Canonical Polyadic Decomposition: From 3-way to N-way

Guoxu Zhou*†, Zhaoshui He†, Yu Zhang*, Qibin Zhao* and Andrzej Cichocki*
**Laboratory for Advanced Brain Signal Processing, RIKEN, Brain Science Institute*
Wako-shi, Saitama 3510198, Japan.
E-mail: {zhouguoxu, cia, yuzhuang,qbzhao}@brain.riken.jp
†*Faculty of Automation, Guangdong University of Technology*
Guangzhou 510006, China. E-mail: zhshhe@gdut.edu.cn.

***Abstract*—Canonical Polyadic (or CANDECOMP/PARAFAC, CP) decompositions are widely applied to analyze high order data, i.e. N-way arrays. Existing CP decomposition methods use alternating least square (ALS) iterations and hence need to compute the inverse of matrices and unfold tensors frequently, which are very time consuming for large-scale data and when the order of the tensor N is large. Fortunately, once at least one factor matrix has been correctly estimated, all the remaining factors can be computed efficiently and uniquely by using a series of rank-1 approximations. Motivated by this fact, to perform a full N-way CP decomposition, we run 3-way CP decompositions on a sub-tensor to estimate two factors first. Then the remaining factors are estimated via an efficient Khatri-Rao product recovering procedure. In this way the whole ALS iterations with respect to each mode are avoided and the efficiency can be significantly improved. Simulations show that, compared with ALS based CP decomposition methods, the proposed method is more efficient and is easier to escape from local solutions for high order tensors.**

***Keywords*-CP (PARAFAC) decompositions, tensor decompositions, Khatri-Rao product, Alternating Least Square.**

I. Introduction

Higher-order tensors (multi-way arrays) have gained increasing importance as they are often more natural representations of multi-dimensional data than matrices in many practical applications. A fundamental problem in these applications is how to find informative representations of tensors, i.e., tensor decompositions. Tensor decompositions are very attractive because they take into account spatial, temporal and spectral information, and provide links among the various extracted factors or latent variables with physical or physiological meaning and interpretation [1], [2].

As one of the most important tensor decomposition models, Canonical Polyadic (CP), also named as CANDECOMP/PARAFAC decomposition [3], [4] has been extensively studied and found many practical applications. One big advantage of CP decomposition is that the factors are essentially unique under mild conditions, which makes it very useful in the cases even when no or only very limited *a priori* information is available on the factors. In the CP model, the matricizations (unfoldings) of a tensor are just the products of one factor matrix with another matrix which is generated by using the other factors in a special way. This has led to the widely adopted alternating least square (ALS) methods for CP decompositions, e.g. see [5]. Unfortunately, in this way we have to unfold the tensor to each mode matrix frequently, which is one major bottleneck for efficiency of CP decomposition methods.

Recently, our results showed that, for CP decompositions, once only one factor has been correctly estimated, all the other factors can be computed uniquely and efficiently by using a series of singular value decomposition (SVD) of rank-1 matrices [1], even if the tensor is ill-posed [6]. This motivated us to perform blind source separation (BSS) on one single mode first and then use efficient rank-1 approximation methods to recover the other factors, which leads to the CP-SMBSS method for CP decompositions. The CP-SMBSS is quite useful if some *a priori* knowledge on at least one mode is available. In this paper we deal with the case where such *a priori* information is completely unavailable.

The following notations will be adopted. Bold capitals (e.g., $\mathbf{A}$) and bold lowercase letters (e.g., $\mathbf{y}$) denote matrices and vectors, respectively. Underlined bold capitals, e.g. $\underline{\mathbf{Y}}$, denote tensors. Unfolding (matricization, flattening) of a tensor $\underline{\mathbf{Y}} \in \mathbb{R}^{I_1\times I_2\times\cdots\times I_N}$ in mode-n is denoted as $\mathbf{Y}_{(n)} \in \mathbb{R}^{I_n\times\prod_{p\neq n} I_p}$, which consists of arranging all possible mode-n tubes (vectors) as the columns of a matrix [2]. $\odot$ denotes the Khatri-Rao product (column-wise Kronecker product) of matrices and $\bigodot_{k=k_1}^{k_2} \mathbf{A}^{(k)} = \mathbf{A}^{(k_1)} \odot \mathbf{A}^{(k_1-1)} \cdots \odot \mathbf{A}^{(k_2+1)} \odot \mathbf{A}^{(k_2)}$ with $k_1 > k_2$. Readers are referred to [2] for detailed tensor notations.

