International Journal of Neural Systems
Special Issue on Issue's Topic
© World Scientific Publishing Company

NONNEGATIVE TENSOR FACTORIZATION FOR CONTINUOUS EEG CLASSIFICATION

Hyekyoung Lee and Yong-Deok Kim
Department of Computer Science,
Pohang University of Science and Technology,
San 31 Hyoja-dong, Nam-gu,
Pohang 790-784, Republic of Korea
{leehk,karma13}@postech.ac.kr

Andrzej Cichocki
Laboratory for Advanced Brain Signal Processing,
Brain Science Institute, RIKEN
2-1 Hirosawa, Wako-shi, Saitama 351-0198, Japan
cia@brain.riken.jp

Seungjin Choi
Department of Computer Science,
Pohang University of Science and Technology,
San 31 Hyoja-dong, Nam-gu,
Pohang 790-784, Republic of Kerea
seungjin@postech.ac.kr

Received (to be inserted
Revised by Publisher)

In this paper we present a method for continuous EEG classification, where we employ nonnegative tensor factorization (NTF) to determine discriminative spectral features and use the Viterbi algorithm to continuously classify multiple mental tasks. This is an extension of our previous work on the use of nonnegative matrix factorization (NMF) for EEG classification. Numerical experiments with two data sets in BCI competition, confirm the useful behavior of the method for continuous EEG classification.

1. Introduction

Brain computer interface (BCI) is a system that is designed to translate a subject's intention or mind into a control signal for a device such as a computer, a wheelchair, or a neuroprosthesis [35]. BCI provides a new communication channel between human brain and computer and adds a new dimension to human computer interface (HCI). It was motivated by the hope of creating new communication channels for disabled persons, but recently draws attention in multimedia communication as well [13].

The most popular sensory signal used for BCI is electroencephalogram (EEG) which is the multivariate time series data where electrical potentials induced by brain activities are recorded in a scalp. Inferring the human intention using EEG is similar to inferring what is going on in a game from the hubbub outside a stadium. If a lot of people in the stadium shout simultaneously when a team scores a goal or loses a goal, we can guess the situation only just hearing hubbub outside the stadium, without being in the stadium. Stimuli make neurons to cheer in chorus, which makes EEG to have certain characteristics.

Exemplary spectral characteristics of EEG, in motor imagery tasks which are considered in this paper, are μ rhythm (8-12 Hz) [35] and β rhythm (18-25 Hz) which decrease during movement or in preparation for movement (event-related desynchronization, ERD) and increase after movement and in relaxation (event-related synchronization, ERS). However those phenomena could happen in different frequency bands, depending on subjects. For instance, they might occur in 16-20 Hz, not in 8-12 Hz [20].

EEG classification using ERD and ERS during motor imagery, has been extensively studied. Along this line, various methods have been developed with promising results [26,25,31,32]. Besides motor imagery task, cognitive tasks has recently been studied in BCI community, including word generation, recall, expectancy, subtraction, and so on. Spectral properties related to cognition and perception are known to involve in the gamma band (30-100 Hz) at posterior and central scalp and to involve in the theta band (3-7 Hz) at bilateral and midline frontal scalp if they are also related with memory [14]. However, such characteristics are not strongly distinguishable, compared to ERD and ERS. Moreover, their variations are very large depending on subjects. Therefore, methods for determining meaningful discriminative features become more important.

Linear data model is a widely-used method for multivariate data analysis, including principal component analysis (PCA), linear discriminant analysis (LDA), and independent component analysis (ICA). Linear data model assumes that the observed multivariate data is represented by a weighted linear sum of basis vectors. Depending on criteria, different meaningful basis vectors are learned from data and appropriate features (corresponding to encoding variables) are determined by simply projecting data onto basis vectors. It was also shown in EEG analysis and classification [23,24,30].

Nonnegative matrix factorization (NMF) is another interesting linear data model, which is more appropriate for handling nonnegative data [21,22]. In contrast to other linear data models, NMF allows only non-subtractive combinations of nonnegative basis vectors, providing a parts-based representation. The time-frequency representation of EEG data computed by short-time Fourier transform or wavelet transform, [3,1,2,16] can be cast into a nonnegative data matrix. Recently, NMF was shown to be useful in determining discriminative basis vectors which well reflect meaningful spectral characteristics without the cross-validation in motor imagery EEG task [25].

Multiway analysis extends aforementioned linear methods, working with multiway data array which is referred to as *tensor*.[*] PARAFAC [19] and multiway SVD (higher-order SVD) [10,11] are exemplary methods which are extensions of factor analysis and SVD. Recently PARAFAC model was exploited in EEG data analysis [28]. Nonnegative tensor factorization (NTF) incorporates nonnegativity constraints into PARAFAC model, extending NMF in the framework of tensor algebra [34,33]. Various information divergences were employed in NTF [9,8].

In this paper, we revisit PARAFAC model and present a NTF algorithm in a compact form, in the framework of tensor algebra. Then we cast the time-frequency representation of multichannel EEG data into a N-way tensor and apply NTF to determine discriminative spectral features. With these features, we use the Viterbi algorithm for continuous EEG classification with no trial structure. The rest of

[*] A vector is a 1-way tensor and a matrix is a 2-way tensor.

this paper is organized as follows. Sec. 2 provides a background for NMF and NTF as well as for fundamental knowledge related to tensor algebra. The proposed NTF-based method is illustrated in detail in Sec. 3. Numerical experiments and results with two data sets in BCI competition, are presented in Sec. 4. Finally conclusions are drawn in Sec. 5.

2. Nonnegative Tensor Factorization

We present a brief overview of NMF and NTF, which is necessary to understand the proposed method. We begin with NMF and updating rules in the case of I-divergence. Then we explain fundamental operations used in tensor analysis and present a NTF algorithm in a compact form.

2.1. *NMF*

Suppose that l observed m-dimensional data points, $\{\boldsymbol{x}_t\}$, $t = 1, \ldots, l$ are available. Denote the data matrix by $\boldsymbol{X} = [\boldsymbol{x}_1 \cdots \boldsymbol{x}_l] = [X_{ij}] \in \mathbb{R}^{m \times l}$. Linear data model seeks a factorization that is of the form

$$\boldsymbol{X} = \widehat{\boldsymbol{X}} \approx \boldsymbol{A}\boldsymbol{S}, \tag{1}$$

where $\boldsymbol{A} \in \mathbb{R}^{m \times n}$ contains basis vectors in its columns and $\boldsymbol{S} \in \mathbb{R}^{n \times l}$ is the associated encoding variable matrix.

