한국뇌학회지, Vol. 2, No. 1, pp. 53－59, June 2002

다채널 뇌파 분류를 위한 주성분 분석 기반 선형동적시스템

이혜경, 최승진
포항공과대학교 컴퓨터공학과

PCA-based Linear Dynamical Systems for Multichannel EEG classification

Hyekyoung Lee, Seungjin Choi
Dept. of Computer Science & Engineering, POSTECH, Korea

Abstract: EEG-based brain computer interface(BCI) provides a new communication channel between human brain and computer. The classification of EEG data is an important task in EEG-based BCI. In this paper we present methods which jointly employ principal component analysis(PCA) and linear dynamical system(LDS) modeling for the task of EEG classification. Experimental study for the classification of EEG data during imagination of a left or right hand movement confirms the validity of our proposed methods.

Key words: brain-computer interface(BCI), EEG, linear dynamical system(LDS), principal component analysis(PCA)

요 약: 뇌파를 기반으로 한 컴퓨터와 인간 간의 인터페이스는 인간의 뇌와 컴퓨터 간의 새로운 통신 채널을 제공하며, 뇌파의 패턴분류는 컴퓨터-인간 인터페이스에서 중요한 역할을 차지한다. 본 논문에서는 주성분분석과 선형동적시스템을 결합한 새로운 뇌파 패턴분류기를 구성하여 오른손이나 왼손의 움직임을 상상하는 데이터를 분류해보았다.

주요어: 뇌파, 선형동적시스템, 주성분분석, 컴퓨터와 인간 간의 인터페이스

Introduction

An important part in EEG-based BCI is the classification of circumscribed and transient EEG changes which are recorded during different types of motor imagery such as imagination of left-hand or right-hand movement. Features such as power spectrum, Hjorth parameters, or adaptive autoregressive parameters are extracted in EEG recordings of overlaying sensorimotor areas located over central and neighboring areas. For the classification of the features, linear discrimination analysis, neural networks, and hidden Markov models (HMMs) are used [1].

PCA is a well-known linear transformation for effective lower dimensional representation of the data. The principal components directions are merely sought by the dominant eigenvectors of the data covariance matrix. It is also known that PCA minimizes the reconstruction error. Because of its simplicity and good performance, PCA has been used in many areas such as image processing, speech processing, and etc for dimensionality reduction or feature extraction [6].

HMM and LDS are widely-used probabilistic methods for modeling time series data and belong to a class of linear Gaussian models [2]. Both HMM and

교신저자: 이혜경
포항공과대학교 컴퓨터공학과, 790-784 경북 포항시 남구 효자동 산 31번지, 포항공대 제2공학관 306호
Tel: 054-279-2924
Fax: 054-279-2299
E-mail: leehk@postech.ac.kr

LDS can represent dynamics by the hidden states which contains information about the past. They assume that the past, present and future observations are statistically independent known the state at any time and hidden states obey the Markov chain. LDS represents the past information through a real-valued hidden state vector, whereas HMM represents it through discrete-valued states. Therefore, LDS can be viewed as a continuous-state analogue of HMM. In LDS, the dependency between the present state vector and the previous state vector is specified through the dynamic equations of the system and the noise model. When these equations are linear and the noise model is Gaussian, the LDS is also known as a Kalman filter model. Almost BCI research group have focused the HMM model, not confirming state dynamicity of EEG signal [3]. In this paper, we employ the LDS as an alternative to HMM for the task of EEG classification.

We use the PCA to preprocess the observation sequence before the data is fed into either HMM or LDS. Our experimental study shows that PCA-based preprocessing accelerates the convergence of learning LDS and improves the classification performance. Detailed description of our proposed methods is illustrated in Section 3.

PCA and LDS

PCA

The PCA is a classical multivariate data analysis method that is useful in linear feature extraction and data compression. The PCA finds a linear transformation $v = Wu$ such that the retained variance is maximized. It can also be viewed as a linear transformation which minimizes the reconstruction error. The row vectors of W correspond to the normalized orthogonal eigenvectors of the data covariance matrix.

