# Canonical Polyadic Decomposition Based on a Single Mode Blind Source Separation

Guoxu Zhou and Andrzej Cichocki

***Abstract*—A new canonical polyadic (CP) decomposition method is proposed in this letter, where one factor matrix is extracted first by using any standard blind source separation (BSS) method and the remainder components are computed efficiently via sequential singular value decompositions of rank-1 matrices. The new approach provides more interpretable factors and it is extremely efficient for ill-conditioned problems. Especially, it overcomes the bottleneck problems, which often cause very slow convergence speed in CP decompositions. Simulations confirmed the validity and efficiency of the proposed method.**

***Index Terms*—CP (PARAFAC) decompositions, blind source separation, tensor decompositions, bottleneck problem.**

## I. Introduction

Recently, 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].

As one of the most important tensor decomposition models, Canonical Polyadic (CP), also named as CANDECOMP/PARAFAC decomposition [2], [3] has been extensively studied and found many practical applications. However, several major problems exist in CP decompositions. First, there may not exist an optimal CP decomposition at all (which achieves the minimum fitting error) [4]. This often causes slow convergence (even divergence) problem and hardly interpretable results [5]. Even if an optimal solution exists, it may be not unique for higher-order tensors. The well-known uniqueness condition was given by Kruskal in 1977 for 3-way tensors and then extended for $N$-way tensors by Sidiropoulos and Bro in 2000 [6], [7]. These uniqueness conditions are mainly based on the algebraic properties such as rank of matrices and tensors. Unfortunately, there is no practical way so far to deal with the problems where the decompositions are not essentially unique. Moreover, even theoretically an optimal solution exists and is unique, some components can be highly collinear and in such case existing methods suffer from a very slow convergence speed. In this letter we attempt to overcome these problems by incorporating *a priori* knowledge about the components in only one mode.

The following notations will be adopted. Underlined bold capitals, e.g. $\underline{\mathbf{Y}}$, denote tensors, and $\mathbf{Y}_{(n)}$ denotes the mode-$n$ unfolding (matricization) of $\underline{\mathbf{Y}}$. Bold capitals (e.g., $\mathbf{A}$) and bold lowercase letters (e.g., $\mathbf{y}$) denote matrices and vectors, respectively. $\odot$ denotes the Khatri-Rao product (columnwise Kronecker product) of matrices and $\odot_{k\neq n}\mathbf{A}^{(k)} = \mathbf{A}^{(N)}\odot\cdots\odot\mathbf{A}^{(n+1)}\odot\mathbf{A}^{(n-1)}\odot\cdots\odot\mathbf{A}^{(1)}$. Readers are referred to [8] for the notations and tensor operations.

## II. 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 formally formulated as

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

where matrices $\mathbf{A}^{(n)} = [\mathbf{a}_1^{(n)},\mathbf{a}_2^{(n)},\cdots,\mathbf{a}_J^{(n)}] \in \mathbb{R}^{I_n\times J}$, $n=1,2,\cdots,N$, consists 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. From (1), the multiway 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 [8]. As the scalar factors $\lambda_j$ can be absorbed into a 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 $\underline{\mathbf{Y}}$:

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

where

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

In standard ALS methods, factor matrices $\mathbf{A}^{(n)}$ are updated as $\mathbf{Y}_{(n)}[\mathbf{B}^{(n)T}]^{\dagger}$ sequentially for $n=1,2,\cdots,N$, where $^{\dagger}$ denotes the Moore-Penrose pseudo inverse of a matrix. ALS methods need to unfold the tensor frequently and often converge slowly. Some authors have proposed various methods (see, e.g., [9], [10]) to improve the performance of CP decomposition algorithms. However, for the *bottleneck* problem, which occurs when two or more components in at least one mode are highly collinear [9], there is still a lack of very efficient and reliable algorithm.

Manuscript received _____.

