Neurocomputing 72 (2009) 3182–3190

Contents lists available at ScienceDirect

Neurocomputing

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

Kernel nonnegative matrix factorization for spectral EEG feature extraction

Hyekyoung Lee [a], Andrzej Cichocki [b], Seungjin Choi [a,*]

[a] Department of Computer Science, Pohang University of Science and Technology, San 31 Hyoja-dong, Nam-gu, Pohang 790-784, Republic of Korea
[b] Laboratory for Advanced Brain Signal Processing, Brain Science Institute, RIKEN 2-1 Hirosawa, Wako-shi, Saitama 351-0198, Japan

ARTICLE INFO

Article history:
Received 25 June 2008
Received in revised form
18 February 2009
Accepted 2 March 2009
Communicated by T. Heskes
Available online 1 April 2009

Keywords:
EEG classification
Feature extraction
Kernel methods
Multiplicative updates
Nonnegative matrix factorization

ABSTRACT

Nonnegative matrix factorization (NMF) seeks a decomposition of a nonnegative matrix $\boldsymbol{X}\geqslant 0$ into a product of two nonnegative factor matrices $\boldsymbol{U}\geqslant 0$ and $\boldsymbol{V}\geqslant 0$, such that a discrepancy between $\boldsymbol{X}$ and $\boldsymbol{UV}^\top$ is minimized. Assuming $\boldsymbol{U}=\boldsymbol{XW}$ in the decomposition (for $\boldsymbol{W}\geqslant 0$), kernel NMF (KNMF) is easily derived in the framework of least squares optimization. In this paper we make use of KNMF to extract discriminative spectral features from the time–frequency representation of electroencephalogram (EEG) data, which is an important task in EEG classification. Especially when KNMF with linear kernel is used, spectral features are easily computed by a matrix multiplication, while in the standard NMF multiplicative update should be performed repeatedly with the other factor matrix fixed, or the pseudo-inverse of a matrix is required. Moreover in KNMF with linear kernel, one can easily perform feature selection or data selection, because of its sparsity nature. Experiments on two EEG datasets in brain computer interface (BCI) competition indicate the useful behavior of our proposed methods.

1. Introduction

Nonnegative matrix factorization (NMF) is a multivariate analysis method which is proven to be useful in learning a faithful representation of nonnegative data such as images, spectrograms, and documents [16,9]. NMF seeks a decomposition of a nonnegative data matrix $\boldsymbol{X}\geqslant 0$ into a product of two nonnegative factor matrices such that $\boldsymbol{X}\approx\boldsymbol{UV}^\top$ for $\boldsymbol{U}\geqslant 0$ and $\boldsymbol{V}\geqslant 0$. NMF allows only nonsubtractive combinations of nonnegative basis vectors to approximate the original nonnegative data, possibly providing a parts-based representation [9].

Electroencephalogram (EEG) is the most popular sensory signal used for brain computer interface (BCI). NMF was shown to be useful in determining discriminative basis vectors which well reflect meaningful spectral characteristics without the cross-validation in a motor imagery task [11]. Exemplary spectral characteristics of EEG involving motor, might be μ rhythm (8–12 Hz) 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) [17]. ERD and ERS could be used as relevant features for the task of motor imagery EEG classification. However, those phenomena might happen in different frequency bands, depending on subjects, for instance, in 16–20 Hz, not in 8–12 Hz [8]. Thus, it is desirable to develop a data-driven feature extraction method which automatically determine discriminative spectral bands. To this end, NMF was proposed as a promising method and was proven to be a good solution [11,3].

Several interesting variations of NMF were recently proposed in [15], including convex-NMF, semi-NMF, kernel NMF (KNMF), and so on. These extensions were mainly discussed in a task of clustering, emphasizing their applicability to a data matrix without sign restriction. However, we pay our attention to KNMF in the case of nonnegative data matrix, which can be easily developed, assuming $\boldsymbol{U}=\boldsymbol{XW}$ for $\boldsymbol{W}\geqslant 0$ in the decomposition of $\boldsymbol{X}=\boldsymbol{UV}^\top$. We derive a multiplicative updating algorithm for KNMF from KKT conditions, as in our recent work [3], where α-divergence was used as an error measure for NMF.

In this paper we make use of KNMF to extract discriminative spectral features from the time–frequency representation of EEG data, which is an important task in EEG classification. The main contribution of this paper is to show how KNMF is applied to extract spectral EEG features, emphasizing its useful aspects that are summarized below:

- When KNMF with linear kernel is used, spectral features are easily computed by a matrix multiplication, while in the standard NMF multiplicative update should be performed repeatedly with the other factor matrix fixed, or the pseudo-inverse of a matrix is required.
- KNMF with linear kernel is a special case of convex-NMF [4], provided that each column in $\boldsymbol{W}$ satisfies the sum-to-one constraint. We show that KNMF yields more sparse

* Corresponding author. Tel.: +82 54 279 2259; fax: +82 54 279 2299.
E-mail address: seungjin@postech.ac.kr (S. Choi).
URL: http://www.postech.ac.kr/~seungjin (S. Choi).

0925-2312/$ - see front matter © 2009 Elsevier B.V. All rights reserved.
doi:10.1016/j.neucom.2009.03.005

representation, compared to the standard NMF in the case of spectral EEG data, confirming the result that was first studied in [4]. In addition, we show that feature selection or data selection can be easily performed in the case of KNNF with linear kernel, because of its sparsity nature.
- All these useful behaviors of KNMF are verified through experiments on two EEG datasets in BCI competition.

The rest of this paper is organized as follows. The next section provides a brief overview of NMF, illustrating how spectral features are extracted from EEG data using NMF. Section 3 explains KNMF and its application to spectral EEG feature extraction. Experiments on two EEG datasets in BCI competition are presented in Section 4. Finally conclusions are drawn in Section 5.

2. NMF for spectral EEG feature extraction

We begin with illustrating how to construct a data matrix from EEG data.

2.1. Data matrix construction

We construct the data matrix $\boldsymbol{X} \in \mathbb{R}^{n\times m}$ from the time-domain EEG signal such that each column vector in $\boldsymbol{X}$ is associated with the frequency profile over trials.

We denote by $c_t^{(k)}$ the time-domain EEG signal measured at the kth channel ($k = 1,\ldots,K$). We also denote by $P_{t,f}^{(k)}$ (time and frequency indices run over $t = 1,\ldots,n$ and $f = 1,\ldots,F$, respectively) the corresponding time–frequency representation of the EEG signal $c_t^{(k)}$, which is typically computed by short-time Fourier transform (STFT) or wavelet transform. Then the spectral matrix is given by $\boldsymbol{P}^{(k)} = [P_{t,f}^{(k)}] \in \mathbb{R}^{n\times F}$. In the case of EEG data with N trials, the time index is given by $n = TN$ where T is the number of samples in each trial.

We provide a brief illustration of how to compute $P_{t,f}^{(k)}$ using complex Morlet wavelet transform which is very popular in handling EEG data. Time–frequency representation of EEG data is computed by filtering it with complex Morlet wavelets, where the mother wavelet is given by

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

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

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

where f is the main receptive frequency. The wavelet transform of $c_t^{(k)}$ at time τ and frequency f is their convolution with scaled and shifted wavelets. The amplitude of the wavelet transform, $P_{\tau,f}^{(k)}$, is given by

$$P_{\tau,f}^{(k)} = \|c_t^{(k)} * \Psi_{\tau,d(f)}(t)\|, \tag{3}$$

for $k = 1,\ldots,K$.

