High-order CPD estimator with dimensionality reduction using a tensor train model
Yassine Zniyed, Remy Boyer, André de Almeida, Gérard Favier

To cite this version:
Yassine Zniyed, Remy Boyer, André de Almeida, Gérard Favier. High-order CPD estimator with dimensionality reduction using a tensor train model. 26th European Signal Processing Conference (EUSIPCO 2018), Sep 2018, Rome, Italy. 10.23919/eusipco.2018.8553466 . hal-01815214

HAL Id: hal-01815214
https://hal-centralesupelec.archives-ouvertes.fr/hal-01815214
Submitted on 13 Jun 2018

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L’archive ouverte pluridisciplinaire HAL, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d’enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.
HIGH-ORDER CPD ESTIMATION WITH DIMENSIONALITY REDUCTION USING A TENSOR TRAIN MODEL

Yassine Zniyed* Rémy Boyer* André L. F. de Almeida† Gérard Favier‡

*L2S, Univ. Paris-Sud, CNRS, CentraleSupélec, Gif-sur-Yvette, France
†Department of Teleinformatics Engineering, Federal University of Fortaleza, Brazil
‡Laboratoire I3S, Université Côte d’Azur, CNRS, Sophia Antipolis, France

ABSTRACT
The canonical polyadic decomposition (CPD) is one of the most popular tensor-based analysis tools due to its usefulness in numerous fields of application. The Q-order CPD is parametrized by Q matrices also called factors which have to be recovered. The factors estimation is usually carried out by means of the alternating least squares (ALS) algorithm. In the context of multi-modal big data analysis, i.e., large order (Q) and dimensions, the ALS algorithm has two main drawbacks. Firstly, its convergence is generally slow and may fail, in particular for large values of Q, and secondly it is highly time consuming. In this paper, it is proved that a Q-order CPD of rank-R is equivalent to a train of Q 3-order CPD(s) of rank-R. In other words, each tensor train (TT)-core admits a 3-order CPD of rank-R. Based on the structure of the TT-cores, a new dimensionality reduction and factor retrieval scheme is derived. The proposed method has a better robustness to noise with a smaller computational cost than the ALS algorithm.

Index Terms— Tensor decompositions, CP decomposition, Tensor train, Big data, Multidimensionnal signal processing, Parameter estimation, Fast algorithms.

I. INTRODUCTION
In this paper, we focus on high-order CPD models [5]. Such models have a great interest in signal processing for blind equalization [7], [4], blind source separation [16], radar [10], wireless communications [12], [3], among many other fields of application. Most methods of factor estimation are ALS-based techniques [2]. Unfortunately these techniques may require several iterations to converge [11], [6], and convergence is increasingly difficult when the order of the tensor increases [1] and it is not even guaranteed [8]. Hence, algorithms that are stable and scalable [15] with the tensor order and dimensions are needed. In this work, we establish an equivalence between a Q-order CPD of canonical rank R and a tensor train (TT) [9] model. Note that the idea of rewriting a CPD into the TT format was briefly mentioned in [9]. In this paper, we exploit this idea to propose a new CPD factor retrieval algorithm. Indeed, we will see that a Q-order CPD of canonical rank R is equivalent to a train of Q 3-order CPD(s) with TT-ranks equal to R. That leads to a new tensor decomposition that we call a CPD-train model, and which can be viewed as a special case of the TT model whose the TT-cores follow a CPD. The advantages of using the TT model in this case is that its storage cost, just like the CPD, scales linearly with Q [9]. The estimation of the TT-ranks amounts to estimating the canonical rank of the associated CPD. And finally, the use of the TT-SVD algorithm [9] allows to overcome the drawbacks of iterative algorithms for very high-order tensors. As a consequence of this equivalence, one could apply one trilinear ALS estimator and Q – 3 bilinear ALS estimators to the (much smaller) TT-cores to recover the original CPD factor matrices, instead of applying the ALS algorithm to the very high-order CPD. Hence, a solution with much less computational cost and better convergence is possible thanks to this TT-CPD equivalence.

