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)

Please cite the original version:

Wang, D., Cong, F., & Ristaniemi, T. (2019). Higher-order Nonnegative CANDECOMP/PARAFAC Tensor Decomposition Using Proximal Algorithm. In ICASSP 2019 : Proceedings of the 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (pp. 3457-3461). IEEE. Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing. https://doi.org/10.1109/ICASSP.2019.8683217

HIGHER-ORDER NONNEGATIVE CANDECOMP/PARAFAC TENSOR DECOMPOSITION USING PROXIMAL ALGORITHM

Deqing Wang*† *Fengyu Cong**† *Tapani Ristaniemi*†

*School of Biomedical Engineering, Faculty of Electronic Information and Electrical Engineering,
Dalian University of Technology, Dalian, China
†Faculty of Information Technology, University of Jyväskylä, Jyväskylä, Finland
deqing.wang@foxmail.com, cong@dlut.edu.cn, tapani.e.ristaniemi@jyu.fi

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 $\geqslant 4$) 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 CANDECOM/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 $\geqslant 4$). 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 $\boldsymbol{x}$; a matrix by boldface uppercase letter, such as $\boldsymbol{X}$;

This work was supported by the National Natural Science Foundation of China (Grant No. 81471742), the Fundamental Research Funds for the Central Universities [DUT16JJ(G)03] in Dalian University of Technology in China, and the scholarship from China Scholarship Council (No. 201600090043). (*Corresponding author: Fengyu Cong*)

and a tensor by boldface Euler script letter, such as $\mathcal{X}$. Operator $\circ$ represents outer product of vectors, $*$ represents the Hadamard product, $\langle\ \rangle$ represents inner product, $[\![\ ]\!]$ represents Kruskal operator, and $\|\ \|_F$ means Frobenius norm.

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

$$\begin{aligned} \min_{\boldsymbol{A}^{(1)},\ldots,\boldsymbol{A}^{(N)}} & \frac{1}{2}\left\|\mathcal{X} - [\![\boldsymbol{A}^{(1)},\ldots,\boldsymbol{A}^{(N)}]\!]\right\|_F^2 \\ \text{s.t. } & \boldsymbol{A}^{(n)} \geqslant 0 \text{ for } n = 1,\ldots,N, \end{aligned} \tag{1}$$

where $\boldsymbol{A}^{(n)} \in \mathbb{R}^{I_n \times R}$ 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 $\boldsymbol{A}^{(n)}$, $n = 1,\ldots,N$, are updated alternatively. Let $\boldsymbol{X}_{(n)} \in \mathbb{R}^{I_n \times \prod_{\tilde{n}=1,\tilde{n}\neq n}^{N} I_{\tilde{n}}}$ represent the mode-n unfolding (matricization) of original tensor $\mathcal{X}$. And the mode-n unfolding of $[\![\boldsymbol{A}^{(1)},\ldots,\boldsymbol{A}^{(N)}]\!]$ can be written as $\boldsymbol{A}^{(n)}\left(\boldsymbol{B}^{(n)}\right)^T$, in which $\boldsymbol{B}^{(n)} = \left(\boldsymbol{A}^{(N)} \odot \cdots \odot \boldsymbol{A}^{(n+1)} \odot \boldsymbol{A}^{(n-1)} \odot \cdots \odot \boldsymbol{A}^{(1)}\right) \in \mathbb{R}^{\prod_{\tilde{n}=1,\tilde{n}\neq n}^{N} I_{\tilde{n}} \times R}$. The updating of $\boldsymbol{A}^{(n)}$ in the kth iteration is solved as the following subproblem:

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

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 $\boldsymbol{A}^{(n)}$. Thus, $\boldsymbol{A}^{(n)}$ might not be full column rank. Although many nonnegative leaset 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 Tiknonov regularization (squared Frobenius norm) [15] is always incorporated into NCP as the following subproblem:

$$\begin{aligned} \boldsymbol{A}_{k+1}^{(n)} = \underset{\boldsymbol{A}^{(n)}\geqslant 0}{\arg\min} \Bigg\{ & \frac{1}{2}\left\|\boldsymbol{X}_{(n)} - \boldsymbol{A}^{(n)}\left(\boldsymbol{B}_k^{(n)}\right)^T\right\|_F^2 \\ & + \frac{\alpha_n}{2}\left\|\boldsymbol{A}^{(n)}\right\|_F^2 \Bigg\}, \end{aligned} \tag{3}$$

