This is a self-archived version of an original article. This version may differ from the original in pagination and typographic details.

Author(s): Wang, Deqing; Cong, Fengyu; Ristaniemi, Tapani

Title: Higher-order Nonnegative CANDECOMP/PARAFAC Tensor Decomposition Using Proximal Algorithm

Year: 2019

Version: Accepted version (Final draft)

Copyright: © 2019 IEEE.

Rights: In Copyright

Rights url: http://rightsstatements.org/page/InC/1.0/?language=en

Please cite the original version:

ABSTRACT

Tensor decomposition is a powerful tool for analyzing multiway data. Nowadays, with the fast development of multisensor technology, more and more data appear in higher-order (order \( \geq 4 \)) and nonnegative form. However, the decomposition of higher-order nonnegative tensor suffers from poor convergence and low speed. In this study, we propose a new nonnegative CANDECOMP/PARAFAC (NCP) model using proximal algorithm. The block principal pivoting method in alternating nonnegative least squares (ANLS) framework is employed to minimize the objective function. Our method can guarantee the convergence and accelerate the computation. The results of experiments on both synthetic and real data demonstrate the efficiency and superiority of our method.

Index Terms— Tensor decomposition, nonnegative CANDECOMP/PARAFAC, proximal algorithm, block principal pivoting, alternating nonnegative least squares

1. INTRODUCTION

In recent years, the widespread application of multisensor technology and the fast development of advanced signal processing methods have promoted the formation of multiway data as higher-order tensor. For example, in a brain signal experiment, the event-related potential (ERP) can be represented even by a seventh-order tensor including modes such as space, frequency, time, trial, subject, condition and group [1]. Tensor decomposition, especially nonnegative CANDECOMP/PARAFAC (NCP) decomposition, is a favourable tool to analyze these data [2]. In order to process such higher-order data efficiently, fast and stable tensor decomposition algorithm is necessary.

Block coordinate descent (BCD) method [3, 4] is a general and important framework to solve tensor decomposition, in which each factor matrix is updated alternatively as a subproblem. Many conventional methods are proposed in BCD framework. For example, hierarchical alternating least squares (HALS) was designed for large scale tense data [5, 6], which showed fast computation. However, the normalization of factor matrices in HALS will spoil the bound-constrained property of NCP and complicate the optimization procedures [7]. Alternating nonnegative least squares (ANLS) is a powerful sub-framwork in BCD for NCP, benefiting from the efficiency of many nonnegative least squares (NNLS) methods such as active set (AS) [8] and block principal pivoting (BPP) [9]. Nevertheless, ANLS often suffers from rank deficiency because of the sparse effect introduced by the nonnegative constraints and the possible appearance of zero components in factor matrices. In recent year, alternating proximal gradient (APG) [3, 10, 11] method has gained in popularity for NMF and third-order tensor decomposition because of its stable convergence, but it still converges very slowly for higher-order tensor (order \( \geq 4 \)). The challenge of higher-order tensor decomposition is to design a solving algorithm that is convergent and efficient.

Recently, proximal algorithm has been applied to unconstrained CP decomposition [12, 13]. The advantage is that the combination of BCD framework and proximal algorithm will satisfy the need for uniqueness of minimum in each subproblem [14]. Therefore, the tensor decomposition will be guaranteed to converge to stationary point [14]. We extend proximal algorithm to the bound-constrained NCP, which had not been adequately analyzed in previous studies. We also find that NCP using proximal algorithm is equivalent to a ANLS problem. Consequently, BPP, as an efficient NNLS method, is employed to solve the ANLS problem. We conduct experiments on both fourth-order synthetic and real data to demonstrated the efficiency and superiority of our method.

2. NCP DECOMPOSITION

In this paper, we denote a vector by boldface lowercase letter, such as \( \mathbf{x} \); a matrix by boldface uppercase letter, such as \( \mathbf{X} \);
and a tensor by boldface Euler script letter, such as $\mathbf{X}$. Operator $\odot$ represents outer product of vectors, $\ast$ represents the Hadamard product, $\langle \rangle$ represents inner product, $\| \|$ represents Frobenius norm.

Given a nonnegative $N$th-order tensor $\mathbf{X} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$, the nonnegative CANDECOMP/PARAFAC (NCP) decomposition is to solve the following minimization problem:

$$
\min_{A^{(1)},\ldots,A^{(N)}} \frac{1}{2} \left\| \mathbf{X} - \left[ A^{(1)}, \ldots, A^{(N)} \right] \right\|_F^2
$$

subject to $A^{(n)} \succeq 0$ for $n = 1, \ldots, N$,

where $A^{(n)} \in \mathbb{R}^{I_n \times R}$ for $n = 1, \ldots, N$ are the estimated factor matrices in different modes, $I_n$ is the size in mode-$n$, and $R$ is the predefined number of components.

