{\alpha}}\,.$$

According to [13, Theorem 5.29], the vectors $(\mathbf{u}(r))_{r\in[\![1,R]\!]}$ forming the support of the sought measure μ are the common zeros of the polynomials with coefficients in $\operatorname{Ker}\mathbf{M}_k$, that is

$$(\mathbf{u}(r))_{r\in[\![1,R]\!]} = \{\mathbf{x}\in\mathbb{R}^n \mid (\forall \mathbf{p}\in\operatorname{Ker}\mathbf{M}_k) \quad p(\mathbf{x}) = 0\}\,.$$

Finding the generating vectors of the CPD is therefore equivalent to solving a multivariate polynomial system.

4.2. Eigenvalue method to solve polynomial system

The eigenvalue method transforms the original polynomial system into a linear algebra problem. Despite being described in a few places [14], this method seems to be widely ignored by the signal processing community. We briefly describe here the different steps in our context. A more general and theoretical explanation of the method can be found in [15].

The first step consists in computing through Gaussian elimination the reduced row echelon form of $\mathbf{M}_k$, which is an upper triangular $N\times N$ matrix whose last $N-R$ rows are composed solely of 0. Dropping the last $N-R$ rows of zeros, we note $\mathbf{U}$ the obtained $R\times N$ matrix and $(\mathbf{u}_{\boldsymbol{\alpha}})_{\boldsymbol{\alpha}\in\mathbb{N}_k^n}$ its N column vectors. We then read from $\mathbf{U}$ the column multi-indices of the pivot elements. We get R pivots whose indices are denoted by $(\boldsymbol{\beta}_r)_{r\in[\![1,R]\!]}$. We have then the following result:

Proposition 1 *For every i in $[\![1,n]\!]$, the i-th coordinates of the R points $(\mathbf{u}(r))_{r\in[\![1,R]\!]}$ are the R eigenvalues of the matrices $\mathbf{N}_i$ extracted from $\mathbf{U}$ such that*

$$\mathbf{N}_i = [\mathbf{u}_{\boldsymbol{\beta}_1+\mathbf{e}_i}\ldots\mathbf{u}_{\boldsymbol{\beta}_R+\mathbf{e}_i}]\,.$$

where $\mathbf{e}_i$ is a multi-index of $\mathbb{N}^n$ whose all elements are equal to zero except its i-th element which is 1.

This result is a direct application of Stickelberger eigenvalue theorem [15, Theorem 4.5]. The matrices $(\mathbf{N}_i)_{i\in[\![1,n]\!]}$ are called the multiplication matrices. The origin of this name can be found in [15].

Example Let $\mathcal{T}$ be a tensor of order $d=4$ in $\mathbb{R}^3$ with rank $R=3$. The associated moment matrix $\mathbf{M}_2$ is a 6×6 matrix that can be seen indexed by the following ordered set

$$\mathbb{N}_2^2 = \{(0,0),(1,0),(0,1),(2,0),(1,1),(0,2)\}\,.$$

We then obtain its reduced row echelon form, e.g.

$$\mathbf{U} = \begin{bmatrix} 1 & 0 & 0 & u_1 & u_4 & u_7\\ 0 & 1 & 0 & u_2 & u_5 & u_8\\ 0 & 0 & 1 & u_3 & u_6 & u_9\end{bmatrix}\,.$$

We read out the indices of the pivots

$$\boldsymbol{\beta}_1 = (0,0)\,,\quad \boldsymbol{\beta}_2 = (1,0)\,,\quad \boldsymbol{\beta}_3 = (0,1)\,,$$

whence the corresponding multiplication matrices

$$\mathbf{N}_1 = \begin{bmatrix} 0 & u_1 & u_4\\ 1 & u_2 & u_5\\ 0 & u_3 & u_6\end{bmatrix}\,,\quad \mathbf{N}_2 = \begin{bmatrix} 0 & u_4 & u_7\\ 0 & u_5 & u_8\\ 1 & u_6 & u_9\end{bmatrix}\,.$$

Note, for instance, that the second column of $\mathbf{N}_1$ is $\mathbf{u}_{(2,0)}$.

