PCA-BASED LINEAR DYNAMICAL SYSTEMS FOR MULTICHANNEL EEG CLASSIFICATION

Hyekyoung Lee and Seungjin Choi

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

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.

1. 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 band power, 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 directions of principal components are merely sought by the eigenvectors of the data covariance matrix having the largest eigenvalues. It is known that PCA also minimizes the reconstruction error. Because of its simpleness and good performance, PCA has been used in many areas such as image processing, speech processing, etc for dimensionality reduction or feature extraction [6].

HMM and LDS are widely-used probabilistic models for time series data and belong to a class of linear Gaussian models [2]. Both HMM and LDS can represent dynamics by the hidden states which contains information about the past. They assume that the past, present and futrue observations are statistically independent known the state at any time and the hidden state obey the Markov independence property. LDS represent the past information through a real-valued hidden state vector, whereas HMM represent 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 state-space model or Kalman filter model. Almost BCI research group have researched model using HMM, 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.

2. PCA AND LDS

2.1. PCA

The PCA is a classical multivariate data analysis method that is useful in linear feature extraction and data compression. It is essentially equivalent to Karhunen-Loéve transformation and closely related to factor analysis. All these methods are based on second-order statistics of the data.

The PCA finds a linear transformation $\boldsymbol{v} = \boldsymbol{W}\boldsymbol{u}$ such that the retained variance is maximized. It can be also viewed as a linear transformation which minimizes the reconstruction error. The row vectors of $\boldsymbol{W}$ correspond to the normalized orthogonal eigenvectors of the data covariance matrix. One simple approach to PCA is to use singular value decomposition (SVD). Let us denote the data covariance matrix by $\boldsymbol{R}_u = E\{\, \boldsymbol{u}\boldsymbol{u}' \}$. Then the SVD of $\boldsymbol{R}_u$ gives

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

where $\boldsymbol{U}_u$ is the eigenvector matrix (i.e.,modal 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

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

For dimensionality reduction, one can choose p dominant column vectors in $\boldsymbol{U}_u$ that are eigenvectors that have the largest eigenvalues to construct a linear transform $\boldsymbol{W}$.

2.2. LDS

A brief review of LDS is given here. For details, refer to [4]. Linear time-invariant dynamical systems (also known as linear Gaussian state space models) are described by

$$\boldsymbol{x}_{t+1} = \boldsymbol{A}\boldsymbol{x}_t + \boldsymbol{w}_t, \qquad \boldsymbol{w}_t \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{Q}), \tag{3}$$

$$\boldsymbol{y}_t = C\boldsymbol{x}_t + \boldsymbol{e}_t, \qquad \boldsymbol{e}_t \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{R}), \tag{4}$$

where $\boldsymbol{A} \in \mathbb{R}^{k \times k}$ is the state transition matrix and $\boldsymbol{C} \in \mathbb{R}^{p \times k}$ is the output matrix. The output $\boldsymbol{y}_t$ is a linear function of the state

$\boldsymbol{x}_t$ which evolves through first-order Markov chain. Both state and output noise, $\boldsymbol{w}_t$ and $\boldsymbol{e}_t$ are zero-mean normally distributed random variables with covariance matrices $\boldsymbol{Q}$ and $\boldsymbol{R}$, respectively. Only the output of the system is observed, the state and all the noise variables are hidden.

We consider a sequence of T output vectors $\{\boldsymbol{y}_t\}$ and state vectors $\{\boldsymbol{x}_t\}$. Due to the Markov property, the joint probability density, $P(\{\boldsymbol{x}_t, \boldsymbol{y}_t\})$ can be described as

$$P(\{\boldsymbol{x}_t, \boldsymbol{y}_t\}) = P(\boldsymbol{x}_1)P(\boldsymbol{y}_1|\boldsymbol{x}_1)\prod_{t=2}^{T}[P(\boldsymbol{x}_t|\boldsymbol{x}_{t-1})P(\boldsymbol{y}_t|\boldsymbol{x}_t)]. \quad (5)$$

