Linked PARAFAC/CP Tensor Decomposition and Its Fast Implementation for Multi-block Tensor Analysis

Tatsuya Yokota[1,2], Andrzej Cichocki[2], and Yukihiko Yamashita[1]

[1] Tokyo Institute of Technology, Tokyo, Japan
[2] RIKEN Brain Science Institute, Saitama, Japan
yokota@yy.ide.titech.ac.jp

Abstract. In this paper we propose a new flexible group tensor analysis model called the linked CP tensor decomposition (LCPTD). The LCPTD method can decompose given multiple tensors into common factor matrices, individual factor matrices, and core tensors, simultaneously. We applied the Hierarchical Alternating Least Squares (HALS) algorithm to the LCPTD model; besides we impose additional constraints to obtain sparse and nonnegative factors. Furthermore, we conducted some experiments of this model to demonstrate its advantages over existing models.

Keywords: Tensor decompositions of multi-block data, PARAFAC/CP model, Group Analysis, Hierarchical Alternating Least Squares (HALS).

1 Introduction

The group (multi-block) tensor decomposition is a very important technique in neuroscience, image analysis, and some multi-modal data processing [3,6,11,7]. The group analysis seeks to identify some factors that are common in two or more blocks in a group [3]. The simultaneous tensor decomposition (STD) is known as one of the methods to extract common factor matrices from a group of subjects. The STD model can be applied into tensor based principal component analysis (PCA) and feature extraction for EEG classification [12].

In this paper, we consider a more flexible decomposition model called the linked tensor decomposition (LTD). The LTD method extracts not only their common factor matrices but also their individual (statistically independent) factor matrices at the same time. The LTD model can be characterized as a generalized model of the STD. In fact, it is an intermediate model between the STD model and the individual tensor decomposition model (i.e. standard tensor decomposition of individual blocks).

In order to implement the LTD model, we applied the Hierarchical Alternating Least Squares (HALS) algorithm with the CP (Canonical Polyadic) constraint and two options of sparsity and non-negativity constraints. We call this method the "Linked CP Tensor Decomposition" (LCPTD). Although the CP model is generally unique we will impose some constraints to obtain more meaningful components.

T. Huang et al. (Eds.): ICONIP 2012, Part III, LNCS 7665, pp. 84–91, 2012.

The rest of this paper is organized as follows. In Section 2, the existing models of tensor analysis are briefly explored. In Section 3, we introduce a novel linked tensor decomposition and its algorithm. In Section 4, we demonstrate experiments using our new method and present the results of these experiments. Finally, we give our conclusions in Section 5.

2 Tensor Decompositions

2.1 Single Tensor Decomposition Based on CP Model

The Canonical Polyadic (CP) model which is also called PARAFAC [8] or CANDECOMP [2] has been well used in positron emission tomography (PET), spectroscopy, chemometrics and environmental science [6,1]. The CP model can be expressed as

$$\underline{\boldsymbol{Z}} \approx \widehat{\underline{\boldsymbol{Z}}} := [\![\underline{\boldsymbol{G}}; \boldsymbol{U}^{(1)}, \boldsymbol{U}^{(2)}, \ \ldots, \boldsymbol{U}^{(N)}]\!] = \sum_{j=1}^{J} g_j \boldsymbol{u}_j^{(1)} \circ \boldsymbol{u}_j^{(2)} \circ \cdots \circ \boldsymbol{u}_j^{(N)}, \qquad (1)$$

where $\underline{\boldsymbol{Z}} \in \mathbb{R}^{I_1 \times \cdots \times I_N}$ is an N-order tensor and model, $\boldsymbol{U}^{(n)} = [\boldsymbol{u}_1^{(n)}, \ldots, \boldsymbol{u}_J^{(n)}] \in \mathbb{R}^{I_n \times J}$ is an n-mode factor matrix with components $\boldsymbol{u}_j^{(n)}$, $\underline{\boldsymbol{G}} = \underline{\boldsymbol{\Lambda}} \in \mathbb{R}^{J \times \cdots \times J}$ is a diagonal core tensor with entries g_j on the main diagonal. The goal of CP decomposition is to estimate factor matrices by minimizing a Frobenius norm of residual tensor $\underline{\boldsymbol{E}} := \underline{\boldsymbol{Z}} - \widehat{\underline{\boldsymbol{Z}}}$. The criterion is given by:

$$\text{minimize} \quad ||\underline{\boldsymbol{E}}||_F^2 = \left|\left| \underline{\boldsymbol{Z}} - \Sigma_{j=1}^{J} g_j \boldsymbol{u}_j^{(1)} \circ \cdots \circ \boldsymbol{u}_j^{(N)} \right|\right|_F^2, \ \text{s.t.}\ ||\boldsymbol{u}_j^{(n)}|| = 1, \qquad (2)$$

for $n = 1, \ldots, N$ and $j = 1, \ldots, J$.