Guoxu Zhou is with the Laboratory for Advanced Brain Signal Processing, RIKEN, Brain Science Institute, Wako-shi, Saitama 3510198, Japan. E-mail: zhouguoxu@brain.riken.jp.

Andrzej Cichocki is with the RIKEN BSI, Japan and Systems Research Institute, Warsaw Poland. E-mail: cia@brain.riken.jp.

## III. CP Decomposition Based on A Single Mode BSS

Most ALS algorithms minimize the fitting error and use algebraic properties of factors. Except the basic algebraic properties, very often some *a priori* knowledge on the components in specific modes is available. This happens, for example, when the components in a specific mode have very clear physical meaning, e.g., they are independent, and/or sparse, nonnegative or have temporal structures. For example, in brain signal processing spectral components are nonnegative and usually sparse, while temporal components are often statistically independent. By incorporating this type of knowledge, the latent components can often be successfully extracted from their arbitrary linear mixtures by using suitable constrained matrix factorization methods such as independent component analysis (ICA), sparse component analysis, and nonnegative matrix factorization (NMF) [11]. Here they are uniformly referred to as blind source separation (BSS). Consider the linear instantaneous model of BSS

$$\mathbf{X} = \mathbf{S}\mathbf{M}^T, \tag{4}$$

where the columns of $\mathbf{S} \in \mathbb{R}^{I\times J}$ consists of $J$ latent source signals (components) and $\mathbf{M}$ denotes a mixing matrix. BSS is quite powerful because it allows us to retrieve the latent source signals $\mathbf{S}$ from their mixtures $\mathbf{X}$ only under very mild conditions, without any knowledge of the mixing matrix $\mathbf{M}$. Let $\Psi$ denote a suitable BSS method. Then there holds that

$$\widehat{\mathbf{S}} = \Psi(\mathbf{X}) = \mathbf{S}\mathbf{P}\mathbf{D}, \tag{5}$$

where $\mathbf{P}$ and $\mathbf{D}$ are $J \times J$ permutation matrix and invertible diagonal matrix, respectively, which denote the unavoidable ambiguity of permutation and scaling. BSS techniques attracted intensive study in the past decades and a lot of efficient BSS methods have been proposed so far based on various assumptions (i.e., *a priori* knowledge) [11].

From (2), the columns of matrix $\mathbf{Y}_{(n)}$ are just linear mixtures of the columns of $\mathbf{A}^{(n)}$. This motivates us to extract $\mathbf{A}^{(n)}$ first by using a proper BSS algorithm $\Psi$ such that

$$\widehat{\mathbf{A}}^{(n)} = \Psi(\mathbf{Y}_{(n)}) = \Psi(\mathbf{A}^{(n)}\mathbf{B}^{(n)T}) = \mathbf{A}^{(n)}\mathbf{P}\mathbf{D}. \tag{6}$$

By using BSS, the factor $\mathbf{A}^{(n)}$ can be recovered as $\widehat{\mathbf{A}}^{(n)}$ up to scaling and permutation ambiguity of columns without any knowledge of the mixing matrix $\mathbf{B}^{(n)}$. After $\mathbf{A}^{(n)}$ has been estimated, the following proposition shows that the remainder component matrices can be computed easily and uniquely:

**Proposition:** Let $\underline{\mathbf{Y}}=[\![\mathbf{A}^{(1)}, \mathbf{A}^{(2)}, \cdots, \mathbf{A}^{(N)}]\!]$ be a CP decomposition of $\underline{\mathbf{Y}}$. For any arbitrary $n$, if $\mathbf{A}^{(n)}$ has been estimated by a suitable BSS method in (6), then all the other component matrices $\mathbf{A}^{(k)}$, $(k \neq n)$, are essentially unique and can be computed via a series of SVD of rank-1 matrices.