Block coordinate descent [3, 4] is an important method to solve NCP problem, in which the factor matrices of $A^{(n)}$, $n = 1, \ldots, N$, are updated alternatively. Let $X^{(n)} \in \mathbb{R}^{I_n \times \prod_{n=1,\neq n}^{N} I_n}$ represent the mode-$n$ unfolding (matri- zification) of original tensor $\mathbf{X}$. And the mode-$n$ unfolding of $[A^{(1)}, \ldots, A^{(N)}]$ can be written as $A^{(n)} \left( B^{(n)} \right)^T$, in which $B^{(n)} = \left( A^{(N)} \odot \cdots \odot A^{(n+1)} \odot A^{(n-1)} \odot \cdots \odot A^{(1)} \right) \in \mathbb{R}^{I_n \times (\prod_{n=1,\neq n}^{N} I_n) \times R}$. The updating of $A^{(n)}$ in the $k$th iteration is solved as the following subproblem:

$$
A^{(n)}_{k+1} = \arg\min_{A^{(n)} \succeq 0} \frac{1}{2} \left\| X^{(n)} - A^{(n)} \left( B^{(n)}_k \right)^T \right\|_F^2.
$$

Essentially, (2) is a bound-constrained optimization problem, for which HALS [5, 6], APG [3, 10, 11] and ANLS [7–9] are popular optimization methods. The nonnegative constraint will naturally lead to sparse results, which might introduce zero components to $A^{(n)}$. Thus, $A^{(n)}$ might not be full column rank. Although many nonnegative least squares (NNLS) methods in ANLS framework usually run very fast, such as active set (AS) [8] and block principal pivoting (BPP) [9], they often suffer from the rank deficiency. In order to prevent the rank deficiency, the Tikhonov regularization (squared Frobenius norm) [15] is always incorporated into NCP as the following subproblem:

$$
A^{(n)}_{k+1} = \arg\min_{A^{(n)} \succeq 0} \left\{ \frac{1}{2} \left\| X^{(n)} - A^{(n)} \left( B^{(n)}_k \right)^T \right\|_F^2 + \frac{\alpha_n}{2} \left\| A^{(n)} \right\|_F^2 \right\},
$$

where $\alpha_n$ is positive regularization parameter in parameter vector $\alpha \in \mathbb{R}^{N \times 1}$. The objective function in (3) can be equivalently rewritten as

$$
\mathcal{F}_1 = \frac{1}{2} \left\| \left( X^{T} \right)_{I_n \times R} - \left( B^{(n)}_k \sqrt{\alpha_n} I_R \right) \left( A^{(n)} \right)^T \right\|_F^2,
$$

where $I$ is the identity matrix and $0$ is zero matrix. Afterwards, NNLS methods, such as AS and BPP, can be employed to minimize the subproblem. Nevertheless, the optimal solution by (3) is not a stationary point of NCP in (1) [13].

APG exhibits efficient convergence properties for third-order tensor, in which the proximal operator is employed to update the factor matrices yielding a close form solution [3]. However, APG still shows slow convergence for higher-order (order $\geq 4$) tensor data.

### 3. NCP USING PROXIMAL ALGORITHM

Proximal algorithm has been successfully utilized in unconstrained CP decomposition, which can guarantee that CP converges to stationary point [12, 13]. Inspired by this idea, we extend the proximal algorithm to the bound-constrained NCP problem. The NCP using proximal algorithm is

$$
\min_{A^{(1)},\ldots,A^{(N)}} \left\{ \frac{1}{2} \left\| \mathbf{X} - \left[ A^{(1)}, \ldots, A^{(N)} \right] \right\|_F^2 + \sum_{n=1}^{N} \frac{\alpha_n}{2} \left\| \tilde{A}^{(n)} - A^{(n)} \right\|_F^2 \right\}
$$

subject to $A^{(n)} \succeq 0$ for $n = 1, \ldots, N$,

where $\tilde{A}^{(n)} \in \mathbb{R}^{I_n \times R}$ is the former version of $A^{(n)}$ in previous iteration. According to block coordinate descent method, $A^{(n)}$ in the $k$th iteration can be updated alternatively by the following subproblem:

$$
A^{(n)}_{k+1} = \arg\min_{A^{(n)} \succeq 0} \left\{ \frac{1}{2} \left\| X^{(n)} - A^{(n)} \left( B^{(n)}_k \right)^T \right\|_F^2 + \frac{\alpha_n}{2} \left\| A^{(n)} \right\|_F^2 \right\}.
$$

