LETTER Communicated by Klaus-Robert Mueller

An Extended EM Algorithm for Joint Feature Extraction and Classification in Brain-Computer Interfaces

Yuanqing Li
yqli2@i2r.a-star.edu.sg
Cuntai Guan
ctguan@i2r.a-star.edu.sg
Neural Signal Processing Lab, Institute for Infocomm Research, Singapore 119613

For many electroencephalogram (EEG)-based brain-computer interfaces (BCIs), a tedious and time-consuming training process is needed to set parameters. In BCI Competition 2005, reducing the training process was explicitly proposed as a task. Furthermore, an effective BCI system needs to be adaptive to dynamic variations of brain signals; that is, its parameters need to be adjusted online. In this article, we introduce an extended expectation maximization (EM) algorithm, where the extraction and classification of common spatial pattern (CSP) features are performed jointly and iteratively. In each iteration, the training data set is updated using all or part of the test data and the labels predicted in the previous iteration. Based on the updated training data set, the CSP features are reextracted and classified using a standard EM algorithm. Since the training data set is updated frequently, the initial training data set can be small (semisupervised case) or null (unsupervised case). During the above iterations, the parameters of the Bayes classifier and the CSP transformation matrix are also updated concurrently. In online situations, we can still run the training process to adjust the system parameters using unlabeled data while a subject is using the BCI system. The effectiveness of the algorithm depends on the robustness of CSP feature to noise and iteration convergence, which are discussed in this article. Our proposed approach has been applied to data set IVa of BCI Competition 2005. The data analysis results show that we can obtain satisfying prediction accuracy using our algorithm in the semisupervised and unsupervised cases. The convergence of the algorithm and robustness of CSP feature are also demonstrated in our data analysis.

1 Introduction

As brain-computer interfaces (BCIs) provide an alternative means of communication and control for people with severe motor disabilities (Birbaumer et al., 1999), research into BCIs has received more attention in recent years, as seen in Blanchard and Blankertz (2004), Donoghue (2002),

Kubler, Kotchoubey, Kaiser, Wolpaw, and Birbaumer (2001), Pfurtscheller et al. (2000), and Wolpaw, Birbaumer, McFarland, Pfurtscheller, and Vaughan (2002). Being noninvasive, electroencephalogram (EEG)-based BCI measures specific components of EEG activity, extracts features, and translates these features into control signals to devices such as a robot arm or a cursor. The features that are commonly used in EEG-based BCIs include visual evoked potentials, slow cortical potentials, P300 evoked potentials, common spatial pattern (CSP) features, mu and beta rhythms, and other activities from sensorimotor cortex and autoregressive parameters (Wolpaw et al., 2002).

CSP features of EEG signals correspond to event-related desynchronization (ERD) and event-related synchronization (ERS) evoked by motor imagery or movements (Pfurtscheller, Neuper, Flotzinger, & Pregenzer, 1997). CSP feature is very effective in discriminating several motor imageries (Blanchard & Blankertz, 2004; Ramoser, Muller-Gerking, & Pfurtscheller, 2000; Wolpaw et al., 2002). However, CSP feature extraction relies on a time-consuming training process to determine a spatial filter matrix, also known as the CSP transformation matrix. Considering the importance of training effort reduction, Muller and his colleagues provided a data set with a small training data set for the BCI competition in 2005 (Dornhege, Blankertz, Curio, & Muller, 2004; see http://ida.first.fraunhofer.de/projects/bci/competition).

In this article, semisupervised learning, which refers to finding a decision rule from both labeled and unlabeled data, will be used to tackle the small training data set problem in BCI systems. Semisupervised learning has gained much appeal in recent years due to its potential in reducing labeling and training effort, which is usually tedious and time-consuming (Nigam & Ghani, 2000; Grandvalet & Bengio, 2004). A necessary condition for a semisupervised learning algorithm, as well as for an unsupervised learning algorithm, is that the applied feature set has sufficient consistency (Zhou, Bousquet, Lal, Weston, & Schölkopf, 2003). Otherwise the algorithm will not work well. In this article, the consistency is reflected in the Fisher ratio of the two classes of features. If a small data set is used for training and CSP features are then directly extracted from the training data set and the test data set, the feature consistency is not sufficient. Consequently, a standard semisupervised learning method cannot be directly employed for classification. To solve this problem, we propose an extended expectation maximization (EM) algorithm by embedding a feature reextraction into the standard EM algorithm.

In each iteration of the proposed algorithm, the training data set is updated using all or part of the test set[1] with labels (predicted in the previous iteration) in order to make the training data become sufficient or expanded.

[1] In this article, the term *test set* refers to unlabeled trials.

Based on the updated training data set, CSP features are then reextracted and classified by a standard EM algorithm. The improvement in prediction accuracy in one iteration leads to a higher consistency of CSP feature in the next iteration, and the latter leads to a further improvement in the subsequent prediction accuracy, and so on. This is the main difference between our algorithm and conventional semisupervised algorithms. The initial training set can be small or even null, that is, the proposed algorithm can be used in both a semisupervised learning case (with a small initial training data set) and a unsupervised learning case (without any initial training data set).

Furthermore, the proposed algorithm can be used to improve the adaptability of BCI systems. In general, if we do not consider the adaptability of a BCI system, its parameters, such as the CSP transformation matrix and the classifier parameters, do not change once determined unless new training is performed. Many researchers think that EEG and other electrophysiological signals typically display short-term and long-term variations linked to several factors, such as time of day, hormonal levels, immediate environment, recent events, fatigue, and illness (McEvoy, Smith, & Gevins, 2000; Polich, 2004; Regan, 1989; Wolpaw et al., 2002). In other words, for BCI systems, the consistency in the features may have only a short-term existence. Adaptability is a recommendation for an effective BCI system, by adjusting the latter's parameters online (Millan & Mourino, 2003; Vidaurre, Schlogl, Cabeza, Scherer, & Pfurtscheller, 2005). This reduces the impact of such spontaneous variations and keeps the consistency of features. As will be seen, the CSP transformation matrix and the Bayes classifier parameters are also updated by using test data and predicted labels in our method. The BCI system parameters can thus be adjusted online without new system training.

The remainder of this article is organized as follows. In section 2, we describe CSP feature extraction and analyze the robustness of CSP feature to noise. In section 3, we introduce our extended EM algorithm. The convergence of the algorithm is also analyzed. Data analysis results in section 4 demonstrate our algorithm's validity. Section 5 concludes with discussions of the data analysis results.

2 CSP Feature Extraction and the Robustness of the CSP Feature

The CSP feature, which is commonly used in EEG-based BCI systems, is very effective for discriminating motor imageries. In this section, we describe CSP feature extraction and present the robustness analysis of CSP feature.

2.1 CSP Feature Extraction. For the convenience of the following analysis, we present the main steps of CSP feature extraction, which can be seen

in Blanchard and Blankertz (2004) and Lemm, Blankertz, Curio, and Muller (2005).

Hereafter, let N_1 and N_2 denote the trial numbers of the training set and the test set respectively.

Define

$$\Sigma^{(1)} = \sum_{j \in C_1} \frac{\mathbf{E}_j * \mathbf{E}_j^T}{trace\left(\mathbf{E}_j * \mathbf{E}_j^T\right)}, \quad \Sigma^{(2)} = \sum_{j \in C_2} \frac{\mathbf{E}_j * \mathbf{E}_j^T}{trace\left(\mathbf{E}_j * \mathbf{E}_j^T\right)}, \tag{2.1}$$

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, and C_1 and C_2 refer to the first class and the second class of trials of training data set, respectively.

The two matrices $\Sigma^{(1)}$ and $\Sigma^{(2)}$ are then jointly diagonalized. Set $\Sigma = \Sigma^{(1)} + \Sigma^{(2)}$, which is a symmetric matrix. Let $\mathbf{V}$ be an orthogonal matrix whose first row vector is nonnegative (if the first entry of a column vector is negative, we use -1 to times the column vector), such that

$$\mathbf{V}^T \Sigma \mathbf{V} = \mathbf{P}, \tag{2.2}$$

where $\mathbf{P}$ is a diagonal matrix composed by the eigenvalues of Σ, in a decreasing order.

Set $\mathbf{U} = (\mathbf{P})^{\frac{1}{2}} \mathbf{V}^T$, $R_1 = \mathbf{U}\Sigma^{(1)}\mathbf{U}^T$, $R_2 = \mathbf{U}\Sigma^{(2)}\mathbf{U}^T$.

It is observed that R_1 is symmetrical; thus, we can find an orthogonal matrix denoted as $\mathbf{Z}$ with its first row vector being nonnegative, such that

$$\mathbf{Z}^T R_1 \mathbf{Z} = D = diag(d_1, \ldots, d_m), \tag{2.3}$$

where the elements in the diagonal lines of D are sorted in decreasing order, and $0 \leq d_1, \ldots, d_m \leq 1$.

Define $\mathbf{W} = \mathbf{Z}^T \mathbf{U}$; then

$$\mathbf{W}\Sigma^{(1)}\mathbf{W}^T = \mathbf{D}, \quad \mathbf{W}\Sigma^{(2)}\mathbf{W}^T = \mathbf{I} - \mathbf{D}. \tag{2.4}$$

$\mathbf{I}$ is the identity matrix.

Next, we construct the CSP transformation matrix $\bar{\mathbf{W}}$, composed by the first l_1 and the last l_2 rows of $\mathbf{W}$. The first l_1 rows of $\mathbf{W}$ correspond to the largest l_1 eigenvalues of $\mathbf{D}$, and the last l_2 rows of $\mathbf{W}$ correspond to the smallest l_2 eigenvalues of $\mathbf{D}$.

For the EEG data matrix $\mathbf{E}_j$ obtained from the jth trial, the CSP feature vector is defined as

$$\mathbf{cf}(j) = diag\left(\bar{\mathbf{W}} \frac{\mathbf{E}_j \mathbf{E}_j^T}{trace\left(\mathbf{E}_j \mathbf{E}_j^T\right)} \bar{\mathbf{W}}^T\right), \tag{2.5}$$

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

Remark 1. In the above CSP feature extraction, the first row vectors of two orthogonal matrices $\mathbf{V}$ and $\mathbf{Z}$ for diagonalizing matrices are set to be nonnegative. In a standard CSP feature extraction, there is no such constraint. As will be seen in section 2.2, $\mathbf{V}$ and $\mathbf{Z}$ are generally unique under this constraint. The uniqueness of $\mathbf{V}$ and $\mathbf{Z}$ is needed to guarantee the robustness of the CSP feature.

2.2 Robustness of the CSP Feature. An extracted feature that can effectively reflect the subject's intentions even in a noisy environment is highly desirable. This is a problem of the feature robustness to noise. In this article, there is no sufficient training data to determine the CSP transformation matrix, so the test data along with the predicted labels are used for training. Since the prediction error of labels is inevitable in each iteration, we need to consider the robustness of the CSP feature in our algorithm.