Then we construct the data matrix $\boldsymbol{X}$ by collecting $n \times F$ spectral matrices computed at K different channels,

$$\boldsymbol{X} = [\boldsymbol{P}^{(1)}, \boldsymbol{P}^{(2)}, \ldots, \boldsymbol{P}^{(K)}] \in \mathbb{R}^{n\times m}, \tag{4}$$

where $m = KF$. Fig. 3 shows the time-domain EEG signals in the upper panel which are measured at eight different channels $(C_3, C_z, C_4, CP_1, CP_2, P_3, P_z, P_4)$ and their corresponding time–frequency representation in the lower panel, where the horizontal axis represents frequency and the vertical axis is associated with time.

2.2. NMF and feature extraction

NMF seeks a decomposition of $\boldsymbol{X} \in \mathbb{R}^{n\times m}$ that is of the form

$$\boldsymbol{X} \approx \boldsymbol{U}\boldsymbol{V}^\top, \tag{5}$$

where $\boldsymbol{U} \in \mathbb{R}^{n\times r}$ and $\boldsymbol{V} \in \mathbb{R}^{m\times r}$ are restricted to be nonnegative matrices as well. Matrices $\boldsymbol{U}$ and $\boldsymbol{V}$, in our problem setting, are interpreted as follows (see also Fig. 4).

- Column vectors in $\boldsymbol{V}$ are *basis vectors* which reflects r representative spectral characteristics learned from an ensemble of EEG data samples.

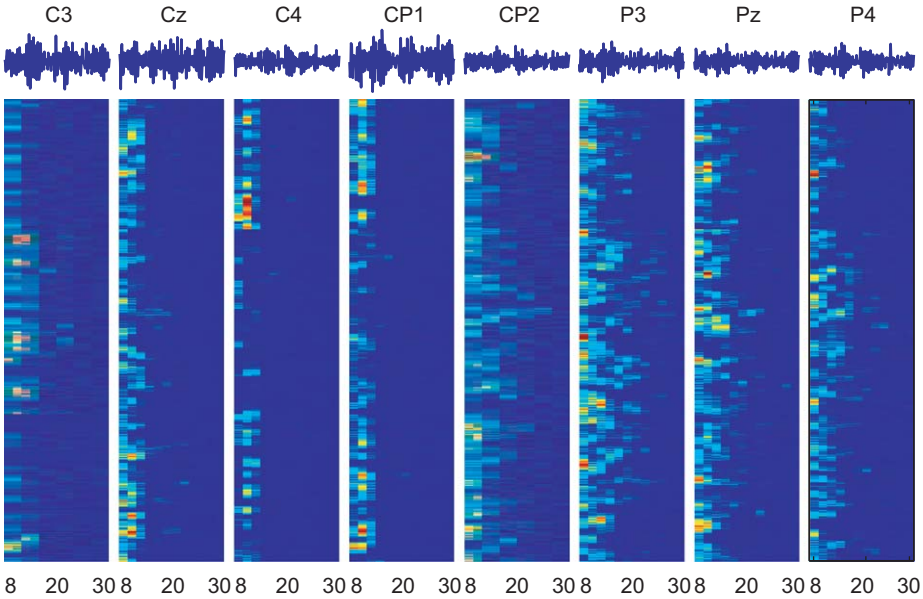

Fig. 1. The classification accuracy averaged over 140 trials is plotted for three methods (NMF, KNMF, and FS). All three methods work well, while KNMF is slightly better than other two methods.

- Row vectors in $\boldsymbol{U}$ are *features*, reflecting how learned spectral patterns are encoded for spectral data vectors (associated with row vectors of $\boldsymbol{X}$).

We provide a simple alternative derivation of multiplicative updates for least squares (LS) NMF, which was originally developed by Lee and Seung [10]. We consider LS objective function given by

$$\mathscr{J} = \tfrac{1}{2}\|\boldsymbol{X} - \boldsymbol{U}\boldsymbol{V}^\top\|^2, \tag{6}$$

where $\|\boldsymbol{X}\|$ denotes the Frobenius norm of a matrix $\boldsymbol{X}$. Taking nonnegative constraints $\boldsymbol{U} \geqslant 0$ and $\boldsymbol{V} \geqslant 0$ into account, the Lagrangian is given by

$$\mathscr{L} = \tfrac{1}{2}\|\boldsymbol{X} - \boldsymbol{U}\boldsymbol{V}^\top\|^2 - \mathrm{tr}\{\boldsymbol{\Lambda}\boldsymbol{U}^\top\} - \mathrm{tr}\{\boldsymbol{\Omega}\boldsymbol{V}^\top\}, \tag{7}$$

where $\boldsymbol{\Lambda} \geqslant 0$ and $\boldsymbol{\Omega} \geqslant 0$ are Lagrangian multipliers.

KKT optimality conditions require

$$\boldsymbol{U}\boldsymbol{V}^\top\boldsymbol{V} - \boldsymbol{X}\boldsymbol{V} = \boldsymbol{\Lambda}, \tag{8}$$

$$\boldsymbol{V}\boldsymbol{U}^\top\boldsymbol{U} - \boldsymbol{X}^\top\boldsymbol{U} = \boldsymbol{\Omega}, \tag{9}$$

which result from $\partial\mathscr{L}/\partial\boldsymbol{U} = 0$ and $\partial\mathscr{L}/\partial\boldsymbol{V} = 0$, respectively. KKT complementary slackness conditions require:

$$[\boldsymbol{U}\boldsymbol{V}^\top\boldsymbol{V} - \boldsymbol{X}\boldsymbol{V}] \odot \boldsymbol{U} = 0, \tag{10}$$

$$[\boldsymbol{V}\boldsymbol{U}^\top\boldsymbol{U} - \boldsymbol{X}^\top\boldsymbol{U}] \odot \boldsymbol{V} = 0, \tag{11}$$

where $\odot$ denotes Hadarmard product (element-wise multiplication). These suggest the following multiplicative updates:

$$\boldsymbol{U} \leftarrow \boldsymbol{U} \odot \frac{\boldsymbol{X}\boldsymbol{V}}{\boldsymbol{U}\boldsymbol{V}^\top\boldsymbol{V}}, \tag{12}$$

$$\boldsymbol{V} \leftarrow \boldsymbol{V} \odot \frac{\boldsymbol{X}^\top\boldsymbol{U}}{\boldsymbol{V}\boldsymbol{U}^\top\boldsymbol{U}}, \tag{13}$$

where the division is performed in an element-wise manner. These rules are exactly same as the one proposed by Lee and Seung [10].

When a test data matrix $\boldsymbol{X}_{test}$ is given, its associated feature matrix $\boldsymbol{U}_{test}$ can be computed in two different ways:

- The feature matrix $\boldsymbol{U}_{test}$ is determined by LS projection,

$$\boldsymbol{U}_{test} = \boldsymbol{X}_{test}[\boldsymbol{V}^\top]^\dagger, \tag{14}$$

  where † represents the pseudo-inverse. In such a case, $\boldsymbol{U}_{test}$ might have negative elements but work well [11].
- We iterate the update rule (12) until convergence, with $\boldsymbol{V}$ (learned in the training phase) fixed.

Sparseness or orthogonality constraints were suggested for more fruitful representation in the framework of NMF [14,6,2].