*Proof:* Let $\widehat{\mathbf{A}}^{(n)}$ be an estimate of $\mathbf{A}^{(n)}$ such that $\widehat{\mathbf{A}}^{(n)} = \mathbf{A}^{(n)}\mathbf{P}\mathbf{D}$. Substitute $\mathbf{A}^{(n)}$ into (2) and we have

$$\mathbf{B}^{(n)} = \widehat{\mathbf{B}}^{(n)}\mathbf{D}\mathbf{P}^T, \tag{7}$$

where

$$\widehat{\mathbf{B}}^{(n)} = \mathbf{Y}_{(n)}^T[\widehat{\mathbf{A}}^{(n)T}]^\dagger \in \mathbb{R}^{(\prod_{k\neq n} I_k)\times J} \tag{8}$$

is fixed as it only depends on the observation data $\mathbf{Y}_{(n)}$ and the estimated matrix $\widehat{\mathbf{A}}^{(n)}$. From (3) and (7), we have

$$\mathbf{B}^{(n)}\mathbf{P}\mathbf{D}^{-1} = [\odot_{k\neq n}\mathbf{A}^{(k)}]\mathbf{P}\mathbf{D}^{-1} = \widehat{\mathbf{B}}^{(n)}. \tag{9}$$

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

$$\widehat{\mathbf{A}}^{(k)} = \mathbf{A}^{(k)}\mathbf{P},\ \forall k \neq n; \quad \widehat{\mathbf{A}}^{(N)} = \mathbf{A}^{(N)}\mathbf{D}^{-1}. \tag{10}$$

From (9) and (10), we have

$$[\odot_{k\neq n}\widehat{\mathbf{A}}^{(k)}] = [\odot_{k\neq n}\mathbf{A}^{(k)}]\mathbf{P}\mathbf{D}^{-1} = \widehat{\mathbf{B}}^{(n)}. \tag{11}$$

In the following we show such $\widehat{\mathbf{A}}^{(k)}$ are actually unique, which naturally leads to the essential uniqueness of $\mathbf{A}^{(k)}$. From (11), we have

$$[\odot_{k\neq n}\widehat{\mathbf{a}}_j^{(k)}] = \widehat{\mathbf{h}}_j^{(N)}, \tag{12}$$

where $[\odot_{k\neq n}\widehat{\mathbf{a}}_j^{(k)}]$ and $\widehat{\mathbf{h}}_j^{(N)}$ are the $j$th column of $[\odot_{k\neq n}\widehat{\mathbf{A}}^{(k)}]$ and the already known matrix $\widehat{\mathbf{B}}^{(n)}$ , respectively. Construct a matrix $\widehat{\mathbf{H}}_j^{(N)}$ by reshaping $\widehat{\mathbf{h}}_j^{(N)}$ such that

$$\widehat{\mathbf{H}}_j^{(N)} = \widehat{\mathbf{a}}_j^{(N)}[\widehat{\mathbf{h}}_j^{(N-1)}]^T \in \mathbb{R}^{I_N\times I_r}, \tag{13}$$

where $I_r = \prod_{k\neq\{n,N\}} I_k$, and $\widehat{\mathbf{h}}_j^{(N-1)} = [\odot_{k\neq\{n,N\}}\widehat{\mathbf{a}}_j^{(k)}]$. From (13) we see that $\widehat{\mathbf{H}}_j^{(N)}$ is a rank-1 matrix, and $\widehat{\mathbf{a}}_j^{(N)}$ and $\widehat{\mathbf{h}}_j^{(N-1)}$ are just its left and right singular vectors corresponding to the *unique* leading singular value, respectively. Consequently, the $j$th 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 $\widehat{\mathbf{H}}_j^{(N)}$. Then reshape $\widehat{\mathbf{h}}_j^{(N-1)}$ such that

$$\widehat{\mathbf{H}}_j^{(N-1)} = \widehat{\mathbf{a}}_j^{(N-1)}[\widehat{\mathbf{h}}_j^{(N-2)}]^T. \tag{14}$$