To analyze the robustness of the CSP feature, we consider two correlation matrices $\Sigma^{(1)} + \varepsilon_1$ and $\Sigma^{(2)} + \varepsilon_2$, where $\Sigma^{(1)}$ and $\Sigma^{(2)}$ are defined in equation 2.1. ε_1 and ε_2 are symmetric matrices related to noise, which can be additive in nature. ε_1 and ε_2 are defined in this article as follows,

$$\varepsilon_1 = \sum_{j \in C_1^r} \frac{\mathbf{E}_j * \mathbf{E}_j^T}{trace\left(\mathbf{E}_j * \mathbf{E}_j^T\right)}, \quad \varepsilon_2 = \sum_{j \in C_2^r} \frac{\mathbf{E}_j * \mathbf{E}_j^T}{trace\left(\mathbf{E}_j * \mathbf{E}_j^T\right)}, \tag{2.6}$$

where C_1^r and C_2^r denote the sets of trials misclassified in the first and second class, respectively.

As in the previous section, we can find a joint diagonalization matrix denoted as $\mathbf{W}(\epsilon)$, such that

$$\mathbf{W}(\epsilon)(\Sigma^{(1)} + \varepsilon_1)\mathbf{W}^T(\epsilon) = \mathbf{D}(\epsilon), \quad \mathbf{W}(\epsilon)(\Sigma^{(2)} + \varepsilon_2)\mathbf{W}^T(\epsilon) = 1 - \mathbf{D}(\epsilon), \tag{2.7}$$

where ϵ denotes $\max\{\|\varepsilon_1\|_1, \|\varepsilon_2\|_1\}$. Note that the 1-norm of a matrix implies the summation of the absolute values of all its entries.

In a noisy environment, the CSP feature (denoted as $\mathbf{cfn}(\epsilon, j)$) for the jth trial is

$$\mathbf{cfn}(\epsilon, j) = diag\left(\bar{\mathbf{W}}(\epsilon)\frac{\mathbf{E}_j\mathbf{E}_j^T}{trace\left(\mathbf{E}_j\mathbf{E}_j^T\right)}\bar{\mathbf{W}}^T(\epsilon)\right). \tag{2.8}$$

Before presenting our results, we present two lemmas. The first lemma can be found in Chen (2000) or other textbooks related to matrix theory.

Lemma 1 (*Bauer-Fike*). *Suppose that* $\boldsymbol{A} = \boldsymbol{Q}\boldsymbol{P}\boldsymbol{Q}^{-1}$, $\boldsymbol{P} = diag(\lambda_1, \ldots, \lambda_m)$. *Then for any eigenvalue u of* $\boldsymbol{A} + \boldsymbol{\Theta}$, *we have*

$$\min_i |\lambda_i - u| \le \| \boldsymbol{Q}^{-1}\boldsymbol{\Theta}\boldsymbol{Q}\|_2, \tag{2.9}$$

where $\boldsymbol{\Theta}$ *is a perturbation matrix with consistent dimension.*

In this article, the matrix norm $\|\mathbf{A}\|_2$ refers to the spectral norm of the matrix $\mathbf{A}$, that is, $\|\mathbf{A}\|_2 = \sqrt{\lambda}$, where λ is the maximum eigenvalue of $\mathbf{A}^T\mathbf{A}$. Spectral norm is consistent with Frobenius vector norm.

Lemma 2. *Suppose that a real symmetric matrix* $\boldsymbol{A} \in R^{m\times m}$ *has m different eigenvalues, and* $\boldsymbol{\Theta} \in R^{m\times m}$ *is a symmetric disturb matrix,* $\theta = \|\boldsymbol{\Theta}\|_2$. $\boldsymbol{G}(\theta)$ *and* $\boldsymbol{G}$ *are two orthogonal matrices with their first row vectors being nonnegative, such that*

$$\boldsymbol{G}^T(\theta)(\boldsymbol{A} + \Theta)\boldsymbol{G}(\theta) = \boldsymbol{Q}(\theta) = diag(q_1(\theta), \ldots, q_m(\theta)), \tag{2.10}$$

$$\boldsymbol{G}^T\boldsymbol{A}\boldsymbol{G} = \boldsymbol{\Lambda} = diag(\lambda_1, \ldots, \lambda_m). \tag{2.11}$$

Then we have

$$\lim_{\theta\to 0}\mathbf{G}(\theta) = \boldsymbol{G}, \ \lim_{\theta\to 0}\boldsymbol{Q}(\theta) = \boldsymbol{\Lambda}. \tag{2.12}$$

The proof is given in appendix A.

Theorem 1. *Considering equations 2.4 and 2.7, we have*

$$\lim_{\epsilon\to 0}\boldsymbol{W}(\epsilon) = \boldsymbol{W}, \ \lim_{\epsilon\to 0}\boldsymbol{D}(\epsilon) = \boldsymbol{D}. \tag{2.13}$$

Furthermore,

$$\lim_{\epsilon\to 0}\boldsymbol{cfn}(\epsilon, j) = \boldsymbol{cf}(j), \tag{2.14}$$

where $\boldsymbol{cfn}(\epsilon, j)$ *and* $\boldsymbol{cf}(j)$ *are defined in equations 2.5 and 2.8, respectively.*

The proof is given in appendix B.

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

3 An Extended EM Algorithm

In this section, an extended EM algorithm is proposed for joint CSP feature extraction and classification. We also discuss how the algorithm is used for both semisupervised and unsupervised learning for online BCI systems. Finally, we present several results on convergence analysis of the iterative algorithm.

Before introducing our algorithm, we present a simplified version of a standard EM algorithm, which can be found in Xu and Jordan (1996). Suppose a gaussian mixture probabilistic model as follows,

$$P(\mathbf{x}|\Omega) = \sum_{q=1}^{2} \alpha_q P(\mathbf{x}|\mathbf{m}^{(q)}, \mathbf{Var}^{(q)}),$$

$$P(\mathbf{x}|\mathbf{m}^{(q)}, \mathbf{Var}^{(q)}) = \frac{exp(-\frac{1}{2}(\mathbf{x}-\mathbf{m}^{(q)})^T(\mathbf{Var}^{(q)})^{-1}(\mathbf{x}-\mathbf{m}^{(q)})}{(2\pi)^{L/2}|\mathbf{Var}^{(q)}|^{1/2}}, \tag{3.1}$$

where $\mathbf{x} \in R^L$, the parameter vector Ω consists of the mixing proportions α_q, the mean vectors $\mathbf{m}^{(q)}$, and the covariance matrices $\mathbf{Var}^{(q)}$, $q = 1, 2$.

Assuming $\alpha_q = \frac{1}{2}$, the EM algorithm can be expressed as

$$h_k^{(q)}(t) = \frac{P\left(\mathbf{x}(t)|\mathbf{m}_k^{(q)}, \mathbf{Var}_k^{(q)}\right)}{\sum_{i=1}^{2} P\left(\mathbf{x}(t)|\mathbf{m}_k^{(i)}, \mathbf{Var}_k^{(i)}\right)},$$

$$\mathbf{m}_{k+1}^{(q)} = \frac{\sum_{t=1}^{N} h_k^q(t)\mathbf{x}(t)}{\sum_{t=1}^{N} h_k^q(t)},$$

$$\mathbf{Var}_{k+1}^{(q)} = \frac{\sum_{t=1}^{N} h_k^q(t)\left[\mathbf{x}(t)-\mathbf{m}_k^{(q)}\right]\left[\mathbf{x}(t)-\mathbf{m}_k^{(q)}\right]^T}{\sum_{t=1}^{N} h_k^q(t)}. \tag{3.2}$$

If we further assume that the two gaussian distributions are well separated, such that the posterior probabilities $h_k^{(q)}(t) \approx 1$ or $h_k^{(q)}(t) \approx 0$, then the above

iterative algorithm becomes

$$h_k^{(q)}(t) = \frac{P\left(\mathbf{x}(t)|\mathbf{m}_k^{(q)}, \mathbf{Var}_k^{(q)}\right)}{\sum_{i=1}^{2} P\left(\mathbf{x}(t)|\mathbf{m}_k^{(i)}, \mathbf{Var}_k^{(i)}\right)},$$

$$\mathbf{m}_{k+1}^{(q)} = \frac{\sum_{t=1}^{N_k^{(q)}} \mathbf{x}^{(q)}(t)}{N_k^{(q)}},$$

$$\mathbf{Var}_{k+1}^{(q)} = \frac{\sum_{t=1}^{N_k^{(q)}} \left[\mathbf{x}^{(q)}(t) - \mathbf{m}_k^{(q)}\right]\left[\mathbf{x}^{(q)}(t) - \mathbf{m}_k^{(q)}\right]^T}{N_k^{(q)}}, \tag{3.3}$$

where $\mathbf{x}^{(q)}(t)$ belongs to the qth class, $N_k^{(q)}$ is the number of samples belonging to the qth class in the kth iteration, and $q = 1, 2$.

Hereafter, equation 3.3 refers to a standard EM iteration.

3.1 Algorithm Steps. We first present a version of our extended EM algorithm for semisupervised learning. In the next section, we show how it is extended for the unsupervised learning case.

This is an iterative algorithm in which a naive Bayes classifier is used. In each iteration, we need to update the trial labels of the test set, the training data set, the CSP feature vectors of the initial training set and test set, and the parameters of Bayes classifier (mean vectors and covariance matrices).

Algorithm 1

Step 1: Initial step. Denote D_0 as the initial training data set. First, we train a CSP transformation matrix based on D_0 and extract CSP features on the initial training data set and test data set. Using the CSP features of the initial training data set, we then calculate two mean vectors and two covariance matrices for both classes as initial parameters of a naive Bayes classifier.

Step 2: The kth iteration ($k = 1, \ldots$) follows steps 2.1 to 2.6.

Step 2.1: With the CSP feature vectors extracted in the $(k-1)$th iteration, perform K_0 standard EM iterations in equation 3.3, where K_0 is a predefined positive integer.

Step 2.2: According to the posterior probabilities obtained in the K_0th standard EM iteration, we perform a classification on the test set (containing N_2 trials). The predicted labels are denoted as $[Label_k(1), \ldots, Label_k(N_2)]$.

Step 2.3: Update the training data set. Select $Int(\alpha N_2)$ trials from the test set that have higher posterior probabilities for retraining, where $\alpha \in (0, 1]$ is a predetermined percentage, and $Int(\alpha N_2)$ defines the largest integer, which is smaller than αN_2. The selected trials along with their predicted

labels are put together with the initial training data set D_0 to form an updated training data set denoted as D_k, which has $N_1 + Int(\alpha N_2)$ trials. Notice that the test set remains unchanged.

Step 2.4: Feature reextraction. Using the training data D_k, regenerate the CSP transformation matrix, and then reextract CSP features for all trials. The CSP feature vector of the ith trial is denoted as $\mathbf{cf}_k(i) = [cf_k(1,i), \ldots, cf_k(L,i)]^T$, where k refers to the kth iteration, L is the dimension number of feature vector, and $i = 1, \ldots, N_1 + N_2$.

Step 2.5: Calculate the mean vectors and covariance matrices of the two classes as new parameters of the Bayes classifier by using the reextracted features of the training data set D_k along with predicted labels.