3. Kernel NMF

3.1. Algorithm

A simple trick to develop kernel NMF is to assume that feature profiles (column vectors of $\boldsymbol{U}$) are convex combinations of frequency profiles (column vectors of $\boldsymbol{X}$) of spectral data in NMF decomposition, i.e.,

$$\boldsymbol{U} = \boldsymbol{X}\boldsymbol{W}, \tag{15}$$

where each column in $\boldsymbol{W}$ satisfies the sum-to-one constraint. Incorporating the assumption (15) into the LS objective function (6) yields

$$\mathscr{J}_K = \tfrac{1}{2}\|\boldsymbol{X} - \boldsymbol{X}\boldsymbol{W}\boldsymbol{V}^\top\|^2. \tag{16}$$

In order to handle inequality constraints $\boldsymbol{W} \geqslant 0$ and $\boldsymbol{V} \geqslant 0$, we consider the Lagrangian, as in (7),

$$\mathscr{L}_K = \tfrac{1}{2}\|\boldsymbol{X} - \boldsymbol{X}\boldsymbol{W}\boldsymbol{V}^\top\|^2 - \mathrm{tr}\{\boldsymbol{\Lambda}\boldsymbol{W}^\top\} - \mathrm{tr}\{\boldsymbol{\Omega}\boldsymbol{V}^\top\}, \tag{17}$$

where $\boldsymbol{\Lambda}$ and $\boldsymbol{\Omega}$ are Lagrangian multipliers.

As in (8) and (10), KKT optimality conditions require:

$$\boldsymbol{X}^\top\boldsymbol{X}\boldsymbol{W}\boldsymbol{V}^\top\boldsymbol{V} - \boldsymbol{X}^\top\boldsymbol{X}\boldsymbol{V} = \boldsymbol{\Lambda}, \tag{18}$$

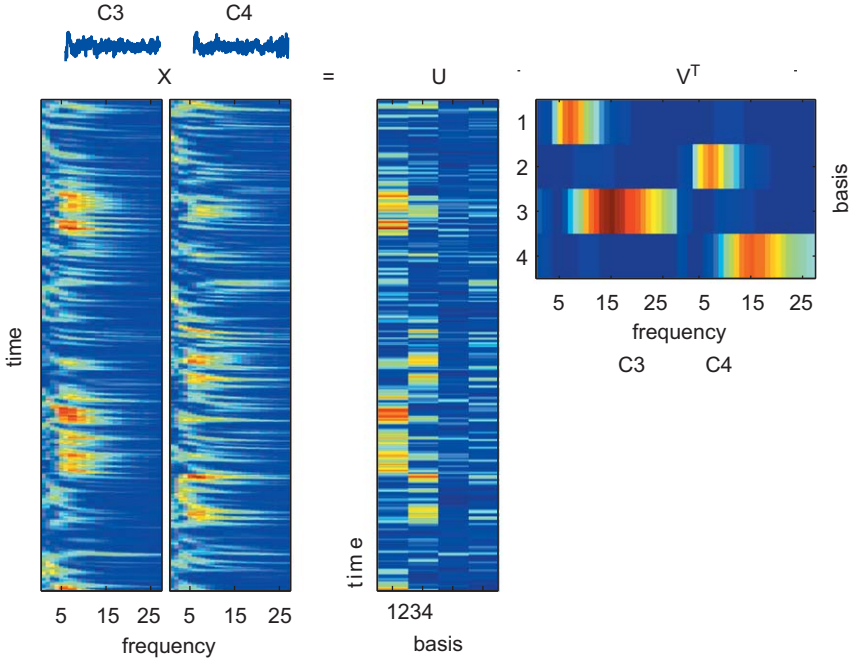

Fig. 2. Basis vectors computed by NMF and KNMF, as well as representative patterns selected by FS, are shown. In each plot, the vertical axis represents frequency components between 4 and 30 Hz. The upper half is for C_3 and the lower half is for C_4. The number of factors was chosen as $r = 6$ for both NMF and KNMF. FS chose 10 features, where blue color is associated with "not-selected". Its selected pattern is closely related to NMF and KNMF factors. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

$$\boldsymbol{V}\boldsymbol{W}^{\top}\boldsymbol{X}^{\top}\boldsymbol{X}\boldsymbol{W}-\boldsymbol{X}^{\top}\boldsymbol{X}\boldsymbol{W}=\boldsymbol{\Omega}, \tag{19}$$

which are the consequences of $\partial\mathscr{L}_K/\partial\boldsymbol{W}=0$ and $\partial\mathscr{L}_K/\partial\boldsymbol{V}=0$, respectively.

Define a kernel matrix (Gram matrix) as $\boldsymbol{K}=\boldsymbol{X}^{\top}\boldsymbol{X}$, the (i,j)-entry of which represents a similarity between two frequency profiles i and j. Then, KKT complementary slackness conditions yield

$$[\boldsymbol{K}\boldsymbol{W}\boldsymbol{V}^{\top}\boldsymbol{V}-\boldsymbol{K}\boldsymbol{V}]\odot\boldsymbol{W}=0, \tag{20}$$

$$[\boldsymbol{V}\boldsymbol{W}^{\top}\boldsymbol{K}\boldsymbol{W}-\boldsymbol{K}\boldsymbol{W}]\odot\boldsymbol{V}=0. \tag{21}$$

These relations lead to the following multiplicative updates:

$$\boldsymbol{V}\leftarrow\boldsymbol{V}\odot\frac{\boldsymbol{K}\boldsymbol{W}}{\boldsymbol{V}\boldsymbol{W}^{\top}\boldsymbol{K}\boldsymbol{W}}, \tag{22}$$

$$\boldsymbol{W}\leftarrow\boldsymbol{W}\odot\frac{\boldsymbol{K}\boldsymbol{V}}{\boldsymbol{K}\boldsymbol{W}\boldsymbol{V}^{\top}\boldsymbol{V}}. \tag{23}$$

The first idea on KNMF was proposed in [4]. We further elaborate it, developing multiplicative updates in a different way as well as applying it to a task of EEG classification. As in [4], one can easily extend KNMF to the case where the data matrix is free from sign restriction. Throughout this paper we use only linear kernel $\boldsymbol{K}=\boldsymbol{X}^{\top}\boldsymbol{X}$. However, the method holds for $\boldsymbol{K}=\boldsymbol{\Phi}^{\top}\boldsymbol{\Phi}$ where $\boldsymbol{\Phi}=[\phi(X_{ij})]$ is a transformed matrix in a feature space.

We also provide an alternative derivation of rules (22) and (23). Suppose that the gradient of the objective function (16) has a decomposition of the form

$$\nabla\mathscr{J}_K=[\nabla\mathscr{J}_K]^{+}-[\nabla\mathscr{J}_K]^{-}, \tag{24}$$

where $[\nabla\mathscr{J}_K]^{+}>0$ and $[\nabla\mathscr{J}_K]^{-}>0$. In such a case, multiplicative update for parameters Θ has the form

$$\Theta\leftarrow\Theta\odot\left(\frac{[\nabla\mathscr{J}_K]^{-}}{[\nabla\mathscr{J}_K]^{+}}\right)^{\cdot\eta}, \tag{25}$$

where $(\cdot)^{\cdot\eta}$ denotes the element-wise power and η is a learning rate ($0<\eta\leqslant 1$). It can be easily seen that the multiplicative update (25) preserves the nonnegativity of the parameter Θ, while $\nabla\mathscr{J}_K=0$ when the convergence is achieved [2].

