31st Annual International Conference of the IEEE EMBS
Minneapolis, Minnesota, USA, September 2-6, 2009

Supervised Adaptive Downsampling for P300-based Brain Computer Interface

Yuya Sakamoto and Masaki Aono

Abstract— **To realize Brain Computer Interface, a recording electroencephalogram (EEG) and determining whether or not P300 is evoked by the presented stimulus have become increasingly important. Using the machine learning method for this classification is effective, but constructing feature vectors with all data points might result in very high-dimensional data. Because such redundant features are undesirable from the viewpoint of computation and classification performance, EEG has been downsampled in several studies. In the present study, we propose a new downsampling method aiming at the improvement of P300 classification accuracy. In particular, each single trial EEG is segmented at non-uniform intervals and then averaged in each segment. The segmentation is decided in such a way that the degree of separating two classes from training data is increased by applying a time series segmentation algorithm. Our experiment using the BCI Competition III P300 Speller paradigm data set demonstrated that our method resulted in higher accuracy than traditional downsampling methods.**

I. INTRODUCTION

The P300 is a kind of positive EEG component, which is supposed to relate to cognition. It occurs as a response to rare target stimuli in a series of non-target stimuli (oddball paradigm). Farwell and Donchin proposed a character input Brain Computer Interface (BCI) system, which they called P300 Speller [4]. In this system, the user is presented a 6×6 matrix of characters as shown in Fig. 1, and focuses attention on the character he wishes to input. Since each row or column is flashed in a random sequence, the desired character is flashed twice out of 12 times (i.e., 10 times non-target stimuli presentation and twice target stimuli presentation). This task is regarded as an oddball paradigm, and the P300 component is evoked when the desired character is flashed. To realize the P300 Speller BCI system, we have to determine whether or not the P300 component is evoked. However, since the signal to noise ratio (SNR) of P300 is very low, classifying with only one flashing sequence may result in low accuracy. Therefore, the sequence has to be repeated several times to produce high accuracy. Another P300-based BCI such as wheelchair control [13] or appliance selection [8] were proposed, based on the same principle.

The machine learning algorithms have been applied for classifying P300 components, and a variety of methods have been investigated to improve classification accuracy. Farwell and Donchin used Stepwise Linear Discriminant Analysis (SWLDA) and considered the output values of the classifier as the score in the reference that proposed P300 Speller [4]. Similarly, Bayesian LDA [8], Gaussian kernel Support Vector Machine (SVM) [9], or ensemble learning approach have been applied [7][14]. Krusienski et al. compared various classification techniques regarding P300, and showed that SWLDA and Fisher Discriminant Analysis (FDA) were superior to other methods including SVM [10].

Y. Sakamoto is with the Department of Information and Computer Sciences, Toyohashi University of Technology, Toyohashi, Aichi 441-8580, JAPAN sakamoto@kde.ics.tut.ac.jp

M. Aono is with the Department of Information and Computer Sciences, Toyohashi University of Technology, Toyohashi, Aichi 441-8580, JAPAN aono@kde.ics.tut.ac.jp

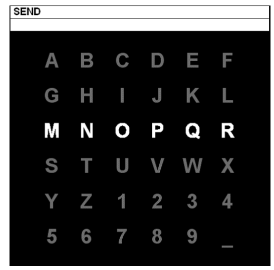


Fig. 1. P300 Speller display

To classify P300, it is customary to define a "feature vector." The feature vector, in this line of research, is usually constructed by concatenating EEG time series data acquired from each electrode. There have been several other feature vector construction methods proposed. Examples include applying Mexican hat wavelet [3], Common Spatial Pattern [11], and subspace approach [15]. However, constructing feature vectors with all sampled data points might result in a very high-dimensional vector, which might make us suffer from the curse of dimensionality. To avoid these problems, many investigators have applied downsampling to reduce the dimension of feature vectors. Typically, two approaches have been primarily used, a decimator approach [7][14], and a downsampling approach with uniform interval segments [12]. In the current study, we propose a new supervised adaptive downsampling method to improve the accuracy of P300-based BCI classification. With this approach, we perform segmentation at non-uniform intervals as illustrated in Fig. 2 and take the average in each segment. The idea behind this strategy is based on our assumptions that effective downsampling for P300-based BCI should make for clear differences between two classes, and at the same time the degree of separation between two classes should become larger. The outline of the rest of the paper is as follows. In Section II, we elaborate our new downsampling method. In Section III, the processing flow and P300 Speller classification methods are described. Section IV presents evaluation experiments and their results, and Section V concludes the paper.

