Proceedings of the 28th IEEE
EMBS Annual International Conference
New York City, USA, Aug 30-Sept 3, 2006

A Semi-supervised SVM Learning Algorithm for Joint Feature Extraction and Classification in Brain Computer Interfaces

Yuanqing Li and Cuntai Guan

Abstract— **In machine learning based Brain Computer Interfaces (BCIs), it is a challenge to use only a small amount of labelled data to build a classifier for a specific subject. This challenge was specifically addressed in BCI Competition 2005 [3]. Moreover, an effective BCI system should be adaptive to tackle the dynamic variations in brain signal. One of the solutions is to have its parameters adjustable while the system is used online. In this paper we introduce a new semi-supervised support vector machine (SVM) learning algorithm. In this method, the feature extraction and classification are jointly performed in iterations. This method allows us to use a small training set to train the classifier while maintaining high performance. Therefore, the tedious initial calibration process is shortened. This algorithm can be used online to make the BCI system robust to possible signal changes. We analyze two important issues of the proposed algorithm, the robustness of the features to noise and the convergence of algorithm. We applied our method to data from BCI competition 2005, and the results demonstrated the validity of the proposed algorithm.**

I. Introduction

Research in Brain-Computer Interfaces (BCIs) has received more and more attentions in recent years [1], [2], because BCI provides an alternative means of communication and control for people with severe motor disabilities. Electroencephalogram (EEG) based BCI measures specific components of EEG activity, extracts their features and translates these features into commands to control a cursor or devices (e.g., a robot arm, a wheelchair, etc.). Among various features extracted from EEG signal, common spatial pattern (CSP) is one of the most effective features in discriminating different classes of motor imagery [1], [2]. To extract CSP feature, an initial calibration process is needed to collect labelled training data for the estimation of the CSP transformation matrix, which is actually a spatial filter. The challenge is, can we reduce the time of this tedious calibration process? This is one of the important objectives in BCI research. A common database for researchers to tackle the challenging small training set problem was provided by Muller and his colleagues in recent BCI competition III [3], [4]. We will propose a solution in this paper and evaluate its effectiveness on this date set.

Semi-supervised learning is a method which finds a decision rule from both labelled and unlabelled data [5]. Since it only needs a small amount of labelled data, it is naturally selected to tackle the small training data set problem in BCI systems. The necessary condition for a semi-supervised learning algorithm (as well as for an unsupervised learning algorithm) is that the applied feature set should be sufficiently consistent [6]. However, in the existing BCI system where CSP feature is used, if the training data set is small, the feature consistency is not guaranteed. This is because we need a sufficient training data set to determine the CSP transformation matrix. In other words, this type of standard semi-supervised learning method dose not work well here. To solve this problem, in this paper we propose a new semi-supervised support vector machine (SVM) learning algorithm in which the feature extraction and classification are jointly performed in iterations. In this way, the consistency of features is ensured and the classification accuracy is improved. We will discuss two important issues of the proposed algorithm, robustness of the features to noise and the convergence of algorithm. We applied our method to data from BCI competition 2005, and the results demonstrated the validity of our algorithm.

Yuanqing Li is with Institute for Infocomm Research, Singapore 119613 yqli2@i2r.a-star.edu.sg

Cuntai Guan is with Institute for Infocomm Research, Singapore 119613 ctguan@i2r.a-star.edu.sg

II. CSP feature extraction and the robustness of CSP feature

In this section, first we introduce CSP feature extraction briefly. The detailed descriptions can be found in [2] and references therein. Next, we show that CSP feature extraction can be explained using the framework of Rayleigh coefficient maximization. Finally, the robustness of CSP feature is discussed.

