Neurocomputing 116 (2013) 280–290

Contents lists available at SciVerse ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom

# A novel method for motor imagery EEG adaptive classification based biomimetic pattern recognition

Yan Wu*, Yanbin Ge

*Department of Computer Science and Technology, Tongji University, Shanghai 201804, China*

ARTICLE INFO

Available online 9 November 2012

*Keywords:*
Biomimetic pattern recognition
Hyper sausage neuron
Brain–computer interface
Motor imagery
Adaptive classification

ABSTRACT

The real on-line BCI is indeed a hotspot at present whose performance however is limited by the problems of non-stationary etc. In this paper, a novel method for the adaptive classification of motor imagery EEG data based Biomimetic Pattern Recognition (BPR) through introducing three adaptive operators is proposed. Considering that the large amounts of labeled samples are difficult to get in the actual application, we also propose a novel unsupervised scheme based the adaptive classifier to solve this problem. Sufficient experiments are conducted on the datasets from previous Brain–Computer Interface Competitions and the actual on-line EEG data in adaptive scheme. The results demonstrate that the new algorithm is efficiency and robust compared with non-adaptive classifiers. Besides, a couple of analyses are made on the selection of parameters in the adaptive BPR, and some advice has been come up with about their selections.



## 1. Introduction

Brain–computer interfaces (BCI), started in the early 1960s, commits to create a direct channel of communication between one subject's brain and one computer, without via the traditional muscle-dependent pathway [1]. Electroencephalogram (EEG), an electrical signal collected from scalp, is now widely used in the field of BCI mainly due to its fine temporal resolution, ease of use, portability and low set-up cost [2]. By analyzing the specific frequency components of EEG, a subject's intention can be recognized and translated it into control commands for external electronic device.

Motor imagery (MI), a kind of spontaneous EEG, is now a research hotspot of BCI. It is broadly accepted that mental imagination of movements involves similar brain regions/functions which are involved in programming and preparing such movements [3]. So it would have been a natural thing to employ motor imagery into the EEG-based BCIs.

When a MI–BCI system is actually applied, a critical problem must be carefully considered—the non-stationary of EEG. It is a common phenomenon due to the subject's brain conditions or environmental changes [4,5]. Moreover, recent researches [6,7] indicate that the distributions of EEG features probably change during on-line sessions, comparing with the training data collected offline, due to the non-stationary problem. So for better interpreting the subject's brain signals, the adaptation of both system and the subject are necessary and significant in practical use. A general approach is to design an adaptive machine learning algorithm for the system and a feedback MI scenario for the subject. In such BCI systems, the machine learning algorithm should own the ability to fit its parameters to the specific characteristics of the subject's brain signals; on the other hand, the subject also needs extensive training to make his signals highly distinguished [8].This article focuses on the former, because the subject's intention (especially in the less training situation) is somewhat uncontrolled; a good flexible adaptation of algorithm can not only help the BCI to reach better effect but also relieve a good amount of the training load from the subject.

By now, there have been lots of literatures concerned about adaptive scheme in BCI, from the feature extraction to the classifiers. Zhao et al. in [9] proposed an Incremental Common Spatial Pattern (ICSP) algorithm to solve the poor adaptability of CSP. The new method interpreted CSP feature extraction using the framework of Rayleigh coefficient maximization. An iterative equation of spatial filter was deduced with the present covariance matrices of the two classes and the old spatial filter. The distinct advantage of ICSP was lower computational cost compared to re-training the whole data, so it was more suitable for development of on-line BCI system. Another feature computation based on adaptive autoregressive parameters (AAR) of order 3 from the bipolar channels C3 and C4 was used in [10]. The Graz BCI team [11] developed the Band Power (BP) features combined with phase information, named Complex Band Power (CBP) features. The advantages of CBP over CSP were much fewer electrodes and far less training data required. Besides, it could work better in the presence of artifacts.

On the other hand, there are more literatures today mainly focused on designing an adaptive classifier since a classifier is the

* Corresponding author.
*E-mail address:* yanwu@tongji.edu.cn (Y. Wu).

0925-2312/$ - see front matter © 2012 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.neucom.2012.03.030

core of pattern recognition based BCI, and online training that involves modifying the classification criteria to cope with changes in signal patterns can be a better way to build an adaptive BCI [12]. A variety of adaptive schemes based some common classifiers such as LDA (Linear Discriminant Analysis) [6,13–14], QDA (Quadratic Discriminant Analysis) [10], SVM (Support Vector Machine) [12,15] etc. had been proposed in recent years. The principles of the above algorithms were aimed at adjusting the separating hyperplane along with the changing distribution of features. Vidaurre et al. had done a comprehensive work in [16], that they proposed two continuously adaptive classifiers based quadratic and LDA, while analyzing the adaptive autoregressive parameters and the logarithmic band power in feature type. The work [17] investigated the effects of feedback and motivation on the performance of BCI, and it represented lots of new interesting results in actual on-line BCI. And the literature [18] proposed two new algorithms to handle missing or erroneous labeling in BCI data; the auxiliary label and the optimal proposal functions had been applied successfully to BCI data. Artificial Neural Network (ANN) [19–21] as a powerful tool in pattern recognition also was applied into the BCI. The paper [22] had designed an adaptive probabilistic neural network (APNN) working in a time-varying environment for classification of EEG signals, reached a higher accuracy level of classification over different sessions and subjects.

In [7,23], another kind of methods were proposed, using the Bayesian classifiers and the stochastic gradient method, based on the covariance matrices in the Gaussian distribution. However, theoretically speaking, the reliability of Bayesian Model relies on a good amount of samples. That may lead to performance degradation when facing the problem that the initial EEG data which is used to constitute the initial training set is limited. As a matter of fact, because of the individual differences and the influence of EEG non-stationary (i.e. even for the same subject, it is hard to employ the same classifier without retraining to classify the data collected on two different time), before starting to use an online BCI, there should be a training procedure for the subject that the data acquired are used for off-line training classifier. Unfortunately, the size of the initial data is usually small for time efficiency and subject fatigue, so how to tackle with the adaptive classification in the circumstance of small-sample also ought to be considered.

In this paper, an adaptive classification technique is presented based biomimetic pattern recognition (BPR). Unlike the traditional classifiers on the basis of partition, BPR is a novel classifier based on high-dimension space geometric coverage which was proposed in 2002 [24]. Its own intrinsical advantages in dealing with adaptive learning and limited samples problem are very suitable for the BCI on-line scenario. This adaptive classifier is applied to a variety of datasets collected from 2002 and 2005 BCI Competitions, and the results testify the validity of our idea. In addition, considering the real-time applications, we try to combine the adaptive method with thought of clustering, and propose an unsupervised scheme to apply into the on-line BCI data collected from our own laboratory, preparing for the future work in automatic vehicle control.

The rest of this paper is organized as follows: Section 2 introduces the basic BPR method; Section 3 explains our adaptive algorithm. In Section 4, we analyze the experiments and its results. Some discussions about our algorithm are provided in Section 5. Section 6 has our conclusions.

## 2. The basics about biomimetic pattern recognition

### 2.1. Basic thoughts of BPR

For the traditional pattern recognition, the starting point of its basic mathematical models is that all available information is in the training set. In other words, if there is any relationship between training samples of the same class is unknown in advance. However, BPR indicates that this kind of prior knowledge does exist and it is called the Principle of Homology-Continuity (PHC). For example, if two samples are of the same class, the differences between them must be gradually changed. Meanwhile, all samples in the course of the transition belong to the same class. The mathematical description of PHC is as follows:

In the feature space $\mathbf{R^n}$, suppose that set $A$ is a point set including all samples in class $A$. $x$, $y$ denotes any two samples in $A$, and $\varepsilon$ is an arbitrary positive value, there must exist set $B$ [24]

$$B=\left\{\begin{array}{l} x_1,x_2,x_3,\ldots,x_n \,|\, x_1=x,\ldots,x_n=y,n\in N,\\ \rho(x_m,x_{m+1})<\varepsilon,\varepsilon>0,\\ n-1\geq m\geq 1,m\in N \end{array}\right\}\subset A \tag{1}$$

In other words, set $B$ contains some samples that gradually change (satisfy the constraint $\rho(x_m,x_{m+1})<\varepsilon,\varepsilon>0$) from $x$ to $y$. And all points in set $B$ should be also classified into class $A$ on the basis of PHC.