Step 2.6: Find out the number of the trials from the test set with different predicted labels in the current and previous iteration,

$$dl_{k-1} = \sum_{i=1}^{N_2} |Label_k(i) - Label_{k-1}(i)|. \tag{3.4}$$

Step 3: Termination step. Given that M_0 is a predetermined positive integer, if $dl_{k-1} < M_0$, the algorithm stops after the kth iteration, and the predicted labels $[Label_k(1), \ldots, Label_k(N_2)]$ of the test set are the final results. Otherwise, perform the $k+1$th iteration.

Note that three parameters need to be preset: α (the percentage of the test set used for retraining), M_0 (the number of testing trials with inconsistent labels in two consecutive iterations), and K_0 (the number of standard EM iterations). These parameters are generally set based on empirical evaluations. According to our extensive simulations, α and M_0 can be set to be 80% and $0.05 * N_2$, respectively; K_0 can be chosen in $\{1, 2, 3\}$ ($K_0 = 3$ in this article). Small adjustments can be made to the above parameters according to the convergence property of the algorithm. The rule of thumb is that if the algorithm converges smoothly after, for example, 20 iterations, we deduce that these parameter settings are reasonable.

The fundamental reason behind our choice of the CSP feature reextraction is that the initial training data set is too small to give a reliable estimation of the CSP transformation matrix in semisupervised learning and unsupervised learning. We can make use of the test set along with the predicted labels to augment the training set and improve the efficiency of feature extraction. Obviously there exists prediction error of the labels. Trials with incorrect labels are considered as noise in the estimation of the CSP transformation matrix. A necessary condition is that the CSP feature should be fairly robust to noise. As analyzed in the previous section, the CSP feature is indeed robust to noise to some extent. Furthermore, the higher the prediction accuracy rate (i.e., the smaller the noise), the better

the CSP feature quality is. Usually the prediction accuracy of labels for the test set is not high initially; thus, the extracted CSP features do not have sufficient consistency, and we need to update the CSP features during later iterations.

As will be seen in our experimental data analysis, the CSP feature reextraction can actually improve the Fisher ratio between the two sets of the CSP features corresponding to the two classes. A higher Fisher ratio implies higher consistency of features and a higher classification efficiency.

The CSP feature reextraction is also motivated by the dynamic characteristics of EEG signals. Even if there is a sufficient training data set, the parameters related to feature extraction of a real-time BCI system should be adjusted if necessary. In the above iterations, the CSP transformation matrix and classifier parameters are kept updated without new system training. This method can be used to improve the adaptability of a BCI system.

Remark 2. (i) Before applying algorithm 1 to an EEG data set, we need to perform data preprocessing, including common average reference (CAR) spatial filtering, frequency filtering, and channel selection (see section 4). From our experience, CAR spatial filtering based on all available channels can improve the accuracy rate. This may be due to denoising. (ii) There are two differences between a standard EM algorithm and the proposed extended EM algorithm. First, our extended EM algorithm is embedded with a CSP feature reextraction. Second, in our extended EM algorithm, only some of the testing data are selected for retraining according to the merits of their classification probabilities. Our experimental analysis results will show that the above two extensions over the standard EM algorithm can improve prediction accuracy significantly.

3.2 Unsupervised and Semisupervised Learning for Online BCI Systems. In the previous section, we presented the extended EM algorithm (algorithm 1) for the semisupervised case. In this section, we first discuss how algorithm 1 can be applied in the unsupervised case. This is followed by a brief discussion of how this algorithm can be used to improve the adaptability of online BCI systems.

Unsupervised learning for a BCI system implies that there is no training phase, that is, no prior training data are available. In the previous section, algorithm 1 was presented for the semisupervised case, but it can be easily extended to the unsupervised case. For off-line data analysis, since an initial training data set is unavailable, we first assign random labels to all the testing trials. The random labels and testing data are then used as the initial training data set. Next, we start the extended EM iterations. In each iteration, the training data set is updated by selecting some testing trials with labels that are predicted in the previous iteration. For the online case,

the initial parameter values of a BCI system can be set default (or as those set previously). The EEG data and real labels are stored while the system is working. When the stored data are sufficient, we use the extended EM algorithm to learn the parameters of the BCI system and update the old ones.

In the semisupervised learning case, there is usually a short training phase. The small number of training trials are used as the initial training data for the extended EM algorithm. The initial parameters of the BCI system can be determined based on the small training data set and updated online, as in the unsupervised case. There exist two ways to set the initial data set for the learning of the parameters. The initial training data set can include only the small training data set with labels or both the small training set with labels and the test set with the labels obtained online.

For both the unsupervised case and the semisupervised case, the labels of the data obtained online may not reflect the user's true intents because of inappropriate BCI system parameters or some other reason. Errors may exist in real-time labels. Thus, we use these labels only as initial values in the online case and update them iteratively in our algorithm (note that in the semisupervised case, the test data set with online labels can also be used as part of the initial training data set).

After the parameters of a BCI system (CSP transformation matrix and classifier parameters) are determined by several iterations, the extracted CSP features have sufficient consistency for classification. Furthermore, the labels obtained online may well reflect the user's true intents. However, this consistency of features may exist only in the short term. When an online BCI system has been used for extended periods, the subject's brain state may change significantly. If so, the consistency (or quality) of the CSP features and classification accuracy will be deteriorated. In this case, the system parameters need to be adjusted to keep or recover the consistency of the CSP features.

Note that the system parameters are updated during the iterations of algorithm 1. Once the iterations terminate, these parameters are determined accordingly. Similar to how the system parameters are determined when the system starts working initially, the proposed extended EM algorithm can also be used to adjust the system parameters online to improve the adaptability of a BCI system.

We now present a simulation example to explain why our algorithm is effective even for the unsupervised case.

Example 1: We generated two artificial data sets, each containing 500 3-by-100 random matrices. Note that each random matrix corresponds to a single trial of EEG data. Each data set is equally divided into two parts corresponding to two classes. The two parts of the first data set are drawn from two uniform distributions with means of 0.5 and 1, respectively; the two parts of the second data set are drawn from two gaussian distributions with means of 0 and 0.5, respectively.

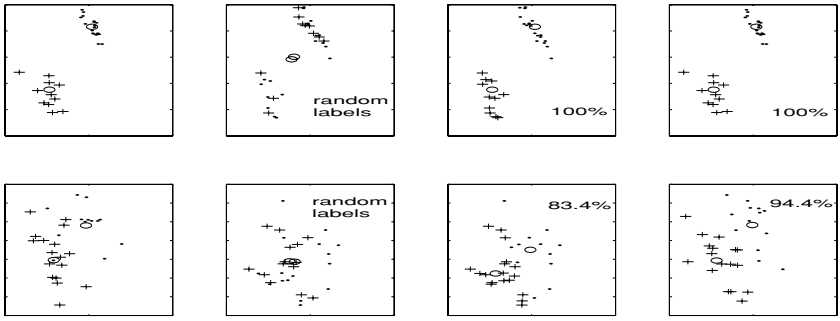

Figure 1: Analysis results for two artificial data sets in the unsupervised case. The two rows correspond to the first and second data sets, respectively. Column 1: True CSP features with true labels. Column 2: CSP features with randomly given labels. Column 3: Features and labels obtained in the first iteration. Column 4: Features and labels obtained in the last iteration. The two circles in each subplot represent two class means.

Assuming that the labels of these data are unknown, we apply our algorithm to predict the labels for all these data matrices. In the first iteration, we randomly assign labels to the data. After extracting three-dimensional CSP features for each data matrix, we predict their labels. These predicted labels are used in the next iteration, and the cycle repeats itself. Feature reextraction and classification are executed in each iteration.

Figure 1 shows our simulation results. The plots in the first and second row correspond to the first and second data set, respectively. From each data set, we first extract three-dimensional CSP features for all data matrices using true labels. These features serve as ground truth for us to compare. Each subplot shows only the first 30 features with labels in Figure 1, and each data point is a two-dimension vector composed by the first two entries of a feature vector. However, the prediction accuracy rates and class means shown below are obtained from all 500 data points. The two circles in each subplot represent the two class means. The two subplots in the first column show the true features with true labels for the two data sets, respectively. We can see that the features of the first data set are separable, while the features of the second data set are overlapped. The second column shows the CSP features derived from random labels. For the first data set, we can see that two separable clusters are obtained, although feature extraction is based on these random labels. The third column depicts the labels obtained in the first classification, noting that the features are the same as those in the second row. The accuracy rates are 100% and 83.4% for the two data sets, respectively. Although the feature extraction and classification here are based on random labels, the prediction accuracy rates obtained are high, especially for the first data set. This is because the features for each data set form two clusters that are somewhat separable. The fourth row shows the

final results. For the first data set, all features and their labels (obtained in the last iteration) are identical to the true ones shown in the first subplot in the first row. For the second data set, the final accuracy rate is 94.4%, noting that the features are different from the true values shown in the first subplot in the second row. Additionally, if a standard EM algorithm (without feature reextraction) is applied to the features extracted from the second data set based on random labels, the classification accuracy is 89.1%.

We also would like to point out that if the semisupervised version of our algorithm is applied to the above two data sets, we can obtain similar results. Note that the first feature extraction and classification are based on the true labels of a small training set rather than randomly given labels.

3.3 Convergence of the Extended EM Algorithm. Although it is local, convergence is an attractive property of a standard EM algorithm. As we pointed out in section 1, it is difficult to have a high-quality CSP feature when the training data are insufficient. If a standard EM is used for classification, CSP features with low consistency (or a low Fisher ratio) may degrade or limit the algorithm performance. In algorithm 1, CSP feature reextraction is embedded in a standard EM algorithm. This may give rise to a convergence problem, which is discussed in this section.

Xu and Jordan (1996) proved that in the EM iterations of equation 3.2, the directions of mean vectors and covariance matrices are the corresponding gradient directions of a log likelihood premultiplied by a positive definite matrix. That is, the log likelihood will increase along with each iteration direction. This guarantees the convergence of a standard EM algorithm.

In this article, the standard EM algorithm in equation 3.2 has been extended for joint CSP feature extraction and classification. Since the feature vectors are updated in each iteration, the log likelihood may not increase as the iterations proceed. However, from our experimental data analysis, the extended EM algorithm still converges.

In the following, we analyze the convergence of the extended EM algorithm. We consider only the unsupervised case, in which all the test data are used in retraining.

Suppose that the labels of test data are known initially. Using these labels, we extract the CSP features of the test data by jointly diagonalizing the two matrices $\Sigma^{(1)}$ and $\Sigma^{(2)}$ (see equations 2.1 to 2.5). In the following, these CSP features, denoted as $\mathbf{cf}^{(q)}(i)\,(q = 1, 2)$, are treated as the true features, which are not affected by prediction error. We now analyze the average error between the true CSP features $\mathbf{cf}^{(q)}(i)$ and those extracted during the iterations of our algorithm.