II. N-way CP Decompositions Based on a 3-way Decomposition

A. Review of CP Decompositions

A CP decomposition (factorization) of a tensor $\underline{\mathbf{Y}}\in \mathbb{R}^{I_1\times I_2\cdots\times I_N}$ can be defined as

$$\underline{\mathbf{Y}} = \sum_{j=1}^{J} \lambda_j\, \mathbf{a}_j^{(1)} \circ \mathbf{a}_j^{(2)} \circ \cdots \circ \mathbf{a}_j^{(N)} + \underline{\mathbf{E}}, \tag{1}$$

[1] A matrix $\mathbf{Y}$ is rank-1 if and only if $\mathbf{Y} = \mathbf{u}\mathbf{v}^T$, where $\mathbf{u}$ and $\mathbf{v}$ are any nonzero vectors.

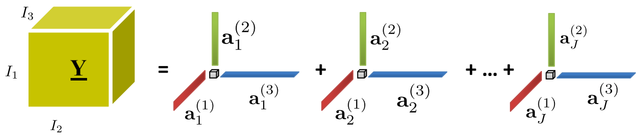


Figure 1: Illustration of CP decompositions of a 3rd-order tensor $\underline{\mathbf{Y}} \in \mathbb{R}^{I_1 \times I_2 \times I_3}$ (ignored the noise), where the factors $\mathbf{A}^{(n)} = [\mathbf{a}_1^{(n)}, \mathbf{a}_2^{(n)}, \ldots, \mathbf{a}_J^{(n)}] \in \mathbb{R}^{I_n \times J}$ contain the latent component $\mathbf{a}_j^{(n)}$ as their columns, $n = 1, 2, 3$.

where matrices $\mathbf{A}^{(n)} = [\mathbf{a}_1^{(n)}, \mathbf{a}_2^{(n)}, \ldots, \mathbf{a}_J^{(n)}] \in \mathbb{R}^{I_n \times J}$, $n = 1, 2, \ldots, N$, consist of unknown components $\mathbf{a}_j^{(n)}$, e.g. latent source signals, $\circ$ denotes the outer product, and $\underline{\mathbf{E}}$ is the fitting error, see Figure 1 for the illustration of CP decompositions of a 3rd-order tensor. From (1), the higher-order tensor is represented as a linear combination of outer products of vectors (i.e., rank one tensors), which can be regarded as a generalization of the matrix singular value decomposition (SVD) to tensors [2]. As the scalar factors λ_j can be absorbed into a factor (or loading) matrix $\mathbf{A}^{(N)}$ by letting $\mathbf{a}_j^{(N)} = \lambda_j \mathbf{a}_j^{(N)}, \forall j$, we also use $\underline{\mathbf{Y}} \approx [\![\mathbf{A}^{(1)}, \mathbf{A}^{(2)}, \cdots, \mathbf{A}^{(N)}]\!]$ as a shorthand notation of (1).

To solve CP decomposition problems, alternating least square (ALS) methods are widely employed. Consider the mode-n matricization of $\underline{\mathbf{Y}}$:

$$\mathbf{Y}_{(n)} = \mathbf{A}^{(n)} \mathbf{B}^{(n)T}, \; n = 1, 2, \ldots, N, \tag{2}$$

where

$$\mathbf{B}^{(n)} = \bigodot_{k \neq n} \mathbf{A}^{(k)}. \tag{3}$$

In standard ALS methods, factor matrices $\mathbf{A}^{(n)}$ are updated as $\mathbf{Y}_{(n)}[\mathbf{B}^{(n)T}]^{\dagger}$ sequentially for $n = 1, 2, \ldots, N$, where '$\dagger$' denotes the Moore-Penrose pseudo inverse of matrices. As the matrix $\mathbf{B}^{(n)}$ is often quite huge, some tricks were proposed to simplify the computation of $\mathbf{Y}_{(n)}[\mathbf{B}^{(n)T}]^{\dagger}$ [2]. ALS methods need to unfold the tensor frequently and often converge slowly.

