

Self-Labeling for P300 Detection

Sangmin Lee*, Yunjun Nam[†] and Seungjin Choi*^{†‡§}

*Division of IT Convergence Engineering,

[†]School of Interdisciplinary Bioscience and Bioengineering,

[‡]Department of Computer Science and Engineering,

[§]Department of Creative IT Excellence Engineering,

Pohang University of Science and Technology,

77 Cheongam-ro, Nam-gu, Pohang 790-784, Korea

{mangdoo,druid,seungjin}@postech.ac.kr

Abstract—The P300 wave refers to a positive peak with a latency of 300 ms, produced in response to task-relevant stimuli. It is a widely-used event related potential (ERP) in practical brain computer interface (BCI) systems, where a classifier is trained using discriminative features extracted from a set of labeled examples, in order to detect the presence of P300. Given a small training examples, the intra- and inter-subject variations in amplitude and latency of P300 degrade the performance of classifier. Thus, it requires a longer training period (calibration time) with more labeled examples for satisfactory performance. In this paper we present a *self-labeling* method, where confident unlabeled data, together with their predicted labels, are gradually added to the training set, in order to re-train the classifier. Linear discriminant analysis with singular value decomposition is used to progressively extract discriminate features. Experiments demonstrate the high performance of our method, especially in the case where a small number of training examples are available. We also apply the method to the zero-calibration P300-based BCI, which removes subject-dependent calibration procedures by using the training set already recorded from other subjects.

Index Terms—Brain computer interface (BCI), linear discriminant analysis (LDA), P300 detection, self-labeling, singular value decomposition (SVD)

I. INTRODUCTION

Brain computer interface (BCI) is a system that makes use of brain signals to translate a subject’s intention or mind into a control signal for a device such as a computer, a wheelchair, or a neuroprosthesis [1], [2], [3]. One of popular modalities in BCI systems is P300 that is a positive peak with a latency of 300 ms on electroencephalography (EEG) sensors, produced in response to task-relevant stimuli [4]. A widely-used P300-based BCI is the P300 speller [5], in which characters are displayed as a 6×6 matrix with rows and columns flashed in a random order and P300 responses are elicited while a specific entry of the matrix corresponding to the character of interest is highlighted. Various methods have been developed for P300 detection [6], [7], [8], [9].

In general, P300 responses are subject-specific, since their peak and latency vary from person to person and time to time, according to the subject’s age and cognitive capability [10]. Intra- and inter-subject variations requires a classifier to be trained using examples labeled by a subject of interest for satisfactory performance. This procedure is referred to as calibration, which is a time-consuming task. Thus, it is desirable to reduce the training period for practical P300-based

BCI systems. To tackle this problem, several techniques have been developed [11], [12], [13]. Rivet et al. [14] developed a P300 speller system with a reduced calibration duration that can automatically switch between the training and testing sessions. Panicker *et al.* [15] reduced the calibration duration by a co-training method where both labeled and unlabeled examples are used together.

In this paper, we present a *self-labeling* method for P300 detection where confident unlabeled data, together with their predicted labels, are gradually and selectively added to the training set, in order to re-train the classifier. We first reduce the dimension of data by a low-rank projection into the space spanned by leading singular vectors of the data matrix, and then discriminative features are extracted using linear discriminant analysis (LDA). We also present results for the case of zero-calibration (no labeled examples by a subject of interest), where the classifier is adapted, starting from the model initialized by other subjects.

II. METHODS

A. Data Acquisition

Four healthy subjects (one female, age: 26 ± 1.82) participated in the study. All subjects were healthy graduate students with no known neurological problems. The EEG signals sampled at 256Hz were recorded at Fz, C3, Cz, C4, P3, Pz, P4 and Oz sites following the international 10-20 system using g.MOBILab+ and g.GAMMASys. The electrodes on the left mastoid and AFz were used as a reference and ground electrode respectively. Fig. 1(a) shows the channel location.

For data acquisition, we implemented the P300-based interface, as shown in Fig. 1(b). The six gray arrows are oriented as a circle on the black screen, and the yellow target arrows are displayed on the upside of the screen to act as a cue. For every sequence, each arrow only flashed white once in a random order, and six sequences were repeated in a trial for selecting one target arrow, as shown in Fig. 2. The intensification lasted 120ms with an inter-stimulus interval of 80ms, and the time pausing between each trial is 5s. Finally, the required time to input a single arrow is 12.2s.