We assume a Gaussian initial state density

$$P(\boldsymbol{x}_1) = (2\pi)^{-k/2}|\boldsymbol{V}_1|^{-1/2} \exp\Big\{-\frac{1}{2}[\boldsymbol{x}_1 - \boldsymbol{\pi}_1]'\boldsymbol{V}_1^{-1}[\boldsymbol{x}_1 - \boldsymbol{\pi}_1]\Big\}. \quad (6)$$

Since both state noise and output noise are also assumed to be Gaussian, we have

$$P(\boldsymbol{y}_t|\boldsymbol{x}_t) = (2\pi)^{-p/2}|\boldsymbol{R}|^{-1/2} \exp\Big\{-\frac{1}{2}[\boldsymbol{y}_t - \boldsymbol{C}\boldsymbol{x}_t]'\boldsymbol{R}^{-1}[\boldsymbol{y}_t - \boldsymbol{C}\boldsymbol{x}_t]\Big\}, \quad (7)$$

$$(8)$$

and

$$P(\boldsymbol{x}_t|\boldsymbol{x}_{t-1}) = (2\pi)^{-k/2}|\boldsymbol{Q}|^{-1/2} \exp\Big\{-\frac{1}{2}[\boldsymbol{x}_t - \boldsymbol{A}\boldsymbol{x}_{t-1}]'\boldsymbol{Q}^{-1}[\boldsymbol{x}_t - \boldsymbol{A}\boldsymbol{x}_{t-1}]\Big\} \quad (9)$$

Therefore, the log-likelihood is given by

$$\begin{aligned}\log P(\{\boldsymbol{x}_t,\boldsymbol{y}_t\}) = &-\sum_{t=1}^{T}\Big(\frac{1}{2}[\boldsymbol{y}_t - \boldsymbol{C}\boldsymbol{x}_t]'\boldsymbol{R}^{-1}[\boldsymbol{y}_t - \boldsymbol{C}\boldsymbol{x}_t]\Big) \\ &-\sum_{t=2}^{T}\Big(\frac{1}{2}[\boldsymbol{x}_t - \boldsymbol{A}\boldsymbol{x}_{t-1}]'\boldsymbol{Q}^{-1}[\boldsymbol{x}_t - \boldsymbol{A}\boldsymbol{x}_{t-1}]\Big) \\ &-\frac{1}{2}[x_1 - \boldsymbol{\pi}_1]'\boldsymbol{V}_1^{-1}[x_1 - \boldsymbol{\pi}_1] - \frac{1}{2}\log|\boldsymbol{V}_1| \\ &-\frac{T}{2}\log|\boldsymbol{R}| - \frac{T-1}{2}\log|\boldsymbol{Q}| - \frac{T(p+k)}{2}\log 2\pi \quad (10)\end{aligned}$$

Given the observation sequence, the parameters $\{\boldsymbol{A}, \boldsymbol{C}, \boldsymbol{Q}, \boldsymbol{R}, \boldsymbol{V}_1, \boldsymbol{\pi}_1\}$ can be learned by the expectation maximization (EM) method. The EM algorithm [4] is summarized below. This procedure iterates an E-step - Kalman smoothing recursions that fixes the current parameters and computes posterior probabilities over the hidden states given the obserations, and an M-step that maximizes the expected log likelihood of the parameters using the posterior distribution computed in E-step.

- Algorithm Outline: LDS

Initialize parameters of the model.

Repeat until bound on log likelihood has converged:

E-step

Denote $\boldsymbol{x}_t^\tau \equiv E(\boldsymbol{x}_t|\{\boldsymbol{y}\}_1^\tau)$ and $\boldsymbol{V}_t^\tau \equiv \mathrm{Var}(\boldsymbol{x}_t|\{\boldsymbol{y}\}_1^\tau)$.