B. *N-way CP Decompositions Based On A Preceding 3-way Decomposition*

Based on the results in [6], once one factor has been correctly estimated, all the remaining factors can be uniquely and efficiently computed via a series of SVD of rank-1 matrices. In [6], we used BSS methods to estimate this factor, provided that some *a priori* knowledge on the components in this factor is available and thanks to the essential uniqueness of BSS. Unfortunately, sometimes we have no such *a priori* knowledge available. To benefit from the advantages of the scheme proposed in [6], we consider to perform CP decompositions on a much smaller tensor first to estimate only partial factors. The uniqueness of CP decompositions guarantees the correctness of estimated partial factors, if the CP model fits well to the data. Then the remaining factors will be estimated by using, for example, the method described in [6]. For convenience, we consider to extract the factor matrix $\mathbf{A}^{(1)}$ and $\mathbf{A}^{(2)}$ first[2]. Consider the mode-1 matricization of $\underline{\mathbf{Y}}$:

$$\begin{aligned} \mathbf{Y}_{(1)} &= \mathbf{A}^{(1)}[\mathbf{A}^{(N)} \odot \cdots \odot \mathbf{A}^{(3)} \odot \mathbf{A}^{(2)}]^T \\ &= \mathbf{A}^{(1)}[\mathbf{G} \odot \mathbf{A}^{(2)}]^T, \end{aligned} \tag{4}$$

where

$$\mathbf{G} \doteq \mathbf{A}^{(N)} \odot \mathbf{A}^{(N-1)} \odot \cdots \odot \mathbf{A}^{(3)} \in \mathbb{R}^{I_r \times J} \tag{5}$$

and

$$I_r = \prod_{k \neq \{1,2\}} I_k. \tag{6}$$

Comparing (4) and (2), we observe that $\mathbf{Y}_{(1)}$ can be viewed as the mode-1 matricization of a 3rd-order tensor $\underline{\mathbf{Y}}_3$ such that

$$\underline{\mathbf{Y}}_3 = \sum_{j=1}^{J} \lambda_j \mathbf{a}_j^{(1)} \circ \mathbf{a}_j^{(2)} \circ \mathbf{g}_j \in \mathbb{R}^{I_1 \times I_2 \times I_r}, \tag{7}$$

where $\mathbf{g}_j$ are the jth column of $\mathbf{G}$. This motivates us to perform 3-way CP decompositions on the tensor $\underline{\mathbf{Y}}_3$ to estimate the factor $\mathbf{A}^{(1)}$ and $\mathbf{A}^{(2)}$ first. Let Ψ denote a CP algorithm such that

$$(\widehat{\mathbf{A}}^{(1)}, \widehat{\mathbf{A}}^{(2)}, \widehat{\mathbf{G}}) = \Psi(\underline{\mathbf{Y}}_3). \tag{8}$$

Here we assume that the corresponding 3-way CP decomposition is essentially unique. Consequently, there holds that

$$\widehat{\mathbf{A}}^{(1)} = \mathbf{A}^{(1)}\mathbf{P}\mathbf{D}_1; \widehat{\mathbf{A}}^{(2)} = \mathbf{A}^{(2)}\mathbf{P}\mathbf{D}_2; \widehat{\mathbf{G}} = \mathbf{G}\mathbf{P}\mathbf{D}_3, \tag{9}$$

where $\mathbf{P}$ and $\mathbf{D}_n$ $(n = 1, 2, 3)$, are a permutation matrix and diagonal matrices, respectively, with $\prod_{n=1}^{3} \mathbf{D}_n = \mathbf{I}$.

We define matrices $\widehat{\mathbf{A}}^{(n)}$ such that

$$\widehat{\mathbf{A}}^{(n)} = \begin{cases} \mathbf{A}^{(n)}\mathbf{P}\mathbf{D}_3^{-1} & n = N, \\ \mathbf{A}^{(n)}\mathbf{P} & n = 1, 2, \ldots, N-1. \end{cases} \tag{10}$$

It can be observed that there are only permutation and scaling ambiguities between $\mathbf{A}^{(n)}$ and $\widehat{\mathbf{A}}^{(n)}$, $\forall n$. From (5), (9) and (10), we have

$$\bigodot_{n=N}^{3} \widehat{\mathbf{A}}^{(k)} = \bigodot_{n=N}^{3} \mathbf{A}^{(k)}\mathbf{P}\mathbf{D}_3^{-1} = \widehat{\mathbf{G}}. \tag{11}$$

In the following we show that such $\widehat{\mathbf{A}}^{(n)}$ are actually unique, which naturally leads to the essential uniqueness of $\mathbf{A}^{(n)}$. We have two ways to see it.