A. CSP feature extraction

Given two sets of data, N_1 and N_2 denote numbers of trials for training and test set, respectively. Define

$$\Sigma^{(1)} = \sum_{j \in C_1} \frac{\mathbf{E}_j * \mathbf{E}_j^T}{trace(\mathbf{E}_j * \mathbf{E}_j^T)}, \ \Sigma^{(2)} = \sum_{j \in C_2} \frac{\mathbf{E}_j * \mathbf{E}_j^T}{trace(\mathbf{E}_j * \mathbf{E}_j^T)}, \tag{1}$$

where $\mathbf{E}_j \in R^{m \times k_2}$ denotes an EEG data matrix of the jth trial, m is the number of selected channels, k_2 is the number of samples in each trial, C_1 and C_2 refer to the first and second class of the training data.

We can find a joint diagonalization matrix $\mathbf{W}$ such that

$$\mathbf{W}\Sigma^{(1)}\mathbf{W}^T = \mathbf{D}, \quad \mathbf{W}\Sigma^{(2)}\mathbf{W}^T = \mathbf{I} - \mathbf{D}, \tag{2}$$

where $\mathbf{I}$ is an identity matrix, $\mathbf{D}$ is a diagonal matrix.

We can then construct a matrix $\bar{\mathbf{W}}$, called CSP transformation matrix, composed of the first l_1 and the last l_2 rows

of $\mathbf{W}$. For the jth trial of EEG data $\mathbf{E}_j$, the CSP feature vector is defined as

$$\mathbf{cf}(j) = diag(\bar{\mathbf{W}} \frac{\mathbf{E}_j \mathbf{E}_j^T}{trace(\mathbf{E}_j \mathbf{E}_j^T)} \bar{\mathbf{W}}^T), \tag{3}$$

where $j = 1, \cdots, N_1 + N_2$.

The above CSP feature extraction can be explained in the framework of Rayleigh coefficient maximization. Maximizing Rayleigh coefficient is to solve the following optimization problem [7],

$$\max J(\mathbf{q}) = \frac{\mathbf{q}^T \mathbf{S}_I \mathbf{q}}{\mathbf{q}^T \mathbf{S}_N \mathbf{q}}, \tag{4}$$

where $J(\mathbf{q})$ is Rayleigh coefficient, $\mathbf{S}_I$ and $\mathbf{S}_N$ are symmetric $m \times m$ matrices designed such that they measure the desired information and the undesired noise along the direction of $\mathbf{q}$.

The ratio in (4) is maximized when one covers as much as possible for the desired information while avoiding the undesired. The solution of (4) can be obtained by solving the following generalized eigenproblem,

$$\mathbf{S}_I \mathbf{q} = \lambda \mathbf{S}_N \mathbf{q}. \tag{5}$$

where λ is a generalized eigenvalue, and $\mathbf{q}$ is a generalized eigenvector corresponding to λ.

Note that there are m generalized eigenvectors, which can be obtained by jointly diagonalizing two matrices $\mathbf{S}_I$ and $\mathbf{S}_N$.

Define

$$\mathbf{S}_I = \Sigma^{(1)} - \Sigma^{(2)}, \ \ \mathbf{S}_N = \Sigma^{(1)} + \Sigma^{(2)}. \tag{6}$$

Then one can find that each row of the CSP transformation matrix $\mathbf{W}$ in (2) is also a local optimal solution of the optimization problem (4). Thus CSP feature extraction can be explained in the framework of Rayleigh coefficient maximization. As seen in Section III, the sum of Rayleigh coefficients will be used as a convergence index (or the index for terminating iterations) of our algorithm.

B. Robustness of CSP feature

Now we consider the robustness of CSP feature. The robustness property of CSP feature will play an important role in our algorithm. When noise is present, we hope the CSP feature is robust enough to distinguish between different classes. Consider two correlation matrices $\Sigma^{(1)} + \varepsilon_{\mathbf{1}}$ and $\Sigma^{(2)} + \varepsilon_{\mathbf{2}}$, where $\Sigma^{(1)}$ and $\Sigma^{(2)}$ are the same as those in (1), $\varepsilon_{\mathbf{1}}$ and $\varepsilon_{\mathbf{2}}$ are symmetric correlation matrices of noise. We can find a joint diagonalization matrix $\mathbf{W}(\epsilon)$, such that

$$\begin{aligned} \mathbf{W}(\epsilon)(\Sigma^{(1)} + \varepsilon_{\mathbf{1}})\mathbf{W}^T(\epsilon) &= \mathbf{D}(\epsilon), \\ \mathbf{W}(\epsilon)(\Sigma^{(2)} + \varepsilon_{\mathbf{2}})\mathbf{W}^T(\epsilon) &= 1 - \mathbf{D}(\epsilon), \end{aligned} \tag{7}$$

where ϵ denotes $max\{||\varepsilon_1||_1, ||\varepsilon_2||_1\}$.