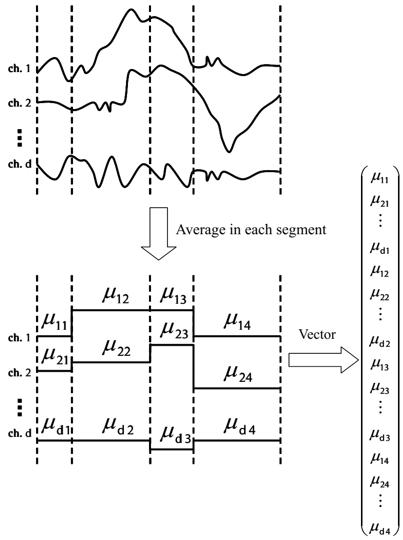


Fig. 2. Adaptive downsampling and feature vector generation

II. SUPERVISED ADAPTIVE DOWNSAMPLING

The downsampling we propose in this paper is a method of dividing the raw time series EEG data adaptively into segments with non-uniform intervals, followed by taking the average for each interval in order to approximate the original data with reduced samples. Our main idea is to determine this interval in a supervised fashion with training data in advance, so that the degree of separation between two classes will become maximum. In the following, we describe the measure of separation between classes, the formulation of our problem, and the algorithm that we developed.

A. Measure of separation between classes

To separate classes, we need a measure of similarity or dissimilarity. In this research, we employ the Fisher Discriminant Criterion (FDC) [5] for this measure. FDC has been used in Fisher Discriminant Analysis (FDA) as one of the linear discriminant analysis methods. FDA is a method of supervised dimensionality reduction, which is a linear mapping that maximizes the degree of separation between classes, based on training data. In FDA, we provide a mapping vector that maximizes the ratio of the squared average difference between classes to the addition of variance between classes. We first define the average vectors.

$$\mathbf{m}_0 = \frac{1}{N_0}\sum_{i:y_i=0}\mathbf{x}_i \ , \ \mathbf{m}_1 = \frac{1}{N_1}\sum_{i:y_i=1}\mathbf{x}_i, \tag{1}$$

where N_0 and N_1 represent the number of training data for each class. Next, we define two variances. Specifically, let the within-class covariance denote

$$\mathbf{W} = \sum_{l=0}^{1}\sum_{i:y_i=l}(\mathbf{x}_i - \mathbf{m}_l)(\mathbf{x}_i - \mathbf{m}_l)^T, \tag{2}$$

while the between-class covariance is denoted by

$$\mathbf{B} = (\mathbf{m}_0 - \mathbf{m}_1)(\mathbf{m}_0 - \mathbf{m}_1)^T. \tag{3}$$

Then, the FDA mapping vector is given by a vector that maximizes the variable denoted by "$\mathbf{a}$" in the following expression:

$$J(\mathbf{a}) = \frac{\mathbf{a}^T\mathbf{B}\mathbf{a}}{\mathbf{a}^T\mathbf{W}\mathbf{a}} \tag{4}$$

The $J(\mathbf{a})$ in this expression is a FDC. The "$\mathbf{a}$" vector for two-class discrimination can be represented by an expression in proportion to some amount of the right-hand formula as below:

$$\mathbf{a} \propto \mathbf{W}^{-1}(\mathbf{m}_0 - \mathbf{m}_1) \tag{5}$$

By substituting equation (5) into equation (4), we obtain a Fisher Discriminant Criterion that we will use in this research:

$$J = (\mathbf{m}_0 - \mathbf{m}_1)^T\mathbf{W}^{-1}(\mathbf{m}_0 - \mathbf{m}_1) \tag{6}$$

As we noted earlier, FDC represents the degree of separation between classes after mapping into lower-dimensional spaces. It is therefore natural to expect FDC to be a measure of separation even in the original higher dimensional spaces.

B. Formulation

Now we formulate the problem of obtaining the "optimal segment." Let $E = \{(\mathbf{S}_i, y_i)|i = 1, \cdots, N\}$ be training samples, where $y_i \in \{0, 1\}$ is a label and $\mathbf{S}_i$ represents EEG data or multi-dimensional time series data. Thus, if a single trial EEG consists of n data points, $\mathbf{S}_i$ is defined as $\mathbf{S}_i = \{\mathbf{s}_i(1), \mathbf{s}_i(2), \cdots, \mathbf{s}_i(n)\}$, where $\mathbf{s}_i(t) \in \mathbf{R}^d$, and d is the number of channels. Assume that the number of segments is denoted by k, the breakpoints between segments can be defined as
$\mathcal{T} = (\tau_0, \tau_1, \cdots, \tau_k)$, where $\tau_j \in \mathbf{N}$, $\tau_j < \tau_{j+1}$, $\tau_0 = 0$, and $\tau_k = n$.

