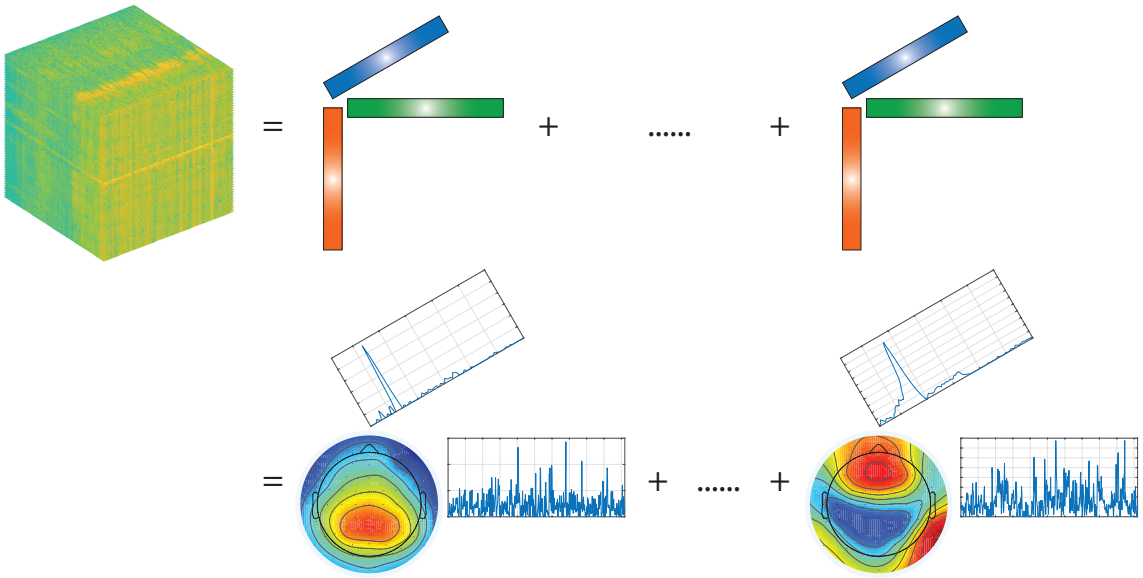
JYU DISSERTATIONS 169

Deqing Wang

# Extracting Meaningful EEG Features Using Constrained Tensor Decomposition



UNIVERSITY OF JYVÄSKYLÄ
FACULTY OF INFORMATION TECHNOLOGY

JYU DISSERTATIONS 169

Deqing Wang

# Extracting Meaningful EEG Features Using Constrained Tensor Decomposition

Esitetään Jyväskylän yliopiston informaatioteknologian tiedekunnan suostumuksella
julkisesti tarkastettavaksi yliopiston Agora-rakennuksen Alfa-salissa
joulukuun 12. päivänä 2019 kello 12.

Academic dissertation to be publicly discussed, by permission of
the Faculty of Information Technology of the University of Jyväskylä,
in building Agora, Alfa hall, on December 12, 2019 at 12 o'clock noon.

JYVÄSKYLÄN YLIOPISTO
UNIVERSITY OF JYVÄSKYLÄ

JYVÄSKYLÄ 2019

Editors
Timo Männikkö
Faculty of Information Technology, University of Jyväskylä
Ville Korkiakangas
Open Science Centre, University of Jyväskylä



Permanent link to this publication: http://urn.fi/URN:ISBN:978-951-39-7968-3

ISBN 978-951-39-7968-3 (PDF)
URN:ISBN:978-951-39-7968-3
ISSN 2489-9003

## ABSTRACT

Wang, Deqing
Extracting Meaningful EEG Features Using Constrained Tensor Decomposition
Jyväskylä: University of Jyväskylä, 2019, 61 p. (+included articles)
(JYU Dissertations
ISSN 2489-9003; 169)
ISBN 978-951-39-7968-3 (PDF)

Electroencephalography (EEG) is a powerful technique for the study of human brain and cognitive neuroscience. Nowadays, more and more EEG data are organized in high-dimension form, which is called tensor. Tensor decomposition is just the suitable tool to exploit the multiway data and extract EEG features that are linked to cognitive processes. Since the high-dimension EEG tensor often contains a large amount of data points, highly efficient tensor decomposition algorithm is desired. In addition, EEG tensor are sometimes nonnegative and the intrinsic features usually have some special properties, such as sparse. In order to extract meaningful feature components, it is necessary to incorporate constraint and regularization to tensor decomposition algorithm.

In this dissertation, we study the CANDECOMP/PARAFAC (CP) tensor decomposition with both nonnegative constraint and sparse regularization, which is abbreviated as sparse NCP. An inexact block coordinate descent (BCD) framework is employed for the non-convex sparse NCP problem. Five optimization methods are employed to solve the sparse NCP, including multiplicative update (MU), alternating nonnegative least squares/quadratic programming (ANLS/ANQP), hierarchical altering least squares (HALS), alternating proximal gradient (APG) and alternating direction method of multipliers (ADMM), all of which are carefully tailored to the sparse regularization problem. In order to improve the stability, we also utilize proximal algorithm particularly for ANLS/ANQP and HALS.

Applications on real-world EEG datasets are carried out. First, we use NCP to decompose a fifth-order event-related potential (ERP) tensor, which was collected by proprioceptive stimuli on human hands. Next, ongoing EEG tensors are analyzed using sparse NCP. The data were collected by naturalistic and continuous music stimulus. Finally, we analyze two modalities of ongoing EEG tensor and music signals simultaneously by N-way partial least square (N-PLS).

In conclusion, our designed tensor decomposition methods with constraint and regularization are able to decompose high-order tensor data efficiently and extract meaningful EEG features linked to cognitive processes.

Keywords: Tensor decomposition, nonnegative CANDECOMP/PARAFAC,
sparse regularization, block coordinate descent, EEG data analysis

# TIIVISTELMÄ (ABSTRACT IN FINNISH)

Wang, Deqing
Oleellisten piirteiden irrottaminen EEG-datasta rajoitetun tensorihajotelman avulla
Jyväskylä: University of Jyväskylä, 2019, 61 s. (+artikkelit)
(JYU Dissertations
ISSN 2489-9003; 169)
ISBN 978-951-39-7968-3 (PDF)

Elektroenkefalografia (EEG) on tehokas tapa tutkia ja mitata ihmisaivojen toimintaa ja edistää kognitiivista neurotiedettä. EEG-mittauksilla saatu data tallennetaan yhä yleisemmin moniuloitteisiin tietorakententeisiin eli tensoreihin. Hajottamalla tensori sen tekijöihin voidaan tutkia monikanavaista dataa ja irrottaa siitä ne EEG-tekijät, jotka liittyvät kognitiivisiin prosesseihin. Koska EEG-tensoreissa on usein paljon data-alkioita monissa ulottuvuuksissa, oleellisten data-alkioiden irrottamiseen tarvitaan tehokas algoritmi tensorin tekijöiden hajottamiseen. Lisäksi EEG-tensorit ovat joskus ei-negatiivisia ja erottamattomasti rakenteeltaan harvoja. Jotta tensorista voitaisiin irrottaa oleelliset piirteet, tensorin hajottamiseen soveltuvaan algoritmiin tulee liittää rajoitteita ja säännönmukaistamista.

Tässä väitöskirjassa tutkitaan CANDECOMP/PARAFAC (CP) hajoittamismenetelmää, joka sisältää sekä ei-negatiivisen rajoituksen että harvan säännönmukaistamisen (sparse NCP). Epäkonveksin harvan säännönmukaistamisen ongelman ratkaisuun käytetään epäeksaktia koordinaattiakselien suuntaisen optimoinnin viitekehystä (block coordinate descent). Harvaan säännönmukaistamiseen käytetään viittä seuraavaa optimointimenetelmää: MU (multiplicative update), ANLS/ANQP (altering nonnegative least squares/quadratic programming), HALS (hierarchical altering least squares), APG (alternating proximal gradient) ja ADMM (alternating direction method of multipliers), joista jokainen on tarkoin suunniteltu harvan säännönmukaistamisen ongelmaan. Stabiliteetin parantamiseksi hyödynnetään ANLS/ANQP- ja HALS-menetelmissä myös nk. lähialgoritmia.

Menetelmiä evaluoitiin todellisilla EEG-aineistoilla. Ensinnäkin NCP:tä käytettiin hajottamaan viidennen asteen herätevastetensori (event-related potential, ERP), joka oli koottu ihmiskäden asentoaistiärsykkeistä. Toisekseen harvaa NCP:tä käytettiin analysoimaan meneillään olevan EEG-mittauksen tuottamia tensoreita, jotka kuvasivat luonnollisia ja jatkuvia, musiikinkuuntelun aiheuttamia ärsykkeitä. Kolmanneksi analysoitiin kahden meneillään olevan EEG-mittauksen tuottamia tensoreita käyttämällä N-PLS-menetelmää.

Suunniteltujen menetelmien avulla voidaan tehokkaasti hajottaa monimutkaista tensoridataa sen tekijöihin ja irrottaa kognitiivisiin prosesseihin liittyviä EEG-tekijöitä.

Avainsanat: tensorin hajottaminen tekijöihin, ei-negatiivinen CANDECOMP/PARAFAC, harva säännönmukaistaminen, koordinaattiakselien suuntainen optimointi, EEG-data-analyysi

**Author**

Deqing Wang
Faculty of Information Technology
University of Jyväskylä
Finland
Email: deqing.wang@foxmail.com
Website: https://deqing.me/
ORCID: https://orcid.org/0000-0002-1333-0928

**Supervisors**

Professor Tapani Ristaniemi
Faculty of Information Technology
University of Jyväskylä
Finland

Professor Fengyu Cong
School of Biomedical Engineering
Dalian University of Technology
China
Faculty of Information Technology
University of Jyväskylä
Finland

**Reviewers**

Professor Guoxu Zhou
School of Automation
Guangdong University of Technology
China

Professor Karen Eguiazarian
Signal Processing Laboratory
Tampere University
Finland

**Opponent**

Professor Pauli Miettinen
School of Computing
University of Eastern Finland
Finland

## ACKNOWLEDGEMENTS

First, I would like to express my sincere gratitude to my supervisors Prof. Tapani Ristaniemi and Prof. Fengyu Cong for their guidance and support during the Ph.D. study. With my supervisors' support, I am very honoured to be selected as a member in the Ph.D. training program that is based on the international cooperation between Dalian University of Technology, China, and University of Jyväskylä, Finland. My supervisors led me to the interesting research fields of tensor decomposition and brain signal processing, to which I have devoted a long time with enthusiasm.

I wish to express my thanks to Prof. Petri Toiviainen for his help on music signal processing and music feature extraction. I wish to thank Dr. Piia Astikainen and Dr. Tiina Parviainen who organized many interesting workshops, summer schools and seminars on brain research. In these academic events, I learned many cutting-edge technologies and had the chances to communicate with other researchers.

I would like to thank my colleagues Xiulin Wang, Yongjie Zhu, Guanghui Zhang, Wenya Liu, Jia Liu, Rui Yan, Ye Ren, Lili Tian, Tiantian Yang and all other new members for their help during my research work. They are erudite and energetic, and I had learned a lot of knowledge and new ideas from the discussions with them. In addition, I am very happy with them during daily life. We had parties and went touring and cycling together. All of the happy times with them are memorable.

I also would like to thank my friend Toni Taipalus for his kind help on the translation of the Finnish text in my dissertation.

I am grateful for the scholarship from China Scholarship Council (CSC), China, and the funding from Faculty of Information Technology, University of Jyväskylä, Finland.

Finally, I am very thankful to my parents for their understanding and support for my Ph.D. study.

Jyväskylä, Finland
September 16, 2019
Deqing Wang

## LIST OF ACRONYMS

| | |
|---|---|
| **ADMM** | Alternating Direction Method of Multipliers |
| **ANLS** | Alternating Nonnegative Least Squares |
| **ANQP** | Alternating Nonnegative Quadratic Programming |
| **APG** | Alternating Proximal Gradient |
| **BCD** | Block Coordinate Descent |
| **BSS** | Blind Source Separation |
| **CP** | CANDECOMP/PARAFAC |
| **CWT** | Complex Wavelet Transform |
| **EEG** | Electroencephalography |
| **ERP** | Event-Related Potential |
| **HALS** | Hierarchical Altering Least Squares |
| **ICA** | Independent Component Analysis |
| **MU** | Multiplicative Update |
| **NCP** | Nonnegative CANDECOMP/PARAFAC |
| **NMF** | Nonnegative Matrix Factorization |
| **PCA** | Principal Component Analysis |
| **STFT** | Short Time Fourier Transform |
| **TFR** | Time-Frequency Representation |

## LIST OF FIGURES



## LIST OF TABLES

# CONTENTS

## LIST OF INCLUDED ARTICLES

PI Deqing Wang, Xiulin Wang, Tapani Ristaniemi and Fengyu Cong. Sparse Nonnegative Tensor Decomposition in Inexact Block Coordinate Descent Framework. *Submitted to a journal*.

PII Deqing Wang, Fengyu Cong and Tapani Ristaniemi. Higher-Order Nonnegative CANDECOMP/PARAFAC Tensor Decomposition Using Proximal Algorithm. *2019 IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP), Brighton, UK*, 2019.

PIII Deqing Wang, Yongjie Zhu, Tapani Ristaniemi and Fengyu Cong. Extracting multi-mode ERP features using fifth-order nonnegative tensor decomposition. *Journal of Neuroscience Methods, Volume 308, p.240-247*, 2018.

PIV Deqing Wang, Xiaoyu Wang, Yongjie Zhu, Petri Toiviainen, Minna Huotilainen, Tapani Ristaniemi and Fengyu Cong. Increasing Stability of EEG Components Extraction Using Sparsity Regularized Tensor Decomposition. *Advances in Neural Networks - the 15th International Symposium on Neural Networks (ISNN 2018), Minsk, Belarus*, 2018.

PV Deqing Wang, Fengyu Cong, Qibin Zhao, Petri Toiviainen, Asoke K. Nandi, Minna Huotilainen, Tapani Ristaniemi and Andrzej Cichocki. Exploiting ongoing EEG with multilinear partial least squares during free-listening to music. *2016 IEEE International Workshop on Machine Learning for Signal Processing (MLSP), Salerno, Italy*, 2016.

# 1 INTRODUCTION

Electroencephalography (EEG) is a powerful technique for noninvasively studying the electrophysiological dynamics of human brain (Cohen, 2017). EEG signals are the electrical activities generated by the firing of neurons in human brain, which can be measured and recorded via electrodes placed on the scalp. Since the recording equipment of EEG is cheaper and more flexible than that of fMRI and MEG, EEG is more common for brain research. EEG has several advantages for studyding cognitive neuroscience (Cohen, 2014):

- EEG is a high-temporal-resolution technique, which is well suited to capture fast, dynamic, and temporally sequenced cognitive events.
- The voltage fluctuations that are measured by EEG are direct reflections of biophysical phenomena at the level of populations of neurons.
- EEG signal is multidimensional, which provides many possibilities for specifying and testing hypothesis that are rooted in both neurophysiology and psychology.

EEG data consist mainly of three categories according to the external stimuli (Cong et al., 2015b): spontaneous EEG, which is measured without external stimuli, for example, in a resting state; event-related potential (ERP), which is recorded by controlled, short, rapidly repeated stimuli; ongoing EEG, which is collected by natural and continuous stimuli, such as dialog, music and movie. The analysis of EEG data is essentially to answer the following question (Cohen, 2017): what are the neural microcircuit functional/anatomical configurations that produce the various spatial/spectral/temporal EEG features that have been linked to cognitive processes? In order to answer this question, many methods had been proposed to extract meaningful EEG features in the past decades (Handy, 2005; Sanei and Chambers, 2007; Luck, 2014; Cong et al., 2015b). Moreover, nowadays, with the popularity of EEG and the increase of computer storage capacity, more EEG signals appear as multidimensional and big data. Therefore, it is necessary to develop advanced signal processing and data analysis methods to exact more meaningful EEG features that are related to cognitive processes.

## 1.1 EEG Signal Processing Methods

EEG data have rich information in many data domains, such as space, frequency, time, subject and experimental condition. In this section, we briefly introduce some popular methods for EEG signal processing related to these domains.

### 1.1.1 Conventional Methods

EEG signals can be represented by time series, spectrum, and spectrogram. Conventionally, EEG signal processing methods include time domain, frequency domain and time-frequency domain method.

#### 1.1.1.1 Time Domain

Averaging is a fundamental time domain method, especially for ERP data analysis. The recorded EEG data contain not only the brain's responses to the stimulus but also other activities that are unrelated to the stimulus, such as spontaneous EEG and noise. By averaging multiple single-trial EEG waveforms, an averaged ERP waveform that contains consistent response elicited by the stimulus will be created, meanwhile those signals unrelated to the stimulus will be suppressed (Luck, 2014). The averaging process is usually conducted separately for each electrode site and each stimulus type, yielding a separate average ERP waveform for each situation (Luck, 2014).

#### 1.1.1.2 Frequency Domain

Many brain activities exhibit constant variations and rhythmic dynamics (Cohen, 2019). The variant and dynamic temporal structure embedded in the time series of EEG can be extracted and quantified using Fourier transform or other spectral analysis method in frequency domain. However, Fourier transform requires the assumption of signal stationarity. In fact, most EEG data are non-stationary. Hence, simple frequency domain method has limitation for EEG data analysis.

#### 1.1.1.3 Time-Frequency Domain

Time-frequency representation (TFR) is a suitable method to analyze non-stationary EEG signals. The primary assumption of TFR is that the signal is roughly stationary over some short (sliding) time window (Cohen, 2019). By time-frequency transform, a time series of EEG data will be represented by a spectrogram, which contains both time and frequency dimension. Three most commonly used TFR methods are short time Fourier transform (SIFT), complex wavelet transform (CWT) and band-pass filtering with Hilbert transform (filter-Hilbert) (Cohen, 2019, 2014). The time-frequency analysis of EEG can be interpreted in terms of neurophysiological mechanisms of neural oscillations (Cohen, 2014). EEG oscillations are very important to analyze brain functions.

TABLE 1 EEG oscillations

| Name | Frequency Bands |
|---|---|
| delta band | $< 4$ Hz |
| theta band | $4 - 8$ Hz |
| alpha band | $8 - 13$ Hz |
| beta band | $13 - 30$ Hz |
| gamma band | $> 30$ Hz |

According to the frequency bands, EEG oscillations are mainly classified into five groups as listed in **Table 1**. More information on time-frequency analysis of EEG data can be found in Herrmann et al. (2014) and Cohen (2019, 2014).

### 1.1.2 Matrix Methods

As mentioned previously, EEG data are recorded via electrodes on the scalp. The recorded EEG series at a electrode site is the sum (mixture) of all source series multiplied by weighting factors (Kappenman and Luck, 2012). Let $\boldsymbol{S} = [\boldsymbol{s}_1, \boldsymbol{s}_2, \ldots, \boldsymbol{s}_N] \in \mathbb{R}^{T \times N}$ denote the source signal matrix, $\boldsymbol{X} = [\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_P] \in \mathbb{R}^{T \times P}$ denote the recorded scalp signal matrix and $\boldsymbol{A} \in \mathbb{R}^{P \times N}$ denote the mixing matrix, where $T, N, P$ represent the number of samples in series, number of sources and number of sensors respectively. The mixing process can be expressed by

$$\boldsymbol{X}^T = \boldsymbol{A}\boldsymbol{S}^T. \tag{1}$$

Here,

$$\boldsymbol{A} = \begin{bmatrix} a_{11} & a_{12} & \ldots & a_{1N} \\ a_{21} & a_{22} & \ldots & a_{2N} \\ \vdots & \vdots & \ddots & \vdots \\ a_{P1} & a_{P2} & \ldots & a_{PN} \end{bmatrix}$$

contains the weighting factors that map the signals from source space to electrode space. The mixing process and the relationship between the source signals and the recorded scalp signals are illustrated in **Figure 1**.

In fact, the mixing matrix $\boldsymbol{A}$ is always unknown. An important task is to recover the underlying source components by computing an unmixing matrix that reverses the mixing process. The unmixing process is called blind source separation (BSS) (Comon and Jutten, 2010). According to the properties of the desired unmixed components, there are several types of BSS methods.

#### 1.1.2.1 Independent Components Analysis

If the separated components are statistically independent to each other and have non-Guassian structure, this type of BSS is called independent compo-

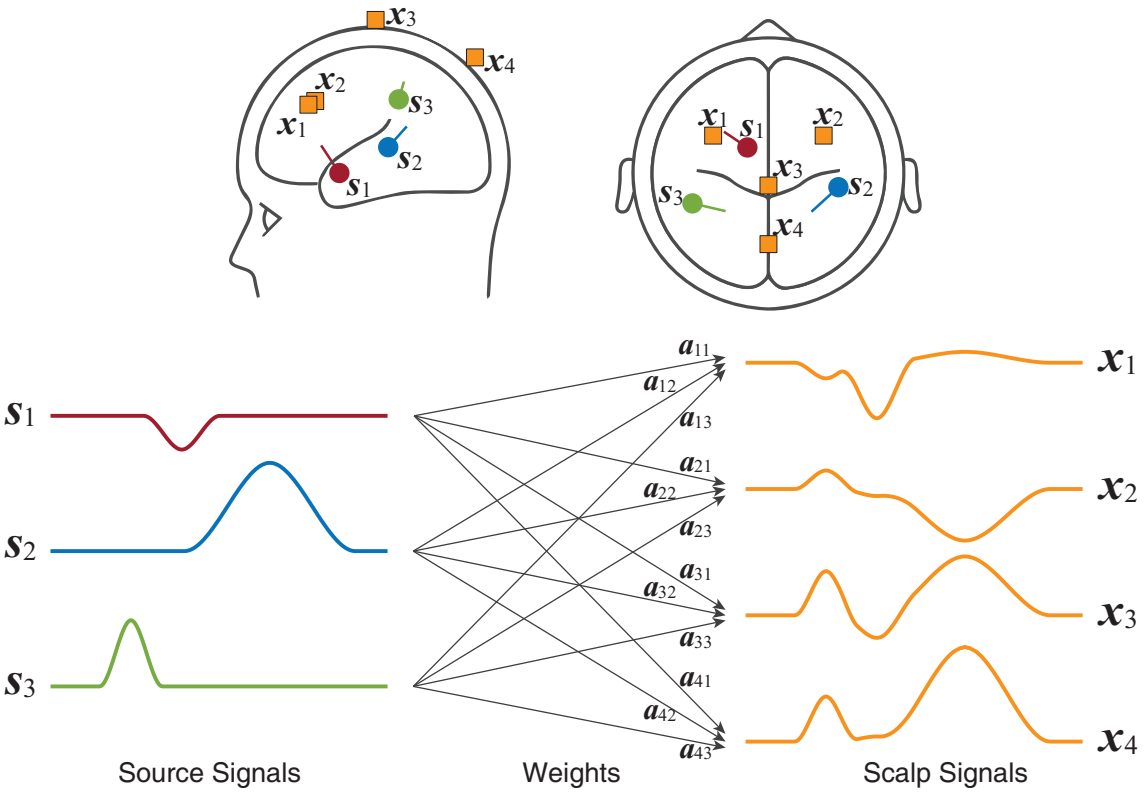



FIGURE 1 Relationship between the source signals and the recorded scalp signals. $s_j$ denotes the source signal series, $x_i$ denotes the recorded signal series and $a_{ij}$ denotes the weighting parameter.

nent analysis (ICA) (Hyvärinen and Oja, 2000; Hyvärinen, 2013). Let $Y = [y_1, y_2, \ldots, y_N] \in \mathbb{R}^{T \times N}$ denote the separated source matrix that contains mutually independent components (IC activations), and let $W$ denote the unmixing matrix. The ICA process can be represented by

$$Y^T = WX^T. \tag{2}$$

Equivalently, we can write the inverse process as

$$X^T = W^{-1}Y^T. \tag{3}$$

Here, the unmixing matrix $W$ is a matrix of spatial filters learned by ICA from the EEG scalp data, and the mixing matrix $W^{-1}$ is a matrix of scalp maps associated with each of the independent components. In addition, don't confuse the mixing matrix $A$ in (1) and the mixing matrix $W^{-1}$ in (3). $A$ is the projection matrix of weights from brain sources to EEG scalp data, which is unknown and usually difficult to compute without source localization method. Whereas $W^{-1}$ is the projection matrix of weights from independent components to EEG scalp data, which can be learned by ICA algorithm.

ICA has been widely used in EEG signal processing. One important role of ICA is artifact detection and removal (Delorme et al., 2007; Radüntz et al., 2017). Since the mixed EEG matrix $X$ consists of two dimensions of space and time, ICA can be performed in two different ways (Zhou et al., 2016): temporal ICA that

takes the data points in temporal dimension as variables and extracts independent time series; spatial ICA that takes the data points in spatial dimension as variables and extracts independent spatial maps. Because of the high temporal resolution, temporal ICA is more common for EEG. More information about ICA for EEG signal processing can be found in Makeig and Onton (2012). Latest tutorials on ICA algorithm can be found in Shlens (2014a) and Tharwat (2018).

#### 1.1.2.2 Principal Components Analysis

Principal component analysis (PCA) is a fundamental dimension reduction method based on singular value decomposition or eigenvalue decomposition. Brief tutorials on PCA can be found in Shlens (2014b) and Smith (2002). Using PCA, the separated components are orthogonal and uncorrelated to each other. PCA also includes two types (Dien, 2012): temporal ICA and spatial PCA. The application of PCA to ERP data analysis can be found in Dien et al. (2005) and Dien (2012).

PCA dimension reduction is a common practice for EEG data in the preprocessing procedures, which is usually followed by ICA (Delorme and Makeig, 2004). However, recently, it is reported that PCA dimension reduction might affect both the dipolarity and stability of independent components (ICs) extracted from high-density EEG data, and degrade the overall capability of ICA to separate functionally identifiable brain and non-brain (artifact) source activities (Artoni et al., 2018, 2019). It seems that PCA-based dimension reduction of EEG should be carefully considered and tested before data preprocessing for ICA decomposition.

#### 1.1.2.3 Nonnegative Matrix Factorization

Nonnegative matrix factorization (NMF) is a constrained BSS method, by which the separated components are required to be nonnnegative. Since in many real-world applications both the data and the intrinsic components are negative, NMF is a very important signal processing method in many areas (Cichocki et al., 2009; Wang and Zhang, 2013; Zhou et al., 2014; Gillis, 2015). NMF was mainly popularized by the paper of Lee and Seung (1999), and had been applied to a wide range of applications, such as image processing (Lee and Seung, 1999), document clustering (Xu et al., 2003; Cai et al., 2011), hyperspectral unmixing (Jia and Qian, 2009; Qian et al., 2011), microarray data analysis (Kim and Park, 2007; Esposito et al., 2019).

EEG data analysis is also one typical application of NMF (Lee and Choi, 2009; Mørup et al., 2006). One time series of EEG data can be converted into a nonnegative two-dimension spectrogram, which can be further reshaped into a one-dimension vector. By joining the vectorized spectrograms from different channels, a nonnegative matrix is generated with two dimensions of space and reshaped spectrogram. Afterwards, NMF can be applied, by which the extracted components in space dimension are the topographies and the extracted components in reshaped spectrogram dimension can be reshaped back to two-dimension spectragrams.

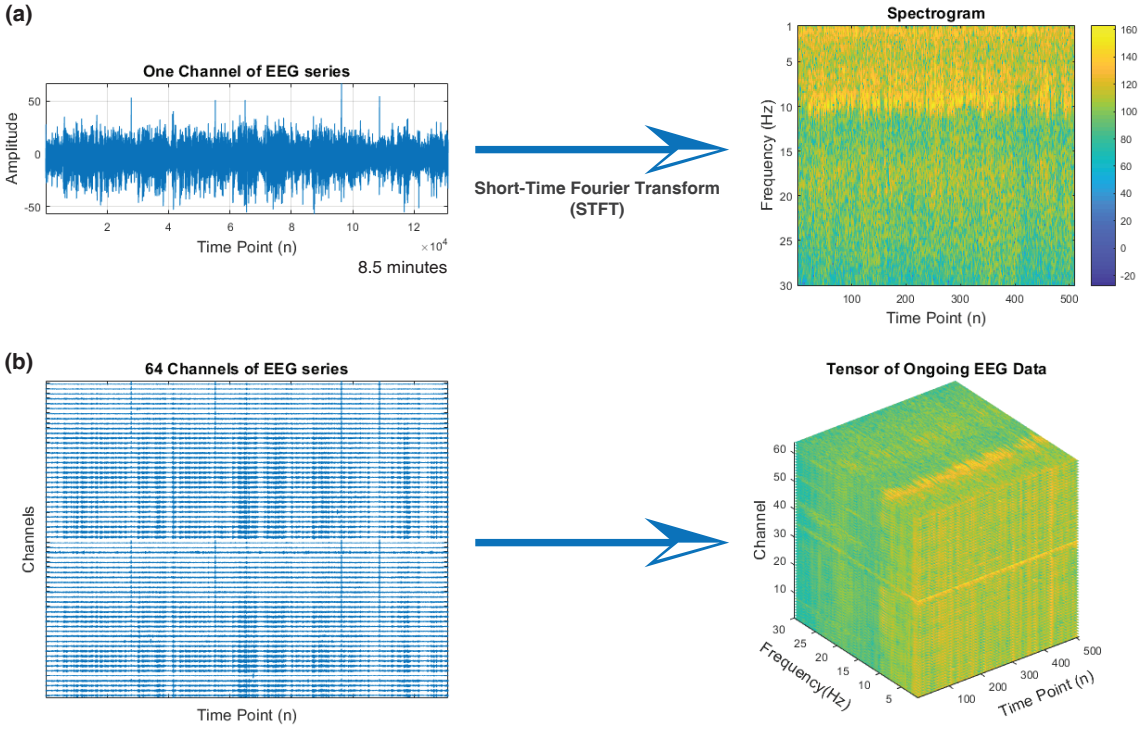



FIGURE 2 Tensor representation of ongoing EEG data. (a) shows the transformation of one channel EEG time series to a spectrogram using short-time Fourier transform. (b) shows the generation of a third-order tensor from 64 channels of EEG times series.

In this section, we mainly introduce one-way (vector-based) and two-way (matrix-based) methods for EEG data analysis. In the next section, we will introduce the multiway (tensor-based) representation of EEG data.

### 1.1.3 Tensor Representation of EEG

Tensor is a representation of multiway (multidimensional) data. The two-way EEG data recorded on the scalp only contain two modes of channel and time. By time-frequency representation, the time series on each channel can be transformed into a two-way spectrogram with two modes of time and frequency. After gathering the spectrograms of all channels, a third-order (three-way) tensor is generated, which has three modes of channel, frequency and time. The generation of a third-order ongoing EEG tensor is illustrated in **Figure 2**. If EEG data consist of more categories of information, a higher-order tensor will be generated. In an experiment, especially ERP experiment, the data can be possibly represented by a higher-order tensor including several modes such as channel/space, frequency, time, trial, subject, condition and group (Cong et al., 2015a).

For high-order EEG data, tensor decomposition is just the right way to extract intrinsic components that are linked to cognitive processes, whereas conventional methods and matrix-based methods are far inferior to tensor decomposition (Cong et al., 2015a; Wang et al., 2018b). The advantage of tensor decomposition will be introduced in the following section.

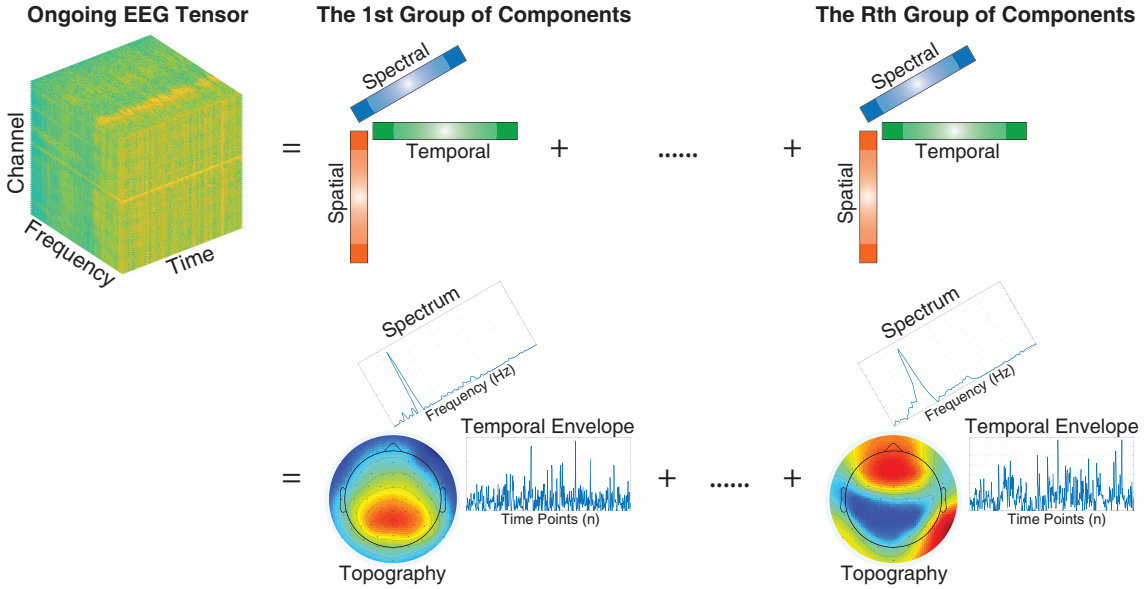



FIGURE 3 Decomposition of a third-order ongoing EEG tensor using NCP. The extracted nonnegative components are related to some brain activities: the spatial components are the topographies that indicate the locations on the scalp; the spectral components are the spectra that reveal the most prominent frequency bands; the temporal components are the envelopes that exhibit the temporal evolution series.

## 1.2 Research Motivation

Tensor decomposition, as a versatile tool for signal processing and machine learning (Cichocki et al., 2015; Sidiropoulos et al., 2017), has become more and more popular for EEG data processing and cognitive neuroscience in recent years (Cong et al., 2015a; Zhou et al., 2016; Mahyari et al., 2017; Idaji et al., 2017; Wang et al., 2018a,b).

If EEG tensor is generated by time-frequency representation, the multiway data must be nonnegative. In the nonnegative tensor, the intrinsic components sometimes are also nonnegative. Therefore, nonnegative constraint should be incorporated into the tensor decomposition. The nonnegative CANDECOMP/PARAFAC (NCP), as a fundamental decomposition method, is just suitable to analyze this type of nonnegative tensor. **Figure 3** illustrates the results of NCP decomposition on a third-order ongoing EEG tensor, which includes three modes of channel, frequency and time. The extracted nonnegative EEG components using NCP are related to some brain activities and have specific meanings: the spatial components are the topographies that indicate the activation locations on the scalp; the spectral components are the spectra that reveal the most prominent frequency bands; the temporal components are the envelopes that exhibit the temporal evolution series.

One major advantage of tensor decomposition for high-order data analysis is that the interactive information among all modes can be well-preserved. By observing **Figure 3** carefully, it can be found that one group of components

are composed of three parts including spatial component, spectral component and temporal component. In fact, these components in one group explain some brain activity from different perspectives. Specifically, the EEG tensor in **Figure 3** has three modes of channel, frequency and time, which interact with each other in the tensor form. Therefore, tensor decomposition will extract three components from all the three modes, which preserve the interactive information that is linked to cognitive process in the brain. If the tensor is reshaped into a two-way matrix, for example by keeping mode of space and merging modes of frequency and time, the interaction information of space-to-frequency, space-to-time and time-to-frequency will be lost.

On the other hand, it is clear to see from **Figure 3** that some components, such as the spectra, are not only nonnegative but also sparse. Hence additional regularization, such as sparsity, will enhance the extraction of meaningful components. Tensor decomposition can be improved by additional regularization items.

However, there are many problems for high-order EEG tensor decomposition, which need careful consideration and further investigation. We list four key issues below.

- The first issue is what type of regularization item should be incorporated into the tensor decomposition model, e.g. the concrete mathematical norm for sparse regularization, which is from the structural point of view.

- The second is which optimization method should be used to guarantee the convergence and stability of the designed decomposition model, which is from the mathematical point of view.

- The third is how to process large-scale tensor data efficiently, which is from the computational point of view.

- The fourth is whether the extracted EEG features are related to concrete brain activities of cognitive processes, which is from the perspective of real applications in brain research and cognitive neuroscience.

Finding the solutions to the above four issues is the motivation of the study in this dissertation.

## 1.3 Research Approaches

In this section, we introduce our proposed approaches to solve the issues mentioned in previous section.

### 1.3.1 Constraint and Regularization

According the nonnegative and sparse properties of the EEG components, we construct the CP decomposition with both nonnegative constraint and sparse regularization, which is abbreviated to sparse NCP. We employ $l_1$-norm as the sparse regularization item, which can yield strong sparsity (Donoho, 2006).

### 1.3.2 Optimization methods

In this dissertation, The main framework for solving sparse NCP is block coordinate descent (BCD) (Bertsekas, 2016; Xu and Yin, 2013; Kim et al., 2014), which is a popular method to solve large-scale tensor decomposition problem. In BCD framework, tensor decomposition is optimized by updating each factor matrix alternatively as a subproblem. We investigated five popular and latest optimization methods to solve the subproblem of tensor decomposition, which are listed as follows:

| Abbreviation | Full Name |
|---|---|
| MU | Multiplicative Update |
| ANLS/ANQP | Alternating Nonnegative Least Squares<br>Alternating Nonnegative Quadratic Programming |
| HALS | Hierarchical Altering Least Squares |
| APG | Alternating Proximal Gradient |
| ADMM | Alternating Direction Method of Multipliers |

MU has a significant influence on nonnegative matrix factorization (NMF) in the past decades (Lee and Seung, 1999), and is still popular today (Cai et al., 2011; Jiang et al., 2019). Hence we extend MU to solve the constrained and regularized tensor decomposition problem.

ANLS/ANQP and HALS are very efficient methods for tensor decomposition (Cichocki et al., 2009; Kim and Park, 2012; Kim et al., 2014). In order to improve the stability, we incorporate proximal algorithm (Li et al., 2013; Wang et al., 2019) into ANLS/ANQP and HALS.

APG and ADMM are also popular for solving tensor decomposition (Xu and Yin, 2013; Zhang et al., 2016; Huang et al., 2016). Both the two methods have proximal operator in the optimization process, which is very flexible to handle a lot of differentiable and non-differentiable regularization items.

### 1.3.3 Inexact Block Coordinate Descent Scheme

Because of the large scale of many EEG tensor data, an algorithm with efficient computation and low time cost is favorable. One the one hand, an excellent optimization algorithm will improve the computational performance of tensor decomposition. On the other hand, some implementation schemes will further

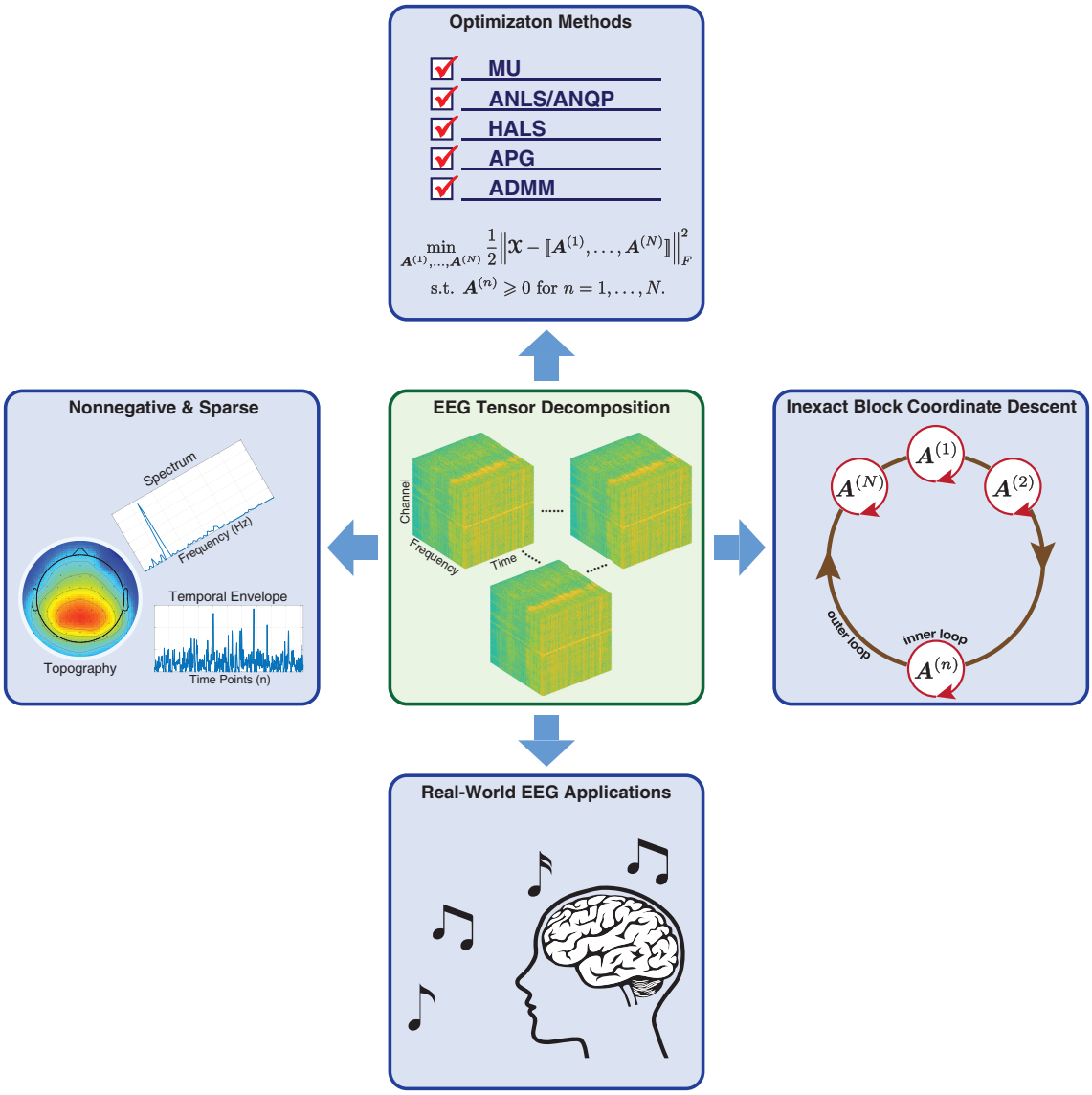



FIGURE 4 Diagram of research approaches in this dissertation.

increase the running speed. Using a optimization method, the subproblem in BCD framework can also be iterated by a limited number of times without yielding a precise solution. This scheme is called inexact block coordinate descent (inexact BCD) (Vervliet and Lathauwer, 2019; Gillis and Glineur, 2012). In this dissertation, we implement all the five optimization methods introduced in section 1.3.2 by the inexact BCD scheme.

### 1.3.4 Real-World EEG Applications

In this dissertation, we apply the constrained and regularized tensor decomposition to several real-world EEG datasets collected in cognitive neuroscience experiments. Our purpose is to extract meaningful EEG features that are linked to cognitive processes.

Firstly, NCP is utilized to decompose a fifth-order event-related potential (ERP) tensor, which was collected by proprioceptive stimuli on human left and

right hand. Many groups of EEG components in five modes, including space, frequency, time, subject and condition, will be extracted. Using the interactive information among the five modes, we search for those EEG features that are obviously elicited by the stimuli on hands.

Secondly, we analyze a set of ongoing EEG tensors using sparse NCP. The data were collected by naturalistic and continuous music stimulus. After decomposition, nonnegative and sparse EEG components will be extracted. Using Monte Carlo method and permutation tests, we select those EEG features whose temporal series are significantly correlated with music features. These components are assumed to be the brain activities elicited by music.

Finally, we analyze two modalities of ongoing EEG tensor and music signals simultaneously by N-way partial least square (N-PLS).

The diagram of all research approaches in this dissertation is shown in **Figure 4**.

## 1.4 Structure of Dissertation

The rest of this dissertation is organized as follows.

Chapter 2 presents the theoretical aspects of constrained tensor decomposition, especially the sparse NCP.

Chapter 3 describes the applications of the constrained tensor decomposition on real EEG tensor analysis, in which meaningful EEG components linked to some brain activity are extracted.

Chapter 4 lists the contribution of each attached paper in this dissertation.

Chapter 5 concludes the whole research work of this dissertation.

# 2 NONNEGATIVE TENSOR DECOMPOSITION

A tensor is a multiway data form or multidimensional array. Nowadays, with the widespread use of multisensor technology and the fast increase of data size, more and more data appear as multiway form. Tensor decomposition is a powerful tool to analyze multiway data and extract intrinsic components from them (Kolda and Bader, 2009; Sidiropoulos et al., 2017). In real-world applications, many tensor data are nonnegative, such as hyperspectral data (Veganzones et al., 2016), electroencephalograph (EEG) data (Cong et al., 2015a; Wang et al., 2018b), fluorescence excitation-emission matrix (EEM) data (Elcoroaristizabal et al., 2015; Vu et al., 2017), neural data (Williams et al., 2018), and many other multiway tensor data (Mørup, 2011; Acar and Yener, 2009). In most cases, the intrinsic components in the nonnegative tensor are also nonnegative. For example, the spectral components in EEG data and the sample concentration components in EEM data are nonnegative. In order to extract meaningful nonnegative component, nonnegative constraint is necessary to be incorporated into tensor decomposition. Nonnegative CANDECOMP/PARAFAC (NCP) is one of the most important decomposition methods for processing nonnegative tensor data.

In this chapter, basic algorithms and computations of tensor decomposition will be introduced firstly. Moreover, we will introduce our contributions to NCP, whose mathematical model is combined with regularization items.

## 2.1 Preliminary

In this dissertation, a vector is denoted by boldface lowercase letter, such as $\boldsymbol{x}$; a matrix is denoted by boldface uppercase letter, such as $\boldsymbol{X}$; and a high order tensor is denoted by boldface Euler script letter, such as $\boldsymbol{\mathcal{X}}$. Operator $\circ$ represents outer product of vectors, $\odot$ represents the Khatri-Rao product, $*$ represents the Hadamard product that is the elementwise matrix product, $\langle\ \rangle$ represents inner product, $[\![\ ]\!]$ represents Kruskal oprator and $[\ ]_+$ represents nonnegative projection. $\|\ \|_F$ denotes Frobenius norm, and $\|\ \|_1$ denotes $l_1$-norm. Basics

of tensor computation and multi-linear algebra can be found in review papers by Kolda and Bader (2009) and Sidiropoulos et al. (2017). Some basic and useful computations are introduced below.

**Mode-$n$ matricization**: An $n$th-order tensor is usually represented by $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$, where $I_n$ is the mode-$n$ size, $n = 1, \ldots, N$. The mode-$n$ matricization will reshape tensor $\mathcal{X}$ into a matrix $\boldsymbol{X}_{(n)} \in \mathbb{R}^{I_n \times \prod_{\tilde{n}=1, \tilde{n} \neq n}^{N} I_{\tilde{n}}}$.