Note that derivatives with respect to $\boldsymbol{W}$ and $\boldsymbol{V}$ are computed as

$$\begin{aligned}\frac{\partial\mathscr{J}_K}{\partial\boldsymbol{W}}&=\left[\frac{\partial\mathscr{J}_K}{\partial\boldsymbol{W}}\right]^{+}-\left[\frac{\partial\mathscr{J}_K}{\partial\boldsymbol{W}}\right]^{-}\\&=\boldsymbol{K}\boldsymbol{W}\boldsymbol{V}^{\top}\boldsymbol{V}-\boldsymbol{K}\boldsymbol{V},\end{aligned}$$

$$\begin{aligned}\frac{\partial\mathscr{J}_K}{\partial\boldsymbol{V}}&=\left[\frac{\partial\mathscr{J}_K}{\partial\boldsymbol{V}}\right]^{+}-\left[\frac{\partial\mathscr{J}_K}{\partial\boldsymbol{V}}\right]^{-}\\&=\boldsymbol{V}\boldsymbol{W}^{\top}\boldsymbol{K}\boldsymbol{W}-\boldsymbol{K}\boldsymbol{W}.\end{aligned}$$

Thus, invoking the relation (25) with $\eta=1$, one can easily derive multiplicative updates (22) and (23).

3.2. Sparsity of KNMF

Our KNMF with linear kernel is a special case of convex-NMF [15]. Thus, it follows from the result in [15] that factor matrices $\boldsymbol{W}$ and $\boldsymbol{V}$ in KNMF are naturally sparse. We briefly review the result on this sparsity [15]. Then we provide an illustrative example, showing a useful behavior of KNMF due to the sparsity property and confirming the sparsity of factor matrices in KNMF when it was applied to spectral EEG data.

Suppose that the singular value decomposition (SVD) of the data matrix $\boldsymbol{X}$ is given by $\boldsymbol{X}=\boldsymbol{P}\boldsymbol{\Sigma}\boldsymbol{Q}^{\top}$. Then, the objective value in

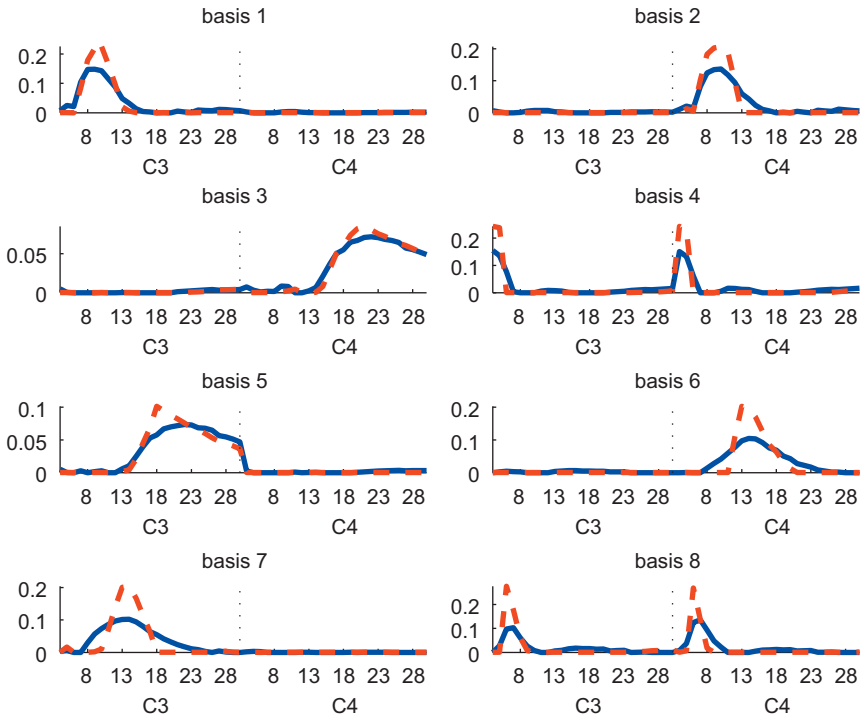

Fig. 3. Exemplary EEG data (IDIAP dataset, the details on which are explained in Section 4.1) are shown in the time-domain (upper panel) and in the time–frequency domain (lower panel). Waveforms of EEG in the time-domain are shown in the upper panel, each of which is measured at eight different channels $(C_3, C_z, C_4, CP_1, CP_2, P_3, P_z, P_4)$. Corresponding time–frequency representations are shown in the lower panel, where frequency (horizontal axis in each plot) ranges over $[8, 10, 12, \ldots, 28, 30]$ (i.e., the number of frequency bands is 12). In this case, the data matrix $\boldsymbol{X}\in\mathbb{R}^{n\times 96}$ is constructed by collecting 12 frequency profiles at each channel $(96=12\times 8)$.

the optimization problem (16) can be rewritten as

$$\|\boldsymbol{X}-\boldsymbol{X}\boldsymbol{W}\boldsymbol{V}^{\top}\|^2 = \mathrm{tr}\{(\boldsymbol{I}-\boldsymbol{V}\boldsymbol{W}^{\top})\boldsymbol{X}^{\top}\boldsymbol{X}(\boldsymbol{I}-\boldsymbol{W}\boldsymbol{V}^{\top})\} = \sum_{j=1}^{m}\sigma_j^2\|\boldsymbol{q}_j^{\top}(\boldsymbol{I}-\boldsymbol{W}\boldsymbol{V}^{\top})\|^2, \quad (26)$$

where σ_j is the jth diagonal entry of $\boldsymbol{\Sigma}$, $\boldsymbol{q}_j$ is the jth column vector of $\boldsymbol{Q}$, and $\boldsymbol{X}^{\top}\boldsymbol{X}=\sum_{j=1}^{m}\sigma_j^2\boldsymbol{q}_j\boldsymbol{q}_j^{\top}$ for $m<n$.

The term $\boldsymbol{q}_j^{\top}(\boldsymbol{I}-\boldsymbol{W}\boldsymbol{V}^{\top})$ in (26) is the projection of $(\boldsymbol{I}-\boldsymbol{W}\boldsymbol{V}^{\top})$ onto a principal direction, so the optimization problem (16) can be considered as the minimization of the sum of the differences between the data and its projected values. One can easily see that in (26) the projection of $(\boldsymbol{I}-\boldsymbol{W}\boldsymbol{V}^{\top})$ onto principal directions has larger weights while the projection of $(\boldsymbol{I}-\boldsymbol{W}\boldsymbol{V}^{\top})$ onto nonprincipal directions has smaller weights. It was pointed out that the sparsity is enforced heavily in the principal subspace and lightly in the nonprincipal subspace. It was shown in [15] that KNMF with linear kernel tends to provide sparse solution.

Fig. 5 shows bases computed by NMF and KNMF from Graz data (the details on which are described in Section 4.2) where C_3 and C_4 channels were used in motor imagery EEG data. Bases are associated with the distributions of frequency components in the range of 4–30 Hz. Certainly, one can easily see that bases in KNMF are sparser than those in NMF.