Traditional pattern recognition aims at getting the optimal partition of different samples classes in the feature space. However, the BPR intends to find the optimal covering of the samples in the same class. And this process can be called "cognition" which is different from "partition" in the traditional pattern recognition. By introducing the PHC, the critical step of BPR is to analyze and "cognitive" the shape of infinite points set made up of all training samples of the same class in the feature space and then construct a high-dimension space geometric coverage as the model of BPR classifier.

Denote $P$ as the region where samples of class $A$ distribute. According to the PHC, other samples near $P$ should be considered as the same class. How to quantify the word "near" is the key to find an appropriate covering in the feature space of class $A$ in BPR. Defined as

$$d(x,P)=\min_{y\in P} d(x,y) \tag{2}$$

which is the distance between the vector $x$ and set $P$. Then the appropriate covering set $P_A$ of class $A$ is

$$P_A=\{x\,|\,d(x,A)\leq k\} \tag{3}$$

where $k$ is the distance threshold. Note that $k$ is a significant factor, and we will discuss it in later section. In order to make it easier to understand the substance of BPR algorithms, the 2-dimension schematic diagram of the difference of back propagation, Radial Basis Function (RBF), and BPR is shown in Fig. 1. More original and detailed information about BPR can be acquired in [24].

### 2.2. Construction of BPR classifier

For using the BPR algorithm into classification, the construction of covering neuron is a key sticking point. Hyper-Sausage Neuron (HSN) [25] is used in this paper to cover the training set. HSN's coverage in the n-dimensional space can be seen as a topological product of a one-dimensional line segment and an n-dimensional hypersphere. HSN is a frequently used neuron model in BPR, which has better generalization capability and wider applicable scope. Shoujue Wang et al. had shown in [26] that the HSN networks can cover a larger region than RBF networks with the same parameters, especially when the distance between samples is large, so that HSN networks can provide much stronger generalization ability. The approximate two-dimensional drawing of HSN coverage is shown in Fig. 2.

For the neuron, parameters including the length of line segment and the radius of hypersphere are needed to consider

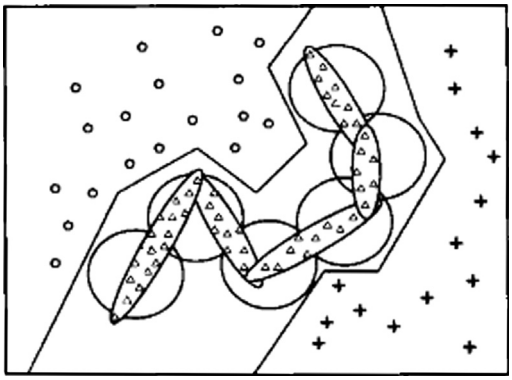

**Fig. 1.** The triangles represent samples to be recognized. The circles and crosses represent samples to be distinguished from triangles. Polygonal line represents the classification manner of traditional back propagation networks. Great circle represents the classification manner of Radial-basis function (RBF) networks. Ellipses represent BPR [25].

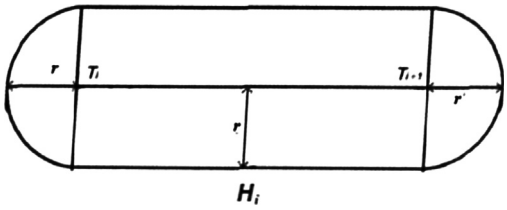


**Fig. 2.** The HSN $H_i$ covers $T_i$ and $T_{i+1}$ in a two-dimension space; the coverage is equivalent to the topological product of line segment $\overline{T_iT_{i+1}}$ and a plane circle whose radius is $r$.

carefully since their values directly affect the accuracy of the classification and generalization of BPR. Note that the radius $r$ in HSN is equivalent to the distance $k$ mentioned in Section 2.1. Besides, the cover process of whole model and the recognition tactic varies widely from case to case. In this paper, we refer to the construction method of BPR which was proposed in another paper [27]. The brief construct procedures are as follows:

- Firstly, for the training set $T$, the Euclid-distance is calculated between any two points of the same class, and then the shortest pair is found, i.e. $T_1$ and $T_2$ for class $A$. Then the first neuron $H_1$ can be constructed by the line segment $(T_1T_2)$ denoted $\theta_1$, topologically multiply a hypersphere of radius $r$. Its coverage is noted as $S_1$.
- Secondly, filter remaining points in $T$ which are covered by $S_1$. Find $T_3$ in the remaining points, which is closest to $H_1$. $\theta_2$ denotes the line segment between $T_3$ and the nearest HSN point (i.e. $T_1$ or $T_2$ in the first neuron $H_1$). $H_2$ then be constructed by $\theta_2$ and a hypersphere. Its coverage is noted as $S_2$.
- Thirdly, repeat the second step and delete the remaining points which are covered by $S_1$, $S_2$, … $S_i$. Find point $T_j$ which is closest to these neurons and build a new HSN $H_{(i+1)}$ to cover it by the topological product of line segment and hypersphere.
- Finally, if all the points in $T$ have been covered, the construct process is terminated. $n$ HSNs are produced, and the coverage of class $A$ is $S_A=S_1\cup S_2\cup\ldots\cup S_n$.

As mentioned in Section 2.1, the basic recognition process is to judge which coverage the test point falls into. Noted that the point has a chance to fall into none or the overlapped coverage [28]; the average distances need to be calculated between the point and every class in this situation. Defined the distance between sample $x$ and class $A$:

$$d(x,A)=\sum_{i=1}^{m} d(x,H_i)/m \tag{4}$$

where $H_i$ is the neuron that belongs to the class $A$, $m$ is the total number of class $A$. Similarly, there are $d(x, B)$, $d(x, C)$…, and $x$ belongs to the class that is closest to it.

It is easy to see the factors specifying the BPR model are the shapes of its neurons. The key parameters of neuron should be the length of line segment $\theta$ and the radius $r$. For this reason, the core problem of the adaptive scheme in BPR is how to enable neurons to be adaptive. In other words, how to adjust the $\theta$ and $r$ is the problem that our mainly concern.

## 3. Adaptive classification

As discussed above, we need a method for the adaptive selection of $\theta$ and $r$. In this paper, a relative value method [27] is adopted instead of traditional fixed value, multiplying the line segment $\theta_i$ of HSN $H_i$ by a distance coefficient $\beta$:

$$r_i=\beta\theta_i \tag{5}$$

Since introducing the relative distance coefficient, we can adjust $r$ by control the value of $\beta$. Meanwhile, in order to enable classifier of self-adaptation, three operators for BPR are introduced: expand, contract and adjunct. Briefly, the first two operators are used to change the coverage area of one neuron; and the last operator is to construct a new neuron and attach to the original BPR model. Since the BPR is a novel method for classification based on high-dimension space geometric coverage, we can explain these operators from the perspective of space geometry.

### 3.1. Adaptive operators

#### 3.1.1. Expand

This operator is to enlarge the coverage area of one neuron by increasing the value of $\beta$. For instance, when a new sample with label is added into the training set, it cannot be covered by any neuron of that class in the original model. The expansion operator is now performed to enlarge the coverage area of neuron which is closest to the new sample. The two-dimensional diagram of "expand" is shown in Fig. 3(a).

#### 3.1.2. Contract

This operator is to reduce the coverage area of one neuron by decreasing the value of $\beta$. For instance, when a new sample with label is added into the training set, it is wrongly covered by one neuron of another class in the original model. The contraction operator is now performed to reduce the coverage area of the wrong neuron so that it can exclude the new sample. The two-dimensional diagram of "contract" is shown in Fig. 3(b).

#### 3.1.3. Adjunct

This operator is to construct a new neuron and attach to the original BPR model. For instance, when a new sample with label is added into the training set, it is not covered by the neurons of its class in the original model. Different from the situation of expanding, the distance between the new sample and the closest neuron is so long that expand operator may cause the overlapping area between classes too large. For this reason the adjunct operator is performed to construct a new neuron by getting together the new sample and the closest point of the original model. The two-dimensional diagram of "adjunct" is shown in Fig. 3(c).