4.3. Computation of the eigenvalues of $(\mathbf{N}_i)_{i\in[\![1,n]\!]}$

Since multiplication matrices all commute pairwise, they preserve the eigenspaces of each others. A numerically stable way to compute their eigenvalues based on Schur factorization [16] is then available and summarized below.

First, build a random linear combination $\mathbf{N}_h$ of the matrices $(\mathbf{N}_i)_{i\in[\![1,n]\!]}$

$$\mathbf{N}_h = \sum_{i=1}^{n} a_i\mathbf{N}_i\,, \tag{4}$$

where $(a_i)_{i\in[\![1,n]\!]}$ are randomly chosen real numbers summing up to one. Now, a key point is that the left eigenspaces of $\mathbf{N}_h$ need all to be one-dimensional in order to avoid to miss any points in the support of μ. Since the rank of the matrix $\mathbf{M}_k$ is R, this holds almost surely [15] for any choice of the $(a_i)_{i\in[\![1,n]\!]}$. Following [16], the left eigenspaces of $\mathbf{N}_h$ are then found by computing an ordered Schur decomposition $\mathbf{Q}\mathbf{T}\mathbf{Q}^\top$ of $\mathbf{N}_h$ where $\mathbf{Q}$ is an orthogonal matrix and $\mathbf{T}$ is upper triangular. The coordinates of the points $(\mathbf{u}(r))_{r\in[\![1,R]\!]}$ are thus given by

$$(\forall i\in[\![1,n]\!])(\forall r\in[\![1,R]\!]) \quad u_i(r) = \mathbf{q}_r^\top\mathbf{N}_i\mathbf{q}_r\,,$$

where $\mathbf{q}_r$ is the r-th column of matrix $\mathbf{Q}$.

Finally the weighting coefficients $(\lambda_r)_{r\in[\![1,R]\!]}$ of (2) are retrieved by solving a linear system. The sketch of the extraction method is provided in Algorithm 1.

Algorithm 1: Extraction of CPD vectors

Inputs : Moment matrix $\mathbf{M}_k$ and rank R of $\mathcal{T}$
Output: Vectors $(\mathbf{u}(r))_{r\in[\![1,R]\!]}$ generating $\mathcal{T}$

1 Compute the reduced row echelon form of $\mathbf{M}_k$ and extract $\mathbf{U}$;
2 Find the column indices of the pivots in $\mathbf{U}$;
3 Read multiplication matrices $(\mathbf{N}_i)_{i\in[\![1,n]\!]}$ from $\mathbf{U}$;
4 Find the common eigenvalues of the multiplication matrices:
5   Compute a random combination $\mathbf{N}_h$ of the multiplication matrices as in (4) ;
6   Compute the ordered Schur decomposition $\mathbf{Q}\mathbf{T}\mathbf{Q}^\top$ of $\mathbf{N}_h$;
7   Read the i-th coordinate of the points $(\mathbf{u}(r))_{r\in[\![1,R]\!]}$ by computing $\mathbf{Q}^\top\mathbf{N}_i\mathbf{Q}$;

---

5. NUMERICAL EXPERIMENTS

5.1. Performance of the proposed method

We generate each symmetric rank-R tensor $\mathcal{T}$ randomly by drawing the coefficients of its vectors $(\mathbf{v}(r))_{r\in[\![1,R]\!]}$ from a uniform distribution on $[-1,1]$. We then apply our method to retrieve the CPD of $\mathcal{T}$ and denote by $\hat{\mathcal{T}}$ the tensor reconstructed from the computed CPD. We assume that the rank R is known (any existing rank detection method can be used) and we focus only on the retrieval of the generating vectors in the CPD.

For each test case, we run 100 simulations and show only the average results. To assess the quality of the reconstruction, we use the relative error between tensors $\mathcal{T}$ and $\hat{\mathcal{T}}$ defined as $\frac{\|\mathcal{T}-\hat{\mathcal{T}}\|_{\mathrm{F}}}{\|\mathcal{T}\|_{\mathrm{F}}}$, where $\|.\|_{\mathrm{F}}$ is the Frobenius norm. Moreover, we also compute a score inspired by [17] to evaluate the reconstruction quality. The score measures the similarity between the original generating vectors $(\mathbf{v}(r))_{r\in[\![1,R]\!]}$ of a symmetric rank-R tensor and the vectors $(\hat{\mathbf{v}}(r))_{r\in[\![1,R]\!]}$ obtained after computing its CPD. It is computed as the product of the correlation between the vectors $(\mathbf{v}(r))_{r\in[\![1,R]\!]}$ and $(\hat{\mathbf{v}}(r))_{r\in[\![1,R]\!]}$, namely

$$\text{score} = \prod_{r=1}^{R} \frac{\langle \mathbf{v}(r) \mid \hat{\mathbf{v}}(r)\rangle}{\|\mathbf{v}(r)\| \, . \, \|\hat{\mathbf{v}}(r)\|} .$$