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

$$\mathcal{F}_1 = \frac{1}{2}\left\|\begin{pmatrix} \boldsymbol{X}_{(n)}^T \\ \boldsymbol{0}_{R\times I_n} \end{pmatrix} - \begin{pmatrix} \boldsymbol{B}_k^{(n)} \\ \sqrt{\alpha_n}\boldsymbol{I}_R \end{pmatrix}\left(\boldsymbol{A}^{(n)}\right)^T\right\|_F^2,$$

where $\boldsymbol{I}$ is the identity matrix and $\boldsymbol{0}$ 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 $\geqslant 4$) 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

$$\begin{aligned} \min_{\boldsymbol{A}^{(1)},\ldots,\boldsymbol{A}^{(N)}} & \Bigg\{\frac{1}{2}\left\|\mathcal{X} - [\![\boldsymbol{A}^{(1)},\ldots,\boldsymbol{A}^{(N)}]\!]\right\|_F^2 \\ & + \sum_{n=1}^{N}\frac{\alpha_n}{2}\left\|\widetilde{\boldsymbol{A}}^{(n)} - \boldsymbol{A}^{(n)}\right\|_F^2\Bigg\} \\ \text{s.t. } & \boldsymbol{A}^{(n)} \geqslant 0 \text{ for } n = 1,\ldots,N, \end{aligned} \tag{4}$$

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

$$\begin{aligned} \boldsymbol{A}_{k+1}^{(n)} = \underset{\boldsymbol{A}^{(n)}\geqslant 0}{\arg\min} \Bigg\{ & \frac{1}{2}\left\|\boldsymbol{X}_{(n)} - \boldsymbol{A}^{(n)}\left(\boldsymbol{B}_k^{(n)}\right)^T\right\|_F^2 \\ & + \frac{\alpha_n}{2}\left\|\boldsymbol{A}_k^{(n)} - \boldsymbol{A}^{(n)}\right\|_F^2 \Bigg\}. \end{aligned} \tag{5}$$

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

$$\mathcal{F}_2 = \frac{1}{2}\left\|\begin{pmatrix} \boldsymbol{X}_{(n)}^T \\ \sqrt{\alpha_n}\left(\boldsymbol{A}_k^{(n)}\right)^T \end{pmatrix} - \begin{pmatrix} \boldsymbol{B}_k^{(n)} \\ \sqrt{\alpha_n}\boldsymbol{I}_R \end{pmatrix}\left(\boldsymbol{A}^{(n)}\right)^T\right\|_F^2.$$

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 $\mathcal{F}_2$

$$\begin{aligned} \frac{\partial \mathcal{F}_2}{\partial \boldsymbol{A}^{(n)}} = & \boldsymbol{A}^{(n)}\left[\left(\boldsymbol{B}_k^{(n)}\right)^T\boldsymbol{B}_k^{(n)} + \alpha_n \boldsymbol{I}_R\right] \\ & - \left[\boldsymbol{X}_{(n)}\boldsymbol{B}_k^{(n)} + \alpha_n \boldsymbol{A}_k^{(n)}\right], \end{aligned} \tag{6}$$

where $\boldsymbol{X}_{(n)}\boldsymbol{B}_k^{(n)}$ is called the *Matricized Tensor Times Khatri-Rao Product* (MTTKRP) [16], and $\left(\boldsymbol{B}_k^{(n)}\right)^T\boldsymbol{B}_k^{(n)}$

Algorithm 1: NCP using proximal algorithm

Input : $\mathcal{X}$, R, $\boldsymbol{\alpha}$

Output: $\boldsymbol{A}^{(n)}$, $n = 1, \ldots, N$

Initialize $\boldsymbol{A}^{(n)} \in \mathbb{R}^{I_n \times R}$, $n = 1, \ldots, N$, using random numbers;

repeat

  for $n = 1$ **to** N **do**

    Make mode-n unfolding of $\mathcal{X}$ as $\boldsymbol{X}_{(n)}$;

    Compute MTTKRP $\boldsymbol{X}_{(n)} \boldsymbol{B}_k^{(n)}$ and $\left(\boldsymbol{B}_k^{(n)}\right)^T \boldsymbol{B}_k^{(n)}$ based on (7);

    $\left(\boldsymbol{B}_k^{(n)}\right)^T \boldsymbol{B}_k^{(n)} \leftarrow \left(\boldsymbol{B}_k^{(n)}\right)^T \boldsymbol{B}_k^{(n)} + \alpha_n \boldsymbol{I}_R$;