During this step, the subjects were comfortably seated in a chair at a distance of approximately 1m from the computer screen. The subjects were asked to focus on a target arrow

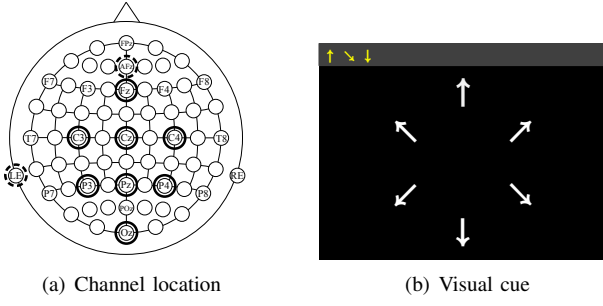


Fig. 1. Channel location and P300 interface

that was cued by an arrow on the upside of the screen in a sequence, and silently count how many times the target arrows flashed. The sessions performed by the subjects comprised 24 target arrows (calibration/training session) and 72 target arrows (test session).

B. Preprocessing

In order to obtain a clear ERP, we preprocess the signal in following 4 steps.

1) *Filtering*: Since P300 is a slow-wave response, EEG signals from each channel are band-pass filtered from 1 to 20 Hz.

2) *Segmentation*: EEG signals were segmented from 100ms pre-stimulus to 600ms post-stimulus, which is enough to contain the P300 waveform.

3) *Normalization*: For the baseline correction, the mean of each pre-stimulus signal was subtracted from each post-stimulus signal, respectively. We then divided the result by the variance of the signals including pre and post-stimulus.

4) *Averaging*: In order to obtain a robustly shaped P300 ERP, the normalized signals were averaged over all channels. Then, this averaged signals for each arrow were also averaged over all sequences in a single trial.

Let us denote the target signal as $\mathbf{x} \in \mathbb{R}^T$, where T is the length of the signal, which is recorded during the flash of the arrow that the subject was focused upon, and denote the non-target signal as $\mathbf{y} \in \mathbb{R}^T$, which is recorded during the flash of other arrows. Then the signals recorded during the n -th trial can be denoted as $\mathbf{S}_n = [\mathbf{x}_n, \mathbf{y}_n^1, \dots, \mathbf{y}_n^5]$ corresponding to 6 kinds of arrows.

C. Feature Extraction: Singular Value Decomposition

For our study, we applied the SVD as a feature extraction method. We used the SVD (instead of the Principal Component Analysis (PCA)), to consider both of the temporal and trial information. The method for using the temporal information will be discussed in our future works.

The goal of this step is to extract the common waveforms from among the target signals in order for them to be used as basis vectors. To train the pattern of P300, we collected target signals including P300, as $\mathbf{X} = [\mathbf{x}_1, \dots, \mathbf{x}_N]$, which is $\mathbf{X} \in \mathbb{R}^{T \times N}$, when \mathbf{S} includes N trials. The remained non-target signals were collected as $\mathbf{Y} \in \mathbb{R}^{T \times 5N}$. Then, the SVD can decompose \mathbf{X} into 3 matrices as

$$\mathbf{X} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^\top, \quad (1)$$

where $\mathbf{U} \in \mathbb{R}^{T \times T}$ and $\mathbf{V} \in \mathbb{R}^{N \times N}$ are orthogonal matrices and $\mathbf{\Sigma}$ is a diagonal matrix. The columns \mathbf{u}_i in \mathbf{U} and \mathbf{v}_i in \mathbf{V} are the left and right singular vectors, respectively and the diagonal elements of $\mathbf{\Sigma}$ are the singular values [16].

We concatenate vectors from the matrix \mathbf{U} corresponding to the l -largest singular values in $\mathbf{\Sigma}$ as a basis matrix denoted by $\hat{\mathbf{U}} = [\mathbf{u}_1, \dots, \mathbf{u}_l]$, which is $\hat{\mathbf{U}} \in \mathbb{R}^{T \times l}$. Then, the feature matrix, $\mathbf{F}_X \in \mathbb{R}^{l \times N}$, for \mathbf{X} is obtained by

$$\mathbf{F}_X = \hat{\mathbf{U}}^\top \mathbf{X}. \quad (2)$$

The first waveform, \mathbf{u}_1 , represents the most dominant and widespread feature of the target signals. Subsequent waveforms occur in descending order of contribution to \mathbf{X} . The exemplary basis vectors can be seen in Fig. 6.