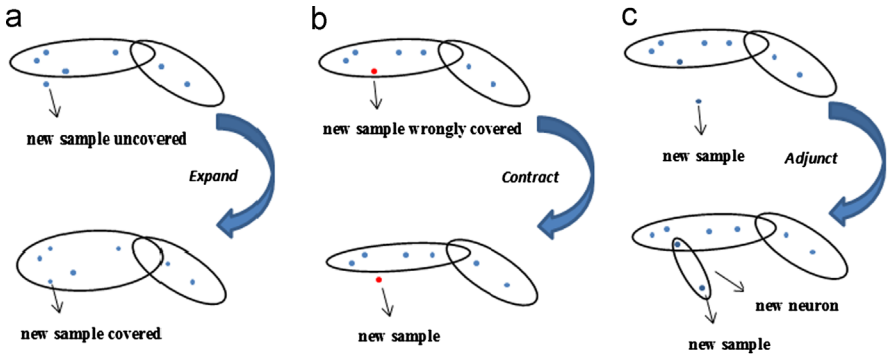


**Fig. 3.** This figure shows the two-dimensional diagram of three adaptive operators. (a) The "expand" operators to enlarge the coverage of HSN. (b) The "contract" operators to reduce the coverage of HSN. (c) The "adjunct" operators to create a new HSN attaching the initial model.

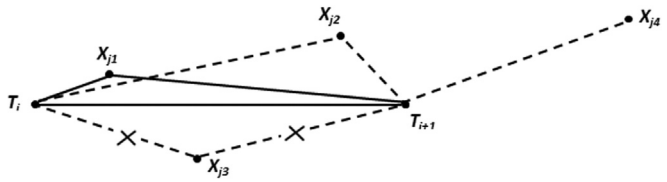


**Fig. 4.** This figure shows the three case of adaptation. Solid-line represents original HSN's 1D topological framework and dash one represents new 1D framework after applying adaptive operators.

### 3.2. Mathematic analysis of each operator

As discussed in Section 2.2, each HSN has a 1D line segment and a hyper-sphere with radius $r$ moving along it. From a different perspective, the samples, which fall into the HSN's coverage, belong to its straight-line segment (i.e. $X_{j1}$ belongs to $\overline{T_iT_{i+1}}$ see Fig. 4).

#### 3.2.1. Expand operator

Assume a new sample $X_{j2}$ is not covered but it belongs to $\overline{T_iT_{i+1}}$; the HSN should enlarge its area by increasing $\beta$:

$$\beta = \sqrt{d^2_{X_{j2}T_i} - \left(\frac{\left(\overrightarrow{T_iX_{j2}}, \overrightarrow{T_iT_{i+1}}\right)}{d_{T_iT_{i+1}}}\right)^2} / d_{T_iT_{i+1}} \tag{6}$$

where $d_{AB}$ denotes the Euclid-distance between point $A$ and $B$ and vector inner product $\left(\vec{X}, \vec{Y}\right) = \sum_{i=1}^{n} x_i y_i$. The numerator of Eq. (6) is actually the distance from point $X_{j2}$ to the line $\overline{T_iT_{i+1}}$ in n-dimension space.

#### 3.2.2. Contract operator

Assume a new sample $X_{j3}$ is wrongly covered, that means it is not belong to the line segment; so the HSN should apply contract operators by decreasing $\beta$:

$$\beta = \sqrt{d^2_{X_{j3}T_i} - \left(\frac{\left(\overrightarrow{T_iX_{j3}}, \overrightarrow{T_iT_{i+1}}\right)}{d_{T_iT_{i+1}}}\right)^2} / d_{T_iT_{i+1}} - \varepsilon \tag{7}$$

where $\varepsilon$ denotes a small positive value, usually equals 0.05.

#### 3.2.3. Adjunct operator

Assume a new sample $X_{j4}$ should be classified to the $\overline{T_iT_{i+1}}$; however the distance between this point and line segment is long, so a new HSN is constructed, that equivalent to a new 1D line segment and $\beta$ have the average value of all HSNs':

$$\beta = \sum_{i=1}^{n} \beta_i / n \tag{8}$$

### 3.3. Choices of operators in supervised scheme

These three operators may not be performed separately in the real on-line adaptation. It is possible that a new sample may cause one class of neuron expanding meanwhile another class of neuron contracting. In the supervised scheme the data with label is used. So first they can be classified by using the original BPR classifier and then the specific process of adaptation is as follows:

#### 3.3.1. Scenario 1

If the fresh data is classified correctly, considering three situations: (1) it falls into the correct neuron coverage; (2) it may fall into none but is closed to the correct class's neuron; (3) it falls into overlapping coverage and is closed to the correct class's neuron. In the first situation, adaptation is unnecessary, because it means our model is fit. For the second case, "expand" is performed to increase $\beta$ to enlarge the neuron's coverage area. Note that the value of $\beta$ is defined in an area with threshold $\delta$, if $\beta > \delta$, the "adjunct" operator should be used to create a new neuron (the selection of $\delta$ will be discussed in Section 5). In the last situation, the aim of adaption is to reduce the overlapping area by performing "contract" in the wrong neuron. $\beta$ is multiplied by a maximum number that can barely exclude the wrong classified data.

#### 3.3.2. Scenario 2

If the fresh data is wrongly classified, also consider three situations: (1) it falls into false neuron coverage; (2) it may fall into none; (3) it falls into overlapping coverage that the false neuron is closer than the correct neuron. In the first situation, the "expand" and "contract" operator should be performed at the same time, that is, "expand" the correct neuron and "contract" the false neuron. If $\beta > \delta$, the "adjunct" operator should be used as well. For the second case, "expand" operator is used and in the last case, the wrong neuron is "contracted".

The framework of applying operators is shown in Fig. 5.

### 3.4. Adaptation in unsupervised scheme

Different from the supervised scenario, unsupervised methods employ data samples without true labels. The information is the fresh data and their predicted labels. Therefore, we introduce another criterion named "density label" to help classify the

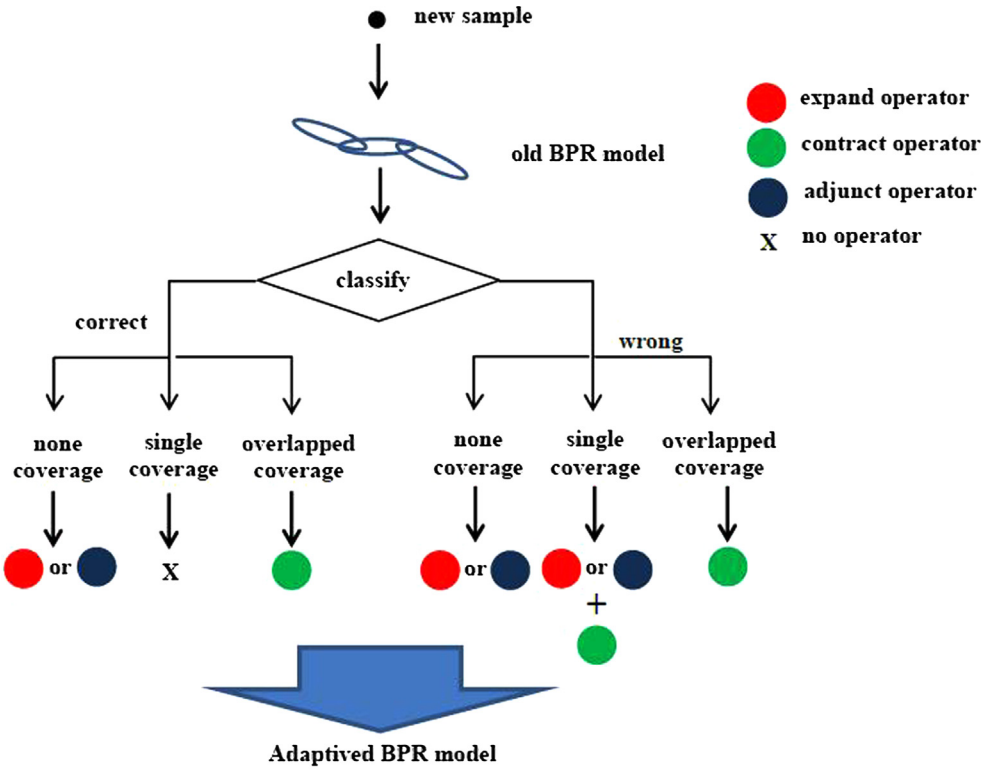


**Fig. 5.** The framework of adaptive algorithms in supervised scheme

unlabeled data. The process to determine the density labels is similar to a clustering algorithm. Suppose $\boldsymbol{A}_1$, $\boldsymbol{A}_2$, … $\boldsymbol{A}_n$ denotes the classes in the sample sets, we first need to calculate each class-density for every new sample $p$.

$$des_{A_i} = \sum_{j=1}^{m} \frac{1}{||x_j - p||}, \quad x_j \in \boldsymbol{A}_i \text{ and } ||x_j - p||^2 \le r^2 \tag{9}$$

where the new data point $p$ is equivalent to a center of a hypersphere and $r$ denotes the radius which equals to average Euclid-distance among all initial training samples. $x_j$ denotes the any initial training sample which belongs to class $\boldsymbol{A}_i$ and locates in the domain of the $p$ hypersphere.