[2] Nevertheless In general, we can actually estimate any two factors by re-permuting the modes of the tensor.

1) Sequential Extraction: We simply follow the procedure described in [6]. From (11), we have

$$\bigodot_{n=N}^{3} \widehat{\mathbf{a}}_j^{(n)} = \widehat{\mathbf{g}}_j, \tag{12}$$

where $\bigodot_{n=N}^{3} \widehat{\mathbf{a}}_j^{(n)}$ and $\widehat{\mathbf{g}}_j$ are the jth column of $\bigodot_{n=N}^{3} \widehat{\mathbf{A}}^{(n)}$ and the already known matrix $\widehat{\mathbf{G}}$, respectively. Construct a matrix $\widehat{\mathbf{G}}_j^{(N)}$ by reshaping $\widehat{\mathbf{g}}_j$ such that

$$\widehat{\mathbf{G}}_j^{(N)} = \widehat{\mathbf{a}}_j^{(N)} [\widehat{\mathbf{g}}_j^{(N-1)}]^T \in \mathbb{R}^{I_N \times I_r^{'}}, \tag{13}$$

where $I_r^{'} = \prod_{k \neq \{1,2,N\}} I_k$, and $\widehat{\mathbf{g}}_j^{(N-1)} = \bigodot_{k=N-1}^{3} \widehat{\mathbf{a}}_j^{(k)}$. From (13) we see that $\widehat{\mathbf{G}}_j^{(N)}$ is a rank-1 matrix, and $\widehat{\mathbf{a}}_j^{(N)}$, $\widehat{\mathbf{g}}_j^{(N-1)}$ are just its left and right singular vectors corresponding to the *unique* leading singular value, respectively. Consequently, the jth column of $\widehat{\mathbf{A}}^{(N)}$, i.e., $\widehat{\mathbf{a}}_j^{(N)}$, can be uniquely estimated first from the leading left singular vector of the rank-1 matrix $\widehat{\mathbf{G}}_j^{(N)}$. Then we reshape $\widehat{\mathbf{g}}_j^{(N-1)}$ such that

$$\widehat{\mathbf{G}}_j^{(N-1)} = \widehat{\mathbf{a}}_j^{(N-1)} [\widehat{\mathbf{g}}_j^{(N-2)}]^T, \tag{14}$$

where $\widehat{\mathbf{g}}_j^{(N-2)} = \bigodot_{k=N-2}^{3} \widehat{\mathbf{a}}_j^{(k)}$. Again, $\widehat{\mathbf{G}}_j^{(N-1)}$ is rank-1 and the jth column of $\widehat{\mathbf{A}}^{(N-1)}$, i.e., $\widehat{\mathbf{a}}_j^{(N-1)}$ is its leading left singular vector. Generally, by running truncated SVD on the rank-1 matrix formed by the reshaped right singular vector $\widehat{\mathbf{g}}_j^{(n)}$, we can estimate all $\widehat{\mathbf{a}}_j^{(n)}$ from the leading left singular vectors sequentially for $n = N, N-1, \ldots, 3$. By this way, the jth columns of all component matrices $\widehat{\mathbf{A}}^{(n)}$, i.e., $\widehat{\mathbf{a}}_j^{(n)}$, $n \geq 3$, are obtained.

Repeat the above procedure for $j = 1, 2, \ldots, J$, all the columns of $\widehat{\mathbf{A}}^{(n)}$ for $n \geq 3$ can be estimated. Note that each time the matrix $\widehat{\mathbf{G}}_j^{(n)}$ is of rank-1 and thus the leading left and right singular vector are essentially unique, thereby leading to the essential uniqueness of $\widehat{\mathbf{A}}^{(n)}$, $n \geq 3$. From (9) and (10), all the factors $\mathbf{A}^{(n)}$ are recovered as $\widehat{\mathbf{A}}^{(n)}$ with only unavoidable uncertainty of scaling and permutation of their columns.