The objective function in (5) can be equivalently rewritten as

$$
\mathcal{F}_2 = \frac{1}{2} \left\| \left( X^{T} \right)_{I_n \times R} - \left( B^{(n)}_k \sqrt{\alpha_n} I_R \right) \left( A^{(n)} \right)^T \right\|_F^2.
$$

Obviously, (5) is still a nonnegative least squares (NNLS) problem. Therefore, we employ the block principal pivoting (BPP) method [9] to solve the subproblem in (5).

Furthermore, we calculate the partial derivative of $\mathcal{F}_2$

$$
\frac{\partial \mathcal{F}_2}{\partial A^{(n)}} = A^{(n)} \left( B^{(n)}_k \right)^T \frac{B^{(n)}_k + \alpha_n I_R}{\sqrt{\alpha_n} I_R} - X^{(n)} B^{(n)}_k - \alpha_n A^{(n)}
$$

where $X^{(n)} B^{(n)}_k$ is called the Matricized Tensor Times Khatri-Rao Product (MTTKRP) [16], and $\left( B^{(n)}_k \right)^T B^{(n)}_k$...
can be computed efficiently by

\[
\left( B_k^{(n)} \right)^T B_k^{(n)} = \left[ \left( A_k^{(N)} \right)^T A_k^{(N)} \right] * \cdots \times \left[ \left( A_k^{(n+1)} \right)^T A_k^{(n+1)} \right] \times \left[ \left( A_k^{(n-1)} \right)^T A_k^{(n-1)} \right] \times \cdots \times \left[ \left( A_k^{(1)} \right)^T A_k^{(1)} \right].
\]

The proposed NCP using proximal algorithm is summarized in Algorithm 1. Our method has several advantages. First, the combination of block coordinate descent and proximal algorithm can guarantee that the NCP converges to stationary point (see Section 3.7.1 in [14]). Second, BPP has proved to be a very efficient NNLS method [9], which will improve the performance of NCP significantly.

**Algorithm 1: NCP using proximal algorithm**

<table>
<thead>
<tr>
<th>Input</th>
<th>Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\mathcal{X}, R, \alpha$</td>
<td>$A^{(n)}$, $n=1, \ldots, N$</td>
</tr>
<tr>
<td>Initialize $A^{(n)} \in \mathbb{R}^{I_n \times R}$, $n=1, \ldots, N$, using random numbers;</td>
<td></td>
</tr>
<tr>
<td>repeat</td>
<td></td>
</tr>
<tr>
<td>for $n=1 \rightarrow N$ do</td>
<td></td>
</tr>
<tr>
<td>Make mode-$n$ unfolding of $\mathcal{X}$ as $X^{(n)}$;</td>
<td></td>
</tr>
<tr>
<td>Compute MTTKRP $X^{(n)} B_k^{(n)}$ and</td>
<td></td>
</tr>
<tr>
<td>$(B_k^{(n)})^T B_k^{(n)}$ based on (7);</td>
<td></td>
</tr>
<tr>
<td>$(B_k^{(n)})^T B_k^{(n)} \leftarrow (B_k^{(n)})^T B_k^{(n)} + \alpha_n I_R$;</td>
<td></td>
</tr>
<tr>
<td>$X^{(n)} B_k^{(n)} \leftarrow X^{(n)} B_k^{(n)} + \alpha_n A_k^{(n)}$;</td>
<td></td>
</tr>
<tr>
<td>Update factor $A^{(n)}$ based on (5) using BPP:</td>
<td></td>
</tr>
<tr>
<td>$A_{k+1}^{(n)} = \arg\min_{A^{(n)} \geq 0} \mathcal{J}<em>2 \left( A^{(n)} \right) \in \text{NNLS}</em>{-BPP} \left( X^{(n)} B_k^{(n)} , (B_k^{(n)})^T B_k^{(n)} \right)$</td>
<td></td>
</tr>
<tr>
<td>until some termination criterion is reached;</td>
<td></td>
</tr>
<tr>
<td>return $A^{(n)}$, $n=1, \ldots, N$.</td>
<td></td>
</tr>
</tbody>
</table>

4. EXPERIMENTS AND RESULTS

We applied the proposed NCP using proximal algorithm (PROX-BPP for short in the following contents) to both fourth-order synthetic and real tensor data. Comparison was made with conventional algorithms of HALS, APG and ANLS with Frobenius-norm regularization based on BPP (ANLS-BPP for short).