**Kronecker product:** Given matrices $\boldsymbol{A} \in \mathbb{R}^{I \times J}$ and $\boldsymbol{B} \in \mathbb{R}^{K \times L}$, the Kronecker product $\boldsymbol{A} \otimes \boldsymbol{B} \in \mathbb{R}^{IK \times JL}$ is computed by

$$\begin{aligned} \boldsymbol{A} \otimes \boldsymbol{B} &= \begin{bmatrix} a_{11}\boldsymbol{B} & a_{12}\boldsymbol{B} & \ldots & a_{1J}\boldsymbol{B} \\ a_{21}\boldsymbol{B} & a_{22}\boldsymbol{B} & \ldots & a_{2J}\boldsymbol{B} \\ \vdots & \vdots & \ddots & \vdots \\ a_{I1}\boldsymbol{B} & a_{I2}\boldsymbol{B} & \ldots & a_{IJ}\boldsymbol{B} \end{bmatrix} \\ &= \begin{bmatrix} \boldsymbol{a}_1 \otimes \boldsymbol{b}_1 & \boldsymbol{a}_1 \otimes \boldsymbol{b}_2 & \boldsymbol{a}_1 \otimes \boldsymbol{b}_3 & \ldots & \boldsymbol{a}_J \otimes \boldsymbol{b}_{L-1} & \boldsymbol{a}_J \otimes \boldsymbol{b}_L \end{bmatrix}. \end{aligned}$$

**Khatri-Rao product:** Given matrices $\boldsymbol{A} \in \mathbb{R}^{I \times K}$ and $\boldsymbol{B} \in \mathbb{R}^{J \times K}$, the Khatri-Rao product $\boldsymbol{A} \odot \boldsymbol{B} \in \mathbb{R}^{IJ \times K}$ is computed by

$$\boldsymbol{A} \odot \boldsymbol{B} = \begin{bmatrix} \boldsymbol{a}_1 \otimes \boldsymbol{b}_1 & \boldsymbol{a}_2 \otimes \boldsymbol{b}_2 & \cdots & \boldsymbol{a}_K \otimes \boldsymbol{b}_K \end{bmatrix}.$$

**Hadamard product:** Hadamard product is the elementwise matrix product. Given matrices $\boldsymbol{A} \in \mathbb{R}^{I \times J}$ and $\boldsymbol{B} \in \mathbb{R}^{I \times J}$, the Khatri-Rao product $\boldsymbol{A} * \boldsymbol{B} \in \mathbb{R}^{I \times J}$ is computed by

$$\boldsymbol{A} * \boldsymbol{B} = \begin{bmatrix} a_{11}b_{11} & a_{12}b_{12} & \cdots & a_{1J}b_{1J} \\ a_{21}b_{21} & a_{22}b_{22} & \cdots & a_{2J}b_{2J} \\ \vdots & \vdots & \ddots & \vdots \\ a_{I1}b_{I1} & a_{I2}b_{I2} & \cdots & a_{IJ}b_{IJ} \end{bmatrix}$$

One interesting property is that $(\boldsymbol{A} \odot \boldsymbol{B})^T (\boldsymbol{A} \odot \boldsymbol{B}) = \left(\boldsymbol{A}^T \boldsymbol{A}\right) * \left(\boldsymbol{B}^T \boldsymbol{B}\right)$.

In the following, we will introduce two fundamental tensor decomposition methods.

**CANDECOMP/PARAFAC (CP) Decomposition:** The CP decomposition has several names and associated abbreviations, such as canonical polyadic decomposition (CPD), canonical decomposition (CANDECOMP), parallel factor analysis (PARAFAC) and CANDECOMP/PARAFAC (CP) Decomposition (Kolda and Bader, 2009; Liavas and Sidiropoulos, 2015). The last one is suggested by Kolda and Bader (2009) and will be used in the whole of this dissertation.

Given an $N$th-order tensor $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$, CP decomposition is to solve the following minimization problem:

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

where $\boldsymbol{A}^{(n)} \in \mathbb{R}^{I_n \times R}$ for $n = 1, \ldots, N$ are the estimated factor matrices in different modes, and $R$ is the selected rank-1 tensor number (component number). The estimated factor matrices in Kruskal operator can be represented by the sum of $R$ rank-1 tensors in outer product form:

$$[\![ \boldsymbol{A}^{(1)}, \ldots, \boldsymbol{A}^{(N)} ]\!] = \sum_{r=1}^{R} \widetilde{\mathcal{X}}_r = \sum_{r=1}^{R} \boldsymbol{a}_r^{(1)} \circ \cdots \circ \boldsymbol{a}_r^{(N)}, \tag{5}$$

where $\boldsymbol{a}_r^{(n)}$ represents the $r$th column of $\boldsymbol{A}^{(n)}$.

The mode-$n$ unfolding of the estimated tensor in Kruskal operator $[\![ \boldsymbol{A}^{(1)}, \ldots, \boldsymbol{A}^{(N)} ]\!]$ can be written as $\boldsymbol{A}^{(n)} \left( \boldsymbol{B}^{(n)} \right)^T$, in which $\boldsymbol{B}^{(n)} = \left( \boldsymbol{A}^{(N)} \odot \cdots \odot \boldsymbol{A}^{(n+1)} \odot \boldsymbol{A}^{(n-1)} \odot \cdots \odot \boldsymbol{A}^{(1)} \right) \in \mathbb{R}^{\prod_{\tilde{n}=1, \tilde{n} \neq n}^{N} I_{\tilde{n}} \times R}$.

**Tucker Decomposition:** Given an $N$th-order tensor $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$, a rank-$(R_1, R_2, \ldots, R_N)$ Tucker decomposition is to solve the following minimization problem:

$$\begin{aligned} \min_{\boldsymbol{A}^{(1)}, \ldots, \boldsymbol{A}^{(N)}} & \frac{1}{2} \left\| \mathcal{X} - [\![ \mathcal{G}; \boldsymbol{A}^{(1)}, \ldots, \boldsymbol{A}^{(N)} ]\!] \right\|_F^2 \\ \text{s.t. } & \mathcal{G} \in \mathbb{R}^{R_1 \times R_2 \times \ldots \times R_N} \\ & \boldsymbol{A}^{(n)} \in \mathbb{R}^{I_n \times R_n} \text{ for } n = 1, \ldots, N, \end{aligned} \tag{6}$$

where $R_n$ is the $n$-mode rank, $n = 1, \ldots, N$.

The work in this dissertation is mainly based on CP decomposition. More knowledge of tensor computation and multilinear algebra can be found in review papers by Kolda and Bader (2009) and Sidiropoulos et al. (2017).

## 2.2 Nonnegative CANDECOMP/PARAFAC Decomposition (NCP)

If the intrinsic components in tensor data are nonnegative, nonnegative constraints should be incorporated into tensor decomposition. Nonnegative CANDECOMP/PARAFAC Decomposition (NCP) is one of the most important model of constrained tensor decomposition.

### 2.2.1 Mathematical Model

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

$$\begin{aligned} \min_{\boldsymbol{A}^{(1)}, \ldots, \boldsymbol{A}^{(N)}} & \frac{1}{2} \left\| \mathcal{X} - [\![ \boldsymbol{A}^{(1)}, \ldots, \boldsymbol{A}^{(N)} ]\!] \right\|_F^2 \\ \text{s.t. } & \boldsymbol{A}^{(n)} \geqslant 0 \text{ for } n = 1, \ldots, N. \end{aligned} \tag{7}$$

The above optimization problem of NCP is usually solved by block coordinate descent (BCD) framework (Xu and Yin, 2013; Kim et al., 2014). In BCD

framework, factor $\boldsymbol{A}^{(n)}$ is updated alternatively by a subproblem in every iteration, which equals to the following minimization problem:

$$\begin{aligned} \min_{\boldsymbol{A}^{(n)}} \mathcal{F}\left(\boldsymbol{A}^{(n)}\right) =& \frac{1}{2}\left\| \boldsymbol{X}_{(n)} - \boldsymbol{A}^{(n)}\left(\boldsymbol{B}^{(n)}\right)^T \right\|_F^2 \\ & \text{s.t. } \boldsymbol{A}^{(n)} \geqslant 0. \end{aligned} \tag{8}$$

Furthermore, the objective function in (8) can be represented using the following form of trace:

$$\begin{aligned} \mathcal{F}\left(\boldsymbol{A}^{(n)}\right) =& \frac{1}{2}\text{tr}\left[\boldsymbol{X}_{(n)}^T \boldsymbol{X}_{(n)}\right] - \text{tr}\left[\left(\boldsymbol{A}^{(n)}\right)^T \boldsymbol{X}_{(n)} \boldsymbol{B}^{(n)}\right] \\ & + \frac{1}{2}\text{tr}\left[\boldsymbol{A}^{(n)}\left(\boldsymbol{B}^{(n)}\right)^T \boldsymbol{B}^{(n)}\left(\boldsymbol{A}^{(n)}\right)^T\right]. \end{aligned} \tag{9}$$

### 2.2.2 Partial Derivative

The partial derivative (or partial gradient) of $\mathcal{F}\left(\boldsymbol{A}^{(n)}\right)$ with respect to $\boldsymbol{A}^{(n)}$ is always used during computation,

$$\frac{\partial}{\partial \boldsymbol{A}^{(n)}} \mathcal{F}\left(\boldsymbol{A}^{(n)}\right) = \boldsymbol{A}^{(n)}\left[\left(\boldsymbol{B}^{(n)}\right)^T \boldsymbol{B}^{(n)}\right] - \boldsymbol{X}_{(n)} \boldsymbol{B}^{(n)}, \tag{10}$$

where $\boldsymbol{X}_{(n)} \boldsymbol{B}^{(n)}$ is called the *Matricized Tensor Times Khatri-Rao Product* (MTTKRP) (Bader and Kolda, 2008). The item $\left(\boldsymbol{B}^{(n)}\right)^T \boldsymbol{B}^{(n)}$ can be computed efficiently by

$$\begin{aligned} \left(\boldsymbol{B}^{(n)}\right)^T \boldsymbol{B}^{(n)} =& \left[\left(\boldsymbol{A}^{(N)}\right)^T \boldsymbol{A}^{(N)}\right] * \cdots * \left[\left(\boldsymbol{A}^{(n+1)}\right)^T \boldsymbol{A}^{(n+1)}\right] \\ & * \left[\left(\boldsymbol{A}^{(n-1)}\right)^T \boldsymbol{A}^{(n-1)}\right] * \cdots * \left[\left(\boldsymbol{A}^{(1)}\right)^T \boldsymbol{A}^{(1)}\right]. \end{aligned} \tag{11}$$

### 2.2.3 Stop Condition and Objective Function

The optimization procedures for tensor decomposition are implemented by iterations. For NCP, a sequence of $\left\{\boldsymbol{A}_k^{(1)}, \ldots, \boldsymbol{A}_k^{(N)}\right\}_{k=1}^{\infty}$ is produced at each iteration. It is necessary to terminate the iteration until some stopping condition is satisfied. Common stopping conditions include the following: predefined maximum number of iterations, predefined maximum running time, the change of objective function value, the change of relative error (data fitting) (Kolda and Bader, 2009; Xu and Yin, 2013).

#### 2.2.3.1 Objective Function and Relative Error

The computations of objective function value and relative error (data fitting) are highly correlated. In the $k$th iteration, the objective function value of NCP problem (7) is

$$\mathcal{F}_{\text{NCP}k} = \frac{1}{2}\left\| \mathcal{X} - [\![\boldsymbol{A}_k^{(1)}, \ldots, \boldsymbol{A}_k^{(N)}]\!] \right\|_F^2, \tag{12}$$

and the relative error (Xu and Yin, 2013) is defined by

$$\text{RelErr}_k = \frac{\left\| \mathcal{X} - [\![ \boldsymbol{A}_k^{(1)}, \ldots, \boldsymbol{A}_k^{(N)} ]\!] \right\|_F}{\| \mathcal{X} \|_F}. \tag{13}$$

Comparing (12) and (13), the relative error can also be computed from the objective function value directly:

$$\text{RelErr}_k = \frac{\sqrt{2\mathcal{F}_{\text{NCP}k}}}{\| \mathcal{X} \|_F}. \tag{14}$$

Meanwhile, the data fitting can be computed by

$$\text{Fit}_k = 1 - \text{RelErr}_k. \tag{15}$$

Based on the objective function value and the relative error, the stopping condition can be set by

$$|\text{RelErr}_{k-1} - \text{RelErr}_k| < \epsilon \tag{16}$$

or

$$|\mathcal{F}_{\text{NCP}k-1} - \mathcal{F}_{\text{NCP}k}| < \epsilon. \tag{17}$$

The threshold of $\epsilon$ can be set by a very small positive value, such as $1e-8$.

#### 2.2.3.2 Accelerated Computation of Objective Function

By mode-$n$ unfolding of tensor, the objective function of NCP in (12) at the $k$th iteration can be represented equivalently as the following:

$$\mathcal{F}_{\text{NCP}k} = \frac{1}{2} \left\| \boldsymbol{X}_{(n)} - \boldsymbol{A}_k^{(n)} \left( \boldsymbol{B}_k^{(n)} \right)^T \right\|_F^2. \tag{18}$$

The works of Guan et al. (2012) and Xu and Yin (2013) introduced a convenient idea to compute the objective function base on the trace computation of matrix. Inspired by this idea, we further represent the objective function in (18) by

$$\begin{aligned} \mathcal{F}_{\text{NCP}k} =& \frac{1}{2} \text{tr} \left\{ \left[ \boldsymbol{X}_{(n)} - \boldsymbol{A}_k^{(n)} \left( \boldsymbol{B}_k^{(n)} \right)^T \right]^T \left[ \boldsymbol{X}_{(n)} - \boldsymbol{A}_k^{(n)} \left( \boldsymbol{B}_k^{(n)} \right)^T \right] \right\} \\ =& \frac{1}{2} \left\{ \| \mathcal{X} \|_F^2 - 2\text{tr} \left[ \boldsymbol{A}_k^{(n)} \left( \boldsymbol{X}_{(n)} \boldsymbol{B}_k^{(n)} \right)^T \right] \right. \\ & \left. + \text{tr} \left[ \left( \left( \boldsymbol{A}_k^{(n)} \right)^T \boldsymbol{A}_k^{(n)} \right) \left( \left( \boldsymbol{B}_k^{(n)} \right)^T \boldsymbol{B}_k^{(n)} \right) \right] \right\}. \end{aligned} \tag{19}$$

Furthermore, the objective function equals to

$$\mathcal{F}_{\text{NCP}k} = \frac{1}{2} \left\{ \| \mathcal{X} \|_F^2 - 2 \sum_{j=1}^{R} \sum_{i=1}^{I_n} \widehat{\boldsymbol{N}}_{i,j} + \sum_{j=1}^{R} \sum_{i=1}^{R} \widehat{\boldsymbol{M}}_{i,j} \right\}, \tag{20}$$

where

$$\widehat{\boldsymbol{N}} = \boldsymbol{A}_k^{(n)} * \left( \boldsymbol{X}_{(n)} \boldsymbol{B}_k^{(n)} \right) \in \mathbb{R}^{I_n \times R}$$

and

$$\widehat{\boldsymbol{M}} = \left( \left( \boldsymbol{A}_k^{(n)} \right)^T \boldsymbol{A}_k^{(n)} \right) * \left( \left( \boldsymbol{B}_k^{(n)} \right)^T \boldsymbol{B}_k^{(n)} \right) \in \mathbb{R}^{R \times R}.$$

Here, $*$ is the Hadamard product.

Since tensor data usually consist of a large amount of data points, the computation of the objective function at each iteration will be time consuming by (12) or (18). For example, in (18) the computational complexity of $\boldsymbol{A}_k^{(n)} \left( \boldsymbol{B}_k^{(n)} \right)^T$ is $O(R \times \prod_{n=1}^{N} I_n)$. On the other hand, according to (10), we find that the items of $\boldsymbol{X}_{(n)} \boldsymbol{B}_k^{(n)}$ and $\left( \boldsymbol{B}_k^{(n)} \right)^T \boldsymbol{B}_k^{(n)}$ have been computed in advance in order to update $\boldsymbol{A}_k^{(n)}$. Therefore, these two items can be employed directly to compute the objective function by (20). In (20), the computational complexity of $\widehat{\boldsymbol{N}}$ and $\widehat{\boldsymbol{M}}$ is only $O(I_n R + R^2)$, which has been reduced significantly.

## 2.3 NCP With Regularization Items

In real applications, adding regularization item to tensor decomposition will improve the performance and make the extracted components more meaningful. NCP with regularization item can be expressed in the following minimization problem:

$$\begin{gathered} \min_{\boldsymbol{A}^{(1)}, \ldots, \boldsymbol{A}^{(N)}} \frac{1}{2} \left\| \mathcal{X} - [\![ \boldsymbol{A}^{(1)}, \ldots, \boldsymbol{A}^{(N)} ]\!] \right\|_F^2 + \sum_{n=1}^{N} \lambda_n \phi_n \left( \boldsymbol{A}^{(n)} \right) \\ \text{s.t. } \boldsymbol{A}^{(n)} \geqslant 0 \text{ for } n = 1, \ldots, N, \end{gathered} \tag{21}$$

where $\lambda_n > 0$ is the positive regularization parameter. Common regularization items include Frobenius norm, sparse regularization, graph regularization, rank regularization, proximal regularization (proximal algorithm), and so on (Cai et al., 2011; Boyd, 2011; Kim and Park, 2012; Shang et al., 2017).

### 2.3.1 Frobenius Norm Regularization

The Frobenius norm regularization can improve the stability of NCP and prevent the elements in factor $\boldsymbol{A}^{(n)}$ to grow too large (Kim et al., 2014). If let $\phi_n \left( \boldsymbol{A}^{(n)} \right) = \left\| \boldsymbol{A}^{(n)} \right\|_F^2$, $\lambda_n = \frac{\alpha_n}{2}$ in (21), the objective function of the subproblem becomes

$$\mathcal{F}_{\text{Frob}} \left( \boldsymbol{A}^{(n)} \right) = \frac{1}{2} \left\| \boldsymbol{X}_{(n)} - \boldsymbol{A}^{(n)} \left( \boldsymbol{B}^{(n)} \right)^T \right\|_F^2 + \frac{\alpha_n}{2} \left\| \boldsymbol{A}^{(n)} \right\|_F^2, \tag{22}$$

The partial derivative of $\mathcal{F}_{\text{Frob}}\left(\boldsymbol{A}^{(n)}\right)$ to $\boldsymbol{A}^{(n)}$ is

$$\frac{\partial}{\partial \boldsymbol{A}^{(n)}}\mathcal{F}_{\text{Frob}}\left(\boldsymbol{A}^{(n)}\right) = \boldsymbol{A}^{(n)}\left[\left(\boldsymbol{B}^{(n)}\right)^{T}\boldsymbol{B}^{(n)} + \alpha_n \boldsymbol{I}_R\right] - \boldsymbol{X}_{(n)}\boldsymbol{B}^{(n)}, \tag{23}$$

where $\boldsymbol{I}_R \in \mathbb{R}^{R\times R}$ is an identity matrix. It is clear to see $\left[\left(\boldsymbol{B}^{(n)}\right)^{T}\boldsymbol{B}^{(n)} + \alpha_n \boldsymbol{I}_R\right]$ is of full rank, which can prevent rank deficiency. Hence, the stability of NCP is enhanced. However, one disadvantage is that the solution of NCP with Frobenius regularization is not exactly the same as the solution of original NCP in (7).

### 2.3.2 Proximal Algorithm

Let $\phi_n\left(\boldsymbol{A}^{(n)}\right) = \left\|\breve{\boldsymbol{A}}^{(n)} - \boldsymbol{A}^{(n)}\right\|_F^2$ and $\lambda_n = \frac{\alpha_n}{2}$ in (21), where $\breve{\boldsymbol{A}}^{(n)} \in \mathbb{R}^{I_n\times R}$ is the former version of $\boldsymbol{A}^{(n)}$ in previous iteration. The objective function of the subproblem becomes

$$\mathcal{F}_{\text{Prox}}\left(\boldsymbol{A}^{(n)}\right) = \frac{1}{2}\left\|\boldsymbol{X}_{(n)} - \boldsymbol{A}^{(n)}\left(\boldsymbol{B}^{(n)}\right)^{T}\right\|_F^2 + \frac{\alpha_n}{2}\left\|\breve{\boldsymbol{A}}^{(n)} - \boldsymbol{A}^{(n)}\right\|_F^2. \tag{24}$$

The partial derivative of $\mathcal{F}_{\text{Prox}}\left(\boldsymbol{A}^{(n)}\right)$ to $\boldsymbol{A}^{(n)}$ is

$$\begin{aligned}\frac{\partial}{\partial \boldsymbol{A}^{(n)}}\mathcal{F}_{\text{Prox}}\left(\boldsymbol{A}^{(n)}\right) =& \boldsymbol{A}^{(n)}\left[\left(\boldsymbol{B}^{(n)}\right)^{T}\boldsymbol{B}^{(n)} + \alpha_n \boldsymbol{I}_R\right] \\ &- \left[\boldsymbol{X}_{(n)}\boldsymbol{B}^{(n)} + \alpha_n \breve{\boldsymbol{A}}^{(n)}\right].\end{aligned} \tag{25}$$

The full rank item $\left[\left(\boldsymbol{B}^{(n)}\right)^{T}\boldsymbol{B}^{(n)} + \alpha_n \boldsymbol{I}_R\right]$ also exists in the derivative, which will guarantee the stability of NCP. What's more, an attracting property is that the solution of NCP with proximal algorithm is equivalent to the solution of original NCP (7) (Li et al., 2013; Wang et al., 2019). There is an intuitive way to understand this property. After adequate iterations, $\boldsymbol{A}^{(n)}$ is very close to $\breve{\boldsymbol{A}}^{(n)}$. Hence, the term $\left\|\breve{\boldsymbol{A}}^{(n)} - \boldsymbol{A}^{(n)}\right\|_F^2$ approaches zeros, which makes the solutions of proximal algorithm version and original version of NCP are equivalent. More theoretical analysis of proximal algorithm for tensor decomposition can be found in (Li et al., 2013). The combination of NCP with proximal algorithm can be find in our original paper **PII**, which shows significant improvement of efficiency and stability.

### 2.3.3 Sparse Regularization

Sparse regularization are very popular for NMF and NCP problems. There are many regularization items that can impose sparsity (Bach et al., 2012). The $l_1$-norm is one of the most classical and important sparse regularization items. We will introduce more about NCP with sparse regularization in the following sections.

## 2.4 Sparse Nonnegative CP Decompositioin

One contribution in this dissertation is the algorithm development of NCP with sparse regularization (sparse NCP). In many cases, the extracted components by NCP are not only nonnegative but also sparse. Take EEG tensor decomposition for example, the extracted spectral components are usually very sparse (Wang et al., 2018a), which represents the narrow-band frequencies related to brain activities. However, the sparsity of NCP is not controllable. It is necessary to add explicit sparse regularization to NCP.

There are mainly two methods for imposing sparsity. One method is to projecte components into sparse vectors at some sparsity level (Hoyer, 2004). However, this method keeps all components at the same fixed sparsity level, which is not in line with the true sparsity level of different components in real-world data. Another method is to incorporate sparse regularization items into the optimization model. The $l_1$-norm is a conventional and effective regularizer to impose sparsity for signal processing(Bruckstein et al., 2009), since for most underdetermined linear equations the optimization problem with $l_1$-norm regularization can yield strong sparsity (Donoho, 2006). In addition, the $l_q$-norm ($0 < q < 1$) item (Xu et al., 2012; Qian et al., 2011; Sigurdsson et al., 2014) and the trace-norm based rank regularization item (Shang et al., 2017) also have become favourable to promote sparsity for NMF in recent years. In this dissertation, we only employ the classical $l_1$-norm as the sparse regularization for NCP.

### 2.4.1 Mathematical Model

We describe the sparse NCP model as follows. Given a nonnegative $N$th-order tensor $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$ and a positive number $R$, the sparse NCP is to solve the following minimization problem:

$$\begin{aligned} \min_{\boldsymbol{A}^{(1)},\ldots,\boldsymbol{A}^{(N)}} & \frac{1}{2}\left\| \mathcal{X} - [\![ \boldsymbol{A}^{(1)}, \ldots, \boldsymbol{A}^{(N)} ]\!] \right\|_F^2 + \sum_{n=1}^{N} \beta_n \sum_{r=1}^{R} \left\| \boldsymbol{a}_r^{(n)} \right\|_1 \\ & \text{s.t. } \boldsymbol{A}^{(n)} \geqslant 0 \text{ for } n = 1, \ldots, N, \end{aligned} \tag{26}$$

In BCD framework, nonnegative factor $\boldsymbol{A}^{(n)}$ is updated alternatively in each outer iteration by the following minimization problem:

$$\begin{aligned} \min_{\boldsymbol{A}^{(n)}} \mathcal{F}\left(\boldsymbol{A}^{(n)}\right) = & \frac{1}{2}\left\| \boldsymbol{X}_{(n)} - \boldsymbol{A}^{(n)} \left(\boldsymbol{B}^{(n)}\right)^T \right\|_F^2 + \beta_n \sum_{r=1}^{R} \left\| \boldsymbol{a}_r^{(n)} \right\|_1 \\ & \text{s.t. } \boldsymbol{A}^{(n)} \geqslant 0. \end{aligned} \tag{27}$$

### 2.4.2 Inexact Block Coordinate Descent Framework

In addition to the outer iteration in BCD, the subproblem (27) is usually solved by optimization methods in inner iterations. However, in many cases, there is

no need to obtain an exact solution of the subproblem (27). For example, at the beginning of tensor decomposition, all factors are usually initialized by random numbers. In the first outer iteration, obtaining an exact solution of the subproblem (27) is not very meaningful. Therefore, it is necessary to terminate the inner iteration with an inexact solution of the subproblem. This is the key idea of inexact block coordinate descent (inexact BCD). There are some stopping criteria for the subproblem in inexact BCD. (Gillis and Glineur, 2012) proposed the stopping criteria based on the computational complexity. A direct way is to set a maximum inner iteration number (`MAX_ITER_NUM`) for the subproblem.

It has been reported that inexact BCD will accelerate the convergence of the optimization problem (Gillis and Glineur, 2012; Vervliet and Lathauwer, 2019). In this dissertation, we employ inexact BCD framework to solve sparse NCP problem. We set `MAX_ITER_NUM` by 5 to terminate the inner iteration. The inexact BCD scheme is shown in in **Algorithm 1**.

**Algorithm 1:** Inexact BCD Scheme For Sparse NCP

```
Initialization and preparation steps;
repeat
    for n = 1 to N do
        Basic computations before updating A^(n);
        repeat
            Update A^(n) by A^(n) = arg min_{A^(n) ⩾ 0} F(A^(n));
        until inner termination criterion is reached;
    end
until outer termination criterion is reached;
```

### 2.4.3 Optimization methods

A lot of optimization methods have been proposed to solve the subproblem in both NCP and NMF. Lee et al. proposed the multiplicative update (MU) method (Lee and Seung, 1999, 2001), which is the most popular and widely applied method for NMF. Cichocki et al. proposed the hierarchical alternating least squares (HALS) method for large-scale problems (Cichocki et al., 2009; Cichocki and Phan, 2009). Xu and Yin (2013) proposed the alternating proximal gradient (APG) method with detailed mathematical convergence proofs. The similar idea as APG was also proposed for NMF in Guan et al. (2012) and for NCP in Zhang et al. (2016). Recently, the alternating direction method of multipliers (ADMM) has become popular (Boyd, 2011; Huang et al., 2016). In addition, the alternating nonnegative least squares (ANLS) method was deeply analyzed in Lin's seminal paper with strong optimization properties (Lin, 2007), which has a significant influence on NMF. A general form of ANLS is alternating nonnegative quadratic programming, which can be abbreviated to ANQP. When ANLS/ANQP method is utilized, the subproblems appear as the nonnegative least squares (NNLS) prob-

lems. Many efficient methods have been devoted to solve the NNLS subproblems, such as Lin's project gradient method (Lin, 2007), quasi-Newton method (Zdunek and Cichocki, 2006; Kim and Park, 2007), active-set method (Kim and Park, 2008), block principal pivoting method (Kim and Park, 2011) and inertial projection neural network (Dai et al., 2018).

In this dissertation, we investigate five popular optimization methods for solving sparse NCP, including MU, ANQP, HALS, APG and ADMM. All of these methods are carefully tailored to sparse NCP problem.

When sparse regularization is imposed, more full-zero components will appear in the factor matrices. Thus, the factor matrices are not full column rank (rank deficiency). MU, APG and ADMM can cope with the rank deficiency and can be used to solve (27) directly. However, ANQP and HALS may suffer from the rank deficiency and cause the tensor decomposition unstable. As mentioned previously, it has been proved that the NCP with proximal algorithm can improve the stability and is equivalent to original NCP (Li et al., 2013). Therefore, instead of solving the subproblem (27), we proposed the improved subproblem as the following minimization problem:

$$\begin{aligned}\min_{\boldsymbol{A}^{(n)}} \mathcal{F}_{\text{prox}}\left(\boldsymbol{A}^{(n)}\right) &= \frac{1}{2}\left\|\boldsymbol{X}_{(n)} - \boldsymbol{A}^{(n)}\left(\boldsymbol{B}^{(n)}\right)^{T}\right\|_{F}^{2} \\ &+ \frac{\alpha_n}{2}\left\|\boldsymbol{A}^{(n)} - \breve{\boldsymbol{A}}^{(n)}\right\|_{F}^{2} + \beta_n \sum_{r=1}^{R}\left\|\boldsymbol{a}_r^{(n)}\right\|_1 \\ &\text{s.t. } \boldsymbol{A}^{(n)} \geqslant 0,\end{aligned} \tag{28}$$

in which $\widetilde{\boldsymbol{A}}^{(n)}$ is the value of factor $\boldsymbol{A}^{(n)}$ in previous iteration during updating and $\alpha_n$ are positive regularization parameters in vectors $\boldsymbol{\alpha} \in \mathbb{R}^{N\times 1}$.

However, the form of the improved subproblem (28) can't be organized in least square form due to the $l_1$-norm item. Therefore, we reform (28) into the nonnegative quadratic programming (NNQP) form:

$$\begin{aligned}\min_{\boldsymbol{A}^{(n)}} \sum_{i=1}^{I_n} &\left\{\frac{1}{2}\left[\boldsymbol{A}^{(n)}\right]_{(i,:)} \boldsymbol{M}\left[\boldsymbol{A}^{(n)}\right]_{(i,:)}^{T} + \boldsymbol{N}_{(i,:)}\left[\boldsymbol{A}^{(n)}\right]_{(i,:)}^{T}\right. \\ &\left. + \frac{1}{2}\left[\boldsymbol{X}_{(n)}\right]_{(i,:)}\left[\boldsymbol{X}_{(n)}\right]_{(i,:)}^{T}\right\} \\ &\text{s.t. } \boldsymbol{A}^{(n)} \geqslant 0,\end{aligned} \tag{29}$$

where $\left[\boldsymbol{A}^{(n)}\right]_{(i,:)}$ represents the $i$th row of $\boldsymbol{A}^{(n)}$, $\boldsymbol{M} = \left(\boldsymbol{B}^{(n)}\right)^{T}\boldsymbol{B}^{(n)} + \alpha_n \boldsymbol{I}_R$ and $\boldsymbol{N} = \beta_n \boldsymbol{E} - \boldsymbol{X}_{(n)}\boldsymbol{B}^{(n)} - \alpha_n \breve{\boldsymbol{A}}^{(n)}$. The subproblem (29) can be solved by optimization methods, such as active-set (AS) and block principal pivoting (BPP).

We summarize the solution of sparse NCP using MU, ANQP, HALS, APG, ADMM in **Table 2**. More details about the computations of these five optimization methods can be found in our attached original paper **PI**.

TABLE 2 Updating rules of optimization methods for sparse NCP

| Algorithm | Updating Rule |
|---|---|
| MU | $\left[\boldsymbol{A}^{(n)}\right]_{ir} = \left[\boldsymbol{A}^{(n)}\right]_{ir} \frac{\left[\boldsymbol{X}_{(n)}\boldsymbol{B}^{(n)}\right]_{(i,r)} + \epsilon}{\left[\boldsymbol{A}^{(n)}\left(\boldsymbol{B}^{(n)}\right)^T\boldsymbol{B}^{(n)} + \beta_n\boldsymbol{E}\right]_{(i,r)} + \epsilon}.$ |
| ANQP-BPP | $\boldsymbol{X}_{(n)}\boldsymbol{B}^{(n)} \leftarrow \boldsymbol{X}_{(n)}\boldsymbol{B}^{(n)} + \alpha_n\check{\boldsymbol{A}}^{(n)} - \beta_n\boldsymbol{E};$<br>$\left(\boldsymbol{B}^{(n)}\right)^T\boldsymbol{B}^{(n)} \leftarrow \left(\boldsymbol{B}^{(n)}\right)^T\boldsymbol{B}^{(n)} + \alpha_n\boldsymbol{I}_R;$<br>$\boldsymbol{A}^{(n)} = \underset{\boldsymbol{A}^{(n)}\geqslant 0}{\operatorname{argmin}} \mathcal{F}_{\text{prox}}\left(\boldsymbol{A}^{(n)}\right)$<br>$= \texttt{NNQP\_BPP}\left(\boldsymbol{X}_{(n)}\boldsymbol{B}^{(n)}, \left(\boldsymbol{B}^{(n)}\right)^T\boldsymbol{B}^{(n)}\right).$ |
| HALS | $\left[\boldsymbol{A}^{(n)}\right]_{(:,r)} =$<br>$\left[\left[\check{\boldsymbol{A}}^{(n)}\right]_{(:,r)} + \frac{\left[\boldsymbol{X}_{(n)}\boldsymbol{B}^{(n)}\right]_{(:,r)} - \boldsymbol{A}^{(n)}\left[\left(\boldsymbol{B}^{(n)}\right)^T\boldsymbol{B}^{(n)}\right]_{(:,r)} - \beta_n\boldsymbol{1}}{\left[\left(\boldsymbol{B}^{(n)}\right)^T\boldsymbol{B}^{(n)}\right]_{(r,r)} + \alpha_n}\right]_+.$ |
| APG | $\widehat{\boldsymbol{A}}^{(n)}$ is the extrapolated point, $\widehat{\boldsymbol{G}}^{(n)}$ is the gradient, and<br>$L^{(n)}$ is Lipschitz constant.<br>$\boldsymbol{A}^{(n)} = \max\left(0, \widehat{\boldsymbol{A}}^{(n)} - \frac{\widehat{\boldsymbol{G}}^{(n)}}{L^{(n)}} - \frac{\beta_n\boldsymbol{E}}{L^{(n)}}\right).$ |
| ADMM | $\boldsymbol{A}^{(n)}$, $\widetilde{\boldsymbol{A}}^{(n)}$ and $\boldsymbol{\Phi}^{(n)}$ are the primal, auxiliary and dual variable.<br>$\mathcal{L}_\rho\left(\boldsymbol{A}^{(n)}, \widetilde{\boldsymbol{A}}^{(n)}, \boldsymbol{\Lambda}^{(n)}\right)$ is the augmented Lagrangian function.<br>$\boldsymbol{A}^{(n)} = \left[\boldsymbol{X}_{(n)}\boldsymbol{B}^{(n)} + \rho_n\left(\widetilde{\boldsymbol{A}}^{(n)} - \boldsymbol{\Phi}^{(n)}\right)\right]\left[\left(\boldsymbol{B}^{(n)}\right)^T\boldsymbol{B}^{(n)} + \rho_n\boldsymbol{I}\right]^{-1};$<br>$\widetilde{\boldsymbol{A}}^{(n)} = \max\left\{0, \boldsymbol{A}^{(n)} + \boldsymbol{\Phi}^{(n)} - \frac{\beta_n}{\rho_n}\boldsymbol{E}\right\};$<br>$\boldsymbol{\Phi}^{(n)} = \boldsymbol{\Phi}^{(n)} + \boldsymbol{A}^{(n)} - \widetilde{\boldsymbol{A}}^{(n)}.$ |

## 2.5 Experiment on Synthetic Tensor

We introduce one experiment on synthetic tensor, by which we test and evaluate the performances of the above five optimization algorithms.

We constructed a synthetic third-order tensor by 10 channels of simulated sparse and nonnegative signals, as shown in **Figure 5**(a). The signals come from the file `VSparse_rand_10.mat` in NMFLAB (Cichocki and Zdunek, 2006). There are 1000 points in each channel, so the sparse signal matrix is $\boldsymbol{S}^{(1)} = [\boldsymbol{s}_1, \ldots, \boldsymbol{s}_{10}] \in \mathbb{R}^{1000\times 10}$. Two uniformly distributed random matrices $\boldsymbol{A}^{(2)}, \boldsymbol{A}^{(3)} \in \mathbb{R}^{100\times 10}$ were employed as mixing matrices, which were generated by `rand` function in MATLAB. Afterwards, we synthesized a third-order tensor by $\mathcal{X} = [\![\boldsymbol{S}^{(1)}, \boldsymbol{A}^{(2)}, \boldsymbol{A}^{(3)}]\!] \in \mathbb{R}^{1000\times 100\times 100}$. Next, nonnegative

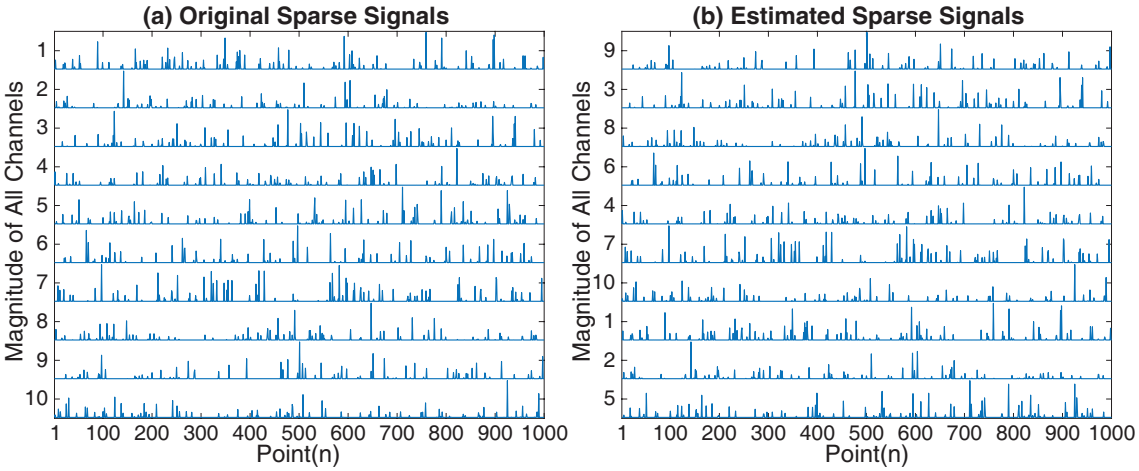



FIGURE 5 Sparse and nonnegative signals used in synthetic tensor. (a) shows the original ten channels of signals. (b) shows the estimated ten channels of signals from the synthetic tensor by sparse NCP based on ANQP-BPP method with $\beta_n = 3$.

noise was added to the tensor with SNR of 40dB, which was generated by MATLAB code `max(0,randn(size(`$\mathcal{X}$`)))`.

We selected a larger value of 20 as the number of components for tensor decomposition. Since 10 channels of signals are mixed in the tensor, naturally, 10 should be selected as the optimal component number. The number of components might also be estimated by some classical methods, such as DIFFIT (Timmerman and Kiers, 2000). However, we intend to recover the 10 channels of true signal just by imposing sparse regularization during decomposition, even though the exact optimal number of components is unknown. We selected values of $\beta_n = 0, 0.1, 0.5, 1, 2, 3$ for all the optimization methods to evaluate their abilities to impose sparsity. The selection of sparse regularization parameters depends on the tensor data. we calculated and evaluated the sparsity level (Wang et al., 2018a) of the factor matrices by

$$\text{Sparsity}_{\boldsymbol{A}^{(n)}} = \frac{\#\left\{\boldsymbol{A}_{i,r}^{(n)} < T_s\right\}}{I_n \times R}, \tag{30}$$

where $T_s$ is a small positive number and $\#\{\cdot\}$ denotes the number of elements that are smaller than the threshold $T_s$ in factor matrix $\boldsymbol{A}^{(n)}$. For this synthetic data, we select $T_s = 1e-3$.

We use the Peak Signal-to-Noise-Ratio (PSNR) (Cichocki et al., 2009) to evaluate the accuracy of the extracted sparse factor matrix compared with original sparse signals. In this experiment, the value of about 85 means an accurate recovery.

After tensor decomposition, the values of objective function value (Obj), relative error (RelErr), running time in second, iteration number (Iter), the number of nonzero components (NNC), sparsity level (Spars) and PSNR of the estimated signal factor matrix were recorded as the performance evaluation criteria. For all optimization methods with each $\beta_n$, the sparse NCP was run 30 times, and the

TABLE 3 Comparison of sparse NCPs on third-order synthetic tensor

| Method | $\beta_n$ | Obj | RelErr | Time | Iter | NNC | Spars | PSNR |
|---|---|---|---|---|---|---|---|---|
| MU | 0 | 9.76e+01 | 0.0082 | 107.9 | 2720.2 | 20.00 | 0.340 | 73.63 |
| | 0.1 | 5.33e+02 | 0.0083 | 78.6 | 1970.0 | 16.17 | 0.431 | 73.57 |
| | 0.5 | 1.65e+03 | 0.0084 | 57.5 | 1449.9 | 12.90 | 0.550 | 79.02 |
| | 1 | 2.47e+03 | 0.0084 | 74.6 | 1889.9 | 11.17 | 0.694 | 81.44 |
| | 2 | 4.17e+03 | 0.0084 | 86.0 | 2181.5 | 10.37 | 0.863 | 82.50 |
| | 3 | 5.91e+03 | 0.0085 | 87.3 | 2212.6 | **10.10** | **0.905** | **83.69** |
| ANQP BPP | 0 | 9.72e+01 | 0.0082 | 8.1 | 170.9 | 20.00 | 0.400 | 69.74 |
| | 0.1 | 5.45e+02 | 0.0083 | 8.9 | 222.8 | 10.20 | 0.548 | 83.21 |
| | 0.5 | 1.37e+03 | 0.0083 | 85.5 | 2178.0 | 10.10 | 0.618 | 84.26 |
| | 1 | 2.16e+03 | 0.0083 | 110.6 | 2811.1 | 10.00 | 0.823 | 84.49 |
| | 2 | 4.00e+03 | 0.0084 | 96.1 | 2435.5 | **10.00** | **0.908** | **85.13** |
| | 3 | 5.87e+03 | 0.0085 | 83.5 | 2116.2 | **10.00** | **0.912** | **85.39** |
| HALS | 0 | 9.74e+01 | 0.0082 | 18.8 | 458.8 | 20.00 | 0.452 | 79.12 |
| | 0.1 | 6.43e+02 | 0.0083 | 11.7 | 287.0 | 15.70 | 0.504 | 79.88 |
| | 0.5 | 1.85e+03 | 0.0084 | 38.2 | 943.4 | 11.70 | 0.634 | 83.27 |
| | 1 | 2.28e+03 | 0.0084 | 101.1 | 2516.1 | 10.20 | 0.819 | 84.38 |
| | 2 | 4.00e+03 | 0.0084 | 95.9 | 2395.9 | **10.00** | **0.913** | **85.10** |
| | 3 | 5.89e+03 | 0.0085 | 80.0 | 2001.8 | **10.00** | **0.917** | **85.42** |
| APG | 0 | 9.73e+01 | 0.0082 | 18.0 | 446.3 | 20.00 | 0.378 | 71.06 |
| | 0.1 | 5.52e+02 | 0.0083 | 11.2 | 278.0 | 11.87 | 0.519 | 79.58 |
| | 0.5 | 1.39e+03 | 0.0083 | 83.1 | 2083.0 | 10.03 | 0.621 | 84.13 |
| | 1 | 2.22e+03 | 0.0083 | 94.1 | 2353.1 | 10.00 | 0.823 | 84.27 |
| | 2 | 4.07e+03 | 0.0084 | 82.0 | 2056.3 | **10.00** | **0.908** | **84.89** |
| | 3 | 5.95e+03 | 0.0084 | 66.1 | 1663.3 | **10.00** | **0.906** | **84.83** |
| ADMM | 0 | 9.76e+01 | 0.0082 | 88.8 | 2219.7 | 20.00 | 0.358 | 73.85 |
| | 0.1 | 6.05e+02 | 0.0083 | 20.4 | 511.3 | 18.60 | 0.505 | 66.19 |
| | 0.5 | 1.75e+03 | 0.0084 | 50.9 | 1279.1 | 11.57 | 0.623 | 80.24 |
| | 1 | 2.45e+03 | 0.0084 | 82.7 | 2085.3 | 10.33 | 0.786 | 83.99 |
| | 2 | 4.02e+03 | 0.0084 | 93.9 | 2378.7 | **10.00** | **0.913** | **85.22** |
| | 3 | 5.89e+03 | 0.0085 | 80.0 | 2030.2 | **10.00** | **0.916** | **85.50** |

Spars = Sparsity level; Ground truth value: Spars=0.9.
NNC = Number of nonzero components.

average values of all criteria were computed. The results are shown in **Table 3**.

From **Table 3**, it can be found that all methods can impose sparsity with proper sparse regularization parameter $\beta_n$. When $\beta_n$ increases, the sparsity level of the mode-1 factor matrix is also increased. With certain sparse regularization, 10 nonzero components are retained in the mode-1 factor matrix, which represent the 10 channels of sparse signals extracted from the synthetic tensor. One of the recovered sparse signal matrix by ANQP-BPP is shown in **Figure 5**(b), in which the PNSR is 85.5251.

The results demonstrate that all the MU, ANQP, HALS, APG and ADMM methods are effective to impose sparsity on factor matrix and recovery the sparse components. The methods of ANQP, HALS and APG especially have superior computational efficiency. More experiments and results about sparse NCP using different optimization methods can be found in our original paper **PI**.

# 3 EXTRACTING MEANINGFUL FEATURES FROM EEG TENSOR

In this Chapter, We present the application of constrained and regularized tensor decomposition, especially NCP and sparse NCP, to real-world EEG tensors. These EEG tensors are collected in real cognitive neuroscience experiments with specific external stimuli. Using NCP or sparse NCP, the EEG components that are related to cognitive processes are extracted. These EEG feature components are good evidences to deeply investigate brain function.

## 3.1 Event-Related Potential Tensor Decomposition

In this experiment, we decompose a fifth-order ERP tensor using NCP. The data come from an open preprocessed dataset associated with ERPWAVELAB toolbox (Mørup et al., 2007), which can be downloaded from www.erpwavelab.org. The data were collected from a proprioceptive experiment, in which two conditions (left and right hand) were manipulated with the increment of handhold load. An important part of the stimuli is the change of applied force on a static muscle contraction, which is conceived as proprioceptive stimulus (Mørup et al., 2006). Fourteen subjects participated in the experiment and 64 scalp electrodes were used to record EEG data. A total of 360 trials (epochs) were obtained from each subject under each condition. All epochs were transformed into time-frequency representation (TFR) by complex Morlet wavelet. In the wavelet transform, only the frequency band from 15 Hz to 75 Hz were analyzed with linear interval of 1Hz. Then, inter-trial phase coherence (ITPC) (Delorme and Makeig, 2004) was computed as an average spectral estimate across all trials. Since the TFR was first applied to each trial and then the average was calculated across trials, ITPC can be seen as induced oscillations of brain (David et al., 2006). Meanwhile, ITPC only takes values between 0 and 1 (Delorme and Makeig, 2004; Cohen, 2014). Finally, a fifth-order nonnegative tensor (channel $\times$ frequency $\times$ time $\times$ subject $\times$ condition = $64 \times 61 \times 72 \times 14 \times 2$) was generated. The 61 frequency points

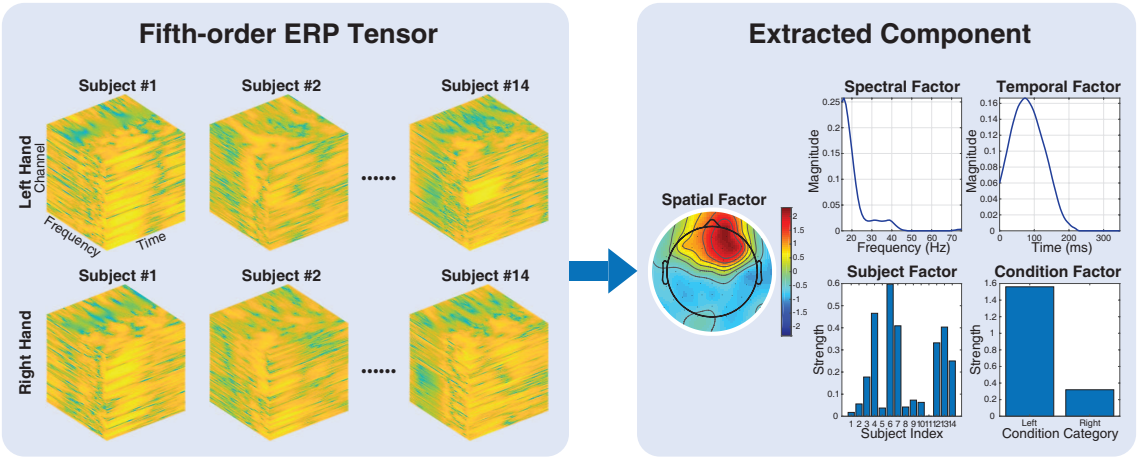



FIGURE 6 Illustration of the fifth-order ERP tensor decomposition.

represent $15 - 75$ Hz, and the 72 time points represent $0 - 346.68$ ms.