When we treat a real-world data, sparsity and non-negativity of factor matrices may play a key role to extract meaningful components. There are many methods for feature extraction and blind source separation using sparsity and non-negativity constraints such as sparse principal component analysis [9] and nonnegative matrix factorization [10,6]. A sparsity constraints are given by $||\boldsymbol{u}_j^{(n)}||_1 < v$, where $||\cdot||_1$ is l_1-norm, and v is a threshold parameter. When it is added into (2), then the criterion provides sparse factor matrices. Next the non-negativity constraint is given by $u_{ji}^{(n)} \geq 0, \ g_j \geq 0 \ \forall j, i, n$. In the same way, when the constraints is added into (2), then the criterion realizes nonnegative tensor factorization (NTF).

2.2 Simultaneous Tensor Decomposition

In this section, we introduce the simultaneous CP tensor decomposition (SCPTD). This is very important to explain the proposed method; since the STD is closely related to our new LCPTD. We discuss multiple tensor decompositions from here; besides we assume that there are S tensors of the same dimensions and we obtain S decompositions. We can consider S as the number of blocks (e.g. each block data represents one subject).

One of the objective of group tensor analysis is to decompose individual tensors one by one based on the CP model which is principally unique. We describe this model as the individual CP tensor decomposition (ICPTD). However, in such case the factor matrices are not directly linked.

On the other hand, it is meaningful to extract some common factors for each block which link block by some common factors. The formulation of the SCPTD is given by $\underline{\boldsymbol{Z}}^{(s)} \approx \widehat{\underline{\boldsymbol{Z}}}^{(s)} := [\![\underline{\boldsymbol{G}}^{(s)}; \boldsymbol{U}^{(1)}, \ldots, \boldsymbol{U}^{(N)}]\!] = \sum_{j=1}^{J} g_j^{(s)} \boldsymbol{u}_j^{(1)} \circ \cdots \circ \boldsymbol{u}_j^{(N)}$. The key-point here is that the basis components ($\boldsymbol{u}_j^{(n)}$ of $\boldsymbol{U}^{(n)}$) are the same for all blocks. Only the core tensors $\underline{\boldsymbol{G}}^{(s)}$ are different for individual blocks which represent features [12].

3 Linked CP Tensor Decomposition

In this section, we propose a new model of simultaneous decomposition called the "Linked CP tensor decomposition"(LCPTD) as

$$\underline{\boldsymbol{Z}}^{(s)} \approx \widehat{\underline{\boldsymbol{Z}}}^{(s)} = [\![\underline{\boldsymbol{G}}^{(s)}; \boldsymbol{U}^{(1,s)}, \ldots, \boldsymbol{U}^{(N,s)}]\!] = \sum_{j=1}^{J} g_j^{(s)} \boldsymbol{u}_j^{(1,s)} \circ \ldots \circ \boldsymbol{u}_j^{(N,s)}, \tag{3}$$

where each factor matrix $\boldsymbol{U}^{(n,s)} = [\boldsymbol{U}_C^{(n)}, \boldsymbol{U}_I^{(n,s)}] \in \mathbb{R}^{I_n \times J}$ is composed of two set of bases: $\boldsymbol{U}_C^{(n)} \in \mathbb{R}^{I_n \times L_n}$ (with $0 \leq L_n \leq J$), which is a common factor matrix for all blocks and corresponds to the same or maximally correlated components and $\boldsymbol{U}_I^{(n,s)} \in \mathbb{R}^{I_n \times J - L_n}$, which corresponds to different individual characteristics.

The LCPTD can be considered as a generalized model of simultaneous decomposition. When we put $L_n = J$, its decomposition is equivalent to the simultaneous common factor decomposition [12]. On the other hand, when $L_n = 0$, its decomposition of each subject is equivalent to the standard tensor decomposition. Then the LTD is an intermediate decomposition between simultaneous and normal tensor decomposition.