    $\boldsymbol{X}_{(n)} \boldsymbol{B}_k^{(n)} \leftarrow \boldsymbol{X}_{(n)} \boldsymbol{B}_k^{(n)} + \alpha_n \boldsymbol{A}_k^{(n)}$;

    Update factor $\boldsymbol{A}^{(n)}$ based on (5) using BPP:

$$\boldsymbol{A}_{k+1}^{(n)} = \underset{\boldsymbol{A}^{(n)} \geqslant 0}{\operatorname{argmin}} \mathcal{F}_2\left(\boldsymbol{A}^{(n)}\right) = \texttt{NNLS\_BPP}\left(\boldsymbol{X}_{(n)} \boldsymbol{B}_k^{(n)}, \left(\boldsymbol{B}_k^{(n)}\right)^T \boldsymbol{B}_k^{(n)}\right).$$

  end

until *some termination criterion is reached*;

return $\boldsymbol{A}^{(n)}$, $n = 1, \ldots, N$.

can be computed efficiently by

$$\left(\boldsymbol{B}_k^{(n)}\right)^T \boldsymbol{B}_k^{(n)} = \left[\left(\boldsymbol{A}_k^{(N)}\right)^T \boldsymbol{A}_k^{(N)}\right] * \cdots * \left[\left(\boldsymbol{A}_k^{(n+1)}\right)^T \boldsymbol{A}_k^{(n+1)}\right] * \left[\left(\boldsymbol{A}_{k+1}^{(n-1)}\right)^T \boldsymbol{A}_{k+1}^{(n-1)}\right] * \cdots * \left[\left(\boldsymbol{A}_{k+1}^{(1)}\right)^T \boldsymbol{A}_{k+1}^{(1)}\right]. \quad (7)$$

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.

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(0,randn(`I_n, R`))`. 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 α_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 α_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].

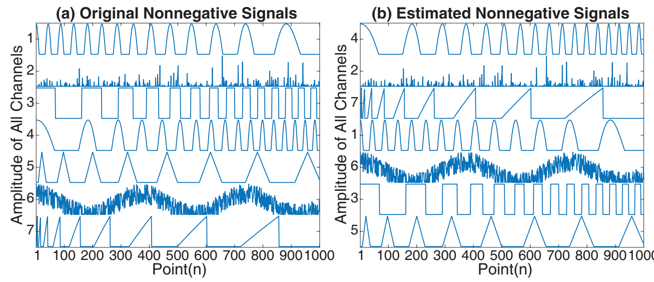

Fig. 1. Simulated signals. (a) shows original signals, and (b) shows the estimated signals by PROX-BPP with α_n =1e-4.

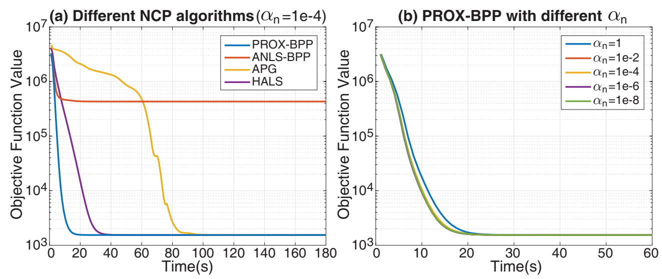

Fig. 2. Convergence of NCP algorithms on the synthetic data.

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 $\mathcal{X}_{\text{Syn}} = [\![\boldsymbol{S}^{(1)}, \boldsymbol{A}^{(2)}, \boldsymbol{A}^{(3)}, \boldsymbol{A}^{(4)}]\!] \in \mathbb{R}^{1000 \times 100 \times 100 \times 5}$, in which $\boldsymbol{S}^{(1)} \in \mathbb{R}^{1000 \times 7}$ is the signal matrix, and $\boldsymbol{A}^{(2)}, \boldsymbol{A}^{(3)} \in \mathbb{R}^{100 \times 7}$, $\boldsymbol{A}^{(4)} \in \mathbb{R}^{5 \times 7}$ 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 α_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 α_n =1e-4 for PROX-

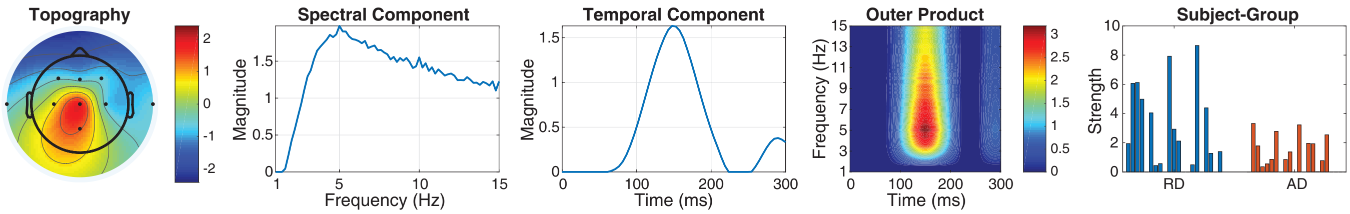

Fig. 3. One group of components extracted from ERP data by PROX-BPP with α_n =1e-4.