The detailed EEG data collection and preprocessing procedures are described in Mørup et al. (2006).

The meaning of the fifth-order ERP tensor decomposition is illustrated in **Figure 6**. Since the tensor has five modes including space, frequency, time, subject and condition, each group of the extracted components consist of spatial, spectral, temporal, subject and condition factor. Thus, the interactive information among the five modes is preserved. For example, the relation between experimental condition and the brain activity location on the scalp. Based on the interactive information, the assumption of the fifth-order tensor decomposition is that the underlying spatial, spectral, and temporal factors are the same among all subjects only with differences in subject-dependent and condition-dependent strength.

The NCP algorithm in this experiment is based on APG method. The initial number of components was set by 20, so 20 components were extracted. In the results, we discovered many pairs of components with symmetric responses on topographies. Two pairs of symmetric components are presented in **Figure 7**. It is clear to see that left-hand stimuli elicit activities in the right hemisphere, and right-hand stimuli elicit activities in the left hemisphere.

More experimental details and results can be found in the attached original paper **PIII**.

## 3.2 Ongoing EEG Tensor Decomposition

Cognitive neuroscience has traditionally relied upon relatively simple parametric tasks using controlled and repeated stimuli. However, these traditional methods do not resemble the complexity and dynamics of stimuli and behaviours in real-life. Studying neural processing and network interactions using naturalistic and dynamic stimuli, such as movie, speech, music and game, may facilitate a deeper understanding of adaptive human brain function (Sonkusare et al., 2019).

Music is associated with many aspects of people's personal and social lives.

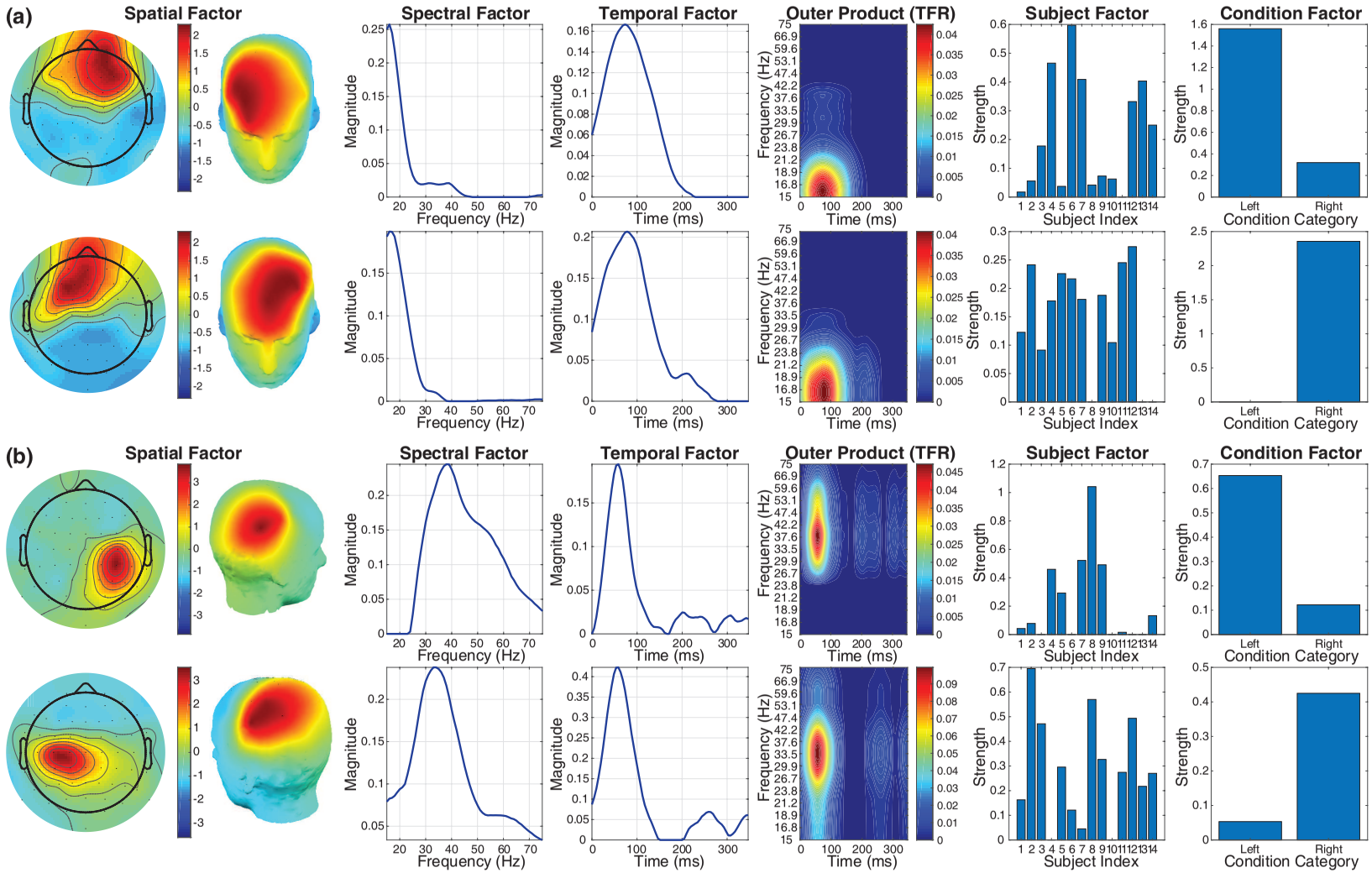



FIGURE 7 Components from a fifth-order ERP tensor using NCP. (a) shows one pair of symmetric components in Beta-band, and (b) shows another pair of components in Gamma-band. Obviously, these extracted brain activities are elicited by contralateral hand stimuli.

Uncovering the neural underpinnings of music processing has become a central theme in cognitive neuroscience in the past decades (Alluri et al., 2012). Some researches have studied music and brain function using functional Magnetic Resonance Imaging (fMRI) (Alluri et al., 2012) and ongoing EEG (Cong et al., 2013).

In this experiment, a set of ongoing EEG tensors are analyzed, which were collected by naturalistic and continuous music stimuli of modern tango. As introduced in Chapter 1, the spectra of the ongoing EEG sometimes are very sparse, so we decompose the data using sparse NCP. The experiment is briefly introduced below.

### 3.2.1 Ongoing EEG Data and Music Features

The ongoing EEG were collected from fourteen right-handed and healthy adults elicited by naturalistic and continuous modern tango music stimulus. Short-time Fourier transform (STFT) was applied to the EEG data, and a third-order tensor was created for each subject with size of space $\times$ frequency $\times$ time $= 64 \times 146 \times 510$. In this study, we analyze the third-order EEG tensor of each subject one by one.

The modern tango music lasts for 8.5 minutes. Five long-term acoustic features, including two tonal features (Mode, Key Clarity) and three rhythmic features (Pulse Clarity, Fluctuation Centroid, Fluctuation Entro-py), were extracted from the music signals. STFT was also used for feature extraction, thus one acoustic feature temporal series contains 510 samples. Detailed acoustic features can be found in Alluri et al. (2012).

### 3.2.2 Correlation Analysis

After tensor decomposition using sparse NCP, EEG features will be extracted including spatial, spectral, and temporal components. Next, we try finding those features that are assumed to be elicited by music features.

We performed correlation analyses by Pearson's correlation coefficient between the time series of long-term acoustic features and the time series of temporal components from EEG tensor decomposition to find stimulus-related components. Monte Carlo method and permutation tests were employed to compute the threshold of correlation coefficient (Alluri et al., 2012; Cong et al., 2013). In the results of EEG tensor decomposition, the temporal components significantly correlated (at level $p<0.05$) with any of the five acoustic features, and their corresponding spectral and spatial components were recorded for further investigations.

### 3.2.3 Stability Analysis

By correlation analysis, we obtain the meaningful components that are assumed to be stimulus-related features, in which some components are sparse. Since the sparse regularization is imposed to the tensor decomposition, more sparse components will be extracted. It is very interesting to evaluate the

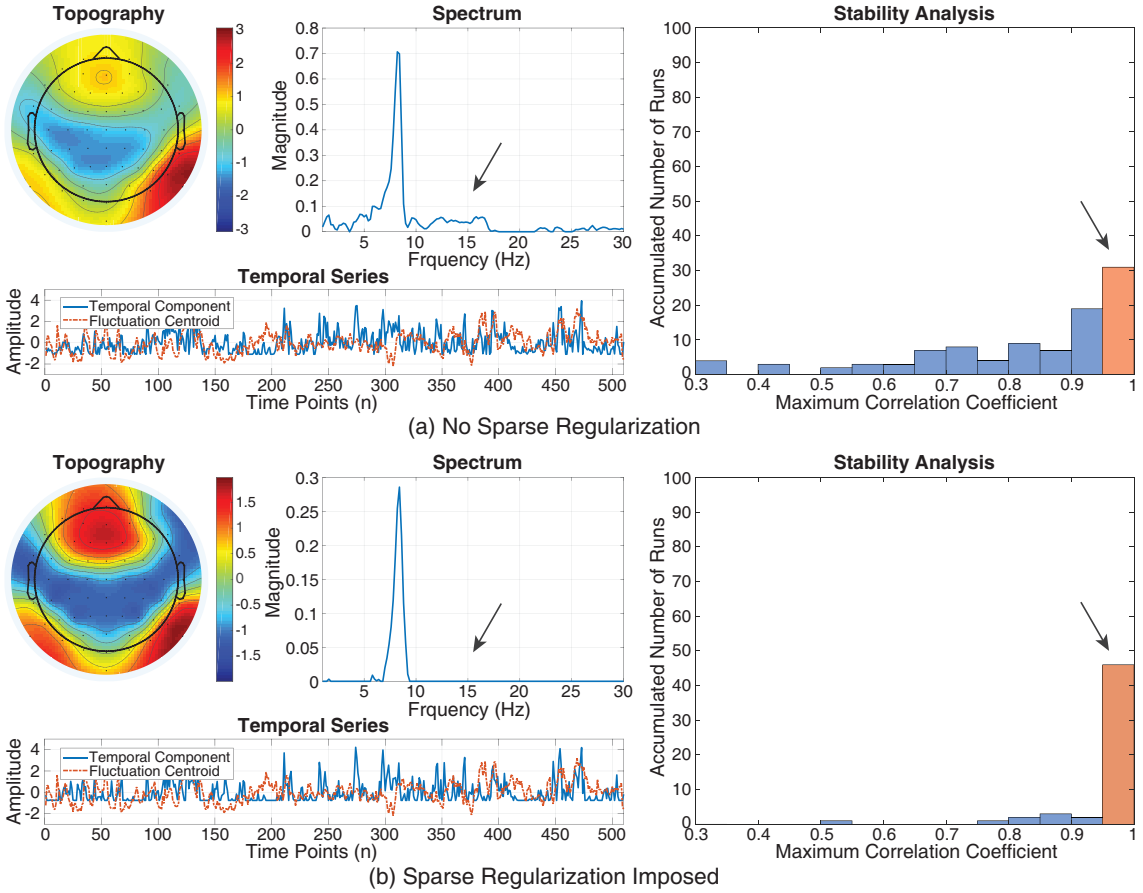



FIGURE 8 Stability analysis of ongoing EEG components. This is a group of EEG components whose temporal series is highly correlated with fluctuation centroid music feature. (a) show the components and their stability without sparse regularization in NCP. (b) shows the results with sparse regularization imposed.

stability of the stimulus-related components.

First, one group components is selected as a template, whose temporal component is highly correlated with one music features. This template can be represented by a rank-1 tensor of their outer product:

$$\mathcal{T} = \boldsymbol{t}^{(\text{Spatial})} \circ \boldsymbol{t}^{(\text{Spectral})} \circ \boldsymbol{t}^{(\text{Temporal})} \tag{31}$$

Second, we run the tensor decomposition for $M = 100$ times. In results of the $m$th decomposition, the correlation coefficient of the $r$th rank-1 tensor and the template $\mathcal{T}$ is computed by

$$\begin{aligned}\rho(r,m) =& \text{corr}(\boldsymbol{a}_r^{(\text{Spatial})}, \boldsymbol{t}^{(\text{Spatial})}) \times \text{corr}(\boldsymbol{a}_r^{(\text{Spectral})}, \boldsymbol{t}^{(\text{Spectral})}) \\ & \times \text{corr}(\boldsymbol{a}_r^{(\text{Temporal})}, \boldsymbol{t}^{(\text{Temporal})})\end{aligned} \tag{32}$$

in which, $\text{corr}(\cdot,\cdot)$ is the calculation of Pearson's correlation coefficient, $\boldsymbol{a}$ is an extracted feature component, $r = 1,\ldots,R$, and $m = 1,\ldots,M$.

Finally, we count the percentage of $\rho(r,m) \geqslant 0.95$ in the $M$ runs, which is record as the stability of the components that are highly related to a music feature.

**Figure 8** presents a group of components whose temporal series is highly correlated with fluctuation centroid music feature. It is clear to see that when sparse regularization is imposed in NCP, some redundant information in the spectrum is removed and the stability of this group of components are increased.

After analysis of the fourteen tensors from fourteen subjects, many groups of components that are significantly correlated with a music feature have increased stability with sparse regularization. More details and results can be found in the attached paper **PIV**.

This experiment on ongoing EEG tensor demonstrated that tensor decomposition with proper constraint and regularization is able to stably extract meaningful EEG components.

## 3.3 N-way Partial Least Squares for Ongoing EEG

In section 3.2.1, we have introduced the ongoing EEG tensor data and music features. The third-order ongoing EEG tensor can be represented by $\mathcal{X} \in \mathbb{R}^{I \times K \times M}$ with $I \times K \times M = 510 \times 146 \times 64$, and the matrix of music features can be represented by $\boldsymbol{Y} \in \mathbb{R}^{I \times J}$ with $I \times J = 510 \times 5$. If a tensor $\mathcal{X}$ and a matrix $\boldsymbol{Y}$ share at least one mode in common, they can be analyzed simultaneously by N-way partial least squares (N-PLS). Using N-PLS, tensor $\mathcal{X}$ and matrix $\boldsymbol{Y}$ can be decomposed by

$$\begin{aligned} \mathcal{X} &= \sum_{r=1}^{R} \boldsymbol{t}_r \circ \boldsymbol{p}_r \circ \boldsymbol{q}_r + \mathcal{E}_{\mathcal{X}} \\ \boldsymbol{Y} &= \sum_{r=1}^{R} d_{rr} \boldsymbol{t}_r \boldsymbol{c}_r^T + \boldsymbol{E}_{\boldsymbol{Y}} \end{aligned} \tag{33}$$

where $\boldsymbol{t} \in \mathbb{R}^{I \times 1}$ represents the temporal components that is shared by $\mathcal{X}$ and $\boldsymbol{Y}$, $\boldsymbol{p}_r \in \mathbb{R}^{K \times 1}$ represents the spectral component, $\boldsymbol{q}_r \in \mathbb{R}^{M \times 1}$ represents the spatial components, and $d_{rr} = \frac{\boldsymbol{u}_r^T \boldsymbol{t}_r}{\boldsymbol{t}_r^T \boldsymbol{t}_r}$ with $\boldsymbol{u}_r = \boldsymbol{Y} \boldsymbol{c}_r$.

The meaning of analyzing ongoing EEG tensor $\mathcal{X}$ and music feature matrix $\boldsymbol{Y}$ by N-PLS is that common temporal components shared by both the EEG data and music features can be extracted. After decomposition using N-PLS, many temporal components will be extracted from the tensor of each subject. Correlation analysis is carried out as introduced in section 3.2.2, then those components significantly correlated with music features are selected.

We present several components extracted by N-PLS from all subjects in **Figure 9**, in which all the temporal series are significantly related with fluctuation centroid music feature. Their corresponding spectral and spatial components are also presented. More information about N-PLS for ongoing EEG analysis can be found in the attached original paper **PV**.

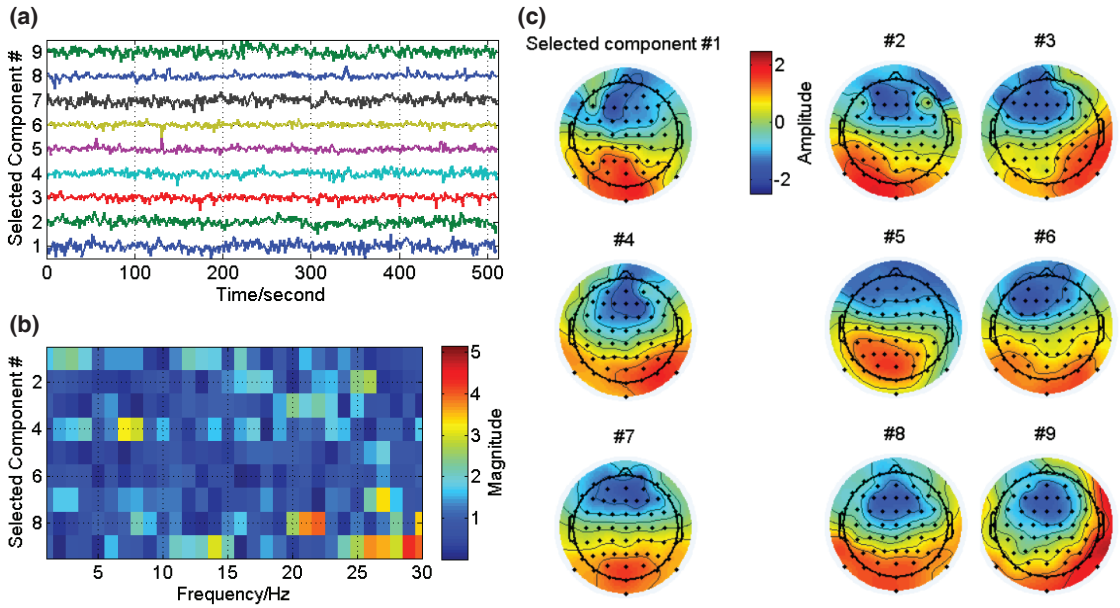



FIGURE 9 Components extracted by N-PLS from ongoing EEG tensor. (a) shows the temporal components that are significantly correlated with fluctuation centroid feature, (b) and (c) are the corresponding spectral and spatial components.

# 4 RESEARCH CONTRIBUTION

The main research works in this dissertation are published in the attached five papers. The titles, abstracts and author contributions of these papers are introduced in this chapter.

## 4.1 Paper PI

**Deqing Wang**, Xiulin Wang, Tapani Ristaniemi and Fengyu Cong. Sparse Nonnegative Tensor Decomposition in Inexact Block Coordinate Descent Framework. *Submitted to a journal*.

**Abstract:** Nonnegative tensor decomposition is a versatile tool for multiway data analysis, by which the extracted components are nonnegative and usually sparse. However, the sparsity is only a side effect, which cannot be controlled without additional regularization. In this paper, we investigate the nonnegative CANDECOMP/PARAFAC (NCP) decomposition with sparse regularization item using $l_1$-norm (sparse NCP). We design sparse NCP using an inexact block coordinate descent scheme, which is able to accelerate the convergence. Five optimization methods in block coordinate descent framework are employed to solve the sparse NCP, including multiplicative update, alternating nonnegative quadratic programming, hierarchical altering least squares, alternating proximal gradient and alternating direction method of multipliers, all of which are carefully tailored to the sparse regularization problem. We evaluate all methods by experiments on both synthetic and real-world tensor data. The experimental results demonstrate that our proposed sparse NCP methods are able to efficiently impose sparsity to factor matrices and extract meaningful sparse components.

**Contributions:**

Deqing Wang proposed all of the algorithms, conducted all experiments and wrote most of the manuscript.

Xiulin Wang analyzed the computational complexity of the algorithms and

revised the manuscript.

Tapani Ristaniemi and Fengyu Cong supervised the whole research work and revised the manuscript.

## 4.2 Paper PII

**Deqing Wang**, Fengyu Cong and Tapani Ristaniemi. Higher-Order Nonnegative CANDECOMP/PARAFAC Tensor Decomposition Using Proximal Algorithm. *2019 IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP), Brighton, UK*, 2019. (**Best Student Paper Award**)

**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.

**Contributions:**

Deqing Wang proposed the algorithm, conducted all experiments and wrote the whole manuscript.

Tapani Ristaniemi and Fengyu Cong supervised the whole research work and revised the manuscript.

## 4.3 Paper PIII

**Deqing Wang**, Yongjie Zhu, Tapani Ristaniemi and Fengyu Cong. Extracting multi-mode ERP features using fifth-order nonnegative tensor decomposition. *Journal of Neuroscience Methods, Volume 308, p.240-247*, 2018.

**Abstract:**

*Background:* Preprocessed Event-related potential (ERP) data are usually organized in multi-way tensor, in which tensor decomposition serves as a powerful tool for data processing. Due to the limitation of computation burden for multi-way data and the low algorithm performance of stability and efficiency, multi-way ERP data are conventionally reorganized into low-order tensor or matrix before further analysis. However, the reorganization may hamper mode specification and spoil the interaction information among different modes.

*New Method:* In this study, we applied a fifth-order tensor decomposition to a set of fifth-order ERP data collected by exerting proprioceptive stimulus on left and right hand. One of the latest nonnegative CANDECOMP/PARAFAC (NCP) decomposition methods implemented by alternating proximal gradient (APG) was employed. We also proposed an improved DIFFIT method to select the optimal component number for the fifth-order tensor decomposition.
*Results:* By the fifth-order NCP model with a proper component number, the ERP data were fully decomposed into spatial, spectral, temporal, subject and condition factors in each component. The results showed more pairs of components with symmetric activation region in left and right hemisphere elicited by contralateral stimuli on hand.
*Comparison with Existing Method(s):* In our experiment, more interesting components and coherent brain activities were extracted, compared with previous studies.
*Conclusions:* The discovered activities elicited by proprioceptive stimulus are in line with those in relevant cognitive neuroscience studies. Our proposed method has proved to be appropriate and viable for processing high-order EEG data with well-preserved interaction information among all modes.

**Contributions:**

Deqing Wang proposed the main idea, conducted all experiments and wrote most of the manuscript.

Yongjie Zhu explained the results from the perspective of cognitive neuroscience and revised the manuscript.

Tapani Ristaniemi and Fengyu Cong supervised the whole research work and revised the manuscript.

## 4.4 Paper PIV

**Deqing Wang**, Xiaoyu Wang, Yongjie Zhu, Petri Toiviainen, Minna Huotilainen, Tapani Ristaniemi and Fengyu Cong. Increasing Stability of EEG Components Extraction Using Sparsity Regularized Tensor Decomposition. Advances in Neural Networks - the 15th International Symposium on Neural Networks (ISNN 2018), Minsk, Belarus, 2018.

**Abstract:** Tensor decomposition has been widely employed for EEG signal processing in recent years. Constrained and regularized tensor decomposition often attains more meaningful and interpretable results. In this study, we applied sparse nonnegative CANDECOMP/PARAFAC tensor decomposition to ongoing EEG data under naturalistic music stimulus. Interesting temporal, spectral and spatial components highly related with music features were extracted. We explored the ongoing EEG decomposition results and properties in a wide range of sparsity levels, and proposed a paradigm to select reasonable sparsity regularization parameters. The stability of interesting components extraction from fourteen sub-

jects' data was deeply analyzed. Our results demonstrate that appropriate sparsity regularization can increase the stability of interesting components significantly and remove weak components at the same time.

**Contributions:**

Deqing Wang proposed the main idea, conducted all experiments and wrote the whole manuscript.

Xiaoyu Wang, Yongjie Zhu, Petri Toiviainen, Minna Huotilainen advised on EEG data analysis and music feature extraction.

Tapani Ristaniemi and Fengyu Cong supervised the whole research work and revised the manuscript.

## 4.5 Paper PV

**Deqing Wang**, Fengyu Cong, Qibin Zhao, Petri Toiviainen, Asoke K. Nandi, Minna Huotilainen, Tapani Ristaniemi and Andrzej Cichocki. Exploiting ongoing EEG with multilinear partial least squares during free-listening to music. 2016 IEEE International Workshop on Machine Learning for Signal Processing (MLSP), Salerno, Italy, 2016.

**Abstract:** During real-world experiences, determining the stimulus-relevant brain activity is excitingly attractive and is very challenging, particularly in electroencephalography. Here, spectrograms of ongoing electroencephalogram (EEG) of one participant constructed a third-order tensor with three factors of time, frequency and space; and the stimulus data consisting of acoustical features derived from the naturalistic and continuous music formulated a matrix with two factors of time and the number of features. Thus, the multilinear partial least squares (PLS) conforming to the canonical polyadic (CP) model was performed on the tensor and the matrix for decomposing the ongoing EEG. Consequently, we found that brain activity of majority of participants was significantly correlated with the musical features in time domain, and that such brain activity showed frontal or central or posterior or occipital distributions along the scalp, and that such brain activity could be of different oscillation bands in frequency domain.

**Contributions:**

Deqing Wang implemented music feature extraction and conducted other experiments with Fengyu Cong. Deqing Wang wrote the first draft and finished the writing with Fengyu Cong.

Fengyu Cong proposed the main idea.

Qibin Zhao, Petri Toiviainen, Asoke K. Nandi, Minna Huotilainen, Tapani Ristaniemi and Andrzej Cichocki advised on the experimental methods and revised the manuscript.

# 5 CONCLUSION

In this dissertation, we investigate constrained and regularized tensor decomposition algorithms and their applications on high-order EEG data analyses that are related to specific cognitive neuroscience experiments.

## 5.1 Tensor Decomposition Algorithms

In this research, the design of tensor decomposition algorithm is based on the properties of EEG tensor data. Because of the popularity of EEG technique in cognitive neuroscience and the fast development of EEG equipment, EEG data are increasing not only in amount but also in dimension. The high-dimension leads EEG data to be naturally represented in tensor form. After some preprocessing procedures, such as time-frequency representation, the EEG tensor becomes nonnegative. What's more, some intrinsic components are also naturally nonnegative and even sparse, such as the spectrum, temporal envelope and the subject strength.

According to the nonnegative and sparse properties, we constructed the nonnegative CANDECOMP/PARAFAC tensor decomposition algorithm with $l_1$-norm based sparse regularization (sparse NCP). The $l_1$-norm item has been reported to be able to impose strong sparsity, which is suitable to extract the intrinsic sparse components in EEG data.

The sparse NCP is a non-convex optimization problem, which can be efficiently solved by block coordinate descent (BCD) method. In BCD framework, the solution of tensor decomposition is achieved by solving subproblems alternatively. Afterwards, the subproblem is solved by an optimization algorithm. Conventional BCD framework might not consider whether the solution of subproblem is an exact one or not. If the subproblem is iterated for limited times with an inexact solution, the computation of the whole tensor decomposition will be increased significantly. This is the key idea of inexact block coordinate descent scheme, which is a key approach in this dissertation.

In the inexact BCD framework, five optimization methods were employed

to solve the sparse NCP, including multiplicative update (MU), alternating nonnegative least squares / quadratic programming ( ANLS/ANQP ), hierarchical altering least squares (HALS), alternating proximal gradient (APG) and alternating direction method of multipliers (ADMM). In addition, we also utilized proximal algorithm particularly for ANLS/ANQP and HALS in order to improve stability.

Experimental results from synthetic and real-world tensor data demonstrate that all our sparse NCP methods are effective and efficient to extract nonnegative and sparse components. The methods of ANLS/ANQP, HALS and APG especially have outstanding performances.

## 5.2 EEG Tensor Decomposition

We applied the constrained and regularized tensor decomposition to a fifth-order ERP dataset and an ongoing EEG dataset. The ERP data were collected by proprioceptive stimuli on human hands. The results of ERP tensor decomposition show that many groups of EEG features in five modes, including space, frequency, time, subject and condition, can be successfully extracted, which are highly related to the stimuli on hands. The ongoing EEG data were collected by naturalistic and continuous music stimulus. Using our method, the nonnegative and sparse EEG components can be extracted, which are highly correlated with music features. Surprisingly, when sparse regularization is imposed, the stability of the music-feature-related components are increased. In addition, the two modalities of ongoing EEG tensor and music signals can be decomposed simultaneously by N-way partial least square (N-PLS), which also yielded meaningful components.

The above analyses on real-world EEG tensors prove that our methods are able to efficiently process high-order EEG data and extract meaningful features that are linked to cognitive processes.

## 5.3 Limitations and Future Directions

The stability of sparse NCP on ongoing EEG is based on selected templates, which is a bit heuristic. More sophisticated stability analysis methods for tensor decomposition are desired.

In future, some latest optimization methods should be considered for the constrained tensor decomposition. Besides the $l_1$-norm based sparse regularization, other regularization item can be incorporated into the tensor decomposition. For example, the combination of graph regularization and tensor decomposition might be useful to brain networks analysis.

# YHTEENVETO (SUMMARY IN FINNISH)

Tässä väitöskirjassa tarkasteltiin rajoitettuja ja säännönmukaistettuja algoritmejä tensoreiden hajottamiseen tekijöihinsä, sekä niiden sovellutuksia korkean asteen EEG-datan analyyseissä. Analyysit liittyivät kognitiivisen neurotieteen kokeisiin.

Tässä tutkimuksessa suunniteltu algoritmi perustuu EEG-tensoridatan ominaisuuksiin. Esiprosessointiproseduurien, kuten aika-frekvenssiesityksen, suorittamisen jälkeen EEG-tensorista tulee ei-negatiivinen. Lisäksi jotkin luontaiset komponentit ovat lähtökohtaisesti ei-negatiivisia ja jopa harvoja esimerkiksi spektriltään. Tässä tutkimuksessa toteutettiin ei-negatiivinen CANDECOMP/PARAFAC-algoritmi tensoreiden hajottamiseen käyttäen harvaa säännönmukaistamista (sparse NCP), joka perustuu $l_1$-normiin. $l_1$-normin on tutkittu olevan sopiva määräämään vahvaa harvuutta (strong sparsity), joten sitä voidaan käyttää harvojen komponenttien irrottamiseen EEG-datasta.

Harva NCP on epäkonveksi optimointiongelma, joka voidaan ratkaista tehokkaasti käyttäen BCD-viitekehystä (block coordinate descent). BCD-viitekehyksen mukaan tensorin hajottaminen tekijöihinsä saavutetaan ratkaisemalla osaongelmia eri järjestyksissä. Tämän jälkeen osaongelma ratkaistaan optimointialgoritmilla. Tavanomainen BCD-viitekehys saattaa jättää huomiotta ratkaisun eksaktiuden. Jos osaongelmaa iteroidaan tietyn määrän epäeksaktilla tuloksella, koko tensorin hajottamiseen käytettävä laskentateho kasvaa huomattavasti. Tämä on erityisen tärkeä huomio liittyen BCD-skeemaan, joka on tämän väitöskirjan keskiössä. Epäeksaktin BCD-viitekehyksen mukaan tässä tutkimuksessa käytettiin viittä optimointimenetelmää harvan säännönmukaistamisen ratkaisemiseen: MU, ANLS/ANQP, HALS, APG ja ADMM. Stabiliteetin parantamiseksi hyödynnettiin ANLS/ANQP- ja HALS-menetelmissä myös nk. lähialgoritmia.

Kokeellisen tutkimuksen tulokset sekä keinotekoisella että todenmukaisella tensoridatalla osoittavat, että kaikki tässä tutkimuksessa esitellyt harvan säännönmukaistamisen menetelmät ovat tehokkaita ei-negatiivisten ja harvojen piirteiden irrottamiseen. Erityisesti ANLS/ANQP-, HALS- ja APG-menetelmien tulokset ovat erinomaisia.

Lisäksi tässä tutkimuksessa käytettiin rajoitettua ja säännönmukaistettua tensoreiden hajottamismenetelmää viidennen asteen ERP-dataan sekä meneillään olevan EEG-mittauksesta saadun datan reaaliaikaiseen analysointiin. ERP-dataan liittyvät tulokset osoittavat, että oleellisia piirteitä voidaan irrottaa EEG-datasta, joka liittyy ihmiskäden asentoaistiärsykkeisiin. Meneillään olevaan EEG-mittaukseen liittyvä data kerättiin kokeen aikana, jossa tutkittiin luonnolliseen ja jatkuvaan musiikinkuunteluun liittyviä aistiärsykkeitä. Tässä tutkimuksessa esiteltyjä menetelmiä käyttäen datasta onnistuttiin irrottamaan ei-negatiiviset ja harvat EEG-komponentit, jotka liittyivät musiikin kuunteluun. Kun dataan käytettiin harvaa säännönmukaistamista, musiikinkuunteluun liittyvien komponenttien stabiliteetti koheni. Lisäksi EEG-tensorin ja musiikkisignaalien modaliteettien hajottaminen samanaikaisesti onnistui N-PLS-menetelmällä, jonka avulla

irrotettiin oleellisia piirteitä tensoreista.

Yllä kuvatut analyysit, joissa käytettiin todenmukaisia EEG-tensoreita, osoittavat että tässä tutkimuksessa kuvatut menetelmät pystyvät tehokkaasti käsittelemään korkean asteen EEG-dataa ja irrottamaan siitä kognitiivisiin prosesseihin liittyviä oleellisia piirteitä.

## REFERENCES

Acar, E. & Yener, B. 2009. Unsupervised multiway data analysis: A literature survey. IEEE Transactions on Knowledge and Data Engineering 21 (1), 6–20. doi:10.1109/tkde.2008.112.

Alluri, V., Toiviainen, P., Jääskeläinen, I. P., Glerean, E., Sams, M. & Brattico, E. 2012. Large-scale brain networks emerge from dynamic processing of musical timbre, key and rhythm. NeuroImage 59 (4), 3677–3689. doi:10.1016/j.neuroimage.2011.11.019.

Artoni, F., Delorme, A. & Makeig, S. 2018. Applying dimension reduction to EEG data by principal component analysis reduces the quality of its subsequent independent component decomposition. NeuroImage 175, 176–187. doi:10.1016/j.neuroimage.2018.03.016.

Artoni, F., Delorme, A. & Makeig, S. 2019. A visual working memory dataset collection with bootstrap independent component analysis for comparison of electroencephalographic preprocessing pipelines. Data in Brief 22, 787–793. doi:10.1016/j.dib.2018.12.022.

Bach, F., Jenatton, R., Mairal, J. & Obozinski, G. 2012. Optimization with sparsity-inducing penalties. Foundations and Trends® in Machine Learning 4 (1), 1–106. doi:10.1561/2200000015.

Bader, B. W. & Kolda, T. G. 2008. Efficient MATLAB computations with sparse and factored tensors. SIAM Journal on Scientific Computing 30 (1), 205–231. doi:10.1137/060676489.

Bertsekas, D. P. 2016. Nonlinear Programming, Third Edition. Belmont, Massachusetts: Athena Scientific.

Boyd, S. 2011. Distributed optimization and statistical learning via the alternating direction method of multipliers. Foundations and Trends® in Machine Learning 3 (1), 1–122. doi:10.1561/2200000016.

Bruckstein, A. M., Donoho, D. L. & Elad, M. 2009. From sparse solutions of systems of equations to sparse modeling of signals and images. SIAM Review 51 (1), 34–81. doi:10.1137/060657704.

Cai, D., He, X., Han, J. & Huang, T. S. 2011. Graph regularized nonnegative matrix factorization for data representation. IEEE Transactions on Pattern Analysis and Machine Intelligence 33 (8), 1548–1560. doi:10.1109/tpami.2010.231.

Cichocki, A., Mandic, D., Lathauwer, L. D., Zhou, G., Zhao, Q., Caiafa, C. & PHAN, H. A. 2015. Tensor decompositions for signal processing applications: From two-way to multiway component analysis. IEEE Signal Processing Magazine 32 (2), 145–163. doi:10.1109/msp.2013.2297439.

Cichocki, A. & Phan, A.-H. 2009. Fast local algorithms for large scale nonnegative matrix and tensor factorizations. IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences E92-A (3), 708–721. doi:10.1587/transfun.e92.a.708.

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

Cichocki, A. & Zdunek, R. 2006. NMFLAB - MATLAB Toolbox for Non-Negative Matrix Factorization.

Cohen, M. X. 2017. Where does EEG come from and what does it mean? Trends in Neurosciences 40 (4), 208–218. doi:10.1016/j.tins.2017.02.004.

Cohen, M. X. 2019. A better way to define and describe morlet wavelets for time-frequency analysis. NeuroImage 199, 81–86. doi:10.1016/j.neuroimage.2019.05.048.

Cohen, M. X. 2014. Analyzing neural time series data: theory and practice. MIT press.

Comon, P. & Jutten, C. 2010. Handbook of Blind Source Separation: Independent component analysis and applications. Academic press.

Cong, F., Alluri, V., Nandi, A. K., Toiviainen, P., Fa, R., Abu-Jamous, B., Gong, L., Craenen, B. G. W., Poikonen, H., Huotilainen, M. & Ristaniemi, T. 2013. Linking brain responses to naturalistic music through analysis of ongoing EEG and stimulus features. IEEE Transactions on Multimedia 15 (5), 1060–1069. doi:10.1109/tmm.2013.2253452.

Cong, F., Lin, Q.-H., Kuang, L.-D., Gong, X.-F., Astikainen, P. & Ristaniemi, T. 2015a. Tensor decomposition of EEG signals: A brief review. Journal of Neuroscience Methods 248, 59–69. doi:10.1016/j.jneumeth.2015.03.018.

Cong, F., Ristaniemi, T. & Lyytinen, H. 2015b. Advanced signal processing on brain event-related potentials: filtering ERPs in time, frequency and space domains sequentially and simultaneously. World Scientific.

Dai, X., Li, C., He, X. & Li, C. 2018. Nonnegative matrix factorization algorithms based on the inertial projection neural network. Neural Computing and Applications. doi:10.1007/s00521-017-3337-5.

David, O., Kilner, J. M. & Friston, K. J. 2006. Mechanisms of evoked and induced responses in MEG/EEG. NeuroImage 31 (4), 1580–1591. doi:10.1016/j.neuroimage.2006.02.034.

Delorme, A. & Makeig, S. 2004. EEGLAB: an open source toolbox for analysis of single-trial EEG dynamics including independent component analysis. Journal of Neuroscience Methods 134 (1), 9–21. doi:10.1016/j.jneumeth.2003.10.009.

Delorme, A., Sejnowski, T. & Makeig, S. 2007. Enhanced detection of artifacts in EEG data using higher-order statistics and independent component analysis. NeuroImage 34 (4), 1443–1449. doi:10.1016/j.neuroimage.2006.11.004.

Dien, J., Beal, D. J. & Berg, P. 2005. Optimizing principal components analysis of event-related potentials: Matrix type, factor loading weighting, extraction, and rotations. Clinical Neurophysiology 116 (8), 1808–1825. doi:10.1016/j.clinph.2004.11.025.

Dien, J. 2012. Applying principal components analysis to event-related potentials: A tutorial. Developmental Neuropsychology 37 (6), 497–517. doi:10.1080/87565641.2012.697503.

Donoho, D. L. 2006. For most large underdetermined systems of linear equations the minimal $\ell_1$-norm solution is also the sparsest solution. Communications on Pure and Applied Mathematics 59 (6), 797–829. doi:10.1002/cpa.20132.

Elcoroaristizabal, S., Bro, R., García, J. A. & Alonso, L. 2015. PARAFAC models of fluorescence data with scattering: A comparative study. Chemometrics and Intelligent Laboratory Systems 142, 124–130. doi:10.1016/j.chemolab.2015.01.017.

Esposito, F., Gillis, N. & Buono, N. D. 2019. Orthogonal joint sparse NMF for microarray data analysis. Journal of Mathematical Biology 79 (1), 223–247. doi:10.1007/s00285-019-01355-2.

Gillis, N. & Glineur, F. 2012. Accelerated multiplicative updates and hierarchical ALS algorithms for nonnegative matrix factorization. Neural Computation 24 (4), 1085–1105. doi:10.1162/neco_a_00256.

Gillis, N. 2015. The why and how of nonnegative matrix factorization. Regularization, Optimization, Kernels, and Support Vector Machines 12 (257), 257–291.

Guan, N., Tao, D., Luo, Z. & Yuan, B. 2012. NeNMF: An optimal gradient method for nonnegative matrix factorization. IEEE Transactions on Signal Processing 60 (6), 2882–2898. doi:10.1109/tsp.2012.2190406.

Handy, T. C. 2005. Event-related potentials: A methods handbook. MIT press.

Herrmann, C. S., Rach, S., Vosskuhl, J. & Strüber, D. 2014. Time–frequency analysis of event-related potentials: A brief tutorial. Brain Topography 27 (4), 438–450. doi:10.1007/s10548-013-0327-5.

Hoyer, P. O. 2004. Non-negative matrix factorization with sparseness constraints. Journal of machine learning research 5 (Nov), 1457-1469.

Huang, K., Sidiropoulos, N. D. & Liavas, A. P. 2016. A flexible and efficient algorithmic framework for constrained matrix and tensor factorization. IEEE Transactions on Signal Processing 64 (19), 5052–5065. doi:10.1109/tsp.2016.2576427.

Hyvärinen, A. & Oja, E. 2000. Independent component analysis: algorithms and applications. Neural Networks 13 (4-5), 411–430. doi:10.1016/s0893-6080(00)00026-5.

Hyvärinen, A. 2013. Independent component analysis: recent advances. Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences 371 (1984), 20110534. doi:10.1098/rsta.2011.0534.

Idaji, M. J., Shamsollahi, M. B. & Sardouie, S. H. 2017. Higher order spectral regression discriminant analysis (HOSRDA): A tensor feature reduction method for ERP detection. Pattern Recognition 70, 152–162. doi:10.1016/j.patcog.2017.05.004.

Jia, S. & Qian, Y. 2009. Constrained nonnegative matrix factorization for hyperspectral unmixing. IEEE Transactions on Geoscience and Remote Sensing 47 (1), 161–173. doi:10.1109/tgrs.2008.2002882.

Jiang, S., Li, K. & Xu, R. Y. D. 2019. Relative pairwise relationship constrained non-negative matrix factorisation. IEEE Transactions on Knowledge and Data Engineering 31 (8), 1595–1609. doi:10.1109/tkde.2018.2859223.

Kappenman, E. S. & Luck, S. J. 2012. ERP components: The ups and downs of brainwave recordings. The Oxford handbook of event-related potential components, 3–30.

Kim, H. & Park, H. 2007. Sparse non-negative matrix factorizations via alternating non-negativity-constrained least squares for microarray data analysis. Bioinformatics 23 (12), 1495–1502. doi:10.1093/bioinformatics/btm134.

Kim, H. & Park, H. 2008. Nonnegative matrix factorization based on alternating nonnegativity constrained least squares and active set method. SIAM Journal on Matrix Analysis and Applications 30 (2), 713–730. doi:10.1137/07069239x.

Kim, J., He, Y. & Park, H. 2014. Algorithms for nonnegative matrix and tensor factorizations: a unified view based on block coordinate descent framework. Journal of Global Optimization 58 (2), 285–319. doi:10.1007/s10898-013-0035-4.

Kim, J. & Park, H. 2011. Fast nonnegative matrix factorization: An active-set-like method and comparisons. SIAM Journal on Scientific Computing 33 (6), 3261–3281. doi:10.1137/110821172.

Kim, J. & Park, H. 2012. Fast nonnegative tensor factorization with an active-set-like method. In High-Performance Scientific Computing. Springer London, 311–326. doi:10.1007/978-1-4471-2437-5_16.

Kolda, T. G. & Bader, B. W. 2009. Tensor decompositions and applications. SIAM Review 51 (3), 455–500. doi:10.1137/07070111x.

Lee, D. D. & Seung, H. S. 1999. Learning the parts of objects by non-negative matrix factorization. Nature 401 (6755), 788–791. doi:10.1038/44565.

Lee, D. D. & Seung, H. S. 2001. Algorithms for non-negative matrix factorization. In Advances in neural information processing systems, 556–562.

Lee, H. & Choi, S. 2009. Group nonnegative matrix factorization for eeg classification. In Artificial Intelligence and Statistics, 320–327.

Li, N., Kindermann, S. & Navasca, C. 2013. Some convergence results on the regularized alternating least-squares method for tensor decomposition. Linear Algebra and its Applications 438 (2), 796–812. doi:10.1016/j.laa.2011.12.002.

Liavas, A. P. & Sidiropoulos, N. D. 2015. Parallel algorithms for constrained tensor factorization via alternating direction method of multipliers. IEEE Transactions on Signal Processing 63 (20), 5450–5463. doi:10.1109/tsp.2015.2454476.

Lin, C.-J. 2007. Projected gradient methods for nonnegative matrix factorization. Neural Computation 19 (10), 2756–2779. doi:10.1162/neco.2007.19.10.2756.

Luck, S. J. 2014. An introduction to the event-related potential technique, second edition. MIT press.

Mahyari, A. G., Zoltowski, D. M., Bernat, E. M. & Aviyente, S. 2017. A tensor decomposition-based approach for detecting dynamic network states from EEG. IEEE Transactions on Biomedical Engineering 64 (1), 225–237. doi:10.1109/tbme.2016.2553960.

Makeig, S. & Onton, J. 2012. ERP features and EEG dynamics: an ICA perspective. Oxford handbook of event-related potential components, 51–87.

Mørup, M., Hansen, L. K. & Arnfred, S. M. 2007. Erpwavelab: A toolbox for multichannel analysis of time–frequency transformed event related potentials. Journal of Neuroscience Methods 161 (2), 361–368. doi:10.1016/j.jneumeth.2006.11.008.

Mørup, M., Hansen, L. K., Parnas, J. & Arnfred, S. M. 2006. Decomposing the time-frequency representation of EEG using non-negative matrix and multi-way factorization. Technical University of Denmark Technical Report. ⟨URL:http://www2.imm.dtu.dk/pubdb/views/edoc_download.php/4144/pdf/imm4144.pdf⟩.

Mørup, M. 2011. Applications of tensor (multiway array) factorizations and decompositions in data mining. Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery 1 (1), 24–40. doi:10.1002/widm.1.

Qian, Y., Jia, S., Zhou, J. & Robles-Kelly, A. 2011. Hyperspectral unmixing via $L_{1/2}$ sparsity-constrained nonnegative matrix factorization. IEEE Transactions on Geoscience and Remote Sensing 49 (11), 4282–4297. doi:10.1109/tgrs.2011.2144605.

Radüntz, T., Scouten, J., Hochmuth, O. & Meffert, B. 2017. Automated EEG artifact elimination by applying machine learning algorithms to ICA-based features. Journal of Neural Engineering 14 (4), 046004. doi:10.1088/1741-2552/aa69d1.

Sanei, S. & Chambers, J. A. 2007. EEG signal processing. John Wiley & Sons.

Shang, R., Liu, C., Meng, Y., Jiao, L. & Stolkin, R. 2017. Nonnegative matrix factorization with rank regularization and hard constraint. Neural Computation 29 (9), 2553–2579. doi:10.1162/neco_a_00995.