Similar to (3), the noisy CSP feature (denoted as $\mathbf{cfn}(\epsilon, j)$) for the jth trial is

$$\mathbf{cfn}(\epsilon, j) = diag(\bar{\mathbf{W}}(\epsilon) \frac{\mathbf{E}_j \mathbf{E}_j^T}{trace(\mathbf{E}_j \mathbf{E}_j^T)} \bar{\mathbf{W}}^T(\epsilon)). \tag{8}$$

Theorem 1: Considering (2)-(8), we have: 1.

$$\lim_{\epsilon \to 0} \mathbf{W}(\epsilon) = \mathbf{W}, \ \lim_{\epsilon \to 0} \mathbf{D}(\epsilon) = \mathbf{D}; \tag{9}$$

2.

$$\lim_{\epsilon \to 0} \mathbf{cfn}(\epsilon, j) = \mathbf{cf}(j). \tag{10}$$

The proof is not difficult and we omitted here due to the limited page space.

From Theorem 1, we can see that CSP feature is robust to additive noise to some degree.

III. A SEMI-SUPERVISED SVM LEARNING ALGORITHM

Based on the CSP feature, we introduce a new semi-supervised SVM learning algorithm where the CSP feature extraction and classification are jointly performed in each iteration. Furthermore, we will present the results on convergence analysis of the iterative algorithm.

A. Algorithm

In the following, we present our algorithm.

Algorithm 1

Step 1. (Initial step) Suppose that initial training data set D_I and test data set D_T contain N_1 and N_2 data examples respectively. The labels of D_I are known, while the labels of D_T are unknown. Initially, we train a CSP transformation matrix $\bar{\mathbf{W}}_1$ using D_I, and extract CSP features on D_I and D_T. With the extracted feature vectors, perform a classification using a standard SVM. The predicted labels are denoted as $[y_1(1), \cdots, y_1(N_2)]$;

Step 2. The kth iteration ($k = 2, \cdots$) follows Steps 2.1-2.6.

Step 2.1. (Update the training data set) Define a new training data set D_k as $D_I + D_T$, where the labels of D_T are the predicted labels $[y_{k-1}(1), \cdots, y_{k-1}(N_2)]$ in the $(k-1)$th iteration.

Step 2.2 (Feature re-extraction) Using the training data D_k, regenerate the CSP transformation matrix $\bar{\mathbf{W}}_k$ and then re-extract CSP features on D_I and D_T. The ith feature vector is denoted as $\mathbf{x}_k(i)$, where k refers to the kth iteration, and $i = 1, \cdots, N_1 + N_2$.

Step 2.3. Calculate the sum of Rayleigh coefficients,

$$R(k) = \sum_{i=1}^{n} \left| \frac{(\mathbf{w}_i^{(k)})^T \mathbf{S}_I^{(k)} \mathbf{w}_i^{(k)}}{(\mathbf{w}_i^{(k)})^T \mathbf{S}_N^{(k)} \mathbf{q} \mathbf{w}_i^{(k)}} \right|, \tag{11}$$

where $\mathbf{w}_i^{(k)}$ is the ith column of the transformation matrix $\bar{\mathbf{W}}_k$, $\mathbf{S}_I^{(k)}$ and $\mathbf{S}_N^{(k)}$ are calculated as in (6) using D_k with the labels predicted in the $(k-1)$th iteration.