Thereby, when the score is close to 1, the vectors $(\hat{\mathbf{v}}(r))_{r\in[\![1,R]\!]}$ are strongly correlated to $(\mathbf{v}(r))_{r\in[\![1,R]\!]}$ and the CPD is accurate. However, if the score is close to 0, the CPD yields a poor quality decomposition.

Table 1 shows that our method can accurately reconstruct the CPD of a symmetric tensor for various combinations of dimension, order and rank. The running time is still fair and scales well with the order or the rank of the tensor.

Table 1. Quality and reconstruction time of our method

$n+1$	d	R	Relative error	Time (s)
10	4	10	4.10e-12	0.02
30	4	10	7.62e-12	7.11
50	4	10	4.51e-12	178.3
100	4	10	9.95e-12	13818
30	4	5	7.83e-13	3.97
30	4	30	1.20e-11	20.27
30	5	10	7.16e-13	7.27
30	6	10	9.06e-13	7.30

Figure 1 shows several cases where the data in the tensor are corrupted with an additive i.i.d. zero-mean Gaussian noise. We first perform a truncated SVD of $\mathbf{M}_k$ at rank R before applying Algorithm 1. For a low noise level, the score is very high, close to 1; the CPD is exactly retrieved. Indeed, in the noiseless case, the decomposition returned by our method is guaranteed to be exact in contrast to some methods such as ALS. However, as the variance increases, the score reduces, the CPD is not accurate anymore and we lose any guarantee. Furthermore, for given dimension and order, we observe that the lower the rank, the lower the sensitivity to the noise.

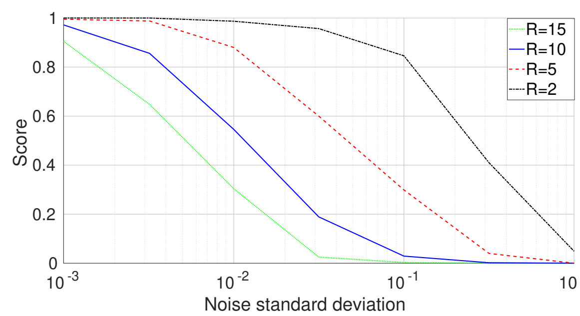

Fig. 1. Reconstruction score for noisy tensor $(n = 29, d = 4)$

5.2. Comparison with other methods

We now compare our method to state-of-the-art CPD algorithms, especially the implementation of ALS, NLS, OPT and GEVD from Tensorlab 3.0 [18]. Table 2 shows a comparison of the relative error for the different algorithms and several different types of symmetric tensors. Results for GEVD have not been reported as they are similar to results for our method. Generally speaking, algebraic methods retrieve faithful CPD but, as shown in Figure 1 for our method, are sensitive to noise. On the other hand, Table 2 shows that methods based on optimization strategies are much less accurate than our method for exact decomposition.

Table 2. Comparison with standard CPD methods

Tensor features			Relative error			
$n+1$	d	R	ALS	OPT	NLS	Our method
10	4	10	9e−3	2e−2	1e−3	4e−10
30	4	10	1e−2	2e−2	5e−4	8e−12
50	4	10	1e−2	2e−2	1e−4	5e−12
100	4	10	1e−2	7e−3	8e−5	1e−13
30	4	5	7e−3	2e−1	1e−3	8e−13
30	4	20	4e−3	5e−2	3e−4	1e−11
30	4	30	4e−3	5e−2	4e−4	1e−11

6. CONCLUSION

By interpreting the problem of the CPD as a moment problem, we propose an algebraic method that guarantees to recover the unique measure solving the problem and allow us to deduce the vectors of the CPD. Finally our simulations show the validity of this method. For CPD in a noiseless context, the method appears to be quite competitive with existing ones.

