

# Bayesian Method for Continuous Cursor Control in EEG-Based Brain-Computer Interface

Xiaoyuan Zhu<sup>1,2</sup>, Cuntai Guan<sup>1</sup>, Jiankang Wu<sup>1</sup>, Yimin Cheng<sup>2</sup> and Yixiao Wang<sup>2</sup>

<sup>1</sup>Institute for Infocomm Research, 21 Heng Mui Keng Terrace, Singapore 119613

<sup>2</sup>Department of Electronic Science and Technology, USTC, Hefei, Anhui, China 230026

E-mail: xiaoyuan\_zhu@ustc.edu; {ctguan, jiankang}@i2r.a-star.edu.sg; {ymcheng, yxwang}@ustc.edu.cn

**Abstract**-To develop effective learning algorithms for continuous prediction of cursor movement using EEG signals is a challenging research issue in Brain-Computer Interface (BCI). To train a classifier for continuous prediction, trials in training dataset are first divided into segments. The difficulty is that the actual intention (label) at each time interval (segment) is unknown. In this paper, we propose a novel statistical approach under Bayesian learning framework to learn the parameters of a classifier. To make use of all the training dataset, we iteratively estimate probability of the unknown label, and use this probability to assist the training process. Experimental results have shown that the performance of the proposed method is equal to or better than the best results so far.

## I. INTRODUCTION

Brain-Computer Interface (BCI) provides a new communication channel to the existing human-computer interface [1]. Electroencephalogram (EEG) is the most prevailing signal for non-invasive BCIs. One of the problems researchers are trying to solve is the cursor movement controlled by EEG signal. The methods for cursor control BCI come under two categories, regression [2,3] and classification [4,5,6]. Each of them has its merits. Here, we adopt classification method to continuously predict cursor movement (up or down in one dimensional (1-D) case as discussed here) using EEG signal. To train the classifier, we divide each trial into segments. Now the key issue is how to train a classifier, with given training dataset where there is no knowledge of actual intended cursor movement at any time intervals during a trial. In other words, for each trial, although the final target label of the trial is known, the true label of each segment is unknown, which imposes great difficulties in the classifier training. In this paper, we denote this issue as "Unlabeled Problem".

In [4] Roberts et al. extracted features using an AR model and classified them into cursor movement. Since

their method is used in the 1-D two targets cursor control scenario, they can label the training dataset, and use standard Bayesian learning method to train the classifier. In 1-D four targets cursor control scenario, which we are discussing here, Cheng et al. [5] proposed an offline method to classify the cursor target position, and reported results on BCI Competition 2003 cursor control Dataset 2a [7]. Blanchard et al. [6] described both continuous method and offline method using Common Spatial Pattern (CSP) for feature extraction, using Fisher discriminant (continuous method) and regularized linear discriminant (offline method) for classification. The results derived from their method won the BCI competition 2003 for cursor control dataset 2a. As proposed in [6], a simple solution to solve the unlabeled problem for continuously predicting cursor movement is to only use trials of top and bottom targets for training the classifier. In this case, they can further label the training dataset by assuming that in trials of top target the subject would try to make the cursor always go up. Then, they can use the classifier trained using partial training dataset (top and bottom targets) to perform 4 targets cursor control. As we have seen from above discussion, [4] and [6] simplified the unlabeled problem by reducing the number of labels from 2 (up and down) to 1 (either up or down depending on the target). We feel that the simplification in [4] and [6] are done at trial level, while the actual cursor control is carried out at finer time interval. For target on top, although the cursor has to go up to reach the final target, it is not necessarily true that the cursor always goes up at all time intervals.

In this paper, we propose a statistical learning method, under Bayesian learning framework. First we divide the training dataset into segments whose labels are not known, and then represent the training dataset by assigning a probability to the possible movement (the label of the segment) at each time interval. To exploit

our prior knowledge of the segment label, we propose to use an estimated probability of the label to assist the training process. During the training process, this probability is updated iteratively. We test our method on the BCI Competition 2003 Cursor Control Dataset 2a. Our result is better than other most recent algorithms on averaged accuracy over different subjects. The experiment results demonstrate that the proposed method can make full use of the training dataset.