Now we provide an illustrative example, showing the sparsity of factor matrices in KNMF is useful in determining discriminative basis vectors, compared to the standard NMF. To this end, we first generate a data matrix $\boldsymbol{X}$:

$$\boldsymbol{X}=\begin{pmatrix}0.3692 & 0.1320 & 0.8212 & 0.4509 & 1.3685\\ 0.1112 & 0.9421 & 0.0154 & 0.5470 & 1.6256\\ 0.7803 & 0.9561 & 0.0430 & 0.2963 & 1.7802\\ 0.3897 & 1.5752 & 1.1690 & 0.7447 & 0.0811\\ 0.2417 & 1.0598 & 1.6491 & 0.1890 & 0.9294\\ 0.4039 & 1.2348 & 1.7317 & 0.6868 & 0.7757\\ 0.0965 & 1.3532 & 1.6477 & 0.1835 & 0.4868\end{pmatrix}.$$

First three rows were generated by adding uniformly distributed ($\mathcal{U}[0,1]$) random numbers to $[0,0,0,0,1]$. Last four rows were constructed by adding uniformly distributed ($\mathcal{U}[0,1]$) random numbers to $[0,1,1,0,0]$.

In an ideal case, it is expected that NMF or KNMF compute two representative basis vectors, each of which is $[0,0,0,0,1]^{\top}$ and $[0,1,1,0,0]^{\top}$, respectively, and identify two clusters (first three rows and last four rows of $\boldsymbol{X}$). Basis matrices computed by NMF and KNMF are given by

$$\boldsymbol{V}_{NMF}=\begin{pmatrix}0.3402 & 0.1865\\ 0.4411 & 1.4140\\ 0.0000 & 2.0089\\ 0.2842 & 0.4462\\ 1.3386 & 0.0441\end{pmatrix},\quad \boldsymbol{V}_{KNMF}=\begin{pmatrix}0.3435 & 0.0442\\ 0.2616 & 0.7895\\ 0.0000 & 1.0587\\ 0.2274 & 0.2324\\ 1.2439 & 0.0000\end{pmatrix}.$$

Each column vector of $\boldsymbol{V}_{NMF}$ and $\boldsymbol{V}_{KNMF}$ resemble ideal basis vectors since the last entry in the first column vector is much larger than the rest of entries and the second and third entries in the second column vector are much larger than the rest of entries. We compute the sparseness for each column vector in $\boldsymbol{V}_{NMF}$ and $\boldsymbol{V}_{KNMF}$ using the measure in [6] defined by

$$\xi(\boldsymbol{x})=\frac{\sqrt{m}-(\sum|x_i|)/\sqrt{\sum x_i^2}}{\sqrt{m}-1}, \quad (27)$$

where x_i is the ith element of the m-dimensional vector $\boldsymbol{x}$. The sparseness is computed as $(0.4846, 0.4926)$ for $\boldsymbol{V}_{NMF}$ and $(0.5518, 0.5278)$ for $\boldsymbol{V}_{KNMF}$. The difference in sparseness between $\boldsymbol{V}_{NMF}$ and $\boldsymbol{V}_{KNMF}$ is not big. In both cases, basis vectors are close to ideal ones.

We compute encoding matrices by NMF and KNMF, which are given by

$$\boldsymbol{U}_{NMF}=\begin{pmatrix}0.8937 & 0.2211\\ 1.2362 & 0.0989\\ 1.4090 & 0.0900\\ 0.2201 & 0.7642\\ 0.5598 & 0.7165\\ 0.5484 & 0.8232\\ 0.2859 & 0.8125\end{pmatrix},$$

$$\boldsymbol{U}_{KNMF}=\boldsymbol{X}\boldsymbol{W}=\boldsymbol{X}\begin{pmatrix}0.2019 & 0.0000\\ 0.0230 & 0.3730\\ 0.0000 & 0.6144\\ 0.0708 & 0.0126\\ 0.7043 & 0.0000\end{pmatrix}=\begin{pmatrix}1.0733 & 0.5595\\ 1.2277 & 0.3677\\ 1.4543 & 0.3868\\ 0.2248 & 1.3152\\ 0.7411 & 1.4109\\ 0.7049 & 1.5332\\ 0.4064 & 1.5194\end{pmatrix}.$$

Each row vector in encoding matrices $\boldsymbol{U}_{NMF}$ and $\boldsymbol{U}_{KNMF}$ are indicator variables for clusters. For example, the first row in $\boldsymbol{U}_{NMF}$ has two entries 0.8937 and 0.2211 where the first element is bigger than the second element, which implies the associated data (the first row vector of $\boldsymbol{X}$) belongs to cluster 1. Investigating $\boldsymbol{U}_{NMF}$ and $\boldsymbol{U}_{KNMF}$ in this way, one can see that both encoding matrices identify the first three row vectors of $\boldsymbol{X}$ as cluster 1 and the last four row vectors of $\boldsymbol{X}$ as cluster 2.

We provide another example with a data matrix in the case of lower signal-to-noise ratio. We consider the data matrix $\boldsymbol{X}$ given by

$$\boldsymbol{X}=\begin{pmatrix}0.3371 & 0.2630 & 0.9133 & 0.1067 & 0.8998\\ 0.1622 & 0.6541 & 0.1524 & 0.9619 & 0.7599\\ 0.7943 & 0.6892 & 0.8258 & 0.0046 & 1.3001\\ 0.3112 & 1.2482 & 1.0383 & 0.7749 & 0.4314\\ 0.5285 & 0.9505 & 1.4961 & 0.8173 & 0.9106\\ 0.1656 & 0.5838 & 0.5782 & 0.8687 & 0.1818\\ 0.6020 & 0.7290 & 0.9427 & 0.0844 & 0.2638\end{pmatrix}.$$

First three rows were generated by adding uniformly distributed ($\mathcal{U}[0,1]$) random numbers to $[0,0,0,0,0.5]$. Last four rows were constructed by adding uniformly distributed ($\mathcal{U}[0,1]$) random numbers to $[0,0.5,0.5,0,0]$. The data matrix was generated in a way similar to previous example, however, a distinction is lower signal-to-noise ratio (0.5 is used instead of 1) with the same pattern of zeros and nonzeros. Basis matrices computed by NMF and KNMF are given by

$$\boldsymbol{V}_{lsnmf}=\begin{pmatrix}0.7356 & 0.0534\\ 0.6297 & 1.0885\\ 1.1661 & 0.6776\\ 0 & 1.3856\\ 1.1416 & 0.1694\end{pmatrix},\quad \boldsymbol{V}_{knmf}=\begin{pmatrix}0.4694 & 0.1771\\ 0.0000 & 1.0268\\ 0.0020 & 1.1943\\ 0.0000 & 0.7247\\ 1.1252 & 0.0000\end{pmatrix}.$$

We used (27) to compute the sparseness of each column vector of $\boldsymbol{V}_{NMF}$ and $\boldsymbol{V}_{KNMF}$ that is given by $(0.2430, 0.3693)$ and $(0.7495, 0.3593)$. The sparseness difference is larger, compared to previous example. Moreover, basis vectors in KNMF still look close to $[0,0,0,0,0.5]^{\top}$ and $[0,0.5,0.5,0,0]^{\top}$, since the last entry in the first column vector is much larger than the rest of entries and the second and third entries in the second column vector are

much larger than the rest of entries, while the second column vector of $\boldsymbol{V}_{NMF}$ does not.