NMF is one of widely-used multivariate analysis methods for nonnegative data, which has many potential applications in pattern recognition and machine learning [29,21,22]. NMF seeks a decomposition (1) of the nonnegative data matrix $\boldsymbol{X}$ with matrices $\boldsymbol{A}$ and $\boldsymbol{S}$ restricted to have only nonnegative elements. Various error measures for the factorization with nonnegativity constraints, can be considered. For example, see [22,6,7,12] for different NMF algorithms with various error functions. In this paper we only consider I-divergence which is given by

$$D[\boldsymbol{X} \parallel \boldsymbol{A}\boldsymbol{S}] = \sum_{i,j} \left[X_{ij} \log \frac{X_{ij}}{[\boldsymbol{A}\boldsymbol{S}]_{ij}} - X_{ij} + [\boldsymbol{A}\boldsymbol{S}]_{ij} \right].$$

NMF involves the following optimization problem:

$$\begin{aligned} &\arg\min_{\boldsymbol{A},\boldsymbol{S}} D[\boldsymbol{X} \parallel \boldsymbol{A}\boldsymbol{S}] && (2) \\ &\text{s.t. } A_{ij}, S_{ij} \geq 0 \ \forall i, j. && (3) \end{aligned}$$

The multiplicative updating rules for iteratively determining a local minimum of (2), are given by

$$S_{ij} \leftarrow S_{ij} \left[\frac{\Sigma_k [A_{ki}(X_{kj}/[\boldsymbol{A}\boldsymbol{S}]_{kl})]}{\Sigma_l A_{li}} \right], \tag{4}$$

$$A_{ij} \leftarrow A_{ij} \left[\frac{\Sigma_k [S_{jk}(X_{ik}/[\boldsymbol{A}\boldsymbol{S}]_{ik})]}{\Sigma_l S_{jl}} \right]. \tag{5}$$

2.2. *NTF*

NTF is a recent multiway extension of nonnegative matrix factorization (NMF), where nonnegativity constraints are incorporated into the PARAFAC model [34,33,17]. Image data, video data, or spectral data of time series naturally fit in 3-way or multiway structure. A multiway data array is referred to as a tensor. A vector is a 1-way tensor, a matrix is a 2-way tensor, a cube is a 3-way tensor, and so on. Spectral EEG data can represented by a tensor whose coordinates correspond to channel, class, trial, and so on, whereas a data matrix is limited to only 2 coordinates in the case of NMF.

2.2.1. *Multiway analysis*

Notations used for tensor analysis are summarized in Table 1.

Table 1. Notations.

Notation	Description
$\boldsymbol{\mathcal{X}}$	N-way tensor
$\boldsymbol{X}$	matrix
$\boldsymbol{X}_{(n)}$	mode-n matricization of tensor $\boldsymbol{\mathcal{X}}$
$\boldsymbol{A}^{(n)}$	mode-n component matrix in Eq. (7)
$\odot$	Khatri-Rao product
$\circ$	outer product
$\circledast$	Hadamard product

The N-way tensor $\boldsymbol{\mathcal{X}} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$ has N indices $(i_1, i_2, \ldots, i_N)$ and its elements are denoted by $x_{i_1,i_2,\ldots,i_N}$ where $1 \leq i_n \leq I_n$. Mode-n vectors of an N-way tensor $\boldsymbol{\mathcal{X}}$ are I_n-dimensional vectors obtained from $\boldsymbol{\mathcal{X}}$ by varying index i_n while keeping the other indices fixed. In matrix, column vectors are referred to as mode-1 vectors and row vectors correspond to mode-2 vectors.

The mode-n vectors are column vectors of the matrix $\boldsymbol{X}_{(n)} \in \mathbb{R}^{I_n \times I_{n+1} I_{n+2} \cdots I_N I_1 I_2 \cdots I_{n-1}}$ which is the mode-n matricization (matrix unfolding) of the tensor $\boldsymbol{\mathcal{X}}$ (Fig. 1).

The scalar product of two tensors $\boldsymbol{\mathcal{X}}, \boldsymbol{\mathcal{Y}}$ is defined as

$$\langle \boldsymbol{\mathcal{X}}, \boldsymbol{\mathcal{Y}} \rangle = \sum_{i_1,i_2,\ldots,i_N} x_{i_1,i_2,\ldots,i_N} y_{i_1,i_2,\ldots,i_N},$$

where $\boldsymbol{\mathcal{X}}, \boldsymbol{\mathcal{Y}} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$. The Frobenius norm of a tensor $\boldsymbol{\mathcal{X}}$ is given by $\| \boldsymbol{\mathcal{X}} \| = \sqrt{\langle \boldsymbol{\mathcal{X}}, \boldsymbol{\mathcal{X}} \rangle}$.

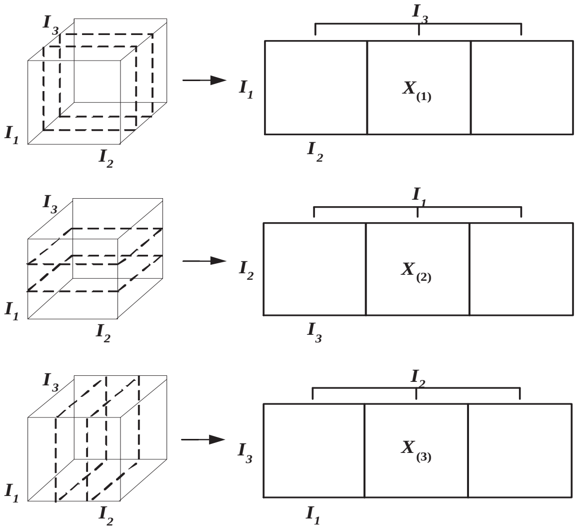


Fig. 1. Unfolding a 3-way tensor $\boldsymbol{\mathcal{X}} \in \mathbb{R}^{I_1 \times I_2 \times I_3}$ leads to $\boldsymbol{X}_{(1)} \in \mathbb{R}^{I_1 \times I_2 I_3}$, $\boldsymbol{X}_{(2)} \in \mathbb{R}^{I_2 \times I_3 I_1}$ and $\boldsymbol{X}_{(3)} \in \mathbb{R}^{I_3 \times I_1 I_2}$.

2.2.2. *PARAFAC model*

An N-way tensor $\boldsymbol{\mathcal{X}} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$ had rank-1 when it equals to the outer product of N vectors:

$$\boldsymbol{\mathcal{X}} = \boldsymbol{a}^{(1)} \circ \boldsymbol{a}^{(2)} \circ \cdots \circ \boldsymbol{a}^{(N)},$$

where $\boldsymbol{a}^{(n)} \in \mathbb{R}^{I_n}$ for $n = 1, 2, \ldots, N$. In an element-wise form, it is written as

$$\mathcal{X}_{i_1,i_2,\ldots,i_N} = a^{(1)}_{i_1} a^{(2)}_{i_2} \cdots a^{(N)}_{i_N},$$

where $a^{(n)}_{i_n}$ denotes the i_nth element of the vector $\boldsymbol{a}^{(n)}$. The rank of an N-way tensor $\boldsymbol{\mathcal{X}}$, denoted $R = \text{rank}(\boldsymbol{\mathcal{X}})$, is the minimal number of rank-1 tensors that is required to yield $\boldsymbol{\mathcal{X}}$:

$$\boldsymbol{\mathcal{X}} \approx \widehat{\boldsymbol{\mathcal{X}}} = \sum_{r=1}^{R} \boldsymbol{A}^{(1)}_{:,r} \circ \boldsymbol{A}^{(2)}_{:,r} \circ \cdots \circ \boldsymbol{A}^{(N)}_{:,r}, \qquad (6)$$

where $\boldsymbol{A}^{(n)}_{:,r}$ represents the rth column vector of the component matrix $\boldsymbol{A}^{(n)} \in \mathbb{R}^{I_n \times R}$.