Shlens, J. 2014a. A tutorial on independent component analysis. arXiv preprint arXiv:1404.2986. ⟨URL:http://arxiv.org/abs/1404.2986⟩.

Shlens, J. 2014b. A tutorial on principal component analysis. arXiv preprint arXiv:1404.1100. ⟨URL:http://arxiv.org/abs/1404.1100⟩.

Sidiropoulos, N. D., Lathauwer, L. D., Fu, X., Huang, K., Papalexakis, E. E. & Faloutsos, C. 2017. Tensor decomposition for signal processing and machine learning. IEEE Transactions on Signal Processing 65 (13), 3551–3582. doi:10.1109/tsp.2017.2690524.

Sigurdsson, J., Ulfarsson, M. O. & Sveinsson, J. R. 2014. Hyperspectral unmixing with $l_q$ regularization. IEEE Transactions on Geoscience and Remote Sensing 52 (11), 6793–6806. doi:10.1109/tgrs.2014.2303155.

Smith, L. I. 2002. A tutorial on principal components analysis. Department of Computer Science, University of Otago. ⟨URL:https://ourarchive.otago.ac.nz/bitstream/handle/10523/7534/OUCS-2002-12.pdf⟩.

Sonkusare, S., Breakspear, M. & Guo, C. 2019. Naturalistic stimuli in neuroscience: Critically acclaimed. Trends in Cognitive Sciences 23 (8), 699–714. doi:10.1016/j.tics.2019.05.004.

Tharwat, A. 2018. Independent component analysis: An introduction. Applied Computing and Informatics. doi:10.1016/j.aci.2018.08.006.

Timmerman, M. E. & Kiers, H. A. L. 2000. Three-mode principal components analysis: Choosing the numbers of components and sensitivity to local optima. British Journal of Mathematical and Statistical Psychology 53 (1), 1–16. doi:10.1348/000711000159132.

Veganzones, M. A., Cohen, J. E., Farias, R. C., Chanussot, J. & Comon, P. 2016. Nonnegative tensor CP decomposition of hyperspectral data. IEEE Transactions on Geoscience and Remote Sensing 54 (5), 2577–2588. doi:10.1109/tgrs.2015.2503737.

Vervliet, N. & Lathauwer, L. D. 2019. Numerical optimization-based algorithms for data fusion. In Data Handling in Science and Technology. Elsevier, 81–128. doi:10.1016/b978-0-444-63984-4.00004-1.

Vu, X., Chaux, C., Thirion-Moreau, N., Maire, S. & Carstea, E. M. 2017. A new penalized nonnegative third-order tensor decomposition using a block coordinate proximal gradient approach: Application to 3d fluorescence spectroscopy. Journal of Chemometrics 31 (4), e2859. doi:10.1002/cem.2859.

Wang, D., Cong, F. & Ristaniemi, T. 2019. Higher-order nonnegative CANDECOMP/PARAFAC tensor decomposition using proximal algorithm. In 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, 3457–3461. doi:10.1109/icassp.2019.8683217.

Wang, D., Wang, X., Zhu, Y., Toiviainen, P., Huotilainen, M., Ristaniemi, T. & Cong, F. 2018a. Increasing stability of EEG components extraction using sparsity regularized tensor decomposition. In Advances in Neural Networks – ISNN 2018. Springer International Publishing, 789–799. doi:10.1007/978-3-319-92537-0_89.

Wang, D., Zhu, Y., Ristaniemi, T. & Cong, F. 2018b. Extracting multi-mode ERP features using fifth-order nonnegative tensor decomposition. Journal of Neuroscience Methods 308, 240–247. doi:10.1016/j.jneumeth.2018.07.020.

Wang, Y.-X. & Zhang, Y.-J. 2013. Nonnegative matrix factorization: A comprehensive review. IEEE Transactions on Knowledge and Data Engineering 25 (6), 1336–1353. doi:10.1109/tkde.2012.51.

Williams, A. H., Kim, T. H., Wang, F., Vyas, S., Ryu, S. I., Shenoy, K. V., Schnitzer, M., Kolda, T. G. & Ganguli, S. 2018. Unsupervised discovery of demixed, low-dimensional neural dynamics across multiple timescales through tensor component analysis. Neuron 98 (6), 1099–1115.e8. doi:10.1016/j.neuron.2018.05.015.

Xu, W., Liu, X. & Gong, Y. 2003. Document clustering based on non-negative matrix factorization. In Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval - SIGIR '03. ACM Press. doi:10.1145/860435.860485.

Xu, Y. & Yin, W. 2013. A block coordinate descent method for regularized multiconvex optimization with applications to nonnegative tensor factorization and completion. SIAM Journal on Imaging Sciences 6 (3), 1758–1789. doi:10.1137/120887795.

Xu, Z., Chang, X., Xu, F. & Zhang, H. 2012. $l_{1/2}$regularization: A thresholding representation theory and a fast solver. IEEE Transactions on Neural Networks and Learning Systems 23 (7), 1013–1027. doi:10.1109/tnnls.2012.2197412.

Zdunek, R. & Cichocki, A. 2006. Non-negative matrix factorization with quasi-newton optimization. In International conference on artificial intelligence and soft computing. Springer, 870–879.

Zhang, Y., Zhou, G., Zhao, Q., Cichocki, A. & Wang, X. 2016. Fast nonnegative tensor factorization based on accelerated proximal gradient and low-rank approximation. Neurocomputing 198, 148–154. doi:10.1016/j.neucom.2015.08.122.

Zhou, G., Cichocki, A., Zhao, Q. & Xie, S. 2014. Nonnegative matrix and tensor factorizations : An algorithmic perspective. IEEE Signal Processing Magazine 31 (3), 54–65. doi:10.1109/msp.2014.2298891.

Zhou, G., Zhao, Q., Zhang, Y., Adali, T., Xie, S. & Cichocki, A. 2016. Linked component analysis from matrices to high-order tensors: Applications to biomedical data. Proceedings of the IEEE 104 (2), 310–331. doi:10.1109/jproc.2015.2474704.

# ORIGINAL PAPERS

# PI

## SPARSE NONNEGATIVE TENSOR DECOMPOSITION IN INEXACT BLOCK COORDINATE DESCENT FRAMEWORK

by

Deqing Wang, Xiulin Wang, Tapani Ristaniemi and Fengyu Cong

Submitted to a journal

# Sparse Nonnegative Tensor Decomposition in Inexact Block Coordinate Descent Framework

Deqing Wang, *Student Member, IEEE,* Xiulin Wang, *Student Member, IEEE,*
Tapani Ristaniemi, *Senior Member, IEEE,* and Fengyu Cong, *Senior Member, IEEE*

**Abstract**—Nonnegative tensor decomposition is a versatile tool for multiway data analysis, by which the extracted components are nonnegative and usually sparse. However, the sparsity is only a side effect, which cannot be controlled without additional regularization. In this paper, we investigate the nonnegative CANDECOMP/PARAFAC (NCP) decomposition with sparse regularization item using $l_1$-norm (sparse NCP). We design sparse NCP using an inexact block coordinate descent scheme, which is able to accelerate the convergence. Five optimization methods in block coordinate descent framework are employed to solve the sparse NCP, including multiplicative update, alternating nonnegative quadratic programming, hierarchical altering least squares, alternating proximal gradient and alternating direction method of multipliers, all of which are carefully tailored to the sparse regularization problem. We evaluate all methods by experiments on both synthetic and real-world tensor data. The experimental results demonstrate that our proposed sparse NCP methods are able to efficiently impose sparsity to factor matrices and extract meaningful sparse components.

**Index Terms**—Tensor decomposition, nonnegative CANDECOMP/PARAFAC (NCP), sparse regularization, inexact block coordinate descent, proximal algorithm

## 1 Introduction

Nonnegative tensor decomposition is a powerful tool in signal processing and machine learning [1], [2]. Nonnegative CANDECOMP/PARAFAC (NCP), as an important decomposition method, has been widely applied to processing multiway data, such as hyperspectral data [3], electroencephalograph (EEG) data [4], [5], fluorescence excitation-emission matrix (EEM) data [6], [7], neural data [8], and many other multiway tensor data [9], [10]. In many cases, the extracted components by NCP are not only nonnegative but also sparse. For example, the spectral components from EEG tensor decomposition are usually very sparse [11], which are the spectra representing the narrowband frequencies of some brain activities. For another example, after decomposing EEM tensor, a component in the sample mode denotes the concentrations of a compound in all samples [12], which is sometimes also sparse. The nonnegative constraint in NCP will naturally lead to sparse results. However, this sparsity is only a side effect, which cannot be controlled to a certain level [13]. Therefore, in order to extract meaningful and accurate sparse components, additional sparse regularization is necessary for NCP tensor decomposition.

The design of NCP decomposition with explicit sparse regularization (sparse NCP) will benefit a lot from the methods in nonnegative matrix factorization (NMF) cases. On the one hand, the method of projecting components into sparse vectors at some sparsity level has been proposed in early study of NMF [13]. However, this method keeps all components at the same fixed sparsity level, which is not in line with the true sparsity of different components in the data. On the other hand, incorporating sparse regularization items into the optimization model is a popular method. The $l_1$-norm is a conventional and effective regularizer to impose sparsity for signal processing [14], since for most underdetermined linear equations the optimization problem with $l_1$-norm regularization can yield strong sparsity [15]. In addition, the $l_q$-norm ($0 < q < 1$) item [16], [17], [18] and the trace-norm based rank regularization item [19] have also become favourable to promote sparsity for NMF in recent years. More information about sparse regularization can be found in [20]. In a word, the method of incorporating appropriate sparse regularization into the optimization model is also viable for designing sparse NCP.

After constructing the sparse NCP model, it is necessary to select an efficient method to solve this optimization problem. Block coordinate descent (BCD) [21], [22] is an important framework for solving tensor decomposition especially constrained decomposition. The advantage of BCD method is that it is efficient for large-scale data and robust to noise. In BCD framework, each factor matrix is updated alternatively as a subproblem. In NMF and NCP studies, a lot of optimization methods have been proposed to solve the subproblem. Lee et al. proposed the multiplicative update (MU) method [23], [24], which is the most popular and widely applied method for NMF. Cichocki et al. proposed the hierarchical alternating least squares (HALS) method for large-scale problems [1], [25]. Xu et al. proposed the alternating proximal gradient (APG) method with detailed mathematical convergence proofs [21]. The similar ideas as

- *D. Wang, X. Wang and F. Cong are with the School of Biomedical Engineering, Faculty of Electronic Information and Electrical Engineering, Dalian University of Technology, Dalian 116024, China, and also with the Faculty of Information Technology, University of Jyväskylä, Jyväskylä 40100, Finland.*
  *E-mail: {deqing.wang, xiulin.wang}@foxmail.com; cong@dlut.edu.cn*
- *T. Ristaniemi is with the Faculty of Information Technology, University of Jyväskylä, Jyväskylä 40100, Finland.*
  *E-mail: tapani.e.ristaniemi@jyu.fi*

*Manuscript received Month Day, Year; revised Month Day, Year.*
*(Corresponding author: Fengyu Cong)*

APG were also proposed for NMF in [26] and NCP in [27]. Recently, the alternating direction method of multipliers (ADMM) has become popular for constrained decomposition [28], [29], [30]. In addition, the alternating nonnegative least squares (ANLS) method was deeply analyzed in Lin's seminal paper with strong optimization properties [31], which has a significant influence. When ANLS method is utilized, the subproblems appear as the nonnegative least squares (NNLS) problems. Many efficient methods have been devoted to solve the NNLS subproblems, such as Lin's project gradient method [31], quasi-Newton method [32], [33], active-set method [34], block principal pivoting method [35], [36] and inertial projection neural network [37]. Most of the above optimization methods can be naturally extended to solve sparse NCP problem.

A multitude of works have been devoted to incorporate sparse regularization to NMF [33], [34], [38], but rare works can be found for tensor decomposition. As far as we know, only a few studies had focused on imposing sparsity by $l_1$-norm regularization to Tucker decomposition [39], [40], [41]. In this work, we only study the sparse regularization with NCP decomposition. We design the sparse NCP using $l_1$-norm, which is the engine to impose sparsity explicitly. Five typical optimization methods are investigated to solve sparse NCP, including multiplicative update (MU), alternating nonnegative quadratic programming (ANQP)[1], hierarchical altering least squares (HALS), alternating proximal gradient (APG) and alternating direction method of multipliers (ADMM). These five methods are carefully tailored to imposing sparsity. Proximal algorithm [42], [43] is utilized to guarantee the convergence particularly for ANQP and HALS. Furthermore, it is reported that inexact block coordinate descent scheme could accelerate the convergence [44], [45]. Specifically, by inexact BCD, the subproblem is iterated several times for updating one factor matrix. Therefore, we design all methods of MU, ANQP, HALS, APG and ADMM using inexact BCD scheme for solving sparse NCP.

In this study, the designed sparse NCP algorithms are targeted at the following two aspects. First, when the sparse regularization parameter is increased, the sparse NCP can increase the sparsity level of factor matrix gradually. Specifically, when sparse regularization is imposed, more weak components will be removed (more full zero components will appear) in the factor matrix, and more weak elements will be suppressed (more zero elements will appear) in strong components. Second, the extracted sparse components should be meaningful and satisfy the properties of real-world applications. We evaluate all our sparse NCP methods on both synthetic and real-world tensor data. By selecting and tuning the sparse regularization properly, we demonstrate the effectiveness and efficiency of the sparse NCP methods to impose sparsity to factor matrices.

The rest of this paper is organized as follows. In Section 2, we describe the mathematical model of sparse NCP and introduce the inexact BCD framework. Section 3 elucidates the solutions to sparse NCP model using MU, ANQP, HALS, APG, and ADMM. In Section 4, we introduce the stopping conditions for both outer and inner loop in inexact BCD framework. Section 5 describes the detailed experiments on synthetic and real-world datasets. Finally, we conclude our paper in Section 6.

1. The alternating nonnegative quadratic programming (ANQP) is a general form of alternating nonnegative least squares (ANLS), which can also be solved by NNLS optimization methods.

## 2 Sparse Nonnegative CP Decomposition

In this paper, operator $\circ$ represents outer product of vectors, $\odot$ represents the Khatri-Rao product, $*$ represents the Hadamard product that is the elementwise matrix product, $\langle\ \rangle$ represents inner product, $[\![\ ]\!]$ represents Kruskal oprator and $[\ ]_+$ represents nonnegative projection. $\|\ \|_F$ denotes Frobenius norm, and $\|\ \|_1$ denotes $l_1$-norm. Basics of tensor computation and multi-linear algebra can be found in review papers [46], [47].

### 2.1 Mathematical Model

We present the mathematical model of the designed sparse NCP in this section.

Given a nonnegative $N$th-order tensor $\boldsymbol{\mathcal{X}} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$ and a positive number $R$, we design the sparse NCP as the following minimization problem:

$$\begin{aligned} \min_{\boldsymbol{A}^{(1)},\ldots,\boldsymbol{A}^{(N)}} \ & \frac{1}{2}\left\|\boldsymbol{\mathcal{X}} - [\![\boldsymbol{A}^{(1)},\ldots,\boldsymbol{A}^{(N)}]\!]\right\|_F^2 + \sum_{n=1}^{N}\beta_n \sum_{r=1}^{R}\left\|\boldsymbol{a}_r^{(n)}\right\|_1 \\ & \text{s.t. } \boldsymbol{A}^{(n)} \geqslant 0 \text{ for } n = 1,\ldots,N, \end{aligned} \tag{1}$$

where $\boldsymbol{A}^{(n)} \in \mathbb{R}^{I_n \times R}$ for $n = 1,\ldots,N$ are the estimated factor matrices in different modes, $\beta_n$ are positive regularization parameters in parameter vectors $\boldsymbol{\beta} \in \mathbb{R}^{N\times 1}$, $I_n$ is the size in mode-$n$, $\boldsymbol{a}_r^{(n)}$ represents the $r$th column of $\boldsymbol{A}^{(n)}$, and $R$ is the initial number of components. We use $\mathcal{F}_{\text{tensor}}(\boldsymbol{A}) = \mathcal{F}_{\text{tensor}}\left(\boldsymbol{A}^{(1)},\ldots,\boldsymbol{A}^{(N)}\right)$ to denote the objective function in (1).

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 $\boldsymbol{\mathcal{X}}$. The mode-$n$ unfolding of the estimated tensor in Kruskal operator $[\![\boldsymbol{A}^{(1)},\ldots,\boldsymbol{A}^{(N)}]\!]$ can be written as $\boldsymbol{A}^{(n)}\left(\boldsymbol{B}^{(n)}\right)^T$, in which $\boldsymbol{B}^{(n)} = \left(\boldsymbol{A}^{(N)} \odot \cdots \odot \boldsymbol{A}^{(n+1)} \odot \boldsymbol{A}^{(n-1)} \odot \cdots \odot \boldsymbol{A}^{(1)}\right) \in \mathbb{R}^{\prod_{\tilde{n}=1,\tilde{n}\neq n}^{N} I_{\tilde{n}} \times R}$. In BCD framework, factor $\boldsymbol{A}^{(n)}$ is updated alternatively by a subproblem in every iteration, which equals to the following minimization problem:

$$\begin{aligned} \min_{\boldsymbol{A}^{(n)}} \mathcal{F}\left(\boldsymbol{A}^{(n)}\right) = & \frac{1}{2}\left\|\boldsymbol{X}_{(n)} - \boldsymbol{A}^{(n)}\left(\boldsymbol{B}^{(n)}\right)^T\right\|_F^2 \\ & + \beta_n \sum_{r=1}^{R}\left\|\boldsymbol{a}_r^{(n)}\right\|_1 \\ & \text{s.t. } \boldsymbol{A}^{(n)} \geqslant 0. \end{aligned} \tag{2}$$

The sparse NCP problem in (1) is non-convex. However, the subproblem (2) with $l_1$-norm is convex [26], which is the key point to solve (1). The optimization methods introduced in the introduction will be applied to (2) directly. The partial gradient (or partial derivative) of $\mathcal{F}\left(\boldsymbol{A}^{(n)}\right)$ with respect to $\boldsymbol{A}^{(n)}$ is

$$\begin{aligned} & \frac{\partial}{\partial \boldsymbol{A}^{(n)}}\mathcal{F}\left(\boldsymbol{A}^{(n)}\right) \\ & = \boldsymbol{A}^{(n)}\left[\left(\boldsymbol{B}^{(n)}\right)^T\boldsymbol{B}^{(n)}\right] - \boldsymbol{X}_{(n)}\boldsymbol{B}^{(n)} + \beta_n\boldsymbol{E}, \end{aligned} \tag{3}$$

where $\boldsymbol{E} \in \mathbb{R}^{I_n \times R}$ is a matrix whose elements are all equal to 1 and $\boldsymbol{X}_{(n)}\boldsymbol{B}^{(n)}$ is called the *Matricized Tensor Times Khatri-Rao Product* (MTTKRP) [48]. The item $\left(\boldsymbol{B}^{(n)}\right)^T \boldsymbol{B}^{(n)}$ can be computed efficiently by

$$\begin{aligned}\left(\boldsymbol{B}^{(n)}\right)^T \boldsymbol{B}^{(n)} &= \left[\left(\boldsymbol{A}^{(N)}\right)^T \boldsymbol{A}^{(N)}\right] * \cdots \\ &* \left[\left(\boldsymbol{A}^{(n+1)}\right)^T \boldsymbol{A}^{(n+1)}\right] * \left[\left(\boldsymbol{A}^{(n-1)}\right)^T \boldsymbol{A}^{(n-1)}\right] \\ &* \cdots * \left[\left(\boldsymbol{A}^{(1)}\right)^T \boldsymbol{A}^{(1)}\right].\end{aligned} \tag{4}$$

### 2.2 Inexact Block Coordinate Descent Method

We solve the sparse NCP model using an inexact BCD scheme. Specifically, the factor matrices $\boldsymbol{A}^{(n)}, n = 1, \ldots, N$, are updated alternatively in outer iterations, meanwhile, in the subproblem (2), one factor $\boldsymbol{A}^{(n)}$ is also updated several times in inner iterations. It is reported that the inexact BCD scheme could accelerate the computation [44], [45]. The procedures are listed in **Algorithm 1**.

**Algorithm 1:** Inexact BCD Scheme

```
Initialization and preparation steps;
repeat
    for n = 1 to N do
        Basic computations before updating A^(n);
        repeat
            Update A^(n) by A^(n) = argmin_{A^(n) ⩾ 0} F(A^(n));
        until inner termination criterion is reached;
    end
until outer termination criterion is reached;
```

## 3 Optimization Methods for Solving Sparse NCP

In this section, we present the solutions to the sparse NCP problem in (1) by the optimization methods of MU, ANQP, HALS, APG and ADMM.

### 3.1 Multiplicative Update

Multiplicative update (MU) was first proposed by Lee et al for NMF [23], [24]. Cai et al. proposed a straightforword way using Lagrange multiplier to solve NMF subproblems [49], where the same update rules as Lee's method can be obtained. We extend Cai's method to tensor case. We represent tensor factor $\boldsymbol{A}^{(n)}$ elementwisely by $\boldsymbol{A}^{(n)} = \left[a_{ir}^{(n)}\right]$ for $i = 1, \ldots, I_n$ and $r = 1, \ldots, R$. Let $\psi_{ir}$ be the Lagrange multiplier for constraint $a_{ir}^{(n)} \geqslant 0$, and $\boldsymbol{\Psi} = [\psi_{ir}]$. Based on (2), the Lagrange $\mathcal{L}$ is

$$\mathcal{L} = \mathcal{F}\left(\boldsymbol{A}^{(n)}\right) + \left\langle \boldsymbol{\Psi}, \boldsymbol{A}^{(n)} \right\rangle. \tag{5}$$

The partial derivative of $\mathcal{L}$ with respect to $\boldsymbol{A}^{(n)}$ is

$$\frac{\partial \mathcal{L}}{\partial \boldsymbol{A}^{(n)}} = -\boldsymbol{X}_{(n)}\boldsymbol{B}^{(n)} + \boldsymbol{A}^{(n)}\left(\boldsymbol{B}^{(n)}\right)^T \boldsymbol{B}^{(n)} + \beta_n \boldsymbol{E} + \boldsymbol{\Psi}. \tag{6}$$

Using KKT condition $\psi_{ir} a_{ir}^{(n)} = 0$, we obtain the following equation for $a_{ir}^{(n)}$:

$$\begin{aligned}&-\left[\boldsymbol{X}_{(n)}\boldsymbol{B}^{(n)}\right]_{(i,r)} a_{ir}^{(n)} + \\ &\left[\boldsymbol{A}^{(n)}\left(\boldsymbol{B}^{(n)}\right)^T \boldsymbol{B}^{(n)} + \beta_n \boldsymbol{E}\right]_{(i,r)} a_{ir}^{(n)} = 0.\end{aligned} \tag{7}$$

This equation leads to the following multiplicative updating rule:

$$a_{ir}^{(n)} \leftarrow a_{ir}^{(n)} \frac{\left[\boldsymbol{X}_{(n)}\boldsymbol{B}^{(n)}\right]_{(i,r)} + \epsilon}{\left[\boldsymbol{A}^{(n)}\left(\boldsymbol{B}^{(n)}\right)^T \boldsymbol{B}^{(n)} + \beta_n \boldsymbol{E}\right]_{(i,r)} + \epsilon}, \tag{8}$$

where $\epsilon$ is a small positive value to avoid zero element. With positive initialization factors, the updating rule (8) is able to introduce zeros in $\boldsymbol{A}^{(n)}$ (impose sparsity), which is different from the updating rule in [44] avoiding zeros in $\boldsymbol{A}^{(n)}$. The implementation of MU method is listed in **Algorithm 2**.

**Algorithm 2:** MU for sparse NCP in (1)

```
Input  : X, R, β
Output: A^(n), n = 1, ..., N
Initialize A^(n) ∈ R^{I_n × R}, n = 1, ..., N, using
  positive random numbers;
repeat
    for n = 1 to N do
        Make mode-n unfolding of X as X_(n) and
          compute MTTKRP X_(n) B^(n);
        Compute (B^(n))^T B^(n) based on (4);
        repeat
            for i = 1 to I_n do
                for r = 1 to R do
                    Update a_ir^(n) according to (8);
                end
            end
        until inner termination criterion is reached;
    end
until outer termination criterion is reached;
return A^(n), n = 1, ..., N.
```

### 3.2 Alternating Nonnegative Quadratic Programming

Alternating Nonnegative Least Squares (ANLS) is an important method for NMF and NCP problems [22], [31]. Many efficient optimization methods were proposed to solve the nonnegative least squares (NNLS) subproblems, such as Lin's projected gradient (PG) [31], active-set (AS) [34], and block principal pivoting (BPP) [35]. However, when sparse regularization is imposed, many full zero columns will appear in the factor matrices $\boldsymbol{A}^{(n)}$. Thus, both $\boldsymbol{A}^{(n)}$ and $\boldsymbol{B}^{(n)}$ cannot guarantee to be of full rank. Therefore, ANLS method is prone to rank deficiency.

In order to avoid the rank deficiency and improve the stability of sparse NCP, we incorporate proximal algorithm

[42], [43] into the objective function in (2). In this way, we have the following minimization problem:

$$\begin{aligned}\min_{\boldsymbol{A}^{(n)}} \mathcal{F}_{\text{prox}}\left(\boldsymbol{A}^{(n)}\right) &= \frac{1}{2}\left\|\boldsymbol{X}_{(n)} - \boldsymbol{A}^{(n)}\left(\boldsymbol{B}^{(n)}\right)^T\right\|_F^2 \\ &+ \frac{\alpha_n}{2}\left\|\boldsymbol{A}^{(n)} - \breve{\boldsymbol{A}}^{(n)}\right\|_F^2 + \beta_n \sum_{r=1}^{R}\left\|\boldsymbol{a}_r^{(n)}\right\|_1 \\ &\text{s.t. } \boldsymbol{A}^{(n)} \geqslant 0,\end{aligned} \tag{9}$$

in which $\breve{\boldsymbol{A}}^{(n)}$ is the value of factor $\boldsymbol{A}^{(n)}$ in previous iteration during updating and $\alpha_n$ are positive regularization parameters in vectors $\boldsymbol{\alpha} \in \mathbb{R}^{N\times 1}$. It has been proved that the solution of (9) is equivalent to the solution of (2) [42]. Afterwards, we try solving (9) combining NNLS methods.

In spite of the efficiency of NNLS, the subproblem of (9) cannot be represented in least squares form due to the $l_1$-norm item. Luckily, inspired by [50], (9) can be represented in the nonnegative quadratic programming (NNQP) form as the following problem:

$$\begin{aligned}\min_{\boldsymbol{A}^{(n)}} \sum_{i=1}^{I_n}\Bigg\{ &\frac{1}{2}\left[\boldsymbol{A}^{(n)}\right]_{(i,:)}\boldsymbol{M}\left[\boldsymbol{A}^{(n)}\right]_{(i,:)}^T + \boldsymbol{N}_{(i,:)}\left[\boldsymbol{A}^{(n)}\right]_{(i,:)}^T \\ &+ \frac{1}{2}\left[\boldsymbol{X}_{(n)}\right]_{(i,:)}\left[\boldsymbol{X}_{(n)}\right]_{(i,:)}^T\Bigg\} \\ &\text{s.t. } \boldsymbol{A}^{(n)} \geqslant 0,\end{aligned} \tag{10}$$

where $\left[\boldsymbol{A}^{(n)}\right]_{(i,:)}$ represents the $i$th row of $\boldsymbol{A}^{(n)}$, $\boldsymbol{M} = \left(\boldsymbol{B}^{(n)}\right)^T\boldsymbol{B}^{(n)} + \alpha_n\boldsymbol{I}_R$ and $\boldsymbol{N} = \beta_n\boldsymbol{E} - \boldsymbol{X}_{(n)}\boldsymbol{B}^{(n)} - \alpha_n\breve{\boldsymbol{A}}^{(n)}$.

The above mentioned optimization methods for NNLS can also be used to solve NNQP problem. In this study, we only use block principal pivoting (BPP) [35] as the NNQP solver. In fact, NNQP is a general case of NNLS [50]. We name the method to solve tensor decomposition using NNQP as alternating nonnegative quadratic programming (ANQP). **Algorithm 3** explicates the ANQP method for sparse NCP.

### 3.3 Hierarchical Alternating Least Squares

Hierarchical alternating least squares (HALS) is a method to update each factor column by column. For the sake of simplification, we use $\boldsymbol{a}_r$ and $\boldsymbol{b}_r$ instead of $\boldsymbol{a}_r^{(n)}$ and $\boldsymbol{b}_r^{(n)}$ in this part, which are the $r$th column of $\boldsymbol{A}^{(n)}$ and $\boldsymbol{B}^{(n)}$ respectively. We also use $\left[\boldsymbol{A}^{(n)}\right]_{(:,r)} = \boldsymbol{a}_r \in \mathbb{R}^{I_n\times 1}$ to represent the column of a matrix, and $\left[\boldsymbol{A}^{(n)}\right]_{(i,r)} = a_{ir}^{(n)}$ to represent an element in a matrix.

In order to improve the stability of HALS, we still employ the proximal algorithm. The subproblem is the same as (9). The objective function in (9) can be further represented as

$$\begin{aligned}\mathcal{F}_{\text{prox}}\left(\boldsymbol{A}^{(n)}\right) =& \frac{1}{2}\left\|\boldsymbol{X}_{(n)} - \sum_{r=1}^{R}\boldsymbol{a}_r\boldsymbol{b}_r^T\right\|_F^2 \\ &+ \frac{\alpha_n}{2}\sum_{r=1}^{R}\|\boldsymbol{a}_r - \breve{\boldsymbol{a}}_r\|_2^2 + \beta_n\sum_{r=1}^{R}\|\boldsymbol{a}_r\|_1,\end{aligned} \tag{11}$$

**Algorithm 3:** ANQP for sparse NCP in (1)

**Input :** $\mathcal{X}$, $R$, $\boldsymbol{\alpha}$, $\boldsymbol{\beta}$
**Output:** $\boldsymbol{A}^{(n)}$, $n = 1, \ldots, N$
Initialize $\boldsymbol{A}^{(n)} \in \mathbb{R}^{I_n\times R}$, $n = 1, \ldots, N$, using nonnegative random numbers;
**repeat**
  **for** $n = 1$ **to** $N$ **do**
    Make mode-$n$ unfolding of $\mathcal{X}$ as $\boldsymbol{X}_{(n)}$ and compute MTTKRP $\boldsymbol{X}_{(n)}\boldsymbol{B}^{(n)}$;
    Compute $\left(\boldsymbol{B}^{(n)}\right)^T\boldsymbol{B}^{(n)}$ based on (4);
    $\boldsymbol{X}_{(n)}\boldsymbol{B}^{(n)} \leftarrow \boldsymbol{X}_{(n)}\boldsymbol{B}^{(n)} + \alpha_n\breve{\boldsymbol{A}}^{(n)} - \beta_n\boldsymbol{E}$;
    $\left(\boldsymbol{B}^{(n)}\right)^T\boldsymbol{B}^{(n)} \leftarrow \left(\boldsymbol{B}^{(n)}\right)^T\boldsymbol{B}^{(n)} + \alpha_n\boldsymbol{I}_R$;
    **repeat**
      Update factor $\boldsymbol{A}^{(n)}$, $n = 1, \ldots, N$, using NNQP based on BPP method:
      $\boldsymbol{A}^{(n)} = \underset{\boldsymbol{A}^{(n)}\geqslant 0}{\operatorname{argmin}} \mathcal{F}_{\text{prox}}\left(\boldsymbol{A}^{(n)}\right)$
      $= \texttt{NNQP\_BPP}\left(\boldsymbol{X}_{(n)}\boldsymbol{B}^{(n)}, \left(\boldsymbol{B}^{(n)}\right)^T\boldsymbol{B}^{(n)}\right)$;
    **until** *inner termination criterion is reached*;
  **end**
**until** *outer termination criterion is reached*;
**return** $\boldsymbol{A}^{(n)}$, $n = 1, \ldots, N$.

where $\breve{\boldsymbol{a}}_r$ is the $r$th column of $\breve{\boldsymbol{A}}^{(n)}$. The minimization problem for (11) can be solved iteratively by columnwise subproblems:

$$\begin{aligned}\min_{\boldsymbol{a}_r} \mathcal{F}_r = \frac{1}{2}\left\|\boldsymbol{Z}_r - \boldsymbol{a}_r\boldsymbol{b}_r^T\right\|_F^2 + \frac{\alpha_n}{2}\|\boldsymbol{a}_r - \breve{\boldsymbol{a}}_r\|_2^2 + \beta_n\|\boldsymbol{a}_r\|_1 \\ \text{s.t. } \boldsymbol{a}_r \geqslant 0,\end{aligned} \tag{12}$$

for $r = 1, \ldots, R$, in which

$$\boldsymbol{Z}_r = \boldsymbol{X}_{(n)} - \sum_{\tilde{r}=1,\tilde{r}\neq r}^{R}\boldsymbol{a}_{\tilde{r}}\boldsymbol{b}_{\tilde{r}}^T. \tag{13}$$

The partial derivative of $\mathcal{F}_r$ with respect to $\boldsymbol{a}_r$ is

$$\begin{aligned}\frac{\partial\mathcal{F}_r}{\partial\boldsymbol{a}_r} &= \left(\boldsymbol{a}_r\boldsymbol{b}_r^T - \boldsymbol{Z}_r\right)\boldsymbol{b}_r + \alpha_n\boldsymbol{a}_r - \alpha_n\breve{\boldsymbol{a}}_r + \beta_n\boldsymbol{1}, \\ &= \left(\boldsymbol{b}_r^T\boldsymbol{b}_r + \alpha_n\right)\boldsymbol{a}_r - \left(\boldsymbol{Z}_r\boldsymbol{b}_r + \alpha_n\breve{\boldsymbol{a}}_r - \beta_n\boldsymbol{1}\right),\end{aligned} \tag{14}$$

where $\boldsymbol{1} \in \mathbb{R}^{I_n\times 1}$ is a vector with all elements equal to 1. When $\frac{\partial\mathcal{F}_r}{\partial\boldsymbol{a}_r} = 0$, nonnegative column vector $\boldsymbol{a}_r$ can be updated as

$$\boldsymbol{a}_r \leftarrow \left[\frac{\boldsymbol{Z}_r\boldsymbol{b}_r + \alpha_n\breve{\boldsymbol{a}}_r - \beta_n\boldsymbol{1}}{\boldsymbol{b}_r^T\boldsymbol{b}_r + \alpha_n}\right]_+, \tag{15}$$

which is a closed form solution [22].

A fast HALS method was proposed to solve large-scale problem [1], [22]. We use the same idea to solve the sparse NCP problem. $\boldsymbol{Z}_r$ in (13) can also be represented as

$$\boldsymbol{Z}_r = \boldsymbol{X}_{(n)} - \sum_{\tilde{r}=1}^{R}\boldsymbol{a}_{\tilde{r}}\boldsymbol{b}_{\tilde{r}}^T + \breve{\boldsymbol{a}}_r\boldsymbol{b}_r^T. \tag{16}$$

Replacing $\boldsymbol{Z}_r$ in (15) by (16), we obtain the new update rule for $\boldsymbol{a}_r$ as shown in (17) in the top of next page.

The procedures of HALS are illustrated in **Algorithm 4**.

$$\begin{aligned}\boldsymbol{a}_r \leftarrow \left[\frac{\left(\boldsymbol{X}_{(n)}-\sum_{\tilde{r}=1}^{R}\boldsymbol{a}_{\tilde{r}}\boldsymbol{b}_{\tilde{r}}^T+\breve{\boldsymbol{a}}_r\boldsymbol{b}_r^T\right)\boldsymbol{b}_r+\alpha_n\breve{\boldsymbol{a}}_r-\beta_n\mathbf{1}}{\boldsymbol{b}_r^T\boldsymbol{b}_r+\alpha_n}\right]_+ &= \left[\breve{\boldsymbol{a}}_r+\frac{\boldsymbol{X}_{(n)}\boldsymbol{b}_r-\sum_{\tilde{r}=1}^{R}\boldsymbol{a}_{\tilde{r}}\boldsymbol{b}_{\tilde{r}}^T\boldsymbol{b}_r-\beta_n\mathbf{1}}{\boldsymbol{b}_r^T\boldsymbol{b}_r+\alpha_n}\right]_+ \\ &= \left[\breve{\boldsymbol{a}}_r+\frac{\left[\boldsymbol{X}_{(n)}\boldsymbol{B}^{(n)}\right]_{(:,r)}-\boldsymbol{A}^{(n)}\left[\left(\boldsymbol{B}^{(n)}\right)^T\boldsymbol{B}^{(n)}\right]_{(:,r)}-\beta_n\mathbf{1}}{\left[\left(\boldsymbol{B}^{(n)}\right)^T\boldsymbol{B}^{(n)}\right]_{(r,r)}+\alpha_n}\right]_+\end{aligned} \tag{17}$$

**Algorithm 4:** HALS for sparse NCP in (1)

**Input** : $\mathcal{X}$, $R$, $\boldsymbol{\alpha}$, $\boldsymbol{\beta}$
**Output:** $\boldsymbol{A}^{(n)}$, $n=1,\ldots,N$
Initialize $\boldsymbol{A}^{(n)}\in\mathbb{R}^{I_n\times R}$, $n=1,\ldots,N$, using nonnegative random numbers;
**repeat**
  **for** $n=1$ **to** $N$ **do**
    Make mode-$n$ unfolding of $\mathcal{X}$ as $\boldsymbol{X}_{(n)}$ and compute MTTKRP $\boldsymbol{X}_{(n)}\boldsymbol{B}^{(n)}$;
    Compute $\left(\boldsymbol{B}^{(n)}\right)^T\boldsymbol{B}^{(n)}$ based on (4);
    **repeat**
      **for** $r=1$ **to** $R$ **do**
        Update $\boldsymbol{a}_r^{(n)}$ using (17);
      **end**
    **until** *inner termination criterion is reached*;
  **end**
**until** *outer termination criterion is reached*;
**return** $\boldsymbol{A}^{(n)}$, $n=1,\ldots,N$.

### 3.4 Alternating Proximal Gradient

The mathematical properties of alternating proximal gradient (APG) method were thoroughly analyzed in works of [21], [41] by Xu et al. APG method has exhibited excellent performances on both NMF and NCP problems [21], [26], [27], and it is also efficient to cope with $l_1$ sparse regularization [41]. Considering updating $\boldsymbol{A}^{(n)}$ at the $k$th iteration, APG is computed as follows.

We take

$$L_{k-1}^{(n)}=\left\|\left(\boldsymbol{B}_{k-1}^{(n)}\right)^T\boldsymbol{B}_{k-1}^{(n)}\right\|_2 \tag{18}$$

as Lipschitz constant, where $\|\boldsymbol{A}\|_2$ is spectral norm of matrix.

The extrapolation weight is computed by

$$\omega_{k-1}^{(n)}=\frac{t_{k-1}^{(n)}-1}{t_k^{(n)}}, \tag{19}$$

where $t_0^{(n)}=1$, $t_k^{(n)}=\frac{1}{2}\left(1+\sqrt{1+4(t_{k-1}^{(n)})^2}\right)$.

Let

$$\widehat{\boldsymbol{A}}_{k-1}^{(n)}=\boldsymbol{A}_{k-1}^{(n)}+\omega_{k-1}^{(n)}\left(\boldsymbol{A}_{k-1}^{(n)}-\boldsymbol{A}_{k-2}^{(n)}\right) \tag{20}$$

denote an extrapolated point, and let

$$\widehat{\boldsymbol{G}}_{k-1}^{(n)}=\widehat{\boldsymbol{A}}_{k-1}^{(n)}\left(\boldsymbol{B}_{k-1}^{(n)}\right)^T\boldsymbol{B}_{k-1}^{(n)}-\boldsymbol{X}_{(n)}\boldsymbol{B}_{k-1}^{(n)} \tag{21}$$

represent the gradient at $\widehat{\boldsymbol{A}}_{k-1}^{(n)}$. Factor $\boldsymbol{A}^{(n)}$ at iteration $k$ is updated by

$$\begin{aligned}\boldsymbol{A}_k^{(n)}=&\underset{\boldsymbol{A}^{(n)}\geqslant 0}{\operatorname{argmin}}\left\{\left\langle\widehat{\boldsymbol{G}}_{k-1}^{(n)},\boldsymbol{A}^{(n)}-\widehat{\boldsymbol{A}}_{k-1}^{(n)}\right\rangle\right. \\ &\left.+\frac{L_{k-1}^{(n)}}{2}\left\|\boldsymbol{A}^{(n)}-\widehat{\boldsymbol{A}}_{k-1}^{(n)}\right\|_F^2+\beta_n\sum_{r=1}^{R}\left\|\boldsymbol{a}_r^{(n)}\right\|_1\right\} \\ =&\operatorname{prox}_{\frac{\beta_n\|\cdot\|_1}{L_{k-1}^{(n)}}}\left(\widehat{\boldsymbol{A}}_{k-1}^{(n)}-\frac{\widehat{\boldsymbol{G}}_{k-1}^{(n)}}{L_{k-1}^{(n)}}-\boldsymbol{A}^{(n)}\right).\end{aligned} \tag{22}$$

The closed form of (22) can be written as

$$\boldsymbol{A}_k^{(n)}=\max\left(0,\ \widehat{\boldsymbol{A}}_{k-1}^{(n)}-\frac{\widehat{\boldsymbol{G}}_{k-1}^{(n)}}{L_{k-1}^{(n)}}-\frac{\beta_n\boldsymbol{E}}{L_{k-1}^{(n)}}\right) \tag{23}$$

APG method for sparse NCP can be implemented by procedures in **Algorithm 5**.

**Algorithm 5:** APG for sparse NCP in (1)

**Input** : $\mathcal{X}$, $R$, $\boldsymbol{\alpha}$, $\boldsymbol{\beta}$
**Output:** $\boldsymbol{A}^{(n)}$, $n=1,\ldots,N$
Initialize $\boldsymbol{A}^{(n)}\in\mathbb{R}^{I_n\times R}$, $n=1,\ldots,N$, using nonnegative random numbers;
**for** $k=1,2,\ldots$ **do**
  **for** $n=1$ **to** $N$ **do**
    Make mode-$n$ unfolding of $\mathcal{X}$ as $\boldsymbol{X}_{(n)}$ and compute MTTKRP $\boldsymbol{X}_{(n)}\boldsymbol{B}_{k-1}^{(n)}$;
    Compute $\left(\boldsymbol{B}_{k-1}^{(n)}\right)^T\boldsymbol{B}_{k-1}^{(n)}$ and $L_{k-1}^{(n)}$ based on (4) and (18);
    **repeat**
      Compute $\omega_{k-1}^{(n)}$, $\widehat{\boldsymbol{A}}_{k-1}^{(n)}$ and $\widehat{\boldsymbol{G}}_{k-1}^{(n)}$ according to (19), (20) and (21);
      Update $\boldsymbol{A}_k^{(n)}$ according to (23);
    **until** *inner termination criterion is reached*;
  **end**
  **if** $\mathcal{F}_{\text{tensor}}(\boldsymbol{A}_k)\geqslant\mathcal{F}_{\text{tensor}}(\boldsymbol{A}_{k-1})$ **then**
    Update $\boldsymbol{A}_k^{(n)}$ again according to (23) with $\widehat{\boldsymbol{A}}_{k-1}^{(n)}=\boldsymbol{A}_{k-1}^{(n)}$, $n=1,\ldots,N$;
  **end**
  **if** *outer termination criterion is reached* **then**
    **return** $\boldsymbol{A}_k^{(n)}$, for $n=1,\ldots,N$;
  **end**
**end**

### 3.5 Alternating Direction Method of Multipliers

In recent years, alternating direction method of multipliers (ADMM) has been widely applied to signal processing and machine learning [28]. ADMM is favoured for NMF and NCP due to its flexibility to handle a wide range of constraints and regularization items [30], [51]. We introduce the solution of sparse NCP in (1) using ADMM as follows.

Introducing an auxiliary variable $\widetilde{\boldsymbol{A}}^{(n)} \in \mathbb{R}^{I_n \times R}$, we reform the subproblem in (2) by

$$\begin{aligned} \min_{\boldsymbol{A}^{(n)}} \frac{1}{2}\left\|\boldsymbol{X}_{(n)} - \boldsymbol{A}^{(n)}\left(\boldsymbol{B}^{(n)}\right)^T\right\|_F^2 + \beta_n \sum_{r=1}^{R}\left\|\widetilde{\boldsymbol{a}}_r^{(n)}\right\|_1 \\ \text{s.t. } \boldsymbol{A}^{(n)} = \widetilde{\boldsymbol{A}}^{(n)}, \widetilde{\boldsymbol{A}}^{(n)} \geqslant 0, \end{aligned} \tag{24}$$

where $\widetilde{\boldsymbol{a}}_r^{(n)}$ is the $r$th column of $\widetilde{\boldsymbol{A}}^{(n)}$. The objective function in (24) is split by

$$f\left(\boldsymbol{A}^{(n)}\right) = \frac{1}{2}\left\|\boldsymbol{X}_{(n)} - \boldsymbol{A}^{(n)}\left(\boldsymbol{B}^{(n)}\right)^T\right\|_F^2,$$
$$\text{and } g\left(\widetilde{\boldsymbol{A}}^{(n)}\right) = \beta_n \sum_{r=1}^{R}\left\|\widetilde{\boldsymbol{a}}_r^{(n)}\right\|_1.$$

We construct the augmented Lagrangian function of (24) by

$$\begin{aligned} &\mathcal{L}_\rho\left(\boldsymbol{A}^{(n)}, \widetilde{\boldsymbol{A}}^{(n)}, \boldsymbol{\Lambda}^{(n)}\right) \\ =&f\left(\boldsymbol{A}^{(n)}\right) + g\left(\widetilde{\boldsymbol{A}}^{(n)}\right) \\ &+ \left\langle \boldsymbol{\Lambda}^{(n)}, \boldsymbol{A}^{(n)} - \widetilde{\boldsymbol{A}}^{(n)}\right\rangle + \frac{\rho_n}{2}\left\|\boldsymbol{A}^{(n)} - \widetilde{\boldsymbol{A}}^{(n)}\right\|_F^2, \end{aligned} \tag{25}$$

in which we employ the Lagrange multiplier (dual variable) $\boldsymbol{\Lambda}^{(n)} \in \mathbb{R}^{I_n \times R}$ and select $\rho_n = \text{tr}\left[\left(\boldsymbol{B}^{(n)}\right)^T \boldsymbol{B}^{(n)}\right]/R$ according to the empirical setting in [30].

We update $\boldsymbol{A}^{(n)}$, $\widetilde{\boldsymbol{A}}^{(n)}$ and $\boldsymbol{\Lambda}^{(n)}$ by inner iterations as follows:

1) Updating the primal variable $\boldsymbol{A}^{(n)}$

$$\boldsymbol{A}^{(n)} = \arg\min_{\boldsymbol{A}^{(n)}} \mathcal{L}_\rho\left(\boldsymbol{A}^{(n)}, \widetilde{\boldsymbol{A}}^{(n)}, \boldsymbol{\Lambda}^{(n)}\right) \tag{26}$$

Supposing $\boldsymbol{\Phi}^{(n)} = \frac{1}{\rho_n}\boldsymbol{\Lambda}^{(n)}$ and computing the partial derivative $\frac{\partial \mathcal{L}_\rho}{\partial \boldsymbol{W}} = 0$, we obtain the solution

$$\begin{aligned} \boldsymbol{A}^{(n)} = &\left[\boldsymbol{X}_{(n)}\boldsymbol{B}^{(n)} + \rho_n\left(\widetilde{\boldsymbol{A}}^{(n)} - \boldsymbol{\Phi}^{(n)}\right)\right] \\ &\left[\left(\boldsymbol{B}^{(n)}\right)^T\boldsymbol{B}^{(n)} + \rho_n\boldsymbol{I}\right]^{-1} \end{aligned} \tag{27}$$

where $\boldsymbol{I}$ is the identity matrix.