D. P300 Detection: LDA and Minimum Distance Classifier

Our objective in this step is to detect the target signal from each test signals. To do this, we used LDA and the minimum distance measure because of their lower computational cost and simplicity [17] [18].

From the feature matrices $\mathbf{F}_X = \hat{\mathbf{U}}^\top \mathbf{X}$ and $\mathbf{F}_Y = \hat{\mathbf{U}}^\top \mathbf{Y}$, we measured the LDA projection vector \mathbf{w} to separate data into two classes. From the one dimensional projected features $\mathbf{w}^\top \mathbf{F}_X$ and $\mathbf{w}^\top \mathbf{F}_Y$, we measured their mean values μ_X and μ_Y . These values are used as referential points for the P300 detection and the self-labeling.

For every trial of test procedure, we obtained 6 signals, $\mathbf{z}_1, \dots, \mathbf{z}_6$, corresponding to the 6 arrows. One of these signals is selected as \mathbf{z}_X which is the target signal regarded as having P300 component, and its corresponding arrow was transmitted to the device as the detection result. In order to decide which signal is the target signal, we measured the distances of their corresponding feature values from μ_X . μ_X is the mean value of $\mathbf{w}^\top \mathbf{F}_X$, which represents the group mean of feature values for the target set, as we mentioned above. That is, the distance d_{X_i} for the \mathbf{z}_i was measured as

$$d_{X_i} = |\mathbf{w}^\top \mathbf{f}_i - \mu_X| \quad (3)$$

where \mathbf{f}_i is $\hat{\mathbf{U}}^\top \mathbf{z}_i$. Among $\mathbf{z}_1, \dots, \mathbf{z}_6$, the signal with the minimum d_{X_i} is selected as the target signal \mathbf{z}_X and others as non-target signals, $\mathbf{z}_Y^1, \dots, \mathbf{z}_Y^5$.

E. Self-Labeling

The goal of our algorithm is reduce the calibration time by using a smaller training set. To compensate for information loss due to the smaller training data, we append the signals from the test procedure to the training set, then retrain the basis vectors and the detection model after every trial. The signal classified as the target set was appended to \mathbf{X} , and the signal classified as the non-target set was appended to \mathbf{Y} .

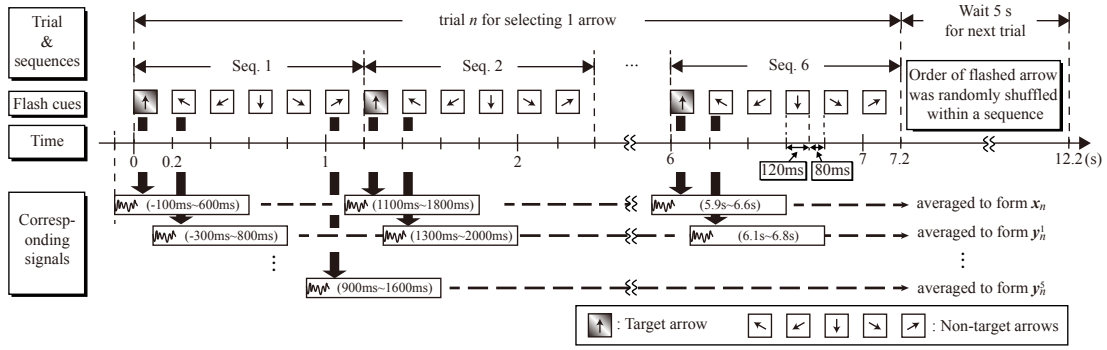


Fig. 2. Cues and corresponding signals for the single trial inputting ‘↑’

