TOPICAL REVIEW

A review of classification algorithms for EEG-based brain–computer interfaces

F Lotte[1], M Congedo[2], A Lécuyer[1], F Lamarche[1] and B Arnaldi[1]

[1] IRISA/INRIA Rennes, Campus universitaire de Beaulieu, Avenue du Général Leclerc,
35042 RENNES Cedex, France
[2] France Telecom R&D, Tech/ONE Laboratory, 28 Chemin du vieux Chêne, InoVallée,
38240 Grenoble, France

E-mail: fabien.lotte@irisa.fr

Abstract

In this paper we review classification algorithms used to design brain–computer interface
(BCI) systems based on electroencephalography (EEG). We briefly present the commonly
employed algorithms and describe their critical properties. Based on the literature, we
compare them in terms of performance and provide guidelines to choose the suitable
classification algorithm(s) for a specific BCI.

1. Introduction

A brain–computer interface (BCI) is a communication system that does not require any peripheral muscular activity [1]. Indeed, BCI systems enable a subject to send commands to an electronic device only by means of brain activity [2]. Such interfaces can be considered as being the only way of communication for people affected by a number of motor disabilities [3].

In order to control a BCI, the user must produce different brain activity patterns that will be identified by the system and translated into commands. In most existing BCI, this identification relies on a classification algorithm [4], i.e., an algorithm that aims at automatically estimating the class of data as represented by a feature vector [5]. Due to the rapidly growing interest in EEG-based BCI, a considerable number of published results are related to the investigation and evaluation of classification algorithms. To date, very interesting reviews of BCI have been published [1, 6] but none has been specifically dedicated to the review of classification algorithms used for BCI, their properties and their evaluation. This review aims at filling this lack. Therefore, one of the main objectives of this review is to survey the different classification algorithms used in EEG-based BCI research and to identify their critical properties. Another objective is to

provide guidelines in order to help the reader with choosing the most appropriate classification algorithm for a given BCI experiment. This amounts to comparing the algorithms and assessing their performances according to the context.

This review is organized as follows: section 2 depicts a BCI as a pattern recognition system and emphasizes the role of classification. Section 3 surveys the classification algorithms used for BCI and, finally, section 4 assesses them and identifies their usability depending on the context.

2. Brain–computer interfaces seen as pattern recognition systems

The very aim of BCI is to translate brain activity into a command for a computer. To achieve this goal, either regression [7] or classification [8] algorithms can be used. Using classification algorithms is the most popular approach. These algorithms are used to identify 'patterns' of brain activity [4]. In this review, we consider a BCI system as a pattern recognition system [5, 9] and focus on the classification algorithms used to design them. The performance of a pattern recognition system depends on both the features and the classification algorithm employed. These two components are highlighted in this section.

2.1. Feature extraction for BCI

In order to select the most appropriate classifier for a given BCI system, it is essential to clearly understand what features are used, what their properties are and how they are used. This section aims at describing the common BCI features and more particularly their properties as well as the way to use them in order to consider time variations of EEG.

2.1.1. Feature properties.
A great variety of features have been attempted to design BCI such as amplitude values of EEG signals [10], band powers (BP) [11], power spectral density (PSD) values [12, 13], autoregressive (AR) and adaptive autoregressive (AAR) parameters [8, 14], time-frequency features [15] and inverse model-based features [16–18]. Concerning the design of a BCI system, some critical properties of these features must be considered:

- *noise and outliers:* BCI features are noisy or contain outliers because EEG signals have a poor signal-to-noise ratio;
- *high dimensionality:* in BCI systems, feature vectors are often of high dimensionality, e.g., [19]. Indeed, several features are generally extracted from several channels and from several time segments before being concatenated into a single feature vector (see the next section);
- *time information:* BCI features should contain time information as brain activity patterns are generally related to specific time variations of EEG (see the next section);
- *non-stationarity:* BCI features are non-stationary since EEG signals may rapidly vary over time and more especially over sessions;
- *small training sets:* the training sets are relatively small, since the training process is time consuming and demanding for the subjects.

These properties are verified for most features currently used in BCI research. However, it should be noted that it may no longer be true for BCI used in clinical practice. For instance, the training sets obtained for a given patient would no longer be small as a huge quantity of data would have been acquired during sessions performed over days and months. As the use of BCI in clinical practice is still very limited [3], this review deals with classification methods used in BCI research. However, the reader should be aware that problems may be different for BCI used outside the laboratories.

2.1.2. Considering time variations of EEG.
Most brain activity patterns used to drive BCI are related to particular time variations of EEG, possibly in specific frequency bands [1]. Therefore, the time course of EEG signals should be taken into account during feature extraction [20]. To use this temporal information, three main approaches have been proposed:

- *concatenation of features from different time segments:* it consists in extracting features from several time segments and concatenating them into a single feature vector [11, 20];

- *combination of classifications at different time segments:* it consists in performing the feature extraction and classification steps on several time segments and then combining the results of the different classifiers [21, 22];
- *dynamic classification:* it consists in extracting features from several time segments to build a temporal sequence of feature vectors. This sequence can be classified using a dynamic classifier [20, 23] (see section 2.2.1).

The first approach is the most widely used, which explains why feature vectors are often of high dimensionality.

2.2. Classification algorithms

In order to choose the most appropriate classifier for a given set of features, the properties of the available classifiers must be known. This section provides a classifier taxonomy. It also deals with two classification problems especially relevant for BCI research, namely, the curse-of-dimensionality and the bias–variance tradeoff.