Table 1. Performances of NCPs On Synthetic Data

NCP	α_n	Obj	RelErr	Time	Iter	Comp
PROX BPP	1	1.5391e+03	0.0083	**47.4**	43.1	7.00
	1e-2	1.5391e+03	0.0083	**44.2**	43.0	7.00
	1e-4	1.5391e+03	0.0083	**43.7**	42.5	7.00
	1e-6	1.5391e+03	0.0083	**43.7**	42.6	7.00
	1e-8	1.5391e+03	0.0083	**44.0**	42.9	7.00
ANLS BPP	1	3.7249e+05	0.0852	264.4	251.0	6.23
	1e-2	2.9504e+05	0.0784	65.4	63.8	6.37
	1e-4	2.1125e+05	0.0633	48.8	47.6	6.53
	1e-6	3.7568e+05	0.0969	50.3	48.9	6.20
	1e-8	4.7112e+05	0.1103	52.7	51.1	6.00
APG	—	1.5392e+03	0.0083	158.1	149.6	7.00
HALS	—	1.5391e+03	0.0083	71.4	67.4	7.00

Table 2. Performances of NCPs On ERP Data

NCP	α_n	Obj	RelErr	Time	Iter	Comp
PROX BPP	1	4.7781e+05	**0.1116**	**24.7**	679.9	40.00
	1e-2	4.7803e+05	**0.1116**	**28.2**	775.8	40.00
	1e-4	4.7845e+05	**0.1117**	**29.2**	805.1	40.00
	1e-6	4.7796e+05	**0.1116**	**27.7**	764.4	40.00
	1e-8	4.7667e+05	**0.1115**	**29.7**	815.2	40.00
ANLS BPP	1	6.0792e+05	0.1250	28.6	611.1	33.33
	1e-2	5.8610e+05	0.1235	26.6	656.2	34.03
	1e-4	5.9085e+05	0.1240	24.9	676.9	33.80
	1e-6	5.9481e+05	0.1245	28.2	721.9	33.50
	1e-8	5.9230e+05	0.1241	30.3	604.8	33.80
APG	—	4.8100e+05	0.1120	99.4	2114.7	40.00
HALS	—	4.7860e+05	0.1117	51.9	1914.4	40.00

BPP and ANLS-BPP. The objective function convergence of PROX-BPP with different α_n is shown in **Fig. 2(b)**.

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].

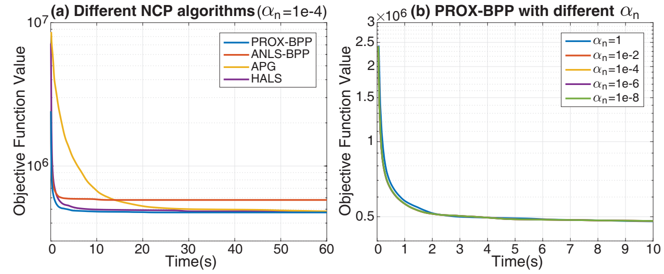

Fig. 4. Convergence of NCP algorithms on the real ERP data.

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 $\geqslant$ 4) tensor.

The choice of parameter α_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 α_n values. We suggest to select 1e-2 $\leqslant \alpha_n \leqslant$ 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. Afterwords, 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 $\geqslant$ 4) tensor data. Our method is very flexible, and can be combined with many other NNLS algorithms.

6. REFERENCES

[1] F. Cong, Q.-H. Lin, L.-D. Kuang, X.-F. Gong, P. Astikainen, and T. Ristaniemi, “Tensor decomposition of EEG signals: A brief review,” *Journal of Neuroscience Methods*, vol. 248, pp. 59–69, Jun. 2015.

[2] D. Wang, Y. Zhu, T. Ristaniemi, and F. Cong, “Extracting multi-mode ERP features using fifth-order nonnegative tensor decomposition,” *Journal of Neuroscience Methods*, vol. 308, pp. 240–247, Oct. 2018.