After obtaining each class-density$des_{\boldsymbol{A}_1}, des_{\boldsymbol{A}_2}, \ldots, des_{\boldsymbol{A}_n}$, we take the maximum value as the density label for the new data. And then, the next step is to balance the predicted label and density label so that we can assign a confident label to the new sample and apply our adaptive operators to it. In this paper, we propose another novel method named "cache samples queue" to help tackle this problem. Our specific steps are as follows:

- Step 1. Construct two cache samples queue, confident queue $Q_1$ and uncertain queue $Q_2$, and the max size of $Q_1$ is $N_Q$;
- Step 2. Compute the predicted label $L_p$ and density label $L_d$ for the new sample $a_i$;
  - Step 2.1. If $L_p = L_d$, then we can simply get the confident label which equals to $L_p$ or $L_d$. Then $a_i$ enters $Q_1$;
  - Step 2.2. If $L_p \neq L_d$ and the new sample falls into none or overlap areas of HSN, we believe that the confidence of $L_p$ is lower in this situation, so the confident label equals to $L_d$. Then $a_i$ enters $Q_1$;
  - Step 2.3. When neither has happened, then we temporarily put the new sample into the uncertain queue $Q_2$;
- Step 3. If $Q_1$ is not full, return to step 2;
- Step 4. We take out all samples in $Q_1$ and apply the adaptive operators into them to adjust BPR classifier. Now $Q_1$ is empty;
- Step 5. The samples in $Q_2$ are taken out and compute their new $L_p$ and $L_d$. Put the adequate samples into $Q_1$ referring to the tragedy of step 2, the rest will enter $Q_2$ again. Return to step 2;
- Step 6. When there are no new samples, we take the remaining samples in $Q_1$ to adapt classifier, and then compute $L_p$ and $L_d$ of samples in $Q_2$. If it is not adequate yet, we reject it. Algorithm ends.

There are three points need to declare. The first point is the algorithm above may perform poorly in the worst case that majority of new samples first enter the $Q_2$ and the $Q_1$ is always not full so that the adaptive process cannot proceed. The second is the radius $r$ in the computing of density label. Note that $r$ needs to update after each adjustment of classifier. Actually, we allocate the computation of $r$ into every computation of predicted label. So there is no need to care about the time-cost of computing average Euclid-distance between all samples (initial samples and new samples in $Q_1$). The third point is considering the non-stationarity of EEG, once a sample will enter $Q_2$ the third time, we need reject it because it may not represent the recent EEG signal.

Another important detail to notice is that we may reject some samples in step 6. It does make sense that some unlabeled samples should not be used for learning because they do not have high degree of distinction, or are bad points due to various interferences. Attempt to learn these samples may get classifier degenerated. For avoiding this bad case, rejecting some uncertain samples is very necessary.

## 4. Experiments and results

As mentioned in section1, we have shown the problems of non-stationarity and limited training data are inevitable in the

on-line BCI classification, and we believe that the adaptive BPR is very suitable for on-line EEG process. In this section, we conduct a range of experiments aimed at quantifying the performance of the adaptive BPR.

### 4.1. Preparation

#### 4.1.1. Datasets

We use five datasets to conduct the experiments of this paper mainly come from previous BCI Competitions (BCIC) and our own lab. For the BCIC data, some more representative subsets of the whole public datasets are selected. The description of datasets are as follows:

- Dataset 1 is from dataset III of 2003BCI Competition, which is provided by the Graz University. It contains a total of 280 groups of right and left hand MI EEG signals.
- Dataset 2 includes the '*aa*', '*al*' and '*aw*' three subsets, which is selected from dataset IVa in 2005 BCIC III. It is provided by Berlin BCI group. Each subset contains a total of 280 groups of right hand and foot MI EEG data. There are two types of visual stimulation: (1) where targets are indicated by letters appearing behind a fixation cross (which might nevertheless induce little target-correlated eye movements), and (2) where a randomly moving object indicated targets (inducing target-uncorrelated eye movements). The chosen datasets covers two different situations that subjects '*al*' and '*aw*' 2 sessions of both types are recorded while subject '*aa*' 3 sessions of type (2) and 1 session of type (1) are recorded.
- Dataset 3 chooses the 'S4' subject's EEG data from the dataset IIIb of 2005 BCIC III, which is specifically recorded for describting the actual non-stationary situation in BCI. It is cued motor imagery with online feedback with 2 classes. There are 3 sessions for 'S4' subject and each session consists of 9 runs. The feedback is basket paradigm which also be used in [10], and it is very suitable and representative for on-line scenario.
- Dataset 4 chooses the '*k3b*' subject which is selected from 2005 BCIC III. This is also a typical dataset consists of recordings of four different motor imagery signals: left hand, right hand, foot and tongue. The MI task is to perform those four imagery movements according to a random cue with 90 groups of each.
- Dataset 5 is our actual on-line MI-EEG data of left and right hand movement which is acquired from the Brain Cognitive and Brain–Computer Interface laboratory of Tongji University. For environmental factors and subject factors, we choose two representative datasets '*xk*' and '*gyb*' to test. '*xk*' contains 160 groups data of 8 runs and '*gyb*' contains 140 groups data of 7 runs. In the experiments, the subjects sit on a chair and relax, facing the monitors to accept visual stimuli. The break time of each run is 1–2 minutes, and each run has 20 single experiments. The duration of every single experiment is 10 seconds (see Fig. 6).

#### 4.1.2. Preprocess

For the raw data, we first use Common Average Reference (CAR) as spatial filter. CAR is a high pass space filter, which can enhance the focal activity from the local sources (e.g. the mu and the beta rhythms) and reduce the widely distributed activity, including that resulting from distant sources [29]. Then select different optimal band for different datasets and used FIR (Finite Impulse Response) filter as the band-pass filter to obtain a significant Event Related Synchronization (ERS)/ Event Related Desynchronization (ERD) feature. At last, intercept the data between 3 and 4 second after the stimulus.

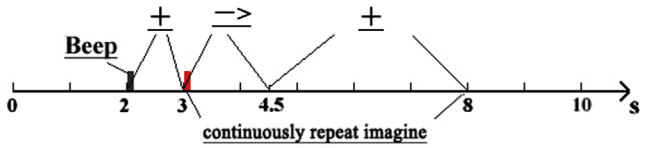


**Fig. 6.** Motor imagery paradigm used for dataset 5. Between 2 s and 3 s, the cross appeared to indicate the position of visual arrow; then left or right arrow appeared during 3 s and 4.5 s. Subject should imagine the left or right hand movement during 3.5 s and 8 s continuously.

#### 4.1.3. Feature extraction

Common Spatial Pattern (CSP) is the feature extraction method we used. The CSP method is based on a decomposition of the raw EEG signals into spatial patterns, which are extracted from two populations of single trial EEG. These patterns maximize the difference between the populations [30]. Given two classes of motor imagery task C1 and C2, two covariance matrices $\Sigma_1$ $\Sigma_2$ are calculated for the two classes.

$$\sum_1 = \sum_{j \in C_1} \frac{T_j^{T*} T_j}{trace\left(T_j^{T*} T_j\right)}\ ,\ \Sigma_2 = \sum_{j \in C_2} \frac{T_j^{T*} T_j}{trace\left(T_j^{T*} T_j\right)} \tag{10}$$

where $T_j \in \mathbf{R}^{s \times c}$ denotes an EEG data matrix of the $j$th trial, $s$ is the number of selected channels, $c$ is the number of samples in each trial.