2) Updating the auxiliary variable $\widetilde{\boldsymbol{A}}^{(n)}$

$$\widetilde{\boldsymbol{A}}^{(n)} = \arg\min_{\widetilde{\boldsymbol{A}}^{(n)} \geqslant 0} \mathcal{L}_\rho\left(\boldsymbol{A}^{(n)}, \widetilde{\boldsymbol{A}}^{(n)}, \boldsymbol{\Lambda}^{(n)}\right) \tag{28}$$

Using the proximal operator, we obtain the following closed form solution

$$\begin{aligned} \widetilde{\boldsymbol{A}}^{(n)} &= \text{prox}_{\frac{\beta_n\|\cdot\|_1}{\rho_n}}\left(\boldsymbol{A}^{(n)} + \frac{1}{\rho_n}\boldsymbol{\Lambda}^{(n)}\right) \\ &= \max\left\{0, \boldsymbol{A}^{(n)} + \boldsymbol{\Phi}^{(n)} - \frac{\beta_n}{\rho_n}\boldsymbol{E}\right\}, \end{aligned} \tag{29}$$

where all the elements in $\boldsymbol{E} \in \mathbb{R}^{I_n \times R}$ equal to one.

3) Updating the Lagrange multiplier (dual variable) $\boldsymbol{\Lambda}^{(n)}$

$$\boldsymbol{\Lambda}^{(n)} = \boldsymbol{\Lambda}^{(n)} + \rho_n\left(\boldsymbol{A}^{(n)} - \widetilde{\boldsymbol{A}}^{(n)}\right). \tag{30}$$

With $\boldsymbol{\Phi}^{(n)} = \frac{1}{\rho_n}\boldsymbol{\Lambda}^{(n)}$, we have

$$\boldsymbol{\Phi}^{(n)} = \boldsymbol{\Phi}^{(n)} + \boldsymbol{A}^{(n)} - \widetilde{\boldsymbol{A}}^{(n)}. \tag{31}$$

**Algorithm 6** shows the implementation of ADMM.

**Algorithm 6:** ADMM for sparse NCP in (1)

**Input :** $\mathcal{X}$, $R$, $\boldsymbol{\beta}$
**Output:** $\boldsymbol{A}^{(n)}$, $n = 1, \ldots, N$
Initialize $\boldsymbol{A}^{(n)} \in \mathbb{R}^{I_n \times R}$, $n = 1, \ldots, N$, using nonnegative random numbers, and initialize $\boldsymbol{\Phi}^{(n)} \in \mathbb{R}^{I_n \times R}$ using zeros ;
**repeat**
  **for** $n = 1$ **to** $N$ **do**
    Make mode-$n$ unfolding of $\mathcal{X}$ as $\boldsymbol{X}_{(n)}$ and compute MTTKRP $\boldsymbol{X}_{(n)}\boldsymbol{B}^{(n)}$;
    Compute $\left(\boldsymbol{B}^{(n)}\right)^T\boldsymbol{B}^{(n)}$ based on (4);
    $\rho_n = \text{tr}\left[\left(\boldsymbol{B}^{(n)}\right)^T\boldsymbol{B}^{(n)}\right]/R$;
    **repeat**
      Update primal variable $\boldsymbol{A}^{(n)}$ using (27);
      Update auxiliary variable $\widetilde{\boldsymbol{A}}^{(n)}$ using (29);
      Update dual variable $\boldsymbol{\Phi}^{(n)}$ using (31);
    **until** *inner termination criterion is reached*;
    $\boldsymbol{A}^{(n)} = \widetilde{\boldsymbol{A}}^{(n)}$;
  **end**
**until** *outer termination criterion is reached*;
**return** $\boldsymbol{A}^{(n)}$, $n = 1, \ldots, N$.

The above sparse NCP methods in BCD framework have very good convergence properties. The subproblem of (2) with $l_1$-norm regularization item is convex [26]. The proximal algorithm especially guarantees (2) to be strongly convex and yields unique minimum [52]. Therefore, the sparse NCP in BCD framework will converge to stationary point [52]. The detailed discussion of the convergence properties of MU, ANQP, HALS, APG and ADMM can be found in [21], [22], [26], [28], [44]. We don't deeply analyze the convergence of above methods in this paper.

We summarize the time complexity of the above five algorithms in **Table 1**. We only count the multiplicative operations for mode-$n$ in one outer iteration. The main time cost of these algorithms is spent on the calculation of MTTKRP $\boldsymbol{X}_{(n)}\boldsymbol{B}^{(n)}$, which consists of two parts: khatri-rao product $\boldsymbol{B}^{(n)}$ and matrix product of $\boldsymbol{X}_{(n)}$ and $\boldsymbol{B}^{(n)}$. The computational complexity of $\boldsymbol{B}^{(n)}$ reaches $R\prod_{\tilde{n}=1,\tilde{n}\neq n}^{N} I_{\tilde{n}}$ and that of $\boldsymbol{X}_{(n)}\boldsymbol{B}^{(n)}$ reaches $R\prod_{n=1}^{N} I_n$. The item $\left(\boldsymbol{B}^{(n)}\right)^T\boldsymbol{B}^{(n)}$ is calculated in an efficient way by (4), whose complexity is $R^2\sum_{\tilde{n}=1,\tilde{n}\neq n}^{N} I_{\tilde{n}}$. As a summary, the complexity of the inner loop is listed in **Table 1**, and $\bar{K}$ is assumed as the average iteration number in the inner loop. In **Table 1**, we can find that the complexity of these algorithms is highly comparable to each other. It can be inferred that the time of convergence is highly related to the number of iterations.

TABLE 1
Computational Complexity of Subproblem (2)

| Method | $\boldsymbol{X}_{(n)}\boldsymbol{B}^{(n)}$ | $\left(\boldsymbol{B}^{(n)}\right)^T\boldsymbol{B}^{(n)}$ | Inner loop |
|---|---|---|---|
| MU | $R\prod_{\tilde{n}=1,\tilde{n}\neq n}^{N} I_{\tilde{n}}$ $+R\prod_{\tilde{n}=1}^{N} I_{\tilde{n}}$ | $R^2\sum_{\tilde{n}=1,\tilde{n}\neq n}^{N} I_{\tilde{n}}$ | $\bar{K}I_nR^2$ |
| ANQP-BPP | | | $\bar{K}\left(I_nR^2+R^3\right)$ |
| HALS | | | $\bar{K}I_nR^2$ |
| APG | | | $\bar{K}I_nR^2$ |
| ADMM | | | $\bar{K}\left(I_nR^2+R^3\right)$ |

## 4 Stopping Conditions

The optimization procedures for tensor decomposition are implemented by iterations. For NCP, a sequence of $\left\{\boldsymbol{A}_k^{(1)},\ldots,\boldsymbol{A}_k^{(N)}\right\}_{k=1}^{\infty}$ is produced at each iteration. It is necessary to terminate the iteration until some stopping condition is satisfied. Common stopping conditions include the following: predefined maximum number of iterations, predefined maximum running time, the change of objective function value, the change of relative error [21], [46].

### 4.1 Stopping Condition for Outer Loop

We terminate the outer loop according to the change of relative error during iteration. Relative error is related to data fitting. In the $k$th outer iteration, the relative error [21] of tensor decomposition is defined by

$$\text{RelErr}_k = \frac{\left\|\mathcal{X}-[\![\boldsymbol{A}_k^{(1)},\ldots,\boldsymbol{A}_k^{(N)}]\!]\right\|_F}{\|\mathcal{X}\|_F}. \tag{32}$$

Based on the relative error, we terminate the outer loop using the following stopping condition

$$|\text{RelErr}_{k-1}-\text{RelErr}_k|<\epsilon. \tag{33}$$

The threshold of $\epsilon$ can be set by a very small positive value, such as $1e-8$.

In addition, we also set a maximum running time for the outer loop.

### 4.2 Stopping Condition for Inner Loop

In the $l$th inner iteration, we define the relative residual of the $n$th factor matrix $\boldsymbol{A}^{(n)}$ as

$$r_l^{(n)} = \frac{\left\|\boldsymbol{A}_l^{(n)}-\boldsymbol{A}_{l-1}^{(n)}\right\|_F}{\left\|\boldsymbol{A}_l^{(n)}\right\|_F}. \tag{34}$$

We terminate the inner loop by the stopping condition of $r_l^{(n)}<\delta^{(n)}$, where $\delta^{(n)}$ is a dynamic positive threshold. If there is only one iteration in the inner loop, we update $\delta^{(n)}$ by $\delta^{(n)}=\delta^{(n)}/10$. We set the initial value by $\delta^{(n)}=0.01$. The ANQP-BPP is an exception, in which the inner loop is terminated according to the columns in feasible region [35].

Since we employ the inexact BCD framework, we also set a maximum number of inner iteration (`MAX_INNER_ITER`) to terminate the inner loop.

We summarize the stopping conditions for both of the outer and inner loop in **Algorithm 7**.

**Algorithm 7:** The stopping conditions

```
Set δ^(n) = 0.01, n = 1,...,N;
for k = 1,2,... do
    % The outer loop starts here.
    for n = 1 to N do
        for l = 1,2,... do
            % The inner loop starts here.
            A^(n) = argmin_{A^(n)≥0} F(A^(n));
            if r_l^(n) < δ^(n) or l ⩾MAX_INNER_ITER then
                Terminate the inner loop;
            end
            % The inner loop ends here.
        end
        if l == 1 then
            δ^(n) = δ^(n)/10;
        end
    end
    if |RelErr_{k-1} − RelErr_k| < ε then
        Terminate the outer loop.
    end
    % The outer loop ends here.
end
```

## 5 Experiments and Results

We carried out the experiments on synthetic tensors, real ongoing EEG tensor and real event-related potentials (ERP) tensor. We compared the abilities of sparse NCP methods to impose sparsity implemented by MU, ANQP-BPP, HALS, APG, and ADMM.

Many experimental parameters and settings will affect the performances of a sparse NCP method. Since our purpose in the experiments is only to test the ability of imposing sparsity, we fix the following settings for all methods.

- Initialization. For ANQP-BPP, HALS, APG, and ADMM, all factor matrices were initialized using nonnegative random numbers by MATLAB function `max(0,randn(`$I_n,R$`))`. Only MU was initialized by `max(0,randn(`$I_n,R$`))+0.1`. All initialized factors were scaled by $\boldsymbol{A}_0^{(n)}=\frac{\boldsymbol{A}_0^{(n)}}{\left\|\boldsymbol{A}_0^{(n)}\right\|_F}\times\sqrt[N]{\|\mathcal{X}\|_F}$.
- The factor updating order was fixed by $1,2,\ldots,N$.
- The maximum inner iteration `MAX_INNER_ITER` was fixed by 5.
- For ANQP-BPP and HALS method, the proximal regularization parameter $\alpha_n$ was fixed by 1e-4.

The $l_1$-norm regularization parameters of $\beta_n, n=1,\ldots,N$, in sparse NCP are the key elements to impose sparsity, which are the most important testing parameters in the experiments. In order to make it convenient to select and test the parameters, we kept $\beta_n, n=1,\ldots,N$, the same in all modes of the tensor. After selecting the $\beta_n$, we calculated and evaluated the sparsity level [11] of the factor matrices by

$$\text{Sparsity}_{\boldsymbol{A}^{(n)}} = \frac{\#\left\{\boldsymbol{A}_{i,r}^{(n)}<T_s\right\}}{I_n\times R}, \tag{35}$$

where $T_s$ is a small positive number and $\#\{\cdot\}$ denotes the number of elements that are smaller than the threshold $T_s$

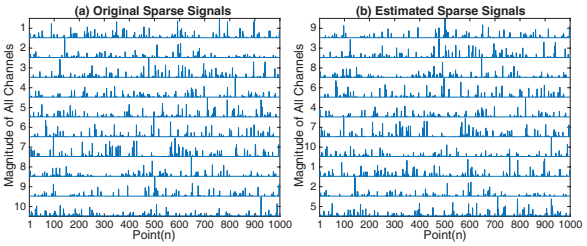



Fig. 1. Sparse and nonnegative signals used in synthetic tensor. (a) shows the original ten channels of signals. (b) shows the estimated ten channels of signals from the synthetic tensor $\mathcal{X}_{\text{SYN1}}$ by sparse NCP based on ANQP-BPP method with $\beta_n = 3$. The PNSR is 85.5251 according to (36).

in factor matrix $\boldsymbol{A}^{(n)}$.

In the synthetic tensor experiments, we used prior sparse matrices to construct the data. After decomposition, the accurary of the recovered sparse signals should be evaluated. Let $\boldsymbol{S}^{(n)} = [\boldsymbol{s}_1, \ldots, \boldsymbol{s}_R] \in \mathbb{R}^{L \times R}$ denote the mode-$n$ prior sparse matrix, where $R$ is the real number of components and $L$ is the length of a component. Let $\boldsymbol{T}^{(n)} = [\boldsymbol{t}_1, \ldots, \boldsymbol{t}_{\tilde{R}}] \in \mathbb{R}^{L \times \tilde{R}}$ represent the mode-$n$ estimated sparse matrix, in which $\tilde{R}$ is the estimated number of nonzero components. We evaluate the accuracy of the estimated matrix $\boldsymbol{T}^{(n)}$ compared with original sparse signals $\boldsymbol{S}^{(n)}$ by Peak Signal-to-Noise-Ratio (PSNR, see Chapter 3 in [1])

$$\text{PSNR} = \frac{1}{\tilde{R}} \sum_{r=1}^{\tilde{R}} 10\log_{10} \left( \frac{L}{\left\| \hat{\boldsymbol{t}}_r - \hat{\boldsymbol{s}}_c \right\|_2^2} \right), \qquad (36)$$

where $\hat{\boldsymbol{t}}_r$ is the $r$th normalized estimated sparse signal, and $\hat{\boldsymbol{s}}_c$ is the normalized reference sparse signal. $\hat{\boldsymbol{s}}_c$ comes from $\boldsymbol{S}^{(n)}$, which has the highest correlation coefficient with $\hat{\boldsymbol{t}}_r$.

All the experiments were conducted on 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 [48], [53], [54].

### 5.1 Synthetic Tensor Data

#### 5.1.1 *Size* $1000 \times 100 \times 100$ *with one sparse factor*

In this experiment, we constructed a synthetic third-order tensor by 10 channels of simulated sparse and nonnegative signals, as shown in Fig. 1(a). The signals come from the file `VSparse_rand_10.mat` in NMFLAB [55]. There are 1000 points in each channel, so the sparse signal matrix is $\boldsymbol{S}^{(1)} = [\boldsymbol{s}_1, \ldots, \boldsymbol{s}_{10}] \in \mathbb{R}^{1000 \times 10}$. Two uniformly distributed random matrices $\boldsymbol{A}^{(2)}, \boldsymbol{A}^{(3)} \in \mathbb{R}^{100 \times 10}$ were employed as mixing matrices, which were generated by `rand` function in MATLAB. Afterwards, we synthesized a third-order tensor by $\mathcal{X}_{\text{SYN1}} = [\![\boldsymbol{S}^{(1)}, \boldsymbol{A}^{(2)}, \boldsymbol{A}^{(3)}]\!] \in \mathbb{R}^{1000 \times 100 \times 100}$. Next, nonnegative noise was added to the tensor with SNR of 40dB, which was generated by MATLAB code `max(0,randn(size(`$\mathcal{X}$`)))`.

For all sparse NCP methods, we set $\epsilon = 1e-8$ as the threshold of outer stopping condition in (33). We set $T_s = 1e-3$ in (35). The maximum running time was set by 180 seconds. We selected a larger value of 20 as the number of components for tensor decomposition[2]. The reason is that we intend to recover the 10 channels of true signal just by imposing sparse regularization during decomposition, even though the exact optimal number of components is unknown. We selected values of $\beta_n = 0, 0.1, 0.5, 1, 2, 3$ for all the optimization methods to evaluate their abilities to impose sparsity. The selection of sparse regularization parameters depends on the tensor data. After tensor decomposition, the values of objective function value (Obj), relative error (RelErr), running time in second, iteration number (Iter), the number of nonzero components (NNC), sparsity level (Spars) and PSNR of the estimated signal factor matrix were recorded as the performance evaluation criteria. For all optimization methods with each $\beta_n$, the sparse NCP was run 30 times, and the average values of all criteria were computed. The results are shown in **Table 2**. We plot the objective function values of all sparse NCP methods with $\beta_n = 0, 0.1, 1, 3$ within the first 15 seconds as shown in **Fig. 2**.

TABLE 2
Comparison of Sparse NCPs on $\mathcal{X}_{\text{SYN1}} \in \mathbb{R}^{1000 \times 100 \times 100}$

| Method | $\beta_n$ | Obj | RelErr | Time | Iter | NNC | Spars | PSNR |
|---|---|---|---|---|---|---|---|---|
| MU | 0 | 9.76e+01 | 0.0082 | 107.9 | 2720.2 | 20.00 | 0.340 | 73.63 |
| | 0.1 | 5.33e+02 | 0.0083 | 78.6 | 1970.0 | 16.17 | 0.431 | 73.57 |
| | 0.5 | 1.65e+03 | 0.0084 | 57.5 | 1449.9 | 12.90 | 0.550 | 79.02 |
| | 1 | 2.47e+03 | 0.0084 | 74.6 | 1889.9 | 11.17 | 0.694 | 81.44 |
| | 2 | 4.17e+03 | 0.0084 | 86.0 | 2181.5 | 10.37 | 0.863 | 82.50 |
| | 3 | 5.91e+03 | 0.0085 | 87.3 | 2212.6 | **10.10** | **0.905** | **83.69** |
| ANQP BPP | 0 | 9.72e+01 | 0.0082 | 8.1 | 170.9 | 20.00 | 0.400 | 69.74 |
| | 0.1 | 5.45e+02 | 0.0083 | 8.9 | 222.8 | 10.20 | 0.548 | 83.21 |
| | 0.5 | 1.37e+03 | 0.0083 | 85.5 | 2178.0 | 10.10 | 0.618 | 84.26 |
| | 1 | 2.16e+03 | 0.0083 | 110.6 | 2811.1 | 10.00 | 0.823 | 84.49 |
| | 2 | 4.00e+03 | 0.0084 | 96.1 | 2435.5 | **10.00** | **0.908** | **85.13** |
| | 3 | 5.87e+03 | 0.0085 | 83.5 | 2116.2 | **10.00** | **0.912** | **85.39** |
| HALS | 0 | 9.74e+01 | 0.0082 | 18.8 | 458.8 | 20.00 | 0.452 | 79.12 |
| | 0.1 | 6.43e+02 | 0.0083 | 11.7 | 287.0 | 15.70 | 0.504 | 79.88 |
| | 0.5 | 1.85e+03 | 0.0084 | 38.2 | 943.4 | 11.70 | 0.634 | 83.27 |
| | 1 | 2.28e+03 | 0.0084 | 101.1 | 2516.1 | 10.20 | 0.819 | 84.38 |
| | 2 | 4.00e+03 | 0.0084 | 95.9 | 2395.9 | **10.00** | **0.913** | **85.10** |
| | 3 | 5.89e+03 | 0.0085 | 80.0 | 2001.8 | **10.00** | **0.917** | **85.42** |
| APG | 0 | 9.73e+01 | 0.0082 | 18.0 | 446.3 | 20.00 | 0.378 | 71.06 |
| | 0.1 | 5.52e+02 | 0.0083 | 11.2 | 278.0 | 11.87 | 0.519 | 79.58 |
| | 0.5 | 1.39e+03 | 0.0083 | 83.1 | 2083.0 | 10.03 | 0.621 | 84.13 |
| | 1 | 2.22e+03 | 0.0083 | 94.1 | 2353.1 | 10.00 | 0.823 | 84.27 |
| | 2 | 4.07e+03 | 0.0084 | 82.0 | 2056.3 | **10.00** | **0.908** | **84.89** |
| | 3 | 5.95e+03 | 0.0084 | 66.1 | 1663.3 | **10.00** | **0.906** | **84.83** |
| ADMM | 0 | 9.76e+01 | 0.0082 | 88.8 | 2219.7 | 20.00 | 0.358 | 73.85 |
| | 0.1 | 6.05e+02 | 0.0083 | 20.4 | 511.3 | 18.60 | 0.505 | 66.19 |
| | 0.5 | 1.75e+03 | 0.0084 | 50.9 | 1279.1 | 11.57 | 0.623 | 80.24 |
| | 1 | 2.45e+03 | 0.0084 | 82.7 | 2085.3 | 10.33 | 0.786 | 83.99 |
| | 2 | 4.02e+03 | 0.0084 | 93.9 | 2378.7 | **10.00** | **0.913** | **85.22** |
| | 3 | 5.89e+03 | 0.0085 | 80.0 | 2030.2 | **10.00** | **0.916** | **85.50** |

Spars = Sparsity level; Ground truth value: Spars=0.9.
NNC = Number of nonzero components.

#### 5.1.2 *Size* $1000 \times 100 \times 100 \times 5$ *with one sparse factor*

Using the sparse signals in `VSparse_rand_10.mat` again, we synthesized a fourth-order tensor $\mathcal{X}_{\text{SNY2}} = [\![\boldsymbol{S}^{(1)}, \boldsymbol{A}^{(2)}, \boldsymbol{A}^{(3)}, \boldsymbol{A}^{(4)}]\!] \in \mathbb{R}^{1000 \times 100 \times 100 \times 5}$, in which $\boldsymbol{A}^{(2)}, \boldsymbol{A}^{(3)} \in \mathbb{R}^{100 \times 10}$ and $\boldsymbol{A}^{(4)} \in \mathbb{R}^{5 \times 10}$ are uniformly

2. Since 10 channels of signals are mixed in the tensor, naturally, 10 should be selected as the optimal component number. The number of components might also be estimated by some classical methods, such as DIFFIT [56]. However, we selected 20 in order to test the performances of sparse regularization.

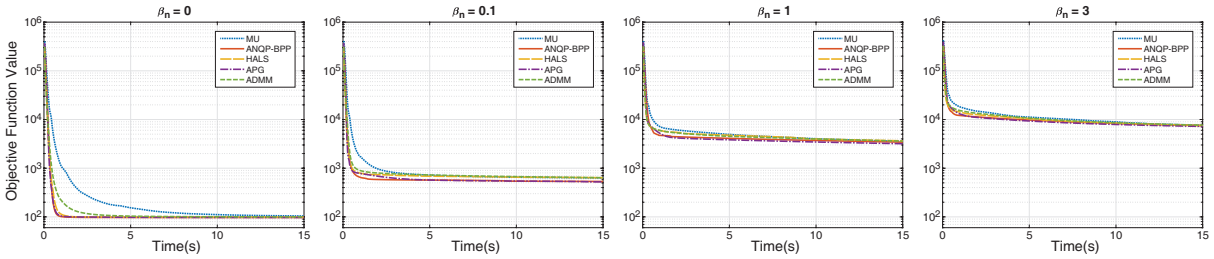



Fig. 2. The Objective Function Value Curves of Sparse NCPs on Third-order Synthetic Tensor $\mathcal{X}_{\text{SYN1}}$ With Fixed Time Limit of 30s.

TABLE 3
Comparison of Sparse NCPs on $\mathcal{X}_{\text{SYN2}} \in \mathbb{R}^{1000\times100\times100\times5}$

| Method | $\beta_n$ | Obj | RelErr | Time | Iter | NNC | Spars | PSNR |
|---|---|---|---|---|---|---|---|---|
| MU | 0 | 1.58e+02 | 0.0084 | 712.0 | 352.5 | 20.00 | 0.210 | 66.64 |
| | 1 | 6.30e+03 | 0.0089 | 862.2 | 427.4 | 12.90 | 0.693 | 75.63 |
| | 2 | 8.62e+03 | 0.0091 | 868.9 | 430.5 | 11.53 | 0.788 | 79.04 |
| | 3 | 1.06e+04 | 0.0093 | 836.4 | 411.2 | 11.10 | 0.847 | 81.37 |
| | 4 | 1.26e+04 | 0.0094 | 822.8 | 403.5 | 11.23 | 0.864 | 81.00 |
| | 5 | 1.39e+04 | 0.0094 | 838.4 | 408.2 | 10.77 | **0.882** | 82.32 |
| ANQP BPP | 0 | 1.51e+02 | 0.0082 | 69.2 | 33.5 | 20.00 | 0.394 | 74.66 |
| | 1 | 6.90e+03 | 0.0088 | 548.3 | 275.5 | 10.00 | 0.787 | 84.76 |
| | 2 | 8.75e+03 | 0.0089 | 803.2 | 403.1 | 10.00 | 0.810 | 87.25 |
| | 3 | 9.90e+03 | 0.0089 | 907.4 | 456.2 | 10.00 | 0.837 | 88.23 |
| | 4 | 1.15e+04 | 0.0089 | 959.3 | 480.3 | **10.00** | **0.852** | **88.67** |
| | 5 | 1.28e+04 | 0.0089 | 991.6 | 496.0 | **10.00** | **0.855** | **88.79** |
| HALS | 0 | 1.51e+02 | 0.0082 | 103.0 | 51.4 | 20.00 | 0.474 | 79.34 |
| | 1 | 8.16e+03 | 0.0090 | 394.8 | 198.3 | 11.63 | 0.800 | 80.59 |
| | 2 | 9.64e+03 | 0.0090 | 656.4 | 327.9 | 10.63 | 0.824 | 84.05 |
| | 3 | 1.03e+04 | 0.0090 | 868.8 | 436.8 | 10.23 | 0.845 | 87.14 |
| | 4 | 1.20e+04 | 0.0090 | 859.2 | 431.9 | **10.27** | **0.861** | **86.93** |
| | 5 | 1.31e+04 | 0.0090 | 947.1 | 474.7 | **10.17** | **0.864** | **87.66** |
| APG | 0 | 1.56e+02 | 0.0084 | 140.9 | 70.6 | 20.00 | 0.122 | 63.19 |
| | 1 | 6.67e+03 | 0.0088 | 354.5 | 177.9 | 10.37 | 0.764 | 86.72 |
| | 2 | 8.04e+03 | 0.0088 | 681.6 | 339.0 | 10.17 | 0.791 | 88.57 |
| | 3 | 9.72e+03 | 0.0089 | 781.7 | 391.2 | 10.07 | 0.828 | 89.26 |
| | 4 | 1.21e+04 | 0.0090 | 792.5 | 394.9 | **10.27** | **0.849** | **88.98** |
| | 5 | 1.33e+04 | 0.0089 | 871.7 | 433.8 | **10.00** | **0.856** | **89.33** |
| ADMM | 0 | 1.59e+02 | 0.0084 | 254.5 | 124.8 | 20.00 | 0.151 | 65.74 |
| | 1 | 7.41e+03 | 0.0089 | 357.7 | 179.4 | 12.33 | 0.782 | 80.64 |
| | 2 | 8.42e+03 | 0.0089 | 801.9 | 401.8 | 11.10 | 0.819 | 87.93 |
| | 3 | 1.00e+04 | 0.0089 | 855.2 | 427.6 | 10.63 | 0.854 | 88.22 |
| | 4 | 1.17e+04 | 0.0090 | 908.3 | 455.9 | 10.57 | 0.869 | 88.45 |
| | 5 | 1.34e+04 | 0.0090 | 876.7 | 438.8 | 10.63 | **0.871** | **87.78** |

Spars = Sparsity level; Ground truth value: Spars=0.9.
NNC = Number of nonzero components.

distributed random matrices. Noise with SNR of 40dB was added.

We set the outer stopping condition by $\epsilon = 1e-6$ and the maximum running time by 1200 seconds. Other settings are the same as those in previous test. The values of $\beta_n = 0, 1, 2, 3, 4, 5$ were tested for all sparse NCP methods. Afterwards, the average evaluation values of 30 runs were computed as shown in **Table 3**.

From **Table 2** and **Table 3**, it can be found that all methods are able to impose sparsity with proper sparse regularization parameter $\beta_n$. When $\beta_n$ increases, the sparsity level of the mode-1 factor matrix also increases. After properly tuning the sparse regularization parameter $\beta_n$, weak components will be removed (set to 0), weak elements in strong components will be prohibited, and the true 10 channels of sparse signals will be recovered.

When $\beta_n$ is increased to a proper value, the PSNR is also increased significantly. In the above two experiments of 5.1.1 and 5.1.2, the value of PSNR around 85 means very good recovery of original sparse components. In **Table 2** and **Table 3**, it is clear to see that ANQP-BPP, HALS, APG, and ADMM have higher PSNR with larger sparse regularization parameters, for example, $\beta_n = 2, 3$ for $\mathcal{X}_{\text{SYN1}}$ and $\beta_n = 4, 5$ for $\mathcal{X}_{\text{SYN2}}$. This means that these four methods recover the 10 channels of sparse signals more precisely. One of the recovered sparse signal matrix from $\mathcal{X}_{\text{SYN1}}$ by ANQP-BPP is shown in Fig. 1(b).

For the synthetic data, the objective function values and relative errors are very similar at the same $\beta_n$. The convergence speed can be concluded from **Table 2**, **Table 3** and **Fig. 2**. MU performs slowly compared with other methods. ADMM performs slowly with $\beta_n = 0$, but it becomes fast with $\beta_n > 0$. All ANQP-BPP, HALS and APG methods perform very well. It can also be concluded from **Table 2** and **Table 3** that the running time is highly related to the number of outer iterations.

#### 5.1.3 *Size* $500 \times 500 \times 500$ *with two sparse factors*

For this third-order tensor, the factor matrices were generated using the following codes.

| Factor | Code | Zeros |
|---|---|---|
| $\boldsymbol{S}^{(1)} \in \mathbb{R}^{500\times100}$ | `max(0,rand(500,100)*10-9);` | 90% |
| $\boldsymbol{S}^{(2)} \in \mathbb{R}^{500\times100}$ | `max(0,rand(500,100)*2-1);` | 50% |
| $\boldsymbol{A}^{(3)} \in \mathbb{R}^{500\times100}$ | `rand(500,100);` | 0% |

Afterwards, a third-order tensor was synthesized by $\mathcal{X}_{\text{SNY3}} = [\![\boldsymbol{S}^{(1)}, \boldsymbol{S}^{(2)}, \boldsymbol{A}^{(3)}]\!]$, whose true number of components was 100. Noise with SNR of 40dB was added.

We set the outer stopping condition by $\epsilon = 1e-6$ and the maximum running time by 600 seconds. 200 was selected as the initial number of components. The average performances of all sparse NCP methods after 30 runs were computed. We only show running time in second, iteration number (Iter), number of nonzero components (NNC) and the sparsity level (Spars) of all estimated factors in **Table 4**.

#### 5.1.4 *Size* $100 \times 100 \times 100 \times 100$ *with three sparse factors*

For this fourth-order tensor, the factor matrices were generated using the following codes.

| Factor | Code | Zeros |
|---|---|---|
| $\boldsymbol{S}^{(1)} \in \mathbb{R}^{100\times20}$ | `max(0,rand(100,20)*10-9);` | 90% |
| $\boldsymbol{S}^{(2)} \in \mathbb{R}^{100\times20}$ | `max(0,rand(100,20)*4-3);` | 75% |
| $\boldsymbol{S}^{(3)} \in \mathbb{R}^{100\times20}$ | `max(0,rand(100,20)*2-1);` | 50% |
| $\boldsymbol{A}^{(4)} \in \mathbb{R}^{100\times20}$ | `rand(100,20);` | 0% |

TABLE 4
Comparison of Sparse NCPs on $\mathcal{X}_{\text{SYN3}} \in \mathbb{R}^{500\times500\times500}$

| Method | $\beta_n$ | Time | Iter | NNC | $\text{Spars}_1$ | $\text{Spars}_2$ | $\text{Spars}_3$ |
|---|---|---|---|---|---|---|---|
| MU | 0 | 600 | 356.8 | 200.00 | 0.330 | 0.508 | 0.011 |
| | 0.1 | 595.3 | 353.6 | 199.63 | 0.374 | 0.506 | 0.012 |
| | 0.5 | 600 | 332.9 | 149.07 | 0.524 | 0.638 | 0.265 |
| | 1 | 600 | 345.1 | 146.27 | 0.535 | 0.635 | 0.280 |
| | 3 | 595.2 | 345.1 | 127.97 | 0.626 | 0.672 | 0.369 |
| | 5 | 593.7 | 340.5 | 118.03 | 0.672 | 0.686 | 0.418 |
| ANQP BPP | 0 | 42.0 | 19.0 | 200.00 | 0.300 | 0.512 | 0.027 |
| | 0.1 | 51.6 | 25.5 | 100.93 | 0.705 | 0.749 | 0.502 |
| | 0.5 | 49.9 | 25.6 | 100.27 | 0.828 | 0.750 | 0.505 |
| | 1 | 173.6 | 97.6 | **100.00** | **0.872** | **0.749** | **0.506** |
| | 3 | 241.3 | 137.5 | **100.00** | **0.933** | **0.725** | **0.505** |
| | 5 | 282.7 | 158.9 | **100.00** | **0.945** | **0.682** | **0.505** |
| HALS | 0 | 56.9 | 32.4 | 200.00 | 0.438 | 0.506 | 0.014 |
| | 0.1 | 80.5 | 46.3 | 159.07 | 0.615 | 0.623 | 0.218 |
| | 0.5 | 114.5 | 65.6 | 123.13 | 0.746 | 0.702 | 0.393 |
| | 1 | 131.1 | 75.2 | 110.47 | 0.790 | 0.728 | 0.454 |
| | 3 | 305.1 | 174.9 | **100.23** | **0.863** | **0.735** | **0.503** |
| | 5 | 345.3 | 195.4 | **100.03** | **0.891** | **0.706** | **0.504** |
| APG | 0 | 121.7 | 72.7 | 200.00 | 0.156 | 0.495 | 0.342 |
| | 0.1 | 122.0 | 72.1 | 200.00 | 0.566 | 0.552 | 0.372 |
| | 0.5 | 140.4 | 83.2 | 151.77 | 0.706 | 0.621 | 0.444 |
| | 1 | 145.2 | 85.2 | 107.80 | 0.719 | 0.695 | 0.466 |
| | 3 | 288.9 | 171.3 | **100.07** | **0.766** | **0.722** | **0.504** |
| | 5 | 440.1 | 263.5 | **100.00** | **0.823** | **0.699** | **0.505** |
| ADMM | 0 | 141.5 | 80.7 | 200.00 | 0.344 | 0.482 | 0.086 |
| | 0.1 | 125.8 | 72.7 | 195.77 | 0.613 | 0.471 | 0.078 |
| | 0.5 | 138.7 | 79.6 | 128.93 | 0.821 | 0.522 | 0.064 |
| | 1 | 125.3 | 73.7 | 110.53 | 0.875 | 0.547 | 0.054 |
| | 3 | 165.9 | 96.7 | 100.37 | 0.939 | 0.646 | 0.138 |
| | 5 | 247.3 | 144.0 | **100.00** | **0.948** | 0.690 | 0.359 |

Ground truth levels: $\text{Spars}_1$=0.95, $\text{Spars}_2$=0.75 and $\text{Spars}_3$=0.5.
$\text{Spars}_n$ = Sparsity level of the mode-$n$ estimated factor.
NNC = Number of nonzero components.

TABLE 5
Comparison of Sparse NCPs on $\mathcal{X}_{\text{SYN4}} \in \mathbb{R}^{100\times100\times100\times100}$

| Method | $\beta_n$ | Time | Iter | NNC | $\text{Spars}_1$ | $\text{Spars}_2$ | $\text{Spars}_3$ | $\text{Spars}_4$ |
|---|---|---|---|---|---|---|---|---|
| MU | 0 | 155.0 | 191.5 | 40.00 | 0.177 | 0.648 | 0.438 | 0.012 |
| | 0.1 | 87.6 | 106.7 | 31.10 | 0.668 | 0.769 | 0.609 | 0.239 |
| | 0.5 | 122.2 | 144.5 | 23.40 | 0.940 | 0.829 | 0.708 | 0.427 |
| | 1 | 114.9 | 135.0 | 21.40 | 0.942 | 0.862 | 0.731 | 0.472 |
| | 2 | 85.9 | 101.7 | **20.50** | **0.935** | **0.863** | **0.735** | **0.492** |
| | 3 | 61.9 | 73.2 | 19.43 | 0.919 | 0.861 | 0.737 | 0.519 |
| ANQP BPP | 0 | 10.5 | 12.6 | 40.00 | 0.580 | 0.613 | 0.434 | 0.129 |
| | 0.1 | 11.3 | 13.8 | **20.00** | **0.831** | **0.857** | **0.751** | **0.504** |
| | 0.5 | 95.0 | 117.5 | **19.87** | **0.943** | **0.864** | **0.751** | **0.507** |
| | 1 | 102.5 | 126.6 | **19.63** | **0.943** | **0.871** | **0.749** | **0.513** |
| | 2 | 90.1 | 111.2 | 19.23 | 0.938 | 0.868 | 0.748 | 0.523 |
| | 3 | 66.9 | 82.2 | 18.27 | 0.921 | 0.852 | 0.739 | 0.547 |
| HALS | 0 | 24.5 | 30.2 | 40.00 | 0.724 | 0.700 | 0.473 | 0.073 |
| | 0.1 | 32.3 | 39.9 | 24.13 | 0.817 | 0.825 | 0.699 | 0.405 |
| | 0.5 | 84.9 | 102.7 | **20.40** | **0.945** | **0.861** | **0.743** | **0.496** |
| | 1 | 95.1 | 116.6 | **19.73** | **0.944** | **0.871** | **0.748** | **0.510** |
| | 2 | 87.9 | 108.5 | 19.33 | 0.941 | 0.871 | 0.748 | 0.520 |
| | 3 | 66.0 | 81.6 | 18.40 | 0.923 | 0.856 | 0.741 | 0.543 |
| APG | 0 | 91.3 | 113.1 | 40.00 | 0.212 | 0.516 | 0.351 | 0.213 |
| | 0.1 | 51.4 | 63.6 | 20.30 | 0.836 | 0.710 | 0.564 | 0.369 |
| | 0.5 | 81.0 | 100.4 | **19.93** | **0.949** | **0.851** | **0.736** | **0.496** |
| | 1 | 76.9 | 95.1 | 19.30 | **0.941** | **0.867** | **0.748** | **0.521** |
| | 2 | 73.7 | 91.1 | 18.67 | 0.930 | 0.858 | 0.745 | 0.535 |
| | 3 | 63.8 | 78.9 | 17.03 | 0.903 | 0.835 | 0.731 | 0.577 |
| ADMM | 0 | 87.9 | 108.5 | 40.00 | 0.145 | 0.521 | 0.385 | 0.210 |
| | 0.1 | 78.0 | 96.2 | 26.50 | 0.814 | 0.602 | 0.465 | 0.284 |
| | 0.5 | 77.4 | 95.7 | 20.60 | 0.943 | 0.728 | 0.542 | 0.358 |
| | 1 | 67.5 | 83.4 | 19.83 | 0.941 | 0.811 | 0.598 | 0.412 |
| | 2 | 57.8 | 71.5 | 19.30 | **0.929** | **0.840** | 0.671 | 0.469 |
| | 3 | 43.3 | 53.5 | 17.60 | 0.896 | 0.824 | 0.701 | 0.527 |

Ground truth: $\text{Spars}_1$=0.95, $\text{Spars}_2$=0.875, $\text{Spars}_3$=0.75 and $\text{Spars}_4$=0.5.
$\text{Spars}_n$ = Sparsity level of the mode-$n$ estimated factor.
NNC = Number of nonzero components.

Afterwards, a fourth-order tensor was synthesized by $\mathcal{X}_{\text{SNY4}} = [\![\boldsymbol{S}^{(1)}, \boldsymbol{S}^{(2)}, \boldsymbol{S}^{(3)}, \boldsymbol{A}^{(4)}]\!]$, whose true number of components was 20. Noise with SNR of 40dB was added.

We set the outer stopping condition by $\epsilon = 1e-6$ and the maximum running time by 240 seconds. 40 was selected as the initial number of components. The average performances after 30 runs were recorded in **Table 5**.

**Table 4** and **Table 5** show that all methods are able to impose sparsity to all factors matrices. ANQP-BPP, HALS, and APG methods performs very well to extract the true number of sparse components, which is 100 for $\mathcal{X}_{\text{SYN3}}$ and 20 for $\mathcal{X}_{\text{SYN4}}$. Surprisingly, the sparsity levels of all extracted factor matrix by ANQP-BPP, HALS, and APG methods are also very close to the ground-truth[3] values with some $\beta_n$. MU and ADMM don't always work well to reach the ground-truth factor sparsity levels. On the other hand, MU shows slower convergence compared with other methods.

### 5.2 Ongoing EEG Tensor Data

In this experiment, we used a real-world third-order ongoing EEG tensor. The data come from one subject elicited by natural continuous music stimulus [11]. The size of this tensor is channel × frequency × time = 64 × 146 × 510. The 64 channel points represent 64 electrodes on the scalp, the 146 frequency points represent the spectrum in 1-30Hz, and the 510 time points represent the duration of stimulus of about 8.5 minutes. The spectra from EEG tensor are usually sparse. We wish to recover the sparse spectral components by sparse regularization.

We set $\epsilon = 1e-8$ in (33) and $T_s = 1e-6$ in (35). The maximum running time was set by 120 seconds. The initial number of components were set by 40 according to previous studies [4], [11]. The values of $\beta_n = 0, 1e5, 5e5, 10e5, 15e5, 20e5$ were tested for all methods. All methods were run 30 times. The averages of performance criteria are recorded in **Table 6**. We also plot the objective function values of all sparse NCP methods with $\beta_n = 0, 1e5, 10e5, 20e5$ within the first 10 seconds as shown in **Fig. 3**. The results show that all methods are effective to impose sparsity with $\beta_n$. ANQP-BPP, HALS, APG, and ADMM have very good computational efficiency.

We selected three groups of extracted components using ANQP-BPP method with $\beta_n = 0, 5e5, 10e5$ respectively as shown in **Fig. 4**. These three groups show the same brain activity. It is very clear to see that the spectra become sparser and sparser when sparse regularization parameter is increased. With $\beta_n = 5e5, 10e5$, more and more redundant elements are removed in the spectra and only the most prominent frequency band is retained. **Fig. 4** demonstrates that our methods are effective to extract meaningful sparse components that are related to some brain activities.

3. Since we use double number of true sparse components as the initial number of tensor decomposition, the ground truth sparsity of the factor matrix is computed by $(x\% + 1)/2$. $x\%$ is the percentage of zeros in a simulated matrix.

TABLE 6
Comparison of Sparse NCPs on Ongoing EEG Tensor $\mathcal{X}_{\text{EEG}} \in \mathbb{R}^{64\times146\times510}$

| Method | $\beta_n \times 10^5$ | Obj | RelErr | Time | Iter | NNC | $\text{Spars}_1$ | $\text{Spars}_2$ | $\text{Spars}_3$ |
|---|---|---|---|---|---|---|---|---|---|
| MU | 0 | 2.99e+10 | 0.2853 | 106.3 | 2790.8 | 40.00 | 0.077 | 0.380 | 0.115 |
| | 1 | 3.34e+10 | 0.2857 | 101.7 | 2708.2 | 40.00 | 0.104 | 0.457 | 0.137 |
| | 5 | 4.58e+10 | 0.2928 | 89.7 | 1269.9 | 39.77 | 0.191 | 0.761 | 0.272 |
| | 10 | 5.87e+10 | 0.3132 | 103.0 | 1034.6 | 31.70 | 0.364 | 0.860 | 0.493 |
| | 15 | 6.79e+10 | 0.3339 | 94.5 | 946.6 | 23.97 | 0.525 | 0.901 | 0.634 |
| | 20 | 7.48e+10 | 0.3505 | 81.5 | 883.3 | **18.10** | **0.635** | **0.922** | **0.720** |
| ANQP BPP | 0 | 2.98e+10 | 0.2847 | 19.2 | 444.4 | 40.00 | 0.085 | 0.381 | 0.104 |
| | 1 | 3.35e+10 | 0.2858 | 34.9 | 824.1 | 39.73 | 0.124 | 0.475 | 0.129 |
| | 5 | 4.59e+10 | 0.2957 | 21.4 | 540.4 | 37.30 | 0.241 | 0.779 | 0.292 |
| | 10 | 5.86e+10 | 0.3145 | 17.6 | 483.9 | 30.53 | 0.383 | 0.872 | 0.495 |
| | 15 | 6.79e+10 | 0.3342 | 15.5 | 448.4 | 23.57 | 0.530 | 0.904 | 0.633 |
| | 20 | 7.46e+10 | 0.3515 | 10.5 | 310.6 | **17.57** | **0.638** | **0.927** | **0.723** |
| HALS | 0 | 2.98e+10 | 0.2848 | 14.2 | 466.9 | 40.00 | 0.086 | 0.376 | 0.103 |
| | 1 | 3.34e+10 | 0.2852 | 18.6 | 601.6 | 39.97 | 0.119 | 0.469 | 0.122 |
| | 5 | 4.58e+10 | 0.2952 | 15.1 | 488.4 | 37.47 | 0.238 | 0.778 | 0.293 |
| | 10 | 5.86e+10 | 0.3158 | 13.1 | 424.2 | 29.83 | 0.399 | 0.872 | 0.503 |
| | 15 | 6.79e+10 | 0.3343 | 15.5 | 456.7 | 23.40 | 0.533 | 0.906 | 0.631 |
| | 20 | 7.47e+10 | 0.3514 | 10.8 | 355.7 | **17.53** | **0.637** | **0.927** | **0.720** |
| APG | 0 | 2.99e+10 | 0.2851 | 9.5 | 301.4 | 40.00 | 0.084 | 0.370 | 0.101 |
| | 1 | 3.34e+10 | 0.2857 | 21.4 | 710.9 | 39.80 | 0.132 | 0.475 | 0.127 |
| | 5 | 4.73e+10 | 0.3117 | 18.4 | 621.0 | 28.37 | 0.426 | 0.783 | 0.445 |
| | 10 | 5.92e+10 | 0.3378 | 19.5 | 655.6 | 19.63 | 0.598 | 0.865 | 0.631 |
| | 15 | 6.76e+10 | 0.3579 | 15.8 | 532.8 | 14.13 | 0.704 | 0.904 | 0.736 |
| | 20 | 7.42e+10 | 0.3681 | 12.9 | 436.7 | **12.20** | **0.745** | **0.924** | **0.781** |
| ADMM | 0 | 2.98e+10 | 0.2849 | 23.0 | 810.4 | 40.00 | 0.081 | 0.385 | 0.102 |
| | 1 | 3.34e+10 | 0.2853 | 20.2 | 685.2 | 39.93 | 0.124 | 0.460 | 0.124 |
| | 5 | 4.60e+10 | 0.2978 | 16.0 | 546.6 | 35.93 | 0.271 | 0.775 | 0.326 |
| | 10 | 5.84e+10 | 0.3218 | 15.3 | 523.7 | 26.23 | 0.468 | 0.860 | 0.547 |
| | 15 | 6.75e+10 | 0.3394 | 14.2 | 486.6 | 20.93 | 0.582 | 0.900 | 0.662 |
| | 20 | 7.43e+10 | 0.3552 | 9.9 | 339.5 | **15.93** | **0.669** | **0.925** | **0.738** |

$\text{Spars}_1$, $\text{Spars}_2$ and $\text{Spars}_3$ are the sparsity levels of spatial, spectral and temporal factor.
NNC = Number of nonzero components.

