Canonical Polyadic Decomposition Based on a Single Mode Blind Source Separation

Guoxu Zhou and Andrzei 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 problem, 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

Reciently, 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., Y, denote tensors, and Y(n) denotes the mode-n unfolding (matricization) of Y. Bold capitals (e.g., A) and bold lowercase letters (e.g., a) denote matrices and vectors, respectively. ⊙ denotes the Khatri-Rao product (columnwise Kronecker product) of matrices and ⊙k̸=n A(k) = A(N) ◦ ··· ◦ A(n+1) ◦ A(n−1) ◦···◦ 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 \( Y \in \mathbb{R}^{I_1 \times I_2 \cdots \times I_N} \) can be formally formulated as

\[
Y = \sum_{j=1}^{J} \lambda_j a_j^{(1)} \odot a_j^{(2)} \cdots \odot a_j^{(N)} + \mathcal{E},
\]

(1)

where matrices \( A^{(n)} = [a_1^{(n)}, a_2^{(n)}, \ldots, a_J^{(n)}] \in \mathbb{R}^{I_n \times J} \), \( n = 1, 2, \ldots, N \), consists of unknown components \( a_j^{(n)} \), e.g. latent source signals, \( \odot \) denotes the outer product, and \( \mathcal{E} \) 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 \( A^{(N)} \) by letting \( a_j^{(N)} = \lambda_j a_j^{(N)} \), \( \forall j \), we also use \( Y \approx [A^{(1)}, A^{(2)}, \ldots, A^{(N)}] \) 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 \( Y \):

\[
Y_{(n)} = A^{(n)} B^{(n)\top}, n = 1, 2, \ldots, N,
\]

(2)

where

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

(3)

In standard ALS methods, factor matrices \( A^{(n)} \) are updated as \( Y_{(n)} [B^{(n)\top}]^\dagger \) sequentially for \( n = 1, 2, \ldots, N \), 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.

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

\[ X = SM^T, \]  

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

\[ \hat{S} = \Psi(X) = SPD, \]  

where \( P \) and \( D \) 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 \( Y(n) \) are just linear mixtures of the columns of \( A(n) \). This motivates us to extract \( A(n) \) first by using a proper BSS algorithm \( \Psi \) such that

\[ \hat{A}(n) = \Psi(Y(n)) = \Psi(A(n)B(n)^T) = A(n)PD. \]  

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

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

**Proof:** Let \( \hat{A}(n) \) be an estimate of \( A(n) \) such that \( \hat{A}(n) = A(n)PD \). Substitute \( \hat{A}(n) \) into (2) and we have

\[ B(n) = \hat{B}(n)DP^T, \]  

where \( \hat{B}(n) = Y(n)^T[\hat{A}(n)^T]^T \in \mathbb{R}(|I_k| \times I_k) \times J \) is fixed as it only depends on the observation data \( Y(n) \) and the estimated matrix \( \hat{A}(n) \). From (3) and (7), we have

\[ B(n)PD^{-1} = [\circ_{k \neq n} A(k)]PD^{-1} = \hat{B}(n). \]  

We define matrices \( \hat{A}(k), (k \neq n) \), such that

\[ \hat{A}(k) = A(k)P, \quad \forall k \neq n, N; \quad \hat{A}(N) = A(N)PD^{-1}. \]  

From (9) and (10), we have

\[ [\circ_{k \neq n} \hat{A}(k)] = [\circ_{k \neq n} A(k)]PD^{-1} = \hat{B}(n). \]  

In the following we show that such \( \hat{A}(k) \) are actually unique, which naturally leads to the essential uniqueness of \( A(k) \), due to the relationship (10). From (11), we have

\[ [\circ_{k \neq n} \hat{a}(j)] = \hat{h}_j(n), \]  

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

\[ \hat{H}_N(n) = \hat{a}(n) \hat{h}_j(N) \hat{h}_j(N)^{-1} = \mathbb{R}^{I \times I_r}, \]  

where \( \hat{h}_j(N^{-1}) = [\circ_{k \neq n(N,N)} \hat{a}(j)] \) and \( I_r = \prod_{k \neq (n,N)} I_k \). From (13) we see that \( \hat{H}_N(n) \) is a rank-1 matrix, and \( \hat{a}(j) \) and \( \hat{h}_j(N^{-1}) \) are just its left and right singular vectors corresponding to the unique leading singular value, respectively. Consequently, the \( j \)-th column of \( \hat{A}(N) \), i.e., \( \hat{a}(j) \), can be uniquely estimated first from the leading left singular vector of the rank-1 matrix \( \hat{H}_N(n) \). Then reshape \( \hat{h}_j(N^{-1}) \) such that

\[ \hat{H}_j(N^{-1}) = \hat{a}(N^{-1}) \hat{h}_j(N^{-1}) \hat{h}_j(N^{-1})^{-1}. \]  

where \( \hat{h}_j(N^{-2}) = [\circ_{k \neq (n,N,N-1)} \hat{a}(j)] \). Again, \( \hat{H}_j(N^{-1}) \) is rank-1 and the \( j \)-th column of \( \hat{A}(N^{-1}) \), i.e., \( \hat{a}(N^{-1}) \) is its leading left singular vector. Generally, by running truncated SVD on the rank-1 matrix formed by the reshaped right singular vector \( \hat{h}_j(N^{-1}) \), we can estimate all \( \hat{a}(p) \) 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 \( \hat{A}(p) \), i.e., \( \hat{a}(p) \), \( p \neq n \), are obtained.