Let us denote the data covariance matrix by $R_u = E[UU^T]$. Then the SVD of R_u gives $R_u = D_u D_u U_u^T$ where U_u is the eigenvector matrix (i.e.,modal matrix) and D_u is the diagonal matrix whose diagonal elements correspond to the eigenvalues of R_u. Then the linear transformation W for PCA is given by $W=U_u^T$. For dimensionality reduction, one can choose p dominant column vectors in D_u that are eigenvectors that have the largest eigenvalues to construct a linear transform W.

LDS

Linear time-invariant dynamical systems (also known as linear Gaussian state space models) are described by

$$x_{t+1} = Ax_t + w_t \qquad w_t \sim N(0,Q) \qquad (1)$$

$$y_t = Cx_t + e_t \qquad e_t \sim N(0,R) \qquad (2)$$

where $A \in \mathrm{R}^{k\times k}$ is the state transition matrix and $C \in \mathrm{R}^{P\times k}$ is the output matrix.

The output y_t is a linear function of the state x_t which evolves through first-order Markov chain. Both state and output noise, w_t and e_t are zero-mean normally distributed random variables with covariance matrices Q and R, respectively. Only the output of the system is observed, the state and all the noise variables are hidden.

In case of HMM, we describe the state equation instead of (1) as

$$x_{t+1} = WTA[Ax_t + w_t] \qquad w_t \sim N(0,Q) \qquad (3)$$

where $WTA[\cdot]$is the winner-take-all nonlinearity defined such that $WTA[x]$ for any vector x is a new vector with unity in the position of the largest coordinates of the input and zeros in all other positions. Therefore, we can represent the equation (3) as a state transition matrix T, where $T_{ij} = P(x_{t+1} = j^{th}\ state \mid x_t = i^{th}\ state)$ [2]. From this, we can know the only difference between LDS and HMM is the dynamics of state - continuous in LDS and discrete in HMM.

The problems are to estimate the hidden states given observations and a model and to learning the model parameters - inference and system identification. Inference and system identification, or learning, can be solved by Kalman smoothing recursions and expectation maximization (EM) method, respectively. More details in appendix.

Proposed Methods

We consider C3 and C4 channels located in sensorimotor cortex related with (left or right) movement as well as imagination of movement. Fig. 1 shows our proposed methods, PCA-LDS1 and PCA-LDS2. Both methods employ data segmentation and feature extraction using PCA. In the PCA-LDS1, only two LDS models are learned, corresponding imagination of either left-hand or right-hand movement. Binary classification is carried out by likelihood scoring. In PCA-LDS2, 4 different LDS models are learned, corresponding either imagination of left-hand movement for C3 and C4 or imagination of right-hand movement for C3 and C4. Thus, four LDS models results in 4 likelihoods. For final decision, we employ the MLP feeding the likelihood scores. PCA-LDS2 does not consider coupling between C3 and C4 channels unlike PCA-LDS1. So, we can verify whether the interaction between channels affect the classification.

Feature Extraction

We decompose the data into N overlapping blocks to construct $M \times N$ data matrix (see Fig. 2) which is used to find a p by M matrix W for PCA.

In our case, we calculate 4 matrices - $W_{C3,\,L}$, $W_{C4,\,L}$, $W_{C3,\,R}$ and $W_{C4,\,R}$ (where subscripts $C3$ and $C4$ denote channels, L and R correspond to imagination of left-hand and right-hand movement, respectively) in