The rest of this paper is organized as follows. Section II recalls the CPD and TT formulations, and gives the main result in terms of the equivalence between the two models. Section III-B shows how to exploit the known structure of the CPD TT-cores, by applying the ALS algorithm on 3-order tensors instead of the original Q-order CPD. In section IV, we compare the computational complexities of the TT-SVD and ALS algorithms for estimating the parameters of a Q-order CPD. Section V presents simulation results that will reinforce the arguments given in the previous sections. Finally, the conclusions are drawn in Section VI.

Notations: Scalars, vectors, matrices and tensors are represented by x, x, X and X, respectively. The symbols (·)T, rank(·), and κ(·) denote, respectively, the transpose, the rank and the dominant complexity cost in flops. The Frobenius norm is defined by || · ||F. Iκ,R denotes the k-order identity tensor of size R × · · · × R, and I2 = I1.

The matrix unfoldkX of size Nk×N1 · · · ×Nκ+1 · · · NQ refers to the k-mode unfolding of X of size N1 × · · · × NQ. The n-mode product is denoted by ×n. The contraction ×q between two tensors A and B of size N1 × · · · × NQ and M1 × · · · × Mp, with Nq = Mp, is a tensor of order...
Proof. Let \( \mathbf{A} \times_q \mathbf{B} \) be any given CPD of order \( Q \) and rank \( P \). Then, the canonical rank of \( (Q + P - 2) \) such that

\[
[\mathbf{A} \times_q \mathbf{B}]_{n_1,\ldots,n_{Q+1},\ldots,n_Q,m_1,\ldots,m_{Q+1},\ldots,m_P} = \sum_{k=1}^{N_Q} [\mathbf{A}]_{n_1,\ldots,n_{Q-k},k,n_{Q-k+1},\ldots,n_Q} \mathbf{B}_{m_1,\ldots,m_{Q-k},k,m_{Q-k+1},\ldots,m_P}.
\]

II. CPD-TRAIN MODEL: EQUIVALENCE BETWEEN THE CPD AND THE TENSOR-TRAIN MODEL

II-A. Canonical Polyadic Decomposition (CPD)

A \( Q \)-order tensor of size \( N_1 \times \ldots \times N_Q \) follows a rank- \( R \) CPD if

\[
\mathbf{X} = \mathcal{I}_{Q,R} \times_1 \mathbf{P}_1 \times_2 \mathbf{P}_2 \cdots \times_Q \mathbf{P}_Q
\]

where the \( k \)-mode factor matrix \( \mathbf{P}_k \) is of size \( N_k \times R \), \( 1 \leq k \leq Q \).

II-B. Tensor-Train Model (TTM) and TT-ranks

Let \( \{R_1, \ldots, R_{Q-1}\} \) be the \( Q-1 \) TT-ranks with bounding conditions \( R_0 = R_Q = 1 \). A \( Q \)-order tensor of size \( N_1 \times \ldots \times N_Q \) admits a decomposition into a train of tensors if

\[
\mathbf{X} = \mathbf{G}_1 \times_1 \mathbf{G}_2 \times_2 \mathbf{G}_3 \times_3 \cdots \times_{Q-1} \mathbf{G}_{Q-1} \times_Q \mathbf{G}_Q
\]

in which the TT-cores are defined according to

\[
\mathbf{G}_1 : N_1 \times R_1, \quad \mathbf{G}_Q^T : N_Q \times R_{Q-1},
\]

\[
\mathbf{G}_k : R_{k-1} \times N_k \times R_k, \quad 2 \leq k \leq Q - 1
\]

with the TT-ranks given by rank(\( \mathbf{G}_1 \)) = \( R_1 \), rank(\( \mathbf{G}_Q \)) = \( R_{Q-1} \), and for \( 2 \leq k \leq Q - 1 \), rank(unfold\(_k\)\( \mathbf{G}_k \)) = \( R_{k-1} \), and rank(unfold\(_4\)\( \mathbf{G}_k \)) = \( R_k \).

II-C. Rewriting of a high order CPD in a TT format

In this section, we present an equivalence between a \( Q \)-order CPD and a train of \( Q \) CPD(s) according to Fig. 1, where the structure of the TT-cores are given in the following Theorem.