The PARAFAC model seeks the rank-R approximation of the tensor $\boldsymbol{\mathcal{X}}$ in (6). In an element-wise form, Eq. (6) is written as

$$\mathcal{X}_{i_1,i_2,\ldots,i_N} \approx \widehat{\mathcal{X}}_{i_1,i_2,\ldots,i_N} = \sum_{r=1}^{R} a^{(1)}_{i_1 r} a^{(2)}_{i_2 r} \cdots a^{(N)}_{i_N r}.$$

3-way PARAFAC model is shown in Fig. 2. The mode-n matricization of $\boldsymbol{\mathcal{X}}$ in the PARAFAC model, is expressed by Khatri-Rao products (column-wise Kronecker product) of component matrices:

$$\begin{aligned} \boldsymbol{X}_{(n)} &\approx \boldsymbol{A}^{(n)} \Big[\boldsymbol{A}^{(n-1)} \odot \cdots \odot \boldsymbol{A}^{(2)} \odot \boldsymbol{A}^{(1)} \\ &\qquad \odot \boldsymbol{A}^{(N)} \odot \cdots \odot \boldsymbol{A}^{(n+2)} \odot \boldsymbol{A}^{(n+1)} \Big]^{\top} \\ &= \boldsymbol{A}^{(n)} \boldsymbol{S}^{(n)}_{A}. \end{aligned} \qquad (7)$$

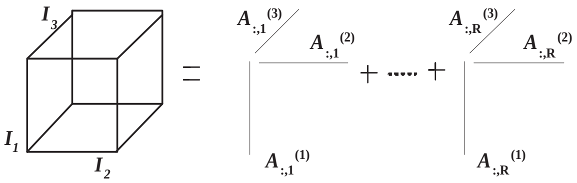


Fig. 2. The rank-R approximation of a 3-way tensor through the PARAFAC model.

2.2.3. *Updating rules*

NTF added nonnegtive constraints of component matrices in the factorization to the PARAFAC model. The objective function of NTF is similar to NMF

$$\begin{aligned} D[\boldsymbol{\mathcal{X}} \,\|\, \widehat{\boldsymbol{\mathcal{X}}}] &= \sum_{i_1,i_2,\ldots,i_N} \Bigg[\mathcal{X}_{i_1,i_2,\ldots,i_N} \log \frac{\mathcal{X}_{i_1,i_2,\ldots,i_N}}{\widehat{\mathcal{X}}_{i_1,i_2,\ldots,i_N}} \\ &\qquad - \mathcal{X}_{i_1,i_2,\ldots,i_N} + \widehat{\mathcal{X}}_{i_1,i_2,\ldots,i_N} \Bigg]. \end{aligned}$$

Multiplicative update rules for iteratively determining a nonnegative component matrices which minimize the objective function is quite similar to algorithms in NMF. Eq. (7) is of the same form as NMF model. Thus, updating rules for $\boldsymbol{A}^{(n)}$ follow NMF updating rules for $\boldsymbol{A}$.

$$\boldsymbol{A}^{(n)} \leftarrow \boldsymbol{A}^{(n)} \circledast \frac{\Big[\boldsymbol{X}_{(n)} / \Big(\boldsymbol{A}^{(n)} \boldsymbol{S}^{(n)}_{A} \Big) \Big] \boldsymbol{S}^{(n)\,T}_{A}}{\boldsymbol{1} \boldsymbol{z}^{T}}, \qquad (8)$$

where / is the element-size division, $\boldsymbol{1} \in \mathbb{R}^{I_n \times 1}$, $\boldsymbol{z} \in \mathbb{R}^{J_n \times 1}$ with $z_i = \sum_j \Big[\boldsymbol{S}^{(n)}_{A} \Big]_{ij}$. Updating rule of each component matrices can be easily derived as in NMF updating rules by matricizing the PARAFAC model into associated modes.

3. Proposed Method

The proposed methods for EEG classification consists of three steps: (1) preprocessing (by wavelet transform); (2) NTF-based feature extraction; (3) classification. Each of these steps is described in detail, following the brief description of two different data sets used in our numerical experiments.

3.1. *Data description*

For our empirical study, we used two data sets: one is the dataset III in BCI competition II, which was provided by the Laboratory of Brain-Computer Interfaces (BCI-Lab), Graz University of Technology [4,26], and the other is the dataset V in BCI competition III, which was provided by the IDIAP Research Institute [18].

3.1.1. *Graz dataset*

The Graz dataset involves left/right imagery hand movements and consists of 140 labelled trials for training and 140 unlabelled trials for test. Each trial has a duration of 9 seconds, where a visual cue (arrow) is presented pointing to the left or the right after 3-second preparation period and imagination task is carried out for 6 seconds. It contains EEG acquired from three different channels (with sampling frequency 128 Hz) C_3, C_z and C_4. In our study we use only two channels, C_3 and C_4, because ERD has contralateral dominance and C_z channel contains little information for discriminant analysis. Requirements for result comparison is to provide a continuous classification accuracy for each time point of trial during imagination session.

3.1.2. *IDIAP dataset*

The IDIAP dataset contains EEG data recorded from 3 normal subjects during 4 non-feedback sessions, which involves three tasks, including the imagination of repetitive self-paced left/right hand movements and the generation of words beginning with the same random letter. All 4 sessions were acquired on the same day, each lasting 4 minutes with 5-10 minutes breaks in between them. The subject performed a given task for about 15 seconds and then switched randomly to another task at the operator's request. In contrast to the Graz dataset, EEG data is not splitted in trials, since the subjects are continuously performing any of the mental tasks (i.e., no trial structure).

Data are provided in two ways: (1) raw EEG signals (with sampling rate = 512 Hz) recorded from 32 electrodes; (2) precomputed features. We use the precomputed features for numerical experiments. They were obtained by the power spectral density (PSD) in the band 8-30 Hz every 62.5 ms, (i.e., 16 times per second) over the last second of data with a frequency resolution of 2 Hz for the eight centro-parietal channels C_3, C_z, C_4, CP_1, CP_2, P_3, P_z, and P_4 after the raw EEG potentials were first spatially filtered by means of a surface Laplacian. As a result, an EEG sample is a 96-dimensional vector (eight channels times 12 frequency components). Requirements for comparative study are to provide an output every 0.5 second using the last second of data.

3.2. *Preprocessing*

We construct a data tensor from spectral EEG data. In what follows, labeled and unlabeled data tensors are denoted by $\boldsymbol{\mathcal{X}}_{train}$ and $\boldsymbol{\mathcal{X}}_{test}$, respectively.

3.2.1. *Graz dataset*

We obtain the time-frequency representation of the EEG data, by filtering it with complex Morlet wavelets, where the mother wavelet is given by

$$\Psi_0(\eta) = \pi^{-1/4} e^{iw_0\eta} e^{-\eta^2/2},$$

where w_0 is the characteristic eigenfrequency (generally taken to be 6). Scaling and temporal shifting of the mother wavelet, leads to $\Psi_{\tau,d(f)}$ controlled by the factor $\eta = (t-\tau)/d(f)$ where

$$d(f) = \frac{w_0 + \sqrt{2 + w_0^2}}{4\pi f},$$

where f is the main receptive frequency.