## II. THE PROPOSED LEARNING ALGORITHM FOR CONTINUOUS PREDICTION

In this section, we derive the proposed learning algorithm under Bayesian learning framework. First we formulate the learning problem as follows. Let  $D = \{x_i, z_i\}_{i=1}^{N_D}$  be the training dataset consisting of  $N_D$  independent and identically distributed (i.i.d.) observations, where  $x_i$  denotes the  $i^{\text{th}}$  trial and  $z_i$  denotes the target label of the  $i^{\text{th}}$  trial. For continuously predicting the cursor movement, each trial is divided into certain number of segments. Let  $x_i = \{x_{i1}, \dots, x_{ij}, \dots, x_{iJ}\}$ , where  $x_{ij}$  denotes the  $j^{\text{th}}$  segment of the  $i^{\text{th}}$  trial and  $J$  is the total number of the segments in a trial. In this learning problem the label of each segment  $x_{ij}$  is not known exactly, although we have some prior knowledge from the target label of  $x_i$ . Let  $y_{ij} \in \Phi$  be the true label of  $x_{ij}$ .  $\Phi$  is the given segment's label set. Let  $\theta$  denotes the parameter of the classifier that maps the feature space of  $x_{ij}$  into the label set  $\Phi$ .

Based on the Bayesian theorem, we can estimate  $\theta$  under MAP criterion:

$$\begin{aligned} \max_{\theta} p(\theta | D) &= \max_{\theta} \left[ \frac{p(D | \theta)p(\theta)}{p(D)} \right] \\ &= \max_{\theta} [p(D | \theta)p(\theta)] \end{aligned} \quad (1)$$

For the sake of simplicity, we assume that  $y_{ij}$  are independent given corresponding segment  $x_{ij}$ . Let  $P(y|x_{ij}, \theta)$  be the probability of label of segment  $x_{ij}$  and  $y \in \Phi$ . Then the posterior  $p(\theta|D)$  can be formulated as follows (strictly we only model the probability of  $\{y_{ij}\}_{i=1, \dots, N_D; j=1, \dots, J}$  as suggested in [10], which is a Conditional Bayesian inference defined in [11]):

$$p(\theta | D) = \frac{1}{Z_D} \prod_{i,j} P(y = y_{ij} | x_{ij}, \theta) p(\theta) \quad (2)$$

where  $Z_D$  is the normalization factor.

However, in the learning problem described above, we do not know the true value of the segments' labels. As an approximation, we propose to use a weighted average to estimate  $P(y=y_{ij}|x_{ij}, \theta)$ .

We define the following expectation:

$$I_{\hat{P}} = E_{\hat{P}(y|x_{ij})} [\ln P(y | x_{ij}, \theta)]$$

where  $\hat{P}(y | x_{ij})$  is the estimated probability of the label of  $x_{ij}$ . (the update procedure of  $\hat{P}(y | x_{ij})$  will be discussed later) It is obvious that  $\exp(I_{\hat{P}}) \in [0, 1]$ . Furthermore, we can see later that under the definition of  $I_{\hat{P}}$ , the given probability  $\hat{P}(y | x_{ij})$  plays a central role in the proposed learning criterion.

Using  $\exp(I_{\hat{P}})$  instead of  $P(y=y_{ij}|x_{ij}, \theta)$  in (2), we obtain:

$$\hat{p}(\theta | D) = \frac{1}{\tilde{Z}_D} \prod_{i,j} \exp(I_{\hat{P}}) p(\theta) \quad (3)$$

where  $\tilde{Z}_D$  is the normalization factor.

Furthermore, as suggested in [10], the prior  $p(\theta)$  is modeled as the Gaussian distribution  $p(\theta) = N[0, \alpha^{-1}I]$ , where  $\alpha$  is the precision parameter. Then  $\theta$  is estimated via maximizing the logarithm of  $\hat{p}(\theta | D)$  as follows:

$$\begin{aligned} \theta^* &= \arg \max_{\theta} \{\ln \hat{p}(\theta | D)\} \\ &= \arg \min_{\theta} \left\{ \sum_{i,j} \sum_{y \in \Phi} \hat{P}(y | x_{ij}) \ln \frac{1}{P(y | x_{ij}, \theta)} + \frac{\alpha}{2} \theta^T \theta \right\} \end{aligned}$$

where the precision parameter  $\alpha$  here acts as a regularization constant. It can be estimated as part of the Bayesian Learning paradigm using a second level of Bayesian inference [8]. To reveal the characteristics of the proposed criterion, since  $\hat{P}(y | x_{ij})$  is independent on  $\theta$ , we rewrite the above criterion as follows:

$$\theta^* = \arg \min_{\theta} \left\{ \sum_{i,j} \sum_{y \in \Phi} \hat{P}(y | x_{ij}) \ln \frac{\hat{P}(y | x_{ij})}{P(y | x_{ij}, \theta)} + \frac{\alpha}{2} \theta^T \theta \right\} \quad (4)$$