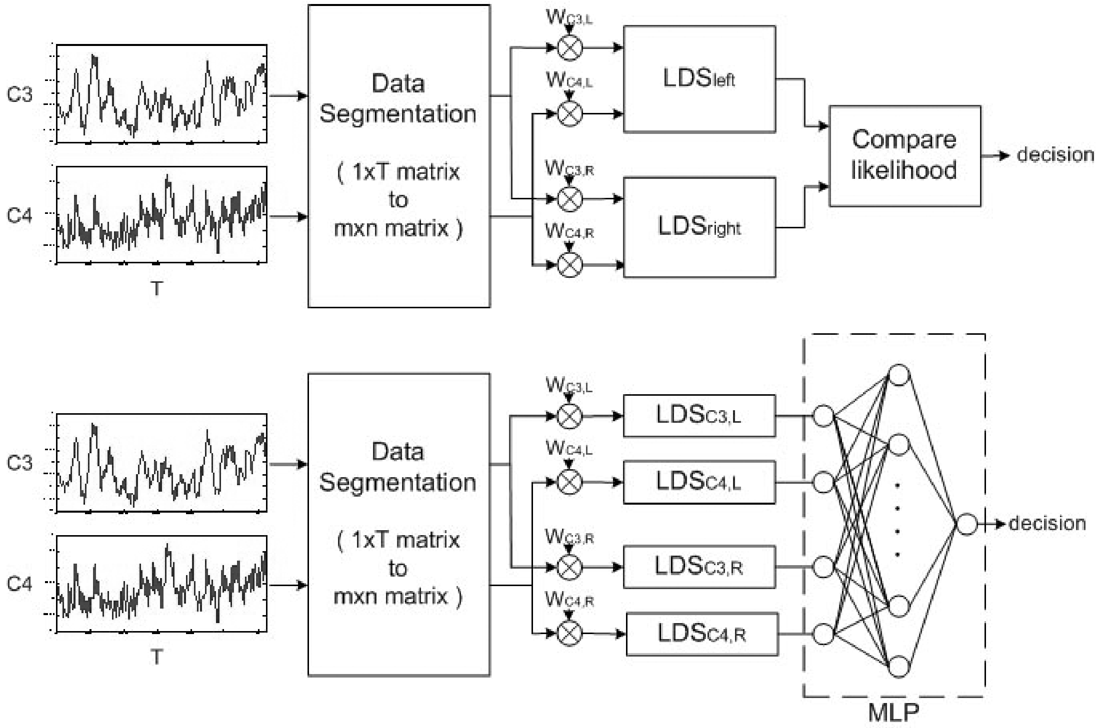

Fig. 1. Schematic diagram for (a) PCA-LDS1 and (b) PCA LDS2

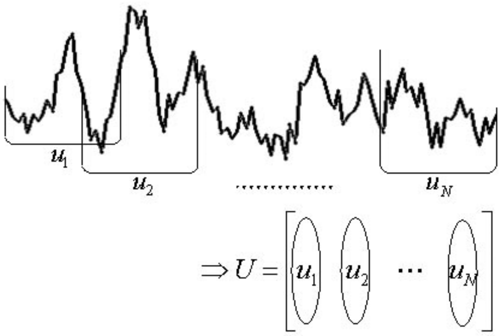

Fig. 2. Data segmentation

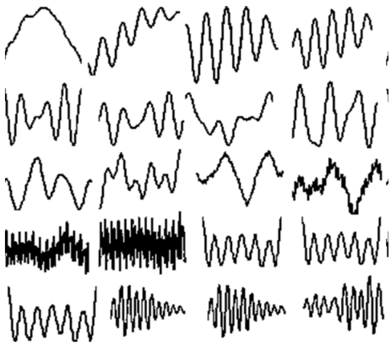

Fig. 3. Principal component - more upper and left, higher principal componetnt

training phase. Then feature vector is computed by

$$v_n = Wu_n \tag{4}$$

The noise will mainly affect the directions of the principal components corresponding to smaller eigenvalues and the information will mainly affect the directions of the others corresponding to higher eigenvalues. Therefore, eliminating redundant components can reduce the artifacts such as eye blinking, eye movement, muscle activity, interference of other channels, and etc. And the other components, known as principal components can function as a kind of filter that extract the information (Fig. 3). Dimensionality reduction also can reduce computational complexity in LDS.

Classification

PCA-LDS1 consist of LDS_{left} and LDS_{right}. Each LDS model is learned from a training set of data recorded during imagining movement, either left-hand or right-hand, respectively. In test phase, it feeds given a set of feature vectors, $Y = \{ y_1, y_2, \cdots, y_N\}$, where

$$y_n = \{ (v_{1,n}, \cdots, v_{p,n})_{C3}, (v_{1,n}, \cdots, v_{p,n})_{C4} \}, \tag{5}$$

$v_{n, C3} = W_{C3}\, u_{n, C3}$ and $v_{n, C4} = W_{C4}\, u_{n, C4}$ which is obtained from the set of test data (see Fig. 1 and equation (4)). And then, each LDS model computes likelihoods, $P(Y \mid LDS_{left})$ and $P(Y \mid LDS_{right})$and , and an appropriate class is assigned depending on which likelihood is larger.