We denote by $C_{1,c,k}(t)$ and $C_{2,c,k}(t)$ the EEG waveforms measured from C_3 and C_4 channels, in the kth trial with label $c \in \{1,2\}$ (corresponding to left/right imagery hand movements). The wavelet transform of $C_{i,c,k}(t)$ $(i = 1,2)$ at time τ and frequency f is their convolution (denoted by $*$) with scaled and shifted wavelets. The amplitude of the wavelet transform is given by

$$x(f,i,\tau,c,k) = \left\| C_{i,c,k}(t) * \Psi_{\tau,d(f)}(t) \right\|,$$

for $f \in \{4,5,\dots,30\}$ Hz, $i = 1,2$ (C_3 and C_4 channels), $\tau = 1,\dots,T$ where T is the number of data points in each trial, $c = 1,2$ (left/right imagery hand movements), and $k = 1,\dots,K$ where K is the number of trials. The data tensor, $\boldsymbol{\mathcal{X}} \in \mathbb{R}^{27\times 2\times 2\times T\times K}$, is given by

$$\mathcal{X}_{f,i,c,\tau,k} \quad = \quad x(f,i,c,\tau,k). \tag{9}$$

3.2.2. *IDIAP dataset*

In this case, we use precomputed features (power spectral densities in the band $8-30$ Hz). Thus the data tensor $\boldsymbol{\mathcal{X}} \in \mathbb{R}^{12\times 8\times 3\times T}$ is constructed by normalizing spectral components $P_i(f,c,t)$ (precomputed features), i.e.,

$$\mathcal{X}_{f,i,c,t} = \frac{P_i(f,c,t)}{\sum_f P_i(f,c,t)}, \tag{10}$$

for $f \in \{8,10,\dots,28,30\}$ Hz, $i = 1,2,\dots,8$ (corresponding to 8 different channels, including C_3, C_z, C_4, CP_1, CP_2, P_3, P_z, and P_4), $c = 1,2,3$ (corresponding to three tasks such as left/right imagery hand movements and word generation), $t = 1,\dots,T$ where T is the number of data points (note that there is no trial structure in this dataset).

3.3. *Feature extraction*

We illustrate how to extract discriminative spectral features by applying NTF to preprocessed data. Depending on a way of constructing a data tensor $\boldsymbol{\mathcal{X}}$, classification results are slightly different.

3.3.1. *Data selection*

Instead of using whole training data, we discard outliers and select a portion of data which is expected to be more discriminative. Then these selected data are used as inputs to NTF, in order to learn component matrices $\boldsymbol{A}^{(n)}$. In some cases where the data tensor involves both temporal and trial coordinates, we use a whole data set without the data selection step, since it destroys the structure (for example, the case of $\boldsymbol{\mathcal{X}} \in \mathbb{R}^{27\times 2\times T\times 2K}$ in Table 2).

We use the nearest neighbor method for data selection. To this end, we calculate mean matrices $\overline{\boldsymbol{X}}_c$ for each class. For instance, given $\boldsymbol{\mathcal{X}} = [\mathcal{X}_{f,i,c,t,k}]$, the mean matrix for each c, $\overline{\boldsymbol{X}}_c$, is given by

$$\overline{\boldsymbol{X}}_c = \sum_{t=1}^{T}\sum_{k=1}^{K} \boldsymbol{\mathcal{X}}_{:,:,c,t,k}.$$

In the case of Graz dataset, $\overline{\boldsymbol{X}}_c \in \mathbb{R}^{27\times 2}$ ($c = 1,2$). For each class, we select slices $\boldsymbol{\mathcal{X}}_{:,:,c,t,k}$ which are nearest neighbors of the mean matrix. Denote by T_s the number of selected slices. We choose T_s which is not less than 43% of TK slices for each class in the case of Graz dataset. In the case of IDIAP dataset, we choose T_s which is not less than 95% of T slices. Imagery hand movements provide more prominent characteristics than the mental task of word generation. That is why the number of selected data points for Graz dataset is much smaller, compared to IDIAP dataset.

The data selection through the nearest neighbor method is useful, especially when the spectral characteristics of a mental task is not known. EEG data involving only motor imagery task, we can use sparseness and energy for data selection [25], because the μ rhythm is strongly activated on motor cortex. However, the spectral characteristics of EEG data involving the word generation is not well-known. Some existing work [14,27] state that such a mental task is related with gamma band between 30-100 Hz. Many BCI tasks have been worked in low frequency bands (below 30 Hz) because the sampling frequency is proportional to the amount of data set, which directly affects the real-time implementation. Cognitive task also increases in theta power between 3-7 Hz at bilateral and midline frontal scalp sites but it is not as prominent as the μ rhythm. If we use the sparseness measure and the power spectrum, these data can be discarded.

3.3.2. *NTF-based feature extraction*

We construct data tensors in various ways. For Graz dataset, $\boldsymbol{\mathcal{X}} \in \mathbb{R}^{27\times 2\times 2\times T\times K}$. For IDIAP dataset, $\boldsymbol{\mathcal{X}} \in \mathbb{R}^{12\times 8\times 3\times T}$. Table 2 summarizes the dimension of N-way data tensors ($N = 2,3,4$) used for NMF and NTF, after the data selection step. For instance, in the case of the 2-way tensor (Graz dataset), frequencies and channels are concatenated in rows and class labels and T_s are concatenated in columns, which leads to $\boldsymbol{X} \in \mathbb{R}^{54 \times 2T_s}$. One can easily figure out a way of constructing the rest of data tensors in Table 2, from their dimension.

Table 2. N-way data tensors used for NTF.

N-way tensor	Graz	IDIAP
2	$\boldsymbol{X} \in \mathbb{R}^{54 \times 2T_s}$	$\boldsymbol{X} \in \mathbb{R}^{96 \times 3T_s}$
3	$\boldsymbol{\mathcal{X}} \in \mathbb{R}^{27 \times 2 \times 2T_s}$	$\boldsymbol{\mathcal{X}} \in \mathbb{R}^{12 \times 8 \times 3T_s}$
4 (time)	$\boldsymbol{\mathcal{X}} \in \mathbb{R}^{27 \times 2 \times T \times 2K}$	
4 (class)	$\boldsymbol{\mathcal{X}} \in \mathbb{R}^{27 \times 2 \times 2 \times T_s}$	$\boldsymbol{\mathcal{X}} \in \mathbb{R}^{12 \times 8 \times 3 \times T_s}$

Applying NTF to data tensors listed in Table 2, leads to component matrices $\boldsymbol{A}^{(n)}$, $n = 1, \ldots, N$, following (7). Note that in the case of NMF, $\boldsymbol{X} = \boldsymbol{A}\boldsymbol{S}$, the encoding variable matrix $\boldsymbol{S}$ serves as features. By analogy with this, in the case of NTF, the Nth component matrix $\boldsymbol{A}^{(N)}$ serves as features. Given test data $\boldsymbol{\mathcal{X}}_{test}$, we have

$$\boldsymbol{A}^{(N)} \boldsymbol{S}_A^{(N)} = [\boldsymbol{\mathcal{X}}_{test}]_{(N)}, \tag{11}$$

where $[\boldsymbol{\mathcal{X}}_{test}]_{(N)}$ is the mode-N matricization of the tensor $\boldsymbol{\mathcal{X}}_{test}$. In the training phase, $\boldsymbol{S}_A^{(N)}$ is determined. Given $\boldsymbol{\mathcal{X}}_{test}$, there are two different ways to determine features $\boldsymbol{A}^{(N)}$:

- One way is to use a simple LS projection, leading to $[\boldsymbol{\mathcal{X}}_{test}]_{(N)} \left[\boldsymbol{S}_A^{(N)}\right]^{\dagger}$ where † represents the pseudo-inverse. In such a case, $\boldsymbol{A}^{(N)}$ is not a nonnegative matrix. However, in the viewpoint of feature extraction, it is acceptable. In fact, such a simple LS projection has been widely used for face recognition using NMF.
- The other way is to apply the NTF algorithm (8) to update only $\boldsymbol{A}^{(N)}$ with fixing other component matrices $\boldsymbol{A}^{(n)}$ for $n = 1, \ldots, N-1$.