In this article, noise comes from the classification error in each iteration. For the kth iteration, we denote $\Sigma_k^{(q)}$ as the normalized correlation matrices corresponding to the two classes (similarly calculated as in equation 2.1) and denote the extracted CSP feature vector as $\mathbf{cf}_k^{(q)}(i) = [cf_k^{(q)}(1, i), \ldots, cf_k^{(q)}(L, i)]^T$, where $q(= 1, 2)$ refers to the qth class, and i is

the trial index. We have the following theorem with respect to the mean feature vectors.

Theorem 2.

$$\|mean(\mathbf{cf}_k^{(q)}) - mean(\mathbf{cf}^{(q)})\|_2 \leq L \left\| \Sigma_k^{(q)} - \Sigma^{(q)} \right\|_2, \quad q = 1, 2. \tag{3.5}$$

The proof is given in appendix C.

Let us recall algorithm 1. In the kth iteration, suppose that the prediction accuracy is $rate_k$. From theorem 2, if $rate_{k+1} > rate_k$, that is, the number of misclassfied trials decreases, then the error matrices in equation 2.6 become smaller. Thus, $\| \Sigma_k^{(q)} - \Sigma^{(q)} \|_2 < \| \Sigma_{k+1}^{(q)} - \Sigma^{(q)} \|_2$ $(q = 1, 2)$, that is, the bounds of $\|mean(\mathbf{cf}_k^{(q)}) - mean(\mathbf{cf}^{(q)})\|_2$ should decrease.

Note that before K_0 standard EM iterations in the $(k+1)$th iteration, the classification accuracy rate is $rate_k$, while the accuracy rate $rate_{k+1}$ is obtained by the K_0 standard EM iterations. Due to the performance of the standard EM algorithm, $rate_{k+1} > rate_k$ in general. If the improvement of accuracy rate between two successive iterations is sufficiently large, the bounds in equation 3.5 will decrease greatly. This may make the errors $\|mean(\mathbf{cf}_k^{(q)}) - mean(\mathbf{cf}^{(q)})\|_2$ decrease. This monotonically decreasing phenomenon will be seen in our real data analysis in section 4. We will provide an explanation here. Furthermore, if the prediction rate approaches 1, then $mean(\mathbf{cf}_k^{(q)})$ will approach $mean(\mathbf{cf}^{(q)})$.

Remark 3. From theorem 1, although we can conclude that the CSP features corrupted by noise will tend to the uncorrupted ones when the noise tends to zero, it is difficult to give an error bound of CSP feature with respect to noise. The errors given in equation 3.5 can be seen as the average error bounds of the CSP feature obtained in each iteration.

We now give the average error bounds for the variances of the CSP features.

For the kth iteration, $\sigma_k^{(q)}(j)$ denotes the variance of $cf_k^{(q)}(j, \cdot)$, the jth element of a CSP feature vector belonging to the qth class, $M_k^{(q)}$ denotes the number of these feature vectors, and $M = \max_{q,k}\{M_k^{(q)}\}$, $\sigma^{(q)}(j)$ denotes the variance of $cf^{(q)}(j, \cdot)$. We have

Theorem 3. *If the prediction accuracy $rate_k$ in the kth iteration is sufficiently large, then*

$$\left| \left[\sigma_k^{(q)}(j)\right]^2 - \left[\sigma^{(q)}(j)\right]^2 \right| < 2(M+1) \left\| \Sigma_k^{(q)} - \Sigma^{(q)} \right\|_2 + (1 - rate_k), \tag{3.6}$$

where $j = 1, \ldots, L, q = 1, 2$.

A sketch of the proof of theorem 3 is given in appendix D.

From theorem 3, we can see that if the prediction accuracy rate is sufficiently close to 1, then $\sigma_k^{(q)}(j)$ will tend to $\sigma^{(q)}(j)$.

Using equation 3.6, we can further estimate the bound of $\|\mathbf{Var}_k^{(q)} - \mathbf{Var}^{(q)}\|_2$, where $\mathbf{Var}_k^{(q)}$ and $\mathbf{Var}^{(q)}$ are the covariance matrices of $\{\mathbf{cf}_k^{(q)}(j)\}$ and $\{\mathbf{cf}^{(q)}(j)\}$, respectively. Due to limited space, we omit the estimate here.

4 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. Muller and 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 description in the following paragraph is from http://ida.first.fraunhofer.de/projects/bci/competition_iii).

This data set was recorded from 118 scalp electrodes at a sampling rate of 1000 Hz from five healthy subjects. Subjects sat in a comfortable chair with arms resting on armrests. This data set contains only data from the four initial sessions without feedback. Visual cues indicated for 3.5 s which of the following two motor imageries the subject should perform: (R) right hand, (F) right foot. The presentations of target cues were separated by periods of random length, 1.75 to 2.25 s, in which the subject could relax. There were two types of visual stimulation: (1) targets were indicated by letters appearing behind a fixation cross (which might nevertheless induce small target-correlated eye movements) and (2) a randomly moving object indicated targets (inducing target-uncorrelated eye movements). For the second and fourth subjects ("al" and "aw"), two sessions of both types were recorded, while for the other three subjects ("aa," "av," and "ay"), three sessions of type 2 and one session of type 1 were recorded. The data were also down-sampled from 1000 Hz into 100 Hz. We use this 100 Hz version in this article.

Due to limited space, we present our detailed analysis results for only three subjects: aa, al, and ay. For convenient analysis, we do not use the competition splitting of data sets. We run our extended EM algorithm using the first 150 trials of the data set for each subject. In the semisupervised case, we perform fivefold cross-validation in which only one fold of data (30 trials) with labels is used for the initial training data set; the other four folds (120 trials) without labels are used for the test data set and retraining. In the unsupervised case, we do not use any labels from the 150 trials. For further demonstration, we use the subsequent 80 trials as an independent test set. Note that in the competition, the number of trials of the five training

sets are 168 (subject aa), 224 (subject al), 84 (subject av), 56 (subject aw), and 28 (subject ay). Except for the fifth training set, the other four training sets are much larger than those used in this article.

In the following, we give a description of preprocessing and then consider the cases of semisupervised learning and unsupervised learning.

4.1 Preprocessing. To ensure good performance, appropriate preprocessing is necessary. The preprocessing in this article includes CAR spatial filtering, frequency filtering, and channel selection.

For every trial, we use data of duration 3.5 sec for analysis. During this period, the cue was visible on the screen, so we have 350 samples for each trial. We then obtain a 118×350 EEG data matrix denoted as E_k for the kth trial. The whole EEG data matrix is given by $\mathbf{E} = [\mathbf{E}_1, \ldots, \mathbf{E}_{N_1}, \mathbf{E}_{N_1+1}, \ldots, \mathbf{E}_{N_1+N_2}]$, where $[\mathbf{E}_1, \ldots, \mathbf{E}_{N_1}]$ denotes the training data set with N_1 trials, and $[\mathbf{E}_{N_1+1}, \ldots, \mathbf{E}_{N_1+N_2}]$ denotes the test set with N_2 trials.

Notice that for the unsupervised case, prior training data are unavailable. All the training data come from the test set. In the semisupervised case, the relatively small initial training data set contains 30 trials.

EEG data matrix $\mathbf{E}$ is first preprocessed by a CAR spatial filter, and the resultant data matrix is denoted as $\bar{\mathbf{E}}$. This filter is useful in reducing some artifacts and noise. After the filtering, a spectral analysis for every EEG channel in the training data (every row of the training data matrix $[\mathbf{E}_1, \ldots, \mathbf{E}_{N_1}]$) is performed. In order to select the proper frequency band, we calculate the Fisher ratio at each frequency bin of the power spectra for each channel. Based on the Fisher ratios, we roughly determine the frequency band (typically in mu or beta bands). This frequency band may be different from subject to subject. In this article, we use only the signals in mu band. The selected frequency bands for the three subjects in our study are 12 Hz to 14 Hz for subjects aa and al and 9 Hz to 13 Hz for subject ay. After determining the frequency band, we further roughly select EEG channels (generally in the sensorimotor area) that exhibit relatively higher Fisher ratios in the determined frequency band. The number of selected channels is denoted as N_0. Note that the above frequency band and channel selection are based on the small training data set in the semisupervised case. For the unsupervised case, we can first use the default settings of the mu frequency band (e.g., 10–14 Hz) and channels (the channels in sensorimotor area); then adjustment is made according to the classification results obtained in the algorithm iterations.

4.2 Semisupervised Learning Case. In this section, we apply the extended EM algorithm to the semisupervised learning case.

As an example, we first describe our analysis procedure and results for subject aa. As stated in the beginning of section 4, we use the first 150 trials with labels for cross-validation and use the subsequent 80 trials as an independent test set. We equally divide the 150 trials into five folds

according to their sequential order. To evaluate the effect of a small training set, we use one fold for the initial training data set. The other four folds, which are used for learning/retraining and testing, are called learning test set in order to distinguish it from the independent test set. The independent test set is to further demonstrate the validity of our algorithm. The process is formulated in this way. In each iteration, besides performing all the tasks stated in our extended EM algorithm, we also extract the CSP features of the independent test set, predict their labels, and calculate the corresponding prediction accuracy rate.

We have two different percentage settings for our extended EM algorithm: (1) 80% of the learning test set is used for retraining and (2) 100% of the learning test set is used for retraining.

In each iteration, we calculate the prediction accuracy rates $accuracy(i, k, j)$ for the learning test set and $accuracy_I(i, k, j)$ for the independent test set, where $i = 1, 2$ refer to the percentage parameters 80% and 100%, respectively; k represents the kth iteration; and $j (= 1, \ldots, 5)$ represents the jth fold used for the initial training set. The average accuracy rates over all folds are calculated as

$$rate(i, k) = \frac{1}{5} \sum_{j=1}^{5} accuracy(i, k, j) \tag{4.1}$$

$$rate_I(i, k) = \frac{1}{5} \sum_{j=1}^{5} accuracy_I(i, k, j), \tag{4.2}$$

where $i = 1, 2$, $k = 1, \ldots, 9$ for subject aa.

For the purpose of comparison, we calculate all these accuracy rates similar to equation 4.1 except that CSP feature reextraction is skipped during the retraining. This corresponds to the performance of a standard EM algorithm. The obtained average accuracy rates for the learning test set are denoted as $\bar{rate}(i, k)$ $(i = 1, 2, k = 1, \ldots, 9)$. From the comparison, we can observe how the feature reextraction contributes to accuracy.

The above analysis results are shown in the first row of Figure 2. In the first subplot, $rate(1, k)$ is depicted as a solid line with asterisks and $rate(2, k)$ as a solid line with circles. Similarly, $\bar{rate}(1, k)$ and $\bar{rate}(2, k)$ are depicted as dotted lines with stars and circles, respectively. Note that these results are obtained from the learning test set. In the second subplot, accuracy rates $rate_I(1, k)$ and $rate_I(2, k)$ for the independent test set are depicted as solid lines with asterisks and circles, respectively.