**Theorem 1.** Assume that a tensor \( \mathbf{X} \) follows a \( Q \)-order CPD of rank- \( R \) given by eq. (1), where the factor matrices \( \mathbf{P}_k \) are full column rank. This tensor admits an equivalent TTM with TT-cores given by:

\[
\mathbf{G}_1 = \mathbf{P}_1, \quad \mathbf{G}_Q^T = \mathbf{P}_Q^T, \quad \mathbf{G}_k = \mathcal{I}_{Q,R} \times_2 \mathbf{P}_k, \quad 2 \leq k \leq Q - 1
\]

Moreover, the TT-ranks are all identical and equal to the canonical rank \( R \).

**Proof.** The TTM of the \( Q \)-order identity tensor \( \mathcal{I}_{Q,R} \) is given by

\[
\mathcal{I}_{Q,R} = \mathbf{I}_R \times_1 \mathcal{I}_{Q-1,R} \times_2 \mathcal{I}_{Q,R} \times_3 \cdots \times_{Q-1} \mathcal{I}_{Q,R} \times_Q \mathbf{I}_R.
\]

Substituting the above TTM into the CPD of eq. (1), we get

\[
\mathbf{X} = (\mathbf{I}_R \times_1 \mathcal{I}_{Q-1,R} \times_2 \mathcal{I}_{Q,R} \times_3 \cdots \times_{Q-1} \mathbf{I}_R) \times_1 \mathbf{P}_1 \cdots \times_Q \mathbf{P}_Q
\]

by expressing the entries of \( \mathbf{X} \) and reorganizing them, we can equivalently write

\[
\mathbf{X} = (\mathbf{P}_1 \mathbf{I}_R) \times_1 (\mathcal{I}_{Q-1,R} \times_2 \mathbf{P}_2) \times_3 \cdots \times_{Q-1} (\mathbf{I}_R \mathbf{P}_Q^T).
\]

By identifying the TT-cores in (4) with those in eq. (2), we can deduce the relations (3). Then, it is straightforward to conclude that the TT-ranks are all identical and equal to the canonical rank \( R \). \( \square \)

As a \( Q \)-order CPD of rank- \( R \) is equivalent to a train of \( Q \)-order CPD(s) of rank- \( R \), we refer to Eq. (4) as a CPD-train.

![Fig. 1. Factor graph of a Q-order CPD and its TTM. The TT-cores are given in Theorem 1.](image)

III. DIMENSIONALITY REDUCTION AND CPD FACTORS ESTIMATION BASED ON A CPD-TRAIN

III-A. Dimensionality reduction: TT-cores recovery

The TT-SVD algorithm presented in [9] computes the TT-cores by extracting in a sequential way the dominant subspaces (thanks to truncated SVDs) of matrix reshapings of one SVD factor calculated at each step of the algorithm. The TT-cores recovery is done up to nonsingular change-of-basis matrices.

In Fig. 2, we illustrate the TT-SVD algorithm applied to a 4-order tensor \( \mathbf{X} \). At each step, the original tensor is reshaped into a matrix having one mode in one dimension and a combination of remaining modes in the other dimension, so that each SVD calculation delivers a TT-core. In our example of Fig. 2, the original \( N_1 \times N_2 \times N_3 \times N_4 \) tensor \( \mathbf{X} \) is first reshaped into an \( N_1 \times (N_2N_3N_4) \) matrix \( \mathbf{X}(1) \) whose the SVD provides the factors \( \mathbf{U}^{(1)} \) that contains the left singular vectors, and \( \mathbf{V}^{(1)} \) that is the product of the diagonal singular values matrix with the matrix containing the right singular vectors. Then, the \( R_1 \times (N_2N_3N_4) \) matrix \( \mathbf{V}^{(1)} \) is reshaped into a matrix...
The factors \( (U^{(2)}, V^{(2)}) \), \( V^{(2)} \) being reshaped into \( V^{(2)}_Q \), and so on. The TT-cores \( G_k \) are sequentially generated from the matrices \( U^{(k)} \).