Note that the matrix $\hat{\mathbf{G}}_j^{(n)}$ becomes gradually smaller with $n = N, N-1, \ldots$. This makes the method quite efficient. Moreover, each column of $\hat{\mathbf{G}}$ is treated separately, which makes it possible to recover the columns in a parallel way.

2) Independent Extraction: From (13), $\widehat{\mathbf{G}}_j^{(N)}$ can be viewed as the mode-(N-2) matricization of the tensor

$$\underline{\hat{\mathbf{G}}}^j = \hat{\mathbf{a}}_j^{(3)} \circ \hat{\mathbf{a}}_j^{(4)} \circ \cdots \circ \hat{\mathbf{a}}_j^{(N)}. \tag{15}$$

Obviously, we can employ any CP decomposition algorithm to decompose $\underline{\hat{\mathbf{G}}}^j$ to estimate $\hat{\mathbf{a}}_j^{(n)}$. Note that $\underline{\hat{\mathbf{G}}}^j$ is actually a rank-1 tensor. Consider its mode-($n-2$) matricization

$$\hat{\mathbf{G}}_{(n-2)}^j = \hat{\mathbf{a}}_j^{(n)} \left[\bigodot_{k \neq \{n,1,2\}} \hat{\mathbf{a}}_j^{(k)} \right]^T, \quad n = 3, 4, \ldots, N, \tag{16}$$

which means that $\hat{\mathbf{a}}_j^{(n)}$ is just the left singular vector associated with the unique largest singular value of $\hat{\mathbf{G}}_{(n-2)}^j$. Hence $\hat{\mathbf{a}}_j^{(n)}$ can be computed from the truncated SVD of $\hat{\mathbf{G}}_{(n-2)}^j$ directly. Note that in this way all the columns of $\hat{\mathbf{A}}^{(n)}$, $n \neq 1, 2$, are estimated uniquely and independently. Consequently, this way can benefit much from parallel computations.

See Algorithm 1 for the pseudo code of the proposed method. We call the new method N3CP as it is based on a preceding 3-way CP decomposition.

Algorithm 1 The N3CP Algorithm

```
Require: $\underline{\mathbf{Y}}$, $J$, and a CP algorithm $\Psi$.
Construct the 3rd-order tensor $\underline{\mathbf{Y}}_3$ by reshaping $\underline{\mathbf{Y}}$ according to (7).
Let$(\widehat{\mathbf{A}}^{(1)}, \widehat{\mathbf{A}}^{(2)}, \widehat{\mathbf{G}}) = \Psi(\underline{\mathbf{Y}}_3)$.
for $j = 1$ to $J$ do
    Estimate $\hat{\mathbf{a}}_j^{(n)}$ by recovering the Khatri-Rao structure of $\widehat{\mathbf{g}}_j$ (or by decomposing the rank-1 tensor $\underline{\hat{\mathbf{G}}}^j$ defined by (15)), for $n = 3, 4, \ldots, N$.
end for
return $\widehat{\mathbf{A}}^{(n)}, n = 1, 2, \ldots, N$.
```

III. Efficient 3-way CP Decomposition

In the N3CP method, the first key step is performing a 3-way CP decomposition on the rearranged tensor $\underline{\mathbf{Y}}_3 \in \mathbb{R}^{I_1 \times I_2 \times I_{\bar{3}}}$ with $I_{\bar{3}} = \prod_{k \neq \{1,2\}} I_k$. For high-order tensors I_r is often huge and $I_{\bar{3}} \gg \max(I_1, I_2)$. Consider the mode-3 matricization of $\underline{\mathbf{Y}}_3$

$$\mathbf{Y}_{3(3)} = \mathbf{G}[\mathbf{A}^{(2)} \odot \mathbf{A}^{(1)}]^T. \tag{17}$$

As $\mathbf{G}$ is significantly larger than $\mathbf{A}^{(1)}$ and $\mathbf{A}^{(2)}$, we consider replacing $\mathbf{G}$ by a much smaller matrix to simplify the computational complexity of the 3-way CP decompositions. After $\mathbf{A}^{(1)}$ and $\mathbf{A}^{(2)}$ have been estimated, we let

$$\mathbf{G} = \mathbf{Y}_{3(3)}[\mathbf{A}^{(2)} \odot \mathbf{A}^{(1)}]^{T\dagger}. \tag{18}$$