Then the CSP algorithm calculates a matrix $\boldsymbol{w}$ and diagonal matrix $\boldsymbol{D}$ with elements in [0, 1] with:

$$\boldsymbol{w}\Sigma_1 \boldsymbol{w}^T = \boldsymbol{D}\ ,\ \boldsymbol{w}\Sigma_2 \boldsymbol{w}^T = \boldsymbol{I} - \boldsymbol{D} \tag{11}$$

where $\boldsymbol{I}$ is an identity matrix. Then the matrix $\boldsymbol{W}$ called CSP transformation matrix can be constructed, composed of the first and last $m$ rows of $\boldsymbol{w}$. Next we extract the $i$-th EEG trial's feature $F(i)$ as follows; each feature has $2m$ dimensions:

$$F(i) = diag\left(\boldsymbol{W} \frac{T_j^{T*} T_j}{trace\left(T_j^{T*} T_j\right)} \boldsymbol{W}^T\right),\ i = 1,2,\ldots(n_t + n_e) \tag{12}$$

where $(n_t + n_e)$ denotes the sum of the number of training and testing samples.

However, the original CSP focuses on classification of two classes. When there are three or more kinds of data need to classify, an extension should be made to overcome its natural weakness. The common method is CSP-OVR paradigm [31] which computes spatial filters for each class against all others. Therefore in our four class problem (i.e. Dataset 4), we develop 4 projection matrices from the training data. And then we capture the first and last rows of each matrix, which leads to an 8- dimensional feature.

#### 4.1.4. Classifier

Considering that BPR is a relatively new algorithm, for the purpose of comparison, two traditional classical classifiers SVM and LDA are used in the experiments.

SVM. We use LIBSVM in this paper, in which RBF kernel function as shown in (13) and 5-fold cross validation are used to determine the $\sigma$ in kernel function and the penalty factor in SVM's discriminant function.

$$K(x,y) = \exp\left(-\frac{\|x-y\|^2}{\sigma^2}\right) \tag{13}$$

LDA. For an $m$-class problem, the between- and within-class scatter matrices $S_b$ and $S_w$ are defined as

$$S_b = \sum_{i=1}^{m} \Pr(C_i)(\mu_i - \mu)(\mu_i - \mu)^T$$
$$S_w = \sum_{i=1}^{M} \Pr(C_i) E[(x - \mu_i)(x - \mu_i)^T] \quad (14)$$

where $Pr(C_i)$ is the prior probability of class $C_i$; $\mu$ is the overall mean vector and $\mu_i$ is the mean vector of class $C_i$. The optimal transformation matrix $\boldsymbol{W}$ can be obtained by solving the generalized eigenvalue problem:

$$S_b \boldsymbol{W} = \lambda S_w \boldsymbol{W} \quad (15)$$

For classification, the linear discriminant functions are ($X$ denotes the arbitrary feature vector):

$$D_i(X) = \boldsymbol{W}(X - \mu_i),\ i = 1,2,\ldots,m \quad (16)$$

**Table 1**
Classification accuracy of SVM, LDA and BPR.

| Amount of training samples | Correct rate (%) | | |
|---|---|---|---|
| | SVM | LDA | BPR |
| 80 | 75.0 | 74.5 | 81.0 |
| 140 | 80.7 | 79.3 | 85.7 |
| 200 | 91.3 | 87.5 | 90.0 |

### 4.2. Comparison of SVM, LDA and BPR (in dataset 1)

A comparison between the traditional classifiers (i.e. SVM and LDA) and the BPR classifier in classification of the BCI data is presented at first. This experiment is aimed to study the offline performance of various classifiers. We have carried out three groups of experiments (independently repeat 10 times in each group) with different sizes of training dataset in Dataset 1. The classifiers are all in their optimal states. The penalty factor and $\sigma$ of SVM are determined by 5-fold cross validation and the distance coefficient $\beta$ of BPR is set to be 0.36. Considering that the effects of different bands in classification result. We choose the optimal bands with 12 and 16 Hz. The average experimental results illustrated in Table 1 indicate that the performance of the BPR is much better than the other classifiers when the training samples are not sufficient. The gap is narrowed when training set grows bigger. As a whole, BPR's performance is approved.

### 4.3. Supervised adaptive classification of two classes (in dataset 2 and 3)

At this experiment, we aim to address two following issues: (*a*) whether the BPR adaptive scheme can demonstrably improve performance over the non-adaptive baseline algorithm; (*b*) the specific impact of using the adaptive BPR compared with the original BPR. The datasets from 2005 BCIC III was used in this experiment.

Fig. 7 compares the classification error rate of each non-adaptive method with the adaptive BPR classifier in Dataset 2.

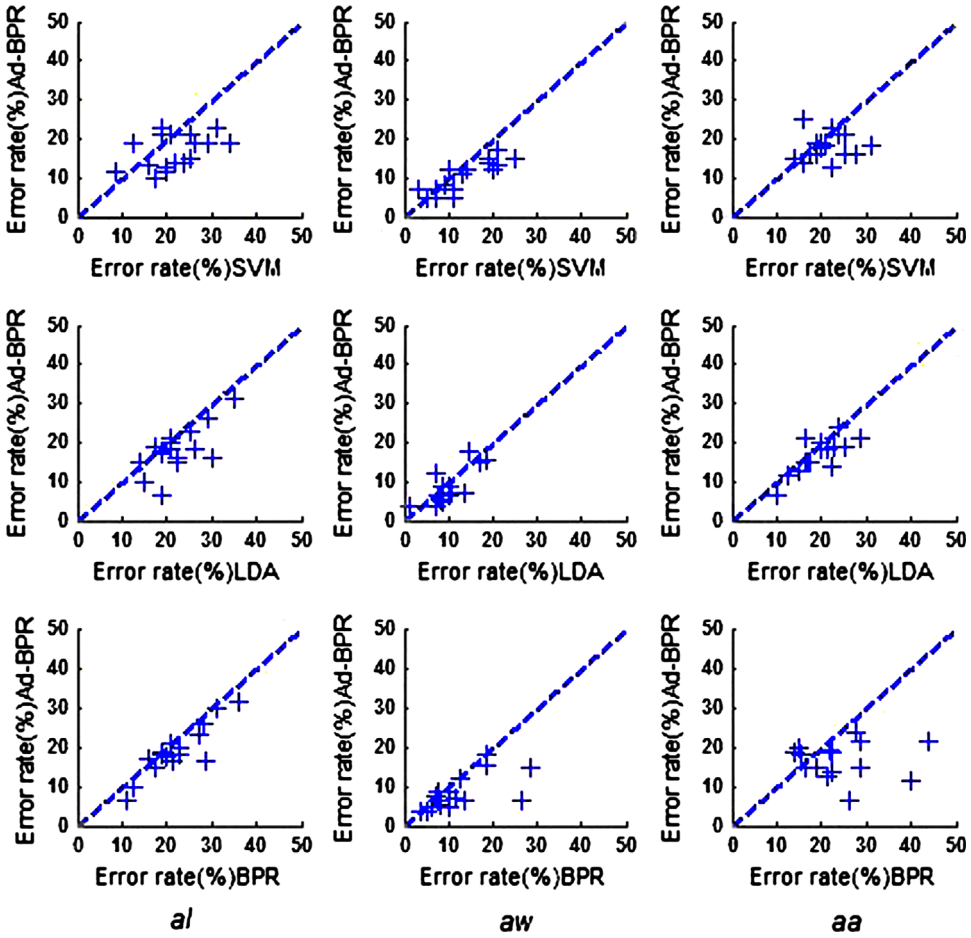


**Fig. 7.** Comparison of non-adaptive classifiers with the adaptive classifier

Each subplot is a scatter plot, with the error rate of the adaptive method on the *y*-axis and the error rate of the method of non-adaptive classifiers on the *x*-axis. Each column presents the three different datasets ('*aa*', '*al*', '*aw*') we used.