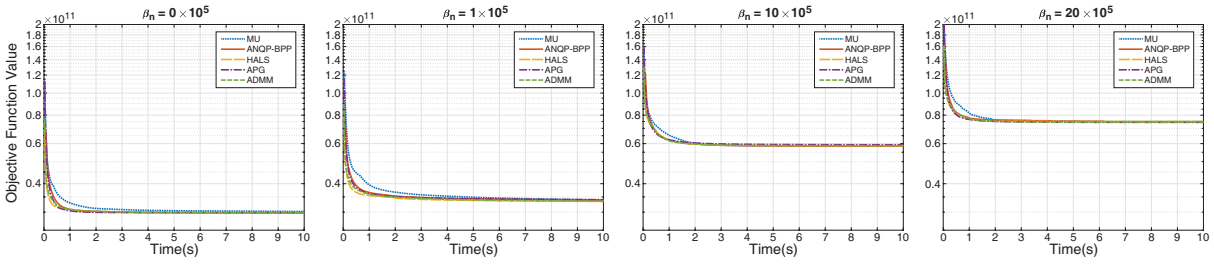



Fig. 3. The Objective Function Value Curves of Sparse NCPs on Third-order Ongoing EEG Tensor.

### 5.3 Event-Related Potentials Tensor Data

In this experiment, we utilized a preprocessed fourth-order event-related potentials (ERP) tensor[4]. The data size is channel × frequency × time × subject-group = 9 × 71 × 60 × 42. The 9 channel points denote the 9 electrodes on the scalp, the 71 frequency points show the spectrum within 1-15Hz, the 60 time points illustrate the temporal energy between 0-300ms, and the 42 subject-group points include 21 subjects with reading disability (RD) and 21 subjects with attention deficit (AD) [57].

In this experiment, the values of $\beta_n = 0, 10, 50, 100, 200, 300$ were tested for all methods. Other setting parameters were the same as those for the ongoing EEG. The average values of performance criteria after 30 runs are recorded in **Table 7**. We also recorded the objective function values of all methods with $\beta_n = 0, 10, 50, 100$ within the first 30 seconds in **Fig. 5**. According to the results, MU performs very slowly and often reach the time limit of 120s. MU is not sensitive to $\beta_n$ for imposing sparsity in this case, and has more nonzero components at the same $\beta_n$ value compared with other methods. If more nonzero components are retained, the objective function value will be smaller. Take the results of $\beta_n = 300$ for example. The number of nonzero components of MU is 31.10 and the objective function value is 1.51e+06. While for ANQP-BPP, the NNC is 15.7 and the Obj is 1.97e+06. This can be further seen from **Fig. 5**. ADMM performs slowly with small $\beta_n$ values (e.g., $0, 10$), but its performance can be improved with large $\beta_n$ values (e.g., $100, 200, 300$). As expected, all ANQP-BPP, HALS, and APG methods perform very well on both of the effectiveness of imposing sparsity and the

4. Data website:
http://www.escience.cn/people/cong/AdvancedSP_ERP.html

TABLE 7
Comparison of Sparse NCPs on ERP Tensor $\mathcal{X}_{\text{ERP}} \in \mathbb{R}^{9\times71\times60\times42}$

| Method | $\beta_n$ | Obj | RelErr | Time | Iter | NNC | $\text{Spars}_1$ | $\text{Spars}_2$ | $\text{Spars}_3$ | $\text{Spars}_4$ |
|---|---|---|---|---|---|---|---|---|---|---|
| MU | 0 | 4.92e+05 | 0.1132 | 120.0 | 3530.2 | 40.00 | 0.437 | 0.130 | 0.241 | 0.217 |
| | 10 | 5.47e+05 | 0.1135 | 120.0 | 3525.5 | 39.97 | 0.382 | 0.128 | 0.235 | 0.214 |
| | 50 | 6.80e+05 | 0.1151 | 120.0 | 3519.3 | 39.87 | 0.377 | 0.141 | 0.229 | 0.232 |
| | 100 | 8.46e+05 | 0.1163 | 120.0 | 3737.7 | 39.07 | 0.438 | 0.173 | 0.242 | 0.239 |
| | 200 | 1.19e+06 | 0.1245 | 113.8 | 3386.1 | 35.17 | 0.493 | 0.257 | 0.309 | 0.314 |
| | 300 | 1.51e+06 | 0.1346 | 92.3 | 2913.0 | 31.10 | 0.564 | 0.330 | 0.385 | 0.391 |
| ANQP BPP | 0 | 4.77e+05 | 0.1115 | 24.2 | 755.8 | 40.00 | 0.522 | 0.141 | 0.234 | 0.178 |
| | 10 | 6.74e+05 | 0.1239 | 31.9 | 1147.0 | 33.93 | 0.576 | 0.266 | 0.333 | 0.291 |
| | 50 | 8.57e+05 | 0.1314 | 34.5 | 1261.2 | 31.03 | 0.606 | 0.325 | 0.384 | 0.349 |
| | 100 | 1.07e+06 | 0.1413 | 32.7 | 1205.0 | 27.70 | 0.637 | 0.390 | 0.446 | 0.418 |
| | 200 | 1.49e+06 | 0.1636 | 23.6 | 883.9 | 21.00 | 0.699 | 0.531 | 0.568 | 0.560 |
| | 300 | 1.97e+06 | 0.1911 | 17.9 | 681.2 | **15.70** | **0.761** | **0.645** | **0.658** | **0.673** |
| HALS | 0 | 4.76e+05 | 0.1114 | 27.7 | 1019.7 | 40.00 | 0.525 | 0.139 | 0.234 | 0.176 |
| | 10 | 6.68e+05 | 0.1223 | 26.4 | 969.2 | 34.57 | 0.572 | 0.254 | 0.317 | 0.278 |
| | 50 | 8.41e+05 | 0.1286 | 32.0 | 1177.5 | 32.13 | 0.586 | 0.307 | 0.366 | 0.329 |
| | 100 | 1.05e+06 | 0.1381 | 30.0 | 1102.8 | 28.77 | 0.625 | 0.373 | 0.426 | 0.395 |
| | 200 | 1.50e+06 | 0.1632 | 22.5 | 828.6 | 21.10 | 0.696 | 0.530 | 0.565 | 0.556 |
| | 300 | 1.94e+06 | 0.1872 | 16.1 | 594.5 | **16.23** | **0.753** | **0.631** | **0.651** | **0.663** |
| APG | 0 | 4.77e+05 | 0.1115 | 19.9 | 704.5 | 40.00 | 0.504 | 0.132 | 0.246 | 0.183 |
| | 10 | 6.82e+05 | 0.1228 | 19.5 | 685.1 | 34.37 | 0.563 | 0.253 | 0.325 | 0.286 |
| | 50 | 1.08e+06 | 0.1494 | 21.7 | 768.5 | 24.93 | 0.655 | 0.446 | 0.502 | 0.472 |
| | 100 | 1.42e+06 | 0.1690 | 19.3 | 691.2 | 19.90 | 0.712 | 0.547 | 0.589 | 0.583 |
| | 200 | 1.97e+06 | 0.1999 | 17.2 | 617.9 | 14.07 | 0.775 | 0.676 | 0.697 | 0.705 |
| | 300 | 2.41e+06 | 0.2225 | 16.6 | 599.7 | **10.97** | **0.804** | **0.745** | **0.762** | **0.765** |
| ADMM | 0 | 4.79e+05 | 0.1117 | 53.4 | 2046.1 | 40.00 | 0.519 | 0.132 | 0.240 | 0.178 |
| | 10 | 6.20e+05 | 0.1200 | 56.4 | 2097.0 | 35.57 | 0.541 | 0.238 | 0.303 | 0.235 |
| | 50 | 9.72e+05 | 0.1435 | 37.5 | 1362.3 | 26.77 | 0.643 | 0.411 | 0.465 | 0.426 |
| | 100 | 1.28e+06 | 0.1604 | 24.9 | 922.7 | 21.87 | 0.688 | 0.512 | 0.554 | 0.538 |
| | 200 | 1.75e+06 | 0.1870 | 23.2 | 859.1 | 16.33 | 0.754 | 0.630 | 0.648 | 0.660 |
| | 300 | 2.29e+06 | 0.2164 | 18.7 | 694.2 | **11.73** | **0.792** | **0.730** | **0.747** | **0.750** |

$\text{Spars}_1$, $\text{Spars}_2$, $\text{Spars}_3$ and $\text{Spars}_4$ are the sparsity levels of spatial, spectral, temporal and subject-group factor matrix.
NNC = Number of nonzero components.

efficiency of computation.

## 6 Conclusion

In this paper, we investigated the nonnegative CANDECOMP/PARAFAC tensor decomposition with $l_1$-norm based sparse regularization (sparse NCP). We employed an inexact block coordinate descent scheme to design sparse NCP algorithm. The optimization methods of multiplicative update (MU), alternating nonnegative quadratic programming (ANQP), hierarchical alternating least squares (HALS), alternating proximal gradient (APG) and alternating direction method of multipliers (ADMM) in block coordinate descent framework were carefully investigated to solve sparse NCP. The experimental results on both synthetic and real-world tensor data demonstrate that our sparse NCP methods are able to successfully impose sparsity and extract meaningful sparse components. The methods of ANQP, HALS, and APG especially have outstanding performances. Our sparse NCP methods can be extended to many other multiway data applications. In future, it is interesting to investigate how to separately control the sparsity levels of different factor matrices using unbalanced sparse regularization parameters. Moreover, it is worth trying incorporating other types of sparse regularization items to our sparse NCP framework besides $l_1$-norm.

## Acknowledgments

This work was supported by the National Natural Science Foundation of China (Grant Nos. 91748105 & 81471742), the Fundamental Research Funds for the Central Universities [DUT2019] in Dalian University of Technology in China and the scholarships from China Scholarship Council (Nos. 201600090043 & 201706060262).

## References

[1] 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.
[2] A. Cichocki, D. Mandic, L. D. Lathauwer, G. Zhou, Q. Zhao, C. Caiafa, and H. A. PHAN, "Tensor decompositions for signal processing applications: From two-way to multiway component analysis," *IEEE Signal Processing Magazine*, vol. 32, no. 2, pp. 145–163, Mar. 2015.
[3] M. A. Veganzones, J. E. Cohen, R. C. Farias, J. Chanussot, and P. Comon, "Nonnegative tensor CP decomposition of hyperspectral data," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 54, no. 5, pp. 2577–2588, May 2016.
[4] 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.
[5] D. Wang, Y. Zhu, T. Ristaniemi, and F. Cong, "Extracting multimode ERP features using fifth-order nonnegative tensor decomposition," *Journal of Neuroscience Methods*, vol. 308, pp. 240–247, Oct. 2018.
[6] S. Elcoroaristizabal, R. Bro, J. A. García, and L. Alonso, "PARAFAC models of fluorescence data with scattering: A comparative study," *Chemometrics and Intelligent Laboratory Systems*, vol. 142, pp. 124–130, Mar. 2015.

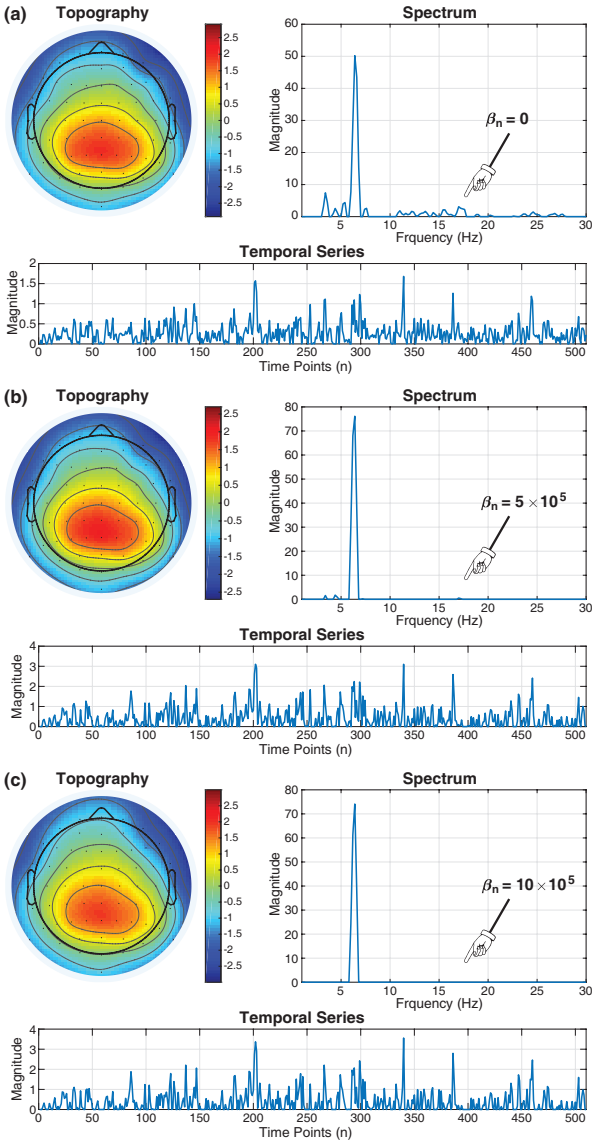



Fig. 4. Selected groups of components from the ongoing EEG tensor using ANQP-BPP based sparse NCP, which show the same brain activity. In the decomposed EEG data, the spatial component is topography, the spectral component is the spectrum, and the temporal component is the energy evolution series. The components in (a) were extracted with $\beta_n = 0$, (b) with $\beta_n = 5 \times 10^5$ and (c) with $\beta_n = 10 \times 10^5$.


[7] X. Vu, C. Chaux, N. Thirion-Moreau, S. Maire, and E. M. Carstea, "A new penalized nonnegative third-order tensor decomposition using a block coordinate proximal gradient approach: Application to 3d fluorescence spectroscopy," *Journal of Chemometrics*, vol. 31, no. 4, p. e2859, Feb. 2017.

[8] A. H. Williams, T. H. Kim, F. Wang, S. Vyas, S. I. Ryu, K. V. Shenoy, M. Schnitzer, T. G. Kolda, and S. Ganguli, "Unsupervised discovery of demixed, low-dimensional neural dynamics across multiple timescales through tensor component analysis," *Neuron*, vol. 98, no. 6, pp. 1099–1115.e8, Jun. 2018.

[9] M. Mørup, "Applications of tensor (multiway array) factorizations and decompositions in data mining," *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, vol. 1, no. 1, pp. 24–40, Jan. 2011.

[10] E. Acar and B. Yener, "Unsupervised multiway data analysis: A literature survey," *IEEE Transactions on Knowledge and Data Engineering*, vol. 21, no. 1, pp. 6–20, Jan. 2009.

[11] D. Wang, X. Wang, Y. Zhu, P. Toiviainen, M. Huotilainen, T. Ristaniemi, and F. Cong, "Increasing stability of EEG components extraction using sparsity regularized tensor decomposition," in *Advances in Neural Networks – ISNN 2018*. Springer International Publishing, 2018, pp. 789–799.

[12] R. Bro and H. A. L. Kiers, "A new efficient method for determining the number of components in PARAFAC models," *Journal of Chemometrics*, vol. 17, no. 5, pp. 274–286, 2003.

[13] P. O. Hoyer, "Non-negative matrix factorization with sparseness constraints," *Journal of machine learning research*, vol. 5, no. Nov, pp. 1457–1469, 2004.

[14] A. M. Bruckstein, D. L. Donoho, and M. Elad, "From sparse solutions of systems of equations to sparse modeling of signals and images," *SIAM Review*, vol. 51, no. 1, pp. 34–81, Feb. 2009.

[15] D. L. Donoho, "For most large underdetermined systems of linear equations the minimal $\ell_1$-norm solution is also the sparsest solution," *Communications on Pure and Applied Mathematics*, vol. 59, no. 6, pp. 797–829, 2006.

[16] Z. Xu, X. Chang, F. Xu, and H. Zhang, "$l_{1/2}$regularization: A thresholding representation theory and a fast solver," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 23, no. 7, pp. 1013–1027, Jul. 2012.

[17] Y. Qian, S. Jia, J. Zhou, and A. Robles-Kelly, "Hyperspectral unmixing via $L_{1/2}$ sparsity-constrained nonnegative matrix factorization," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 49, no. 11, pp. 4282–4297, Nov. 2011.

[18] J. Sigurdsson, M. O. Ulfarsson, and J. R. Sveinsson, "Hyperspectral unmixing with $l_q$ regularization," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 52, no. 11, pp. 6793–6806, Nov. 2014.

[19] R. Shang, C. Liu, Y. Meng, L. Jiao, and R. Stolkin, "Nonnegative matrix factorization with rank regularization and hard constraint," *Neural Computation*, vol. 29, no. 9, pp. 2553–2579, Sep. 2017.

[20] F. Bach, R. Jenatton, J. Mairal, and G. Obozinski, "Optimization with sparsity-inducing penalties," *Foundations and Trends® in Machine Learning*, vol. 4, no. 1, pp. 1–106, 2012.

[21] 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.

[22] 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.

[23] D. D. Lee and H. S. Seung, "Learning the parts of objects by non-negative matrix factorization," *Nature*, vol. 401, no. 6755, pp. 788–791, Oct. 1999.

[24] ——, "Algorithms for non-negative matrix factorization," in *Advances in neural information processing systems*, 2001, pp. 556–562.

[25] 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.

[26] 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.

[27] Y. Zhang, G. Zhou, Q. Zhao, A. Cichocki, and X. Wang, "Fast nonnegative tensor factorization based on accelerated proximal gradient and low-rank approximation," *Neurocomputing*, vol. 198, pp. 148–154, Jul. 2016.

[28] S. Boyd, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Foundations and Trends® in Machine Learning*, vol. 3, no. 1, pp. 1–122, 2011.

[29] A. P. Liavas and N. D. Sidiropoulos, "Parallel algorithms for constrained tensor factorization via alternating direction method of multipliers," *IEEE Transactions on Signal Processing*, vol. 63, no. 20, pp. 5450–5463, Oct. 2015.

[30] K. Huang, N. D. Sidiropoulos, and A. P. Liavas, "A flexible and efficient algorithmic framework for constrained matrix and tensor factorization," *IEEE Transactions on Signal Processing*, vol. 64, no. 19, pp. 5052–5065, Oct. 2016.

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

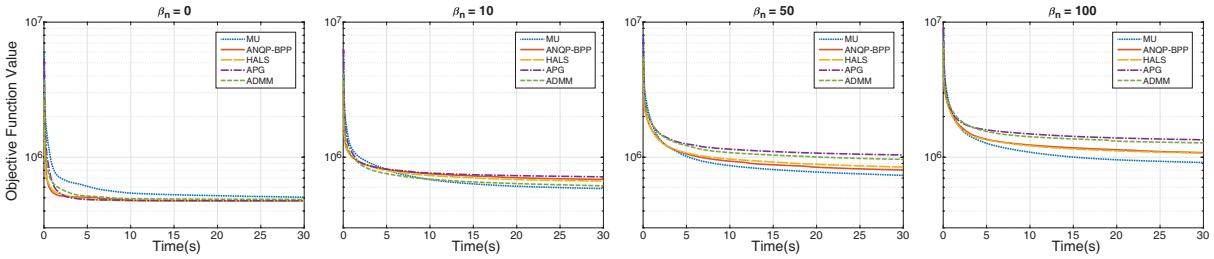



Fig. 5. The Objective Function Value Curves of Sparse NCPs on Fourth-order ERP Tensor.


[32] R. Zdunek and A. Cichocki, "Non-negative matrix factorization with quasi-newton optimization," in *International conference on artificial intelligence and soft computing*. Springer, 2006, pp. 870–879.

[33] H. Kim and H. Park, "Sparse non-negative matrix factorizations via alternating non-negativity-constrained least squares for microarray data analysis," *Bioinformatics*, vol. 23, no. 12, pp. 1495–1502, May 2007.

[34] ——, "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.

[35] 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.

[36] ——, "Fast nonnegative tensor factorization with an active-set-like method," in *High-Performance Scientific Computing*. Springer London, 2012, pp. 311–326.

[37] X. Dai, C. Li, X. He, and C. Li, "Nonnegative matrix factorization algorithms based on the inertial projection neural network," *Neural Computing and Applications*, Jan. 2018.

[38] P. O. Hoyer, "Non-negative sparse coding," in *Proceedings of the 12th IEEE Workshop on Neural Networks for Signal Processing*. IEEE, 2002.

[39] M. Mørup, L. K. Hansen, and S. M. Arnfred, "Algorithms for sparse nonnegative tucker decompositions," *Neural Computation*, vol. 20, no. 8, pp. 2112–2131, Aug. 2008.

[40] J. Liu, J. Liu, P. Wonka, and J. Ye, "Sparse non-negative tensor factorization using columnwise coordinate descent," *Pattern Recognition*, vol. 45, no. 1, pp. 649–656, Jan. 2012.

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

[42] 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.

[43] D. Wang, F. Cong, and T. Ristaniemi, "Higher-order nonnegative CANDECOMP/PARAFAC tensor decomposition using proximal algorithm," in *2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, May 2019, pp. 3457–3461.

[44] N. Gillis and F. Glineur, "Accelerated multiplicative updates and hierarchical ALS algorithms for nonnegative matrix factorization," *Neural Computation*, vol. 24, no. 4, pp. 1085–1105, Apr. 2012.

[45] N. Vervliet and L. D. Lathauwer, "Numerical optimization-based algorithms for data fusion," in *Data Handling in Science and Technology*. Elsevier, 2019, pp. 81–128.

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

[47] N. D. Sidiropoulos, L. D. Lathauwer, X. Fu, K. Huang, E. E. Papalexakis, and C. Faloutsos, "Tensor decomposition for signal processing and machine learning," *IEEE Transactions on Signal Processing*, vol. 65, no. 13, pp. 3551–3582, Jul. 2017.

[48] 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.

[49] D. Cai, X. He, J. Han, and T. S. Huang, "Graph regularized nonnegative matrix factorization for data representation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, no. 8, pp. 1548–1560, Aug. 2011.

[50] Y. Li and A. Ngom, "The non-negative matrix factorization toolbox for biological data mining," *Source Code for Biology and Medicine*, vol. 8, no. 1, p. 10, 2013.

[51] Y. Xu, W. Yin, Z. Wen, and Y. Zhang, "An alternating direction algorithm for matrix completion with nonnegative factors," *Frontiers of Mathematics in China*, vol. 7, no. 2, pp. 365–384, Apr. 2012.

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

[53] 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/

[54] 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.

[55] A. Cichocki and R. Zdunek, "NMFLAB - MATLAB toolbox for non-negative matrix factorization," 2006.

[56] M. E. Timmerman and H. A. L. Kiers, "Three-mode principal components analysis: Choosing the numbers of components and sensitivity to local optima," *British Journal of Mathematical and Statistical Psychology*, vol. 53, no. 1, pp. 1–16, May 2000.

[57] 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.


**Deqing Wang** (S'18) received the B.E. degree in automation and the M.E. degree in pattern recognition and intelligent system from Harbin Engineering University, Harbin, China, in 2009 and 2012, respectively. He was appointed as assistant engineer and engineer in Dalian Scientific Test and Control Technology Institute, China Shipbuilding Industry Corporation (CSIC), Dalian, China, in 2012 and 2014, respectively. He is currently a joint training doctoral student in both Dalian University of Technology, China, and University of Jyväskylä, Finland. His research interests include signal processing, machine learning, tensor decomposition and EEG data analysis.

**Xiulin Wang** (S'19) received the B.E. degree in communication engineering from Shandong University, Weihai campus, China in 2012. He received the M.E. degree in signal and information processing from Dalian University of Technology, China, in 2015. He was a junior engineer of software development in Beijing Huiqing Techology Co., Ltd, China, from 2015 to 2016. Currently, he is pursuing the doctoral degrees in both Dalian University of Technology, China, and University of Jyväskylä, Finland. His research interests include algebraic methods for multiset signal processing, coupled matrix/tensor decomposition and brain signal processing.

**Tapani Ristaniemi** (SM'11) received his M.Sc. in 1995 (Mathematics), Ph.Lic. in 1997 (Applied Mathematics) and Ph.D. in 2000 (Wireless Communications), all from University of Jyväskylä, Jyväskylä, Finland. In 2001 he was appointed as Professor in Faculty of Information Technology, University of Jyväskylä. In 2004 he moved to Department of Communications Engineering, Tampere University of Technology, Tampere, Finland, where he was appointed as Professor in Wireless Communications. In 2006 he moved back to University of Jyväskylä to take up his appointment as Professor in Computer Science. He is an Adjunct Professor of Tampere University of Technology. In 2013 he was a Visiting Professor in the School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore. Currently, he is a full professor at University of Jyväskylä. He is also a co-founder of Magister Solutions Ltd. He has authored or co-authored over 220 publications in international journals, book chapters and conference proceedings. His research interests include brain signal processing and wireless communication systems.

**Fengyu Cong** (M'08-SM'13) received the B.S. degree in Power and Thermal Dynamic Engineering and the Ph.D. degree in Mechanical Design and Theory from Shanghai Jiao Tong University, China, in 2002 and 2007, and the Ph.D. degree in Mathematical Information Technology from University of Jyväskylä, Finland, in 2010. From 2007 to 2010 he was a postdoctoral researcher and junior lecturer, and from 2011 he holded a tenure-track faculty position in University of Jyväskylä. In May 2012, he was conferred the title of Docent (Adjunct Professor) in Signal Processing in Faculty of Information Technology, University of Jyväskylä. Since 2013, he has been a full professor in School of Biomedical Engineering, Dalian University of Technology, Dalian, China.

He has authored or co-authored over 80 publications in international journals, book chapters and conference proceedings. As the first author, he published one monograph entitled "Advanced Signal Processing on Brain Event-Related Potentials: Filtering ERPs in Time, Frequency and Space Domains Sequentially and Simultaneously" in World Scientific in 2015. His research interests include brain/acoustic signal processing, blind source separation/independent component analysis, tensor decomposition, higher-order partial least squares, and sequential Monte Carlo.

# PII

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

by

Deqing Wang, Fengyu Cong and Tapani Ristaniemi 2019

2019 IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP), Brighton, UK

(Best student paper award)

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 $N$th-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 $k$th iteration is solved as the following subproblem:

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

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:

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

where $\alpha_n$ 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 $k$th iteration can be updated alternatively by the following subproblem:

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

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 $\alpha_n$, $n = 1, \ldots, N$ 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].

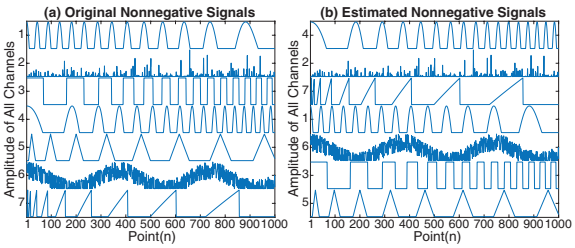



**Fig. 1**. Simulated signals. (a) shows original signals, and (b) shows the estimated signals by PROX-BPP with $\alpha_n$ =1e-4.

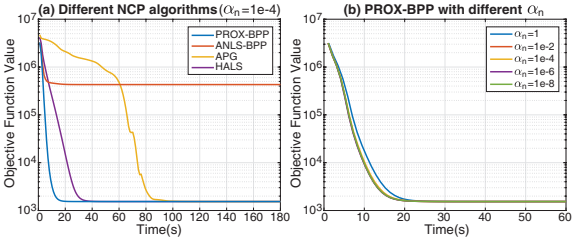



**Fig. 2**. Convergence of NCP algorithms on the synthetic data.

### 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 $\mathcal{X}_{\text{Syn}} = [\![\boldsymbol{S}^{(1)}, \boldsymbol{A}^{(2)}, \boldsymbol{A}^{(3)}, \boldsymbol{A}^{(4)}]\!] \in \mathbb{R}^{1000\times100\times100\times5}$, in which $\boldsymbol{S}^{(1)} \in \mathbb{R}^{1000\times7}$ is the signal matrix, and $\boldsymbol{A}^{(2)}, \boldsymbol{A}^{(3)} \in \mathbb{R}^{100\times7}$, $\boldsymbol{A}^{(4)} \in \mathbb{R}^{5\times7}$ 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-

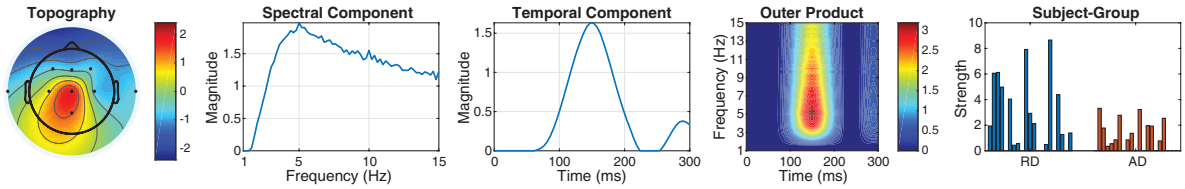



**Fig. 3**. One group of components extracted from ERP data by PROX-BPP with $\alpha_n$ =1e-4.

**Table 1**. Performances of NCPs On Synthetic Data

| NCP | $\alpha_n$ | 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 | $\alpha_n$ | 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 $\alpha_n$ 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].

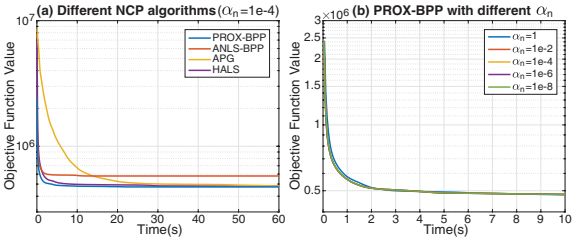



**Fig. 4**. Convergence of NCP algorithms on the real ERP data.

### 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 $\alpha_n$ 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 $\leqslant \alpha_n \leqslant$ 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.

# PIII

# EXTRACTING MULTI-MODE ERP FEATURES USING FIFTH-ORDER NONNEGATIVE TENSOR DECOMPOSITION

by

Deqing Wang, Yongjie Zhu, Tapani Ristaniemi and Fengyu Cong 2018

Journal of Neuroscience Methods, Volume 308, p.240-247

https://doi.org/10.1016/j.jneumeth.2018.07.020

Journal of Neuroscience Methods 308 (2018) 240–247

Contents lists available at ScienceDirect

Journal of Neuroscience Methods

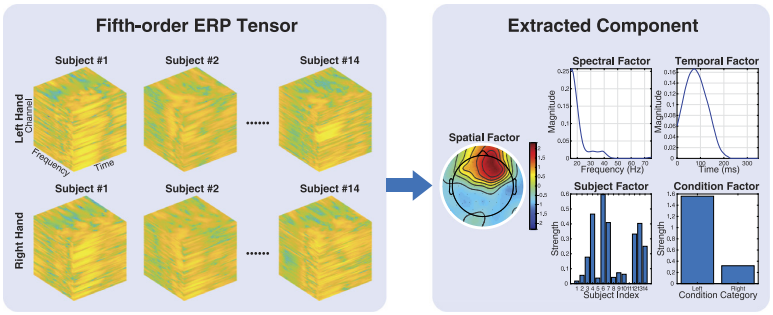
journal homepage: www.elsevier.com/locate/jneumeth

# Extracting multi-mode ERP features using fifth-order nonnegative tensor decomposition

Deqing Wang [a,b], Yongjie Zhu [a,b], Tapani Ristaniemi [b], Fengyu Cong [a,b,*]

[a] School of Biomedical Engineering, Faculty of Electronic Information and Electrical Engineering, Dalian University of Technology, Dalian, China
[b] Faculty of Information Technology, University of Jyväskylä, Jyväskylä, Finland

## GRAPHICAL ABSTRACT




## ARTICLE INFO

*Keywords:*
Nonnegative tensor decomposition
CANDECOMP/PARAFAC
Event-related potential
Multi-mode features
Component number selection

## ABSTRACT

**Background:** Preprocessed Event-related potential (ERP) data are usually organized in multi-way tensor, in which tensor decomposition serves as a powerful tool for data processing. Due to the limitation of computation burden for multi-way data and the low algorithm performance of stability and efficiency, multi-way ERP data are conventionally reorganized into low-order tensor or matrix before further analysis. However, the reorganization may hamper mode specification and spoil the interaction information among different modes.

**New Method:** In this study, we applied a fifth-order tensor decomposition to a set of fifth-order ERP data collected by exerting proprioceptive stimulus on left and right hand. One of the latest nonnegative CANDECOMP/PARAFAC (NCP) decomposition methods implemented by alternating proximal gradient (APG) was employed. We also proposed an improved DIFFIT method to select the optimal component number for the fifth-order tensor decomposition.

**Results:** By the fifth-order NCP model with a proper component number, the ERP data were fully decomposed into spatial, spectral, temporal, subject and condition factors in each component. The results showed more pairs of components with symmetric activation region in left and right hemisphere elicited by contralateral stimuli on hand.

**Comparison with Existing Method(s):** In our experiment, more interesting components and coherent brain activities were extracted, compared with previous studies.

**Conclusions:** The discovered activities elicited by proprioceptive stimulus are in line with those in relevant cognitive neuroscience studies. Our proposed method has proved to be appropriate and viable for processing high-order EEG data with well-preserved interaction information among all modes.

* Corresponding author.
*E-mail addresses:* deqing.wang@foxmail.com (D. Wang), yongjie.zhu@foxmail.com (Y. Zhu), tapani.e.ristaniemi@jyu.fi (T. Ristaniemi), cong@dlut.edu.cn (F. Cong).
*URLs:* http://www.deqing.me (D. Wang), http://www.users.jyu.fi/~riesta (T. Ristaniemi), http://www.escience.cn/people/cong (F. Cong).

https://doi.org/10.1016/j.jneumeth.2018.07.020
Received 23 April 2018; Received in revised form 8 July 2018; Accepted 31 July 2018
Available online 2 August 2018

## 1. Introduction

Tensor decomposition, as a versatile tool for signal processing and machine learning (Cichocki et al., 2015; Sidiropoulos et al., 2017), has become more and more popular for EEG data processing and cognitive neuroscience in recent years (Cong et al., 2015; Zhou et al., 2016; Mahyari et al., 2017; Idaji et al., 2017; Wang et al., 2018). Event-related potential (ERP) is a time-locked EEG activity measuring brain response elicited by perceptual, cognitive or motor events (Handy, 2005). ERP can be represented naturally in tensor form. In these studies (Zhang et al., 2013; Idaji et al., 2017; Vanderperren et al., 2012; Niknazar et al., 2014), ERP data are arranged in channel × time × trial tensor. If ERP data are transformed into time-frequency representation (by Continuous Wavelet Transform), a new frequency mode will be introduced. Consequently, the ERP data of single-trial or the average of trials can be represented by channel × frequency × time tensor (Mørup et al., 2006a; Weis et al., 2009; Zhao et al., 2011; Cong et al., 2012, 2013). In an experiment, ERP data can be represented by high-order tensor including modes such as space, frequency, time, trial, subject, condition and group (Cong et al., 2015). Afterwards, tensor decomposition can be performed for the multi-way ERP data.

CANDECOMP/PARAFAC (CP), as a basic tensor decomposition method (Sidiropoulos et al., 2017), has remarkable advantage in processing high-order EEG data, in which a group of related feature factors can be extracted from each mode (a feature factor refers to a column of the matrix in some mode after tensor decomposition). When time-frequency representation is applied on EEG time series, the transformed data are nonnegative, and nonnegative CANDECOMP/PARAFAC (NCP) is preferred with nonnegative constraints in each mode (Cichocki et al., 2009). The rationale for using NCP in time-frequency transformed EEG tensor is that: after decomposition, (1) the temporal factor representing temporal envelope is nonnegative; (2) the spectral factor representing spectrum is nonnegative; and (3) the spatial factor representing topography can also be nonnegative. Specifically, for a brain activity, the temporal envelope exhibits the temporal evolution; the spectrum reveals the most prominent frequency band; and the topography indicates the location on the scalp. The meaning of NCP for third-order EEG tensor (channel × frequency × time) is illustrated in Fig. 1. When higher-order (> 3) EEG tensor is decomposed by NCP, more feature factors from different modes (e.g. subject, condition) will be extracted simultaneously, with the degree of strength given by score vectors being nonnegative.

In general, the processing of high-order EEG tensor data is time-consuming, in which the stability and convergence of tensor decomposition algorithms cannot be guaranteed. Hence high-order EEG data are often reshaped into low-order tensor by merging several modes together, or unfolded into two-order matrix. In a study of wavelet transformed ERP (Mørup et al., 2006b), the fifth-order ERP data (channel × frequency × time × subject × condition) were reorganized into a third-order tensor (channel × time-frequency × subject-condition). In (Cong et al., 2012), a fourth-order ERP tensor (channel × frequency × time × subject-condition) was generated with subject and condition modes being merged. However, the merging or unfolding of modes can potentially hamper data interpretation, dismiss mode specification, and spoil the interaction information among these modes (Cong et al., 2015; Mørup et al., 2006b).

Mørup et al. applied fifth-order NCP to decompose another fifth-order ERP data, the condition mode size of which is two (Mørup et al., 2006a). However, there existed an unreasonable assumption that the data could only entail a two-component CP model (Mørup et al., 2006a). The rationale behind this assumption might be that the component number was no more than the minimal mode size of two. In fact, the selection of component number for CP model is related to the rank of tensor. Since the rank of tensor can be larger than the maximal mode size (Sidiropoulos et al., 2017), a larger component number can be selected.

In this study, we analyze a set of fifth-order ERP data (channel × frequency × time × subject × condition) elicited by proprioceptive stimulus. Fifth-order NCP is applied with a large component number. The data are fully decomposed in each mode, and the interaction information among these five modes is retained. One of the latest NCP models implemented by alternating proximal gradient (APG) method is employed, which has been proved to be mathematically convergent and stable (Xu and Yin, 2013).

It is nontrivial to determine the component number by the tensor rank directly. Some studies have focused on the selection of component number for multi-way models, such as core consistency diagnostic (CORCONDIA) (Bro and Kiers, 2003) and automatic relevance determination

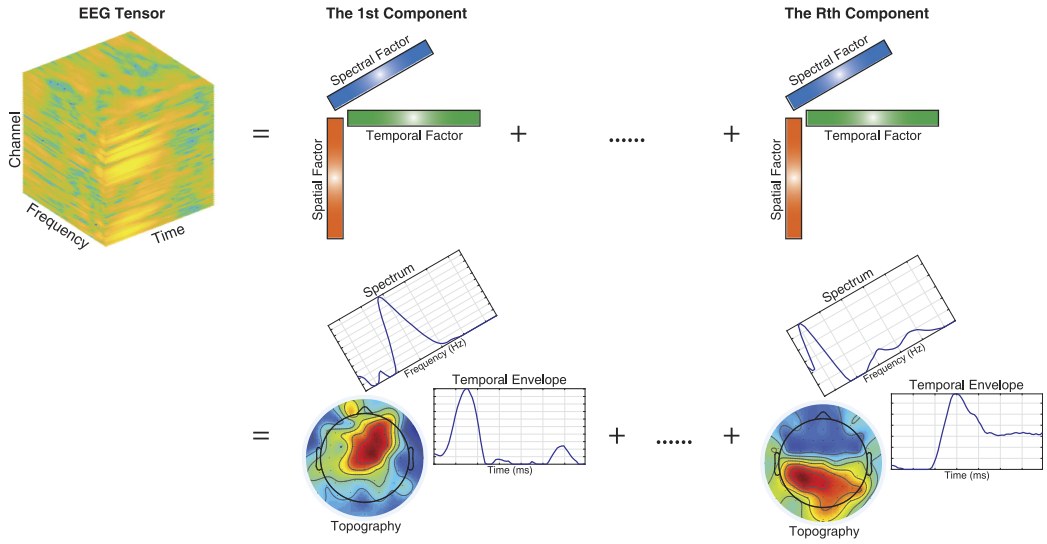



**Fig. 1.** Tensor decomposition of third-order EEG tensor (channel × frequency × time) by NCP. The time-frequency transformed EEG tensor is nonnegative. After decomposition, the temporal factor representing temporal envelop is nonnegative, the spectral factor representing spectrum is nonnegative, and the spatial factor representing topography can also be nonnegative.

(ARD) (Mørup and Hansen, 2009). Nevertheless, these methods often indicate very few components, which are not adequate to reveal the physiological properties of EEG signals (Cong et al., 2015). Conventional DIFFIT (Timmerman and Kiers, 2000) is a preferred method for component number selection in EEG data processing (Cong et al., 2015), but it usually fails to provide useful information due to fluctuations on DIFFIT curve. Therefore, we propose an improved smooth DIFFIT method that can select a proper number of components to make sure that the most important components are included.

The assumption of the fifth-order analysis is that the underlying spatial, spectral, and temporal factors are the same among all subjects, however, with a subject-dependent strength given by the subject score vector and with a variable strength in all conditions (Mørup et al., 2006a). Our method satisfies the above assumption and meanwhile reveals more interesting components and coherent activities compared with previous study (Mørup et al., 2006b). Our findings are also in line with related cognitive neuroscience explanations of proprioceptive stimulus (Herrmann et al., 2004b; Arnfred et al., 2011).

## 2. Nonnegative CANDECOMP/PARAFAC decomposition

### 2.1. Notation

In this paper, we denote a vector by boldface lowercase letter, such as $\boldsymbol{x}$; a matrix by boldface uppercase letter, such as $\boldsymbol{X}$; and a high order tensor by boldface script letter, such as $\mathcal{X}$. Operator $\circ$ represents outer product of vectors, $\langle\ \rangle$ represents inner product, $[\![\ ]\!]$ represents Kruskal operator, and $\|\ \|_F$ means Frobenius norm. Nonnegative CANDECOMP/PARAFAC decomposition is abbreviated as NCP for convenience in following contents.

### 2.2. Mathematical model

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

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

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 selected rank-1 tensor number (component number). The estimated factor matrices in Kruskal operator can be represented by the sum of $R$ rank-1 tensors in outer product form:

$$[\![ \boldsymbol{A}^{(1)}, \ldots, \boldsymbol{A}^{(N)} ]\!] = \sum_{r=1}^{R} \widetilde{\mathcal{X}}_r = \sum_{r=1}^{R} \boldsymbol{a}_r^{(1)} \circ \cdots \circ \boldsymbol{a}_r^{(N)}, \tag{2}$$

where $\boldsymbol{a}_r^{(n)}$ represents the $r$th column of $\boldsymbol{A}^{(n)}$.

### 2.3. Optimization scheme

Conventionally, there are many optimization methods that can be applied to solve NCP problem, such as multiplicative updating (MU), alternating least squares (ALS), hierarchical alternating least squares (HALS) (Cichocki et al., 2009). Recently, Xu and Yin (2013) have proposed the alternating proximal gradient (APG) method to solve nonnegative matrix and tensor decomposition problems in block coordinate descend (BCD) framework. APG outperforms many conventional methods both in accuracy and efficiency. More importantly, it has better convergence properties. Therefore, APG method is employed to solve NCP in our study. We don't make a comparison among different optimization methods in this study, which might be done in future.

Supposing that $\widehat{\boldsymbol{A}}^{(n)}$ is an extrapolated point, $\widehat{\boldsymbol{G}}^{(n)}$ is the block-partial gradient at $\widehat{\boldsymbol{A}}^{(n)}$ and $L^{(n)}$ is a Lipschitz constant, factor matrix $\boldsymbol{A}^{(n)}$ can be updated alternatively by

$$\boldsymbol{A}^{(n)} \leftarrow \arg\min_{\boldsymbol{A}^{(n)} \geqslant 0} \left\langle \widehat{\boldsymbol{G}}^{(n)}, \boldsymbol{A}^{(n)} - \widehat{\boldsymbol{A}}^{(n)} \right\rangle + \frac{L^{(n)}}{2} \left\| \boldsymbol{A}^{(n)} - \widehat{\boldsymbol{A}}^{(n)} \right\|_F^2. \tag{3}$$

Furthermore, (3) can be rewritten in the closed form

$$\boldsymbol{A}^{(n)} \leftarrow \max\left(0, \widehat{\boldsymbol{A}}^{(n)} - \widehat{\boldsymbol{G}}^{(n)} / L^{(n)}\right). \tag{4}$$

The detailed solution of NCP model (1) using APG and its convergence analysis can be found in (Xu and Yin, 2013).

## 3. Component number estimation

Before applying tensor decomposition to EEG data, it is required to determine a proper component number, which is the rank-1 tensor number $R$ in (2). The component number is closely related to the rank of tensor. For a two-way matrix $\boldsymbol{X} \in \mathbb{R}^{I \times J}$, it follows $\text{rank}(\boldsymbol{X}) \leqslant \min(I, J)$, but for a multi-way tensor $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$, $\text{rank}(\mathcal{X})$ can be even larger than $\max(I_1, I_2, \ldots, I_N)$ (Sidiropoulos et al., 2017). Therefore, the component number for tensor decomposition can be selected with a large number, which is not restricted by the size in each mode. However, it is nontrivial to determine $R$ by tensor rank. DIFFIT is a conventional method to determine component number, and has been used in ERP tensor data decomposition (Cong et al., 2012).

### 3.1. DIFFIT method

DIFFIT refers to the difference in data fitting, and is calculated based on relative error/residual and the explained variance (or the explained sum of squares) (Timmerman and Kiers, 2000; Mørup and Hansen, 2009). The relative error/residual of NCP is defined as

$$\text{RelErr} = \frac{\left\| \mathcal{X} - [\![ \boldsymbol{A}^{(1)}, \ldots, \boldsymbol{A}^{(N)} ]\!] \right\|_F}{\|\mathcal{X}\|_F}. \tag{5}$$

Let component number $R \in [1, \mathcal{R}]$, where $\mathcal{R}$ is the empirically maximal component number. When $R$ is selected, the explained variance of the estimated $R$ components from NCP is

$$\text{ExpVar}(R) = 1 - \frac{\left\| \mathcal{X} - [\![ \boldsymbol{A}^{(1)}, \ldots, \boldsymbol{A}^{(N)} ]\!] \right\|_F^2}{\|\mathcal{X}\|_F^2}. \tag{6}$$

In order to obtain a more precise value, NCP model is usually run many times (e.g. $T$ times) for each $R$ and the average explained variance is calculated yielding an averaged sequence

$$\boldsymbol{e} = \left[ \overline{\text{ExpVar}}(1), \ldots, \overline{\text{ExpVar}}(R), \ldots, \overline{\text{ExpVar}}(\mathcal{R}) \right]. \tag{7}$$

Then, calculate the DIF (difference of explained variance) and DIFFIT as follows:

$$\text{DIF}(R) = \overline{\text{ExpVar}}(R) - \overline{\text{ExpVar}}(R-1), \tag{8}$$

$$\text{DIFFIT}(R) = \frac{\text{DIF}(R)}{\text{DIF}(R+1)}. \tag{9}$$

Sometimes, NCP model is run tens or hundreds of times for each $R$, which is very time-consuming and even worse for big data or higher-order tensor. Meanwhile, since the discrete sequence of averaged explained variances has only $\mathcal{R}$ values (usually several tens), there probably exist serious fluctuations and fake spikes on the final DIFFIT curve. Then, we propose an improved method named "smoothed DIFFIT".