Here we introduce two alternative methods to reduce the size of $\mathbf{G}$:

1) **Truncated SVD.** Note that $\mathbf{Y}_{3(3)}$ is of rank-J in noiseless cases. Let $\mathbf{Y}_{3(3)} = \mathbf{U}\mathbf{D}\mathbf{V}^T$ be the truncated SVD of $\mathbf{Y}_{3(3)}$ where $\mathbf{D} \in \mathbb{R}^{J \times J}$ is a diagonal matrix whose diagonal elements consist of the leading J singular values. Then $\mathbf{Y}_{3(3)}$ is updated by the singular matrix $\mathbf{V}$, which is equivalent to replacing the factor $\mathbf{G}$ by $\mathbf{D}^{-1}\mathbf{U}^T\mathbf{G}$. Then we reshape the new matrix $\mathbf{Y}_{3(3)}$ to form a reduced tensor $\underline{\mathbf{Y}}_3 \in \mathbb{R}^{I_1 \times I_2 \times J}$, which is often significantly smaller than the original 3rd-order tensor. This can significantly reduce the computational complexity of 3rd-order tensor decompositions.

2) **Sampling.** Sometimes we hope to maintain the physical meaning of the original data, e.g., nonnegativity[3]. In this case we can achieve dimensionality reduction by sampling the rows of the unfolded data matrix $\mathbf{Y}_{3(3)}$ [8], which is equivalent to sampling the rows of $\mathbf{G}$. We randomly sample p rows (empirically we select $p \geq 5J$, see, e.g., [1]) from $\mathbf{Y}_{3(3)}$ to form the smaller 3rd-order tensor.

After we construct the 3rd-order tensor $\underline{\mathbf{Y}}_3$, we can employ any efficient CP decomposition methods to estimate $\mathbf{A}^{(1)}$ and $\mathbf{A}^{(2)}$ and then use (18) to compute the original $\mathbf{G}$. Finally the remaining factors are estimated by using the Khatri-Rao product recovering methods described in Section II.B.

It is worth noticing that the proposed approach is also an efficient CP decomposition methods for 3rd-order tensors incorporating dimensionality reduction techniques. In practice, one mode, say $\mathbf{G} = \mathbf{A}^{(3)}$, can be of extremely large size. We can reduce the size of $\mathbf{G}$ first and then estimate $\mathbf{A}^{(3)}$ from (18). In this way, we can achieve a trade-off between accuracy and efficiency.

IV. Simulations

Two performance indices (PI) were used to evaluate the performance of the proposed algorithm. The first one is the signal-to-interference ratio (SIR), which is defined by

$$\mathrm{SIR}(\mathbf{a}, \widehat{\mathbf{a}}) = 10 \log_{10} \frac{\sum_t a_t^2}{\sum_t (a_t - \widehat{a}_t)^2}, \tag{19}$$

where $\mathbf{a}$, $\widehat{\mathbf{a}}$ are normalized random variables with zero mean and unit variance, and $\widehat{\mathbf{a}}$ is an estimate of $\mathbf{a}$. The value of SIR reflects how well the estimated component (source) match the true original one. The second PI measures the fit of the estimated tensor to the original raw data tensor which is defined as

$$\mathrm{Fit}(\underline{\mathbf{Y}}, \widehat{\underline{\mathbf{Y}}}) = 1 - \frac{\|\underline{\mathbf{Y}} - \widehat{\underline{\mathbf{Y}}}\|_F}{\|\underline{\mathbf{Y}}\|_F}, \tag{20}$$

where $\widehat{\underline{\mathbf{Y}}}$ is an estimate of $\underline{\mathbf{Y}}$, $\mathrm{Fit}(\underline{\mathbf{Y}}, \widehat{\underline{\mathbf{Y}}}) = 1$ if and only if $\widehat{\underline{\mathbf{Y}}} = \underline{\mathbf{Y}}$. For synthetic data, $\underline{\mathbf{Y}}$ denotes the original noiseless data instead the noisy data in order to evaluate how robust of the proposed method in respect to additive noise. All the experiments were done in MATLAB 2008a on a computer with Intel 7i 3.33GHz CPU and 24GB memory running Windows 7.

