PCA+HMM+SVM FOR EEG PATTERN CLASSIFICATION

Hyekyung Lee and Seungjin Choi

Department of Computer Science and Engineering, POSTECH, Korea
{leehk, seungjin}@postech.ac.kr

ABSTRACT

Electroencephalogram (EEG) pattern classification plays an important role in the domain of brain computer interface (BCI). Hidden Markov model (HMM) might be a useful tool in EEG pattern classification since EEG data is a multivariate time series data which contains noise and artifacts. In this paper we present methods for EEG pattern classification which jointly employ principal component analysis (PCA) and HMM. Along this line, two methods are introduced: (1) PCA+HMM; (2) PCA+HMM+SVM. Usefulness of principal component features and our hybrid method is confirmed through the classification of EEG that is recorded during the imagination of a left or right hand movement.

1. INTRODUCTION

The automatic classification of EEG patterns plays an important role in an EEG-based BCI system. It provides a new communication channel between human brain and computer. In general, EEG data is very noisy and contains several types of artifacts. Moreover, EEG data consists of mixtures of several brain sources (which are invisible to us) and noisy sources, which makes the problem even harder.

Several attempts have been made to build an EEG-based BCI system. The system consists of two procedure: (1) feature extraction; (2) classification. For feature extraction, adaptive autoregressive model (AAR), Hjorth parameters, power spectrum have widely been used. As a classifier, linear discriminant analysis (LDA), neural networks, and recently HMM were used [6].

In this paper, we consider principal component features which capture the second-order statistical structure of the data. Although PCA has been mainly used to analyze spatial data, however, in this paper we show that PCA is also a useful tool for time series data. Since PCA retains maximum variance, it is expected to provide features that are robust to small noise.

Based on principal component features, we employ a HMM classifier that is a popular tool for modelling time series data. A recent work on a HMM-based BCI system can be found in [5] where Hjorth parameters were used. In this paper we show that principal component features improves the classification performance of a HMM (PCA+HMM). In addition we also present a hybrid method which combines HMM and support vector machine (SVM). These two methods are described in Sec. 3 and their usefulness is confirmed by computer simulations.

2. BACKGROUND

2.1. PCA

PCA aims to find a linear orthogonal transformation $\boldsymbol{v} = \boldsymbol{W}\boldsymbol{u}$ (where $\boldsymbol{u}$ is the observation vector) such that the retained variance is maximized. Alternatively, PCA is viewed as a minimizer of reconstruction error It turned out that these principles (variance maximizer or reconstruction error minimizer) leads to a symmetric eigenvalue problem. The row vectors of $\boldsymbol{W}$ correspond to the normalized orthogonal eigenvectors of the data covariance matrix.

A simple approach to PCA is to use singular value decomposition (SVD). Let us denote the data covariance matrix by $\boldsymbol{R}_u = E\{\, \boldsymbol{u}\boldsymbol{u}^T\}$ where the superscript T denotes the transpose of vector or matrix. Then the SVD of $\boldsymbol{R}_u$ has the form

$$\boldsymbol{R}_u = \boldsymbol{U}_u \boldsymbol{D}_u \boldsymbol{U}_u^T, \tag{1}$$

where $\boldsymbol{U}_u$ is the eigenvector matrix and $\boldsymbol{D}_u$ is the diagonal matrix whose diagonal elements correspond to the eigenvalues of $\boldsymbol{R}_u$. Then the linear transformation $\boldsymbol{W}$ for PCA is given by

$$\boldsymbol{W} = \boldsymbol{U}_u^T. \tag{2}$$

For dimensionality reduction, one can choose p dominant column vectors in $\boldsymbol{U}_u$ which are the eigenvectors associated with the p largest eigenvalues in order to construct a linear transform $\boldsymbol{W}$. Many different methods for PCA have been developed. See [1, 4] for further details on PCA.

2.2. HMM

HMM is a widely-used probabilistic method which is useful in modelling time series data. It has been extensively used in speech recognition and computational biology. It is a simple dynamic Bayesian network which can represent probability distributions over sequences of observations.