Forward recursions:

$$\begin{aligned}\boldsymbol{x}_t^{t-1} &= \boldsymbol{A}\boldsymbol{x}_{t-1}^{t-1}, \\ \boldsymbol{V}_t^{t-1} &= \boldsymbol{A}\boldsymbol{V}_{t-1}^{t-1}\boldsymbol{A}' + \boldsymbol{Q}, \\ \boldsymbol{K}_t &= \boldsymbol{V}_t^{t-1}\boldsymbol{C}'(\boldsymbol{C}\boldsymbol{V}_t^{t-1}\boldsymbol{C}' + \boldsymbol{R})^{-1}, \\ \boldsymbol{x}_t^t &= \boldsymbol{x}_t^{t-1} + \boldsymbol{K}_t(\boldsymbol{y}_t - \boldsymbol{C}\boldsymbol{x}_t^{t-1}), \\ \boldsymbol{V}_t^t &= \boldsymbol{V}_t^{t-1} - \boldsymbol{K}_t\boldsymbol{C}\boldsymbol{V}_t^{t-1},\end{aligned}$$

where $\boldsymbol{x}_1^0 = \boldsymbol{\pi}_1$ and $\boldsymbol{V}_1^0 = \boldsymbol{V}_1$.

Backward recursions;

$$\begin{aligned}\boldsymbol{J}_{t-1} &= \boldsymbol{V}_{t-1}^{t-1}\boldsymbol{A}'(\boldsymbol{V}_t^{t-1})^{-1} \\ \boldsymbol{x}_{t-1}^T &= \boldsymbol{x}_{t-1}^{t-1} + \boldsymbol{J}_{t-1}(\boldsymbol{x}_t^T - \boldsymbol{A}\boldsymbol{x}_{t-1}^{t-1}) \\ \boldsymbol{V}_t^T &= \boldsymbol{V}_{t-1}^{t-1} + \boldsymbol{J}_{t-1}(\boldsymbol{V}_t^T - \boldsymbol{V}_{t-1,t})\boldsymbol{J}_{t-1}' \\ \boldsymbol{V}_{t-1,t-2}^T &= \boldsymbol{V}_{t-1}^{t-1}\boldsymbol{J}_{t-2}' \\ &\quad +\boldsymbol{J}_{t-1}(\boldsymbol{V}_{t,t-1}^T - \boldsymbol{A}\boldsymbol{V}_{t-1,t-1})\boldsymbol{J}_{t-2}'\end{aligned}$$

which is initialized as $\boldsymbol{V}_{T,T-1}^T = (I - \boldsymbol{K}_T\boldsymbol{C})\boldsymbol{A}\boldsymbol{V}_{T-1}^{T-1}$.

Calculate

$$\begin{aligned}\hat{\boldsymbol{x}}_t &\equiv \boldsymbol{x}_t^T, \\ \boldsymbol{P}_t &\equiv \boldsymbol{V}_t^T + \boldsymbol{x}_t^T\boldsymbol{x}_t^{T'}, \\ \boldsymbol{P}_{t,t-1} &\equiv \boldsymbol{V}_{t,t-1}^T + \boldsymbol{x}_t^T\boldsymbol{x}_{t-1}^{T'}.\end{aligned}$$

M-step

Update:

$$\begin{aligned}\boldsymbol{C}^{new} &= \left(\sum_{t=1}^{T}\boldsymbol{y}_t\hat{\boldsymbol{x}}_t'\right)\left(\sum_{t=1}^{T}\boldsymbol{P}_t\right)^{-1} \\ \boldsymbol{R}^{new} &= \frac{1}{T}\sum_{t=1}^{T}\left(\boldsymbol{y}_t\boldsymbol{y}_t' - \boldsymbol{C}^{new}\hat{\boldsymbol{x}}_t'\boldsymbol{y}_t'\right) \\ \boldsymbol{A}^{new} &= \left(\sum_{t=2}^{T}\boldsymbol{P}_{t,t-1}\right)\left(\sum_{t=2}^{T}\boldsymbol{P}_{t-1}\right)^{-1} \\ \boldsymbol{Q}^{new} &= \frac{1}{T-1}\left(\sum_{t=2}^{T}\boldsymbol{P}_t - \boldsymbol{A}^{new}\sum_{t=2}^{T}\boldsymbol{P}_{t-1,t}\right) \\ \boldsymbol{\pi}_1^{new} &= \hat{\boldsymbol{x}}_1 \\ \boldsymbol{V}_1^{new} &= \boldsymbol{P}_1 - \hat{\boldsymbol{x}}_1\hat{\boldsymbol{x}}_1'\end{aligned}$$