3.1 LCPTD-HALS Algorithm

In this section, we introduce a new HALS algorithm for LCPTD. Optimization criterion for LCPTD is given by

$$\text{minimize} \quad \sum_{s=1}^{S} \left|\left| \underline{\boldsymbol{Z}}^{(s)} - \sum_{j=1}^{J} g_j^{(s)} \boldsymbol{u}_j^{(1,s)} \circ \ldots \circ \boldsymbol{u}_j^{(N,s)} \right|\right|_F^2, \tag{4}$$

$$\text{subject to} \quad \boldsymbol{u}_j^{(n,1)} = \cdots = \boldsymbol{u}_j^{(n,S)} \text{ for } j \leq L_n, \ ||\boldsymbol{u}_j^{(n,s)}|| = 1, \tag{5}$$

for all n, s, and j. Furthermore, we add $||\boldsymbol{u}_j^{(n,s)}||_1 < v$ or $u_{ji}^{(n,s)} \geq 0$, $g_j^{(s)} \geq 0 \ \ \forall i, j, n, s$ into (5) if we want to get sparse or nonnegative components.

The Hierarchical ALS (HALS) algorithm was first proposed for the Nonnegative Matrix Factorization and Nonnegative Tensor Factorization (NTF) in

[5]. The HALS algorithm were applied to the CP model and it achieved good performances in [4]. In this algorithm, we consider J local-problems and solve them sequentially and iteratively instead of solving (4) and (5), directly. Let $\underline{\boldsymbol{Y}}_j^{(s)} := \underline{\boldsymbol{Z}}^{(s)} - \sum_{i \neq j} g_i^{(s)} \boldsymbol{u}_i^{(1,s)} \circ \ldots \circ \boldsymbol{u}_i^{(N,s)}$, the j-th local problem is given by

$$\text{minimize} \quad \sum_{s=1}^{S} ||\underline{\boldsymbol{Y}}_j^{(s)} - g_j^{(s)} \boldsymbol{u}_j^{(1,s)} \circ \ldots \circ \boldsymbol{u}_j^{(N,s)}||_F^2, \tag{6}$$

$$\text{subject to} \quad \boldsymbol{u}_j^{(n,1)} = \cdots = \boldsymbol{u}_j^{(n,S)} \text{ if } j \leq L_n, \; ||\boldsymbol{u}_j^{(n,s)}|| = 1, \tag{7}$$

for all n and s. The LTD-HALS algorithm can be summarized as Algorithm 1; note it does not require matrix inversion and is solved by only simple calculation.

Algorithm 1. LTD-HALS algorithm

Input: $\{\underline{\boldsymbol{Z}}^{(s)}\}_{s=1}^{S}$, J, and $\{L_n\}_{n=1}^{N}$
Initialize: $\{\boldsymbol{g}^{(s)}, \; \{\boldsymbol{U}^{(n,s)}\}_{n=1}^{N}\}_{s=1}^{S}$.
$\underline{\boldsymbol{E}}^{(s)} = \underline{\boldsymbol{Z}}^{(s)} - \Sigma_{j=1}^{J} g_j^{(s)} \boldsymbol{u}_j^{(1,s)} \circ \cdots \circ \boldsymbol{u}_j^{(N,s)}$ for all s;
repeat
  for $j = 1, \ldots, J$ **do**
    $\underline{\boldsymbol{Y}}_j^{(s)} = \underline{\boldsymbol{E}}^{(s)} + g_j^{(s)} \boldsymbol{u}_j^{(1,s)} \circ \cdots \circ \boldsymbol{u}_j^{(N,s)}$ for all s;
    for $n = 1, \ldots, N$ **do**
      Updating $\boldsymbol{u}_j^{(n,s)}$:

$$\boldsymbol{u}_j^{(n,s)} \leftarrow g_j^{(s)} \underline{\boldsymbol{Y}}_j^{(s)} \times_1 \boldsymbol{u}_j^{(1,s)} \cdots \times_{n-1} \boldsymbol{u}_j^{(n-1,s)} \times_{n+1} \boldsymbol{u}_j^{(n+1,s)} \cdots \times_N \boldsymbol{u}_j^{(N,s)} \text{ for all } s; \tag{8}$$

      if $j \leq L_n$, $\boldsymbol{t} \leftarrow \sum_{s=1}^{S} \boldsymbol{u}_j^{(n,s)}$; $\boldsymbol{u}_j^{(n,s)} \leftarrow \boldsymbol{t}$ for all s; **end if**
      Normalize $\boldsymbol{u}_j^{(n,s)} \leftarrow \boldsymbol{u}_j^{(n,s)} / ||\boldsymbol{u}_j^{(n,s)}||$ for all s;
    end for
    Update $g_j^{(s)}$:

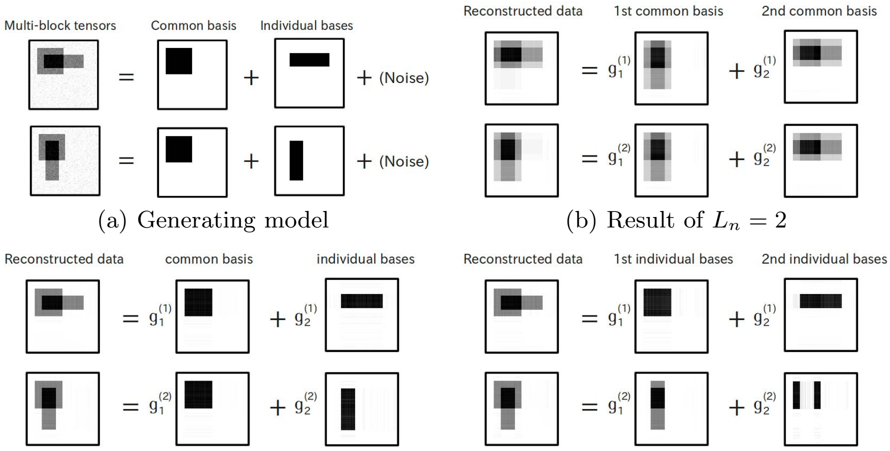
$$g_j^{(s)} \leftarrow \underline{\boldsymbol{Y}}_j^{(s)} \times_1 \boldsymbol{u}_j^{(1,s)} \cdots \times_N \boldsymbol{u}_j^{(N,s)} \text{ for all } s; \tag{9}$$

    $\underline{\boldsymbol{E}}^{(s)} = \underline{\boldsymbol{Y}}_j^{(s)} - g_j^{(s)} \boldsymbol{u}_j^{(1,s)} \circ \cdots \circ \boldsymbol{u}_j^{(N,s)}$ for all s;
  end for
until $\sum_{s=1}^{S} ||\underline{\boldsymbol{E}}^{(s)}||_F^2$ converge
Output: $\{\boldsymbol{g}^{(s)}, \; \{\boldsymbol{U}^{(n,s)}\}_{n=1}^{N}\}_{s=1}^{S}$

If we want to obtain sparse components, we implement the following updates after (8):

$$\boldsymbol{u}_j^{(n,s)} \leftarrow \text{sign}(\boldsymbol{u}_j^{(n,s)}) \circledast [\text{abs}(\boldsymbol{u}_j^{(n,s)}) - \xi_n \mathbf{1}]_+ \text{ for all } s; \tag{10}$$

where ξ_n is a positive parameter deciding on their sparsity.



(a) Generating model (b) Result of $L_n = 2$

(c) Result of $L_n = 1$ (d) Result of $L_n = 0$

Fig. 1. Linked Multi-block Tensor Factorization

If we want to obtain nonnegative components, we implement the following updates after (8) and (9):

$$\boldsymbol{u}_j^{(n,s)} \leftarrow [\boldsymbol{u}_j^{(n,s)}]_+ \text{ for all } s, \tag{11}$$

$$g_j^{(s)} \leftarrow [g_j^{(s)}]_+ \text{ for all } s. \tag{12}$$

4 Experiments

4.1 Toy Problem for Linked Multi-block Tensor Factorization

In this part, we applied the LCPTD to a toy problem (benchmark) for linked multi-block tensor factorization. We generated two block data tensors consisting of a one common basis factor and two individual basis factors with noise (see Fig. 1(a)). And we decompose them by our LCPTD model with nonnegative constraints for various number of common bases $L_n \in \{2, 1, 0\}$ for $n = 1, 2$. Fig. 1(b,c,d) depict the results of this experiment. It is obvious that the result of $L_n = 2$ couldn't represent the original data tensors since the degree of freedom of model is not sufficient. On the other hand, the result of $L_n = 0$ could represent the original data tensors, but each basis is not matched, completely. The result of $L_n = 1$ could represent not only the original data tensors, but also each basis; besides, the additive noise were reduced.