Let us denote a sequence of observations $\{\boldsymbol{y}_t\}$ and a sequence of hidden states $\{\boldsymbol{s}_t\}$ where $t = 1, \ldots, T$. A HMM assumes two sets of conditional independence relations: (1) the observation $\boldsymbol{y}_t$ is independent of all other observations and states given $\boldsymbol{s}_t$; (2) the state $\boldsymbol{s}_t$ depends on only $\boldsymbol{s}_{t-1}$, i.e., states satisfy the first-order Markov property. It follows from these conditional independence relation that the joint probability distribution of states and observations can be factorized as

$$\begin{aligned} &P(\boldsymbol{s}_1, \ldots, \boldsymbol{s}_T, \boldsymbol{y}_1, \ldots, \boldsymbol{y}_T) \\ &\quad = P(\boldsymbol{s}_1) P(\boldsymbol{y}_1|\boldsymbol{s}_1) \prod_{t=2}^{T} P(\boldsymbol{s}_t|\boldsymbol{s}_{t-1}) P(\boldsymbol{y}_t|\boldsymbol{s}_t). \end{aligned} \tag{3}$$

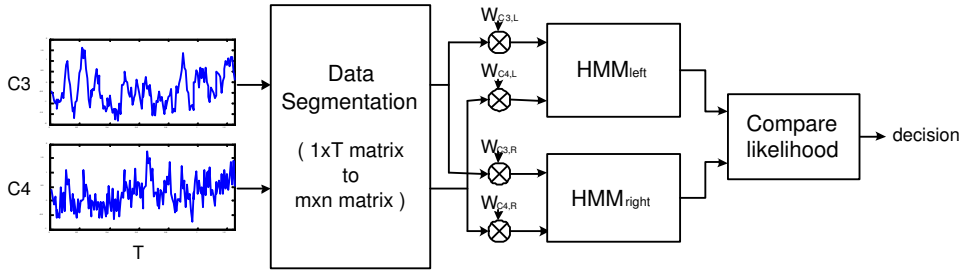


Figure 1: Schematic diagram for PCA-HMM1: Raw data is preprocessed in the stage of data segmentation to extract principal components which are used to learn two HMMs, each of which corresponds to left hand or right hand movement; Final decision resorts to the likelihood scores calculated by two HMMs.

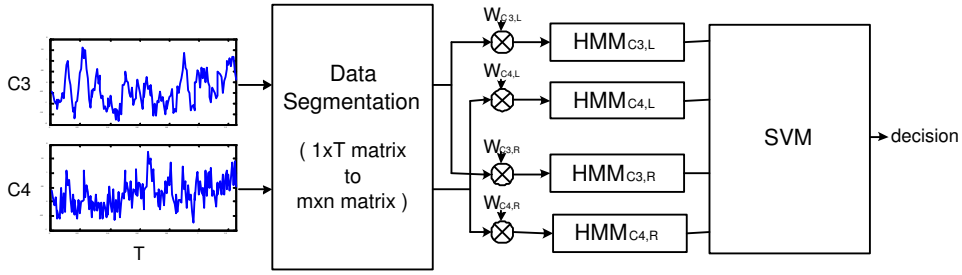


Figure 2: Schematic diagram for PCA-HMM2: Raw data is preprocessed in the stage of data segmentation to extract principal components; Principal components extracted from each channel are used to learn two HMMs, thus, in total, four HMMs are trained; Likelihood scores are fed into SVM to make a final decision.

A HMM assumes that hidden state variables are discrete-valued, i.e., $\boldsymbol{s}_t \in \{1, \ldots, K\}$. The state vector $\boldsymbol{s}_t$ is a K-dimensional vector with only one element being unity and the rest of elements being zeros. In other words, which element of the state vector is unity, depends on which state value is active. Then $P(\boldsymbol{s}_t|\boldsymbol{s}_{t-1})$ can be represented by a $K \times K$ state transition matrix that is denoted by $\boldsymbol{\Phi}$. $P(\boldsymbol{s}_1)$ is a K dimensional vector for initial state probability that is denoted by $\boldsymbol{\pi}$.

A HMM allows either discrete-valued observations (discrete HMM) or real-valued observations (continuous HMM). In this paper, we only consider a continuous HMM because EEG is real-valued data. For real-valued observation vectors, $P(\boldsymbol{y}_t|\boldsymbol{s}_t)$ can be modelled in many different forms such as a Gaussian, mixture of Gaussians, or a neural network.

Learning HMM consists of two steps: (1) inference step where the posterior distribution over hidden states is calculated; (2) learning step where parameters (such as initial state probability, state transition probability, and emission probability) are identified. The well-known forward-backward recursion allows us to infer the posterior over hidden states efficiently. More details on HMM can be found in [7, 2]

2.3. SVM

Support vector machine (SVM) has been widely used in pattern recognition and regression due to its computational efficiency and good generalization performance. It was originated from the idea of the structural risk minimization that was developed by Vapnik in 1970's [9].