All classifiers were first trained on the training set with the random number of 60–100 in each dataset. And then a new hundred fresh data continuously entered with time for adaptive training. Noted that it was just only for the adaptive BPR, while non-adaptive classifiers without this process. After that, the remaining samples in the dataset were used for testing to evaluate the efficiency of classifiers.

Inspecting each row, we see that the adaptive BPR generally outperforms the other three classifiers. However, not all the points are below the dividing lines, that means the adaptive scheme may lead to degradation in some situations. The probably reason is that the EEG data's non-stationary that new fresh data may mislead the classifier and make BPR adapt in a wrong orientation.

For studying the second issue, we adopt an experimental scheme which was named "incremental adaption", proposed by [12]. It can clearly illustrate the performance between the original BPR and the adaptive BPR. The scheme process is as follows:

For each dataset, we divide it into *m* consecutive subsets, the first subset is used for offline training and the rest are used first for testing and then for online training. In non-adaptive situation, the accuracy of classification over each subset is calculated using the BPR trained by the first subset; while in adaptive situation, the accuracy of classification over each subset is calculated using the adaptive BPR (i.e., incrementally trained) by the pervious subset. The classification accuracy over each subset is computed by the rate of properly classified samples to all the samples in each subset.

Fig. 8 depicts the classification performance of the adaptive BPR as well as the non-adaptive classifiers over the three datasets ('*aa*', '*al*', '*aw*'). Each dataset is divided into 12 subsets; the first subset contains 60 samples, and the following subsets have the same number of 20 samples. As it is shown, the adaptive algorithm improves the classification accuracy by about 10% average. Also, on examining all three plots carefully, we see that the former is more stable and less affected by non-stationary of EEG data. From all this, the adaptive method with BPR can indeed improve performance.

We have also conducted the same experiment in Dataset 3, which can be divided into 3 sessions according to trigger time (the starting time of each trial). The time interval between each session is 20 s. Session1 (360 samples) is our training set and session 2 and 3 (720 samples) are our testing sets. The experimental result also shows that the adaptive BPR performs nearly 10% better than the original BPR, whose accuracy rate reaches 87.3% average while the original BPR just 78.7%.

### 4.4. Supervised adaptive classification of multi-class (in dataset 4)

Considering the actual situation in the real BCI that MI does not restrict in 2 classes, we use Dataset 4, a four class problem, to testify our algorithm. The rough EEG recording is made with a 64-channel EEG amplifier from Neuroscan, sampled at 250 Hz and is filtered between 1 and 50 Hz. Four imagery movements (left and right hand, foot, tongue) are included. Each trial lasts about 8 s with 3–7 s is the imagine time. The channel selection is important in this case since 60 channels are too many that may lead to misleading information and high time cost of computation. As consensus, C3, C4 and Cz are the core channels in motor imagery BCI. So we select totally 27 channels (17–25, 27–35, 37–45) around these three electrodes, specific location as can be seen in Fig. 9(a). We select 8–12 Hz as our filter bank for CSP-OVR and time window 6–7 s as the best interval for recognition of EEG

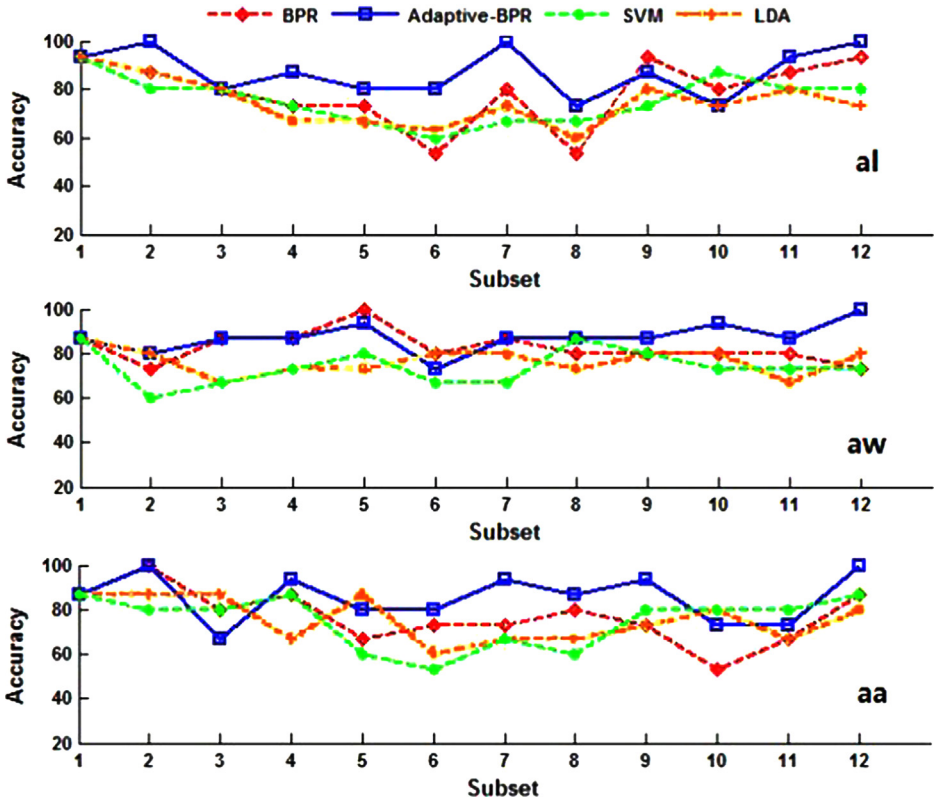


**Fig. 8.** Comparison of BPR with the adaptive BPR

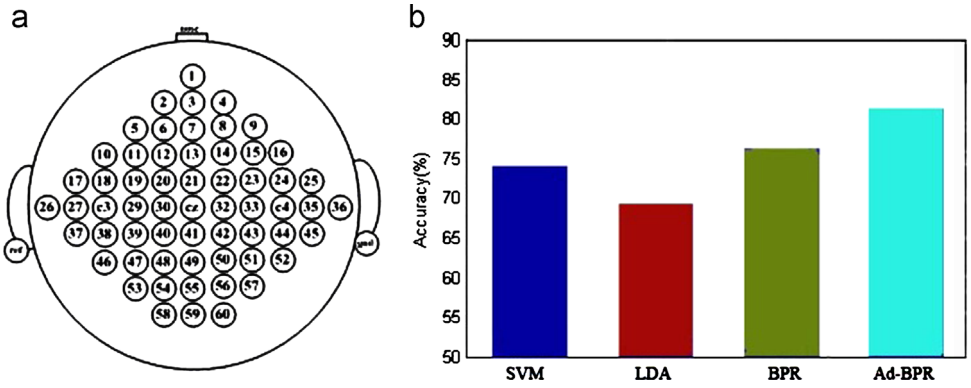


**Fig. 9.** (a) The electrodes locations of dataset 4, No. 1–No. 60 are the original recording channels and No. 17–No. 25, No. 27–No. 45, No. 37–No. 45 are the actual channels we used. (b) Mean accuracy of the non-adaptive classifiers and the adaptive BPR (Ad-BPR) in four class scheme.

data. The training set is 180, 45 of each class. Also, we compare the average classification accuracy of LDA, SVM, BPR and the adaptive-BPR in their optimal state (see Fig. 9(b)). Obviously, in the four class problem, the common decrease of accuracy is in all classifiers. LDA's result is the lowest, about 68.9%; BPR is the highest among the non-adaptive classifiers, 76.67% average. And the adaptive BPR reaches 81.1%. However in this experiment, the adaptive algorithm is barely above the 80%, not very high. The First reason is that in the multi-class situation, the overlapped area may more easily appear in our BPR model. In this case, our algorithm needs to call Eq. (4) to classify the data and it is just a simple criterion whose classification accuracy is not as well as those fall in the coverage of BPR. Another reason is that we use a non-adaptive feature extraction method while an adaptive method is actually necessary in real adaptive BCI. Our future work will focus on improving the rule of overlapped case and the adaptation of feature extraction.