We define a $\mathit{MakeVector}(\mathbf{S}_i; \mathcal{T})$ function that downsamples an $\mathbf{S}_i$ and converts it into a vector:

$$\begin{aligned}\mathbf{x}_i(\mathcal{T}) &= \mathit{MakeVector}(\mathbf{S}_i; \mathcal{T}) \\ &= \begin{pmatrix} \frac{1}{\tau_1-\tau_0}\sum_{t=\tau_0+1}^{\tau_1}\mathbf{s}_i(t) \\ \frac{1}{\tau_2-\tau_1}\sum_{t=\tau_1+1}^{\tau_2}\mathbf{s}_i(t) \\ \vdots \\ \frac{1}{\tau_k-\tau_{k-1}}\sum_{t=\tau_{k-1}+1}^{\tau_k}\mathbf{s}_i(t) \end{pmatrix}\end{aligned} \tag{7}$$

$\mathbf{x}_i$ obtained by this function is a $d \times k$ dimensional vector. Consider the training data set $\tilde{E}(\mathcal{T}) = \{(\mathbf{x}_i(\mathcal{T}), y_i)|i = 1, \cdots, N\}$, which is obtained by the transformation defined by equation (7).

Since $\mathbf{m}_0$, $\mathbf{m}_1$, and $\mathbf{W}$ which we define previously are generated by $\tilde{E}(\mathcal{T})$, they are all functions of $\mathcal{T}$. The degree of separation between two classes based on feature vectors generated by $\mathcal{T}$ is rephrased by equation (6) as follows:

$$J(\mathcal{T}) = (\mathbf{m}_0(\mathcal{T}) - \mathbf{m}_1(\mathcal{T}))^T\mathbf{W}(\mathcal{T})^{-1}(\mathbf{m}_0(\mathcal{T}) - \mathbf{m}_1(\mathcal{T})) \tag{8}$$

Finally, the problem of finding "segments" that maximize the degree of separation between classes is reduced to finding $\hat{\mathcal{T}}$, which is given by the following formula:

$$\hat{\mathcal{T}} = \underset{\mathcal{T}}{\operatorname{argmax}}\, J(\mathcal{T}) = \underset{\tau_0,\tau_1,\cdots,\tau_k}{\operatorname{argmax}}\; J(\tau_0,\tau_1,\cdots,\tau_k)$$
$$\text{subject to } \tau_j \in \mathbf{N},\ \tau_j < \tau_{j+1},\ \tau_0 = 0,\ \tau_k = n \quad (9)$$

C. Algorithm

To solve the equation (9), we seek the optimal $\hat{\mathcal{T}}$ by modifying Local Iterative Replacement (LIR) [6] into our time series data segmentation.

In LIR, we first provide the initial value for $\mathcal{T}$, and then seek the optimal solution. LIR consists of repeated computation of the selection of boundary between segments, and the search of new boundary location before or after the selected boundary, until we obtain a solution that satisfies equation (9), which is assured to be a local optimum. The convergence of LIR is described in [6]. The pseudo code of the algorithm is given below:

Algorithm LIR(E,k)
Input E: training set, k: segment number
Return $\hat{\mathcal{T}}$: optimal boundary location

```
Initialize boundary T = (τ0, τ1, ··· , τk). Note that τ0 =
  0 and τk = n are always fixed.
repeat
  Substitute T into T^(pre)
  for i = 1 to k − 1 do
    for r = τ(i−1) + 1 to τ(i+1) − 1 do
      Assign T_r to be T except that i-th element (τi)
        is replaced by r
      Transform E into Ẽ(T_r)
      Save J(T_r) and T_r
    end for
    T = argmax_{T_r} J(T_r)
    Reset J(T_r) and T_r
  end for
until T^(pre) == T
T̂ = T
```

III. P300 SPELLER CLASSIFICATION METHOD

In this section, we describe a method of learning and P300 Speller classification using feature vectors. For classification, we employ logistic regression. In logistic regression, assume that we are given a vector $\mathbf{x}$, the probability of a data item belonging to a class "1" is represented by the following:

$$p(y=1|\mathbf{x}) = \sigma(\psi(\mathbf{x})) = \frac{1}{1+e^{-\psi(\mathbf{x})}}, \quad (10)$$

where

$$\psi(\mathbf{x}) = \boldsymbol{\theta}^T\mathbf{x} + \theta_0. \quad (11)$$

From training set $\{(\mathbf{x}_i, y_i)|i = 1,\cdots,N\}$, we let the system learn $\boldsymbol{\theta}, \theta_0$ of equation (11) so that equation (10) produces the correct probability.

$$[\boldsymbol{\theta}, \theta_0] = \underset{\boldsymbol{\theta},\theta_0}{\operatorname{argmin}} \left(-\sum_{i=1}^{N} \ln p(y_i|\mathbf{x}_i) \right) \quad (12)$$

That is, the problem is reduced to finding $\boldsymbol{\theta}$ and θ_0.