Step 2.4. (Classification) With the extracted feature vectors, perform a classification using a standard SVM. Note that the training set here is the feature set of D_k. The predicted labels for the test data set are denoted as $[y_k(1), \cdots, y_k(N_2)]$;

Step 3. (Termination step) Given that α is a pre-determined positive constant, if $|R(k) - R(k-1)| < \alpha$, the algorithm stops after the kth iteration, and the predicted labels $[y_k(1), \cdots, y_k(N_2)]$ of the test set are the final classification results. Otherwise, perform the $(k+1)$th iteration.

In this algorithm, the test set is used to augment the training set in order to improve the efficiency of feature

extraction and classification. Naturally, there are possible errors in the prediction of the test set labels. A necessary condition is that CSP feature should be robust to the errors (seen as noise here) to some degree. We have shown in Section 2 that CSP feature is indeed robust to noise to some extent, so we believe this type of errors should not spoil the effectiveness of our algorithm. Experiment results in Section 4 show that this is true.

Remark 1: Comparing Algorithm 1 to those existing standard semi-supervised algorithms, there are a major improvement in our algorithm. That is, CSP features are kept updated by feature re-extraction in our algorithm.

Remark 2: In algorithm 1, the system parameters (CSP transformation matrix and classifier parameters) are updated iteratively with the help of test data. With the similar concept, this method can be used online to adjust the system parameters with available test data. Therefore, this algorithm can be used to improve the adaptability of a BCI system.

B. Algorithm convergence

We now analyze the convergence of Algorithm 1 under the following two assumptions.

Assumption 1: Suppose that two classes of data are generated from two Gaussian distributions $N(e_1, \sigma^2)$ and $N(e_2, \sigma^2)$ respectively, where e_1 and e_2 are two different means, σ^2 is the variance.

Assumption 2: Suppose that in the training data set and the test data set, the number of the data examples in the first class is equal (or close) to that of the data examples in the second class. Furthermore, in each iteration of Algorithm 2, the labels of the two classes are predicted correctly with equal probabilities.

Assumption 3: In Algorithm 1, suppose that the improvement of the sum of Rayleigh coefficients will lead to the increase of the prediction accuracy rate, i.e., $R(k+1) > R(k)$ will lead to $rate_{k+1} > rate_k$.

On the convergence of Algorithm 1, we have the following theorems.

Theorem 2: Under Assumptions 1 and 2, (1) if $rate_k > rate_{k-1}$, then $R(k+1) > R(k)$, where $R(k)$ is defined in (11), $rate_k$ is the prediction accuracy rate in the kth iteration of Algorithm 2; (2) $\{R(k)\}$ is bounded.

The proof of Theorem 2 is omitted here due to the limit of page space.

Theorem 3: Under Assumptions 1, 2 and 3,

$$R(1) < R(2) < \cdots < R(k) < \cdots. \tag{12}$$

(2) The algorithm 1 is convergent.

Proof: First, we prove that

$$R(2) > R(1). \tag{13}$$

In the Step 1 of Algorithm 1, we first obtain a transformation matrix denoted as $\bar{\mathbf{W}}_1$ and features for all trials of data. The predicted labels $[y_1(1), \cdots, y_1(N_2)]$ for test data set and a prediction accuracy $rate_1$ are then obtained a standard SVM classification.

Using $\bar{\mathbf{W}}_1$, training data set and test data set with the predicted labels $[y_1(1), \cdots, y_1(N_2)]$, we calculate the sum of Rayleigh coefficients $R(1)$,

$$R(1) = \sum_{i=1}^{n} \left| \frac{(\mathbf{w}_i^{(1)})^T \mathbf{S}_I^{(2)} \mathbf{w}_i^{(1)}}{(\mathbf{w}^{(1)})^T \mathbf{S}_N^{(2)} \mathbf{w}^{(1)}} \right|, \tag{14}$$

where $\mathbf{S}_I^{(2)}$ and $\mathbf{S}_N^{(2)}$ are calculated as in (11).