We can see that the LCPTD model can be very useful assuming that some components are common in generating model. The blind source separation can separate into two original sources from two observed signals. It is very interesting that the LCPTD can achieve separation of three bases (i.e., a common and two individual

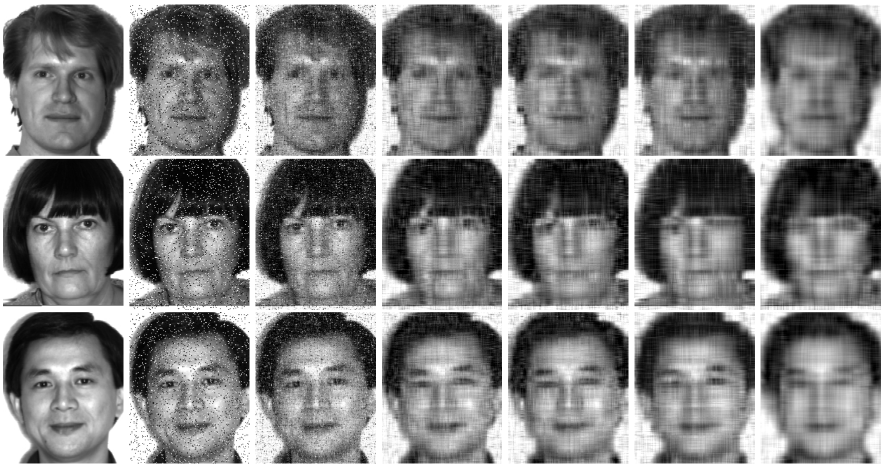
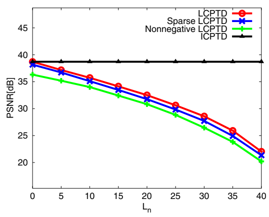
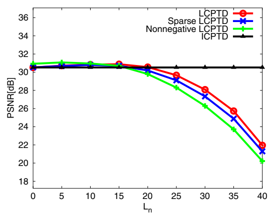


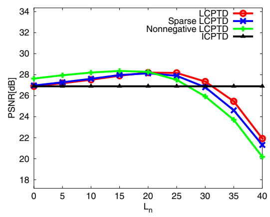
Fig. 2. Face images corrupted by additive noise and the reconstructed images (PSNR= 15 dB, $J = 40$): 1st column: original images, 2nd column: noisy images, 3rd column: ICPTD model, 4th column: LCPTD ($L_n = 35$), 5th column: LCPTD with sparse constraint ($L_n = 35$), 6th column: LCPTD with nonnegative constraint ($L_n = 35$), 7th column: SCPTD model.



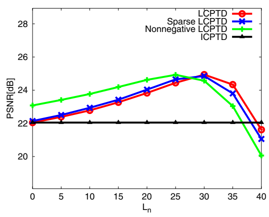
(a) Noise free



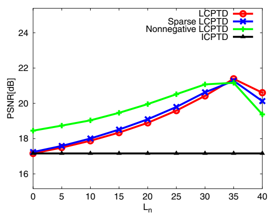
(b) PSNR=30 dB



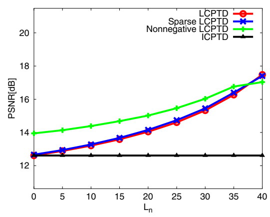
(c) PSNR=25 dB



(d) PSNR=20 dB



(e) PSNR=15 dB



(f) PSNR=10 dB

Fig. 3. PSNRs for various noisy data sets

bases) from only two observed tensors. We should note that the selection of L_n could be very important deciding factor to obtain proper decomposition for the LCPTD method from this experiment.

4.2 Images (Faces) Reconstruction and Denoising

In this part, the LCPTD was applied to face reconstruction problems and the performances were compared with other models. The Yale face database consists of 165 gray-scale images of 15 individuals. There are 11 images per subject with different facial expressions or configurations. In this experiments, we used 15 full-face images; we took a one image from each subject. Size of images are 215×171 pixels, then we considered that $I_1 = 215$, $I_2 = 171$, and $S = 15$. Furthermore, we prepared salt-and-pepper noised data sets: SNR $\in \{5,\ 10,\ 20\}$ [dB].