In our current task, we let the system learn "1" if there is a letter that a subject is thinking of and would like to convey to the system, which is included in the flashed rows; otherwise let the system learn "0".

When we input the feature vectors of testing data into the equation (10), in which the learning phase has been done, the system outputs the probability of the class being "1", i.e., the probability of the target letter exposed to the user as a stimulus.

Since the main task of P300 Speller is to predict the letter which a subject is thinking of, we have to determine the letter using our discriminant criterion. Here, we have adopted the majority consensus method using the score of the discriminant function, analogous to the methods proposed by previous researchers such as [9] and [12]. Specifically, by using equation (11), we define the following expressions:

$$\hat{r}(RepNum) = \underset{r}{\operatorname{argmax}} \left(\sum_{rep=1}^{RepNum} \psi(\mathbf{x}_{rep}^{row}(r)) \right) \quad (13)$$

$$\hat{c}(RepNum) = \underset{c}{\operatorname{argmax}} \left(\sum_{rep=1}^{RepNum} \psi(\mathbf{x}_{rep}^{column}(c)) \right) \quad (14)$$

Here, $RepNum$ denotes the number of repetitions, $r = \{1,2,3,4,5,6\}$ are the row numbers, and $c = \{1,2,3,4,5,6\}$ are the column numbers. $\mathbf{x}_{rep}^{row}(r)$ and $\mathbf{x}_{rep}^{column}(c)$ represent the feature vectors when r-th row (c-th column) is flashed and the number of repetitions is rep.

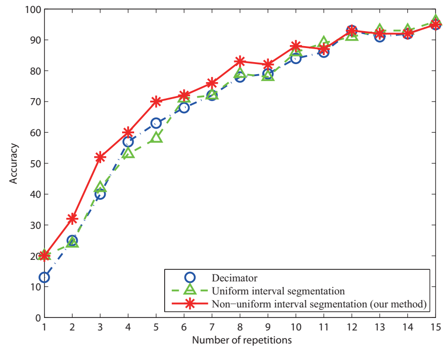
IV. PERFORMANCE EVALUATION

A. Experimental Settings

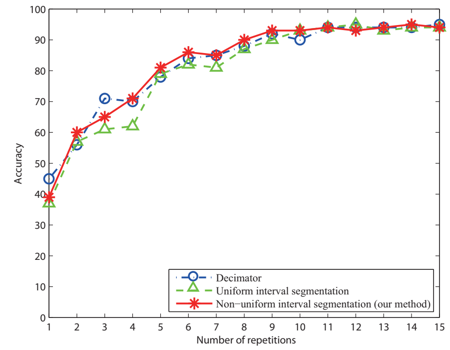
To evaluate the classification accuracy of our method, we employed a data set, "BCI Competition III dataset II P300 Speller paradigm" [1][2]. This data set consists of training data with predefined labels and testing data with no labels. BCI Competition III was a task to predict the input letter of a subject from EEG time series data. The sampling frequency of the original data set was 240 Hz, and 64 electrodes were used to cover different locations of the human head. The period of a stimulus was 175 ms and the number of repetitions per letter was 15 times at maximum. The idea is to predict letters with as few repetitions as possible and as precisely as possible.

In experiments, we extracted EEG data during 667 ms from the time a row or a column in the P300 Speller display was flashed. Since the interval between flashes was 175 ms, there were overlaps between adjacent flashing. Before our downsampling algorithm was applied, we normalized the value into the range $[-1, 1]$ per electrode, and applied a band-path filter with 0.1 Hz-20 Hz. We employed the fourth-order type-one Chebyshef filter.

For comparison, we also conducted experiments using downsampling with a decimator, and downsampling with uniform interval segmentation.



(a) Subject A, No. of segments: 9



(b) Subject B, No. of segments: 9

Fig. 3. Comparative accuracy of our downsampling method with other methods. The vertical axis represents the percentage of correctly recognized characters under different number of repetitions.