From the proposed criterion we can see that  $\hat{P}(y | x_{ij})$  is used to supervise the optimization process by minimizing the Kullback-Leibler distance between  $\hat{P}(y | x_{ij})$  and  $P(y|x_{ij}, \theta)$ , which will let  $P(y|x_{ij}, \theta)$  close to  $\hat{P}(y | x_{ij})$ . In the proposed method  $\hat{P}(y | x_{ij})$  is initialized using the prior knowledge of the segment label. In the optimization process, we update  $\hat{P}(y | x_{ij})$  iteratively to transform the prior into posterior by extracting information from the dataset. Furthermore, we can see that if we set  $\hat{P}(y | x_{ij})$  to be the delta function  $\delta(y, y_{ij})$ , where  $y_{ij}$  is the true label of  $x_{ij}$ , then the proposed criterion (4) will degenerate to:

$$\begin{aligned} \theta^* = \arg \min_{\theta} \{ & -\sum_{ij} [\delta(y, y_{ij}) \ln P(y_{ij} | x_{ij}, \theta) \\ & + (1 - \delta(y, y_{ij})) \ln(1 - P(y_{ij} | x_{ij}, \theta))] + \frac{\alpha}{2} \theta^T \theta \} \end{aligned} \quad (5)$$

This criterion has been used in [4]. To show the improvement of the proposed algorithm by fully using the dataset, we take criterion (5) as baseline.

To derive the algorithm from the above framework, we have to model  $P(y|x_{ij}, \theta)$  first. For simplicity let us model  $P(y|x_{ij}, \theta)$  in the binary case. Let  $\Phi \in \{C_1, C_0\}$  denote the segment label set, where  $C_1$  and  $C_0$  stand for the two classes. The probability  $P(y|x_{ij}, \theta)$  can be model as follows:

$$P(C_1 | x_{ij}, \theta) = \frac{1}{1 + \exp(-\theta^T x_{ij})} = g(\theta^T x_{ij}) = g(a) \quad (6)$$

$$P(C_0 | x_{ij}, \theta) = 1 - P(C_1 | x_{ij}, \theta)$$

where  $a$  denotes  $\theta^T x_{ij}$ . It should be noted that, without confusion, we use  $x_{ij}$  to denote the feature extracted from the  $j^{\text{th}}$  segment of the  $i^{\text{th}}$  trial when operating with other variables in a function. The motivation for using this form of discriminant has been discussed in [10].

However, if  $P(y|x_{ij}, \theta)$  is used as a classifier directly, the estimation error of  $\theta$  will make the output of the classifier overconfident. To solve this problem, as suggested in [9], the following equation is used as our classifier:

$$P(C_1 | x_{ij}, D) = g(k(\sigma^2(x_{ij}))a^*(x_{ij})) \quad (7)$$

where  $a^*(x_{ij}) = \langle \theta^*, x_{ij} \rangle$ ,  $\langle \cdot, \cdot \rangle$  denotes the dot product.  $k(\sigma^2) = (1 + \pi\sigma^2/8)^{-1/2}$ ,  $\sigma^2 = x_{ij}^T H^{-1} x_{ij}$ ,  $H$  is Hessian matrix.

It is worth mentioning that in the aforementioned learning problem, the segment label of  $x_{ij}$  is not known, so it is difficult to set  $\hat{P}(y | x_{ij})$  individually for each segment  $x_{ij}$ . Thus we set  $\hat{P}(y | x_{ij})$  with the same value for all  $x_{ij}$   $s$  which belong to the same trial.

To update  $\hat{P}(y | x_{ij})$  from training dataset, we minimize the KL distance  $D_{\text{KL}}(\hat{P} \| P)$  with respect to  $\hat{P}$  under constrain  $\sum_{y \in \Phi} \hat{P}(y | x_{ij}) = 1$ .