Here we use the former method (LS projection) due to its lower complexity. In our numerical experiments, the former method gives slightly better classification accuracy ($1-2\%$ better) than the latter one.

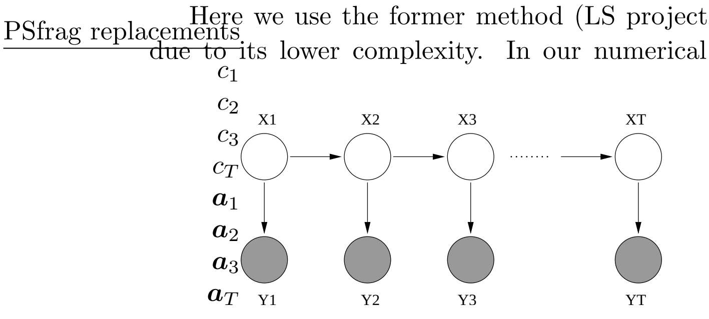


Fig. 3. The graphical model involving the Viterbi algorithm is shown.

Note that test data do not have label information. Thus, in constructing the test data tensor $\boldsymbol{\mathcal{X}}_{test}$, we duplicate test data C times in order to fill in the coordinate involving the class label.

3.4. *Classification*

3.4.1. *Graz data with trial structure*

For the single-trial online classification for Graz data (with trial structure), we use a Gaussian probabilistic model-based classifier [26] where Gaussian class-conditional probabilities for a single point in time t are integrated temporally by taking the expectation of the class probabilities with respect to the discriminative power at each point in time. In the case of the 4-way data tensor (with time) in Table 2, we do not need the integration step because it can find temporal bases suitable for classification for each channel and frequency. In such a case, the mode-4 matricization of the test data $\boldsymbol{\mathcal{X}} \in \mathbb{R}^{27 \times 2 \times t \times 1}$ at time t will make $\boldsymbol{\mathcal{X}}_{(4)} \in \mathbb{R}^{1 \times 54t}$ and its basis matrix $\boldsymbol{S}_A^{(4)} \in \mathbb{R}^{R \times 54t}$ calculated by $\boldsymbol{A}^{(1)} \in \mathbb{R}^{27 \times R}$, $\boldsymbol{A}^{(2)} \in \mathbb{R}^{2 \times R}$ and $\boldsymbol{A}^{(3)}_{(1:t,:)} \in \mathbb{R}^{t \times R}$.

3.4.2. *IDIAP data with no trial structure*

For the on-line classification for IDIAP data which consist of uncued EEG signals, we use the Viterbi algorithm [15] that is a dynamic programming algorithm for finding a most probable sequence of hidden states that explains a sequence of observations. The graphical model involving the Viterbi algorithm is shown in Fig. , where hidden states follow the first-order Markov chain and an observed variable at time t depends on only a hidden state at time t.

The dependency between hidden states at $t-1$ and t is defined by a transition probability $P(c_t|c_{t-1})$ and the dependency between observation at t and hidden state at t is defined by an emission probability $P(\boldsymbol{a}_t|c_t)$. In our case, hidden states correspond to class labels, i.e., $c_t \in \{1, 2, 3\}$ which are related to imagery left/right hand movements and the imagination of word generation. Therefore, transition probability can be defined by the 3×3 transition matrix $\Phi(c_t, c_{t-1})$ satisfied $\sum_{c_t} \Phi(c_t, c_{t-1}) = 1$. We should define the initial probability of hidden states $\pi(c_1)$ satisfied $\sum_{c_1} \pi(c_1) = 1$. Observed data $\boldsymbol{a}_t$ can be both discrete and continuous, in our case, observed data is continuous value that obtained by feature extraction, that is, $\boldsymbol{a}_t$ is the column vector of $\boldsymbol{A}^{(N)}$ in (11). We define the emission probability is normal distribution with mean $\boldsymbol{\mu}_{c_t}$ and covariance matrix $\boldsymbol{\Sigma}_{c_t}$,

$$\begin{aligned} P(\boldsymbol{a}_t|c_t) &= \mathcal{N}(\boldsymbol{a}_t|\boldsymbol{\mu}_{c_t}, \boldsymbol{\Sigma}_{c_t}) \\ &= (2\pi)^{-\frac{R}{2}} |\boldsymbol{\Sigma}_{c_t}|^{-\frac{1}{2}} \\ & \exp\left(-\frac{1}{2} (\boldsymbol{a}_t - \boldsymbol{\mu}_{c_t})^T \boldsymbol{\Sigma}_{c_t}^{-1} (\boldsymbol{a}_t - \boldsymbol{\mu}_{c_t}) \right). \end{aligned}$$

The initial probability $\pi(c_t)$, the transition matrix $\Phi(c_t, c_{t-1})$ and mean $\boldsymbol{\mu}_{c_t}$ and covariance matrix $\boldsymbol{\Sigma}_{c_t}$ of emission probability can be calculated by the features of training data.

$$\pi(c_1) = \frac{N_{c_1}}{\sum_{c_1=1}^{3} N_{c_1}}, \quad (12)$$

$$\Phi(c_t, c_{t-1}) = \frac{N_{c_{t-1},c_t}}{\sum_{c_t=1}^{3} N_{c_{t-1},c_t}}, \quad (13)$$

where N_c is the number of data in class $c \in \{1, 2, 3\}$ and $N_{c,d}$ is the number of transition from class $c \in \{1, 2, 3\}$ to class $d \in \{1, 2, 3\}$.

The mean $\boldsymbol{\mu}_{c_t}$ and the covariance matrix $\boldsymbol{\Sigma}_{c_t}$ of emission probability can be calculated by the feature of training data.

$$\begin{aligned} \boldsymbol{\mu}_c &= \frac{1}{N_c} \sum_{\boldsymbol{a}_t \in \mathcal{C}_c} \boldsymbol{a}_t, \\ \boldsymbol{\Sigma}_c &= \frac{1}{N_c - 1} \sum_{\boldsymbol{a}_t \in \mathcal{C}_c} (\boldsymbol{a}_t - \boldsymbol{\mu}_c)(\boldsymbol{a}_t - \boldsymbol{\mu}_c)^T, \end{aligned}$$

where $c \in \{1, 2, 3\}$, $t \in \{1, \ldots, T\}$ and $\mathcal{C}_c$ is a data set involved in the cth class.