### 4.5. Unsupervised experiment (in dataset 5)

The dataset in this experiment is the actual on-line EEG data referring to Section 4.1. By analyzing the EEG band of subjects, the phenomenon of ERS/ERD of '*xk*' is clear between 8–12 Hz while '*gyb*' is between 12–24 Hz. We choose the first 4 runs of each dataset for the training set to generate initial BPR classifier. And then we put the remaining samples (unlabeled) into the classifier to proceed the unsupervised adaptive learning. At last, after all samples are learned, we test our new classifier's performance by using all samples in the datasets except the ones that for training. The $N_Q$ set to be 8 and 14 for '*xk*' and '*gyb*'. For comparison, the correct rates of non-adaptive classifiers (SVM, LDA and BPR) also are tested. Since the unsupervised scheme is actually based on the adaptive BPR, we call this algorithm U-ABPR for short below. The experiment results are all in their optimal parameters and repeat 10 times for rejecting the randomness. From Fig. 10, the performance of the non-adaptive classifiers is obviously lower than U-ABPR because these classifiers do not use the unlabeled data and have the adaptive learning step. The accuracy of the non-adaptive classifiers is mostly lower than 80%, however the U-ABPR reaches 84.4% and 86.7% in '*xk*' and '*gyb*', 5% higher average. That amply justifies the validity of our method.

As shown in Section 3.4, the parameter $N_Q$ denotes the max size of the confident queue $Q_1$ which influences the performance of our unsupervised scheme. And different sizes influence the experiment results obviously. For this, we set different values in our following experiment to investigate the impact of $N_Q$ to accuracy. The results can be seen in Tables 2 and 3.

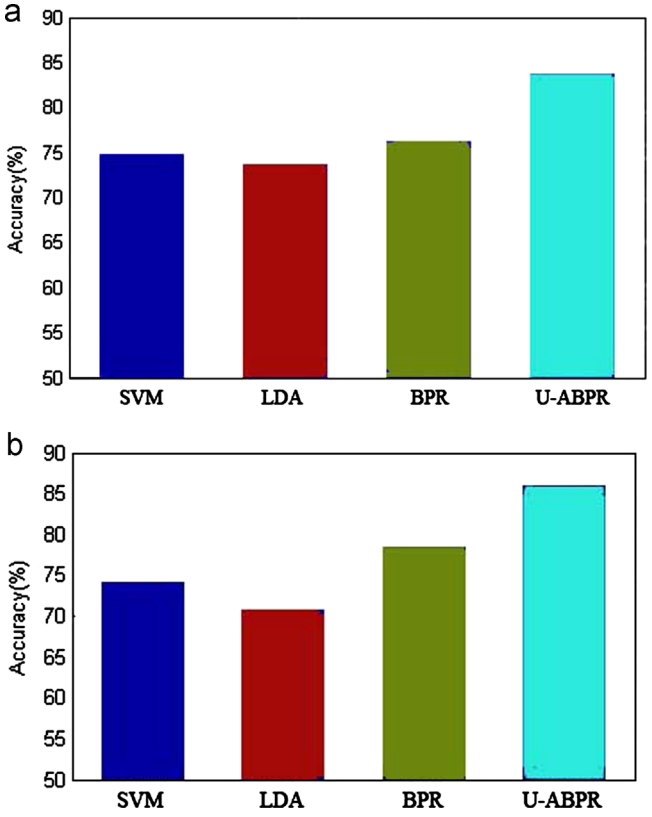


**Fig. 10.** Mean accuracy of the traditional classifier (SVM and LDA), the non-adaptive BPR (BPR) and the adaptive BPR in unsupervised scheme (U-ABPR). Those are with the optimal states and noncausual. (a) The experiment results of subject '*xk*'. (b) The experiment results of subject '*gyb*'.

**Table 2**
Experiment result of '*xk*' in unsupervised scheme.

| Size of the queue $Q_1$ | Average correct rate (%) |
|---|---|
| 1 | 75.3 |
| 5 | 82.1 |
| **8** | **84.4** |
| 10 | 83.8 |
| 15 | 83.8 |
| 20 | 81.6 |
| 25 | 76.8 |
| 30 | 77.3 |

**Table 3**
Experiment result of '*gyb*' in unsupervised scheme

| Size of the queue $Q_1$ | Average correct rate (%) |
|---|---|
| 1 | 79.8 |
| 5 | 81.2 |
| 10 | 83.6 |
| **14** | **86.7** |
| 15 | 86.4 |
| 20 | 86.1 |
| 25 | 81.3 |
| 30 | 79.3 |

Note that, we represent the eight different values of $N_Q$ in this experiment, including the best ones (8 and 14). From these results, we can get a general conclusion that different $N_Q$ indeed leads to different results. Also we can find that too large ( $>20$) or too small ( $<5$) value of $N_Q$ makes the performance of classifier degraded, and the result is not as well as the $N_Q$ between 5 and 20. Different lengths lead to different results, some relations must be behind this interesting feature. And we will discuss it in Section 5.2.

## 5. Discussions

### 5.1. Influence of parameters in the adaptive BPR

In this part, we mainly examine how much influence that the selection of parameters will have on the performance of BPR adaptive classifier. For our algorithm, the distance of coefficient $\beta$ is no doubt a key factor. Its threshold $\delta$ determines whether the 'expand' operator or the 'adjunct' operator is applied. It is clear that different operator must lead to different classifier model. Given $Ac^{non\text{-}adaptive}$ and $Ac^{adaptive}$ denote the accuracy rate of BPR and adaptive-BPR. The improved accuracy of using adaptive scheme compared with non-adaptive scheme is quantified by:

$$\psi = \frac{1}{n}\sum_{i=1}^{n}\left(Ac_i^{adaptive} - Ac_i^{non-adaptive}\right) \tag{17}$$

where $n$ denotes the number of tests. Its positive or negative value represents improvement or degradation of classification hit rate of the applied adaptive BPR, respectively. We employed first three datasets in our paper and observed its average performance.

Fig. 11 shows the result of this experiment, the $x$-axis is the ratio of $\delta$ to $\beta$'s initial value. When ratio is less than 1, it means no "expand" operator is applied; when ratio is much larger, then "adjunct" operator is hardly used. As it is shown, the performance of the adaptive BPR is best when the ratio is between 1.5 and 2. Thus, our recommendation of $\delta$ should be$1.5\beta \le \delta \le 2\beta$. Moreover, the figure also indicates that if the threshold is set to be oversize and ratio above 5, degradation of classifier will be quite obvious.

### 5.2. Different length of confident queue in unsupervised scheme

As shown in Section 4.5, the accuracy in our unsupervised scheme has some relation with the length of confident queue $Q_1$. From the theoretical consideration, the size actually determines when and how often the classifier adjusts. According our unsupervised scheme, its essential purpose is to keep the useful samples and reject bad samples. So if the selection of $N_Q$ is not proper so that useful samples rejected falsely or bad samples are not rejected. For example, if the value is too large, the worst case may easily appear that the classifier cannot update since $Q_1$ is not always full, making the adaptive-BPR degenerate into non-adaptive BPR. If the value is too small, for example, the unsupervised process is equivalent to adapt samples one by one. And another worse case may occur that some useful points are rejected since they are judged by uncertain samples three times before the classifier becomes good enough to receive them. Generally the $N_Q$ between 5 and 20 may be a proper range of the queue. However, this range is a little broad and for the specific dataset, such as '*xk*' and '*gyb*', the proper ranges are actually between [5,15] and [10,20]. The selection of length seems to be with some degree of randomness. From plenty of experiments, we recommended that the length around 15 is proper value for less time cost in adaptation while accuracy is guaranteed at the same time. But it is a still common experience suggestion, not has a really strong grounding in mathematical reasoning. The more rigorous mathematical analysis will be our future work.

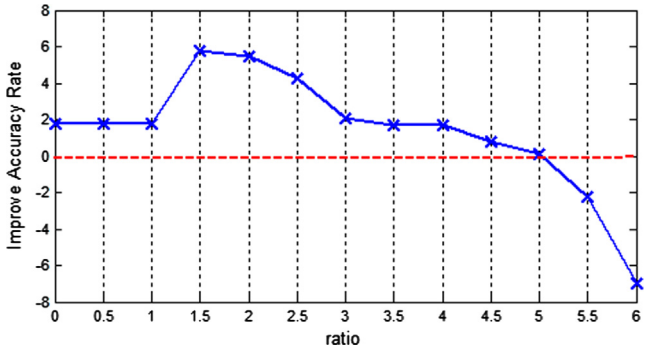


**Fig. 11.** Influence of parameter in adaptive BPR.

## 6. Conclusion

An adaptive BCI classifier has been proposed based BPR in this paper. We apply three adaptive operators into BPR and make it self-adaptive in the supervised situation. Also considering getting the labeled data is often either too expensive or entirely impossible in real-time applications, we attempt to implement adaptive BPR into unsupervised scheme. In Section 4, sufficient experiments are conducted to compare the performance of different classifiers in non-adaptive scheme with adaptive scheme. The results show the effectiveness and robustness of the algorithm. Besides, we have made studies on the selection of parameters in adaptive BPR, and come up with the reference range about its value.