Corresponding encoding matrices are computed as

$$\boldsymbol{U}_{NMF} = \begin{pmatrix} 0.6901 & 0.0124 \\ 0.1759 & 0.5521 \\ 0.9624 & 0 \\ 0.4251 & 0.6951 \\ 0.7794 & 0.5652 \\ 0.1049 & 0.5817 \\ 0.5485 & 0.1917 \end{pmatrix},$$

$$\boldsymbol{U}_{KNMF} = \boldsymbol{X}\boldsymbol{W} = \boldsymbol{X}\begin{pmatrix} 0.2539 & 0.0408 \\ 0.0001 & 0.3361 \\ 0.0016 & 0.3848 \\ 0.0000 & 0.2366 \\ 0.7445 & 0.0017 \end{pmatrix} = \begin{pmatrix} 0.7569 & 0.4804 \\ 0.6072 & 0.5139 \\ 1.1709 & 0.5852 \\ 0.4019 & 1.0158 \\ 0.8146 & 1.1117 \\ 0.1784 & 0.6313 \\ 0.3508 & 0.6528 \end{pmatrix},$$

where one can see that $\boldsymbol{U}_{KNMF}$ reflects correct clusters (in the first three rows, left elements are bigger than right elements, and in the last four rows, right elements are bigger than left elements), while $\boldsymbol{U}_{NMF}$ fails to identify clusters correctly.

3.3. KNMF for spectral feature extraction

Given a basis matrix $\boldsymbol{V}$ learned by NMF, the feature matrix $\boldsymbol{U}$ is computed by iterating the update (12) only with $\boldsymbol{V}$ fixed. Or LS projection is applied to compute $\boldsymbol{U} = \boldsymbol{X}[\boldsymbol{V}^\top]^\dagger$, while $\boldsymbol{U}$ is not an nonnegative matrix.

On the other hand, the feature matrix is computed as $\boldsymbol{U} = \boldsymbol{X}\boldsymbol{W}$ in KNMF. Thus, features associated with unseen data $\boldsymbol{x}_{test} \in \mathbb{R}^m$ are easily computed by

$$\boldsymbol{u}_{test} = \boldsymbol{x}_{test}^\top \boldsymbol{W}. \tag{28}$$

We can use KNMF for data selection or feature selection. Recall that our data matrix has the form $\boldsymbol{X} = [\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n]^\top$ where $\boldsymbol{x}_i$ is m-dimensional data vector. We define a relative importance of column vectors in $\boldsymbol{X}$ as

$$R_j = \frac{\|\boldsymbol{W}_{j,:}\|}{\|\boldsymbol{W}\|}, \tag{29}$$

where $\boldsymbol{W}_{j,:}$ (for $j = 1, \ldots, m$) is the jth row vector of $\boldsymbol{W}$. For data selection, we sort column vectors of $\boldsymbol{X}$ in descending order by values of R_j and choose column vectors associated with $k \leqslant m$ largest values of R_j. The same idea can be applied to $\boldsymbol{X}^\top$, which is *feature selection*.

4. Numerical experiments

Numerical experiments for EEG classification are conducted, consisting of three steps where pre-processing and classification are common for three different spectral feature extraction methods considered for comparison:

(1) Pre-processing: Transformation of time-domain EEG signals to the time–frequency domain by wavelet transform or short-time Fourier transform.
(2) Spectral feature extraction:
   (a) NMF: The basis matrix $\boldsymbol{V}$ is learned by multiplicative updates (12) and (13) and the feature matrix associated with the test data matrix $\boldsymbol{X}_*$ is computed by $\boldsymbol{U}_* = \boldsymbol{X}_*[\boldsymbol{V}^\top]^\dagger$.
   (b) KNMF: The factor matrix $\boldsymbol{W}$ is learned by multiplicative updates (22) and (23) and the feature matrix associated with the test data matrix $\boldsymbol{X}_*$ is computed by $\boldsymbol{U}_* = \boldsymbol{X}_*\boldsymbol{W}$.
   (c) Feature selection using KNMF, referred to as FS: Apply KNMF to $\boldsymbol{X}^\top = [\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n] \in \mathbb{R}^{m \times n}$ (instead of $\boldsymbol{X}$), estimating the factor matrix $\boldsymbol{W}$. Choose row vectors of $\boldsymbol{X}$ associated with k largest values of relative importance R_j computed by (29). Row vectors of the feature matrix correspond to these selected row vectors of $\boldsymbol{X}$.
(3) Classification: The time-dependent linear discriminant analysis [13,11] or Viterbi algorithm [12], depending on tasks or data types.

Table 1
Classification accuracy of IDIAP data.

Methods	Sub1	Sub2	Sub3	Avg.
NMF	84.93	70.51	55.28	70.24
KNMF	85.16	75.58	58.26	73.00
FS	83.56	76.73	54.13	71.47
BCI comp. winner	79.60	70.31	56.02	68.65

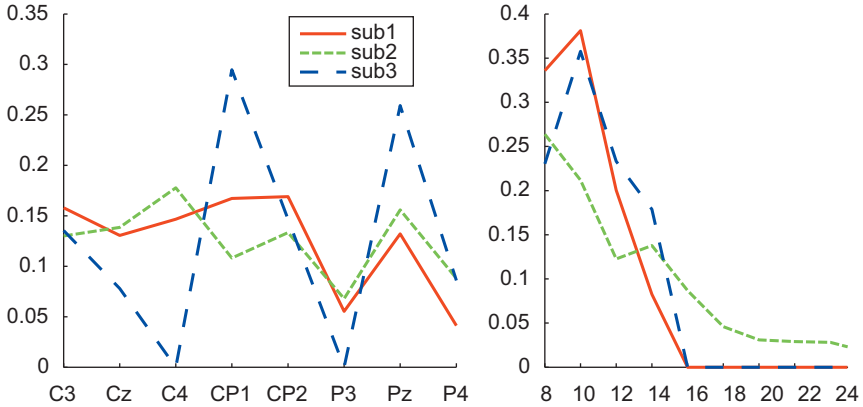

Fig. 4. The pictorial illustration of feature extraction using NMF is shown in the case of Graz dataset (the details on the dataset is explained in Section 4.2). The data matrix $\boldsymbol{X}$ is constructed by collecting frequency profiles (from two channels C_3 and C_4) in its columns. NMF is applied to determine two factor matrices $\boldsymbol{U}$ and $\boldsymbol{V}$ with $r = 4$. Row vectors of $\boldsymbol{V}^\top$ are basis vectors, each of which represents discriminative frequency pattern. The first two rows reveal the characteristics of μ rhythm (8–12 Hz) for C_3 and C_4 channels. The last two rows represent β rhythm around 15 Hz for both channels. Columns of $\boldsymbol{U}$ are corresponding feature profiles.

For our empirical study, we used two datasets: one is the dataset V in BCI competition III, which was provided by the IDIAP Research Institute [7], and the other is the dataset III in BCI competition II, which was provided by the Laboratory of Brain–Computer Interfaces (BCI-Lab), Graz University of Technology [1,13].

4.1. IDIAP dataset

4.1.1. Data description

The IDIAP dataset contains EEG data recorded from three normal subjects and 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. In contrast to the Graz dataset, EEG data are not split into trials, since the subjects are continuously performing any of the mental tasks (i.e., no trial structure).

We use the precomputed features which 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).