Next, we present our analysis results on the convergence of our algorithm. We consider only the case in which 80% of the learning test set is used for retraining. We define $mean_q(k, j)$ and $\text{var}_q(k, j)$ as the mean vectors and variance matrices of a Bayes classifier, where $q = 1, 2$ are class indices, $k = 1, \ldots, 9$ are iteration indices, and $j = 1, \ldots, 5$ refer to the jth

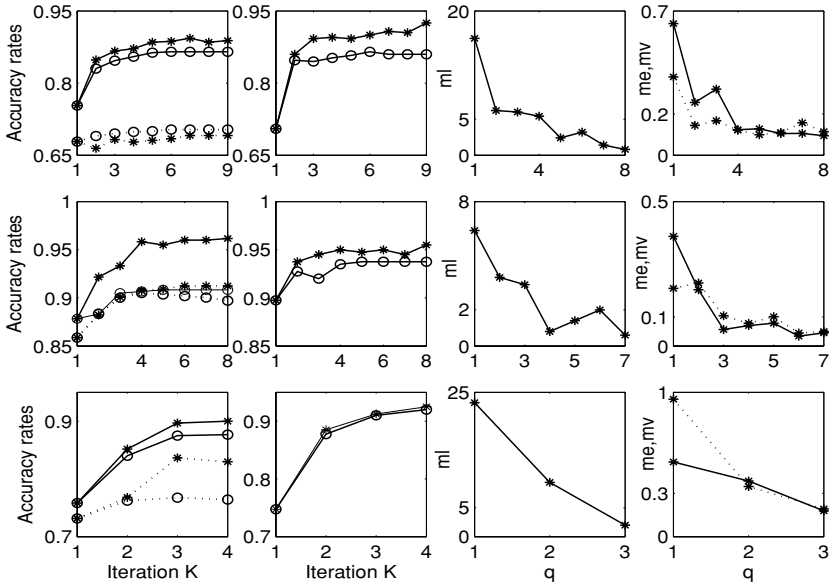

Figure 2: Analysis results in the semisupervised case. The first, second, and third rows are for subjects aa, al, and ay, respectively. The first column shows prediction accuracy rates for the learning test set obtained by our algorithm (solid lines) and the standard EM algorithm (dotted lines), where the lines with asterisks and circles refer to the percentage settings of 80% and 100%, respectively. The second column shows prediction accuracy rates obtained by our algorithm for the independent test set, where the lines with asterisks and circles refer to the percentage settings of 80% and 100%, respectively. The third column depicts the curves of label convergence index ml in equation 4.5. The fourth column shows the curves of average errors for the mean (solid lines) and covariance (dotted lines) of the classifier.

fold used for the initial training data set. We can find the average difference of $mean_q(k, j)$ and $\mathrm{var}_q(k, j)$ between two successive iterations over five folds and two classes as follows,

$$me(k) = \frac{1}{2}\sum_{q=1}^{2}\frac{1}{5}\sum_{j=1}^{5}\|mean_q(k, j) - mean_q(k+1, j)\| \tag{4.3}$$

$$mv(k) = \frac{1}{2}\sum_{q=1}^{2}\frac{1}{M}\sum_{j=1}^{5}\|var_q(k, j) - var_q(k+1, j)\|, \tag{4.4}$$

where $k = 1, \ldots, 8$ for subject aa. In equations 4.3 and 4.4, $\|\cdot\|$ represents the Frobenius norm of a vector or a matrix.

We can also observe the convergence from the consistency of labels for the learning test set predicted in two successive iterations. Let $Label(k, j, \cdot)$ be the label vector predicted in the kth iteration, where j $(= 1, \ldots, 5)$ refers to the jth fold used for the training data set. We calculate the average number of different labels between two successive iterations over five folds, which we call the label convergence index,

$$ml(k) = \frac{1}{5} \sum_{q=1}^{5} \sum_{n=1}^{N_2} |Label(k, j, n) - Label(k+1, j, n)|, \tag{4.5}$$

where $k = 1, \ldots, 8$, N_2 is the number of predicted labels (i.e., the number of testing trials).

ml is shown in the third subplot of the first row of Figure 2, and *me* and *mv* are shown in the fourth subplot with the solid and dotted line, respectively. These iterative curves in these two subplots illustrate the convergence of our algorithm.

For subjects al and ay, we performed a similar analysis. The corresponding results are presented in the second and the third rows of Figure 2, respectively. Note that the numbers of iterations for subjects al and ay are 8 and 4, respectively.

Remark 4. Besides the above three subjects, we also applied our algorithm to the data sets from the other two subjects (aw and av). Under the same settings as above, the final average prediction accuracy rates of the learning test sets are 91.2% and 76.4% for the two subjects aw and av, respectively, while the final corresponding average accuracy rates of the independent test sets are 88.8% and 75.3%. The result for subject av is not so satisfactory as those for the other subjects here. This case also happened for the results obtained by the winner in the BCI2005 competition. We think it is due to the quality of the data.

4.3 Unsupervised Learning Case. In this section, we consider the unsupervised learning case. In this article, unsupervised learning for a BCI system implies that there are no initial training data. Our extended EM algorithm can be used in the unsupervised learning case as stated in section 3.2. We evaluate our algorithm using data for subjects aa and al due to limited space, although we have also obtained satisfactory results for the other data sets. For each of the two subjects, we use the first 150 trials for the learning test set and the subsequent 80 trials for the independent test set. In the initialization step, we assign random labels to the learning test set, extract the CSP features of the learning test set according to these random labels, and set the initial values of the Bayes classifier

parameters. Next we apply the extended EM algorithm. In each iteration, we extract the CSP features for both the learning test and the independent test set and predict their labels. We have two iteration settings: (1) 80% of the learning test trials are used for the training set in each iteration, and (2) 100% of the learning test trials are used for the training set in each iteration.

First, we calculate the prediction accuracy rates, $rate(i,k)$, for the learning test set and $rate_I(i,k)$ for the independent test set, where $i = 1, 2$ represent the percentage settings of 80% and 100%, respectively, and k refers to the kth iteration. Next, we calculate the number of different labels for the learning test set between two successive iterations, which is the indicator for terminating the iteration,

$$dL_i(k) = \sum_{n=1}^{N_2} |Label_i(k,n) - Label_i(k+1,n)|, \tag{4.6}$$

where $Label_i(k,\cdot)$ is the label vector predicted in the kth iteration and under the ith ($i = 1, 2$) percentage setting. N_2 is the number of predicted labels.

We now consider the convergence of the algorithm. From theorems 2 and 3, we can conclude that the mean vectors and covariance matrices will tend to the true ones if the improvement of prediction accuracy in each iteration is sufficiently large.

We now demonstrate this conclusion by data analysis results. For each of the two subjects, we use all 150 trials of the learning test set and their true labels to extract the CSP feature vectors $\mathbf{cf}^{(q)}(j)$, where $q = 1, 2$ is the class index (i.e., label), and j refers to the jth trial of the qth class. The covariance matrix of $\{\mathbf{cf}^{(q)}(j)\}$ is denoted as $\mathbf{Var}^{(q)}$. These CSP feature vectors, the mean vectors, and class covariance matrices are treated as true ones without being affected by prediction error.

We then calculate the average errors for mean vectors and covariance matrices,

$$ME_i(k) = \frac{1}{2}\sum_{q=1}^{2} \left\| mean\left(\mathbf{cf}_k^{(q)}(\cdot)\right) - mean\left(\mathbf{cf}^{(q)}(\cdot)\right) \right\|_2, \tag{4.7}$$

$$MV_i(k) = \frac{1}{2}\sum_{q=1}^{2} \left\| \mathbf{Var}_k^{(q)} - \mathbf{Var}^{(q)} \right\|_2, \tag{4.8}$$

where i refers to the ith percentage setting; the notations $\mathbf{cf}_k^{(q)}(j)$ (CSP feature vector) and $\mathbf{Var}_k^{(q)}$ (covariance matrix) can be seen in section 3.3.

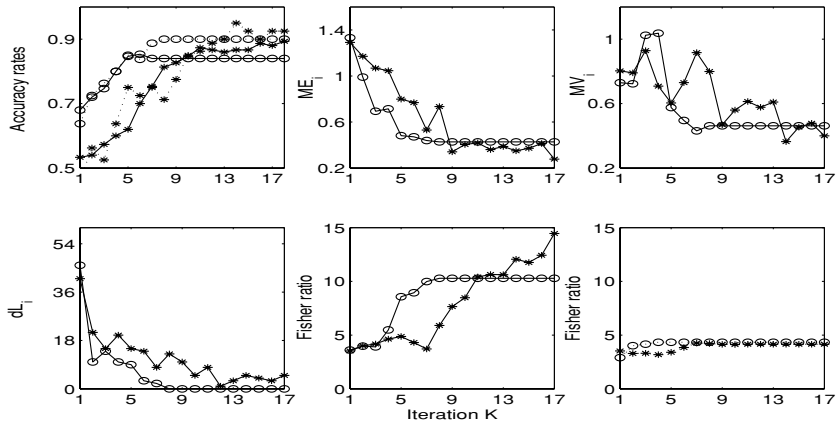

Figure 3: Analysis results for subject aa in the unsupervised case. In the first row, the left subplot shows curves of prediction accuracy rates for the learning test set (solid lines) and the independent test set (dotted lines). Note that for all subplots in this figure, the percentage settings of 80% and 100% are represented by asterisks and circles, respectively. The middle and right subplots in the first row show the curves of average errors for mean ($ME_i(k)$ in equation 4.7) and covariance ($MV_i(k)$ in equation 4.8) of the classifier. The subplots in the second row show curves of label convergence index $dL_i(k)$ in equation 4.6 (left subplot), Fisher ratios obtained from the 150 trials of the learning test set by our algorithm (middle subplot), and Fisher ratios obtained from the same data set by the standard EM algorithm (right subplot).

To further demonstrate that feature reextraction in the iterations can improve the consistency of features, we calculate the Fisher ratios,

$$FR_i(k) = \frac{\left\| mean\left(\mathbf{cf}_k^{(1)}(\cdot)\right) - mean\left(\mathbf{cf}_k^{(2)}(\cdot)\right) \right\|}{\left(\left\| \mathbf{Var}_k^{(1)} \right\| + \left\| \mathbf{Var}_k^{(2)} \right\| \right)^{\frac{1}{2}}}, \tag{4.9}$$

where i implies the ith percentage setting and k implies the kth iteration.

For comparison, we perform several standard EM iterations (without feature reextraction) using the CSP features extracted in the third iteration of our algorithm. Similar to equation 4.9, we also calculate the Fisher ratios, which are denoted as $\bar{FR}_i(k)$.