Nevertheless, this adaptive classifier is not very mature; the selections of some parameters mostly rely on the summary of many experiment results, lacking of some specific theoretic support. Moreover, try to make the feature extraction method adaptable is very necessary and will be another main goal of our future work.

## References

[1] J.R. Wolpaw, N. Birbaumer, W.J. Heetderks, et al., Brain–computer interface technology: a review of the first international meeting, IEEE Trans. Rehab. Eng 8 (2) (2000) 164–173.
[2] Brain–computer interface. Available from: 〈http://en.wikipedia.org/wiki/Brain-computer_interface〉.
[3] M.J. Jeannerod, Mental imagery in the motor context, Neuropsychologia 33 (11) (1995) 1419–1432.
[4] T.M. Vaughan, W.J. Heetderks, L.J. Trejo, W.Z. Rymer, et al., Brain–computer interface technology: a review of the second international meeting, IEEE Trans. Rehab. Eng 11 (2003) 94–109.
[5] B. Blankertz, K.R. Müller, D.J. Krusienski, G. Schalk, et al., The BCI competition III: validating alternative approaches to actual BCI problems, IEEE Trans. Rehab. Eng 14 (2) (2006) 153–159.
[6] C. Vidaurre, M. Kawanabe1, P. von Bünau, B. Blankertz, K.R. Müller, Toward an unsupervised adaptation of LDA for brain–computer interfaces, IEEE Trans. Biomed. Eng. 58 (3) (2011) 587–597.

[7] P. Shenoy, M. Krauledat, B. Blankertz, Rajesh P.N. Rao, K.R. Müller, Towards adaptive classification for BCI, J. Neural. Eng. 3 (2006) 13–23.
[8] J. del R. Millan, On the need for on-line learning in brain computer interfaces, in: Proceedings of the IJCNN, Budapest, Hungary, 2004, pp. 2877–2882.
[9] Q.B. Zhao, L.Q. Zhang, C. Andrzej J. Li, Incremental common spatial pattern algorithm for BCI, in: Proceedings of the International Joint Conference on Neural Networks, 2008, pp. 2656–2659.
[10] C. Vidaurre, A. Schlöogl, R. Cabeza, R. Scherer, G. Pfurtscheller, A full on-line adaptive BCI, IEEE Trans. Biomed. Eng. 53 (6) (2006) 1214–1219.
[11] G. Pfurtscheller, G.R. Müller-Putz, A. Schlogl, B. Graimann, et al., 15 years of BCI research at Graz University of Technology: current projects, IEEE. Trans. Neural Syst. Rehab. Eng. 14 (2) (2006) 205–210.
[12] M.A. Oskoei, J.Q. Gan, Huosheng Hu, Adaptive schemes applied to online SVM for BCI data classification, in: Proceedings of the 31st Annual International Conference of the IEEE EMBS, 2009, pp. 2600–2603.
[13] C.S.L. Tsui, J.Q. Gan, Comparison of three methods for adapting LDA classifiers with BCI applications, in: Proceedings of the 4th International Workshop on Brain–Computer Interfaces, Graz, Austria, 2008, pp. 116–121.
[14] C.S.L. Tsui , J.Q. Gan, Asynchronous BCI control of a robot simulator with supervised online training, in: Proceedings of the International Conference on Intelligent Data Engineering and Automated Learning, IDEAL, 2007, Birmingham, UK, pp. 125–134.
[15] G.G. Molina, BCI adaptation using incremental-SVM learning, in: Proceedings of the 3rd International IEEE/EMBS Conference on Neural Engineering, 2007, pp. 337–341.
[16] C. Vidaurre, A. Schlogl, R. Cabeza, et al., Study of on-line adaptive discriminant analysis for EEG-based brain computer interfaces, IEEE. Trans. Biomed. Eng. 54 (3) (2007) 550–556.
[17] B. Grychtol, H. Lakany, G. Valsan, et al., Human behavior integration improves classification rates in real-time BCI, IEEE Trans. Neural. Syst. Rehabil. Eng. 18 (4) (2010) 362–368.
[18] J.W. Yoon, S.J. Roberts, M. Dyson, et al., Adaptive classification for brain computer interface systems using sequential Monte Carlo sampling, Neural. Net. 22 (2009) 1286–1294.
[19] D.S. Huang, Systematic Theory of Neural Networks for Pattern Recognition, Publishing House of Electronic Industry of China, Beijing, 1996.
[20] D.S. Huang, Radial basis probabilistic neural networks: model and application, Int. J. Pattern Recognition Artif. Intell. 13 (7) (1999) 1083–1101.
[21] D.S. Huang, J.X. Du, A constructive hybrid structure optimization methodology for radial basis probabilistic neural networks, IEEE. Trans. Neural Networks 19 (12) (2008) 2099–2115.
[22] M.K. Hazrati, A. Erfanian, An online EEG-based brain–computer interface for controlling hand grasp using an adaptive probabilistic neural network, Med. Eng. Phys. 32 (2010) 730–739.
[23] S.L. Sun, Y. Lu, Y.G. Chen, The stochastic approximation method for adaptive Bayesian classifiers: towards online brain–computer interfaces, Neuro. Comput. Appl. 20 (2011) 31–40.
[24] S.J. Wang, Bionic (topological) pattern recognition—a new model of pattern recognition theory and its applications (in Chinese), Acta Electron. Sin. 30 (2002) 1417–1420.
[25] S.J. Wang, J.L. Lai, Geometrical learning, descriptive geometry, and biomimetic pattern recognition, Neurocomputing 67 (2005) 9–28.
[26] S.J. Wang, X.T. Zhao, Biomimetic pattern recognition theory and its applications, Chinese J. Electronics 13 (3) (2004) 373–377.
[27] K. Xu, Y. Wu, Motor imagery EEG recognition based on biomimetic pattern recognition, in: Proceedings of the 3rd International Conference on Biomedical Engineering and Informatics, BMEI, pp. 955–959, 2010.
[28] Y. Wu, X. Yao, S.J. Wang, Relative division of overlapping space based biomimetic pattern recognition, Pattern Recognition Artif. Intell. 21 (3) (2008) 346–350. (in Chinese).
[29] A. Bashashati, M. Fatourechi, R.K. Ward, G.E. Birch, A survey of signal processing algorithms in brain–computer interfaces based on electrical brain signals, J. Neural. Eng. 4 (2007) R32–57.
[30] M.A. Li, J.Y. Liu, D.M. Hao, J.F. Yang, An improved CSP algorithm and application in motor imagery recognition, in: Proceedings of the 5th International Conference on Natural Computation, Tianjin, China, 2009, pp.113–117.
[31] G. Dornhege, B. Blankertz, G. Curio, K.R. Müller, Boosting bit rates in noninvasive EEG single-trial classifications by feature combination and multiclass paradigms, IEEE. Trans. Biomed. Eng. 51 (6) (2004) 993–1002.

**Yan Wu** received her Ph.D. degree in traffic information engineering and control from Shanghai Tiedao University, China, in 1999. From 2000 to 2003, she had worked as a Postdoctoral Research Fellow in Department of Electric Engineering, Fudan University, China. Now, She is a full professor and Doctoral Advisor in the Department of Computer Science and Technology, Tongji University, Shanghai, China. She has published more than 90 papers on important national and international journals and conference proceedings. Now she is mainly engaged in artificial neural networks, intelligent systems, pattern recognition.

**Yanbin Ge** received the B.S. degree in computer science and now stays on for M.S. degree in pattern recognition and intelligence system from the Tongji University, Shanghai, China. His research interests are focused on pattern recognition and neural network.