We applied our new LCPTD model, a sparse LCPTD, and a nonnegative LCPTD with various numbers of common components for the noise free and noised data sets; the number of bases was fixed as $J = 40$, and number of common components was changed in $L_n \in \{0,\ 5,\ 10,\ \ldots,\ 40\}$. We computed the PSNR between the original faces and the reconstructed faces.

Fig. 2 depicts the results of face reconstruction. We can see that the ICPTD model couldn't reduce the noise well, and the SCPTD model was robust with respect to noise but reconstructed faces were too fuzzy (distorted). However, the LCPTD based methods gave the appropriate and intermediate decompositions for both models.

Fig. 3 depicts the graphs of PSNR for various number of common components. We can see that if the noise level becomes larger, the maximum points of PSNR move to right. In noise free data set Fig. 3(a), the maximum PSNR was obtained at $L_n = 0$ for all methods; so the ICPTD model is the best for them in this case. On the other hand, the maximum PSNRs were obtained by the LCPTD based methods in Fig. 3(b,c,d,e). It is also interesting that the nonnegative LCPTD kept high PSNR for smaller number of common components in comparison with the other methods in high noise level (see Fig. 3(d,e,f)). It can be considered that the nonnegative constraint is useful for this problem.

In general, because real data includes often some noise factors, the proposed method could be very useful and practical for the real tensor data analysis. The higher noise level requires large number of common components, but multitude of common components often only hampers the fitting. However, we have to select the best parameter of L_n and the open problem is how to select it. We may be able to select L_n depending on PSNR if it is known as prior information.

5 Conclusion

We have presented a method of linked CP tensor decomposition (LCPTD) including sparse and nonnegative factorization by using the HALS algorithm. LCPTD can be considered as a generalized model of simultaneous CP tensor decomposition with common factors, and it provides some improvement over existing methods by selecting optimal parameters of L_n and ξ_n. The parameter selection can be considered as a key issue of flexible model. The Bayesian method or cross validation method may be able to determine such parameters. Its detail and application are reserved for our future works.

References

1. Bro, R.: Multi-way analysis in the food industry - models, algorithms, and applications. Tech. rep., MRI, EPG and EMA, Proc. ICSLP 2000 (1998)
2. Carroll, J., Chang, J.J.: Analysis of individual differences in multidimensional scaling via an n-way generalization of eckart-young decomposition. Psychometrika 35, 283–319 (1970)
3. Cichocki, A.: Tensor decompositions: New concepts for brain data analysis? Journal of Control, Measurement, and System Integration (SICE) 7, 507–517 (2011)
4. Cichocki, A., Phan, A.: Fast local algorithms for large scale nonnegative matrix and tensor factorizations. IEICE Trans. Fundamentals of Electronics, Communications and Computer Sciences E92-A (3), 708–721 (2009)
5. Cichocki, A., Zdunek, R., Amari, S.-I.: Hierarchical ALS Algorithms for Nonnegative Matrix and 3D Tensor Factorization. In: Davies, M.E., James, C.J., Abdallah, S.A., Plumbley, M.D. (eds.) ICA 2007. LNCS, vol. 4666, pp. 169–176. Springer, Heidelberg (2007)
6. Cichocki, A., Zdunek, R., Phan, A.H., Amari, S.: Nonnegative Matrix and Tensor Factorizations: Applications to Exploratory Multi-way Data Analysis and Blind Source Separation. Wiley Publishing (2009)
7. Guo, Y., Pagnoni, G.: A unified framework for group independent component analysis for multi-subject fMRI data. NeuroImage 42(3), 1078–1093 (2008)
8. Harshman, R.: Foundations of the parafac procedure: Model and conditions for an 'explanatory' multi-mode factor analysis. UCLA Working Papers in phonetics 16, 1–84 (1970)
9. Jolliffe, I.T., Trendafilov, N.T., Uddin, M.: A modified principal component technique based on the LASSO. Journal of Computational and Graphical Statistics 12(3), 531–547 (2003)
10. Lee, D.D., Seung, H.S.: Learning the parts of objects by non-negative matrix factorization. Nature 401(6755), 788–791 (1999)
11. Lee, H., Choi, S.: Group nonnegative matrix factorization for EEG classification. Journal of Machine Learning Research - Proceedings Track 5, 320–327 (2009)
12. Phan, A., Cichocki, A.: Tensor decompositions for feature extraction and classification of high dimensional datasets. IEICE, NOLTA 1(1), 37–68 (2010)