Proceedings of the 2005 IEEE
Engineering in Medicine and Biology 27th Annual Conference
Shanghai, China, September 1-4, 2005

Enhancing Feature Extraction with Sparse Component Analysis for Brain-Computer Interface

Yuanqing Li, Cuntai Guan and Jianzhao Qin

Abstract— **Feature extraction is very important to EEG-based brain computer interfaces (BCI) in helping achieve high classification accuracy. Preprocessing of EEG signals plays an important role, because an effective preprocessing method will help enhance the efficiency of the feature extraction. In this paper, sparse component analysis (SCA) is employed as a preprocessing method for EEG based BCI. A combined feature vector is constructed. This feature vector consists of a dynamical power feature and a dynamical common spatial pattern (CSP) feature. The dynamical power feature is extracted from selected SCA components, while the dynamical CSP feature is extracted from raw EEG data. Using the presented preprocessing and feature extraction method, we analyze the data for a cursor control BCI carried out at Wadsworth Center. Our results show that SCA preprocessing is the most effective in extracting a component which reflects the subject's intention, and demonstrate the validity of SCA preprocessing for the enhancement of feature extraction.**

I. INTRODUCTION

Brain-computer Interface (BCI) provides an alternative communications and control method for those people with severe motor disabilities. The core component in a BCI system is an effective translation mechanism which converts brain activities into commands. Particularly, as a non-invasive BCI, EEG-based BCI measures specific EEG activities, extracts features and translates these features into command signals to control cursor movement or devices (for instance, robotic arm, wheelchair, etc.). Until now, researchers have developed many effective approaches for EEG-based BCI using different features, for example, event-related potential (ERP), slow cortex potential (SCP), movement-related potential (MRP), event-related (de-)synchronization (ERD/ERS), common spatial pattern (CSP), etc.

Preprocessing of EEG signals plays an important role in EEG-based BCI. A good preprocessing method can improve the performance of BCI (e.g., accuracy rate and speed, or information transferring rate (ITR)). Until now, ones have developed or used several preprocessing methods in EEG-based BCI including Spatial filtering, temporal filtering, Principle Component Analysis (PCA), Independent Component Analysis (ICA), etc. Spatial filtering method is a common preprocessing method, in which alternative spatial filters include a standard ear-reference, a common average reference (CAR), a small Laplacian reference (3 cm to set of surrounding electrodes), a large Laplacian reference (6 cm to set of surrounding electrodes), a local average technique [1], [2]. Many methods, e.g, low pass filter, time averaging, downsampling, baseline correction, etc, belong to temporal filtering methods [3]. PCA is a kind of de-correlation method, which can be used as a signal preprocessing method in many fields including BCI [14]. ICA is often used to find the independent components in EEG [4], [5], which is also an alternative preprocessing method in BCI [6].

In this paper, we introduce sparse component analysis (SCA) or sparse factorization as a preprocessing method for EEG-based BCI. EEG data matrix can be factorized into the product of two matrices by SCA, one of which is the mixing matrix, and the other is the source component matrix[9], [10]. Based on the source component obtained from the spare factorization, we define a dynamical feature vector consisting of power and CSP features. In the extraction of these dynamical features, time-bin selection is introduced. The power feature is extracted from SCA components, and the CSP feature is extracted from raw EEG data. We give offline analysis on a cursor control BCI data set, which was provided by Wadsworth Center for BCI competition 2003. We demonstrate the validity of SCA preprocessing and feature extraction method by comparing with several other methods in the literature.

The remainder of this paper is organized as follows. In section 2, we present the SCA preprocessing method for EEG-based BCI, and the feature extraction. In section 3, we present off-line analysis results for the cursor control dataset mentioned above. Discussions and conclusions are given in Section 4.

II. METHODS

The method presented in this section consists three parts, preprocessing, feature extraction and classification. The processing method is SCA, the feature vectors consists of dynamical power and CSP features, and classification method is a standard SVM.