where $\widehat{\mathbf{h}}_j^{(N-2)} = [\odot_{k\neq\{n,N,N-1\}}\widehat{\mathbf{a}}_j^{(k)}]$. Again, $\widehat{\mathbf{H}}_j^{(N-1)}$ is rank-1 and the $j$th 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 $\widehat{\mathbf{h}}_j^{(p)}$, we can estimate all $\widehat{\mathbf{a}}_j^{(p)}$ from the leading left singular vectors sequentially for $p = N, \cdots, n+1, n-1, \cdots, 1$. By this way, the $j$th columns of all component matrices $\widehat{\mathbf{A}}^{(p)}$, i.e., $\widehat{\mathbf{a}}_j^{(p)}$, $p \neq n$, are obtained.

Repeat the above procedure for $j = 1, 2, \cdots, J$, all the columns of $\widehat{\mathbf{A}}^{(k)}$ for $p \neq n$ can be estimated. Note that each time the matrix $\widehat{\mathbf{H}}_j^{(p)}$ is of rank-1 and thus the leading left and right singular vector are essentially unique, thereby leading to the essential uniqueness of $\widehat{\mathbf{A}}^{(p)}$, $p \neq n$. From (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. ■

In summary, after the components in a single mode $\mathbf{A}^{(n)}$ have been estimated by using BSS, the remainder factor matrices can be obtained efficiently and essentially uniquely, no matter whether they have some highly collinear columns (bottlenecks) or even proportional columns. The new method consists of two major steps: in step 1, BSS is performed on a single mode and in step 2, the Khatri-Rao product structure is recovered. We call this method CP-SMBSS as running BSS on a single mode plays a key role (See Algorithm 1 for the pseudo-code). The success of CP-SMBSS relies on proper *a priori* knowledge of one mode. If such knowledge is

not available or we have used incorrect priori knowledge, (6) and consequently (7) may not hold. In this case the method fails. Once such *a priori* knowledge is available, however, CP-SMBSS can be employed to achieve faster convergence speed and overcome bottleneck problems. Moreover, the results given by CP-SMBSS are often more interpretable as the specific physical meaning of data has been incorporated.

A somewhat similar tensorial probabilistic ICA (pICA) method was proposed in [12] for 3-way tensors based on independence assumptions on one factor, where pICA and Khatri-Rao product structure recovering are performed iteratively. However, this kind of procedure not only increases computational cost but also often causes divergence problems. In this letter, we proved that these iterations can be avoided. We also clarified under which conditions the method succeeds. Finally, in the proposed method we can incorporate any suitable BSS techniques without limitation on the order of tensors.

Note that in the BSS stage of (6), as $\mathbf{Y}_{(n)} \in \mathbb{R}^{I_n \times \prod_{k\neq n} I_k}$ is often huge and $J \ll \prod_{k\neq n} I_k$, so we often apply dimensionality reduction methods on $\mathbf{Y}_{(n)}$ such as truncated SVD first to significantly reduce the computational complexity and filter out noise. In the second stage each column is computed directly and thus tensor based ALS iterations are completely avoided. Consequently, CP-SMBSS can enjoy very high efficiency in practice, especially for highly collinear data.

## IV. Fast Rank-1 Approximation with Optional Nonnegativity Constraints

Generally, the optimal rank-1 approximation of a matrix can be achieved by truncated SVD. Here, we briefly introduce another alternative method which is not only faster but also capable of accommodating nonnegative constraints naturally. To solve optimization problem

$$\min_{\mathbf{u},\mathbf{v}} \|\widehat{\mathbf{H}}_j^{(p)} - \mathbf{u}\mathbf{v}^T\|_F^2, \ p \neq n, \tag{15}$$

where $\widehat{\mathbf{H}}_j^{(p)} \in \mathbb{R}^{I_p \times I_r}$, $\mathbf{u} = \widehat{\mathbf{a}}_j^{(p)}$, and $\mathbf{v} = \widehat{\mathbf{h}}_j^{(p-1)}$. We fix $\mathbf{u}$ first and the optimal $\mathbf{v}$ is obtained from

$$\mathbf{v} = \frac{1}{\mathbf{u}^T\mathbf{u}} \widehat{\mathbf{H}}_j^{(p)T}\mathbf{u}. \tag{16}$$