Figure 3 shows the above analysis results for subject aa. In the first row, the left subplot shows the curves of average accuracy rates, $rate(i,k)$ for the learning test set in solid lines, and $rate_I(i,k)$ for the independent test set in dotted lines. Note that for all subplots in this figure, the settings of 80% and 100% are represented by asterisks and circles, respectively. The average errors of mean vectors $ME_i(k)$ and covariance matrices $MV_i(k)$ are shown

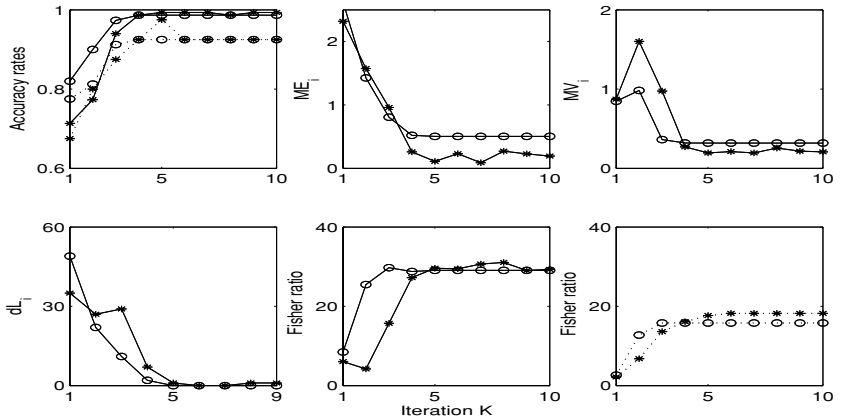

Figure 4: Analysis results for subject al in the unsupervised case. In the first row, the left subplot shows the curves of prediction accuracy rates for the learning test set (solid lines) and the independent test set (dotted lines). Note that for all subplots in this figure, the percentage settings of 80% and 100% are represented by asterisks and circles, respectively. The middle and right subplots in the first row show the curves of average errors for the mean ($ME_i(k)$ in equation 4.7) and covariance ($MV_i(k)$ in equation 4.8) of the classifier. The three subplots in the second row show the curves of label convergence index $dL_i(k)$ in equation 4.6 (left subplot), Fisher ratios obtained from the 150 trials of the learning test set by our algorithm (middle subplot), and Fisher ratios obtained from the same data set by standard EM algorithm (right subplot).

in the middle and right subplots, respectively. The iteration-terminating indicator $dL_i(k)$, Fisher ratios $FR_i(k)$, and $\overline{FR}_i(k)$ obtained from the 150 trials of learning test set are shown in the three subplots in the second row, separately.

We perform data analysis for subject al as was done for subject aa. The corresponding results are shown in Figure 4.

4.4 Discussion. In this section, we present our discussions based on the experimental analysis results shown in Figures 2, 3, and 4 for both the semisupervised and the unsupervised cases.

1. Our analysis results for the semisupervised case are shown in Figure 2, in which the three rows correspond to the three subjects' data. The first and second column of Figure 2 display accuracy rate curves for the learning test set and the independent test set, respectively. Figures 3 and 4 present our analysis results for the unsupervised case for subjects aa and al. Accuracy rate curves for both the learning test set and the independent test set are shown in the first subplots of Figures 3 and 4.

From all these accuracy curves obtained by the extended EM algorithm under the percentage settings of 80% (solid lines with asterisks in the first two columns of Figure 1, lines with asterisks in the first subplots of Figures 2 and 3), we see that satisfying prediction accuracy rates are obtained by several iterations of our proposed algorithm. The validity of our algorithm is thus demonstrated. This suggests that the extended EM algorithm presented in this article may be used in a BCI system when training data are not sufficient or are even unavailable.

2. From the analysis results shown in the three subplots of the first column in Figure 1, we see that the highest accuracy rates are obtained when the percentage of the learning test data for retraining is 80% instead of 100%. In one iteration, the higher the posterior probability of a trial from the learning test set, the more confident its predicted label. Thus, we do not use those trials with very low posterior probabilities for retraining. Therefore, for the semisupervised case, an appropriate percentage setting can improve the performance of our algorithm. This phenomenon is also suggested by Figures 3 and 4 for the unsupervised case. Thus, we recommend setting the percentage to be less than 100%. One question might be how this percentage can be determined. Through extensive experiments, we found that for the percentages in a broad range (e.g., from 60% to 90%), the performance does not vary too much. To choose a suitable value, we can start with an initial value of 80%, for example. If the iterations converge smoothly, we take this percentage value as our choice. Otherwise, we search for another one.

3. We now compare the accuracy rates obtained by our extended EM algorithm, which is embedded with a feature reextraction, with the accuracy rates obtained by the standard EM algorithm in equation 3.3. From the three subplots of the first column of Figure 1, we can see that $rate(1, k)$, $rate(2, k)$, and $rate(3, k)$ (solid curves) are higher than $\bar{rate}(1, k)$, $\bar{rate}(2, k)$, and $\bar{rate}(3, k)$ (dotted curves), respectively. This means that feature reextraction can improve the performance of classification for the semisupervised case. In the unsupervised case, since the initial labels are given randomly, it is theoretically necessary to reextract the CSP feature during the iterations.

For any two consecutive iterations, feature reextraction in the first iteration can improve the feature consistency (expressed by Fisher ratio; see the later discussion), and this leads to a higher classification accuracy. The latter will result in a further improvement of feature consistency after feature reextraction in the next iteration.

4. We consider iteration convergence in the semisupervised case. From all subplots in the second and third columns in Figure 2, the parameters (mean vectors and variance matrices) of the Bayes classifiers and the predicted label vectors are convergent for all three subjects. This implies a satisfying convergence property of our algorithm and also demonstrates the validity of our criterion for the termination of iterations. Convergence analysis for the semisupervised version of our algorithm can be similar to theorems 2 and 3.

On iteration convergence in the unsupervised case, theorems 2 and 3 and their corresponding proofs (see the appendixes) tell us that the mean vectors and covariance matrices of CSP feature vectors for both classes will tend to the true ones (unaffected by noise) if the improvement of prediction accuracy in each iteration is sufficiently large. This has been demonstrated in the second and third subplots in the first rows of Figures 3 and 4. In addition, the first subplots in the second rows of Figures 3 and 4 show the convergence of the predicted label vectors.

5. From the second subplots in the second rows of Figures 3 and 4, we can find that the Fisher ratios between two classes of CSP features can be improved significantly during the extended EM iterations. By comparing the Fisher ratio curves (obtained by the standard EM iterations) shown in the third subplots in the second rows of Figures 3 and 4 with those shown in the second subplots, feature reextraction causes a significant improvement of the Fisher ratios. Thus, our method of retraining with feature reextraction can improve both the classification performance and the quality of feature.

5 Conclusion

In this article, we present an extended EM algorithm that can be used for both semisupervised learning and unsupervised learning in BCI systems. The first objective is to reduce or even skip the training phase entirely. The second objective is to improve the adaptability of BCI systems.

Two key problems—the robustness of the CSP feature to noise and algorithm convergence—are addressed here. In our proposed algorithm, the labels predicted in each iteration are used for reextracting the CSP features. Since prediction error of the labels (treated as noise in this article) is inevitable, we need to consider the robustness of the CSP feature to noise. According to our analysis, the feature is somewhat robust to noise. Furthermore, during the iterations of the extended EM algorithm, if the prediction accuracy rates tend to one, then the reextracted CSP features will tend to the true values, which are unaffected by prediction errors. It is well known that the convergence plays a key role for an iterative algorithm to work. From our theoretical and experimental data analysis, our extended EM algorithm also has satisfying convergence property. This is due to the convergence property of the standard EM algorithm and the robustness of CSP feature to noise.

The main difference between our algorithm and a standard EM algorithm is that there is a feature reextraction in each iteration of our algorithm. When the initial training data set is small or null, the CSP features extracted in the beginning have low consistency and thus are not reliable. According to our analysis results, feature reextraction can improve the consistency (expressed by Fisher ratio) of CSP features and classification accuracy.

Appendix A: Proof of Lemma 2

The second conclusion can be obtained directly from lemma 1.

We now prove the first conclusion. Note that equation 2.10 is equivalent to

$$(\mathbf{A}+\mathbf{\Theta})\mathbf{g}_i(\theta)=q_i(\theta)\mathbf{g}_i(\theta),\ \ i=1,\ldots,m, \tag{A.1}$$

where $\mathbf{g}_i(\theta)$ is the ith column vector of $G(\theta)$.

Noting that $\mathbf{g}_i^T(\theta)\mathbf{g}_i(\theta)=1$, the vector function $\mathbf{g}_i(\theta)$ is bounded. Thus, $\mathbf{g}_i(\theta)$ has convergent subsequences. Suppose that $\{\mathbf{g}_i(\theta_j),\ j=1,\ldots,\}$ is a convergent subsequence of $\{\mathbf{g}_i(\theta)\}$, that is, $\lim_{j\to\infty}\theta_j=0$, $\lim_{j\to\infty}\mathbf{g}_i(\theta_j)=\bar{\mathbf{g}}_i$. We first have

$$(\mathbf{A}+\mathbf{\Theta}_j)\mathbf{g}_i(\theta_j)=q_i(\theta_j)\mathbf{g}_i(\theta_j). \tag{A.2}$$

Noting that $\lim_{j\to\infty}q_i(\theta_j)=\lambda_i$, we have

$$\mathbf{A}\bar{\mathbf{g}}_i=\lambda_i\bar{\mathbf{g}}_i. \tag{A.3}$$

It follows from equation A.3 that $\bar{\mathbf{g}}_i$ is an eigenvector of $\mathbf{A}$ corresponding to λ_i. Since $\{\mathbf{g}_i(\theta_j)\}$ are normalized vectors with their first entries being nonnegative, $\|\bar{\mathbf{g}}_i\|_2=1$, and the first entry of $\bar{\mathbf{g}}_i$ is nonnegative. $\mathbf{A}$ has m different eigenvalues and the first entry of $\mathbf{g}_i$ is nonnegative; thus, $\bar{\mathbf{g}}_i=\mathbf{g}_i$. From the above analysis, we can see that any convergent subsequence of $\mathbf{g}_i(\theta)$ tends to $\mathbf{g}_i$. Thus, $\lim_{\theta\to 0}\mathbf{g}_i(\theta)=\mathbf{g}_i$. Lemma 2 is proven.

Appendix B: Proof of Theorem 1

Reconsidering the joint diagonalization procedure of the two noisy correlation matrices $\Sigma^{(1)}+\varepsilon_{\mathbf{1}}$ and $\Sigma^{(2)}+\varepsilon_{\mathbf{2}}$, we have

$$\mathbf{V}^T(\epsilon)\Sigma(\epsilon)\mathbf{V}(\epsilon)=\mathbf{P}(\epsilon), \tag{B.1}$$

where $\Sigma(\epsilon)=\Sigma^{(1)}+\varepsilon_{\mathbf{1}}+\Sigma^{(2)}+\varepsilon_{\mathbf{2}}$, $\mathbf{P}(\epsilon)$ is a diagonal matrix composed by the eigenvalues of $\Sigma(\epsilon)$ in a decreasing order.

Set $\mathbf{U}(\epsilon)=(\mathbf{P}(\epsilon))^{\frac{1}{2}}\mathbf{V}^T(\epsilon)$, $\mathbf{R}_1(\epsilon)=\mathbf{U}(\epsilon)(\Sigma^{(1)}+\varepsilon_{\mathbf{1}})\mathbf{U}^T(\epsilon)$.