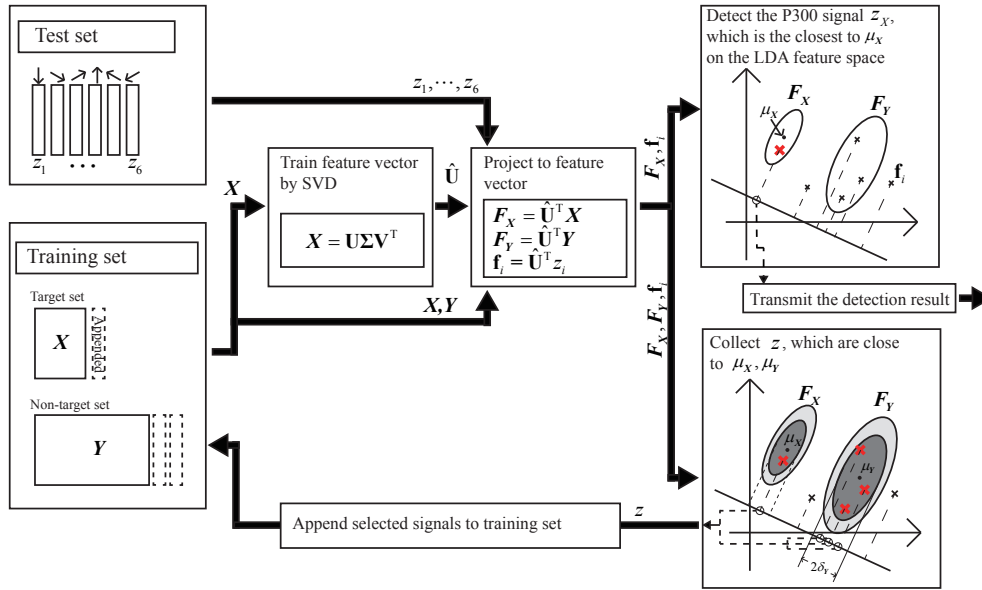


Fig. 3. Algorithm Overview

However, we could not guarantee that the detection results of the test procedure were always correct. Furthermore, trials of the test procedure can be invalid for various reasons. For example, if the subject miss the flash on the cued arrow, there will be no P300-contained trial in the test set. However, even in this case, the system will try to append one of the test signals to the target set. Then, the performance of the system will have deteriorated, because the signal without P300 is included in the target set within the training data. In addition, the signals in the test procedure could be corrupted by various artifacts, such as may arise from electrooculogram or electromyogram. Usually, these artifacts exhibit amplitudes with a range of a few millivolts, while event-related potentials exhibits amplitudes ranging between 2 and $20\mu V$. Generally, LDA is vulnerable to such outliers.

To tackle this problem, we selectively append the signals that are similar to the original target and non-target sets in the existed training set. The similarities to the original target and non-target sets are measured by d_{X_i} and d_{Y_i} , which are the

distance of feature values from the group mean of target and non-target sets. Likewise d_{X_i} in Eq. 3, d_{Y_i} is defined to

$$d_{Y_i} = |\mathbf{w}^\top \mathbf{f}_i - \mu_Y|, \quad (4)$$

where μ_Y is the mean value of $\mathbf{w}^\top \mathbf{F}_Y$.

With this similarity measure, to decide whether the detected target signal z_X is appended to \mathbf{X} or not, we used σ_X , the standard deviation corresponding to the feature values of the target set, $\mathbf{w}^\top \mathbf{F}_X$.

That is, σ_X was defined to

$$\sigma_X = \sqrt{\frac{1}{N_X} \sum_{n=1}^{N_X} (\mathbf{w}^\top \mathbf{F}_X - \mu_X)^2} \quad (5)$$

where N_X is the number of trials of the training set.

z_X is appended to \mathbf{X} , only when its corresponding feature value, $\mathbf{w}^\top \hat{\mathbf{U}}^\top z_X$, is located within the range of $\mu_X - \sigma_X$ and $\mu_X + \sigma_X$.

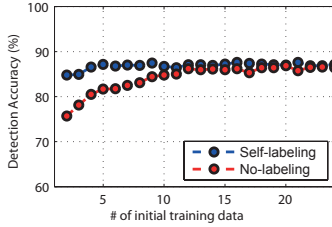


Fig. 4. Averaged accuracies in relation to the number of training data

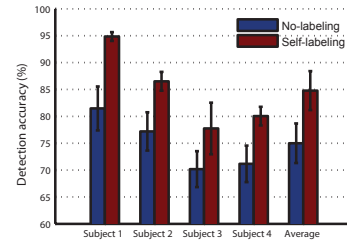


Fig. 5. Error bars of the result trained with 2 data

Likewise, non-target signals, z_Y^1, \dots, z_Y^5 , were appended only when their corresponding feature values are located within the range of $\mu_Y - \sigma_Y$ and $\mu_Y + \sigma_Y$.