A. Data description

An EEG-based cursor control experiment was carried out in Wadsworth Center. The recorded data set was available from BCI Competition 2003, which can be downloaded from the web site: http://ida.first.fraunhofer.de/projects/bci/competition. The details of the experiment can be found in the web. We give a brief description as follws. During the experiment, the subjects sat in a reclining chair facing a video screen. They

Yuanqing Li is with Institute for Infocomm Research, Singapore 119613 yqli2@i2r.a-star.edu.sg
Cuntai Guan is with Institute for Infocomm Research, Singapore 119613 ctguan@i2r.a-star.edu.sg
Jianzhao Qin is with Institute of Automation Science and Engineering, Southchina University of Technology, Guangzhou, 510640,China

used mu or beta rhythm amplitude (i.e., frequencies between 8-12 Hz or 18-24 Hz, respectively) to control vertical cursor movement toward the vertical position of targets located at the right edge of the video screen. There were four target positions. The Data were collected from each subject for 10 sessions of 30 min each. Each session consisted of six runs, separated by one-minute breaks, and each run consisted of about 32 individual trials. Thus the total trial number was 192 in each session.

B. SCA preprocessing

First, we assume that EEG signals can be modelled by the following linear model neglecting additive noise.

$$\mathbf{X} = \mathbf{AS}, \tag{1}$$

where $\mathbf{X} = [\mathbf{x}(1), \cdots, \mathbf{x}(N)] = [x_i(j)]_{n\times N}$ is a known EEG data matrix of which each row is an EEG channel signal, N is the number of samples, $\mathbf{A} = [\mathbf{a}_1 \cdots \mathbf{a}_m]$ is a $n \times m$ mixing constant matrix, $\mathbf{S} = [\mathbf{s}(1), \cdots, \mathbf{s}(N)] = [s_i(j)]_{m\times N}$ is a source components matrix, of which the rows represent brain sources, artifacts, etc. If n is not too large, we assume that $m > n$, which implies that the model is overcomplete.

In this paper, a SCA approach is used to find the mixing matrix and source components. This approach contains two parts: the first estimates the mixing matrix, the second estimates source components. As our analysis shown in [10], sparsity of source components plays a key role in this approach. Although EEG signals and their source components are not sparse in the time domain, they are sparse in the time frequency domain. The following discussion mainly deals with the time frequency domain. To obtain sparser time frequency representation, we apply wavelet packets transformation to (1) instead of a general wavelet transformation. Noting that the wavelet packets transformation is a linear transformation, we can obtain

$$\tilde{\mathbf{X}} = \mathbf{A}\tilde{\mathbf{S}}, \tag{2}$$

where each row of $\tilde{\mathbf{X}}$ is a time-frequency representation of a corresponding EEG channel signal in $\mathbf{X}$, each row of $\tilde{\mathbf{S}}$ is a time-frequency representation of corresponding source components in $\mathbf{S}$.

Algorithm outline 1: Estimating the mixing matrix

Step 1. De-noise by deleting the columns in $\tilde{\mathbf{X}}$ with their 2-norm less than a positive constant e.g. $\frac{M}{3}$, where M is the maximum of 2-norms of all the columns, a submatrix of $\tilde{\mathbf{X}}$ is obtained denoted as $\bar{\mathbf{X}}$;

Step 2. Normalize the data column vectors of $\bar{\mathbf{X}}$ to length one;

Step 3. Set a source number m and begin a $K-$means clustering iteration followed by normalization to estimate the mixing matrix denoted as $\bar{\mathbf{A}}$.