Suppose that $\mathbf{Z}(\epsilon)$ is a orthogonal matrix such that

$$\mathbf{Z}^T(\epsilon)\mathbf{R}_1(\epsilon)\mathbf{Z}(\epsilon)=\mathbf{D}(\epsilon)=diag(d_1(\epsilon),\ldots,d_m(\epsilon)). \tag{B.2}$$

Note that the first row vectors of above two orthogonal matrices $\mathbf{V}(\epsilon)$ and $\mathbf{Z}(\epsilon)$ are set to be nonnegative.

Define $\mathbf{W}(\epsilon) = \mathbf{Z}^T(\epsilon)\mathbf{U}(\epsilon)$; then we have equation 2.7.

For practical data, we can say that Σ and $\mathbf{R}_1$ have m different eigenvalues respectively (with probability one).

It follows from lemma 2 that

$$\lim_{\epsilon \to 0} \mathbf{V}(\epsilon) = \mathbf{V}, \ \lim_{\epsilon \to 0} \mathbf{P}(\epsilon) = \mathbf{P}. \tag{B.3}$$

Thus, we have

$$\lim_{\epsilon \to 0} \mathbf{U}(\epsilon) = \mathbf{U}, \ \lim_{\epsilon \to 0} \mathbf{R}_1(\epsilon) = \mathbf{R}_1, \ \lim_{\epsilon \to 0} \mathbf{R}_2(\epsilon) = \mathbf{R}_2. \tag{B.4}$$

From equation B.4 and lemma 2, $\lim_{\epsilon \to 0} \mathbf{Z}(\epsilon) = \mathbf{Z}$ and $\lim_{\epsilon \to 0} \mathbf{D}(\epsilon) = \mathbf{D}$. In view of the definition of $\mathbf{W}(\epsilon)$, $\lim_{\epsilon \to 0} \mathbf{W}(\epsilon) = \mathbf{W}$.

The second conclusion can be directly obtained from equation 2.13 and the definitions of CSP features in equations 2.5 and 2.8.

The theorem is thus proven.

Appendix C: Proof of Theorem 2

Noting that the sum $\Sigma_k^{(1)} + \Sigma_k^{(2)}$ does not change in every iteration, that is, $\Sigma_k^{(1)} + \Sigma_k^{(2)} = \Sigma$, where Σ is the same as in equation 2.2.

We denote $R_k^{(1)} = (\mathbf{P})^{\frac{1}{2}} \mathbf{V}^T \Sigma_k^{(1)} \mathbf{V} (\mathbf{P})^{\frac{1}{2}}$, $R_k^{(2)} = (\mathbf{P})^{\frac{1}{2}} \mathbf{V}^T \Sigma_k^{(2)} \mathbf{V} (\mathbf{P})^{\frac{1}{2}}$, where the matrices $\mathbf{V}$, $\mathbf{P}$ are defined in equation 2.2. We also denote $M_k^{(q)}$ the number of trials belonging to the qth class in the kth iteration and $M^{(q)}$ the true number of trials belonging to the qth class.

Suppose that $R_k^{(1)} = \mathbf{Z}_k^T \mathbf{D}_k \mathbf{Z}_k$, where $\mathbf{Z}_k$ is an orthogonal matrix and $\mathbf{D}_k$ is a diagonal matrix with its elements in the diagonal line in decreasing order. Let $\mathbf{W}_k = \mathbf{Z}_k^T (\mathbf{P})^{\frac{1}{2}} \mathbf{V}^T$; then $\mathbf{W}_k$ can jointly diagonalize the matrices $\Sigma_k^{(1)}$ and $\Sigma_k^{(2)}$. The submatrix $\bar{\mathbf{W}}_k$ is constructed by using the first l_1 rows and the last l_2 rows of $\mathbf{W}_k$; then it is a CSP transformation matrix in the kth iteration (similarly as in equation 2.5).

By the definition of CSP feature,

$$\begin{aligned} \frac{1}{M_k^{(1)}} \sum_{i=1}^{M_k^{(1)}} cf_k^{(1)}(j,i) &= \frac{1}{M_k^{(1)}} \sum_{i=1}^{M_k^{(1)}} \bar{\mathbf{w}}_k(j) \frac{\mathbf{S}_i^{(1)} \left(\mathbf{S}_i^{(1)}\right)^T}{trace\left(\mathbf{S}_i^{(1)} \left(\mathbf{S}_i^{(1)}\right)^T\right)} \bar{\mathbf{w}}_k^T(j) \\ &= \bar{\mathbf{w}}_k(j) \left[\frac{1}{M_k^{(1)}} \sum_{i=1}^{M_k^{(1)}} \frac{\mathbf{S}_i^{(1)} \left(\mathbf{S}_i^{(1)}\right)^T}{trace\left(\mathbf{S}_i^{(1)} \left(\mathbf{S}_i^{(1)}\right)^T\right)} \right] \bar{\mathbf{w}}_k^T(j) \end{aligned}$$

$$= \bar{\mathbf{w}}_k(j)\Sigma_k^{(1)}\bar{\mathbf{w}}_k^T(j)$$
$$= d_k(n_j), \tag{C.1}$$

where $\bar{\mathbf{w}}_k(j)$ is the jth row vector of $\bar{\mathbf{W}}_k$ which is assumed to be the n_jth row of $\mathbf{w}_k(j)$, and $d_k(n_j)$ is the n_jth eigenvalue of $\Sigma_k^{(1)}$ (i.e., the n_jth element of the diagonal line of $\mathbf{D}_k$).

Similarly,

$$\frac{1}{M^{(1)}}\sum_{i=1}^{M^{(1)}} cf^{(1)}(j,i) = \bar{\mathbf{w}}_j\Sigma^{(1)}\bar{\mathbf{w}}_j^T = d(j). \tag{C.2}$$

It follows from equations C.1 and C.2 and lemma 1 that

$$|\frac{1}{M_k^{(1)}}\sum_{i=1}^{M_k^{(1)}} cf_k^{(1)}(j,i) - \frac{1}{M^{(1)}}\sum_{i=1}^{M^{(1)}} cf^{(1)}(j,i)| = |d_k(n_j) - d(n_j)|$$
$$\leq \|\Sigma_k^{(1)} - \Sigma^{(1)}\|_2. \tag{C.3}$$

Furthermore, we have

$$\|mean(\mathbf{cf}_k^{(1)}(\cdot)) - mean(\mathbf{cf}^{(1)}(\cdot))\|_2$$
$$= \left[\sum_{j=1}^{L} |\frac{1}{M_k^{(1)}}\sum_{i=1}^{M_k^{(1)}} cf_k^{(1)}(j,i) - \frac{1}{M^{(1)}}\sum_{i=1}^{M^{(1)}} cf^{(1)}(j,i)|^2\right]^{\frac{1}{2}}$$
$$\leq L\|\Sigma_k^{(1)} - \Sigma^{(1)}\|_2. \tag{C.4}$$

Similarly, we have the following conclusion for the second-class mean vector:

$$\|mean\big(\mathbf{cf}_k^{(2)}(\cdot)\big) - mean\big(\mathbf{cf}^{(2)}(\cdot)\big)\|_2 \leq L\|\Sigma_k^{(2)} - \Sigma^{(2)}\|_2. \tag{C.5}$$

Theorem 2 is proved.

Appendix D: Sketch of Proof of Theorem 3

As in the proof of theorem 2, we denote $M_k^{(q)}$ the number of trials belonging to the qth class in the kth iteration and $M^{(q)}$ the true number of trials belonging to the qth class, $M = \max_{q,k}\{M_k^{(q)}\}$, $m_k^{(q)}(j) = mean(cf_k^{(q)}(j,\cdot))$, $m^{(q)}(j) = mean(cf^{(q)}(j,\cdot))$.

The variances of $cf_k^{(q)}(j,\cdot)$ and $cf^{(q)}(j,\cdot)$ are calculated as

$$\begin{aligned}\left(\sigma_k^{(q)}(j)\right)^2 &= \frac{1}{M_k^{(q)}}\sum_{i=1}^{M_k^{(q)}}\left(cf_k^{(q)}(j,i)-m_k^{(q)}(j)\right)^2\\ \left(\sigma^{(q)}(j)\right)^2 &= \frac{1}{M^{(q)}}\sum_{i=1}^{M^{(q)}}\left(cf^{(q)}(j,i)-m^{(q)}(j)\right)^2,\end{aligned} \tag{D.1}$$

where $j=1,\ldots,L, q=1,2$.

Suppose that in the kth iteration, there are $\bar{M}_k^{(1)}$ trials of the first class with correct labels. Since $rate_k$ is sufficiently large, then $\bar{M}_k^{(1)}$ is close to $M_k^{(1)}$ and $M^{(1)}$. Thus, we have

$$\begin{aligned}&\frac{1}{\bar{M}_k^{(1)}}\sum_{i=1}^{\bar{M}_k^{(1)}} cf_k^{(1)}(j,i)\simeq m_k^{(1)}(j),\\ &\frac{1}{\bar{M}_k^{(1)}}\sum_{i=1}^{\bar{M}_k^{(1)}} cf^{(1)}(j,i)\simeq m^{(1)}(j).\end{aligned} \tag{D.2}$$

Without loss of generality, suppose that $m_k^{(1)}(j)>m^{(1)}(j)$; we have

$$\begin{aligned}&\frac{1}{\bar{M}_k^{(1)}}\left[\sum_{i=1}^{\bar{M}_k^{(1)}} cf_k^{(1)}(j,i)\right]^2-\frac{1}{\bar{M}_k^{(1)}}\left[\sum_{i=1}^{\bar{M}_k^{(1)}} cf^{(1)}(j,i)\right]^2\\ &=\frac{1}{\bar{M}_k^{(1)}}\sum_{i=1}^{\bar{M}_k^{(1)}}\left[\left(cf_k^{(1)}(j,i)\right)^2+cf_k^{(1)}(j,i)\sum_{l\neq i}cf_k^{(1)}(j,l)\right]\\ &\quad-\frac{1}{\bar{M}_k^{(1)}}\sum_{i=1}^{\bar{M}_k^{(1)}}\left[\left(cf^{(1)}(j,i)\right)^2+cf^{(1)}(j,i)\sum_{l\neq i}cf^{(1)}(j,l)\right]\end{aligned} \tag{D.3}$$

and

$$\begin{aligned}&\frac{1}{\bar{M}_k^{(1)}}\sum_{i=1}^{\bar{M}_k^{(1)}}\left[cf_k^{(1)}(j,i)\sum_{l\neq i}cf_k^{(1)}(j,l)\right]-\frac{1}{\bar{M}_k^{(1)}}\sum_{i=1}^{\bar{M}_k^{(1)}}\left[cf^{(1)}(j,i)\sum_{l\neq i}cf^{(1)}(j,l)\right]\\ &\simeq\sum_{i=1}^{\bar{M}_k^{(1)}}\left[cf_k^{(1)}(j,i)m_k^{(1)}(j)\right]-\sum_{i=1}^{\bar{M}_k^{(1)}}\left[cf^{(1)}(j,i)m^{(1)}(j)\right]\\ &\simeq\bar{M}_k^{(1)}(m_k^{(1)}(j))^2-\bar{M}_k^{(1)}(m^{(1)}(j))^2>0.\end{aligned} \tag{D.4}$$