### 3.2. Smoothed DIFFIT

Run NCP fewer times ($T$ = 10,20, or other values according to the computer's computation ability) for each $R$, and obtain vector $\boldsymbol{e}$ based on (7). From vector $\boldsymbol{e}$, a smooth curve is generated by polynomial curve fitting (PCF) using a $m$th-order equation

$$p(x) = p_1 x^m + p_2 x^{m-1} +, \ldots, + p_m x^1 + p_{m+1}. \tag{10}$$

After polynomial curve fitting, the new sequence is

$$e_{\mathrm{PCF}} = \left[\mathrm{ExpVar}_{\mathrm{PCF}}(1), \ldots, \mathrm{ExpVar}_{\mathrm{PCF}}(\mathcal{R})\right]. \tag{11}$$

We recompute the DIF and DIFFIT as follows:

$$\mathrm{DIF}_{\mathrm{PCF}}(R) = \mathrm{ExpVar}_{\mathrm{PCF}}(R) - \mathrm{ExpVar}_{\mathrm{PCF}}(R-1), \tag{12}$$

$$\mathrm{DIFFIT}_{\mathrm{PCF}}(R) = \frac{\mathrm{DIF}_{\mathrm{PCF}}(R)}{\mathrm{DIF}_{\mathrm{PCF}}(R+1)}. \tag{13}$$

## 4. Experiments and results

### 4.1. Data description

The ERP data in our experiment come from an open preprocessed dataset associated with ERPWAVELAB toolbox (Mørup et al., 2007), which can be downloaded from www.erpwavelab.org. The data were collected from a proprioceptive experiment, in which two conditions (left and right hand) were manipulated with the increment of handhold load. An important part of the stimuli is the change of applied force on a static muscle contraction, which is conceived as proprioceptive stimulus (Mørup et al., 2006b). Fourteen subjects participated in the experiment and 64 scalp electrodes were used to record EEG data. A total of 360 trials (epochs) were obtained from each subject under each condition. All epochs were transformed into time-frequency representation (TFR) by complex Morlet wavelet. In the wavelet transform, only the frequency band from 15 Hz to 75 Hz were analyzed with linear interval of 1 Hz. Then, inter-trial phase coherence (ITPC) (Delorme and Makeig, 2004) was computed as an average spectral estimate across all trials. Since the TFR was first applied to each trial and then the average was calculated across trials, ITPC can be seen as induced oscillations of brain (David et al., 2006). Meanwhile, ITPC only takes values between 0 and 1 (Delorme and Makeig, 2004; Cohen, 2014). Finally, a fifth-order nonnegative tensor (channel × frequency × time × subject × condition = 64 × 61 × 72 × 14 × 2) was generated. The structure of the fifth-order tensor is shown in Fig. 2. The 61 frequency points represent 15–75 Hz, and the 72 time points represent 0–346.68 ms.

The detailed EEG data collection and preprocessing procedures are described in (Mørup et al., 2006b).

### 4.2. Component number selection

Both the original DIFFIT and our proposed smoothed DIFFIT methods were tested. The results are shown in Fig. 3.

We tested NCP on the fifth-order tensor data by increasing $R$ from 1 to 50 ($\mathcal{R} = 50$). With each selected component number $R$, NCP was run 20 times. The averaged explained variance vector $\boldsymbol{e}$, DIF and DIFFIT curves are shown in Fig. 3(a)–(c). Obviously, many fluctuations appear on the DIF and DIFFIT curves. From Fig. 3(b) we can find that the DIF values at $R = 31$ and $R = 44$ are very close to 0. According to (9), this will cause large DIFFIT values with two long fake spikes in Fig. 3(c) at $R = 30$ and $R = 43$. The DIFFIT curve in Fig. 3(c) entails very limited useful information.

We fit $\boldsymbol{e}$ by a 7th-order polynomial, and the obtained polynomial is

$$\begin{aligned} p(x) =& 2.0486^{-11}x^7 - 3.9518^{-9}x^6 + 3.0904^{-7}x^5 \\ & - 1.2586^{-5}x^4 + 2.8604^{-4}x^3 - 3.6465^{-3}x^2 \\ & + 2.6685^{-2}x + 7.3496^{-1}. \end{aligned}$$

The fitted curve $e_{\mathrm{PCF}}$ is shown in Fig. 3(d). By visual inspection, the results of 5th and 6th-order polynomials were under-fitting, while 8th and 9th-order were over-fitting. 7th-order polynomial has the best fitting for the data in our experiment.

Subsequently, $\mathrm{DIF}_{\mathrm{PCF}}$ and $\mathrm{DIFFIT}_{\mathrm{PCF}}$ were calculated, as shown in Fig. 3(e) and (f). In Fig. 3(f), when the DIFFIT value is larger than 1, DIF curve decreases; when it is smaller than 1, DIF curve increases; and when it equals to 1, DIF curve doesn't change. The two local maximums on DIFFIT curve at $R = 4$ and $R = 24$ indicate two positions on DIF curve that have fast dropping rate. The values after $R = 45$ can be ignored due to the poor fitting at the end of the curve. From Fig. 3(e) and (f) we find that at $R = 20$ the DIF value decreases rapidly again approaching to a rather low value, which indicates that, after $R = 20$,

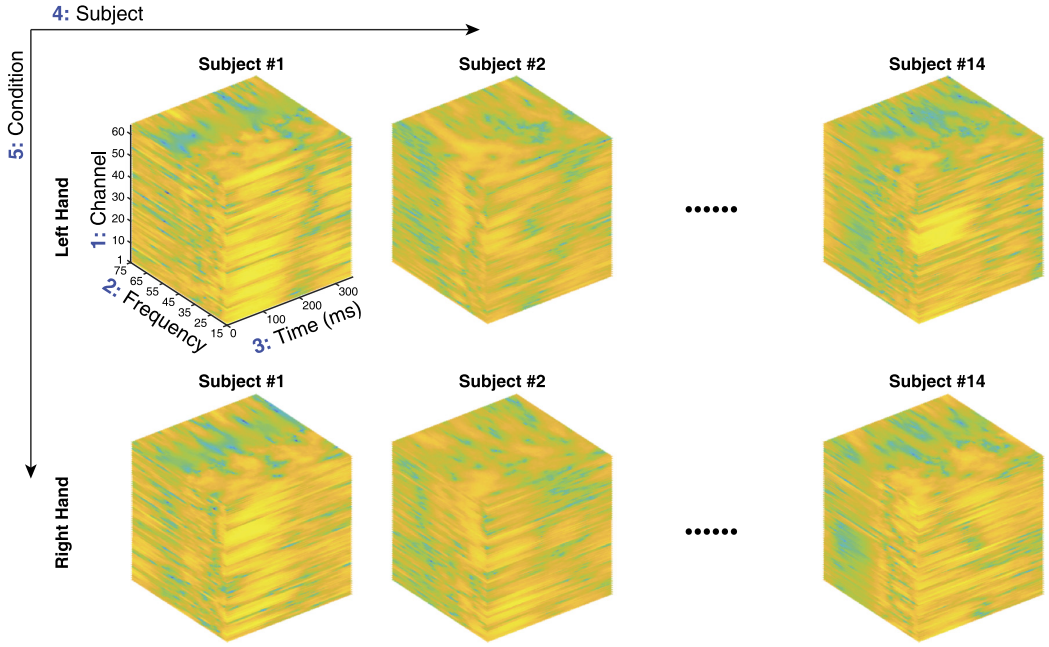



**Fig. 2.** Fifth-order ERP tensor (channel × frequency × time × subject × condition). This tensor has 64 channels, 61 frequency points representing 15–75 Hz, 72 time points representing 0–346.68 ms, 14 subjects and 2 conditions of stimuli on left and right hand.

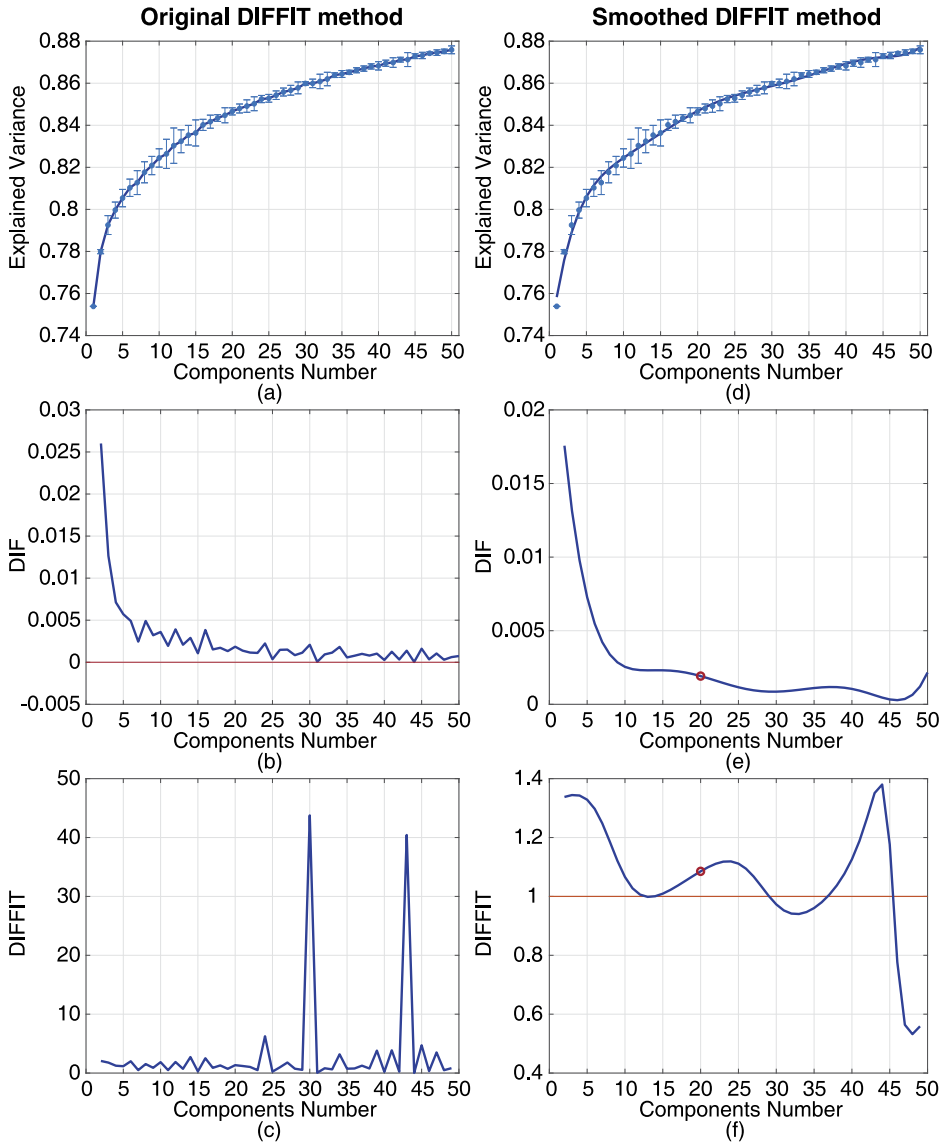



**Fig. 3.** DIFFIT for component number selection. (a)–(c) represent original DIFFIT method. (d)–(e) represent the proposed smoothed DIFFIT method.

the explained variance doesn't increase that much. Based on above analysis, we assume that, at $R = 20$, the estimated factors of tensor decomposition contain the most important EEG components. Therefore, $R = 20$ is selected as the optimal component number in NCP model for this fifth-order tensor data. In fact, our smoothed DIFFIT method can estimate a narrow range of proper component numbers. For example, in case of $R = 19$ or 21, similar results can be obtained as $R = 20$. However, if $R = 30$, more unimportant components, such as weak signals or noise, might be included in the results.

### 4.3. Tensor decomposition implementation

#### 4.3.1. Factor initialization

All factors of the five modes were initialized using normally distributed random numbers as in (Xu and Yin, 2013).

#### 4.3.2. Stop criteria

Based on (5), the relative error/residual of NCP at the $k$th iteration was defined as $\mathrm{RelErr}_k = \frac{\left\|\mathcal{X} - [\![\boldsymbol{A}_k^{(1)}, \ldots, \boldsymbol{A}_k^{(N)}]\!]\right\|_F}{\|\mathcal{X}\|_F}$. We terminated the NCP decomposition process when the following condition between two iterations $k$ and $k+1$ was satisfied:

$$T_{\text{stop}} = |\mathrm{RelErr}_k - \mathrm{RelErr}_{k+1}| \leq \epsilon,$$

where $\epsilon = 10^{-6}$ in this study.

### 4.4. Components analysis

In this section, we name the $r$th rank-1 tensor $\widetilde{\mathcal{X}}_r$ after decomposition as the $r$th component. A component further contains spatial, temporal, temporal, subject, and condition factor, which can be represented by $\widetilde{\mathcal{X}}_r = \boldsymbol{a}_r^{(spatial)} \circ \boldsymbol{a}_r^{(spectral)} \circ \boldsymbol{a}_r^{(temporal)} \circ \boldsymbol{a}_r^{(subject)} \circ \boldsymbol{a}_r^{(condition)}$ according to (2).

After NCP decomposition, a total of 20 components were obtained. The assumption of the fifth-order analysis is that the underlying spatial, spectral, and temporal factors are the same among all subjects only with differences in subject-dependent and condition-dependent strength. In the results, we discovered five pairs of components (10 components) with symmetric responses on topographies, which are shown in Figs. 4 and 5. It is clear that left-hand stimuli elicit activities in the right hemisphere, and right-hand stimuli elicit activities in the left hemisphere. We also calculated time-frequency representation (TFR) for each component, which is the outer product of the spectral and temporal factor. The frequency of TFR is presented in logarithm scale.

Fig. 4 shows the components with frequency oscillations in beta-band (15–30 Hz). Fig. 4(a) displays a pair of components with symmetric activities in right and left frontal lobe elicited by contralateral hand stimuli. Both components appear within 15–20 Hz at 75 ms. This pair of components was also found in (Mørup et al., 2006b). Fig. 4(b) shows symmetric activities in right and left temporal lobe within 15–20 Hz at 75 ms. Activities in Fig. 4(c) appear in right and left frontal lobe within 25–30 Hz at 60 ms. These beta-band activities emerge before 100 ms after stimulus onset, which is consistent with previous study about beta-frequency oscillations of proprioceptive information processing (Arnfred et al., 2011).

Fig. 5 demonstrates the components with frequency oscillations in gamma-band (30–75 Hz). Fig. 5(a) shows symmetric activities in right and left temporal lobe, and they occur within 30–40 Hz at 60 ms. This pair of components was also found in Mørup et al. (2006b). In Fig. 5(b), the other two components with the frequency of 40–75 Hz in the region between parietal and temporal lobe appear respectively at 75 ms and 45 ms. It has been reported in (Mørup et al., 2006b; Arnfred et al., 2011) that proprioceptive stimulus could elicit gamma-band activities (GBA, 30–80 Hz). Our fifth-order tensor decomposition can extract

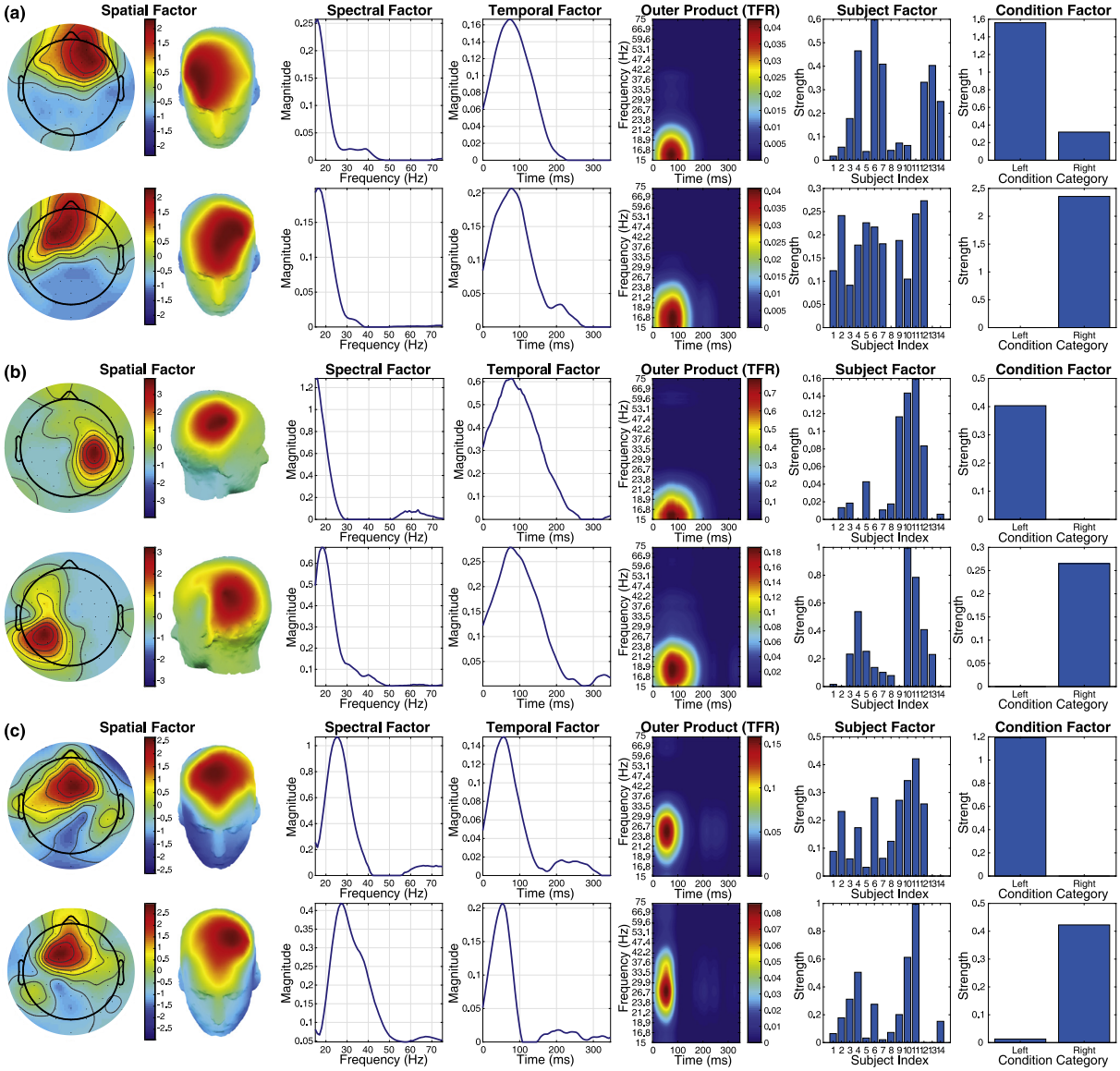



**Fig. 4.** Components of beta-band oscillations. These components have contralateral brain activities on topographies elicited by left-hand and right-hand stimuli respectively. Activities in (a) appear in frontal lobe within 15–20 Hz at 75 ms. Activities in (b) occur in temporal lobe within 15–20 Hz at 75 ms. Activities in (c) emerge in frontal lobe within 25–30 Hz at 60 ms.

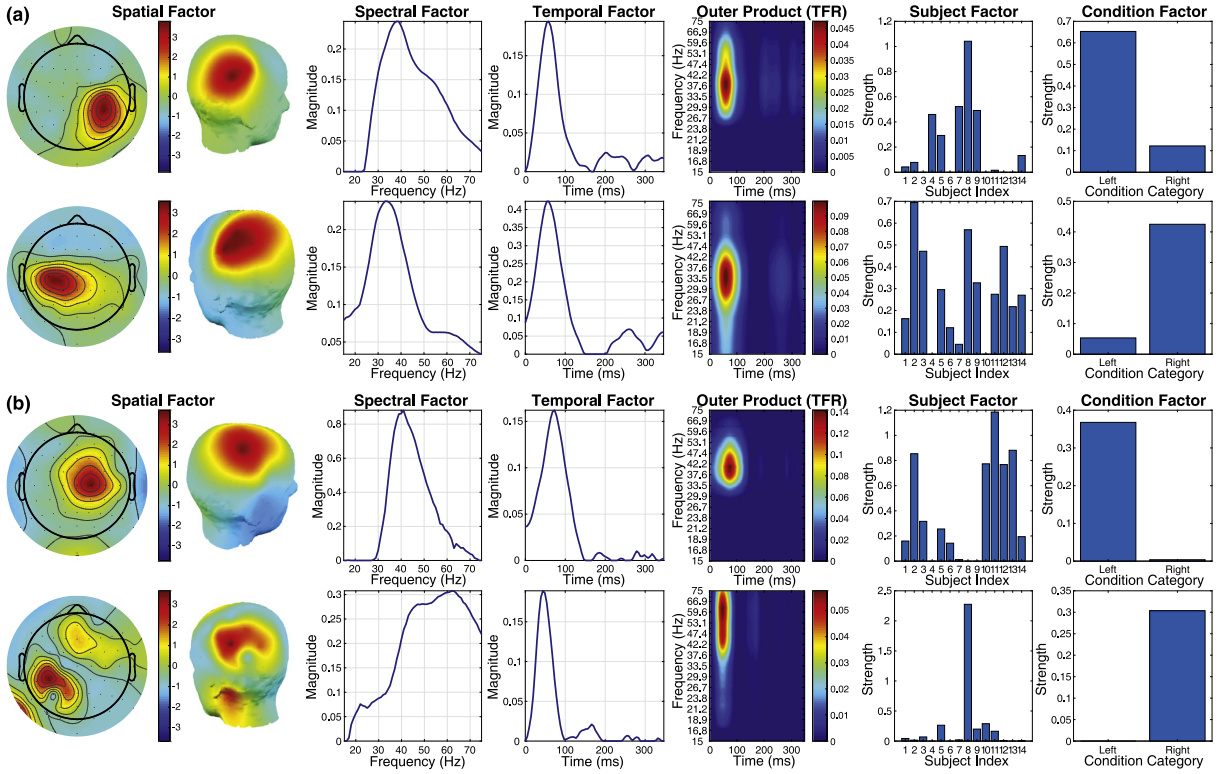



**Fig. 5.** Components of gamma-band oscillations. These components have contralateral brain activities on topographies elicited by left-hand and right-hand stimuli respectively. Activities in (a) appear in frontal lobe within 30–40 Hz at 60 ms. Activity of the first component in (b) occurs between right parietal and temporal lobes at 40 Hz and 75 ms. Activity of the second component in (b) occurs between left parietal and temporal lobes within 40–75 Hz at 45 ms.

more gamma-band components compared with previous study (Mørup et al., 2006b). These gamma-band activities, occurring before 100 ms after stimulus onset in parietal and temporal region, are in accordance with the match step of match-and-utilization model (MUM) in cognitive neuroscience (Herrmann et al., 2004a,b). It is assumed in MUM that the 'early' gamma-band response reflects the matching of stimulus-related information with memory contents that occur rapidly (before 150 ms) after stimulus onset (Herrmann et al., 2004b). We infer that, when a weight increment is exerted on subjects' hand repeatedly during the experiment, there will be a matching and comparison process between the new proprioceptive-stimulus-related signals and existing memory contents. These gamma-band activities in Fig. 5 can be regarded as the results of the matching and comparison process.

In addition, we also discovered two extra components of high strength in both left-hand and right-hand conditions. One of the components occurs in occipital lobe with the frequency of 22 Hz at 80 ms, while another occurs at the center of parietal lobe with the frequency of 28 Hz at 130 ms. Perhaps, these two components were elicited by non-proprioceptive stimulus.

## 5. Conclusion

In this study, we applied fifth-order NCP model to decompose a set of fifth-order ERP tensor data collected by exerting proprioceptive stimulus on left and right hand. The data were fully decomposed in all modes (space, frequency, time, subject, condition), and the interaction information among these modes was well retained. We also proposed an improved smoothed DIFFIT method to select a proper component number for NCP with the most important components being well-preserved after decomposition. Compared with previous study, we discovered more interesting components with symmetric and coherent activities elicited by left-hand and right-hand stimuli. Our findings are consistent with the explanations in related cognitive neuroscience studies. In future studies, some aspects still need further consideration. From the perspective of signal processing algorithm, it would be interesting to compare the performance and stability of NCP tensor decomposition in fifth-order ERP data implemented by different optimization methods, such as APG, MU, ALS, and HALS. From the perspective of cognitive neuroscience methodology, it is worth analyzing experiments results using diverse data forms, such as fifth-order tensor analysis, reshaped third-order tensor analysis, subject-based analysis and group-based analysis.

## Acknowledgment

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 scholarships from China Scholarship Council (Nos. 201600090043 and 201600090042). The authors would like to express their thanks to Lili Tian for refining the written English.

## References

Arnfred, S.M.H., Mørup, M., Thalbitzer, J., Jansson, L., Parnas, J., 2011. Attenuation of beta and gamma oscillations in schizophrenia spectrum patients following hand posture perturbation. Psychiatry Res. 185, 215–224. http://dx.doi.org/10.1016/j.psychres.2009.10.005.

Bro, R., Kiers, H.A.L., 2003. A new efficient method for determining the number of components in PARAFAC models. J. Chemometrics 17, 274–286. http://dx.doi.org/10.1002/cem.801.

Cichocki, A., Mandic, D., Lathauwer, L.D., Zhou, G., Zhao, Q., Caiafa, C., Phan, H.A., 2015. Tensor decompositions for signal processing applications: From two-way to multiway component analysis. IEEE Signal Process. Mag. 32, 145–163. http://dx.doi.org/10.1109/msp.2013.2297439.

Cichocki, A., Zdunek, R., Phan, A.H., Amari, S.-i., 2009. Nonnegative Matrix and Tensor Factorizations: Applications to Exploratory Multi-Way Data Analysis and Blind Source Separation. John Wiley & Sons.

Cohen, M.X., 2014. Analyzing Neural Time Series Data: Theory and Practice. In: Issues in Clinical and Cognitive Neuropsychology, MIT press.

Cong, F., Lin, Q.-H., Kuang, L.-D., Gong, X.-F., Astikainen, P., Ristaniemi, T., 2015. Tensor decomposition of EEG signals: A brief review. J. Neurosci. Methods 248, 59–69. http://dx.doi.org/10.1016/j.jneumeth.2015.03.018.

Cong, F., Phan, A.-H., Astikainen, P., Zhao, Q., Wu, Q., Hietanen, J.K., Ristaniemi, T., Cichocki, A., 2013. Multi-domain feature extraction for small event-related potentials through nonnegative multi-way array decomposition from low dense array EEG. Int. J. Neural Syst. 23, 1350006. http://dx.doi.org/10.1142/s0129065713500068.

Cong, F., Phan, A.H., Zhao, Q., Huttunen-Scott, T., Kaartinen, J., Ristaniemi, T., Lyytinen, H., Cichocki, A., 2012. Benefits of multi-domain feature of mismatch negativity extracted by non-negative tensor factorization from EEG collected by low-density array. Int. J. Neural Syst. 22, 1250025. http://dx.doi.org/10.1142/s0129065712500256.

David, O., Kilner, J.M., Friston, K.J., 2006. Mechanisms of evoked and induced responses in MEG/EEG. NeuroImage 31, 1580–1591. http://dx.doi.org/10.1016/j.neuroimage.2006.02.034.

Delorme, A., Makeig, S., 2004. EEGLAB: an open source toolbox for analysis of single-trial EEG dynamics including independent component analysis. J. Neurosci. Methods 134, 9–21. http://dx.doi.org/10.1016/j.jneumeth.2003.10.009.

Handy, T.C., 2005. Event-Related Potentials: A Methods Handbook. MIT press.

Herrmann, C.S., Lenz, D., Junge, S., Busch, N.A., Maess, B., 2004a. Memory-matches evoke human gamma-responses. BMC Neurosci. 5, 13. http://dx.doi.org/10.1186/1471-2202-5-13.

Herrmann, C.S., Munk, M.H., Engel, A.K., 2004b. Cognitive functions of gamma-band activity: memory match and utilization. Trends Cogn. Sci. 8, 347–355. http://dx.doi.org/10.1016/j.tics.2004.06.006.

Idaji, M.J., Shamsollahi, M.B., Sardouie, S.H., 2017. Higher order spectral regression discriminant analysis (HOSRDA): A tensor feature reduction method for ERP detection. Pattern Recognit. 70, 152–162. http://dx.doi.org/10.1016/j.patcog.2017.05.004.

Mahyari, A.G., Zoltowski, D.M., Bernat, E.M., Aviyente, S., 2017. A tensor decomposition-based approach for detecting dynamic network states from EEG. IEEE Trans. Biomed. Eng. 64, 225–237. http://dx.doi.org/10.1109/tbme.2016.2553960.

Mørup, M., Hansen, L.K., 2009. Automatic relevance determination for multi-way models. J. Chemometrics 23, 352–363. http://dx.doi.org/10.1002/cem.1223.

Mørup, M., Hansen, L.K., Arnfred, S.M., 2007. ERPWAVELAB: A toolbox for multi-channel analysis of time–frequency transformed event related potentials. J. Neurosci. Methods 161, 361–368. http://dx.doi.org/10.1016/j.jneumeth.2006.11.008.

Mørup, M., Hansen, L.K., Herrmann, C.S., Parnas, J., Arnfred, S.M., 2006a. Parallel factor analysis as an exploratory tool for wavelet transformed event-related EEG. NeuroImage 29, 938–947. http://dx.doi.org/10.1016/j.neuroimage.2005.08.005.

Mørup, M., Hansen, L.K., Parnas, J., Arnfred, S.M., 2006b. Decomposing the Time-Frequency Representation of EEG Using Non-Negative Matrix and Multi-Way Factorization. Technical University of Denmark Technical Report, pp. 1–28.

Niknazar, M., Becker, H., Rivet, B., Jutten, C., Comon, P., 2014. Blind source separation of underdetermined mixtures of event-related sources. Signal Process. 101, 52–64. http://dx.doi.org/10.1016/j.sigpro.2014.01.031.

Sidiropoulos, N.D., Lathauwer, L.D., Fu, X., Huang, K., Papalexakis, E.E., Faloutsos, C., 2017. Tensor decomposition for signal processing and machine learning. IEEE Trans. Signal Process. 65, 3551–3582. http://dx.doi.org/10.1109/tsp.2017.2690524.

Timmerman, M.E., Kiers, H.A.L., 2000. Three-mode principal components analysis: Choosing the numbers of components and sensitivity to local optima. Br. J. Math. Statist. Psychol. 53, 1–16. http://dx.doi.org/10.1348/000711000159132.

Vanderperren, K., Mijović, B., Novitskiy, N., Vanrumste, B., Stiers, P., den Bergh, B.R.H.V., Lagae, L., Sunaert, S., Wagemans, J., Huffel, S.V., Vos, M.D., 2012. Single trial ERP reading based on parallel factor analysis. Psychophysiology 50, 97–110. http://dx.doi.org/10.1111/j.1469-8986.2012.01405.x.

Wang, D., Wang, X., Zhu, Y., Toiviainen, P., Huotilainen, M., Ristaniemi, T., Cong, F., 2018. Increasing stability of EEG components extraction using sparsity regularized tensor decomposition. In: Advances in Neural Networks - ISNN 2018. Springer International Publishing, pp. 789–799. http://dx.doi.org/10.1007/978-3-319-92537-0_89.

Weis, M., Romer, F., Haardt, M., Jannek, D., Husar, P., 2009. Multi-dimensional space–time-frequency component analysis of event related EEG data using closed-form PARAFAC. In: 2009 IEEE International Conference on Acoustics, Speech and Signal Processing. IEEE, http://dx.doi.org/10.1109/icassp.2009.4959592.

Xu, Y., Yin, W., 2013. A block coordinate descent method for regularized multiconvex optimization with applications to nonnegative tensor factorization and completion. SIAM J. Imaging Sci. 6, 1758–1789. http://dx.doi.org/10.1137/120887795.

Zhang, Y., Zhou, G., Jin, J., Wang, M., Wang, X., Cichocki, A., 2013. L1-regularized multiway canonical correlation analysis for SSVEP-based BCI. IEEE Trans. Neural Syst. Rehabil. Eng. 21, 887–896. http://dx.doi.org/10.1109/tnsre.2013.2279680.

Zhao, Q., Caiafa, C.F., Mandic, D.P., Zhang, L., Ball, T., Schulze-bonhage, A., Cichocki, A.S., 2011. Multilinear subspace regression: An orthogonal tensor decomposition approach. In: Shawe-Taylor, J., Zemel, R.S., Bartlett, P.L., Pereira, F., Weinberger, K.Q. (Eds.), Advances in Neural Information Processing Systems, vol. 24. Curran Associates, Inc., pp. 1269–1277, URL: http://papers.nips.cc/paper/4328-multilinear-subspace-regression-an-orthogonal-tensor-decomposition-approach.pdf.

Zhou, G., Zhao, Q., Zhang, Y., Adali, T., Xie, S., Cichocki, A., 2016. Linked component analysis from matrices to high-order tensors: Applications to biomedical data. Proc. IEEE 104, 310–331. http://dx.doi.org/10.1109/jproc.2015.2474704.

# PIV

## INCREASING STABILITY OF EEG COMPONENTS EXTRACTION USING SPARSITY REGULARIZED TENSOR DECOMPOSITION

by

Deqing Wang, Xiaoyu Wang, Yongjie Zhu, Petri Toiviainen, Minna Huotilainen, Tapani Ristaniemi and Fengyu Cong 2018

Advances in Neural Networks - the 15th International Symposium on Neural Networks (ISNN 2018), Minsk, Belarus

https://doi.org/10.1007/978-3-319-92537-0_89

# Increasing stability of EEG Components Extraction Using Sparsity Regularized Tensor Decomposition

Deqing Wang[1,2], Xiaoyu Wang[1], Yongjie Zhu[1,2], Petri Toiviainen[3], Minna Huotilainen[4], Tapani Ristaniemi[2], and Fengyu Cong[1,2(✉)]

[1] Department of Biomedical Engineering, Faculty of Electronic Information and Electrical Engineering, Dalian University of Technology, Dalian 116024, China
[2] Faculty of Information Technology, University of Jyväskylä, Jyväskylä 40100, Finland
{deqing.wang, xiaoyu.wang0207, yongjie.zhu}@foxmail.com
cong@dlut.edu.cn, tapani.e.ristaniemi@jyu.fi
[3] Department of Music, Art and Culture Studies, Faculty of Humanities and Social Sciences, University of Jyväskylä, Jyväskylä 40700, Finland
petri.toiviainen@jyu.fi
[4] CICERO Learning network and Cognitive Brain Research Unit, Faculty of Educational Sciences, University of Helsinki, Helsinki 00014, Finland
minna.huotilainen@helsinki.fi

**Abstract.** Tensor decomposition has been widely employed for EEG signal processing in recent years. Constrained and regularized tensor decomposition often attains more meaningful and interpretable results. In this study, we applied sparse nonnegative CANDECOMP/PARAFAC tensor decomposition to ongoing EEG data under naturalistic music stimulus. Interesting temporal, spectral and spatial components highly related with music features were extracted. We explored the ongoing EEG decomposition results and properties in a wide range of sparsity levels, and proposed a paradigm to select reasonable sparsity regularization parameters. The stability of interesting components extraction from fourteen subjects' data was deeply analyzed. Our results demonstrate that appropriate sparsity regularization can increase the stability of interesting components significantly and remove weak components at the same time.

**Keywords:** Tensor Decomposition, Sparse Regularization, Nonnegative Constraints, Ongoing EEG, Stability Analysis.

## 1 Introduction

In recent years, tensor decomposition [1] has gained more and more popularity for EEG data processing [2]. By time-frequency representation, multi-channel EEG data can be converted into a third-order (time × frequency × channel) tensor. In some EEG experiments, a seventh-order tensor may be generated potentially [2]. But sometimes basic tensor decompositions on EEG data can't guarantee meaningful factors. For example, the third-order EEG tensor mentioned above is a nonnegative tensor essentially. The extracted temporal components should be energy series which are

nonnegative. The spectral components should be spectra which are nonnegative and usually very sparse. The spatial components are topographies which are sometimes also sparse. Constrained and regularized tensor decomposition can make the results meaningful and interpretable [1].

Nonnegativity is often achieved by constrained optimization methods. Nonnegative constrains can naturally give sparse results [3], but this sparsity is only a side effect and not controllable. $l_1$-norm regularization is an effective and widely applied method to impose sparsity explicitly [4], and has been employed for sparse and nonnegative tensor decomposition [5-7]. But these works regarding sparse regularization in tensor decomposition [5-7] only tested a few groups of sparsity regularization parameters, and demonstrated that their model can successfully impose sparsity on factors. No work, as far as we know, has studied how the change of sparsity level affects the results of tensor decomposition and the physical meanings in real application.

In this study, ongoing EEG tensor data under naturalistic modern tango music stimulus are analyzed. Sparse nonnegative CANDECOMP/PARAFAC decomposition is employed to extract groups of interesting components whose temporal components are highly correlated with music features and whose spatial components have dipolar topographies. In order to reveal a clear picture of EEG components' properties at different sparsity level, a large range of sparsity regularization parameters are tested. We propose a method to select reasonable regularization parameters that can best balance the data fitting and sparsity. We find that when sparsity regularization is imposed on tensor decomposition, the stability of interesting components is increased significantly. In our results, appropriate sparsity regularization can also remove weak components and elements on nonzero sparse components such as spectra.

## 2 Sparse Nonnegative CANDECOMP/PARAFAC Decomposition

### 2.1 Notation

In this paper, a boldface lowercase letter, such as $\boldsymbol{x}$, denotes a vector; a boldface uppercase letter, such as $\boldsymbol{X}$, denotes a matrix; and a boldface Euler script letter, such as $\boldsymbol{\mathcal{X}}$, denotes a high order tensor. Operator $\circ$ denotes outer product of vectors, and $[\![\ ]\!]$ denotes Kruskal operator. $\|\cdot\|_F$ represents Frobenius norm, and $\|\boldsymbol{x}\|_1$ represents $l_1$-norm of vector $\boldsymbol{x}$. We call sparse nonnegative CANDECOMP/PARAFAC decomposition "sparse NCP" and the version without sparsity regularization "NCP" for short.

### 2.2 Mathematical Model

Given a tensor $\boldsymbol{\mathcal{X}} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$, sparse NCP is to solve the following problem:

$$\min_{\boldsymbol{A}^{(1)},\cdots,\boldsymbol{A}^{(N)}} \frac{1}{2}\left\|\boldsymbol{\mathcal{X}} - [\![\boldsymbol{A}^{(1)},\cdots,\boldsymbol{A}^{(N)}]\!]\right\|_F^2 + \sum_{n=1}^{N} \beta_n \sum_{j=1}^{K} \left\|\boldsymbol{a}_j^{(n)}\right\|_1 \tag{1}$$

$$\text{s.t. } \boldsymbol{A}^{(n)} \geq 0 \text{ for } n = 1,\cdots,N,$$

where $\boldsymbol{A}^{(n)} \in \mathbb{R}^{I_n \times K}$ for $n = 1, \cdots, N$ are estimated factors, and $\boldsymbol{a}_j^{(n)}$ represents the $j$th column of $\boldsymbol{A}^{(n)}$. $\beta_n$ for $n = 1, \cdots, N$ are parameters of sparsity regularization items. $K$ is the selected rank-1 tensor number, and the estimated factors in Kruskal operator can be represented by sum of $K$ rank-1 tensors in outer product form:

$$[\![\boldsymbol{A}^{(1)}, \cdots, \boldsymbol{A}^{(N)}]\!] = \sum_{j=1}^{K} \boldsymbol{a}_j^{(1)} \circ \cdots \circ \boldsymbol{a}_j^{(N)} \tag{2}$$

### 2.3 Model solution

Sparse NCP in (1) is a highly nonlinear and nonconvex model, which is non-trivial to solve and converge. Recently, Xu applied an efficient alternating proximal gradient (APG) method to solve nonnegative matrix and tensor decomposition [8]. Later, Xu extended APG method to solve nonnegative Tucker decomposition with $l_1$-norm sparsity regularizations [7]. Inspired by Xu's work [7], we utilize the same updating method in block coordinate descend (BCD) framework to solve problem (1).

Supposing $\widehat{\boldsymbol{A}}^{(n)}$ is an extrapolated point, $\widehat{\boldsymbol{G}}^{(n)}$ is the block-partial gradient at $\widehat{\boldsymbol{A}}^{(n)}$ and $L^{(n)}$ is a Lipschitz constant, factor matrix $\boldsymbol{A}^{(n)}$ is updated by

$$\boldsymbol{A}^{(n)} \leftarrow \underset{\boldsymbol{A}^{(n)} \geq 0}{\operatorname{argmin}} \left[ \langle \widehat{\boldsymbol{G}}^{(n)}, \boldsymbol{A}^{(n)} - \widehat{\boldsymbol{A}}^{(n)} \rangle + \frac{L^{(n)}}{2} \left\| \boldsymbol{A}^{(n)} - \widehat{\boldsymbol{A}}^{(n)} \right\|_F^2 + \beta_n \sum_{j=1}^{K} \left\| \boldsymbol{a}_j^{(n)} \right\|_1 \right] \tag{3}$$

which can be written in the closed form

$$\boldsymbol{A}^{(n)} \leftarrow \max\left(0, \widehat{\boldsymbol{A}}^{(n)} - \frac{\widehat{\boldsymbol{G}}^{(n)}}{L^{(n)}} - \frac{\beta_n \mathbf{1}_{I_n \times K}}{L^{(n)}}\right) \tag{4}$$

The detailed solution and convergence properties of APG can be found in [7, 8].

## 3 Materials and methods

### 3.1 EEG data and music signal

**Data description.** The data in this study are ongoing EEG of fourteen right-handed and healthy adults under continuous and naturalistic modern tango music stimulus. Short-time Fourier transform (STFT) was applied to the EEG data, and a third-order tensor was created for each subject with size of 510×146×64 on temporal, spectral and spatial mode respectively. In this paper, we represent the estimated factors of the third-order EEG tensor as $\boldsymbol{A}^{(\text{Temporal})}$, $\boldsymbol{A}^{(\text{Spectral})}$ and $\boldsymbol{A}^{(\text{Spatial})}$.

**Acoustic features.** Five long-term acoustic features, including two tonal features (Mode, Key Clarity) and three rhythmic features (Pulse Clarity, Fluctuation Centroid, Fluctuation Entropy), were extracted from the tango music [9]. STFT was also used for feature extraction and one acoustic feature temporal series contains 510 samples.

The detailed data collection experiment paradigm and data preprocessing procedures can be found in [10, 11]. Detailed acoustic features can be found in [9].

### 3.2 Correlation Analysis

According to previous studies [9, 10], we hypothesize that acoustic features (tonal and rhythmic components) can activate certain brain areas. We performed correlation analyses (by Pearson's correlation coefficient) between the time series of long-term acoustic features and the time series of temporal components from EEG tensor decomposition to find stimulus-related activations. Monte Carlo method and permutation tests were employed to compute the threshold of correlation coefficient [9, 10]. In the results of EEG tensor decomposition, the temporal components significantly correlated (at level $p < 0.05$) with any of the five acoustic features, and their corresponding spectral and spatial components were recorded for further investigations.

### 3.3 Sparsity Parameter Selection

When applying sparse NCP model to decompose EEG tensor, a key point is the selection of sparsity regularization parameters $\beta_n$ in model (1), which balance the data fitting and sparsity level. Data fitting at iteration $k$ is defined by

$$\mathrm{Fit}_k = 1 - \frac{\left\| \boldsymbol{\mathcal{X}} - [\![ \boldsymbol{A}_k^{(1)}, \cdots, \boldsymbol{A}_k^{(N)} ]\!] \right\|_F}{\|\boldsymbol{\mathcal{X}}\|_F}, \tag{5}$$

which is a measure of similarity of estimated factors to original data tensor. The sparsity level of estimated factor $\boldsymbol{A}_k^{(n)}$ at iteration $k$ is defined by

$$\mathrm{Sparsity}_{\boldsymbol{A}_k^{(n)}} = \frac{\#\left\{ \boldsymbol{A}_k^{(n)}(i,j) < T_S \right\}}{I_n \times K}, \tag{6}$$

where $\#\{\cdot\}$ means the number of elements in factor $\boldsymbol{A}_k^{(n)}$ that satisfy the assumption. Strictly speaking, the factor sparsity should be measured by the number of elements that equal to zero. But in practice it is better to select a small positive sparsity threshold $T_s$. In this study, we select $T_s = 1\mathrm{e} - 6$.

In order to reveal a broad picture of data properties and results at different sparsity levels, we tested a large range of sparsity regularization parameters of $\beta_n$s for sparse NCP. The procedures are shown in the following steps.

**Step 1.** For the temporal and spatial factor, select $\beta_{\mathrm{temporal}} = \beta_{\mathrm{spatial}} = e^{\boldsymbol{\lambda}_1}$ and $\beta_{\mathrm{spectral}} = e^{\boldsymbol{\lambda}_2}$. $\boldsymbol{\lambda}_1$ and $\boldsymbol{\lambda}_2$ contain $N_1$ and $N_2$ linearized numbers, then there will be $N_1 \times N_2$ different groups of parameter combinations $[\beta_{\mathrm{temporal}}, \beta_{\mathrm{spectral}}, \beta_{\mathrm{spacial}}]$. $\beta_{\mathrm{temporal}}$ and $\beta_{\mathrm{spatial}}$ are set by small numbers, and don't have high sparsity effects. They are made the same in order to reduce the parameter group number.

**Step 2.** Using each group of parameters, run sparse NCP 10 times, and record the average nonzero components number of spectral factor, the average fitting value using (5), and the average sparsity level of spectral factor using (6). Because the spectral components are usually very sparse, only the spectral factor is considered.

**Step 3.** Reorder all of groups of results based on spectral factor sparsity level in ascend order with in $[0,1]$, and generate the 'Fit-Sparsity' curve which reveal the fitting change at different spectral sparsity value.

**Step 4.** According to the 'Fit-Sparsity' curve, identify the maximum effective sparsity level based on the relative fitting change (slope) defined as

$$\text{slope} = \frac{\Delta\text{Fit}}{\Delta\text{Sparsity}} = \frac{\text{Fit}(\text{Sparsity}_1) - \text{Fit}(\text{Spasity}_2)}{\text{Sparsity}_1 - \text{Sparsity}_2}. \tag{7}$$

When the slope at some sparsity point is very close to $0.5$, the sparsity value and its corresponding sparsity parameters group $\beta_n$s are selected. We assume that the slope should not be less than $-0.5$, because after that the fitting value become poor dramatically and the tensor decomposition results may be not accurate.

### 3.4 Stability Analysis

By correlation analysis introduced in section 3.2, we can find the interesting components which are assumed to be stimulus-related activations. Tensor decomposition models of NCP with and without sparsity regularization were evaluated respectively. We evaluated the stability of these components using the following steps.

**Step 1.** Run sparse NCP 5 times for one subject's EEG tensor, and record those groups of temporal, spectral and spatial components whose temporal courses are significantly correlated with any of five music features. Keep those groups of components whose topographies (spatial components) are dipolar as templates [10]. We assume 5 times or more may yield more accurate templates. One group components of template can be represented by a rank-1 tensor of their outer product:

$$\boldsymbol{\mathcal{T}} = \boldsymbol{t}^{(\text{Temporal})} \circ \boldsymbol{t}^{(\text{Spectral})} \circ \boldsymbol{t}^{(\text{Spatial})}$$

**Step 2.** Based on (2), after the tensor decomposition, $K$ rank-1 tensors in outer product form will be obtained. Supposing it is the $r$th time decomposition, according to [12], the correlation coefficient of the $j$th rank-1 tensor and the template is

$$\rho(j,r) = \text{corr}\left(\boldsymbol{a}_j^{(\text{Temporal})}, \boldsymbol{t}^{(\text{Tempral})}\right) \times \text{corr}\left(\boldsymbol{a}_j^{(\text{Spectral})}, \boldsymbol{t}^{(\text{Spectral})}\right) \times \text{corr}\left(\boldsymbol{a}_j^{(\text{Spatial})}, \boldsymbol{t}^{(\text{Spatial})}\right) \tag{8}$$

where $\text{corr}(\cdot,\cdot)$ is the calculation of Pearson's correlation coefficient, $j = 1, \cdots, K$ and $\rho(j,r) \in [0,1]$. Then, calculate the maximum correlation coefficient [12] of the $K$ rank-1 tensors with the template as:

$$P(r) = \max_{j=1,\cdots,K} \rho(j,r) \tag{9}$$

**Step 3.** Run sparse NCP 100 times, and record all $P(r)$, $r = 1, \cdots, 100$. Make a histogram of the 100 $P(r)$ values. Because $P(r) \in [0,1]$, the number of occurrences within $[0.95,1]$ is recorded as stability measurement criterion for further analysis. The higher this criterion number is, the more stable sparse NCP performs.

**Step 4.** Within the 5 times runs in Step 1 many groups of templates that have very similar temporal, spectral and spatial components will be found. The highest criterion number of these templates after 100 times test will be kept.