Since the estimated mixing matrix $\bar{\mathbf{A}}$ is not a square matrix with its column number being larger than its row number, the source component matrix can not be obtained by solving linear equations. In this paper, we first estimate the time frequency representation matrix $\tilde{\mathbf{S}}$ in (2) by solving a optimization problem, and then calculate the source component matrix $\mathbf{S}$ using an inverse wavelet packets transformation. That is, for the estimated mixing matrix $\bar{\mathbf{A}}$ in (1), the time frequency representation matrix $\tilde{\mathbf{S}}$ of the source component matrix can be found by solving the following optimization problem:

$$\min \sum_{i=1}^{m}\sum_{j=1}^{N} |\tilde{s}_i(j)|, \text{ subject to } \bar{\mathbf{A}}\tilde{\mathbf{S}} = \tilde{\mathbf{X}}. \tag{3}$$

Thus we have

Algorithm outline 2:

Step 1. Transform the n time course signals (n rows of $\mathbf{X}$) into n time-frequency signals by a wavelet packets transformation, a time frequency matrix $\tilde{\mathbf{X}}$ is obtained.

Step 2. Estimate the mixing matrix $\bar{\mathbf{A}} \in R^{n\times m'}$ using the Algorithm 1.

Step 3. Use the estimated mixing matrix $\bar{\mathbf{A}}$ and the matrix $\tilde{\mathbf{X}}$, estimate the time frequency representation matrix $\tilde{\mathbf{S}}$ in (2) by solving the optimization problem (3).

Step 4. Reconstruct source component matrix $\mathbf{S}$ in (1) using the inverse wavelet packets transformation. End.

C. Power feature extraction

Before extracting features, we define ten time bins for each trial which are partially overlapped. Each time bin contains 120 samples, and every two neighboring bins are overlapped by 92 samples. Hence the total number of the used samples is 372. In this section and section 2.4, we extract power and CSP features based on these ten time bins.

Through SCA preprocessing, we obtain m components ($m = 15$ in this paper). For every components, we extract two power features in each time bin as follows, one is from μ rhythm, the other is from β rhythm,

$$PF_i^{\mu} = \sum_{f\in[11,14]} P_i(f), \; PF_i^{\beta} = \sum_{f\in[22,26]} P_i(f), \tag{4}$$

where $P_i(f)$ is the power spectral of the component in the $i-th$ time bin, $i = 1, \cdots, 10$.

In this paper, we also implement ICA and PCA methods to compare with SCA for preprocessing. The power feature extraction of ICA and PCA components is similar to what is described above. However, the number of ICA components is equal to n ($n = 9$ in this paper), so is the number of PCA components.

Until now, we have extracted two power feature vectors based on μ sand β rhythm, respectively. Their entries are defined in 10 time bins. In the following, we perform a selection of time bins.

We only use the features of 5 time bins and construct the power feature vectors.

$$\mathbf{PF}^{\mu}(i_1, \cdots, i_5) = [PF_{i_1}^{\mu}, \cdots, PF_{i_5}^{\mu}], \tag{5}$$

$$\mathbf{PF}^{\beta}(i_1, \cdots, i_5) = [PF_{i_1}^{\beta}, \cdots, PF_{i_5}^{\beta}]. \tag{6}$$

We now perform time bin selection using nearest neighborhood classifier with Euclidean distance (described in the next subsection) based on the data of training sessions 1-6. Noting that we have total C_1^50 combinations of five time bins when we choose 5 from 10 ten time bins. We chose the combination of five time bins which corresponding to the highest classification rate. Denote the indices of the five selected time bins as $I_1^\mu, \cdots, I_5^\mu$ corresponding to the power of μ band. Similarly, considering β rhythm based power feature, we can also choose five time bins with their indices denoted as $I_1^\beta, \cdots, I_5^\beta$. The power feature vectors with time bin selection $\mathbf{PF}^\mu(I_1^\mu, \cdots, I_5^\mu)$ and $\mathbf{PF}^\beta(I_1^\beta, \cdots, I_5^\beta)$ will be used in our combined features.

D. CSP feature extraction

Since using power feature alone cannot give us satisfying result in offline analysis, we combine it with CSP feature.

First, we extract CSP feature based on μ rhythm. We perform a spatial filtering with common average reference to raw EEG signals of 64 channels, then apply temporal filtering to get μ frequency band ($11 - 14$Hz) signal. The following CSP feature extraction is based on the filtered signals.