The above method can be denoted by the following algorithm.

Algorithm

After the training, we are left with a target set, \mathbf{X} , and a non-target set, \mathbf{Y} , as the initial training data. During the test procedure, new signals, z_1, \dots, z_6 , are recorded for every trial.

Step 1. From \mathbf{X} , we extract the l -dimensional basis matrix, $\hat{\mathbf{U}}$, by SVD and calculate the feature matrices, \mathbf{F}_X and \mathbf{F}_Y , by Eq. (2).

Step 2. From \mathbf{F}_X and \mathbf{F}_Y , we train the LDA projection vector w and mean values, μ_X, μ_Y , and standard deviations, σ_X, σ_Y , for each group.

Step 3. For z_1, \dots, z_6 , we calculate their feature values by using SVD and LDA.

Step 4. We decide the target signal z_i with the minimum distance, d_{X_i} , as the target signal, z_X , then the system recognize the corresponding arrow as the detection result. We also collect other z s as non-target signals, z_Y^1, \dots, z_Y^5 .

Step 4.1. If d_{X_i} corresponding to z_X is located within the range of $\mu_X - \sigma_X$ and $\mu_X + \sigma_X$, we append z_X to \mathbf{X} .

Step 4.2. If d_{Y_i} corresponding to z_1, \dots, z_6 is located within the range of $\mu_Y - \sigma_Y$ and $\mu_Y + \sigma_Y$, then append the z_i to \mathbf{Y} .

Step 5. We then go back to step 1 and repeat the sequence for the next trial.

By the above algorithm, we can collect valid P300 trials from the test set and then retrain the detection model. The implemented P300 classifier based on our proposed algorithm, shown in Fig. 3, could work properly with a smaller training set and shorter calibration time.

III. EXPERIMENTS

For evaluating the performance of our algorithm, we carried out two kinds of experiments in order to compare its detection accuracies with those of an algorithm without self-labeling. The algorithm without self-labeling (no-labeling algorithm) also uses the same detection model without self-labeling. Instead, it merely detects the target signals from all of the test data by using the detection model from the initial training data.

We used the 2 largest singular values to derive basis vectors ($l = 2$) in both algorithms.

A. Training with Small Data Set

In the first experiment, to validate that self-labeling method can improve the detection accuracy, especially when the amount of training data is small, we performed the off-line P300 detection with various amounts of initial training data. The amount of training data (the number of trials) varied from 2 to 24 and the amount of test data was 72 for each subject. The initial training data are randomly selected from each subject's training data.

The detection accuracy for this experiment is shown in Table I, and their averaged result of all 4 subjects in Fig. 4. On average, as the amount of training data increased, the detection accuracies of the two methods also increased. Also the detection accuracies of both algorithms converged at a certain level when using plentiful amount of initial training data ($N > 12$).

However, when using small amount of initial training data, our algorithm could obtain higher accuracy than no-labeling algorithm. In particular, using only 2 initial training data, which takes 24.4 s and can be considered extremely small, our algorithm improved the detection accuracy from 75.70% to 84.78%.

Also, we compared the error bar of the accuracies when using only 2 training data. As shown in Fig. 5, our algorithm could obtain higher mean accuracy than the no-labeling algorithm for all subjects, and its standard deviation shows that the results from our algorithm are statistically significant.

From the viewpoint of calibration time, to reach 86.5% accuracy, the no-labeling algorithm needs at least 5 times as many initial training data as with our algorithm for subject 2. Since the time for collecting 1 trial is 12.2s in our system, our algorithm can reduce the calibration time as 97.6s in this case.