It follows from equations D.3 and D.4 that

$$\frac{1}{\bar{M}_k^{(1)}}\left[\sum_{i=1}^{\bar{M}_k^{(1)}} cf_k^{(1)}(j,i)\right]^2 - \frac{1}{\bar{M}_k^{(1)}}\left[\sum_{i=1}^{\bar{M}_k^{(1)}} cf^{(1)}(j,i)\right]^2$$
$$\geq \frac{1}{\bar{M}_k^{(1)}}\sum_{i=1}^{\bar{M}_k^{(1)}}(cf_k^{(1)}(j,i))^2 - \frac{1}{\bar{M}_k^{(1)}}\sum_{i=1}^{\bar{M}_k^{(1)}}(cf^{(1)}(j,i))^2. \tag{D.5}$$

From equation D.1, we have for $j = 1, \ldots, L$,

$$\begin{aligned}
&\left|\left[\sigma_k^{(1)}(j)\right]^2 - [\sigma^{(1)}(j)]^2\right| \\
&= \left|\frac{1}{M_k^{(1)}}\sum_{i=1}^{M_k^{(1)}}\left(cf_k^{(1)}(j,i) - m_k^{(1)}(j)\right)^2 - \frac{1}{M^{(1)}}\sum_{i=1}^{M^{(1)}}(cf^{(1)}(j,i) - m^{(1)}(j))^2\right| \\
&= \left|\frac{1}{M_k^{(1)}}\sum_{i=1}^{M_k^{(1)}}\left(cf_k^{(1)}(j,i)\right)^2 - (m_k^{(1)}(j))^2 - \frac{1}{M^{(1)}}\sum_{i=1}^{M^{(1)}}(cf^{(1)}(j,i))^2 + (m^{(1)}(j))^2\right| \\
&\leq \left|\frac{1}{M_k^{(1)}}\sum_{i=1}^{\bar{M}_k^{(1)}}\left(cf_k^{(1)}(j,i)\right)^2 - \frac{1}{M^{(1)}}\sum_{i=1}^{\bar{M}_k^{(1)}}(cf^{(1)}(j,i))^2\right| + \left|\frac{1}{M_k^{(1)}}\sum_{i=\bar{M}_k^{(1)}+1}^{M_k^{(1)}}\left(cf_k^{(1)}(j,i)\right)^2\right. \\
&\quad \left. - \frac{1}{M^{(1)}}\sum_{i=\bar{M}_k^{(1)}+1}^{M^{(1)}}(cf^{(1)}(j,i))^2| + |(m^{(1)}(j))^2 - \left(m_k^{(1)}(j)\right)^2\right| \\
&\simeq \left|\frac{1}{\bar{M}_k^{(1)}}\sum_{i=1}^{\bar{M}_k^{(1)}}\left(cf_k^{(1)}(j,i)\right)^2 - \frac{1}{\bar{M}_k^{(1)}}\sum_{i=1}^{\bar{M}_k^{(1)}}(cf^{(1)}(j,i))^2\right| + \left|\frac{1}{M_k^{(1)}}\sum_{i=\bar{M}_k^{(1)}+1}^{M_k^{(1)}}\left(cf_k^{(1)}(j,i)\right)^2\right. \\
&\quad \left. - \frac{1}{M^{(1)}}\sum_{i=\bar{M}_k^{(1)}+1}^{M^{(1)}}(cf^{(1)}(j,i))^2| + |(m^{(1)}(j))^2 - \left(m_k^{(1)}(j)\right)^2\right|.
\end{aligned} \tag{D.6}$$

In view that $cf_k^{(1)}(j,i) \leq 1$,

$$\begin{aligned}
&\left|\frac{1}{M_k^{(1)}}\sum_{i=\bar{M}_k^{(1)}+1}^{M_k^{(1)}}\left(cf_k^{(1)}(j,i)\right)^2 - \frac{1}{M^{(1)}}\sum_{i=\bar{M}_k^{(1)}+1}^{M^{(1)}}(cf^{(1)}(j,i))^2\right| \\
&\leq \max\left\{\frac{M_k^{(1)} - \bar{M}_k^{(1)}}{M_k^{(1)}}, \frac{M^{(1)} - \bar{M}_k^{(1)}}{M^{(1)}}\right\} \simeq 1 - rate_k.
\end{aligned} \tag{D.7}$$

In view of equations D.6, D.7, and D.5,

$$\begin{aligned}
&\left|\left[\sigma_k^{(1)}(j)\right]^2 - \left[\sigma^{(1)}(j)\right]\right|^2 \le \left| \frac{1}{\bar{M}_k^{(1)}} \sum_{i=1}^{\bar{M}_k^{(1)}} \left(cf_k^{(1)}(j,i)\right)^2 - \frac{1}{\bar{M}_k^{(1)}} \sum_{i=1}^{\bar{M}_k^{(1)}} (cf^{(1)}(j,i))^2 \right| \\
&\quad +1 - rate_k + \left|(m^{(1)}(j))^2 - \left(m_k^{(1)}(j)\right)^2\right| \\
&\le \left| \frac{1}{\bar{M}_k^{(1)}} \left[\sum_{i=1}^{\bar{M}_k^{(1)}} cf_k^{(1)}(j,i)\right]^2 - \frac{1}{\bar{M}_k^{(1)}} \left[\sum_{i=1}^{\bar{M}_k^{(1)}} cf^{(1)}(j,i)\right]^2 \right| + 1 - rate_k \\
&\quad + \left|(m^{(1)}(j))^2 - \left(m_k^{(1)}(j)\right)^2\right| \\
&\simeq \left|\bar{M}_k^{(1)}\left(m_k^{(1)}(j)\right)^2 - \bar{M}_k^{(1)}(m^{(1)}(j))^2\right| + 1 - rate_k + \left|(m^{(1)}(j))^2 - \left(m_k^{(1)}(j)\right)^2\right| \\
&\le (M+1)\left|\left(m_k^{(1)}(j)\right)^2 - (m^{(1)}(j))^2\right| + 1 - rate_k \\
&\le 2(M+1)\left|m_k^{(1)}(j) - m^{(1)}(j)\right| + 1 - rate_k \\
&\le 2(M+1)\left\|\Sigma_k^{(1)} - \Sigma^{(1)}\right\|_2 + 1 - rate_k, \qquad \text{(D.8)}
\end{aligned}$$

where the last inequality is from equation C.3.

Similarly,

$$\left|\left[\sigma_k^{(2)}(j)\right]^2 - \left[\sigma^{(2)}(j)\right]^2\right| < 2(M+1)\left\|\Sigma_k^{(2)} - \Sigma^{(2)}\right\|_2 + 1 - rate_k, \qquad \text{(D.9)}$$

where $j = 1, \ldots, L$.

Thus we have the conclusion in theorem 3.

Acknowledgments

We are grateful for the anonymous reviewers for their insightful comments. We are also grateful to Chin Zheng Yang for his efforts to improve the presentation of this article.

References

Birbaumer, N., Ghanayim, N., Hinterberger, T., Iversen, I., Kotchoubey, B., Kubler, A., Perelmouter, J., Taub, E., & Flor, H. (1999). A spelling device for the paralysed. *Nature, 398*, 297–298.

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

Chen, Y. P. (2000). *Matrix theory*. Xian, China: Northwest Chian University of Technology Publisher.

Donoghue, J. P. (2002). Connecting cortex to machines: Recent advances in brain interfaces. *Nature Neuroscience Supplement, 5*, 1085–1088.

Dornhege, G., Blankertz, B., Curio, G., & Muller, K. R. (2004). Boosting bit rates in non-invasive EEG single-trial classifications by feature combination and multiclass paradigms. *IEEE Trans. Biomed. Eng., 51*(6), 993–1002.

Grandvalet, Y., & Bengio, Y. (2004). Semi-supervised learning by entropy minimization. *Advances in neural information processing systems, 16*. Cambridge, MA: MIT Press.

Kubler, A., Kotchoubey, B., Kaiser, J., Wolpaw, J. R., & Birbaumer, N. (2001). BrainC-computer communication: Unlock the locked-in. *Psychol. Bull., 127*(3), 358C375.

Lemm, S., Blankertz, B., Curio, G., & Muller, K. R. (2005). Spatio-spectral filters for improving the classification of single trial EEG. *IEEE Transactions on Biomedical Engineering, 52*(9), 1541–1548.

McEvoy, L. K., Smith, M. E., & Gevins, A. (2000). Test-retest reliability of task-related EEG. *Clinical Neurophysiology, 1*, 457–463.

Millan, J. R., & Mourino, J. (2003). Asynchronous BCI and local neural classifiers: An overview of the Adaptive Brain Interface project. *IEEE Trans. on Neural Systems and Rehabilitation Engineering, 11*(2), 159–161.

Nigam, K., & Ghani, R. (2000). Analyzing the effectiveness and applicability of co-training. In *Proceedings of 9th International Conference on Information and Knowledge Management* (pp. 86–93). McLean, VA.

Pfurtscheller, G., Neuper, C., Guger, C., Harkam, W., Ramoser, H., Schlogl, A., Obermaier, B., & Pregenzer, M. (2000). Current trends in Graz brain-computer interface research. *IEEE Trans. on Rehabilitation Engineering, 8*(2), 216–218.

Pfurtscheller, G., Neuper, C., Flotzinger, D., & Pregenzer, M. (1997). EEG-based discrimination between imagination of right and left hand movement? *Electroencephalogr. Clin. Neurophysiol., 103*, 642–651.

Polich, J. (2004). Neuropsychology of P3a and P3b- A theoretical overview. In N. C. Moore & K. Arikan (Eds.), *Brainwaves and mind—Recent developments* (pp. 15–29). Wheaton, IL: Kjellberg.

Ramoser, H., Muller-Gerking, J., & Pfurtscheller, G. (2000). Optimal spatial filtering of single trial EEG during imagined hand movement. *IEEE Trans. on Rehabilitation Engineering, 8*(4), 441–446.

Regan, D. (1989). *Human brain electrophysiology: Evoked potentials and evoked magnetic fields in science and medicine*. Dordrecht: Elsevier Science Publishing.

Vidaurre, C., Schlogl, A., Cabeza, R., Scherer, R., & Pfurtscheller, G. (2005). Adaptive on-line classification for EEG-based brain-computer interfaces with AAR parameters and band power estimates. *Biomed. Tech. (Berl.), 50*(11), 350–354.

Wolpaw, J. R., Birbaumer, N., McFarland, D. J., Pfurtscheller, G., & Vaughan, T. M. (2002). Brain-computer interfaces for communication and control. *Clinical Neurophysiology, 113*, 767–791.

Xu, L., & Jordan, M. I. (1996). On convergence properties of the EM algorithm for gaussian mixtures. *Neural Computation, 1*, 129–151.

Zhou, D., Bousquet, O., Lal, T. N., Weston, J., & Schölkopf, B. (2003). Learning with local and global consistency. In S. Becker, S. Thrün, & K. Obermayer (Eds.), *Advances in Neural Information Processing Systems, 15*. Cambridge, MA: MIT Press.

Received September 14, 2005; accepted February 25, 2006.