For each trial, we have defined 10 overlapped time bins in the previous subsection. For each time bin, we calculate a CSP feature vector as follows.

The CSP analysis in the ith time bin consists of calculating a matrix $\mathbf{W}_i$ and diagonal matrix $\mathbf{D}_i$ through a joint diagonalization method ($i = 1, \cdots, 10$):

$$\mathbf{W}_i \Sigma_i^1 \mathbf{W}_i^T = \mathbf{D}_i, \quad \mathbf{W}_i \Sigma_i^4 \mathbf{W}_i^T = 1 - \mathbf{D}_i, \tag{7}$$

where Σ_i^1 and Σ_i^4 are 64 by 64 dimensional normalized covariance matrix derived from EEG data matrices $\mathbf{E}_i^1$ and $\mathbf{E}_i^4$. Using all trials with target code 1 of the training sessions (sessions 1-6), we construct the matrix $\mathbf{E}_i^1$ by trial-concatenating the filtered EEG data in the ith time bins of every trial. $\mathbf{E}_i^4$ is obtained similarly except that it corresponds to the trials with target code 4. The reason why we use data from the trials with target code 1 and 4 is that these two targets are at the top and bottom position of the screen, so they are most separable.

After obtaining the transformation matrix $\mathbf{W}_i$, we now extract CSP feature in the ith time bin of a trial ($i = 1, \cdots, 10$). We first calculate a covariance matrix using the filtered EEG signals in the ith time bin, then normalize it. We take the first 2 and the last 2 main diagonal elements of the transformed (by $\mathbf{W}_i$) and normalized covariance matrix. Note that the first 2 diagonal elements correspond to 2 largest eigenvalues in the diagonal matrix $\mathbf{D}_i$ above, the other 2 correspond to its 2 smallest eigenvalues. Thus we obtain a 4 dimensional CSP feature for each time bin.

Until now, we have described CSP feature based on μ rhythm in each time bin. The CSP feature in the ith bin is denoted as $\mathbf{CF}^\mu(i) \in R^4$, where $i = 1, \cdots, 10$.

We also extract CSP feature based on β rhythm ($22 - 26$Hz) In the similar way as above. The extracted CSP feature in the ith time bin is denoted as $\mathbf{CF}_i^\beta$.

Similarly as in the previous section, we perform two selection of 5 time bins for μ rhythm based and β rhythm based CSP features, respectively. The indices of selected 5 time bins are denoted as $J_1^\mu, \cdots, J_5^\mu$ for μ rhythm based CSP feature, the corresponding CSP feature vector is denoted as $\mathbf{CF}^\mu(J_1^\mu, \cdots, J_5^\mu)$. For β rhythm based CSP feature, the indices of selected 5 time bins are denoted as $J_1^\beta, \cdots, J_5^\beta$, the corresponding CSP feature vector is denoted as $\mathbf{CF}^\beta(J_1^\beta, \cdots, J_5^\beta)$.

E. Construction of feature vector

In this subsection, we combine power feature and CSP feature to construct the final feature vector for each trial.

As stated in Subsection 2.3, we extracted power feature for every SCA components. However, we use only one component's power feature in the combined feature vector. The selection of component is carried out by using an nearest neighborhood classifier with Euclidean distance to perform cross validation to the data of the training sessions 1-6. The component with the highest accuracy is chosen.

In the combination of power and CSP features, we only use the features of 5 time bins (the method for time bin selection stated in the two previous subsections). First, we combine power and CSP features from μ and β bands as follows, respectively,

$$\mathbf{F}^\mu = [\mathbf{PF}^\mu(I_1^\mu, \cdots, I_5^\mu), \mathbf{CF}^\mu(J_1^\mu, \cdots, J_5^\mu)], \tag{8}$$

$$\mathbf{F}^\beta = [\mathbf{PF}^\beta(I_1^\beta, \cdots, I_5^\beta), \mathbf{CF}^\beta(J_1^\beta, \cdots, J_5^\beta)]. \tag{9}$$

Then, we combine the above two feature vectors and obtain,

$$\mathbf{F}^{\mu,\beta} = [\mathbf{F}^\mu, \mathbf{F}^\beta]. \tag{10}$$

Again, we perform the selection of the three types of feature vectors above. The selection method is similar to that of time bin selection, which is also carried out using nearest neighborhood classifier with Euclidean distance.