In order to show how our algorithm improves the performance, we plotted the accumulative detection accuracy in Fig 6(a). Fig. 6(a) shows the alteration of accumulative detection accuracies of the two methods when the 72 test trials for subject 2 are processed one by one using only 2 initial training data. Whenever the target signal is appended to the existing training data, we draw a vertical black line on the corresponding test trial. While the accuracy of the no-labeling algorithm gradually decreased during the test, our algorithm

TABLE I
DETECTION ACCURACY WITH THE SUBJECT'S OWN TRAINING DATA

Method	No-labeling						Self-Labeling						
	# of training characters	2	3	4	10	20	24	2	3	4	10	20	24
Subject 1	80.44%	89.35%	91.03%	94.00%	97.21%	97.22%	97.22%	94.85%	94.91%	95.72%	96.29%	96.52%	97.22%
Subject 2	77.19%	75.58%	76.19%	86.00%	87.03%	87.50%	87.50%	86.51%	88.90%	88.19%	88.90%	88.90%	88.89%
Subject 3	70.17%	75.23%	78.01%	79.00%	78.47%	79.12%	79.12%	77.74%	79.12%	82.06%	80.21%	80.90%	80.56%
Subject 4	75.00%	74.65%	76.00%	77.50%	80.12%	80.56%	80.56%	80.04%	80.91%	80.32%	80.44%	84.26%	87.50%
Average	75.70%	78.70%	80.49%	84.12%	85.78%	86.10%	86.10%	84.78%	85.12%	86.57%	86.46%	87.64%	88.54%

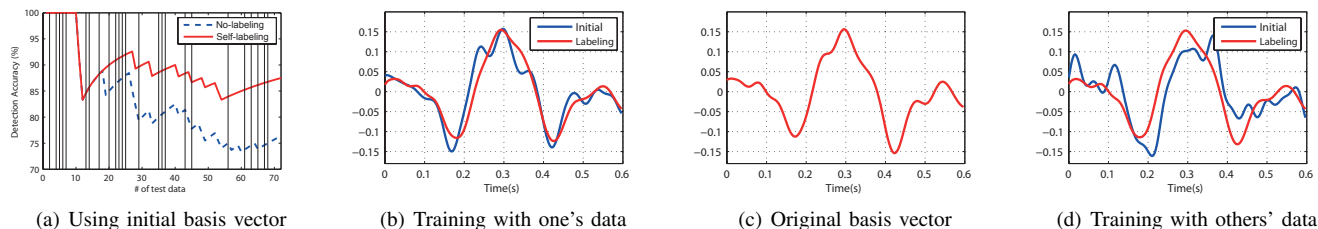


Fig. 6. Comparing the basis vectors

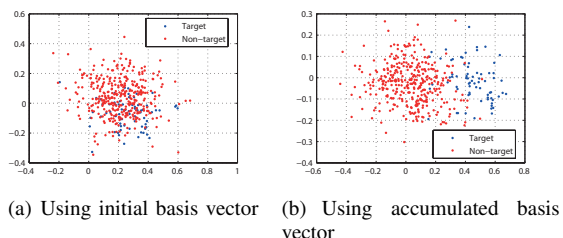


Fig. 7. Feature space

keep the accuracy from decreasing by retraining basis vectors.

To analyze the effect of our algorithm, we presented the basis vectors and feature space during the test procedure. Fig. 6(b) shows the most dominant basis vectors from the initial and the final training data for the result of Fig. 6(a). The basis vectors from the initial training data are not similar to the original basis vector, which is extracted from the subject 2's all training data as shown in Fig. 6(c), because of small amount of training data. However, our algorithm can retrain the basis vector to be similar to the original basis vector. Therefore our algorithm can extract more accurate basis vector for better detection accuracy.

Moreover, as the basis vector retrained, the feature space is also changed. Fig. 7 shows the 2-dimensional feature spaces of all test data, which are derived by the basis vectors from initial training data and accumulated training data by using our algorithm. As shown in Fig. 7, we can easily notice that the retrained basis vector, which has more similar shape with P300 waveform than the initial training data, obtained more separable feature space of test data than the initial basis vector. It means that our algorithm is able to extract a more separable feature space than the no-labeling algorithm with

TABLE II
DETECTION ACCURACY WITH OTHER SUBJECTS' TRAINING DATA

From	Other subjects		Subject Oneself
	no-labeling	Self-labeling	no-labeling
Subject 1	75.23%	88.90%	97.20%
Subject 2	79.26%	84.72%	87.50%
Subject 3	69.44%	73.61%	79.16%
Subject 4	66.42%	72.65%	80.01%
Average	72.58%	79.97%	85.96%

better accuracy.