Then the optimization algorithm can be summarized as follows. First we estimate  $\theta^{(t+1)}$  using the following criterion:

$$\theta^{(t+1)} = \arg \min_{\theta} \left\{ \sum_{i,j} \sum_{y \in \Phi} \hat{P}^{(t)}(y | x_{ij}) \ln \frac{\hat{P}^{(t)}(y | x_{ij})}{P(y | x_{ij}, \theta)} + \frac{\alpha}{2} \theta^T \theta \right\}$$

Then, we update  $\hat{P}^{(t+1)}$  as follows:

$$\begin{aligned} \hat{P}^{(t+1)}(C_1 | x_{ij}) &= \frac{\sqrt{\prod_{m \in \{C_1, C_0\}} \prod_{n=1}^N P(C_1 | x_{nm}, \theta^{(t+1)})}}{\sqrt{\prod_{m \in \{C_1, C_0\}} \prod_{n=1}^N P(C_1 | x_{nm}, \theta^{(t+1)})} + \sqrt{\prod_{m \in \{C_1, C_0\}} \prod_{n=1}^N P(C_0 | x_{nm}, \theta^{(t+1)})}} \\ \hat{P}^{(t+1)}(C_0 | x_{ij}) &= 1 - \hat{P}^{(t+1)}(C_1 | x_{ij}) \end{aligned}$$

where  $N$  is the total number of segments belong to the trials whose target labels are the same.

We perform these two iterative steps to estimate the parameter vector  $\theta$ . It can be proved that the algorithm is convergent (The proof is beyond the scope of this paper and therefore not provided here for the sake of brevity).

### III. IMPLEMENTATION OF THE PROPOSED ALGORITHM TO CURSOR CONTROL PROBLEM

In this paper we evaluate the proposed algorithm in cursor control problem, specifically for 1-D control with four targets based on Mu/Beta rhythms. First we introduce the scenario of 1-D four targets cursor control BCI. There are four targets on the right side of the screen. Targets 1 to 4 are aligned from top to bottom position. The starting position of the cursor is on the middle of the left side screen. The subject's task is to move the cursor to the pre-decided target using their EEG signal. The cursor was moved every hundreds of milliseconds. Our aim is to use the classifier to continuously predict the cursor movement at each time interval as well as the final target.

To perform the continuous cursor control, first we extract the power features centralizing on the alpha and beta bands, from the EEG data. Then we learn the parameter  $\theta^*$  of the classifier and use (7) to carry out the prediction of the cursor movement at each control step. Let  $d_{ij}$  denotes the displacement of cursor movement at the  $j^{\text{th}}$  time interval of the  $i^{\text{th}}$  trial, we have:

$$d_{ij} = \frac{1}{J} [P(C_1 | x_{ij}, D) - \frac{1}{2}] \quad (8)$$

where  $J$  is the total number of the segments of the  $i^{\text{th}}$  trial. Then we formulate the vertical displacement  $D_{ij}$  between the middle line of the screen and the cursor at the  $j^{\text{th}}$  time interval of the  $i^{\text{th}}$  trial as follows:

$$D_{ij} = \frac{1}{J} \sum_{k=1}^j [P(C_1 | x_{ik}, D) - \frac{1}{2}] \quad (9)$$

To evaluate the performance of our algorithm, we choose three thresholds  $t_3 < t_2 < t_1$  to classify the final distance  $D_{ij}$  controlled by  $x_i$  into four categories, such that trial  $x_i$  belongs to target 1, if  $t_1 < D_{ij}$ , and trial  $x_i$  belongs to target 2, if  $t_2 < D_{ij} < t_1$ , and so on.