**Theorem 2.** Let us consider the CPD-train defined in (4). Applying the TT-SVD algorithm to recover the TT-cores, we obtain the following equations:

\[
\begin{align*}
G_1 &= P_1 M_1^{-1}, \\
G_k &= \mathcal{I}_{3,R} \times_1 M_{k-1} \times_2 P_k \times_3 M_k^{-T}, \quad 2 \leq k \leq Q - 1 \\
G_Q &= M_{Q-1} P_Q^T
\end{align*}
\]

where \( M_k \in \mathbb{C}^{R \times R} \) is a nonsingular change-of-basis matrix.

**Proof.** The proof of this theorem is constructive, i.e., it is based on the algebraic structure of the TT-SVD algorithm applied to a Q-order CPD tensor of a known canonical rank \( R \). The proof is omitted due to a lack of space. \( \square \)

This means that if a Q-order tensor admits a rank-R CPD with full column rank factors, then its TTM involves a train of Q 3-order CPD(s) having all identical TT-ranks such as \( R_1 = \ldots = R_{Q-1} = R \). The factors can be derived straightforwardly from the TT-cores up to two change-of-basis matrices.

**Remark.** Note that each TT-core for \( 2 \leq k \leq Q - 1 \), follows a CPD linked to its previous TT-core (see Fig. 3). Indeed, the knowledge of the matrix \( M_{k-1} \) estimated from the TT-core \( G_{k-1} \) can be exploited for computing the CPD of \( G_k \).

The recursion property is exploited in the sequel to propose a new factor retrieval scheme. Note that the involved structure differs from the one of coupled tensors [13].

**III-B. CPD factors retrieval based on a CPD-train**

A strategy for CPD factors retrieval based on a CPD-train is derived in this section. Once the dimensionality reduction step is done, the CPD factors can be derived using the model equivalence presented in Theorem 2. More precisely, the results of Theorem 2 allow us to derive the factors retrieval scheme presented in Algorithm 1. In this Algorithm, we denote by Tri-ALS the ALS algorithm applied to a 3-order tensor, while Bi-ALS denotes the ALS algorithm applied to a 3-order tensor using the knowledge of one factor. The use of the Bi-ALS algorithm is justified relatively to the Remark III-A.

**Remark III-C. Permutation and scaling ambiguities**

The factors of the 3-order CPDs are estimated up to trivial ambiguities, i.e. up to column permutation and scaling [14].
Theorem 3. The proposed factors retrieval scheme estimates the factors up to
1) a unique column permutation matrix denoted by Π,
2) diagonal scaling matrices satisfying the following relation:
\[ \Lambda_1 \Lambda_2 \Lambda_3 \cdots \Lambda_{Q-1} \Lambda_Q = I_R \]
where \( \Lambda_k \) is the scaling ambiguity for the \( k \)-mode factor \( P_k \).

Proof. Firstly, it is straightforward to check that if a change-of-basis matrix \( M_k^T \) is estimated up to a column permutation matrix \( \Pi \) and a diagonal scaling matrix \( \Gamma_k \), then \( M_k \) is known up to the same column permutation matrix and a diagonal scaling matrix \( \Gamma_k^{-1} \). So, we can deduce that the column permutation matrix is unique and common to all the TT-cores due to the recursion property. This proves the first item of the theorem. Secondly, we have
\[ G_1 = P_1 \Pi \Lambda_1 (M_1 \Pi \Gamma_1^{-1})^{-1} \]
\[ G_2 = \mathcal{I}_{3,R} \times_1 M_1 \Pi \Gamma_1^{-1} \]
\[ \times_2 P_2 \Pi \Lambda_2 \times_3 M_2^{-T} \Pi \Gamma_2 \]
\[ G_3 = \mathcal{I}_{3,R} \times_1 M_2 \Pi \Gamma_2^{-1} \]
\[ \times_2 P_3 \Pi \Lambda_3 \times_3 M_3^{-T} \Pi \Gamma_3 \]
\[ \vdots \]
\[ G_{Q-1} = \mathcal{I}_{3,R} \times_1 M_{Q-2} \Pi \Gamma_{Q-2}^{-1} \]
\[ \times_2 P_{Q-1} \Pi \Lambda_{Q-1} \times_3 M_{Q-1}^{-T} \Pi \Gamma_{Q-1} \]
\[ G_Q = M_{Q-1} \Pi \Gamma_{Q-1}^{-1} (P_Q \Pi \Lambda_Q)^T. \]