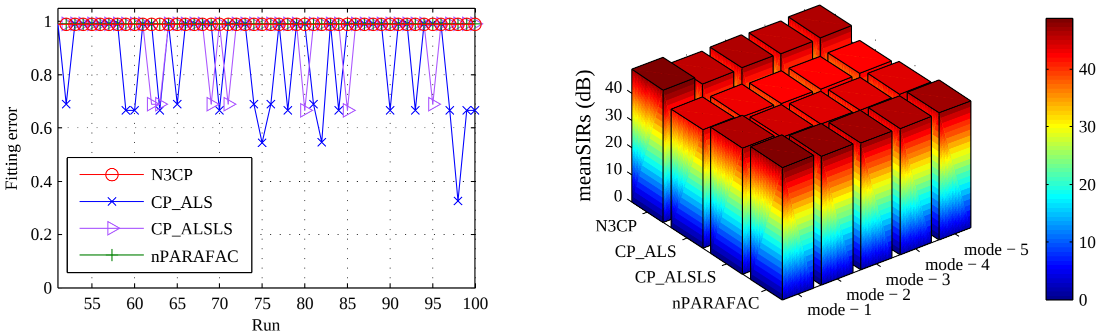
Simulation 1: We generated a 5th-order tensor $\underline{\mathbf{Y}} \in \mathbb{R}^{21 \times 22 \times 23 \times 24 \times 25}$ using the CP model. The elements of each factor $\mathbf{A}^{(n)}$ were drawn from independent standard normal distributions. Finally, independent and very heavy Gaussian noises with SNR=0dB was added to the observation tensor. The proposed method was compared with the standard CP method based on ALS iterations (CP-ALS) included in [5], the CP-ALS combined with line search (CP-ALSLS) [10], and the PARAFAC algorithm included in the N-Way tensor toolbox for MATLAB [11](nPARAFAC, ver. 3.20). In CP-ALS, the maximum iteration number was 500. In the N3CP method, we constructed a 3rd-order tensor $\underline{\mathbf{Y}}_3$ with respect to mode-1 and mode-2. Then, we used the SWATLD method [12] to decompose the tensor $\underline{\mathbf{Y}}_3$ to estimate the first two factors and $\mathbf{G}$. Finally, we used SVD to retrieve the remaining factors. Their performance over 100 Monte Carlo runs was illustrated in Figure 2. It can be seen that in the most cases all the methods achieved almost the same fit. Unfortunately, CP-ALS and CP-ALSLS often stuck in local minima (33% and 14%, respectively). Also, they converged very slow. Both N3CP and nPARAFAC achieved satisfying fit of data, but N3CP was much faster than the nPARAFAC algorithm.

Table I: Performance comparison of the four CPD methods when they were applied to real image data clustering. The N3CP method was significantly faster than the other algorithms and achieved better clustering accuracy.

Algorithm	Fit	Runtime (s)	Accuracy
N3CP	0.58	7.0	61.6%±1.8
CP-ALS	0.58	356.3	52.1%±3.9
CP-ALSLS	0.58	593.3	49.0%±1.6
nPARAFAC	0.58	616.9	49.6%±1.7

Simulation 2: We applied the above methods to real image data analysis, namely the COIL-100 database [13]. The COIL-100 database consists of 7200 color images of 100 objects, and 72 images per object which were taken from 72 different angles. For simplicity, we selected the first 20 objects for test, and each image was scaled with the size of 128×128. Then, we had a tensor $\underline{\mathbf{Y}}$ with the size of $128 \times 128 \times 3 \times 1440$, where the number 3 denotes the RGB channels. We set rank $J = 3$ for all the methods. In the N3CP we used the SWATLD method to perform 3-way CP decompositions to estimate the mode-1 ($\mathbf{A}^{(1)}$) and mode-4 ($\mathbf{A}^{(4)}$) factors (we used truncated SVD to reduce the size of the sub-tensor). For the other methods the maximum iteration number was 100. Finally, we used the factor $\mathbf{A}^{(4)}$ as features to cluster the original images. As K-means is prone to be influenced by initial centers of clusters, we replicated K-means 20 times for each method. See TABLE I for their performance over 20 Monte Carlo runs. From the table, we see that the N3CP method not only converged much faster than the CP-ALS method but also achieved better clustering accuracy.