TABLE I

A COMPARISON OF THE AVERAGED ACCURACY OF DIFFERENT METHODS

	Accuracy (%)			
	AA	BB	CC	Avg.
Proposed method	70.3	67.6	71.2	69.7
Baseline method	68.6	63.8	66.3	66.2
Tsinghua's method	69.3	60.2	68.4	66.0
Blanchard's method	70.4	65.0	71.0	68.8

#### IV. EXPERIMENTAL RESULTS

To evaluate the performance of our method, we tested it on the BCI Competition 2003, Dataset 2a. This dataset consisted of ten 30-min sessions for three subjects (A, B, C). In each session, there are 192 trials. The training set consisted of all the trials of 1—6 sessions. The test set consisted of 7—10 sessions. Both the proposed method and the baseline method were applied to this dataset. To set the initial value of  $\hat{P}(y | x_{ij})$  in the proposed method, a six fold cross-validation was performed on the training dataset. Since in the baseline method we had to assign label to each segment, as proposed in [6], we assume the labels for the segments of target 1 belong to  $C_1$  and labels of target 4 belong to  $C_0$ , and used the trials belong to the first and the fourth targets of the first six sections to train the classifier. The three thresholds  $\{t_i\}_{i=1,2,3}$  are chosen in the training process.

In order to benchmark the performance of our proposed algorithm, we compared it with baseline method, Tsinghua's method [5] and Blanchard's method [6]. The averaged accuracies of each of these methods are listed in Tab.I. In Table I, AA, BB and CC denote different subjects. Avg. denotes the averaged accuracy over all three subjects. It is worth mentioning that since our proposed method is continuous, here we compare our results with the results of continuous method in [6]. It is clear from Tab.I, that the proposed method outperforms the baseline method and Tsinghua's method on every subject. For subject BB the results of the proposed method are better than all the other three methods. This comparison shows that the accuracy of our method is among the best.

#### V. CONCLUSIONS

In this paper, we proposed a new statistical method, based on the Bayesian learning framework, to learn parameters of a classifier under MAP criterion. To solve the unlabeled problem, we used the estimated probability  $\hat{P}(y | x_{ij})$  to assist the optimization process,

and update  $\hat{P}(y | x_{ij})$  iteratively. Experiments showed that the accuracy of our algorithm is among the best, and the proposed method can make use of all the training data.

#### ACKNOWLEDGMENT

The authors would like to thank Dr Yuanqing Li, Mr Wenjie Xu and Dr Haihong Zhang for their fruitful discussions.

#### REFERENCES

- [1] Wolpaw J R, Birbaumer N, MacFarland D J, Pfurtscheller G and Vaughan T M: 'Brain-computer interface for communication and control'. *Clin. Neurophysiol.*, vol. 113, pp. 767-791, 2002.
- [2] D.J. McFarland and J.R. Wolpaw, "EEG-based communication and control: speed-accuracy relationships", *Appl. Psychophysiol. Biofeedback*, vol. 28, pp. 217-31, 2003.
- [3] J.R. Wolpaw and D.J. McFarland, "Control of a two-dimensional movement signal by a noninvasive brain-computer interface in humans", *PNAS*, vol. 101, pp17849—17854, 2004.
- [4] S.J. Roberts and W.D. Penny, "Real-time Brain Computer Interfacing: a preliminary study using Bayesian learning", *Medical and Biological Engineering and Computing*, vol. 38, No. 1, pp.56-61, 2000.
- [5] Ming Cheng, Wenyan Jia, Xiaorong Gao, Shangkai Gao, Fusheng Yang, "Mu rhythm-based cursor control: an offline analysis", *Clinical Neurophysiology*, vol. 115, pp. 745-751, 2004.
- [6] G. Blanchard and B. Blankertz, "BCI Competition 2003—Data Set IIa: Spatial Patterns of Self-Controlled Brain Rhythm Modulations", *IEEE Trans. Biomed. Eng.*, vol. 51, No. 6, pp. 1062-1066, June 2004.
- [7] B. Blankertz, KR. Müller, G. Curio, T.M. Vaughan, G. Schalk, J.R. Wolpaw, et. al. "The BCI Competition 2003: Progress and Perspectives in Detection and Discrimination of EEG Single Trials", *IEEE Trans. Biomed. Eng.*, vol. 51, pp. 1044-1051, June 2004.
- [8] D.J.C. MacKay, "Bayesian interpolation", *Neural Computation*, vol. 43, pp. 415-447, 1991.
- [9] D.J.C. MacKay, "The evidence framework applied to classification networks", *Neural Computation*, vol. 45, pp. 689-714, 1991.
- [10] C.M. Bishop, *Neural Networks for Pattern Recognition*, Oxford University Press, Oxford, 1995.
- [11] Tony Jebara, *Machine Learning Discriminative and Generative*, Kluwer Academic Publishers, 2004.