3. PROPOSED METHODS

We consider C3 and C4 channels located in sensorimotor cortex related with (left or right) movement as well as imagination of movement. Figure 1 and figure 2 shows our proposed methods, PCA-LDS1 and PCA-LDS2, respectively. 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.

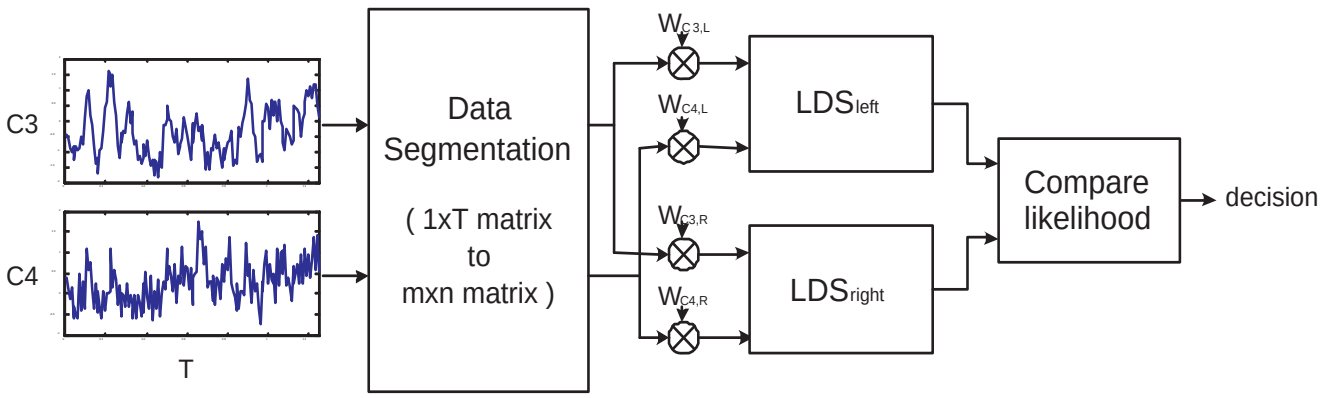

Figure 1: Schematic diagram for PCA-LDS1.