In the case of PCA-LDS2, feature vector for each LDS model, $LDS_{C3, L}$, $LDS_{C4, L}$, $LDS_{C3, R}$ and $LDS_{C4, R}$ is given by

$$y_n = \{ (v_{1,n}, v_{2,n}, \cdots, v_{p,n}) \}, \tag{6}$$

where $v_n = Wu_n$. Each LDS model compute likelihoods, $P(Y \mid LDS_{CS, left})$, $P(Y \mid LDS_{CS, left})$, $P(Y \mid LDS_{CS, right})$, and $P(Y \mid LDS_{C4, right})$. These likelihood scores are fed into MLP to make a decision. And the MLP is trained in such a way that if the data is left-imagination, then the output is −1, otherwise the output is +1.

We can assume the PCA-LDS1 considers the interaction between channels, but has more complexity than PCA-LDS2 because its dimension of feature vector is twice larger than PCA-LDS2.

Experimental Results

Two bipolar EEG-channels were recorded over left and right sensorimotor hand areas, close to electrode positions C3 and C4. The EEG are sampled at 128 Hz and bandpass filtered between 0.5 and 30 Hz. Course of the experimental trial is followed. From 0 to 2 sec a fixation cross was presented, followed by the cue at 2 sec. At 3 sec an arrow was displayed at the centre of the monitor for 1.25 sec. 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 sec. 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 sec are only used. Detailed description on data can be found in [5].

In order to show that PCA is a good feature extractor, we compare the PCA-based features with Hjorth parameters [3] and raw data. We also compare LDS to continuous HMM, where methods are called PCA-HMM1 and PCA-HMM2. Methods based on the Hjorth parameter or raw data, are called as HJORTH-LDS1 or RAW-HMM1.

In the case of PCA and HJORTH, the window size is 0.5 sec, 64 points and each window is overlapped by 87.5%, i.e. shifted by 8 points. In the case of PCA, we reduce the dimension to 32. Each experiments is repeated 10 times using cross-validation - the training and test data are selected randomly and non-overlappingly among 160 trials with the ratio of training set and the testing set, 1 to 1. Figs. 4 and 5 show mean, maximum, and minimum of classification accuracy for several numbers or dimensions of state in order to obtain the optimal number and dimension of state for feature, using PCA-HMM1/2 and PCA-LDS1/2, respectively. In these figures, the x-axis of (a) shows the number of state values for HMM, and the x-axis of (b) represents the dimension of the state vector in LDS. The y-axis of both (a) and (b) represents the classification accuracy. Finding the optimal number and dimension of state, we compare the classification accuracy for each feature and each classifiers in Table 1. In the case using Hjorth parameter which is known suitable feature for EEG, the result is worse than that using raw data, because Hjorth parameter extract wrong information using the data mixed some artifact.