Based on the above expressions and using the property that the product of scaling matrices is identity [5], we have \( \Gamma_{Q}^{-1} \Lambda_2 \Gamma_2 = I_R \) from eq. (6), and \( \Gamma_2 = A_2 \Gamma_3 \) from eq. (7). Based on the two above properties, we obtain \( \Gamma_{Q}^{-1} \Lambda_2 \Gamma_3 = I_R \). Continuing the same reasoning until the \( k \)-th and \( k+1 \)-th TT-cores, we obtain \( \Gamma_{Q}^{-1} \Lambda_2 \cdots \Lambda_k \Gamma_k = I_R \), and \( \Gamma_k^{-1} \Lambda_{k+1} \cdots \Lambda_{Q-1} = I_{Q-1} \). As \( \Gamma_{Q}^{-1} = \Lambda_1 \), and \( \Gamma_{Q-1} = \Lambda_Q \), it is easy to deduce the relation (5). □

IV. COMPUTATIONAL COMPLEXITY

In this section, we compare the computational complexity of the proposed CPD-train algorithm with that of the ALS algorithm applied to a very large CPD tensor. Hypercubic Q-order tensors of rank \( R = 3 \) and dimensions \( N \times \cdots \times N \) following a CPD were generated with random factor matrices whose elements were drawn from a Gaussian distribution with zero mean and unit variance. The rank \( R \) is assumed to be perfectly known. Let \( f(\mathbf{X}^{(t)}) = \|\mathbf{X} - \mathbf{X}^{(t)}\|_F \), where \( \mathbf{X}^{(t)} \) denotes the estimated tensor at the \( t \)-th iteration. The convergence test was chosen such that \( \frac{f(\mathbf{X}^{(t)}) - f(\mathbf{X}^{(t-1)})}{f(\mathbf{X}^{(t)})} < 2 \cdot 10^{-16} \) or when the number of iterations exceeds 1000. The computation time given for the simulations is the average time of 20 realizations. In Table I, we fix \( N = 6 \) and vary the tensor order. It is clear that the gain increases with the order.

Table I. Comparison of the execution time for \( R = 3, N = 6 \).

<table>
<thead>
<tr>
<th>Tensor order</th>
<th>Proposed scheme</th>
<th>ALS</th>
<th>Gain</th>
</tr>
</thead>
<tbody>
<tr>
<td>( Q = 6 )</td>
<td>0.63 (s)</td>
<td>7.48 (s)</td>
<td>11.87</td>
</tr>
<tr>
<td>( Q = 7 )</td>
<td>1.07 (s)</td>
<td>62.57 (s)</td>
<td>58.47</td>
</tr>
<tr>
<td>( Q = 8 )</td>
<td>1.31 (s)</td>
<td>431.95 (s)</td>
<td>329.73</td>
</tr>
</tbody>
</table>

The conclusion is the same, i.e., the gain in the computation time increases when the dimensions are increased. On the
other hand, one can see that the convergence of the ALS is increasingly slow, when the tensor order and dimensions increase. In Fig. 4, the MSE is depicted for the CPD-train algorithm, the ALS algorithm, and the ALS applied after a preprocessing step for noise reduction. The MSEs depicted were obtained by averaging the results over 300 independent Monte Carlo runs, truncated from 5% worst and 5% best MSEs (because the ALS algorithm may not converge in some cases). Note that, due to the use of SVDs, the CPD-

\[
\begin{array}{cccc}
\text{Tensor dimension} & \text{Proposed scheme} & \text{ALS} & \text{Gain} \\
N = 4 & 0.79 (s) & 19.26 (s) & 24.37 \\
N = 5 & 0.91 (s) & 114.16 (s) & 125.45 \\
N = 6 & 1.31 (s) & 431.95 (s) & 329.73 \\
\end{array}
\]

### Table II. Comparison of the computation time for \( R = 3, Q = 8 \).