Suppose we have a set of training data $\{(\boldsymbol{x}_1, z_1), \ldots, (\boldsymbol{x}_N, z_N)\}$. The decision function $f(\boldsymbol{x})$ has the form

$$f(\boldsymbol{x}) = \text{sgn}\left(\sum_{i=1}^{N} z_i \alpha_i k(\boldsymbol{x}, \boldsymbol{x}_i) + b\right), \tag{4}$$

where $\{\alpha_i\}$ are embedding coefficients and $k(\boldsymbol{x}, \boldsymbol{x}_i)$ is kernel that is represented by the dot product, i.e.,

$$k(\boldsymbol{x}, \boldsymbol{x}_i) = \langle \boldsymbol{x}, \boldsymbol{x}_i \rangle, \tag{5}$$

where $< \cdot, \cdot >$ denotes the dot product.

The optimal decision function is computed by the following quadratic programming

$$\text{maximize} \qquad \mathcal{J} = \sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_i \alpha_j z_i z_j k(\boldsymbol{x}_i, \boldsymbol{x}_j) \tag{6}$$

$$\text{subject to} \qquad \alpha_i \geq 0, \quad i = 1, \ldots, N, \quad \text{and} \sum_{i=1}^{N} \alpha_i z_i = 0. \tag{7}$$

More details on SVM can be found in [8].

3. METHODS

We present two methods which jointly employ PCA and HMM for EEG pattern classification. In our methods, we consider only C_3 and C_4 channels located in sensorimotor cortex because we focus

on binary classification of EEG patterns that are recorded during the imagination of either a left or right hand movement.

Schematic diagrams for our methods which are named as PCA-HMM1 and PCA-HMM2, are shown in Fig. 1 and Fig. 2, respectively. Both methods employ data segmentation procedure where time series data is decomposed into overlapping blocks in order to extract principal components. In PCA-HMM1, principal component features extracted separately from C_3 and C_4 channels are concatenated, then are fed into the corresponding HMM (which models either left-movement or right-movement) for training. On the other hand, in PCA-HMM2, principal component features from each channel are fed into two HMMs separately, which results in four HMMs in total. The SVM is employed to make a final decision from the likelihood scores computed by HMMs.

3.1. Feature Extraction: PCA

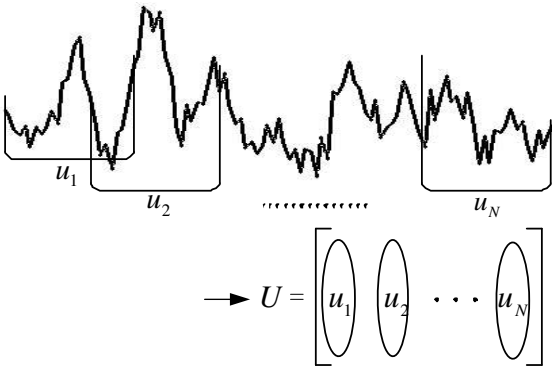


Figure 3: Data segmentation: Time series data is converted into a data matrix.

The time series data is decomposed into N overlapping blocks to construct $M \times N$ data matrix (where M is the number of data points in the data block, see Fig. 3) which is used to find a p by M matrix $\boldsymbol{W}$ for PCA. In our case, we calculate 4 matrixes - $\boldsymbol{W}_{C_3,L}$, $\boldsymbol{W}_{C_4,L}$, $\boldsymbol{W}_{C_3,R}$ and $\boldsymbol{W}_{C_4,R}$ (where subscripts C_3 and C_4 denote channels, L and R correspond to the imagination of left-hand and right-hand movement, respectively) in training phase. Then feature vector is computed by $\boldsymbol{v}_n = \boldsymbol{W}\boldsymbol{u}_n$.

The small noise will mainly appear in minor component directions which correspond to minor eigenvalues. Useful information is expected to lie in principal directions. Therefore we can expect PCA can reduce some noise effect as well as extracting useful features from time series data. Exemplary basis functions learned by PCA are shown in Fig. 4, which looks similar to wavelet basis functions. A major difference between PCA and wavelet transform is that the former learns basis functions from the ensemble of data, whereas the latter uses basis functions that are fixed in advance.