Then $\mathbf{v}$ is fixed and the optimal $\mathbf{u}$ is computed as

$$\mathbf{u} = \frac{1}{\mathbf{v}^T\mathbf{v}} \widehat{\mathbf{H}}_j^{(p)}\mathbf{v}. \tag{17}$$

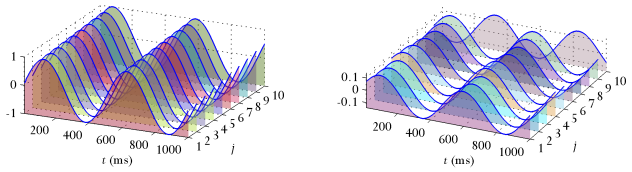
We run (16) and (17) iteratively till convergence, the optimal $\mathbf{u}$ and $\mathbf{v}$ emerge. It can be verified the computational complexity of them is $\mathcal{O}(MN)$. In the case where nonnegative components are desired, we simply let

$$\mathbf{u} = \max(\mathbf{u}, \epsilon), \ \mathbf{v} = \max(\mathbf{v}, \epsilon), \tag{18}$$

where $\epsilon \geq 0$ is very small. See [1], [13] for more details and convergence analysis. By employing this approach the optimal nonnegative rank-1 approximation can be obtained.

**Algorithm 1** The CP-SMBSS Algorithm

**Require:** $\underline{\mathbf{Y}}$, $J$, $n$, and a BSS algorithm $\Psi$.
1: Let $\widehat{\mathbf{A}}^{(n)} = \Psi(\mathbf{Y}_{(n)})$ and $\widehat{\mathbf{B}}^{(n)} = \mathbf{Y}_{(n)}^T[\widehat{\mathbf{A}}^{(n)}]^\dagger$.
2: **for** $j = 1$ to $J$ **do**
3: Let $\widehat{\mathbf{h}}_j^{(N)}$ be the $j$th column of $\widehat{\mathbf{B}}^{(n)}$. $I_r = \Pi_{k\neq\{n,N\}} I_k$.
4: **for** $p = N$ to 1 with $p \neq n$ **do**
5: Construct $\widehat{\mathbf{H}}_j^{(p)} \in \mathbb{R}^{I_p \times I_r}$ by reshaping $\widehat{\mathbf{h}}_j^{(p)}$;
6: Obtain $\widehat{\mathbf{a}}_j^{(p)}$, $\widehat{\mathbf{h}}_j^{(p-1)}$ from the rank-1 approximation of $\widehat{\mathbf{H}}_j^{(p)}$ (by using, e.g., (16)-(18)).
7: $I_r \leftarrow I_r / I_p$.
8: **end for**
9: **end for**
10: **return** $\widehat{\mathbf{A}}^{(p)}, p = 1, 2, \cdots, N$.




(a) Highly colinear source signals in mode-1.

(b) Estimated signals in mode-1. SIR=56.7dB.

Fig. 1: Performance of CP-SMBSS in the decomposition of a 3-way tensor with double bottlenecks. Although the sources are highly correlated, CP-SMBSS estimated all signals very well.