After estimating all probabilities in the phase of training, how can we inference the hidden label given the test sequence? Given first data points, the hidden class information c_1 can calculate like this :

$$\begin{aligned} c_1^* &= \arg\max_{c_1} P(c_1|\boldsymbol{a}_1) \\ &= \arg\max_{c_1} P(c_1, \boldsymbol{a}_1) \\ &= \arg\max_{c_1} P(\boldsymbol{a}_1|c_1) P(c_1). \end{aligned}$$

The second equality is established because $P(\boldsymbol{a}_1)$ is not related with c_1. Continuing the next data points,

$$\begin{aligned} c_1^*, c_2^* &= \arg\max_{c_1,c_2} P(c_2, c_1, \boldsymbol{a}_1, \boldsymbol{a}_2) \\ &= \arg\max_{c_1,c_2} P(\boldsymbol{a}_2|c_2) P(c_2|c_1) P(c_1, \boldsymbol{a}_1) \\ &= \arg\max_{c_1,c_2} P(\boldsymbol{a}_2|c_2) P(c_2|c_1) \delta_1(c_1, \boldsymbol{a}_1) \\ & \vdots \\ c_1^*, \ldots, c_t^* &= \\ & \arg\max_{c_1,\ldots,c_t} P(\boldsymbol{a}_t|c_t) P(c_t|c_{t-1}) \delta_{t-1}(c_{t-1}, \boldsymbol{a}_{t-1}). \end{aligned}$$

We can calculate the most likely current hidden state only keeping the past probability $\delta_{t-1}(c_{t-1}, \boldsymbol{a}_{t-1})$ because dependency exists only between time t and time $t-1$. We feed the last seconds of data (16 data points) every 0.5 sec (every 8 data points) into Viterbi algorithm, then infer the most likely class.

4. Numerical Experiments

We apply our methods to Graz dataset in BCI competition II with single-trial classification of motor imagery task and IDIAP dataset in BCI competition III with continuous EEG classification of three classes. The overall structure is basically (1) preprocessing, (2) feature extraction with data selection, and (3) classification. Above mentioned, the analysis procedure is slightly different according to the data set and the data sturucture. In the case of Graz dataset, we need to change the data form from temporal one to spectral one using morlet Wavelet during preprocessing. However, in the case of IDIAP dataset, we do only normalization because we use the precomputed features. The other difference between data set is the classifier. The Graz dataset use the Gaussian probabilistic model combining the information across time (except for the 4-way tensor (time) structure in Table 2, because the temporal information can be explained by the the component matrix $\boldsymbol{A}^{(3)}$ which involved in the temporal dimension, in this case, we just use the Gaussian probabilistic model only taking a single time point into account), while the IDIAP dataset use the Viterbi algorithm because it has no trial structure, thus, there is no fixed tendency across time.

For Graz dataset, NMF is enough to extract the meaningful features which discriminate between classes, thus, the extension to NTF is not necessary. For

IDIAP dataset, the classification accuracy of NTF is better than one of NMF, moreover, the proposed method combining NTF and Viterbi algorithm works better than the winner's method proposed in BCI competition III. Thus, we will focus on comparing the bases for Graz dataset and the classification results for IDIAP dataset by doing various way tensor analyses.

4.1. *NMF v.s. NTF*

The basis vectors, the row vectors of $\boldsymbol{S}_A^{(N)}$ in (11) are illustrated in Fig. 6. they are obtained by NMF, 3-way NTF, and NTF with 4-way tensor (class) in Table 2 (from top to bottom) and the number of basis $r = 2, 4, 5, 6$ (from left to right). In each plot, top 1/2 is associated with C_3 and bottom 1/2 is contributed by C_4.

The first line shows the results of NMF for the number of basis, $r = 2, 4, 5, 6$ (from (a) to (d)). They finds the concatenated basis vector of spectral C3 and C4 channels [25]. As the number of basis vector increases, the spectral components such as μ rhythm (8-12 Hz), β rhythm (18-22 Hz), and sensori-motor rhythm (12-16 Hz), appear in the order of their importance. All rhythms have the property of contralateral dominance.

The second line shows the result of 3-way tensor. The basis vectors are the row vectors of $\boldsymbol{S}_A^{(3)}$ of 3-way tensor in (11), $\boldsymbol{\mathcal{X}} \in \mathbb{R}^{\text{freq.}\times\text{channel}\times\text{trial}}$. While NMF finds the sparse basis vectors which one of C3 (upper half part) C4 (lower half part) channels are activated, NTF finds the basis vectors which both channels are activated because it finds the basis for each channel separately by mode-n matricization as (8). The order of the activated frequecy bands is similar to the one of NMF.

The last line shows the basis vectors of NTF with 4-way tensor (class), $\boldsymbol{\mathcal{X}} \in \mathbb{R}^{\text{freq.}\times\text{channel}\times\text{class}\times\text{trial}}$. It consists of two lines: the upper one is for 'left' class and the lower one is for 'right' class. We can find that C3 channel (the upper part in one figure) of basis vectors are higher power than C4 channel (the upper part in one figure) for 'left' class and vice versa for 'right' class. These shows that ERD phenomenon has the contralateral property (ERD means event-related desynchronization, thus, it is correct that C3 channel located on the left hemisphere has higher power than C4 channel located on the right hemisphere).

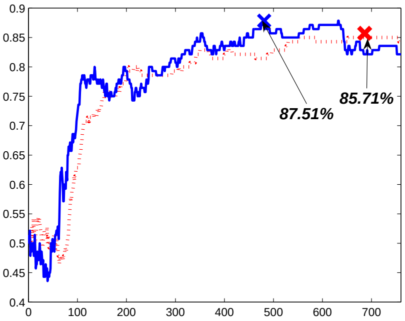


Fig. 4. The on-line classification result of Graz data. x- and y-axis means time and classification accuracy, repectively. The blue line shows the result across time using NTF with 4-way tensor for time and the red line shows the result using the Gaussian probabilistic model for each time point.