4.1.2. Preprocessing

The data matrix $\boldsymbol{X}_{train} \in \mathbb{R}^{10\,528\times 96}$ is constructed by normalizing spectral components $P_{t,f}^{(k)}$ (precomputed features), i.e.,

$$\tilde{P}_{t,f}^{(k)} = \frac{P_{t,f}^{(k)}}{\sum_f P_{t,f}^{(k)}}, \tag{30}$$

for $f \in \{8, 10, \ldots, 28, 30\}$ Hz, $k = 1, 2, \ldots, 8$ (corresponding to eight different channels, including C_3, C_z, C_4, CP_1, CP_2, P_3, P_z, and P_4), $t = 1, \ldots, 10\,528$ where 10 528 is the number of data points in the training set (note that there is no trial structure in this dataset). In the same way, we make the test data matrix, $\boldsymbol{X}_{test} \in \mathbb{R}^{3504\times 96}$.

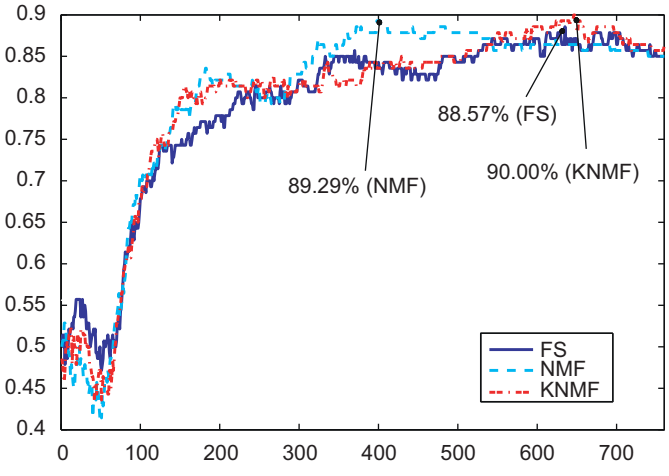

Fig. 5. Eight basis vectors computed by NMF (blue solid line) and by KNMF (red dotted line) are shown using Graz dataset. Basis vectors are sorted according to the mutual information between them of NMF and of KNMF. KNMF produces more sparse factors. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

4.1.3. Feature extraction and classification

Feature matrices $\boldsymbol{U} \in \mathbb{R}^{n\times r}$ computed by three different methods, NMF, KNMF, and FS are used as inputs to Viterbi algorithm [5] for on-line classification of un-cued EEG signals. More details on the Viterbi classifier are found in [12].

Classification accuracy is summarized in Table 1. KNMF outperforms NMF across all subjects and shows the best performance across most of cases. FS method shows compatible performance to NMF, while FS provides easily-interpretable features. Fig. 6 shows the relative importance of spatial (related to channel) and spectral (related to frequency components) features for each subject. Each subject has his/her own spatial and spectral features.

4.2. Graz dataset

4.2.1. Data description

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 s, where a visual cue (arrow) is presented pointing to the left or the right after 3-s 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 even-related desynchronization has contralateral dominance and C_z channel contains little information for discriminant analysis. Requirements for result comparison are to provide a continuous classification accuracy for each time point of trial during imagination session.

4.2.2. Preprocessing

By filtering it with complex Morlet wavelets [11], we obtain the time–frequency representation of the EEG data, $P_{i,t,f}^{(k)}$ for $f \in \{4, 5, \ldots, 30\}$ Hz, $k = 1, 2$ (C_3 and C_4 channels), $t = 1, \ldots, T$ and $i = 1, \ldots, N$. T is the number of data points in each trial and N is the number of trials. The data matrix is reshaped by $\boldsymbol{X}_{train} = \boldsymbol{X}_{test} \in \mathbb{R}^{(27*2)\times(2*760*140)}$, where $T = 760$ and $N = 140$.

4.2.3. Feature extraction and classification

The procedure of feature extraction is same as above. For the single-trial online classification for Graz data (with trial structure), we use a Gaussian probabilistic model-based classifier [13] 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. The classification results are shown in Fig. 1. The basis vectors of NMF and KNMF are shown in left and middle figures of Fig. 2, respectively. The averaged sparseness of KNMF is 0.7104, while the averaged sparseness of NMF is 0.3227. Thus, KNMF factors are sparser than NMF factors. Right figure of Fig. 2 shows the selected features. Only using 10-selected features, we can get the similar classification result in Fig. 1. Fig. 7 shows the 100-selected data from 106 400 data points (only 7000 randomly selected data points are plotted in figure). For visualization, we reduce the dimension from 54 to 2 using KNMF. Its basis vectors are shown in left figure of Fig. 7. Although the data are selected in unsupervised framework, they are representative and discriminative data of each class.

5. Conclusions

We have presented multiplicative updates for KNMF that is a kernelized version of the standard NMF. Assuming $\boldsymbol{U} = \boldsymbol{XW}$, KNMF was easily derived in the framework of LS optimization

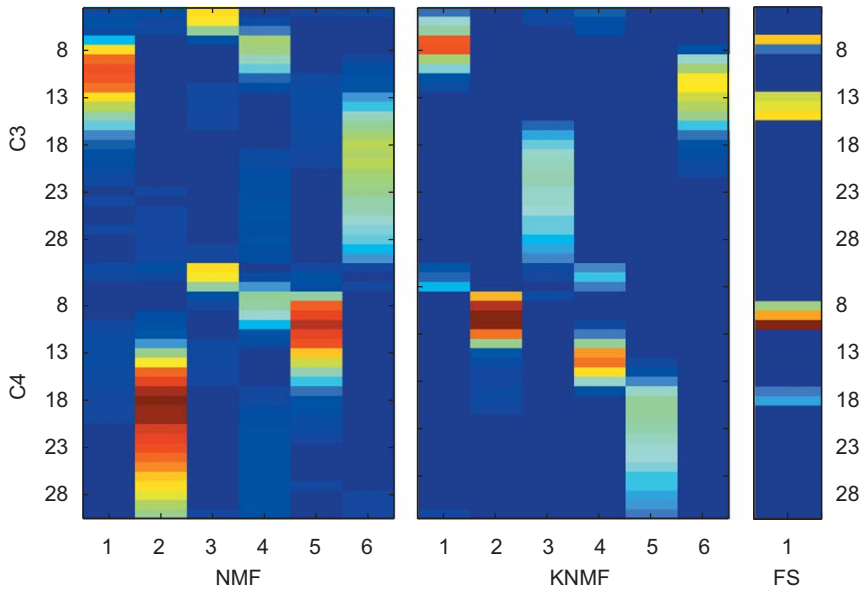

Fig. 6. The relative importance of spatial and spectral features is shown for three different subjects, in the left (channels) and right (frequency components) panel, respectively. That is, the left panel shows which channels are more important, while the right panel shows which frequency components are more important in a given task.