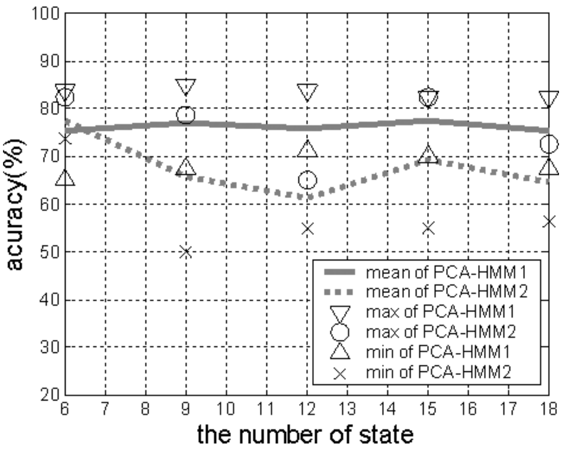

Fig. 4. Classification performance for PCA-based features: PCA-HMM1 and PCA-HMM2

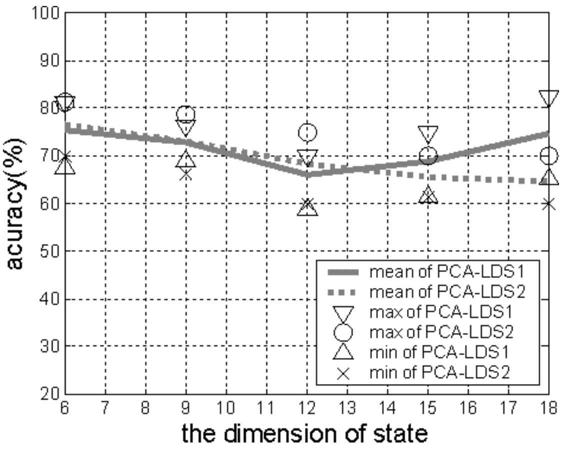

Fig. 5. Classification performance for PCA-based features: PCA-LDS1 and PCA-LDS2

Table 1. Classification accuracy (%).

	HMM1	HMM2	LDS1	LDS2
PCA	77.50	77.50	75.25	76.50
RAW	60.63	64.38	64.44	71.25
HJORTH	56.88	62.50	58.75	59.50

Table 2. Convergence speed [sec] which is estimated by matlab m-files on a Pentium IV 1.7GHz.

HMM1	HMM2	LDS1	LDS2
124.2660	78.9530	63.6250	42.1250

But in the case using PCA, we observed that the performance was improved by almost 10%, that the convergence speed of learning classifier was faster than others. So we confirm PCA is a suitable feature extractor for EEG signal. Both HMM and LDS showed similar performance, which might imply that the state dynamicity of EEG signal is not either purely continuous or purely discrete. However, as Table 2 where each classifier is compared the convergence speed for the optimal number and dimension of state found in Fig. 4 and 5, LDS required less complexity than HMM in the context of learning and it has more stable result than HMM in test phase. The case of using one classifier and different classifiers for each channel also showed similar performance, which might imply that the interaction between channels does not affect the results.

Conclusion

In this paper we presented LDS-based methods for multichannel EEG classification. We also employed PCA-based preprocessing so that LDS were trained from PCA-based features. We observed that PCA-based features had good performance and accelerated the convergence of learning LDS. Although the classification results of LDS and HMM were not different, the LDS is less expensive than HMM in complexity. Currently we are investigating switching state space models which is a combination of LDS and HMM for EEG classification.

Appendix

Algorithm Outline: LDS

We consider a sequence of T output vectors y_t and state vectors x_t. Due to the Markov property, the joint probability density, $P(x_t, y_t)$ can be described as

$$P(x_t, y_t) \tag{7}$$

We assume a Gaussian initial state density with mean π_1 and covariance matrixV_1 , $P(x_1)$ ~ $N(\pi_1, V_1)$. Since both state noise and output noise are also assumed to be Gaussian, we have $P(y_t | x_t)$ ~ $N(Cx_\tau, R)$ and $P(x_t | x_{t\text{-}1})$ ~ $N(Ax_{t-1}, Q)$. Therefore, the log-likelihood is given by

$$\begin{aligned} &\log P(\{x_t, y_t\}) = \\ &-\sum_{t=1}^{T}(\frac{1}{2}[y_t - Cx_t]^T R^{-1}[y_t - Cx_t]) \\ &-\sum_{t=2}^{T}(\frac{1}{2}[x_t - Ax_{t-1}]^T Q^{-1}[x_t - Ax_{t-1}]) \\ &-\frac{1}{2}[x_1 - \pi_1]^T V_1^{-1}[x_1 - \pi_1] - \frac{1}{2}\log|V_1| \\ &-\frac{T}{2}\log|R| - \frac{T-1}{2}\log|Q| - \frac{T(p+k)}{2}\log 2\pi \end{aligned} \tag{8}$$