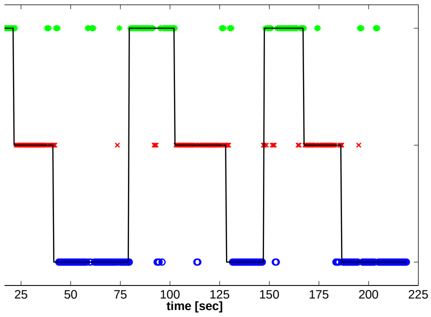
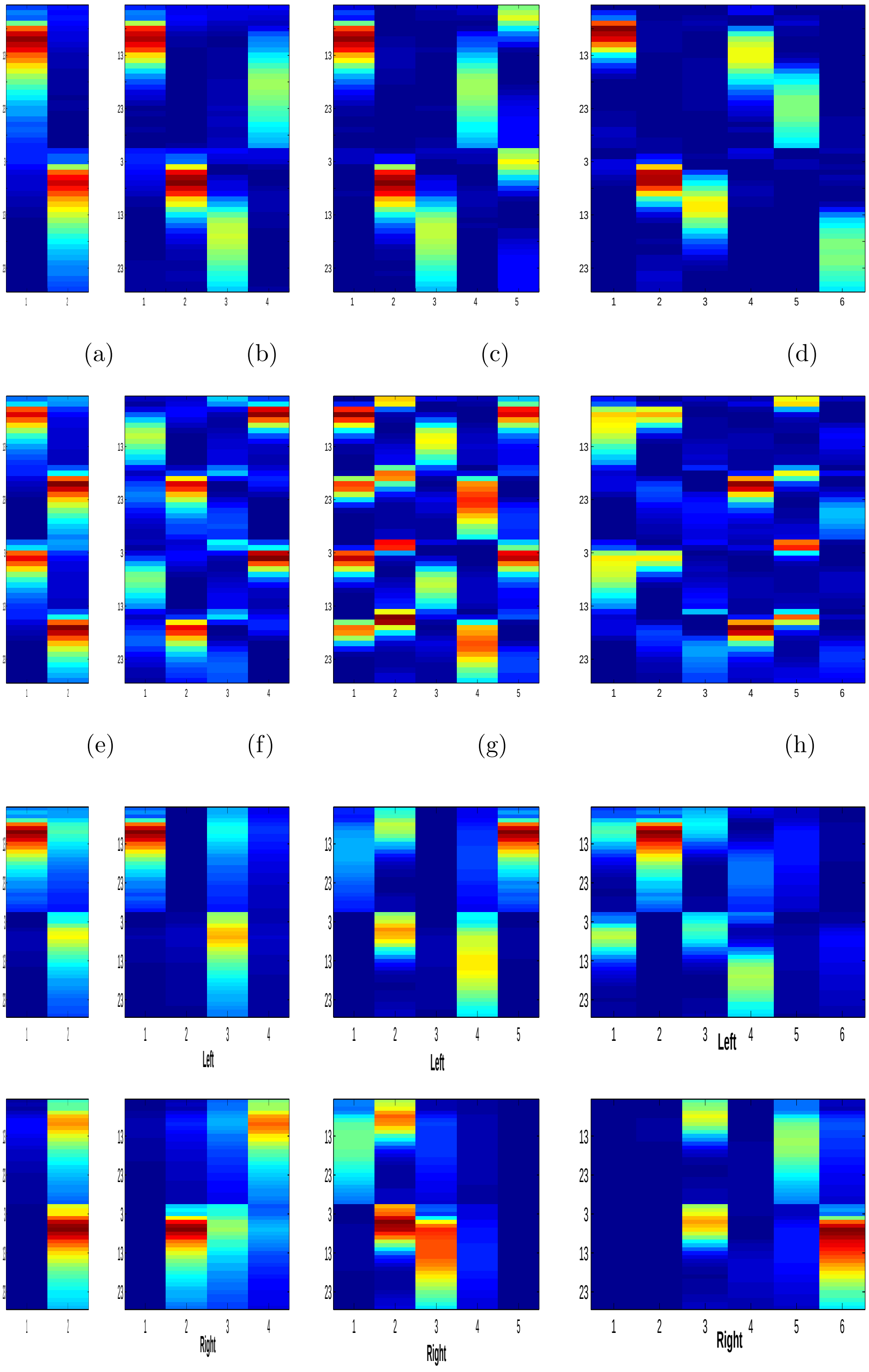


Fig. 5. Classification result across time. x- and y-axis represents time and class, respectively. The black line means true label and the points are classification result of Viterbi algorithm. The classification accuracy is 85.62 %.

4.2. *Classification results*

For the Graz data, the maximum result across time of 4-way NTF is 88.57 % at 5.36 sec that is same with NMF. But the maximum result of 3-way NTF is 81.51 %. From this, the sparse and discriminative basis vectors as shown in Fig. 6 can find more informative features and can improve the classification accuracy.

An interesting result of NTF is for 4-way tensor analysis that contains the temporal dimension (Table 2). In Fig. , the blue thick line shows the result across time when NTF with 4-way tensor (time) and the Gaussian probabilistic model no considering the temporal information are used. The red dotted line shows the result when NMF and the Gaussian probabilistic model combining the weight according to the



(a) (b) (c) (d)

(e) (f) (g) (h)

(i) (j) (k) (l)

Fig. 6. Basis vectors determined by NMF, NTF with 3-way tensor and NTF with 4-way tensor (from top to bottom) and the number of basis $r = 2, 4, 5, 6$ (from left to right). In each plot, top 1/2 is associated with C_3 and bottom 1/2 is contributed by C_4. In each of those, the vertical axis represents frequencies between 4 and 30 Hz, the horizontal axis is related to the number of basis vectors.

error rate on each time point [26,25] are used. Although the maximum result of NTF is less than the best result of NMF about 1 %, it shows that NTF can find the weighted importance of basis vectors across time automatically.

Table 3. Classification accuracy of IDIAP data

without Viterbi	sub1	sub2	sub3	avg
NMF ($\alpha = 0$)	75.34	39.63	38.53	51.17
NTF (3-way)	75.57	62.67	50.69	62.98
NTF (4-way)	77.63	65.67	52.52	65.27
with Viterbi				
NMF ($\alpha = 0$)	86.07	67.97	51.61	68.55
NTF (3-way)	85.62	69.35	53.44	69.47
NTF (4-way)	85.62	71.66	53.44	70.24
BCI comp. winner	79.60	70.31	56.02	68.65

Table 3 shows the classification results of IDIAP data. It separates the results into 'with Viterbi' and 'without Viterbi' according to whether the Viterbi algorithm is used or not. In the case of 'without Viterbi', we use the Gaussian probabilistic model. The winner of BCI competition III use the canonical variates transform for feature extraction and the DB discriminator working with an Euclidean metric for classification. From the classification results in Table 3, we can verify that the Viterbi algorithm which considers the temporal dependency works very well for continuous EEG classification. The transition probability $\Phi(c_t, c_{t-1})$ in (12) is like this:

$$\mathbf{\Phi} = \begin{bmatrix} 0.9976 & 0.0005 & 0.0019 \\ 0.0024 & 0.9969 & 0.0007 \\ 0.0009 & 0.0020 & 0.9971 \end{bmatrix}.$$

The probability that transfer the same class is much higher than others, this means that transition is not random, but highly dependent on previous state. Classification result across time with Viterbi algorithm for subject1 is shown in Fig. .

NTF with 4-way tensor that contains the class information (70.24 %) is little better than one with 3-way tensor (69.47 %). Both methods are better than the best result of BCI competition III (68.65 %) [5]. The result of subject1 is always better than the other, thus, we can say that the data set of subject1 is more informative than one of the other subjects. For subject1, the result of NMF is slightly better than the one of NTF. But for the other subject, NTF is better than NMF. Thus, we guess that NTF is more robust than NMF to find the hidden patterns from noisy training data.

5. Conclusions

We have presented an NTF-based method of feature extraction and the Viterbi algorithm for continuous EEG classification. Linear data model is convenient for selecting discriminative spectral features without the cross-validation several times. The NMF-based method, linear data model with nonnegativity constraint, could find discriminative and representative basis vectors (which reflected appropriate spectral characteristics) without cross-validation, which improved the on-line classification accuracy. In this paper, we considered more general framework that takes the multiway structure into account. NTF can find the hidden structures for new dimension such as time or class.

Continuous EEG classification can reduce the restriction of EEG experiment since it don't need the trial structure. The Viterbi algorithm can use the classifier to infer the possible hidden labels from observed data sequence. Our experiments show that it could be applied to uncued EEG classification successfully.

Acknowledgments

This work was supported by KOSEF International Cooperative Research Program and KOSEF Basic Research Program (grant R01-2006-000-11142-0).