B. Training with Other Subject's Data

In the first experiment, we proved that our algorithm outperformed than the no-labeling algorithm with small initial training data. However, the ultimate goal of this paper is detecting P300 without any calibration time, which means removing this burdensome task. In order to achieve this, we use the training data from other subjects for obtaining the detection model in this experiment. That is, we randomly select the signals from other subjects' training data and average them over subjects to make a general waveform. Suppose we want to classify the data of subject A with n training data. Then, n number of training data are chosen from the remaining 3 subjects respectively and averaged over all subjects. Finally, with n number of averaged signals as the initial training data, we classify the test data from subject A. In this experiment, we classified 72 test data for each subject, and used all of the training data from remaining subjects', which means the amount of initial training data was 24.

By using this subject transfer approach, we obtained the

accuracy of 75.23%, 79.26%, 69.44% and 66.42% for each subject, as shown in the first column of the Table II. Be compared to the result of ‘Subject Oneself’, which uses the training data from their own, we could see that the accuracies are deteriorated. This problem is due to the inter-subject pattern differences of P300 waveform, which is plotted in Fig. 6(d) and 6(c). Fig. 6(d) shows the basis vectors obtained from the initial training set collected from the subject 1, 3, and 4. Otherwise the Fig. 6(c) is the original basis vector obtained from the training data of the subject 2. They share similar N100 and P300 fluctuating patterns, but they are not exactly the same. To correct this inconsistency, we applied self-labeling algorithm to this subject transfer problem. Our approach selectively accumulated the P300 signals during the test procedure, which is similar to the target set from other subjects and also has subject 2’s own P300 wave characteristics. The final basis vector after all test procedure was also plotted in Fig. 6(d). We could see that the basis vector was retrained, which is much similar to the Fig. 6(c), the original waveform of subject 2’s P300. As a result, as shown in the second column of Table II, the accuracies were improved to 88.90%, 84.72%, 73.61% and 72.65% for each subject. Totally self-labeling method improved the accuracy by 7.39% ($\pm 3.70\%$). This result implies that our algorithm can be used for no-calibration P300 paradigm.

IV. CONCLUSIONS

In this paper, we have proposed a novel algorithm for a P300 based BCI system that can reduce the required calibration time with a small set of training data. Our algorithm uses singular value decomposition and linear discriminant analysis for feature extraction, then applies minimum distance classifier. For shortening the required calibration time, we selectively accumulate detected P300 to the training data and consecutively retrain the detection model, which is mentioned as ‘Self-labeling’. Experimental results suggest that our algorithm can obtain better detection accuracy with less amount of training data than ordinary algorithms. Furthermore, we also prove that it can be used for a BCI without the calibration. Therefore, our algorithm can shorten the calibration time with less deterioration of performance. For future work, we will implement the real-time BCI system using the proposed method to improve its usefulness on the on-line procedure. In addition to this, we will use the SVD more sophisticatedly, to extract the information about the which trial is close to the canonical P300 waveform. In order to increase the speed for the P300-based BCI, we plan on reducing the required number of sequences in a single trial. In our experiment, a single trial takes 12.2s for six sequences. If we could achieve the system with less sequences, then the speed of the system will be improved. In addition to this, although we just averaged signals of all channels to obtain the robust P300 ERP, if we exploit the channel information, the performance of the system may be increased.

ACKNOWLEDGMENTS

This work was supported by the National Research Foundation (NRF) of Korea (2012-0005785, 2012-0005786), MKE-

NIPA ‘IT Consilience Creative Program’ (C1515-1121-0003), and NRF World Class University Program (R31-10100).