For all algorithms, the factor matrices were initialized using nonnegative normally distributed random numbers by command $\max \left( 0, \text{randn} \left( I_n, R \right) \right)$. The stopping condition was based on the change of relative error [2], in which the tolerance was set by $1e^{-8}$. The maximum running time was 600s. For PROX-BPP and ANLS-BPP, we kept $\alpha_n$, $n=1, \ldots, N$ the same in all modes with fixed value during the iteration. The values of 1, $1e^{-2}$, $1e^{-4}$, $1e^{-6}$, and $1e^{-8}$ were tested for $\alpha_n$. The objective function value, relative error, running time, iteration, and nonzero component number of the first factor matrix were used to measure the performance of the algorithms. The results of 30 independent runs were recorded and the average was computed.

All experiments were conducted on a computer with Intel Core i5-4590 3.30GHz CPU, 8GB memory, 64-bit Windows 10 and MATLAB R2016b. The fundamental tensor computation was based on Tensor Toolbox 2.6 [16–18].

4.1. Fourth-order Synthetic Data

We synthesized a fourth-order nonnegative tensor by 7 channels of simulated signals, which come from the AC-7,2noi file in NMFLAB [19] as shown in Fig. 1(a). The tensor was constructed by $X_{\text{Syn}} = [\mathbf{S}(1), \mathbf{A}(2), \mathbf{A}(3), \mathbf{A}(4)] \in \mathbb{R}^{100 \times 100 \times 100 \times 5}$, in which $\mathbf{S}(1) \in \mathbb{R}^{10 \times 100 \times 100 \times 5}$ is the signal matrix, and $\mathbf{A}(2), \mathbf{A}(3) \in \mathbb{R}^{100 \times 7}, \mathbf{A}(4) \in \mathbb{R}^{5 \times 7}$ are random matrices in uniform distribution. Next, nonnegative Gaussian noise was added to the tensor with SNR of 40dB.

For all algorithms on this synthetic data, the number of components is set by 7. The average results of 30 independent runs are recorded in Table 1. One of the estimated signal matrix by PROX-BPP with $\alpha_n = 1e^{-4}$ is shown in Fig. 1(b). We compare the objective function convergence of all algorithms within the first 180s with the same initialized factor matrices as shown in Fig. 2(a), in which we set $\alpha_n = 1e^{-4}$ for PROX-
4.2. Fourth-order ERP Data

We utilized a set of preprocessed fourth-order event-related potential (ERP) data (channel × frequency × time × subject-group = 9 × 71 × 60 × 42). The 9 channel points represent 9 electrodes on the scalp, the 71 frequency points represent 1-15Hz, the 60 time points represent 0-300ms, and the 42 subject-group points include 21 subjects with reading disability (RD) and 21 subjects with attention deficit (AD) [20].

For all algorithms, the number of components is set by 40. The experimental procedures are the same as that for the synthetic data. The results are shown in Table 2 and Fig. 4. One group of components extracted by PROX-BPP is shown in Fig. 3, which represents typical brain activity [20].

4.3. Discussion

From the results of both fourth-order synthetic data and real ERP data, we find that our proposed PROX-BPP method outperforms all other methods with high efficiency and accuracy. ANLS-BPP method has high objective function value and large relative error, and often yields fewer meaningful components than the predefined ones. Although HALS has satisfying accuracy, it is inferior to PROX-BPP in running time. APG, which has excellent performance for third-order tensor, shows very low convergence for higher-order (order ⩾ 4) tensor.

The choice of parameter $\alpha_n$ for PROX-BPP is said to be related the noise level in the data [12, 13]. Surprisingly, our PROX-BPP is very robust with different $\alpha_n$ values. We suggest to select $1e-2 \leq \alpha_n \leq 1e-4$, since too large value may affect the objective function and too small value might still cause rank deficiency.

5. CONCLUSION

In this study, we proposed a new NCP method using proximal algorithm in block coordinate descent framework. Afterwards, one of the efficient NNLS methods implemented by block principal pivoting (BPP) was employed to solve the model. The proposed method exhibited high efficiency and outperformed conventional methods on higher-order (order ⩾ 4) tensor data. Our method is very flexible, and can be combined with many other NNLS algorithms.
6. REFERENCES