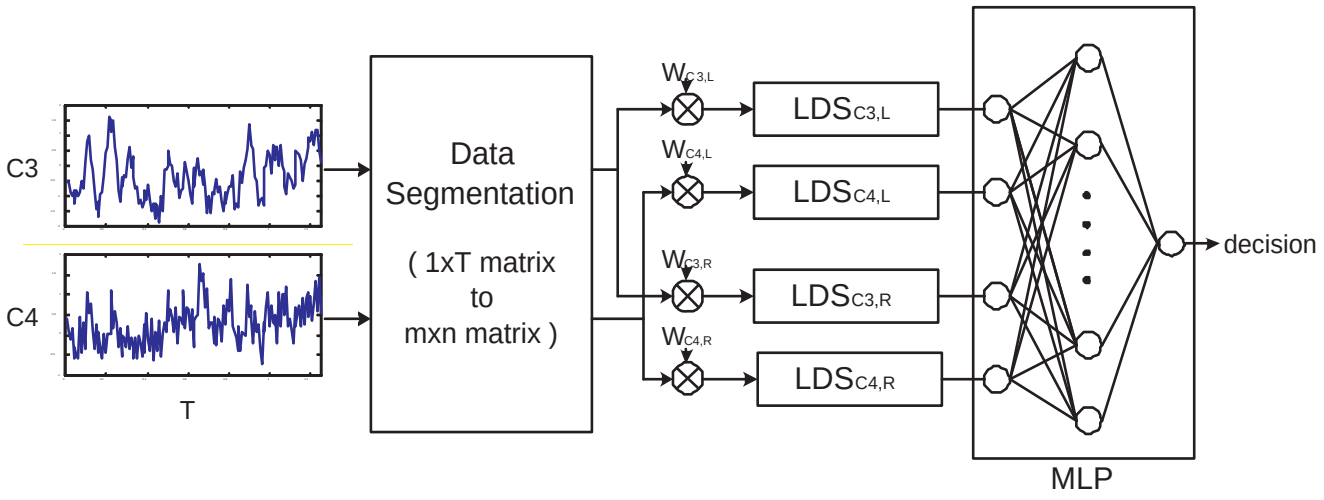

Figure 2: Schematic diagram for PCA-LDS2.

3.1. Feature Extraction

We decompose the data into N overlapping blocks to construct $M \times N$ data matrix (see figure 3) which is used to find a p by M matrix $\boldsymbol{W}$ for PCA. In our case, we calculate 4 matrixes - $\boldsymbol{W}_{C3,L}$, $\boldsymbol{W}_{C4,L}$, $\boldsymbol{W}_{C3,R}$ and $\boldsymbol{W}_{C4,R}$ (where subscripts C_3 and C_4 denote channels, L and R correspond to imagination of left-hand and right-hand movement, respectively) in training phase. Then feature vector is computed by $\boldsymbol{v}_n = \boldsymbol{W}\boldsymbol{u}_n$.

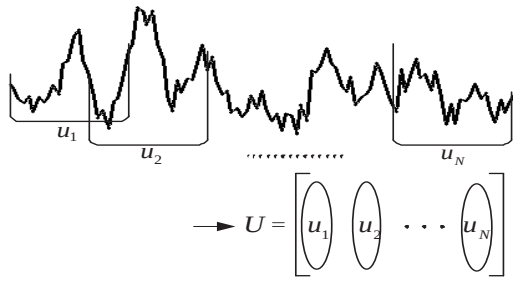

Figure 3: Data segmentation.

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, etc. And the other components, known as principal components can function as a kind of filter that extract the information (figure 4). Dimensionality reduction also can reduce computational complexity in LDS.

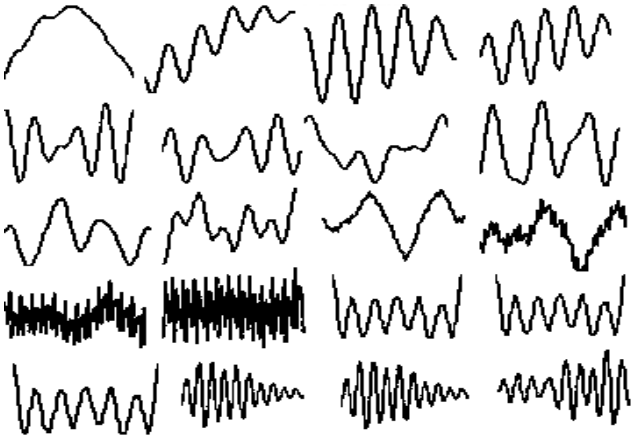

Figure 4: Principal Component : more upper and left, higher principal component.