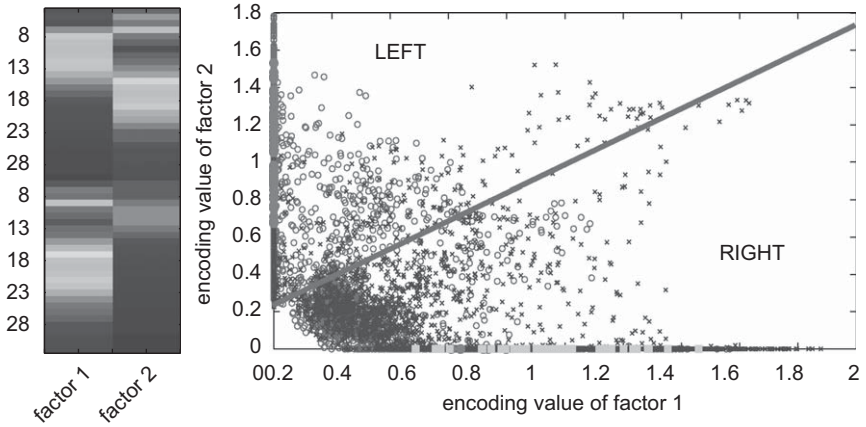

Fig. 7. The left panel shows two basis vectors determined by KNMF with $r=2$. Each of them reveals μ rhythm in C_3 C_4 channels, respectively. The right panel shows associated encoding variables (linear low-dimensional embeddings into two-dimensional Euclidean space), where red circles represent "LEFT" class and blue crosses represent "RIGHT" class. The green solid line is a hyperplane which bisects the line connecting two centers determined by k-means clustering, with considering the scatter of each cluster. Among 10 000 data points, we selected 100 data points by our data selection scheme. Those selected data points were clustered along either horizontal axis (cyan) or vertical axis (magenta), implying that selected data points are representative and discriminative.(For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

with nonnegativity constraints. We have successfully applied KNMF to a task of learning discriminative spectral feature from EEG data for classification. Experiments on two benchmark EEG datasets confirmed the performance gain over standard NMF. Although KNMF was first proposed in [4], we further elaborated it here and confirmed its useful behavior (easiness in feature extraction and sparsity) with real-world EEG data.

Acknowledgments

This work was supported by KOSEF Basic Research Program (Grant R01-2006-000-11142-0) and National Core Research Center for Systems Bio-Dynamics.

References

[1] 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, N. Birbaumer, The BCI competition 2003: progress and perspectives in detection and discrimination of EEG single trials, IEEE Transactions on Biomedical Engineering 51 (6) (2004).

[2] S. Choi, Algorithms for orthogonal nonnegative matrix factorization, in: Proceedings of the International Joint Conference on Neural Networks, Hong Kong, 2008.

[3] A. Cichocki, H. Lee, Y.-D. Kim, S. Choi, Nonnegative matrix factorization with α-divergence, Pattern Recognition Letters 29 (9) (2008) 1433–1440.

[4] C. Ding, T. Li, M.I. Jordan, Convex and semi-nonnegative matrix factorizations, Technical Report 60428, Lawrence Berkeley National Laboratory, 2006.

[5] G.D. Forney, The Viterbi algorithm, Proceedings of the IEEE 61 (1973) 268–278.

[6] P.O. Hoyer, Non-negative matrix factorization with sparseness constraints, Journal of Machine Learning Research 5 (2004) 1457–1469.

[7] J. del R. Millán, On the need for on-line learning in brain–computer interfaces, in: Proceedings of the International Joint Conference on Neural Networks, Budapest, Hungary, 2004.

[8] T.N. Lal, M. Schroder, T. Hinterberger, J. Weston, M. Bogdan, N. Birbaumer, B. Schölkopf, Support vector channel selection in BCI, Technical Report 120, Max Planck Institute for Biological Cybernetics, 2003.

[9] D.D. Lee, H.S. Seung, Learning the parts of objects by non-negative matrix factorization, Nature 401 (1999) 788–791.

[10] D.D. Lee, H.S. Seung, Algorithms for non-negative matrix factorization, in: Advances in Neural Information Processing Systems, 13, MIT Press, Cambridge, 2001.
[11] H. Lee, A. Cichocki, S. Choi, Nonnegative matrix factorization for motor imagery EEG classification, in: Proceedings of the International Conference on Artificial Neural Networks, Athens, Greece, Springer, Berlin, 2006.
[12] H. Lee, Y.-D. Kim, A. Cichocki, S. Choi, Nonnegative tensor factorization for continuous EEG classification, International Journal of Neural Systems 17 (4) (2007) 305–317.
[13] S. Lemm, C. Schäfer, G. Curio, BCI competition 2003-data set III: probabilistic modeling of sensorimotor μ rhythms for classification of imaginary hand movements, IEEE Transactions on Biomedical Engineering 51 (6) (2004).
[14] S.Z. Li, X.W. Hou, H.J. Zhang, Q.S. Cheng, Learning spatially localized parts-based representation, in: Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition, Kauai, Hawaii, 2001, pp. 207–212.
[15] T. Li, C. Ding, The relationships among various nonnegative matrix factorization methods for clustering, in: Proceedings of the IEEE International Conference on Data Mining, Hong Kong, 2006.
[16] P. Paatero, U. Tapper, Positive matrix factorization: a non-negative factor model with optimal utilization of error estimates of data values, Environmetrics 5 (1994) 111–126.
[17] J.R. Wolpaw, N. Birbaumer, D.J. McFarland, G. Pfurtscheller, T.M. Vaughan, Brain-computer interfaces for communication and control, Clinical Neurophysiology 113 (2002) 767–791.

Hyekyoung Lee received the B.S. degree in electrical engineering in 2000, and the M.S. and Ph.D. degrees in computer science in 2003 and 2009, from Pohang University of Science and Technology, Pohang, Korea. Her research interests include statistical machine learning and EEG-based brain computer interface.

Andrzej Cichocki received the M.Sc. (with honors), Ph.D. and Dr.Sc. (Habilitation) degrees, all in electrical engineering from Warsaw University of Technology (Poland). Since 1972, he has been with the Institute of Theory of Electrical Engineering, Measurement and Information Systems, Faculty of Electrical Engineering at the Warsaw University of Technology, where he obtained a title of a full Professor in 1995. He spent several years at University Erlangen-Nuerenberg (Germany), at the Chair of Applied and Theoretical Electrical Engineering directed by Professor Rolf Unbehauen, as an Alexander-von-Humboldt Research Fellow and Guest Professor. In 1995–1997 he was a team leader of the laboratory for Artificial Brain Systems, at Frontier Research Program RIKEN (Japan), in the Brain Information Processing Group. He is currently the head of the laboratory for Advanced Brain Signal Processing, at RIKEN Brain Science Institute (JAPAN) in the Brain-Style Computing Group directed by Professor Shun-ichi Amari. He is co-author of more than 250 scientific papers and three internationally recognized monographs (two of them translated to Chinese): Adaptive Blind Signal and Image Processing (Wiley, April 2003-revised edition), CMOS Switched-Capacitor and Continuous-Time Integrated Circuits and Systems (Springer, 1989) and Neural Networks for Optimizations and Signal Processing (Teubner-Wiley, 1994). He is Editor in Chief of International Journal Computational Intelligence and Neuroscience.

Seungjin Choi received the B.S. and M.S. degrees in Electrical Engineering from Seoul National University, Korea, in 1987 and 1989, respectively, and the Ph.D. degree in Electrical Engineering from the University of Notre Dame, Indiana, in 1996. He was a Visiting Assistant Professor in the Department of Electrical Engineering at University of Notre Dame, Indiana, during the Fall semester of 1996. He was with the Laboratory for Artificial Brain Systems, RIKEN, Japan, in 1997 and was an Assistant Professor in the School of Electrical and Electronics Engineering, Chungbuk National University from 1997 to 2000. He is currently a Professor of Computer Science at Pohang University of Science and Technology, Korea. His primary recent research has focused on statistical machine learning and probabilistic models, including manifold learning, matrix factorization, semi-supervised learning, kernel machines, and independent component analysis, with applications to brain computer interface, bioinformatics, information retrieval and pattern recognition.