3.2. Classification: HMM+SVM

The principal component feature vector from C_3 channel for HMM_L (L means "left hand movement") is given by

$$\boldsymbol{v}_{C_3,L} = \boldsymbol{W}_{C_3,L}\boldsymbol{u}, \tag{8}$$

and $\boldsymbol{v}_{C_4,L}$ is also computed by $\boldsymbol{v}_{C_4,L} = \boldsymbol{W}_{C_4,L}\boldsymbol{u}$.

In the case of PCA-HMM1, the feature vector for HMM_L, $\boldsymbol{y}_L$, consists of principal components which concatenate $\boldsymbol{v}_{C_3,L}$ and $\boldsymbol{v}_{C_4,L}$, i.e., $\boldsymbol{y}_L = \left[\boldsymbol{v}_{C_3,L}^T, \boldsymbol{v}_{C_4,L}^T\right]^T$. The feature vector for HMM_R, $\boldsymbol{y}_R$, is constructed in the same manner. Both HMM_L and HMM_R are learned from a corresponding set of features that are computed from EEG data which is recorded during the imagination of either a left or right hand movement. Given a test EEG data, we compute likelihood values, i.e., $P(\boldsymbol{y}|HMM_L)$ and $P(\boldsymbol{y}|HMM_R)$ and assign it to the class which gives bigger likelihood value.

In the case of PCA-HMM2, the principal component feature vectors, $\boldsymbol{v}_{C_3,L}, \boldsymbol{v}_{C_3,R}, \boldsymbol{v}_{C_4,L}, \boldsymbol{v}_{C_4,R}$ are used separately to train the corresponding HMMs. For classification, given a test data $\boldsymbol{y}$, four different likelihood values are produced from $HMM_{C_i,L}$ and $HMM_{C_i,R}$ ($i = 3, 4$). These likelihood values are fed into the SVM to make a final decision. Although the PCA-HMM1 considers the interaction between channels, the dimension of its feature vector is twice larger than PCA-HMM2, which cause more complexity.

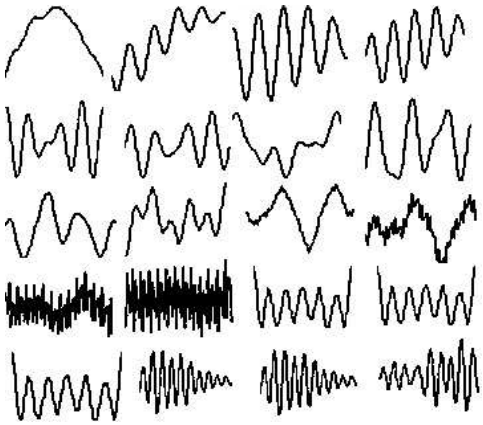
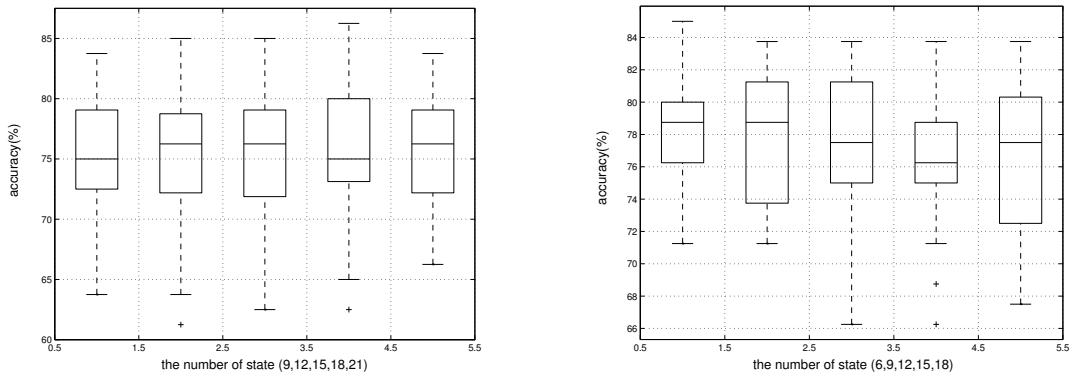


Figure 4: Basis functions calculated by PCA are shown. From left to right and from top to bottom, basis functions are in order of the size of corresponding eigenvalues.