7. REFERENCES

[1] H. Becker, L. Albera, P. Comon, R. Gribonval, F. Wendling, and I. Merlet, "Brain-source imaging: From sparse to tensor models," *IEEE Signal Process. Mag.*, vol. 32, no. 6, pp. 100–112, Nov. 2015.

[2] L. Qi, H. Chen, and Y. Chen, *Tensor Eigenvalues and Their Applications*, Springer Singapore, 2018.

[3] N. D. Sidiropoulos, L. D. Lathauwer, X. Fu, K. Huang, E. E. Papalexakis, and C. Faloutsos, "Tensor decomposition for signal processing and machine learning," *IEEE Trans. Signal Process.*, vol. 65, no. 13, pp. 3551–3582, July 2017.

[4] I. V. Oseledets, "Tensor-train decomposition," *SIAM J. Sci. Comput.*, vol. 33, no. 5, pp. 2295–2317, Jan. 2011.

[5] L. Sorber, M. V. Barel, and L. D. Lathauwer, "Optimization-based algorithms for tensor decompositions: Canonical polyadic decomposition, decomposition in rank-$(l_r, l_r, 1)$ terms, and a new generalization," *SIAM J. Optim.*, vol. 23, no. 2, pp. 695–720, Jan. 2013.

[6] E. Sanchez and B. R. Kowalski, "Tensorial resolution: A direct trilinear decomposition," *J. Chemometrics*, vol. 4, no. 1, pp. 29–45, Jan. 1990.

[7] A. Marmin, M. Castella, and J.-C. Pesquet, "Detecting the rank of a symmetric tensor," in *27th European Signal Processing Conference (EUSIPCO)*. Sept. 2019, IEEE.

[8] A. Bernardi, J. Brachat, P. Comon, and B. Mourrain, "General tensor decomposition, moment matrices and applications," *J. Symbolic Computat.*, vol. 52, pp. 51–71, May 2013.

[9] J. Nie, "Generating polynomials and symmetric tensor decompositions," vol. 17, no. 2, pp. 423–465, Nov. 2015.

[10] G. Tang and P. Shah, "Guaranteed tensor decomposition: A moment approach," in *Proc. Int. Conf. Mach. Learn.*, F. Bach and D. Blei, Eds., Lille, France, 07–09 Jul 2015, vol. 37 of *Proceedings of Machine Learning Research*, pp. 1491–1500, PMLR.

[11] Q. Zhao, G. Zhou, L. Zhang, A. Cichocki, and S. Amari, "Bayesian robust tensor factorization for incomplete multiway data," vol. 27, no. 4, pp. 736–748, Apr. 2016.

[12] B. Recht, M. Fazel, and P. A. Parrilo, "Guaranteed minimum-rank solutions of linear matrix equations via nuclear norm minimization," *SIAM Rev.*, vol. 52, no. 3, pp. 471–501, Jan. 2010.

[13] M. Laurent, "Sums of squares, moment matrices and optimization over polynomials," in *Emerging Applications of Algebraic Geometry*, pp. 157–270. Springer New York, Sept. 2008.

[14] D. Henrion and J.-B. Lasserre, "Detecting global optimality and extracting solutions in GloptiPoly," in *Positive Polynomials in Control*, vol. 312, pp. 293–310. Springer Berlin Heidelberg, Sept. 2005.

[15] D. A. Cox, J. Little, and D. O'Shea, *Using Algebraic Geometry*, Springer-Verlag, 2005.

[16] R. M. Corless, P. M. Gianni, and B. M. Trager, "A reordered Schur factorization method for zero-dimensional polynomial systems with multiple roots," in *Proceedings of the 1997 International Symposium on Symbolic and Algebraic Computation*, New York, NY, USA, July 1997, pp. 133–140, ACM Press.

[17] B. W. Bader, T. G. Kolda, et al., "Matlab tensor toolbox version 3.0-dev," Available online, Oct. 2017.

[18] N. Vervliet, O. Debals, L. Sorber, M. Van Barel, and L. De Lathauwer, "Tensorlab 3.0," Mar. 2016, Available online.