Using the above $\mathbf{S}_I^{(2)}$ and $\mathbf{S}_N^{(2)}$, we obtain the transformation matrix $\bar{\mathbf{W}}_2$ by maximizing Rayleigh coefficient in (4). The corresponding sum of Rayleigh coefficients is $R(2)$. Compared with $R(1)$, $R(2)$ is an optimal value. Thus we have (13).

It follows from Assumption 3 and (13) that $rate_2 > rate_1$. According to Theorem 2, this leads to

$$R(3) > R(2). \tag{15}$$

Repeating the above deduction, we have (12).

Since $\{R(k)\}$ are bounded (Theorem 2), thus it is convergent. This implies the convergence of Algorithm 2. The theorem is proven.

The inequalities in (12) has been demonstrated in our experimental data analysis example.

Remarks 4: (1) The above proof of Theorem 3 also shows the working mechanism of Algorithm 2. That is, the sum of Rayleigh coefficients and prediction accuracy rates are improved alternatively. (2) Note that the improvement of Rayleigh coefficient may not strictly lead to a higher prediction accuracy rate, thus Assumption 3 is generally but not always valid.

IV. Experimental Results

In this section, we evaluate our methods with the following data set: Data set IVa in BCI Competition 2005, provided by K. R. Mller, B. Blankertz (Fraunhofer FIRST, Intelligent Data Analysis Group), and G. Curio (Neurophysics Group, Department of Neurology, Campus Benjamin Franklin of the Charit - University Medicine Berlin). This data set is provided for researchers to evaluate their algorithm performance when only a small amount of labeled training data is available. The details of the experiment and data set can be found from the website: http://ida.first.fraunhofer.de /projects/bci/competition. In each trial of the experiment, visual cues indicated for 3.5 s which of the following 2 motor imageries the subject should perform: (R) right hand, (F) right foot. The presentation of target cues were intermitted by periods of random length, 1.75 to 2.25 s, in which the subject could relax. In this paper, we present our analysis results for 3 of the five subjects in this paper ("aa", "al" and "ay").

Apart from the feature extraction and classification, proper preprocessing is also important to ensure good performance. In this paper, the preprocessing methods include Common Average Reference (CAR) for spatial filtering, frequency filtering (in mu band). For each trial, we use 3.5 sec length of data for analysis. During this period, the cue was visible on the screen.

As an example, we first describe our analysis procedure and results for subject "aa". We use the first 160 trials to train the model with "cross-validation", and the subsequent 80 trials as an independent test set. We divide the 160 trials into 5 folds according to their sequential order. We use one of the 5 folds as the initial training set and the rest 4 folds as test set. To distinguish this test set from the independent test set, we call the 4 folds learning-test set. During each iteration, we perform all the steps of Algorithm 1 on the cross-validation data, at the same time, we also extract the CSP features of independent test set and calculate the corresponding prediction accuracy.

In each iteration, we calculate the prediction accuracy rates $ac(k,j)$ for learning-test set, and $ac_I(k,j)$ for independent test set, where k represents the kth iteration, $j(=1,\cdots,5)$ represents the jth fold which is used as the initial training set. The average accuracy rates over all folds are calculated as

$$rate(k)=\frac{1}{5}\sum_{j=1}^{5}ac(k,j),\ rate_I(k)=\frac{1}{5}\sum_{j=1}^{5}ac_I(k,j). \tag{16}$$

In each iteration of Algorithm 1, we also calculate the sum of the Rayleigh coefficients as in (11). When cross-validation is finished, the average Rayleigh coefficients is then obtained.