4. RESULTS

Two bipolar EEG-channels were recorded over left and right sensorimotor areas, close to electrode positions C_3 and C_4. The EEG are sampled at 128 Hz and bandpass filtered between 0.5 and 30 Hz. The experimental trial is as follows. From 0 to 2 s, a fixation cross was presented, followed by the cue at 2 s. At 3 s an arrow was displayed at the center of the monitor for 1.25 s. Depending on the direction of the arrow presented left or right the subject was instructed to imagine a movement of either the left or the right hand. And then, feedback session continues from 4.25 to 8.0 s. One session constitutes 40 times repeating the course of the trial (20-left and 20-right). The total session is 4, so the number of trial is 160: 80-left and 80-right. We did not use feedback session. So the data from 3 to 4.25 s are used [3]. The window size for each data block is 0.5 s with overlapped portion being 87.5%, and the dimension are reduced by half of window size.



(a) (b)

Figure 5: Classification performance with respect to the number of states: (a) PCA-HMM1; (b) PCA-HMM2.

	HMM1	HMM2
PCA	**75.70**	**78.15**
Raw	60.63	64.38
Hjorth	56.88	66.50

Table 1: Performance comparison for PCA features, raw EEG data, and Hjorth parameters.

Fig. 5 show the box plots of classification accuracy for PCA-HMM1 and PCA-HMM2 when the number of states varies. The median is shown as a line across the box and the lower quartile and the upper quartile are also shown as lower and upper boundary of box. The minimum and maximum points are shown as lines. The performance of HMM did not vary much depending on the number of hidden states. The performance comparison for three different features (PCA, raw data, and Hjorth parameters) in PCA-HMM1 and PCA-HMM2 is summarized in Table 1. One can observe that PCA features outperform other features. PCA-HMM2 gave slight better performance compared to PCA-HMM1. The reason being is that in PCA-HMM2, separate HMMs were trained by principal component features that were separate channels.

In all cases, we didn't use the feedback session, but used the cue session between 3 and 4.25 s. In the case using Hjorth parameter, the results are worse than the result using raw data, because it extracts wrong information when the EEG data are mixed with some artifacts. Principal component features improved the performance of HMM by almost 10% and speeded up the convergence in the training of HMM. Hence PCA is a suitable feature extractor for EEG signal.

5. CONCLUSION

In this paper we have presented two methods for EEG pattern classification which jointly employ PCA and HMM (with SVM for PCA-HMM2). Experimental study showed that PCA was a good feature extractor for time series data.

Currently we are investigating theoretically why PCA gives better performance when it is combined with HMM for EEG pattern classification. In addition, a new structure (PCA-HMM2) showed slightly better performance compared to PCA-HMM1. The reason being is that PCA is applied to each channel separately so that separate HMMs model the data better.

6. ACKNOWLEDGMENT

We thank Graz BCI research group for sharing their EEG data. This work was supported by ETRI, by Korea Ministry of Science and Technology under Brain Science and Engineering Research Program and International Cooperative Research Program, and by Brain Korea 21.

7. REFERENCES

[1] K. I. Diamantaras and S. Y. Kung, *Principal Component Neural Networks: Theory and Applications*. John Wiely & Sons, INC, 1996.

[2] Z. Ghahramani, "An introduction to hidden markov models and Bayesian networks," *Journal of Pattern Recongition and Artificial Intelligence*, vol. 15, no. 1, pp. 9–42, 2001.

[3] C. Guger, A. Schlogl, C. Neuper, D. Walterspacher, T. Strein, and G. Pfurtscheller, "Rapid prototyping of an EEG-based Brain-Computer-Interface (BCI)," *IEEE Trans. Rehab. Engineering*, vol. 9, no. 1, pp. 49–58, 2001.

[4] I. T. Jolliffe, *Principal Component Analysis, 2nd Edition*. Spinger, 2002.

[5] B. Obermaier, C. Guger, C. Neuper, and G. Pfurtscheller, "Hidden markov models for online classification of single trial EEG data," *Pattern Recongnition Letters*, vol. 22, pp. 1299–1309, 2001.

[6] G. Pfurtscheller and C. Neuper, "Motor imagery and direct brain-computer communication," *Proc. of the IEEE*, vol. 89, no. 7, pp. 1123–1134, 2001.

[7] L. R. Rabiner and B. H. Juang, "An introduction to hidden Markov models," *IEEE Trans. Acoustics, Speech, and Signal Processing Magazine*, vol. 3, pp. 4–16, 1986.

[8] B. Schölkopf and A. J. Smola, *Learning with Kernels*. MIT Press, 2002.

[9] V. N. Vapnik, *The Nature of Statistical Learning Theory*. Spinger, 1995.