In this paper, two classifiers, nearest neighborhood classifier with Euclidean distance and support vector machine (SVM) classifier with RBF kernel are used. The first classifier is mainly used for three feature selections, that is, the selection of components, the selection of time bins and the selection of the three types of feature vectors (defined in the previous subsection). A combined feature vector for each subject is obtained. After constructing the feature vector, we use SVM for classification to obtain the final results.

III. RESULTS

In this section, we present our offline analysis results for three subjects (”A”, ”B” and ”C”) in the Wadsworth Center dataset. Before applying SCA preprocessing, we constructed a data matrix $\mathbf{X}$ using EEG data from 9 selected EEG channels with channel umbers $\{8, 9, 10, 15, 16, 17, 48, 49, 50\}$. The locations of the 9 electrodes covers left sensorimotor area. After setting the number of source components 15, we estimated the mixing matrix and 15 components using Algorithms 1 and 2.

We then constructed the combined feature vectors and use SVM for classification. As explained in Subsection 2.5, we selected the best SCA component for power feature extraction. For three subjects, the selected components are: component 8 for Subject A, component 5 for Subject B, component 8 for Subject C.

Subsequently, we extracted CSP feature using the method in Subsection 2.4. Using dynamical power and CSP features, we constructed the feature vectors as in Subsection 2.5. After we obtained the combined feature vectors for all trials in training sessions and test sessions (session 7 to 10), we then trained a SVM using training set, and performed classification for the test sessions. Comparing with the target codes of the 4 test sessions, we calculated the average classification rates (averaged across the 4 test sessions).

In comparison with the SCA preprocessing, we also evaluated prevailing ICA and PCA methods by plugging them in to replace SCA preprocessing. The average classification rates for these two methods were obtained by using the combined feature vectors and SVM classifier. The three average classification rates for Subject "A" are shown in the second row of Table 1, while the offline analysis results for the data sets of subjects "B" and "C" are shown in the last two rows of Table 1. The online results are also listed in the last column of this table for comparison.

TABLE I

CLASSIFICATION RATES (%) WITH DIFFERENT PREPROCESSING METHODS AVERAGED ACROSS TESTING SESSIONS 7-10

Subjects	SCA	ICA	PCA	Online result
A	73.04	70.81	69.90	73.4
B	64.01	65.71	65.84	77.2
C	72.44	68.85	67.18	69.00

IV. CONCLUSION

From Table 1, we can see that for subjects "A" and "C", the average classification rates obtained by SCA preprocessing are the highest, which are very close to or better than the online results. For Subject "B", the results obtained by three preprocessing methods are much lower than the online result. This could be due to the reason that subject "B" varied his brain wave during training sessions more than other subjects. The online model could be well trained with more data or better adapted with online adaptation. In fact, all reported offline analysis results in the literature are much poorer than the online result [14]. Because of the nonstationarity of the EEG data for the subject, the offline analysis method without adaptation does not work well for the subject.

It is well known that ICA is based on the assumption of independent sources. In recent years, ICA has been widely used in EEG signals, for example, to reveal source components, to remove artifacts, to preprocess for event-related potential based BCI, etc. Many promising results were obtained from it. However, it is reasonable to assume that not all EEG sources are mutually independent. Similarly, it is also reasonable to assume that not all brain source components are mutually uncorrelated, which is the basis of PCA preprocessing. Compared to ICA and PCA, SCA has the following important advantages: 1) sources are not assumed to be mutually independent or uncorrelated; actually, sources can even be non-stationary; 2) the number of sources can be greater than the number of sensors. We believe that SCA is an alternative and very promising approach for analyzing EEG, especially as a preprocessing method for EEG-based BCI systems.