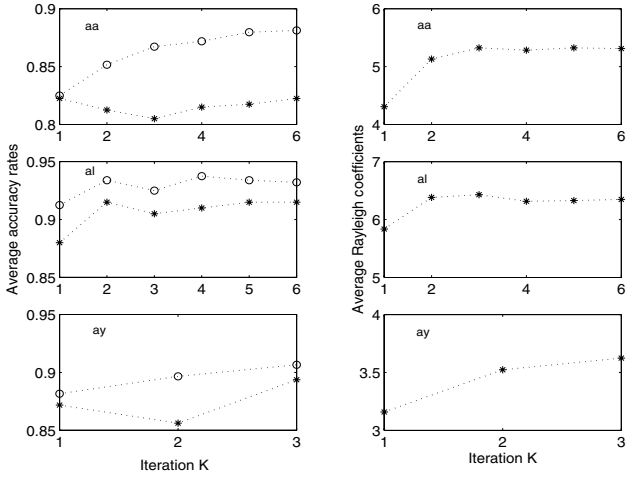

Fig. 1. Data analysis results. The 1st, 2nd and 3rd rows are for Subjects "aa", "al" and "ay", respectively. The first column shows average prediction accuracy rates for learning-test set (lines with "o") and average prediction accuracy rates for independent test set (lines with "*"). The 2nd column depicts the curves of average Rayleigh coefficients, which is used as an convergence index in our algorithm.

The results are shown in the first row of Fig. 1. In the left subplot, $rate(k)$ obtained from learning-test set is depicted in line with circles. For independent test set, the average accuracy rates $rate_I(k)$ are depicted in the line with with stars. The curve of average Rayleigh coefficients is shown in the right subplot of the first row. This curve demonstrated the convergence of our algorithm.

For subjects "al" and "ay", we performed the same analysis as above. The corresponding results are shown in the second and the third row of Fig. 1. Note that the iteration numbers for subjects "aa," "al" and "ay" are 6, 6 and 3 , respectively.

Until now, we have presented our experimental data analysis results. Discussions on these experiment results will be given in the next section.

V. Discussions and conclusions

In this paper, we have presented a semi-supervised SVM learning algorithm for BCI systems. This algorithm is mainly meant to reduce the training time during calibration. At the same time, this algorithm can also be used to improve the adaptability of the BCI system. We addressed two key issues of the proposed algorithms, namely the robustness of the features to noise and the algorithm convergence.

From the experimental results, we may come to the following conclusions:

1. Compare the accuracy rates obtained with Algorithm 1 (the rates in the last iteration), which is embedded with a feature re-extraction process, to those obtained with a standard SVM classification algorithm (the rates in the first iteration). From Fig.1, we can see from the three subplots of the first column that, the proposed method consistently outperform the standard SVM method. This indicates that semi-supervised SVM with feature re-extraction do help improve the performance of classification.

2. From subplots of the 2nd column in Fig. 1, we can see that the algorithm is convergent in terms of the sum of the Rayleigh coefficients. This also provides a useful criterion for iteration termination.

References

[1] J. R. Wolpaw, N. Birbaumer, D. J. McFarland, G. Pfurtscheller, T. M. Vaughan, "Brain-computer interfaces for communication and control," Clinical Neurophysiology, vol. 113, pp. 767-791, 2002.

[2] G. Blanchard, B. Blankertz, "BCI competition 2003-data set IIa: spatial patterns of self-controlled brain rhythm modulations," *IEEE Transactions on Biomedical Engineering*, Vol. 51(6), pp. 1062-1066, 2004.

[3] http://ida.first.fraunhofer.de/projects/bci/competitioniii.

[4] G. Dornhege, B. Blankertz, G. Curio, and K. R. Muller. "Boosting bit rates in non-invasive EEG single-trial classifications by feature combination and multi-class paradigms," *IEEE Trans. Biomed. Eng.*, 51(6):993-1002, June 2004.

[5] K. Nigam and R. Ghani, "Analyzing the effectiveness and applicability of co-training," *Proceedings of 9th International Conference on Information and Knowledge Management,* pp. 86-93, 2000.

[6] D. Zhou, O. Bousquet, T. N. Lal, J. Weston, and B. Scholkopf, "Learning with Local and Global Consistency," *Advances in Neural Information Processing Systems*, vol. 15. Cambridge, MA: MIT Press, 2003.

[7] S. Mika, *Kernel Fisher Discriminants*, Ph.D. thesis, 2002.

[8] C. C. Chang and C. J. Lin, "LIBSVM – A Library for Support Vector Machines," *http://www.csie.ntu.edu.tw/ cjlin/libsvm/.*