References

1. H. Adeli, S. Ghosh-Dastidar, and N. Dadmehr. Disease: Models of computation and analysis of EEGs. *Clinical EEG and Neuroscience*, 36(3):131–140, 2005.
2. H. Adeli, S. Ghosh-Dastidar, and N. Dadmehr. A Wavelet-Chaos methodology for analysis of EEGs and EEG sub-bands to detect seizure and epilepsy. *IEEE Trans. Biomedical Engineering*, 54(2):205–211, Feb. 2007.
3. H. Adeli, Z. Zhou, and N. Dadmehr. Analysis of EEG records in an epileptic patient using wavelet transform. *Journal of Neuroscience Methods*, 123(1):69–87, 2003.
4. B. Blankertz, K. -R. Müller, G. Curio, T. M. Vaughan, G. Schalk, J. R. Wolpaw, A. Schlögl, C. Neuper, G. Pfurtscheller, T. Hinterberger, M. Schroder, and N. Birbaumer. The BCI competition 2003: Progress

and perspectives in detection and discrimination of EEG single trials. *IEEE Trans. Biomedical Engineering*, 51(6), 2004.

5. B. Blankertz, K. -R. Müller, D. J. Krusierski, G. Schalk, J. R. Wolpaw, A. Schlögl, G. Pfurtscheller, and N. Birbaumer. The BCI competition III: Validating alternative approaches to actual BCI problems. *IEEE Trans. Neural Systems and Rehabilitation Engineering*, 14:153–159, 2006.
6. A. Cichocki, R. Zdunek, and S. Amari. Csiszár's divergences for non-negative matrix factorization: Family of new algorithms. In *Proc. Int'l Conf. Independent Component Analysis and Blind Signal Separation*, Charleston, South Carolina, 2006.
7. A. Cichocki, R. Zdunek, and S. Amari. New algorithms for non-negative matrix factorization in applications to blind source separation. In *Proc. IEEE Int'l Conf. Acoustics, Speech, and Signal Processing*, Toulouse, France, 2006.
8. A. Cichocki, R. Zdunek, S. Choi, R. J. Plemmons, and S. Amari. Non-negative tensor factorization using alpha and beta divergences. In *Proc. IEEE Int'l Conf. Acoustics, Speech, and Signal Processing*, Honolulu, Hawaii, 2007.
9. A. Cichocki, R. Zdunek, S. Choi, R. J. Plemmons, and S. Amari. Novel multi-layer non-negative tensor factorization with sparsity constraints. In *Proc. Int'l Conf. Adaptive and Natural Computing Algorithms*, Warsaw, Poland, 2007.
10. L. de Lathauwer, B. de Moor, and J. Vandewalle. A multilinear singular value decomposition. *SIAM J. Matrix Anal. Appl.*, 21(4):1253–1278, 2000.
11. L. de Lathauwer, B. de Moor, and J. Vandewalle. One the best rank-1 and rank-$(R_1, R_2, \ldots, R_N)$ approximation of higher-order tensros. *SIAM J. Matrix Anal. Appl.*, 21(4):1324–1342, 2000.
12. I. S. Dhillon and S. Sra. Generalized nonnegative matrix approximations with Bregman divergences. In *Advances in Neural Information Processing Systems*, volume 18. MIT Press, 2006.
13. T. Ebrahimi, J. -F. Vesin, and G. Garcia. Brain-computer interface in multimedia communication. *IEEE Signal Processing Magazine*, 20(1):14–24, Jan. 2003.
14. S. P. Fitzgibbon, K. J. Pope, L. Mackenzie C. R. Clark, and J. O. Willoughby. Cognitive tasks augment gamma EEG power. *Clinical Neurophysiology*, 115(8):1802–1809, 2004.
15. G. D. Forney. The Viterbi algorithm. *Proc. of the IEEE*, 61:268–278, 1973.
16. S. Ghosh-Dastidar and H. Adeli. Improved spiking neural networks for EEG classification and epilepsy and seizure detection. *Integrated Computer-Aided Engineering*, 14(3), 2007.
17. T. Hazan, S. Polak, and A. Shashua. Sparse image coding using a 3D non-negative tensor factorization. In *Proc. Int'l Conf. Computer Vision*, Beijing, China, 2005.
18. J. del R. Millán. On the need for on-line learning in brain-computer interfaces. In *Proc. Int'l Joint Conf. Neural Networks*, Budapest, Hungary, 2004.
19. H. A. L. Kiers. Towards a standardized notation and terminology in multiway analysis. *Journal of Chemometrics*, 14:105–122, 2000.
20. T. N. Lal, M. Schroder, T. Hinterberger, J. Weston, M. Bogdan, N. Birbaumer, and B. Schölkopf. Support vector channel selection in BCI. Technical Report 120, Max Planck Institute for Biological Cybernetics, 2003.
21. D. D. Lee and H. S. Seung. Learning the parts of objects by non-negative matrix factorization. *Nature*, 401:788–791, 1999.
22. D. D. Lee and H. S. Seung. Algorithms for non-negative matrix factorization. In *Advances in Neural Information Processing Systems*, volume 13. MIT Press, 2001.
23. H. Lee and S. Choi. PCA-based linear dynamical systems for multichannel EEG classification. In *Proc. Int'l Conf. Neural Information Processing*, pages 745–749, Singapore, 2002.
24. H. Lee and S. Choi. PCA+HMM+SVM for EEG pattern classification. In *Proc. Int'l Symp. Signal Processing and Its Applications*, pages 541–544, Paris, France, 2003.
25. H. Lee, A. Cichocki, and S. Choi. Nonnegative matrix factorization for motor imagery EEG classification. In *Proc. Int'l Conf. Artificial Neural Networks*, Athens, Greece, 2006. Springer.
26. S. Lemm, C. Schäfer, and G. Curio. BCI competition 2003-data set III: Probabilistic modeling of sensorimotor μ rhythms for classification of imaginary hand movements. *IEEE Trans. Biomedical Engineering*, 51(6), 2004.
27. H. Liu, J. Wang, C. Zheng, and P. He. Study on the effect of different frequency bands of EEG signals on mental tasks classification. In *Proc. 27th Conf. IEEE Eng. in Medicine and Biology*, Shanghai, China, 2005.
28. M. Mørup, L. K. Hansen, C. S. Herrmann, J. Parnas, and S. M. Arnfred. Parallel factor analysis as an exploratory tool for wavelet transformed event-related EEG. *NeuroImage*, 29:938–947, 2006.
29. P. Paatero and U. Tapper. Least squares formulation of robust non-negative factor analysis. *Chemometrics Intelligent Laboratory Systems*, 37:23–35, 1997.
30. L. C. Parra, C. D. Spence, and A. D. Gerson. Recipies for the linear analysis of EEG. *NeuroImage*, 28:326–341, 2005.
31. G. Pfurtscheller and F. H. Lopes da Silva. Event-related EEG/MEG synchronization and desynchronization: Basic principles. *Clinical Neurophysiology*, 110(11):1842–1857, 1999.
32. H. Ramoser, J. Müller-Gerking, and G. Pfurtscheller. Optimal spatial filtering of single trial EEG during imagined hand movement. *IEEE Trans. Rehabilitation Engineering*, 8:441–446, 2000.

33. A. Shashua and T. Hazan. Non-negative tensor factorization with applications to statistics and computer vision. In *Proc. Int'l Conf. Machine Learning*, Bonn, Germany, 2005.
34. M. Welling and M. Weber. Positive tensor factorization. *Pattern Recognition Letters*, 22:1255–1261, 2001.
35. J. R. Wolpaw, N. Birbaumer, D. J. McFarland, G. Pfurtscheller, and T. M. Vaughan. Brain-computer interfaces for communication and control. *Clinical Neurophysiology*, 113:767–791, 2002.