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

In summary, after the components in a single mode \( A(n) \) 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 \( Y_{(n)} \in \mathbb{R}^{I_n \times \Pi_{k \neq n} I_k} \) is often huge and \( J \ll \prod_{k \neq n} I_k \), so we often apply dimensionality reduction methods on \( Y_{(n)} \) 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

\[
\min_{u,v} \|H_j^{(p)} - uv^T\|_F^2, \quad p \neq n,
\]

where \( H_j^{(p)} \in \mathbb{R}^{I_p \times I_T} \), \( u = \tilde{a}_j^{(p)} \), and \( v = \tilde{h}_j^{(p-1)} \). We fix \( u \) and the optimal \( v \) is obtained from

\[
v = \frac{1}{u^T \tilde{H}_j^{(p)} u} \tilde{H}_j^{(p)T} u.
\]

Then \( v \) is fixed and the optimal \( u \) is computed as

\[
u = \frac{1}{v^T \tilde{H}_j^{(p)} v} \tilde{H}_j^{(p)} v.
\]

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

\[
u = \max(u, \epsilon), \quad v = \max(v, \epsilon),
\]

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:** \( Y, J, n, \) and a BSS algorithm \( \Psi \).

1. Let \( \hat{A}^{(n)} = \Psi(Y_{(n)}) \) and \( B^{(n)} = Y_{(n)}^* [\hat{A}^{(n)}]^t \).
2. for \( j = 1 \) to \( J \) do
3. Let \( \hat{H}_j^{(N)} \) be the \( j \)th column of \( B^{(n)} \). \( I_r = \Pi_{k \neq \{n,N\}} I_k \).
4. for \( p = N \) to 1 with \( p \neq n \) do
5. Construct \( \hat{H}_j^{(p)} \in \mathbb{R}^{I_p \times I_T} \) by reshaping \( \hat{H}_j^{(N)} \).
6. Obtain \( \tilde{a}_j^{(p)} \) and \( \tilde{h}_j^{(p-1)} \) from the rank-1 approximation of \( \hat{H}_j^{(p)} \) (by using, e.g., (16)-(18)).
7. \( I_r \leftarrow I_r / I_p \).
8. end for
9. end for
10. return \( \hat{A}^{(p)}, p = 1, 2, \ldots, N \).

---

**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

\[
\text{SIR}(a, \hat{a}) = 10 \log_{10} \frac{\sum \alpha_k^2}{\sum (\alpha_k - \hat{\alpha}_k)^2},
\]

where \( a, \hat{a} \) are normalized random variables with zero mean and unit variance, and \( \hat{a} \) is an estimate of \( a \). 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

\[
\text{Fit}(\hat{Y}, \hat{Y}) = 1 - \frac{||Y - \hat{Y}||_F}{||Y||_F},
\]

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

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

---

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

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>mSIR1</th>
<th>mSIR2</th>
<th>mSIR3</th>
<th>Fit</th>
<th>Runtime (s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>SWATLD</td>
<td>37.5967</td>
<td>11.0672</td>
<td>3.7856</td>
<td>0.8564</td>
<td>77.3</td>
</tr>
<tr>
<td>ALS-ELS</td>
<td>32.3156</td>
<td>9.9660</td>
<td>5.3962</td>
<td>0.9813</td>
<td>515.3</td>
</tr>
<tr>
<td>CP-ALS</td>
<td>30.8252</td>
<td>10.0586</td>
<td>2.2586</td>
<td>0.9804</td>
<td>85.3</td>
</tr>
<tr>
<td>CP-SMBSS</td>
<td>56.6947</td>
<td>48.6985</td>
<td>45.8784</td>
<td>0.9990</td>
<td>0.7</td>
</tr>
</tbody>
</table>

---
the benchmark acsin10d of ICALAB [14]. The components in $A^{(1)} \in \mathbb{R}^{1000 \times 10}$ consisted of 10 sine waves with slightly shifted phases but the same frequency $f = 2\text{Hz}$. That is, $a^{(1)}_j(t) = \sin(2\pi ft + j\frac{2\pi}{10})$, $j = 1, 2, \ldots, 10$. 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 $A^{(2)}$, i.e., $a^{(2)}_j = v_j$ for $j = 1$; and $a^{(2)}_j = a^{(2)}_{j-1} + 0.5v_j$ for $j = 2, 3, \ldots, 10$, where the entries of $v_j$ were drawn from independent standard normal distributions. By these configurations, the every neighboring components in $A^{(1)}$ and $A^{(2)}$ 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 mSIR, 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 $Y$ with the size of $128 \times 128 \times 72K$. We decomposed the tensor $Y$ with $J = K$ first and then used the factor $A^{(3)}$ 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 $A^{(1)}$ (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.

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 confirmed the validity of the proposed method.

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.

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Clustering accuracy (%)</th>
<th>Runtime (s)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$K = 5$</td>
<td>$K = 10$</td>
</tr>
<tr>
<td>NCP-ALS</td>
<td>89.7</td>
<td>84.3</td>
</tr>
<tr>
<td>NTF-HALS</td>
<td>90.1</td>
<td>85.5</td>
</tr>
<tr>
<td>CP-SMBSS</td>
<td>95.8</td>
<td>85.7</td>
</tr>
<tr>
<td>CP-ALS</td>
<td>85.0</td>
<td>80.9</td>
</tr>
</tbody>
</table>

REFERENCES