3.2. Classification

In the case of PCA-LDS1, feature vector in each LDS is

$$\boldsymbol{y}_n = \left\{(v_{1,n}, \ldots, v_{p,n})_{C3}, (v_{1,n}, \ldots, v_{p,n})_{C4}\right\}. \quad (11)$$

PCA-LDS1 consist of LDS_{left} and LDS_{right}. Each LDS model learned from a training set of data recorded during imagining movement of left-hand and right-hand, respectively. Feeding given a set of feature vectors obtained from the set of test data, $\boldsymbol{Y} = \{\boldsymbol{y}_1, \boldsymbol{y}_2, \ldots, \boldsymbol{y}_N\}$, each LDS model compute likelihoods, $P(\boldsymbol{Y}|LDS_{left})$ and $P(\boldsymbol{Y}|LDS_{right})$, and an appropriate class is assigned depending on which likelihood is larger.

In the case of PCA-LDS2, feature vector for each LDS is given by

$$\boldsymbol{y}_n = \{(v_{1,n}, v_{2,n}, \ldots, v_{p,n})\}. \quad (12)$$

Each LDS model compute likelihoods, $P(\boldsymbol{Y}|LDS_{C3,left})$, $P(\boldsymbol{Y}|LDS_{C4,left})$, $P(\boldsymbol{Y}|LDS_{C3,right})$ and $P(\boldsymbol{Y}|LDS_{C4,right})$. These likelihood scores are fed into MLP to make a decision. 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.

4. 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 s a fixation cross was presented, followed by the cue at 2 s. At 3 s an arrow was displayed at the centre 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 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, and we call them PCA-HMM1 and PCA-HMM2 Methods based on the Hjorth parameter or raw data, are called as RAW-HMM1,HJORTH-LDS1.

In the case of PCA and HJORTH, the window size is 0.5 sec with overlapped portion being 0.875. And in the case of PCA, we reduce the dimension to 20. Figure 5 compares the classification accuracy using each feature extractor and each classifier. And figure 6 and figure 7 show mean, maximum, and minimum of classification accuracy for PCA-based feature using clssifier - HMM and LDS. 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.

In the case using Hjorth parameter which is known suitable feature for EEG, the result are worse than the result using raw data. 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, LDS required less complexity than HMM in the context of learning and it has more stable result than HMM in test phase.

	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

Figure 5: Compare the classification accuracy : column means feature extractor and row means classifier

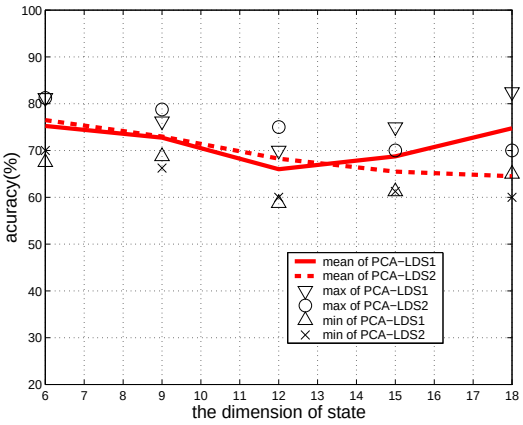

Figure 6: Classification performance for PCA-based features: PCA-LDS1 and PCA-LDS2

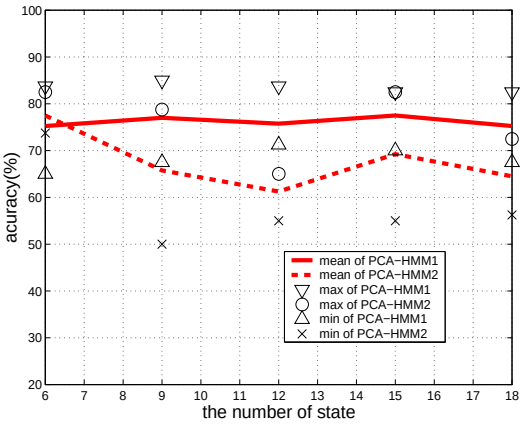

Figure 7: Classification performance for PCA-based features: PCA-HMM1 and PCA-HMM2.

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.

5. 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.

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 by Ministry of Education of Korea for its financial support toward the Electrical and Computer Engineering Division at POSTECH through its BK21 program.

7. 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," *A Wiley-Interscience Publication*, Second Edition, 2001.