V. ACKNOWLEDGMENTS

The work of the third author was supported by the National Natural Science Foundation of China (No. 60475004), Guangdong Province Science Foundation for Research Team Program (No.04205789), China

REFERENCES

[1] D. J. McFarland, L. M. McCane, S. V. David, and J. R. Wolpaw, Spatial filter selection for EEG-based communication. Electroenceph. clin. Neurophysiol., 1997b, 103: 386-394.

[2] B. O. Peters, G. Pfurtscheller, and H. Flyvbjerg, "Automatic differentiation of multichannel EEG signals," *IEEE Trans on Biomedical Engineering*, vol. 48, no. 1, pp. 111-116, 2001.

[3] B. Blankertz, G. Curio, and K. R. Muller, "Classifying single trial EEG: towards brain-computer interfacing," in: T. G. Diettrich, S. Becker, and Z. Ghahramani, eds., *Advances in Neural Inf. Proc. Systems (NIPS 01)*, vol. 14, 2002.

[4] S. Makeig, M. Westerfield, T. P. Jung, S. Enghoff, J. Townsend, E. Courchesne & T. J. Sejnowski, " Dynamic brain sources of visual evoked responses," *Science*, vol. 295, pp. 690-694, 2002.

[5] T. P. Jung, S. Makeig, M. Westerfield, J. Townsend, E. Courchesne, & T. J. Sejnowski, "Removal of eye activity artifacts from visual event-related potentials in normal and clinical subjects," *Clinical Neurophysiology*, vol. 111, pp.1745-1758, 2000.

[6] B. J. Culpepper, R. M. Keller, "Enabling Computer Decisions based on EEG input," *IEEE Trans. on Neural Systems and Rehabilitation Engineering*, vol. 11, no. 4, pp. 354-360, 2003.11.

[7] S. Vorobyov and A. Cichocki, "Blind noise reduction for multisensory signals using ICA and subspace filtering, with application to EEG analysis," Biological Cybernetics, vol. 86, no. 4, pp. 293-303, Apr. 2002.

[8] S. A. Crucez-Alvarez, A. Cichocki, and S. Amari, "From blind signal extraction to blind instantaneous signal separation: criteria, algorithms and stability," IEEE Transactions on Neural Networks, Special Issue on Information Theoretical Learning, vol. 15, no. 4, pp. 859-873, 2004.

[9] M. Zibulevsky, & B. A. Pearlmutter, "Blind Source Separation by Sparse Decomposition," *Neural Computations*, vol. 13(4), pp.863-882, 2001.

[10] Y. Q. Li, A. Cichocki and S. Amari, "Sparse representation and blind source separation," *Neural Computation*, vol. 16, no. 6, 2004.

[11] K. R. Muller, S. Mika, G. Ratsch, K. Tsuda, and B. Scholkopf, "An Introduction to Kernel bsed learning algorithm," *IEEE Trans. on Neural Networks*, vol. 12, no. 2, pp. 181-201, 2001.

[12] J. R. Wolpaw, N. Birbaumer, D. J. McFarland, G. Pfurtscheller, T. M. Vaughan, "Brain-computer interfaces for communication and control," Clinical Neurophysiology, vol. 113, pp. 767-791, 2002.

[13] D. J. McFarland, A. T. Lefkowicz, and J. R. Wolpaw, Design and "operation of an EEG-based brain-computer interface with digital signal processing technology," *Behav. Res. Methods Instrum. Comput.*, 1997a, 29: 337-345.

[14] Cheng M., Jia W. Y., Gao X. R., Gao S. K. and Yang F. S., "Mu-rhythm-based cursor control: an offline analysis," *Clinical Neurophysiology*, vol. 115, pp. 745-751, 2004.