TABLE I
COMPARISON BETWEEN OURS AND BCI COMPETITION III WINNER

Algorithm	No. of segments	5 Rep	15 Rep
winning method	-	73.5	96.5
Our Algorithm	7	72.5	93.0
	8	**74.5**	93.0
	9	**75.5**	94.5
	10	**75.5**	94.5
	11	**74.0**	**96.5**

B. Results

Fig. 3 shows the accuracy of classification with our methods. For subject A, our method turned out to be very effective if the number of repetitions of the flashing was small, while for subject B, our method was almost equal to or slightly better than previous methods such as a decimator and a method with uniform interval downsampling. The comparison between our method and the winner of the BCI Competition III is summarized in Table I. It is demonstrated that our methods outperform the method by Rakotomamonjy (winner) [14], if the repetition number is 5 except when the number of segments is 7, and if the repetition number is 15 and the number of segments is 11.

V. CONCLUSION

We proposed a new supervised adaptive downsampling method and evaluated our method using BCI Competition III P300 Speller Test data. We demonstrated that our method outperformed a decimator method and a downsampling method with uniform interval segments. In addition, by using a single discriminator based on logistic regression, we showed that our system was superior to the winner of the BCI III Competition, who used multiple SVMs. It would be possible to incorporate our method into a sophisticated ensemble discriminator, since our method relies on downsampling alone.

REFERENCES

[1] B. Blankertz, "BCI Competition III," http://ida.first.fraunhofer.de/projects/bci/competition_iii/, 2004
[2] B. Blankertz, K.-R. Mueller, D. Krusienski, G. Schalk, J. Wolpaw, A. Schloegl, G. Pfurtscheller, J. del R. Millan, M. Schroeder, N. Birbaumer, "The BCI competition iii: Validating alternative approaches to actual bci problems," IEEE Transactions on Neural Systems and Rehabilitation Engineering, Vol.14, No.2, pp.153-159, June 2006.
[3] V. Bostanov, "BCI competition 2003-data sets Ib and IIb: feature extraction from event-related brain potentials with the continuous wavelet transform and the t-value scalogram," IEEE Transactions on Biomedical Engineering, Vol.51, No.6, pp.1057-1061, June 2004
[4] L. A. Farwell, 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, December 1988.
[5] D. H. Foley, J. W. Sammon Jr., "An Optimal Set of Discriminant Vectors," IEEE Transactions on Computers, Vol.24, No.3, pp.281-289, March 1975.
[6] J. Himberg, K. Kalle, H. Mannila, J. Tikanmaki, "Time series segmentation for context recognition in mobile devices," Proc. IEEE International Conference on Data Mining, pp.203-210, December 2001
[7] U. Hoffmann, G. Garcia, J. M. Vesin, K. Diserens, T. Ebrahimi, "A Boosting Approach to P300 Detection with Application to Brain-Computer Interfaces," Proc. 2nd International IEEE EMBS Conference on Neural Engineering, pp.97-100, March 2005
[8] U. Hoffmann, J. M. Vesin, T. Ebrahimi, K. Diserens, "An efficient P300-based brain-computer interface for disabled subjects," Journal of Neuroscience Methods, vol.167, No.1, pp.115-125, January 2008
[9] M. Kaper, P. Meinicke, U. Grossekathoefer, T. Lingner, 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, June 2004
[10] D. J. Krusienski, E. W. Sellers, F. Cabestaing, S. Bayoudh, D. J. McFarland, T. M. Vaughan, J. R. Wolpaw, "A comparison of classification techniques for the P300 Speller," Journal of Neural Engineering, Vol.3, No.4, pp.299-305, December 2006
[11] D. J. Krusienski, E. W. Sellers, T. M. Vaughan, "Common Spatio-Temporal Ptterns for the P300 Speller," Proc. 3rd International IEEE EMBS Conference on Neural Engineering, pp.421-424, May 2007
[12] D. J. Krusienski, E. W. Sellers, D. J. McFarland, T. M. Vaughan, J. R. Wolpaw, "Toward enhanced P300 speller performance," Journal of Neuroscience Methods, Vol.167, No.1, pp.15-21, January 2008
[13] G. Pires, M. Castelo-Branco, U. Nunes, "Visual P300-based BCI to steer a wheelchair:A Bayesian approach," Proc. 30th Annual Int. Conference of the IEEE EMBS, pp.658-661, August 2008
[14] A. Rakotomamonjy, V. Guigue, "BCI Competition III: Dataset II - Ensemble of SVMs for BCI P300 Speller," IEEE Transactions on Biomedical Engineering, Vol.55, No.3, pp.1147-1154, March 2008
[15] B. Rivet, A. Souloumiac, "Subspace estimation approach to P300 detection and application to Brain-Computer Interface," Proc. 29th Annual International Conference of the IEEE EMBS, pp.5071-5074, August 2007