REFERENCES

- [1] J. R. Wolpaw, N. Birbaumer, D. J. McFarland, G. Pfurtscheller, and T. M. Vaughan, “Brain-computer interfaces for communication and control,” *Clinical Neurophysiology*, vol. 113, pp. 767–791, 2002.
- [2] T. Ebrahimi, J. F. Vesin, and G. Garcia, “Brain-computer interface in multimedia communication,” *IEEE Signal Processing Magazine*, vol. 20, no. 1, pp. 14–24, Jan. 2003.
- [3] A. Cichocki, Y. Washizawa, T. Rutkowski, H. Bakardjian, A. H. Phan, S. Choi, H. Lee, Q. Zhao, L. Zhang, and Y. Li, “Noninvasive BCIs: Multiway signal-processing array decompositions,” *IEEE Computer*, vol. 41, no. 10, pp. 34–42, 2008.
- [4] S. Sutton, M. Braren, and J. Zubin, “Evoked-potential correlates of stimulus uncertainty,” *Science*, vol. 150, pp. 1187–1188, November 1965.
- [5] L. A. Farwell and E. Donchin, “Talking off the top of your head: Toward a mental prosthesis utilizing event-related brain potentials,” *Electroencephalography and Clinical Neurophysiology*, vol. 70, no. 6, pp. 510–523, 1988.
- [6] B. Rebsamen, E. Burdet, C. Guan, H. Zhang, C. L. Teo, Q. Zeng, M. Ang, and C. Laugier, “A brain-controlled wheelchair based on P300 and path guidance,” in *The First IEEE/RAS-EMBS International Conference on Biomedical Robotics and Biomechanics*, Pisa, Italy, 2006.
- [7] U. Hoffmann, J. M. Vesin, T. Ebrahimi, and K. Diserens, “An efficient P300-based brain-computer interface for disabled subjects,” *Journal of Neuroscience Methods*, vol. 167, no. 1, pp. 115–125, 2008.
- [8] F. Nijboer, E. W. Sellers, J. Mellinger, M. A. Jordan, T. Matuz, A. Furdea, S. Halder, U. Mochty, D. J. Krusienski, T. M. Vaughan, J. R. Wolpaw, N. Birbaumer, and A. Kublera, “A P300-based brain-computer interface for people with amyotrophic lateral sclerosis,” *Clinical Neurophysiology*, vol. 119, no. 8, pp. 1909–1916, 2008.
- [9] M. Kaper, P. Meinicke, U. Grossekhoefer, T. Lingner, and H. Ritter, “BCI competition 2003-data set Iib: Support vector machines for the P300 speller paradigm,” *IEEE Transactions on Biomedical Engineering*, vol. 51, no. 6, pp. 1073–1076, 2004.
- [10] J. Polich, “On the relationship between EEG and P300: Individual differences, aging, and ultradian rhythms,” *International Journal of Physiology*, vol. 26, pp. 299–317, 1997.
- [11] S. T. Ahi, N. Yoshimura, H. Kambara, and Y. Koike, “Utilizing fuzzy-SVM and a subject database to reduce the calibration time of P300-based BCI,” in *Proceedings of the International Conference on Neural Information Processing (ICONIP)*, Sydney, Australia, 2010.
- [12] I. Iturrate, L. Montesano, R. Chavarriga, J. del R. Millan, and J. Minguez, “Minimizing calibration time using inter-subject information of single-trial recognition of error potentials in brain-computer interfaces,” in *Proceedings of the Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, 2011.
- [13] F. Lotte and C. Guan, “An efficient P300-based brain-computer interface with minimal calibration time,” in *Proceeding of NIPS’09 Symposium on Assistive Machine Learning for People with Disabilities symposium*, 2009.
- [14] B. Rivet, H. Cecotti, M. Perrin, E. Maby, and J. Mattout, “Adaptive training session for a P300 speller brain-computer interface,” *Journal de Physique (Paris)*, vol. 105, no. 1-3, pp. 123–129, 2011.
- [15] R. C. Panicker, S. Puthusserypady, and Y. Sun, “Adaptation in P300 brain-computer interfaces: A two-classifier cotraining approach,” *IEEE Transactions on Biomedical Engineering*, vol. 57, no. 12, pp. 2927–2935, Dec. 2010.
- [16] G. Golub and W. Kahan, “Calculating the singular values and pseudo-inverse of a matrix,” *Journal of the Society for Industrial and Applied Mathematics: Series B, Numerical Analysis*, vol. 2, no. 2, pp. 205–224, 1965.
- [17] D. J. Krusienski, E. W. Sellers, F. Cabesting, S. Bayouth, D. J. McFarland, T. M. Vaughan, and J. R. Wolpaw, “A comparison of classification techniques for the P300 speller,” *Journal of Neural Engineering*, vol. 3, no. 4, pp. 188–305, 1990.
- [18] M. E. Hodgson, “Reducing the computational requirements of the minimum-distance classifier,” *Remote Sensing of Environment*, vol. 25, no. 1, pp. 117–128, 1988.