TABLE I: Performance of four CP algorithms in the decomposition of a 3-way tensor with double bottlenecks.

| Algorithm | $\text{mSIR}_1$ | $\text{mSIR}_2$ | $\text{mSIR}_3$ | Fit | Runtime (s) |
|---|---|---|---|---|---|
| SWATLD | 37.5967 | 11.0672 | 3.7856 | 0.8564 | 77.3 |
| ALS-ELS | 32.3156 | 9.9660 | 5.3962 | 0.9813 | 515.3 |
| CP-ALS | 30.8252 | 10.0586 | 2.2586 | 0.9804 | 85.3 |
| CP-SMBSS | **56.6947** | **48.6985** | **45.8784** | **0.9990** | **0.7** |

## V. 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

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

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

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

where $\underline{\widehat{\mathbf{Y}}}$ is an estimate of $\underline{\mathbf{Y}}$, $\text{Fit}(\underline{\mathbf{Y}}, \underline{\widehat{\mathbf{Y}}}) = 1$ if and only if $\underline{\widehat{\mathbf{Y}}} = \underline{\mathbf{Y}}$.

**Simulation 1:** We constructed a tensor with double bottlenecks as follows: The columns of $\mathbf{A}^{(3)}$ were chosen as independent sine waves with different frequencies included in

the benchmark acsin10d of ICALAB [14]. The components in $\mathbf{A}^{(1)} \in \mathbb{R}^{1000\times 10}$ consisted of 10 sine waves with slightly shifted phases but the same frequency $f = 2$Hz. That is, $\mathbf{a}_j^{(1)}(t) = \sin(2\pi ft + j\frac{\pi}{50})$, $j = 1, 2, \cdots, 10$. The sampling time was from 0 to 990 ms with the interval of 10 ms (see Fig.1(a)). We used the technique adopted in [9] to generate colinear components in $\mathbf{A}^{(2)}$, i.e., $\mathbf{a}_j^{(2)} = \mathbf{v}_j$ for $j = 1$; and $\mathbf{a}_j^{(2)} = \mathbf{a}_{j-1}^{(2)} + 0.5\mathbf{v}_j$ for $j = 2, 3, \cdots, 10$, where the entries of $\mathbf{v}_j$ were drawn from independent standard normal distributions. By these configurations, the every neighboring signals in $\mathbf{A}^{(1)}$ and $\mathbf{A}^{(2)}$ have correlations higher than 0.9. The ALS method with enhanced linear search (ALS-ELS) [9], SWATLD [15] and the CP-ALS [16] were compared with the proposed method, and their maximum iteration number was set to 500. The performance of the compared algorithms is detailed in Table I, where $\text{mSIR}_i$ denotes the mean value of SIR in mode-$i$. It can be seen that, due to the bottlenecks, SWATLD, CP-ALS and ALS-ELS converged very slowly and did not estimate some of the true sources. On the contrary, CP-SMBSS in which we applied SOBI [14] in mode-3 estimated all the components correctly using much less runtime, see Fig.1 (b) for the recovered components (the waveforms of mode 2 and 3 are omitted due to space limitation).

**Simulation 2:** We applied our algorithm to the COIL-20 [17] objects clustering. The COIL-20 database consists of 1440 images of 20 objects, 72 images per object. We used the first $K$ objects ($K = 5, 10, 15, 20$) to form the observation tensor $\underline{\mathbf{Y}}$ with the size of $128 \times 128 \times 72K$. We decomposed the tensor $\underline{\mathbf{Y}}$ with $J = K$ first and then used the factor $\mathbf{A}^{(3)}$ as the features for clustering. We used the $K$-means and replicated it 20 times in each run to reduce the influence of initial guess of centers. In the CP-SMBSS we used the lraNMF method [13] to extract $\mathbf{A}^{(1)}$ (used 5000 randomly sampled fibers, see [13]) and applied the HALS iterations (16)-(18) to extract the other *nonnegative* components. We compared our method with the Nonnegative CP-ALS (NCP-ALS) method [16], which is a quite typical implementation of nonnegative CP decompositions, and the NTF-HALS method [1], where the maximum number of iterations was set 1000. The unconstrained CP-ALS [16] was also compared. TABLE II shows their clustering accuracy and runtime (for $K = 20$) averaged over 50 runs. We can see that the CP-SMBSS provided the the highest clustering accuracy for all $K$ while the unconstrained CP-ALS obtained the lowest. Note that CP-SMBSS had slightly higher fitting error, which may suggest some physical properties of factors play more critical role than minimum fitting error in practical applications. Moreover, the CP-SMBSS consumed 11.7 seconds , which is the fastest one among all the compared nonnegative CP decomposition methods.