The EM algorithm of LDS is below. This procedure iterates an E-step, which is also called Kalman smoothing recursions which is the method for inference and an M-step. In E step, we fix the current parameters and compute the posterior probabilities over the hidden states given the observations,

$$Q = E\ [\log\ P(\{x\},\ \{y\}\ |\ \{y\})]\ , \tag{9}$$

which depends on $E\ [x_t|\{y\}]$, $E\ [x_t\, x_{t\text{-}1}'|\{y\}]$ and denoted by $\hat{x}_t$, P_t and $P_{t,\,t\text{-}1}$, respectively. In M step, we obtain the parameters $\{A,\ C,\ Q,\ R,\ V_1,\ \pi_1\}$by maximizing the expected log likelihood of the parameters, (3) using the posterior distribution computed in E-step. See [4] for more details.

- Select the dimension of state.
- Initialize parameters of the model.
- Repeat until bound on log likelihood has converged:
- E-step

Denote $x_t^\tau \equiv E\ (x_t|\{y\}^{\tau_1})$ and $V_t^\tau \equiv Var\ (x_t|\{y\}^{\tau_1})$.

- Forward recursions:

where $x_1^0 = \pi_1$ and $V_1^0 = V_1$.

- Backward recursions;

$$J_{t-1} = V_{t-1}^{t-1} A' (V_t^{t-1})^{-1}$$
$$x_{t-1}^T = x_{t-1}^{t-1} + J_{t-1}(x_t^T - A x_{t-1}^{t-1})$$
$$V_t^T = V_{t-1}^{t-1} + J_{t-1}(V_t^T - V_{t-1,t}) J'_{t-1}$$
$$V_{t-1,t-2}^T = V_{t-1}^{t-1} J'_{t-2} + J_{t-1}(V_{t,t-1}^T - A V_{t-1,t-1}) J'_{t-2}$$

which is initialized as .

- Calculate

$$\hat{x}_t \equiv x_t^T,$$
$$P_t \equiv V_t^T + x_t^T (x_t^T)',$$
$$P_{t,t-1} \equiv V_{t,t-1}^T + x_t^T (x_{t-1}^T)'.$$

■ M-step

Update:

$$C^{\neq w} = (\sum_{t=1}^{T} y_t \hat{x}_t^{'})(\sum_{t=1}^{T} P_t)^{-1},$$
$$T^{\neq w} = \frac{1}{T}(\sum_{t=1}^{T} y_t y_t^{'} - C^{\neq w} \hat{x}_t y_t^{'}),$$
$$A^{\neq w} = (\sum_{t=2}^{T} P_{t,t-1})(\sum_{t=2}^{T} P_{t-1})^{-1},$$
$$Q^{\neq w} = \frac{1}{T-1}(\sum_{t=2}^{T} P_t - A^{\neq w} \sum_{t=2}^{T} P_{t-1,t}),$$
$$\pi_1^{\neq w} = \hat{x}_1,$$
$$V_1^{\neq w} = P_1 - \hat{x}_1 \hat{x}_1^{'}.$$

Acknowledgment

We thank Graz BCI research group for sharing their EEG data. This work was supported by ETRI and by Ministry of Education of Korea for its financial support toward the Electrical and Computer Engineering Division at POSTECH through its BK21 program.

References

1. G. Pfurtscheller and C. Neuper, Motor imagery and direct brain-computer communication, Proceedings of IEEE, vol.89, no.7, pp. 1123-1134, July 2001.
2. S. Roweis and Z. Ghahramani, A unifying review of linear Gaussian models, Neural Computation, vol. 11, no. 2, pp. 305-345, 1999.
3. B. Obermaier, C.Guger, C. Neuper, and G. Pfurtscheller, Hidden Markov models for online classification of single trial EEG data, Pattern Recognition Letters, vol. 22, pp. 1299-1309, 2001.
4. Z. Ghahramani, and G.E. Hinton, Parameter estimation for linear dynamical systems, University of Toronto Technical Report, CRG-TR-96-2, 1996.
5. 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. Engng., vol.9, no.1, pp. 49-58, 2001.
6. R. O. Duda, P. E. Hart, and D. G. Stork, Pattern Classification, Second Edition, A Wiley-Interscience Publication, 2001.