[3] Y. Xu and W. Yin, “A block coordinate descent method for regularized multiconvex optimization with applications to nonnegative tensor factorization and completion,” *SIAM Journal on Imaging Sciences*, vol. 6, no. 3, pp. 1758–1789, Jan. 2013.

[4] J. Kim, Y. He, and H. Park, “Algorithms for nonnegative matrix and tensor factorizations: a unified view based on block coordinate descent framework,” *Journal of Global Optimization*, vol. 58, no. 2, pp. 285–319, Mar. 2014.

[5] A. Cichocki, R. Zdunek, A. H. Phan, and S.-i. Amari, *Nonnegative matrix and tensor factorizations: applications to exploratory multi-way data analysis and blind source separation*. John Wiley & Sons, 2009.

[6] A. Cichocki and A.-H. Phan, “Fast local algorithms for large scale nonnegative matrix and tensor factorizations,” *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, vol. E92-A, no. 3, pp. 708–721, 2009.

[7] C.-J. Lin, “Projected gradient methods for nonnegative matrix factorization,” *Neural Computation*, vol. 19, no. 10, pp. 2756–2779, Oct. 2007.

[8] H. Kim and H. Park, “Nonnegative matrix factorization based on alternating nonnegativity constrained least squares and active set method,” *SIAM Journal on Matrix Analysis and Applications*, vol. 30, no. 2, pp. 713–730, Jan. 2008.

[9] J. Kim and H. Park, “Fast nonnegative matrix factorization: An active-set-like method and comparisons,” *SIAM Journal on Scientific Computing*, vol. 33, no. 6, pp. 3261–3281, Jan. 2011.

[10] Y. Xu, “Alternating proximal gradient method for sparse nonnegative tucker decomposition,” *Mathematical Programming Computation*, vol. 7, no. 1, pp. 39–70, May 2015.

[11] N. Guan, D. Tao, Z. Luo, and B. Yuan, “NeNMF: An optimal gradient method for nonnegative matrix factorization,” *IEEE Transactions on Signal Processing*, vol. 60, no. 6, pp. 2882–2898, Jun. 2012.

[12] C. Navasca, L. De Lathauwer, and S. Kindermann, “Swamp reducing technique for tensor decomposition,” in *2008 16th European Signal Processing Conference*. IEEE, 2008, pp. 1–5. [Online]. Available: https://www.eurasip.org/Proceedings/Eusipco/Eusipco2008/papers/1569105418.pdf

[13] N. Li, S. Kindermann, and C. Navasca, “Some convergence results on the regularized alternating least-squares method for tensor decomposition,” *Linear Algebra and its Applications*, vol. 438, no. 2, pp. 796–812, Jan. 2013.

[14] D. P. Bertsekas, *Nonlinear Programming, Third Edition*. Belmont, Massachusetts: Athena Scientific, 2016.

[15] L.-H. Lim and P. Comon, “Nonnegative approximations of nonnegative tensors,” *Journal of Chemometrics*, vol. 23, no. 7-8, pp. 432–441, Jul. 2009.

[16] B. W. Bader and T. G. Kolda, “Efficient MATLAB computations with sparse and factored tensors,” *SIAM Journal on Scientific Computing*, vol. 30, no. 1, pp. 205–231, Jan. 2008.

[17] B. W. Bader, T. G. Kolda *et al.*, “Matlab tensor toolbox version 2.6,” Available online, Feb. 2015. [Online]. Available: http://www.sandia.gov/~tgkolda/TensorToolbox/

[18] B. W. Bader and T. G. Kolda, “Algorithm 862: MATLAB tensor classes for fast algorithm prototyping,” *ACM Transactions on Mathematical Software*, vol. 32, no. 4, pp. 635–653, Dec. 2006.

[19] A. Cichocki and R. Zdunek, “Nmflab – matlab toolbox for non-negative matrix factorization,” 2006. [Online]. Available: http://www.bsp.brain.riken.jp/ICALAB/nmflab.html

[20] F. Cong, A. H. Phan, Q. Zhao, T. Huttunen-Scott, J. Kaartinen, T. Ristaniemi, H. Lyytinen, and A. Cichocki, “Benefits of multi-domain feature of mismatch negativity extracted by non-negative tensor factorization from EEG collected by low-density array,” *International Journal of Neural Systems*, vol. 22, no. 06, p. 1250025, Dec. 2012.