[3] However, it does not mean that the first method cannot be used for nonnegative data analysis. Due to the essentially uniqueness of CP decompositions, the resulting factors $\mathbf{A}^{(1)}$ and $\mathbf{A}^{(2)}$ are essentially unique and hence can be nonnegative after adjusting the signs of their columns accordingly, no matter whether $\underline{\mathbf{Y}}_3$ and $\mathbf{G}$ are negative or not. See also [7] for a related treatment.



(a) Fit

(b) Mean SIR

Figure 2: Performance comparison over 100 Monte Carlo runs when they were applied to decompose a 5th-order tensor with 0dB additive noise. It can be observed that the N3CP and nPARAFAC are more robust with respect to additive Gaussian noise than CP-ALS and CP-ALSLS. Moreover, N3CP is much faster than the other algorithms.

V. Conclusion

Existing CP decomposition methods are often based on alternating least square (ALS) iterations. In ALS methods, we have to compute inverse of matrices and unfold the observation tensor frequently, which are the major bottlenecks of high efficiency. In our previous study we showed that CP decomposition can be solved efficiently as soon as at least one factor has been correctly estimated. Based on this result, in this paper we proposed a new method to perform CP decompositions of high order tensors. We perform a 3-way CP decomposition on a much smaller sub-tensor to estimate two pre-selected factors first, thanks to the essential uniqueness of CP decompositions. After that, the remaining factors are efficiently computed via a Khatri-Rao structure recovering procedure which can be achieved by a series of SVD of rank-1 matrices. In this way, frequently unfolding to each mode matrix is avoided. Simulations confirmed the efficiency and validity of the proposed method.

References

[1] A. Cichocki, R. Zdunek, A.-H. Phan, and S. Amari, *Nonnegative Matrix and Tensor Factorizations: Applications to Exploratory Multi-way Data Analysis and Blind Source Separation*. Chichester: Wiley, 2009.

[2] T. G. Kolda and B. W. Bader, “Tensor decompositions and applications,” *SIAM Review*, vol. 51, no. 3, pp. 455–500, 2009.

[3] J. Carroll and J.-J. Chang, “Analysis of individual differences in multidimensional scaling via an N-way generalization of Eckart-Young decomposition,” *Psychometrika*, vol. 35, no. 3, pp. 283–319, September 1970.

[4] R. A. Harshman, “Foundations of the PARAFAC procedure: Models and conditions for an ’explanatory’ multi-modal factor analysis,” *UCLA Working Papers in Phonetics*, vol. 16, no. 1, 1970.

[5] B. W. Bader and T. G. Kolda, “MATLAB tensor toolbox version 2.5,” Feb. 2012. [Online]. Available: http://csmr.ca.sandia.gov/~tgkolda/TensorToolbox/

[6] G. Zhou and A. Cichocki, “Canonical polyadic decomposition based on a single mode blind source separation,” *IEEE Signal Processing Letters*, vol. 19, no. 8, pp. 523 –526, Aug. 2012.

[7] G. Zhou, A. Cichocki, and S. Xie, “Fast nonnegative matrix/tensor factorization based on low-rank approximation,” *IEEE Transactions on Signal Processing*, vol. 60, no. 6, pp. 2928 –2940, June 2012.

[8] M. W. Mahoney and P. Drineas, “CUR matrix decompositions for improved data analysis,” *Proceedings of the National Academy of Sciences*, vol. 106, no. 3, pp. 697–702, Jan. 2009.

[9] D. Nion and L. De Lathauwer, “An enhanced line search scheme for complex-valued tensor decompositions. application in DS-CDMA,” *Signal Processing*, vol. 88, no. 3, pp. 749 – 755, 2008.

[10] C. A. Andersson and R. Bro, “The N-way toolbox for MATLAB,” 2000. [Online]. Available: http://www.models.life.ku.dk/source/nwaytoolbox/

[11] Z.-P. Chen, H.-L. Wu, and R.-Q. Yu, “On the self-weighted alternating trilinear decomposition algorithm – the property of being insensitive to excess factors used in calculation,” *Journal of Chemometrics*, vol. 15, no. 5, pp. 439 – 453, 2001.

[12] S. A. Nene, S. K. Nayar, and H. Murase, “Columbia object image library (COIL-100),” 1996. [Online]. Available: http://www.cs.columbia.edu/CAVE/software/softlib/coil-100.php