TABLE II: Comparison of performance in the clustering of the COIL-20 objects over 50 runs. The CP-SMBSS is the fastest one among all nonnegative CP decomposition methods.

| Algorithm | Clustering accuracy (%) | | | | Runtime (s) |
|---|---|---|---|---|---|
| | $K = 5$ | $K = 10$ | $K = 15$ | $K = 20$ | ($K = 20$) |
| NCP-ALS | 89.7 | 84.3 | 74.0 | 63.7 | 1230.8 |
| NTF-HALS | 90.1 | 85.5 | 75.4 | 64.9 | 46.4 |
| CP-SMBSS | **95.8** | **85.7** | **76.6** | **71.9** | **11.7** |
| CP-ALS | 85.0 | 80.9 | 67.4 | 57.1 | 2.4 |

## VI. Conclusion

In this letter we proposed a new CP decomposition method, assuming that some *a priori* knowledge about the components at least in one mode is available. We run a standard BSS algorithms on this mode and estimate the components in this mode first. The remainder component matrices are then computed directly via a series of rank-1 singular value decompositions. This method is extremely efficient and overcomes the bottleneck issue of CP decomposition. Particularly, for the case where the ordinary CP decompositions are not essentially unique, our method provides more interpretable results with physical meaning. Simulation results justified 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] 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.

[3] 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.

[4] V. De Silva and L.-H. Lim, "Tensor rank and the ill-posedness of the best low-rank approximation problem," *SIAM Journal on Matrix Analysis and Applications*, vol. 30, no. 3, pp. 1084–1127, 2008.

[5] A. Stegeman and L. D. Lathauwer, "A method to avoid diverging components in the CANDECOMP/PARAFAC model for generic $i \times J \times 2$ arrays," *SIAM Journal on Matrix Analysis and Applications*, vol. 30, no. 4, pp. 1614–1638, 2009.

[6] J. B. Kruskal, "Three-way arrays: rank and uniqueness of trilinear decompositions, with application to arithmetic complexity and statistics," *Linear Algebra and its Applications*, vol. 18, no. 2, pp. 95 – 138, 1977.

[7] N. D. Sidiropoulos and R. Bro, "On the uniqueness of multilinear decomposition of n-way arrays," *Journal of Chemometrics*, vol. 14, no. 3, pp. 229 – 239, 2000.

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

[9] P. Comon, X. Luciani, and A. L. F. de Almeida, "Tensor decompositions, alternating least squares and other tales," *Journal of Chemometrics*, vol. 23, no. 7-8, pp. 393–405, 2009.

[10] S. Kindermann and C. Navasca, "Analysis and approximation of the Canonical Polyadic tensor decomposition," *ArXiv e-prints*, Sep. 2011.

[11] A. Cichocki and S. Amari, *Adaptive Blind Signal and Image Processing: Learning Algorithms and Applications*. John Wiley & Sons, Ltd, 2002.

[12] C. F. Beckmann and S. M. Smith, "Tensorial extensions of independent component analysis for multisubject fMRI analysis," *NeuroImage*, vol. 25, no. 1, pp. 294 – 311, 2005.

[13] 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, 2012.

[14] A. Cichocki, S. Amari, K. Siwek, and T. Tanaka, "ICALAB toolbox," 2007. [Online]. Available: http://www.bsp.brain.riken.jp/ICALAB.

[15] 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.

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

[17] S. A. Nene, S. K. Nayar, and H. Murase, "Columbia object image library (COIL-20)," 1996. [Online]. Available: http://www.cs.columbia.edu/CAVE/software/softlib/coil-20.php