# 4 Experiments and Results

## 4.1 Tensor Decomposition Implementation

**Factor Initialization.** All the factors were initialized using normally distributed rand numbers as in [8].

**Stop Criteria.** We stopped the iteration during tensor decomposition by criterion of relative residual change (fit change) according to (5), when the following condition between two iterations are satisfied:

$$T_{\text{stop}} = |\text{Fit}_k - \text{Fit}_{k+1}| < \epsilon, \tag{10}$$

where, $\epsilon = 1\text{e} - 6$ in this study.

**Components Number Selection.** Before decomposing each participant's EEG tensor, we should determine the components number for sparse NCP (1). A simple and convenient way was employed in this study. We made spatial mode unfolding of the third-order EEG tensor yielding a 64×74460 matrix where temporal and spectral modes were merged. Then we performed PCA along spatial mode on this matrix and recorded the principal components number that give 99% explained variance for each subject's EEG tensor. The numbers for 14 subjects' EEG tensor are listed in **Table 1**.

**Table 1.** Components number selection of 14 subjects' EEG tensors

| Subject | #1 | #2 | #3 | #4 | #5 | #6 | #7 | #8 | #9 | #10 | #11 | #12 | #13 | #14 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Number | 37 | 44 | 52 | 34 | 38 | 34 | 51 | 38 | 45 | 31 | 57 | 40 | 53 | 43 |

## 4.2 Sparsity Regularization Parameters

For the temporal and spatial factor, we selected $\beta_{\text{temporal}} = \beta_{\text{spatial}} = e^{\boldsymbol{\lambda}_1}$, where vector $\boldsymbol{\lambda}_1 = [-Inf, -5\text{:}\,0.2\text{:}\,0]$ in MATLAB format; for the spectral factor, we selected $\beta_{\text{spectral}} = e^{\boldsymbol{\lambda}_2}$, where $\boldsymbol{\lambda}_2 = [-Inf, -5\text{:}\,0.2\text{:}\,1]$. The linear ranges of $\boldsymbol{\lambda}_1$ and $\boldsymbol{\lambda}_2$ were selected by try and error method, and can be adjusted for different data. In our EEG data test, we found that using exponential form of a linearized vector parameters, the spectral factor sparsity level were approximately uniformly distributed within [0,1], which helps to analyze the fitting change at different sparsity levels. Vector $\boldsymbol{\lambda}_1$ contains 27 numbers, and $\boldsymbol{\lambda}_2$ contains 32 numbers, so there will be 27×32=864 groups of parameters. We expect to add strong sparsity parameters on spectral factor and weak sparsity parameters on temporal and spatial factors. All groups of parameters with $\boldsymbol{\lambda}_1 > \boldsymbol{\lambda}_2$ were removed, and finally, 509 groups were kept for test.

Each of the 509 parameter groups were tested 10 times on sparse NCP, and the average nonzero components number of spectral factor, tensor fitting value, and sparsity level of spectral factor were recorded. Fig. 1. shows the results of subject #1's tensor data after all the steps in Section 3.4. From Fig. 1. (a) we can find that, when more sparsity is added to spectral factor, some components become zeros, which are weak signal components. Fig. 1. (b) shows the tensor decomposition fitting changes at

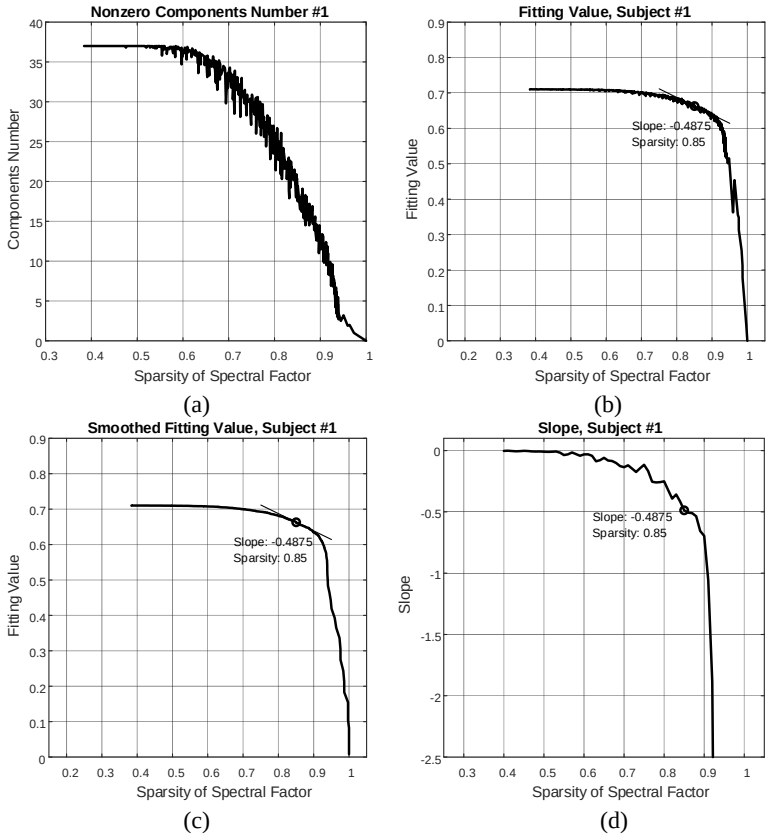



**Fig. 1.** Sparsity regularization parameters selection for subject #1

**Table 2.** Results of sparsity regularization parameters selection

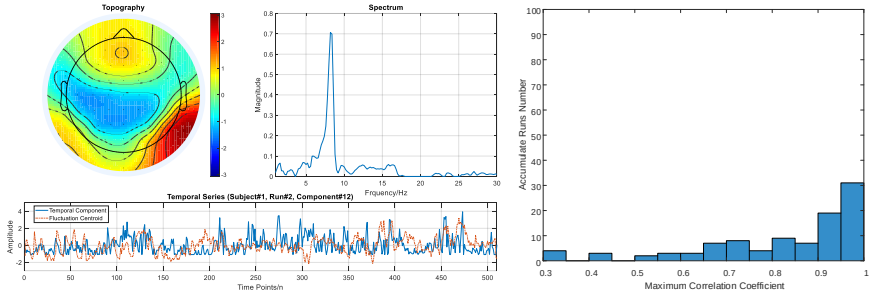
| Subject | Identified Sparsity | Nonzero Comps | Fitting Value | $[e^{\lambda_1}, e^{\lambda_2}, e^{\lambda_1}]$ | $[\lambda_1, \lambda_2, \lambda_1]$ |
|---|---|---|---|---|---|
| #1 | 0.85 | 20.4 | 0.6667 | [0.0608,1.2214,0.0608] | [-2.8,0.2,-2.8] |
| #2 | 0.87 | 28.5 | 0.6991 | [0.0123,0.3012,0.0123] | [-4.4,-1.2,-4.4] |
| #3 | 0.86 | 42.5 | 0.6726 | [0.0000,2.7183,0.0000] | [-Inf,1.0,-Inf] |
| #4 | 0.84 | 28.7 | 0.7683 | [0.0183,0.2466,0.0183] | [-4.0,-1.4,-4.0] |
| #5 | 0.90 | 18.4 | 0.7241 | [0.0150,0.1653,0.0150] | [-4.2,-1.8,-4.2] |
| #6 | 0.86 | 17.4 | 0.8084 | [0.0123,0.0907,0.0123] | [-4.4,-2.4,-4.4] |
| #7 | 0.90 | 31.8 | 0.8095 | [0.0067,0.1353,0.0067] | [-5.0,-2.0,-5.0] |
| #8 | 0.89 | 24.5 | 0.6870 | [0.0183,1.8221,0.0183] | [-4.0,0.6,-4.0] |
| #9 | 0.88 | 28.8 | 0.6811 | [0.0334,0.3679,0.0334] | [-3.4,-1.0,-3.4] |
| #10 | 0.88 | 19.5 | 0.7435 | [0.0498,0.6703,0.0498] | [-3.0,-0.4,-3.0] |
| #11 | 0.93 | 37.2 | 0.7188 | [0.0498,0.4493,0.0498] | [-3.0,-0.8,-3.0] |
| #12 | 0.90 | 19.8 | 0.7110 | [0.0150,1.4918,0.0150] | [-4.2,0.4,-4.2] |
| #13 | 0.92 | 33.4 | 0.6992 | [0.0224,0.2466,0.0224] | [-3.8,-1.4,-3.8] |
| #14 | 0.90 | 20.4 | 0.6899 | [0.0123,0.6703,0.0123] | [-4.4,-0.4,-4.4] |

different spectral factor sparsity levels. When sparsity increases, fitting value decreases gradually, because some weak components are removed. But after some sparsity point, the fitting value drops dramatically. In order to find this point, we smoothed the fitting curve in (b) by low-pass filter, as is shown in (c). Next, we computed the slope according to equation (7), as is shown in (d). Based on curve (d), the largest sparsity level value 0.85 before slope $-0.5$ was selected. Then we searched in all groups of sparsity regularization parameters and found the group $[\boldsymbol{\lambda}_1, \boldsymbol{\lambda}_2, \boldsymbol{\lambda}_1] = [-2.8, 0.2, -2.8]$ could attain the sparsity level 0.85 best. All subjects' results are shown in Table 2.

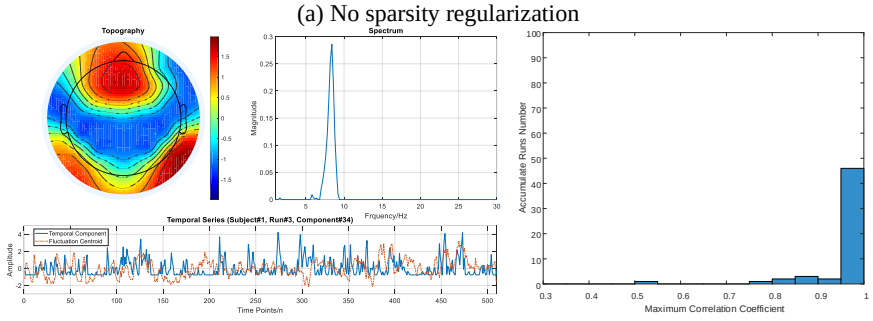
### 4.3 Sparsity and Stability Comparison

From Table 2 we identified the sparsity regularization parameters group $[\beta_{\text{temporal}}, \beta_{\text{spectral}}, \beta_{\text{spacial}}] = [e^{\boldsymbol{\lambda}_1}, e^{\boldsymbol{\lambda}_2}, e^{\boldsymbol{\lambda}_1}]$ for each subject's tensor data, using which we run tensor decomposition model (1). In order to make a comparison, we also tested the original NCP without any sparsity regularizations by setting $[\beta_{\text{temporal}}, \beta_{\text{spectral}}, \beta_{\text{spacial}}] = [0,0,0]$.

Fig 2 shows two templates of subject #1 whose temporal components are both correlated with fluctuation centroid music feature series. One template has sparsity regularization imposed, while another doesn't. The histograms of stability analyses are also included. The template in Fig 2 (a) is selected from the 5th run of NCP and the 14th rank-1 components, and has the highest stability measurement number (the number between [0.95,1] on histogram) than other similar templates in the 5 runs.




(a) No sparsity regularization




(b) Sparsity regularization imposed

**Fig. 2.** Fluctuation centroid templates and their stability analysis from subject #1

**Table 3.** Stability analyses of sparsity regularization for all subjects

| Subject | Music Feature | No Sparsity Regularization | | With Sparsity Regularization | |
|---|---|---|---|---|---|
| | | Template Index | Stability | Template Index | Stability |
| #1 | PulseClarity | [Run#5, Comp#14] | 42% | [Run#5, Comp#23] | **74%** |
| | FlucCentroid | [Run#2, Comp#12] | 31% | [Run#3, Comp#34] | **46%** |
| #4 | FlucCentroid | [Run#3, Comp#9] | 86% | [Run#1, Comp#24] | **97%** |
| #7 | Key | [Run#1, Comp#21] | 57% | [Run#2, Comp#30] | **63%** |
| #8 | Mode | [Run#4, Comp#35] | 29% | [Run#1, Comp#33] | **60%** |
| | Mode | [Run#5, Comp#3] | 16% | [Run#3, Comp#15] | **68%** |
| | Key | [Run#2, Comp#37] | 17% | [Run#1, Comp#22] | **57%** |
| #9 | Mode | [Run#3, Comp#37] | 10% | [Run#4, Comp#23] | **54%** |
| #10 | PulseClarity | [Run#4, Comp#27] | 48% | [Run#2, Comp#25] | **49%** |
| | Key | [Run#4, Comp#1] | **57%** | [Run#2, Comp#20] | 51% |
| #11 | FlucCentroid | [Run#5, Comp#55] | 14% | [Run#3, Comp#40] | **55%** |
| | Mode | [Run#3, Comp#1] | **73%** | [Run#1, Comp#56] | 48% |
| #13 | FlucEntropy | [Run#2, Comp#17] | 29% | [Run#1, Comp#34] | **58%** |
| | FlucCentroid | [Run#1, Comp#5] | 16% | [Run#5, Comp#52] | **37%** |
| | Key | [Run#3, Comp#51] | 30% | [Run#2, Comp#20] | **72%** |

The template in Fig 2 (b) is selected from the 5th run of sparse NCP and the 23th rank-1 components. All of the results for 14 subjects are summarized in Table 3 with selected templates and stability analyses[1]. The selected pairs of templates for comparison have very similar topographies and spectra, and appear in both situations with and without sparsity imposed on spectral components.

Comparing the histograms in Fig 2 (a) and (b), we find that, with sparsity regularization the stabilities of selected templates components have obvious increases. Table 3 further demonstrates that, for most of the subjects, when sparsity regularization is imposed on NCP for EEG data, the stability of extracting interesting components highly correlated with some certain music features are improved significantly.

From Fig 1 (a) we observe that when high sparsity regularization is imposed on spectral factor, more components of full zeros appear. This is a general phenomenon for all subjects according to the nonzero components number in Table 2 and the selected components numbers in Table 1. By carefully observing the spectrum in Fig 2 (b) compared to that in (a), we also find that small elements are suppressed when sparsity regularization is imposed. We believe adding sparsity regularization not only can remove weak components but also help to suppress weak information on nonzero components for nonnegative tensor decomposition.

## 5 Conclusion

In this study, we applied sparse nonnegative CANDECOMP/PARAFAC decomposition to ongoing EEG tensor data collected under naturalistic music stimulus. Interest-

[1] The stability was measured by number of maximum correlation coefficient within [0.95,1]. But there are two exceptions of "#11-FlucCentroid" and "#13-FlucCentroid", which were measured within [0.9,0.95], because no significant value appeared within [0.95,1].

ing temporal components correlated with music features and corresponding spectral and spatial components were extracted. Mathematical properties and physical meanings of the decomposition with sparsity regularization were deeply analyzed in a large range of sparsity levels. We proposed a method to select reasonable sparsity regularization parameters based on the derivative of fitting-sparsity curve. It can be concluded from our results that appropriate sparsity regularization on tensor decomposition can improve the stability of interesting components and suppress weak signals.

**Acknowledgements.** 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 scholarships from China Scholarship Council (Nos. 201600090043 and 201600090042).

## References

1. Sidiropoulos, N.D., De Lathauwer, L., Fu, X., et al.: Tensor decomposition for signal processing and machine learning. IEEE Transactions on Signal Processing. 65(13), 3551-3582 (2017).
2. Cong, F., Lin, Q.-H., Kuang, L.-D., et al.: Tensor decomposition of EEG signals: a brief review. Journal of neuroscience methods. 248, 59-69 (2015).
3. Lee, D.D. and Seung, H.S.: Learning the parts of objects by non-negative matrix factorization. Nature. 401(6755), 788-791 (1999).
4. Donoho, D.L.: For most large underdetermined systems of linear equations the minimal L1 ‐ norm solution is also the sparsest solution. Communications on pure and applied mathematics. 59(6), 797-829 (2006).
5. Mørup, M., Hansen, L.K., and Arnfred, S.M.: Algorithms for sparse nonnegative Tucker decompositions. Neural computation. 20(8), 2112-2131 (2008).
6. Liu, J., Liu, J., Wonka, P., et al.: Sparse non-negative tensor factorization using columnwise coordinate descent. Pattern Recognition. 45(1), 649-656 (2012).
7. Xu, Y.: Alternating proximal gradient method for sparse nonnegative Tucker decomposition. Mathematical Programming Computation. 7(1), 39-70 (2015).
8. Xu, Y. and Yin, W.: A block coordinate descent method for regularized multiconvex optimization with applications to nonnegative tensor factorization and completion. SIAM Journal on imaging sciences. 6(3), 1758-1789 (2013).
9. Alluri, V., Toiviainen, P., Jääskeläinen, I.P., et al.: Large-scale brain networks emerge from dynamic processing of musical timbre, key and rhythm. Neuroimage. 59(4), 3677-3689 (2012).
10. Cong, F., Alluri, V., Nandi, A.K., et al.: Linking brain responses to naturalistic music through analysis of ongoing EEG and stimulus features. IEEE Transactions on Multimedia. 15(5), 1060-1069 (2013).
11. Wang, D., Cong, F., Zhao, Q., et al.: Exploiting ongoing EEG with multilinear partial least squares during free-listening to music. In: 2016 IEEE 26th International Workshop on Machine Learning for Signal Processing (MLSP), pp. 1-6. IEEE, Salerno, Italy (2016).
12. Cong, F., Phan, A.H., Zhao, Q., et al.: 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. 22(06), 1250025 (2012).

# PV

## EXPLOITING ONGOING EEG WITH MULTILINEAR PARTIAL LEAST SQUARES DURING FREE-LISTENING TO MUSIC

by

Deqing Wang, Fengyu Cong, Qibin Zhao, Petri Toiviainen, Asoke K. Nandi, Minna Huotilainen, Tapani Ristaniemi and Andrzej Cichocki 2016

2016 IEEE International Workshop on Machine Learning for Signal Processing (MLSP), Salerno, Italy

https://doi.org/10.1109/MLSP.2016.7738849

2016 IEEE INTERNATIONAL WORKSHOP ON MACHINE LEARNING FOR SIGNAL PROCESSING, SEPT. 13–16, 2016, SALERNO, ITALY

# EXPLOITING ONGOING EEG WITH MULTILINEAR PARTIAL LEAST SQUARES DURING FREE-LISTENING TO MUSIC

*Deqing Wang[1,2], Fengyu Cong[1,2], Qibin Zhao[3], Petri Toiviainen[4], Asoke K. Nandi[5,6], Minna Huotilainen[7] Tapani Ristaniemi[2], Andrzej Cichocki[3]*

1. Department of Biomedical Engineering, Faculty of Electronic Information and Electrical Engineering, Dalian University of Technology, China
2. Department of Mathematical Information Technology, University of Jyväskylä, Finland
3. Laboratory for Advanced Brain Signal Processing, RIKEN Brain Science Institute, Japan
4. Finnish Centre of Excellence in Interdisciplinary Music Research, University of Jyväskylä, Finland
5. Department of Electronic and Computer Engineering, Brunel University, UK
6. College of Electronics and Information Engineering, Tongji University, Shanghai, China
7. Finnish Institute of Occupational Health, Helsinki, Finland

## ABSTRACT

During real-world experiences, determining the stimulus-relevant brain activity is excitingly attractive and is very challenging, particularly in electroencephalography. Here, spectrograms of ongoing electroencephalogram (EEG) of one participant constructed a third-order tensor with three factors of time, frequency and space; and the stimulus data consisting of acoustical features derived from the naturalistic and continuous music formulated a matrix with two factors of time and the number of features. Thus, the multilinear partial least squares (PLS) conforming to the canonical polyadic (CP) model was performed on the tensor and the matrix for decomposing the ongoing EEG. Consequently, we found that brain activity of majority of participants was significantly correlated with the musical features in time domain, and that such brain activity showed frontal or central or posterior or occipital distributions along the scalp, and that such brain activity could be of different oscillation bands in frequency domain.

***Index Terms—*** Ongoing EEG, music, multilinear partial least squares, tensor decomposition,

## 1. INTRODUCTION

Electroencephalography is one of the most commonly used brain imaging methods since 1920s [1]. The corresponding recording by electrodes along the scalp, electroencephalogram (EEG), can be divided into three categories including spontaneous EEG without any stimulus, for example, under the resting state or sleeping [2], event-related potentials (ERPs) elicited by controlled and rapidly repeated stimuli [3], and ongoing EEG during real-world experiences [4, 5]. ERPs indeed can be further grouped into averaged EEG data over single trials and single-trial EEG data, and the former is mainly for cognitive neuroscience research [3] and the latter has been largely used in brain computer interface [6]. In contrast to spontaneous EEG and ERPs, ongoing EEG is not well studied yet, mainly due to the difficulty for data processing and analysis, as well as a lack of prior knowledge with regard to such data.

During real-world experience, one of interesting topics is to determine the stimulus-relevant brain activity. The most straightforward way is to analyze correlation coefficient of two temporal courses representing the oscillations of EEG data and acoustic features of stimulus data [4]. Actually, relationship between two sets of variables representing by matrices and vectors can be also revealed by canonical correlation coefficient (CCA) [7] and partial least squares (PLS) [8] by maximizing coherence and covariance. When the two sets of the variables are multi-way data, multilinear PLS, the extension of PLS, can provide such an opportunity to analyze interactions of all modes directly [9].

Multilinear PLS can be used for the analysis of one tensor and one matrix or two tensors [9]. Spectrograms of spontaneous EEG data can be represented by a third-order tensor with factors of time, frequency and space. Multilinear PLS was used to analyze the EEG data and fMRI data represented by a matrix with factors of time and space, and related information between two modalities of brain data were examined [10]. In another study [11] spontaneous EEG and single-trial ERP data were represented by a third-order tensor with factors of epoch, feature and space, and the feature factor included features of EEG in different domains. Multilinear PLS was performed on the tensor and labels of different epochs for seizure recognition [11] and also for the simultaneous EEG and electromyography (EMG) [12].

Hence, like PLS, multilinear PLS can be used as PLS correlation and PLS regression in brain imaging [13].

In this study for each participant in a free-listening experiment using a 512-second modern tango, we performed multilinear PLS on a third-order tensor (with factors of time, frequency and space of ongoing EEG elicited by naturalistic and continuous music) and a matrix of musical features. For comparison, we also applied PLS on the vectorized brain data and stimulus data since it has been applied to analyze brain imaging data widely [13]. We found music-associated patterns of brain activity of majority of participants in such naturalistic and continuous brain data.

## 2. METHOD

### 2.1 Data description

Fourteen right-handed and healthy adults aged 20 to 46 years old in Finland participated in the free-listening experiment and none of participants had musical expertise. The stimulus was an 8.5-minute long musical piece of modern tango by Astor Piazzolla [4]. During the experiment, they were told to listen to music and sit as still as possible with eyes open. Ongoing EEG data were collected by 10-20 system with BioSemi bioactive electrode caps (64 electrodes in the cap plus 5 external electrodes at the tip of the nose, left and right mastoids and around the right eye both horizontally and vertically). Between each measuring electrode and the Common Mode Sense electrode, the direct-current mean value was kept under ±25 $\mu$V. The reference was the external electrode of the nose, and EEG were collected with the sampling rate of 2048 Hz for off-line processing.

Ongoing EEG data of our study were first preprocessed using EEGLAB [14]. They were down-sampled to 256 Hz to reduce the size of datasets without losing important data information. Subsequently, the 1 Hz high-pass and 30 Hz low-pass filters were applied on data of each participant at each channel. Next, the ongoing EEG data were visually checked and no obvious artifacts from head movement were found. The data were then used for further analysis. In order to remove EOG (i.e., eye blinks), independent component analysis (ICA) was performed on ongoing EEG data of each participant. The application of ICA in this dataset has been introduced in [4]. Hereinafter, when ongoing EEG is mentioned, it means the preprocessed one.

### 2.2 Third-order tensor of ongoing EEG

The short-time Fourier transform (STFT) was applied to the ongoing EEG in order to obtain the spectrogram, i.e., time-frequency representation. The duration of the window was three seconds, and the overlap between two adjacent windows was two seconds. This was consistent with acoustical feature extraction [4]. In addition, a Hamming window was used to segment the ongoing EEG, and the number of points for Fourier transform was 1284 which was five times the sampling frequency (i.e., 256 Hz). As the data were filtered by 1 Hz high-pass and 30 Hz low-pass filters, the spectrogram between 1 Hz and 30 Hz was used for further analysis. Later, a third-order tensor of ongoing EEG was created for each participant. Factors of the tensor were time (510 samples), frequency (146 bins) and space (64 electrodes).

### 2.3 Musical features

Five acoustic features [15] were examined as tonal and rhythmic musical features here. Acoustical features were extracted using a frame-wise approach. The duration of each frame was 3 seconds and the overlap between two adjacent frames was two seconds. Therefore, one temporal course with 510 samples was produced for each feature.

The five acoustic features can be divided into two groups including two tonal and three rhythmic musical features. The tonal ones consist of Mode (strength of major or minor mode) and Key Clarity (the measure of the tonal clarity), while the rhythmic features include Fluctuation Centroid, Fluctuation Entropy, and Pulse Clarity [15]. Fluctuation centroid is the geometric mean of the fluctuation spectrum. It is for the global repartition of rhythm periodicities within the range of 0–10, indicating the average frequency of these periodicities. Fluctuation entropy is the Shannon entropy of the fluctuation spectrum. It represents the global repartition of rhythm periodicities and is a measure of the noisiness of the fluctuation spectrum. Pulse Clarity naturally estimates clarity of the pulse. Hereinafter, acoustical features of music are called musical features.

### 2.4 Partial least squares

The standard PLS [13] seeks the common latent vectors from brain data $\mathbf{X} \in \mathfrak{R}^{I\times K}$ and behavior data $\mathbf{Y} \in \mathfrak{R}^{I\times J}$, and the constraint is that those latent vectors mostly explain the covariance between the brain and the behavior data. Hence, the standard PLS reads

$$\mathbf{X} = \mathbf{T}\mathbf{P}^T + \mathbf{E}_\mathbf{X}, \quad (1)$$

$$\mathbf{Y} = \mathbf{U}\mathbf{C}^T + \mathbf{E}_\mathbf{Y}, \quad (2)$$

where $\mathbf{T} = [\mathbf{t}_1, \mathbf{t}_2, \cdots, \mathbf{t}_R] \in \mathfrak{R}^{I\times R}$, $\mathbf{T}^T\mathbf{T} = \mathbf{I}$, $\mathbf{I}$ is the identity matrix, $\mathbf{T}$ includes the extracted orthonormal latent vectors from brain data $\mathbf{X}$; $\mathbf{U} = [\mathbf{u}_1, \mathbf{u}_2, \cdots, \mathbf{u}_R] \in \mathfrak{R}^{I\times R}$ consists of latent vectors from behavior data $\mathbf{Y}$; $\mathbf{P}$ and $\mathbf{C}$ correspond to loadings; and $\mathbf{E}_\mathbf{X}$ and $\mathbf{E}_\mathbf{Y}$ represent the residuals from brain data $\mathbf{X}$ and behavior data $\mathbf{Y}$, respectively. By PLS, $\mathbf{U}$ column-wisely has the maximum covariance with $\mathbf{T}$, and the simplest model to mostly represent $\mathbf{Y}$ is given by

$$\mathbf{U} \approx \mathbf{T}\mathbf{D}, \quad (3)$$

where $\mathbf{D}$ is a diagonal matrix and its elements are $d_{rr} = \mathbf{u}_r^T\mathbf{t}_r / \mathbf{t}_r^T\mathbf{t}_r (r = 1, \cdots, R)$ which are the regression coefficients to predict $\mathbf{Y}$.

### 2.5 Multilinear partial least squares

In our study, a third order tensor $\underline{\mathbf{X}} \in \Re^{I\times K\times M}$ represents brain activity with factors of time, frequency and space, and musical stimulus data are denoted by a matrix $\mathbf{Y} \in \Re^{I\times J}$. Multilinear PLS in the form of $N$-way PLS ($N$-PLS) [9, 16] is used in this study. $N$-PLS is indeed based on the canonical polyadic (CP) model [17]. Thus, the tensor $\underline{\mathbf{X}}$ and the matrix $\mathbf{Y}$ in $N$-PLS can be decomposed as below,

$$\underline{\mathbf{X}} = \sum_{r=1}^{R} \mathbf{t}_r \circ \mathbf{p}_r \circ \mathbf{q}_r + \underline{\mathbf{E}}_{\underline{\mathbf{X}}} \quad (4)$$

$$\mathbf{Y} = \sum_{r=1}^{R} d_{rr}\mathbf{t}_r\mathbf{c}_r^T + \mathbf{E}_{\mathbf{Y}}, \quad (5)$$

where '$\circ$' denotes the outer product of two vectors, $\mathbf{t}_r \in \Re^{I\times 1}$ represents the temporal component and is shared by $\underline{\mathbf{X}}$ and $\mathbf{Y}$, $\mathbf{p}_r \in \Re^{K\times 1}$ is the spectral component, $\mathbf{q}_r \in \Re^{M\times 1}$ denotes the spatial component of ongoing EEG, $d_{rr} = \mathbf{u}_r^T\mathbf{t}_r / \mathbf{t}_r^T\mathbf{t}_r (r = 1, \cdots, R)$, and $\mathbf{u}_r = \mathbf{Y}\mathbf{c}_r$. For simplicity, we remove index $r$ and present how each component is sequentially optimized. Under $N$-PLS, $\forall\, r, r = 1, \cdots, R$, the vectors $\mathbf{p}$, $\mathbf{q}$, and $\mathbf{c}$ satisfy

$$\{\mathbf{p}, \mathbf{q}, \mathbf{c}\} = \arg\max_{\widetilde{\mathbf{p},\mathbf{q},\mathbf{c}}} [\mathrm{cov}(\mathbf{t}, \mathbf{u})], \quad (6)$$

where s.t. $t_i = \sum_{k=1}^{K}\sum_{m=1}^{M} x_{ikm}p_k q_m$, $\|\mathbf{p}\|_2^2 = \|\mathbf{q}\|_2^2 = 1$.

Multilinear PLS was first implemented through alternating least squares (ALS) [9, 16]. Recently, we found that the fast higher-order orthogonal iteration (HOOI) based higher-order partial least squares (HOPLS) [18] can also be used for $N$-PLS due to high computing efficiency. Therefore, the latter is applied for $N$-PLS here instead of ALS.

It should be noted that the variance and polarity of an extracted component by multilinear PLS are indeterminate, which exists inherently in most of matrix and tensor decompositions. In this study, we do not aim to correct such indeterminacy since we do not compare power of brain activity among different participants.

### 2.6 Data processing for PLS

In order to reveal the strength of multilinear PLS, the standard PLS was applied to decompose the brain data and stimulus data here. For PLS, factors of frequency and space were vectorized, therefore, the third-order tensor was reshaped into a matrix with the factors of time and frequency-space. The number of extracted components for PLS was referenced to the number for multilinear PLS (see the next sub-section). After decomposition, each frequency-space component represented by a vector was reshaped into a matrix with the factors of frequency and space. Then, according to the frequency band of each oscillation among delta (1-4Hz), theta (4-8Hz), alpha (8-13Hz) and beta (13-30Hz), the integral over an oscillation frequency band in the matrix was applied to produce the spatial component of one oscillation. Next, residual dipole variance (RDV) in brain imaging was analyzed for each spatial component to examine the reliability of the component because reliable brain activity is assumed to be dipolar [19].

### 2.7 Data processing for multilinear PLS

Based on a tensor representation of brain activity and a matrix of the music stimulus data, multilinear PLS was performed to extract temporal, spectral and spatial components of brain activity given a predefined number of components. Then, reliable spatial components were selected according to the analysis of RDV. Correspondingly, the temporal and spectral components were automatically chosen due to the applied CP model for multilinear PLS. Next, the selected temporal components were correlated with each musical feature of the music stimulus, and the threshold for significant correlation coefficients was determined. Subsequently, the temporal components significantly correlated with musical features were chosen, as well as the corresponding spectral and spatial components in the CP model. Hence, determining brain activity relevant to the music stimulus depended on two criteria: whether its spatial map showed dipolar brain activity, and whether its temporal component was significantly correlated with any of musical features of the music stimulus.

Following this process, brain activity relevant to the music stimulus can be discovered and its multi-factor patterns in the time, frequency and spatial domains can be known for each participant individually.

### 2.8 Data decomposition

The number of components in multi-way data decomposition is a critical parameter. This study implemented the method called DIFFIT (difference of fit) [20] to determine it. Here, fits of stimulus data were used for DIFFIT and the numbers of components extracted by multilinear PLS ranged from 2 to 70. Then, DIFFIT suggested one number of components, $R$, for decomposition. In this study, before decomposition, each temporal course was centered both in brain data and stimulus data. Singular value decomposition (SVD) was used to initialize the factor matrices for HOOI. Due to this determined initialization, the decomposition was only run once in this study, and the first $R$ temporal components from ongoing EEG, as well as the corresponding spectral and spatial components, were used for further analysis after the implementation of DIFFIT.

### 2.9 Residual dipole variances of spatial components

When EEG data are decomposed by linear or multilinear transformations, the reliability of an extracted component should be examined to determine the appropriateness of the decomposition procedure. Recently, reliability of an ICA component has been investigated with a single equivalent dipole model in brain imaging [19]. It has been assumed that brain activity should be dipolar, resulting in low RDV, when a spatial component along scalp, i.e., topography, is used for source localization. Although the single equivalent dipole model may not be accurate enough, many spatial

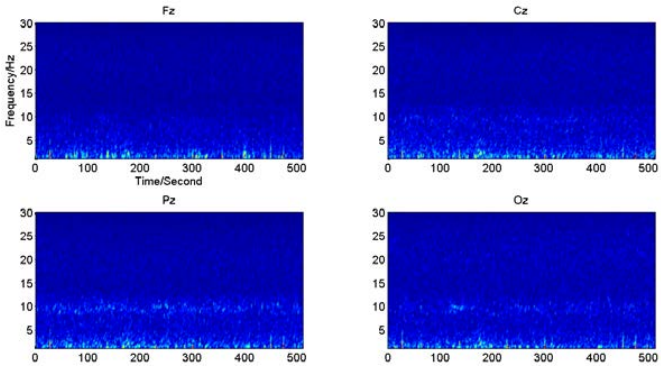


Fig.1 Spectrogram of ongoing EEG

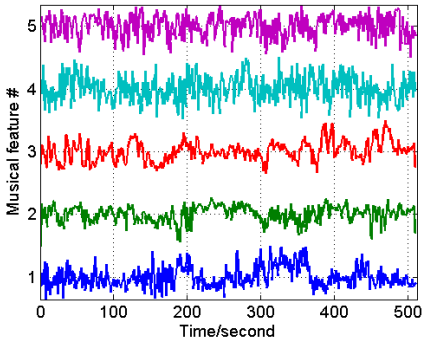


Fig.2 Temporal courses of five musical features

components with high RDV can be immediately rejected since their possibilities showing real brain activity are very low. Furthermore, as mentioned earlier, we do not have enough knowledge from ongoing EEG data elicited by naturalistic and continuous music. Therefore, such an approach can assist us to find relatively reliable spatial components. Here, spatial components with RDV less than 10% were chosen for further analysis, as well as the corresponding temporal and spectral components. Such a threshold was used in [19].

### 2.10 Threshold for correlation coefficient

Statistically, examination of the significance of the correlation coefficient between two temporal courses is important. One threshold was derived from one musical feature and $L$ ($L$ is the number of diploe spatial maps among $R$ extracted spatial components by PLS or multilinear PLS, $L \leq R$) temporal courses for each participant with correction for multiple comparisons based on Monte-Carlo method [15] and the permutation test procedure. Temporal components of interest were determined as the ones possessing significant correlations ($p < 0.001$) with musical features.

### 2.11 Group-level data analysis

Given one musical feature of the music stimulus, the temporal components from ongoing EEG (significantly correlated with the musical feature) from different participants can be found. Subsequently, the similarity and dissimilarity of the corresponding spatial components from those participants can be revealed, and so do the associated spectral components. Here, if more than half of all participants possess such temporal components given any musical feature, brain activity and the musical feature are reported hereinafter.

## 3. RESULTS

### 3.1 Spectrogram of ongoing EEG

Fig.1 shows the spectrogram of averaged ongoing EEG over 14 participants in the experiment at four representative electrodes. The alpha oscillation clearly presented at Pz. However, the correlation coefficient between the temporal courses of EEG oscillations and the temporal courses of musical features shown in Fig.2 was not significant for most of participants [4]. Thus, no further analysis was done.

### 3.2 Decomposition of PLS

As we have mentioned in last section, the number of extracted components by PLS was referenced to multilinear PLS. For each oscillation, among all 514 (the mean over 14 participants was about 37 and the standard deviation (SD) was about 15) extracted spatial components by PLS for all 14 participants, the number of spatial components with RDV less than 10% was very small. They were 19 (about 4% of 514) for delta, 32 (about 6% of 514) for theta, 75 (about 15% of 514) for alpha, and 21 (about 4% of 514) for beta. Since the number of reliable spatial components was not considered to be enough for any oscillation, we did not analyze the corresponding temporal or spectral components.

### 3.3 Decomposition of multilinear PLS

***Residual diploe variances of spatial components***

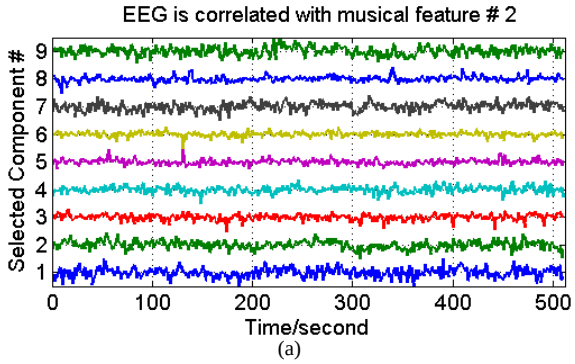
The number of spatial components with RDV less than 10% was 97 (the mean over 14 participants was about 7 and SD was about 4, and each participant had at least one component). This means about 19% of all 514 spatial components of all participants showed dipolar brain activity. Such results were comparable with several blind source separation methods for ERPs [19].

Then, the 97 spatial components' corresponding temporal components and spectral components were further analyzed.

***Brain activity relevant to music***

For the musical feature of the music stimulus, Pulse Clarity, the number of participants whose temporal components significantly correlated with it was five, and was less than half of 14 participants. Therefore, the components relevant to this feature are not shown hereinafter. For any of other four features, the number of participants whose temporal components significantly correlated with it was at least eight.

Figs. 2 and 3 respectively show the five musical features and selected temporal components which were significantly correlated with the musical feature #2, as well as their corresponding spatial and spectral components. It should be noted these temporal components were selected from all 514 components of all 14 participants, and their corresponding spatial maps showed dipolar brain activity. Each temporal course/component was centered and normalized to its own SD. The averaged correlation coefficient between the musical feature 'Fluctuation Entropy' and the selected temporal components was 0.26 (SD: 0.058). For 'Fluctuation Centroid', 'Mode', and 'Key', they were 0.31 (SD: 0.058), 0.21 (SD: 0.029), and 0.22 (SD: 0.043), respectively.



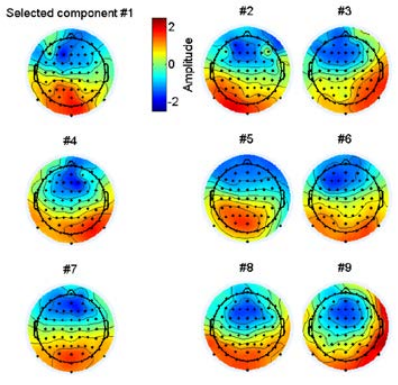

(a)

As for the corresponding spectral components associated with the selected temporal components in Fig. 3, power of different spectral components from different participants maximized at different frequency ranges. In contrast to the spectrum of ongoing EEG shown in Fig. 1, brain activity relevant to music appeared with higher possibility in higher frequency bands by multilinear PLS although power of low frequency bands dominated the ongoing EEG in the electrode field.

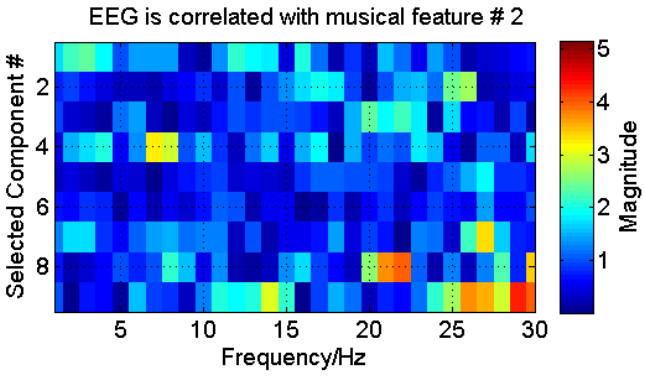
## 4. DISCUSSION

We proposed an efficient approach in order to decompose ongoing EEG in a free-listening experiment mainly by multilinear PLS conforming to the canonical polyadic (CP) model. From ongoing EEG of normal adults listening to a piece of 512-second modern tango, we found brain activity of interest which was significantly correlated with the music stimulus. Actually, such coherence between ongoing EEG and music can be expected since theta and alpha oscillations in an ICA study on the same dataset were found significantly associated with the same music [4].

The proposed approach is different from the previous reports where temporal courses of EEG oscillations based on spectrogram of ICA components [4] and temporal components extracted by tensor decomposition from just the EEG tenor [5] were correlated with musical features. As shown in Eq. (4), the proposed approach using multilinear PLS allows examination of spectral components relevant to the music stimulus, providing the opportunity to observe the spectral structure of brain activity particularly of wider frequency band in the frequency domain. Furthermore, without surprise, multilinear PLS outperformed standard PLS here. If the beta oscillation, which ranges from 13Hz to 30Hz, was calculated based on the integral of frequency in this range in the spectrogram of ongoing EEG, it would be unknown in which specific sub-band brain activity could be in the frequency domain.




(b)




(c)

Fig.3 Selected temporal components (a) which were significantly correlated with four musical feature #2 'Fluctuation Centroid', as well as their corresponding spatial (b) and spectral components (c).

With the proposed data processing approach, patterns of brain activity relevant to the music stimulus across majority of participants were found diverse both in different oscillation bands in the frequency domain and in hemispheres and locations along the scalp in the spatial domain. Generally speaking, in ERP and spontaneous EEG experiments, it is expected that different participants in one group can share similar patterns of brain activity in the time, frequency, and spatial domains. Such difference between ERP/spontaneous EEG and ongoing EEG is assumed to be associated with experimental designs. In the free-listening experiment, natural brain activity is recorded due to the naturalistic and continuous auditory stimulus, and it is reasonable that different people respond to the same stimulus differently. Indeed, brain activity of different participants in the same group shares a similar pattern is desired when the goal is to reveal the difference in brain activity among different groups. For such a purpose, different constraints including independence, nonnegativity, and sparsity, and so on, can be exploited when ongoing EEG data are decomposed.

## 5. ACKNOWLEDGEMENT

This work was partially supported by TEKES (Finland) grant 40334/10 ‘Machine Learning for Future Music and Learning Technologies’, the Fundamental Research Funds for the Central Universities [DUT16JJ(G)03] in Dalian University of Technology in China, and National Natural Science Foundation of China (Grant No. 81471742). Cong F. thanked Dr. Elvira Brattico and Dr. Vinoo Alluri for data collection and discussion about the reliability of findings of ongoing EEG.

## 6. REFERENCES

[1] H. Berger, "Ueber das Elektrenkephalogramm des Menschen," *Archives fur Psychiatrie Nervenkrankheiten*, vol. 87, pp. 527-570, 1929.

[2] E. Niedermeyer and F. Lopes da Silva, *Electroencephalography : basic principles, clinical applications, and related fields*. Baltimore, MD: Williams & Wilkins, 2005.

[3] S. J. Luck, *An Introduction to the Event-Related Potential Technique*, 2nd ed.: The MIT Press, 2014.

[4] F. Cong, V. Alluri, A. K. Nandi, P. Toiviainen, R. Fa, B. Abu-Jamous, *et al.*, "Linking Brain Responses to Naturalistic Music through Analysis of Ongoing EEG and Stimulus Features," *IEEE Transactions on Multimedia*, vol. 15, pp. 1060-1069, 2013.

[5] F. Cong, A. H. Phan, Q. Zhao, A. K. Nandi, V. Alluri, P. Toiviainen, *et al.*, "Analysis of Ongoing EEG Elicited by Natural Music Stimuli Using Nonnegative Tensor Factorization," *Proceeding of The 2012 European Signal Processing Conference (EUSIPCO-2012), Bucharest, Romania, August 27-31, 2012*, vol. Bucharest, Romania, pp. 494-498, 2012.

[6] D. S. Tan and A. Nijholt, *Brain-Computer Interfaces: Applying our Minds to Human-Computer Interaction*. London: Springer, 2010.

[7] H. Hotelling, "Relations between two sets of variables," *Biometrika*, vol. 28, pp. 321-377, 1936.

[8] H. Wold, "Soft modelling, the basic design and some extensions," in *Systems under Indirect Observation (Vols I and II)*, K. G. Jöreskog and H. Wold, Eds., ed Amsterdam: North-Holland, 1982, pp. 181-196.

[9] R. Bro, "Multiway calibration. Multilinear PLS," *J. Chemometr.*, vol. 10, pp. 47-61, 1996.

[10] E. Martinez-Montes, P. A. Valdes-Sosa, F. Miwakeichi, R. I. Goldman, and M. S. Cohen, "Concurrent EEG/fMRI analysis by multiway Partial Least Squares," *NeuroImage*, vol. 22, pp. 1023-1034, 2004.

[11] E. Acar, C. A. Bingol, H. Bingol, R. Bro, and B. Yener, "Seizure recognition on epilepsy feature tensor," *Conference proceedings : ...Annual International Conference of the IEEE Engineering in Medicine and Biology Society.IEEE Engineering in Medicine and Biology Society.Conference*, vol. 2007, pp. 4273-4276, 2007.

[12] J. Chiang, Z. Wang, and M. McKeown, "A multiblock PLS model of cortico-cortical and corticomuscular interactions in Parkinson's disease," *NeuroImage*, vol. 65, pp. 1498-1509, 2012.

[13] A. Krishnan, L. J. Williams, A. R. McIntosh, and H. Abdi, "Partial Least Squares (PLS) methods for neuroimaging: a tutorial and review," *NeuroImage*, vol. 56, pp. 455-475, 2011.

[14] A. Delorme and S. Makeig, "EEGLAB: an open source toolbox for analysis of single-trial EEG dynamics including independent component analysis," *Journal of neuroscience methods*, vol. 134, pp. 9-21, 2004.

[15] V. Alluri, P. Toiviainen, I. P. Jaaskelainen, E. Glerean, M. Sams, and E. Brattico, "Large-scale brain networks emerge from dynamic processing of musical timbre, key and rhythm," *NeuroImage*, vol. 59, pp. 3677-3689, 2012.

[16] C. A. Andersson and R. Bro, "The N-way Toolbox for MATLAB," *Chemometrics and Intelligent Laboratory Systems*, vol. 52, pp. 1-4, 2000.

[17] F. L. Hitchcock, "The Expression of a Tensor or a Polyadic as a Sum of Products," *Journal of Mathematics and Physics*, vol. 6, pp. 164-189, 1927.

[18] Q. Zhao, C. F. Caiafa, D. P. Mandic, Z. C. Chao, Y. Nagasaka, N. Fujii, *et al.*, "Higher-Order Partial Least Squares (HOPLS): A Generalized Multi-Linear Regression Method," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, pp. 1660-1673, 2013.

[19] A. Delorme, J. Palmer, J. Onton, R. Oostenveld, and S. Makeig, "Independent EEG sources are dipolar," *PloS one*, vol. 7, p. e30135, 2012.

[20] F. Cong, A. H. Phan, Q. Zhao, T. Huttunen-Scott, J. Kaartinen, T. Ristaniemi, *et al.*, "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, pp. 1-19, 2012.