2.2.1. Classifier taxonomy.
Several definitions are commonly used to describe the different kinds of available classifiers:

Generative–discriminative. Generative (also known as informative) classifiers, e.g., Bayes quadratic, learn the class models. To classify a feature vector, generative classifiers compute the likelihood of each class and choose the most likely. Discriminative ones, e.g., support vector machines, only learn the way of discriminating the classes or the class membership in order to classify a feature vector directly [24, 25];

Static–dynamic. Static classifiers, e.g., multilayer perceptrons, cannot take into account temporal information during classification as they classify a single feature vector. In contrast, dynamic classifiers, e.g., the hidden Markov model, can classify a sequence of feature vectors and thus catch temporal dynamics [26].

Stable–unstable. Stable classifiers, e.g., linear discriminant analysis, have a low complexity (or capacity [27]). They are said to be stable as small variations in the training set do not considerably affect their performance. In contrast, unstable classifiers, e.g., multilayer perceptron, have a high complexity. As for them, small variations of the training set may lead to important changes in performance [28].

Regularized. Regularization consists in carefully controlling the complexity of a classifier in order to prevent overtraining. A regularized classifier has good generalization performances and is more robust with respect to outliers [5, 9].

2.2.2. Main classification problems in BCI research.
While performing a pattern recognition task, classifiers may be facing several problems related to the feature properties such as outliers, overtraining, etc. In the field of BCI, two main

problems need to be underlined: the curse-of-dimensionality and the bias–variance tradeoff.

The curse-of-dimensionality. The amount of data needed to properly describe the different classes increases exponentially with the dimensionality of the feature vectors [9, 29]. Actually, if the number of training data is small compared to the size of the feature vectors, the classifier will most probably give poor results. It is recommended to use, at least, five to ten times as many training samples per class as the dimensionality [30, 31]. Unfortunately this cannot be applied in all BCI systems as generally the dimensionality is high and the training set small (see section 2.1.1). Therefore this 'curse' is a major concern in BCI design.

The bias–variance tradeoff. Formally, classification consists in finding the true label y^* of a feature vector x using a mapping f. This mapping is learnt from a training set T. The best mapping f^* that has generated the labels is, of course, unknown. If we consider the mean square error (MSE), classification errors can be decomposed into three terms [28, 29]:

$$
\begin{aligned}
\text{MSE} &= E[(y^* - f(x))^2] \\
&= E[(y^* - f^*(x) + f^*(x) - E[f(x)] \\
&\quad + E[f(x)] - f(x))^2] \\
&= E[(y^* - f^*(x))^2] + E[(f^*(x) - E[f(x)]^2)] \\
&\quad + E[(E[f(x)] - f(x))^2] \\
&= \text{Noise}^2 + \text{Bias}(f(x))^2 + \text{Var}(f(x)). \quad (1)
\end{aligned}
$$

These three terms describe three possible sources of classification error:

- *noise:* represents the noise within the system. This is an irreducible error;
- *bias:* represents the divergence between the estimated mapping and the best mapping. Therefore, it depends on the method that has been chosen to obtain f (linear, quadratic, ...);
- *variance:* reflects the sensitivity to the training set T used.

To attain the lowest classification error, both the bias and the variance must be low. Unfortunately, there is a 'natural' bias–variance tradeoff. Actually, stable classifiers tend to have a high bias and a low variance, whereas unstable classifiers have a low bias and a high variance. This can explain why simple classifiers sometimes outperform more complex ones. Several techniques, known as stabilization techniques, can be used to reduce the variance. Among them, we can quote combination of classifiers [28] and regularization (see section 2.2.1).

EEG signals are known to be non-stationary. Training sets coming from different sessions are likely to be relatively different. Thus, a low variance can be a solution to cope with the variability problem in BCI systems.

3. Survey of classifiers used in BCI research

This section surveys the classification algorithms used to design BCI systems. They are divided into five different categories: linear classifiers, neural networks,
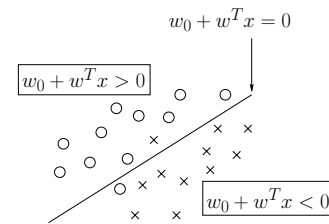

Figure 1. A hyperplane which separates two classes: the 'circles' and the 'crosses'.

nonlinear Bayesian classifiers, nearest neighbor classifiers and combinations of classifiers. The most popular are briefly described and their most important properties for BCI applications are highlighted.

3.1. Linear classifiers

Linear classifiers are discriminant algorithms that use linear functions to distinguish classes. They are probably the most popular algorithms for BCI applications. Two main kinds of linear classifiers have been used for BCI design, namely, linear discriminant analysis (LDA) and support vector machine (SVM).

3.1.1. Linear discriminant analysis. The aim of LDA (also known as Fisher's LDA) is to use hyperplanes to separate the data representing the different classes [5, 32]. For a two-class problem, the class of a feature vector depends on which side of the hyperplane the vector is (see figure 1).

LDA assumes normal distribution of the data, with equal covariance matrix for both classes. The separating hyperplane is obtained by seeking the projection that maximizes the distance between the two classes' means and minimizes the interclass variance [32]. To solve an N-class problem ($N > 2$) several hyperplanes are used. The strategy generally used for multiclass BCI is the 'one versus the rest' (OVR) strategy which consists in separating each class from all the others.

This technique has a very low computational requirement which makes it suitable for the online BCI system. Moreover this classifier is simple to use and generally provides good results. Consequently, LDA has been used with success in a great number of BCI systems such as motor imagery based BCI [33], P300 speller [34], multiclass [35] or asynchronous [36] BCI. The main drawback of LDA is its linearity that can provide poor results on complex nonlinear EEG data [37].

A regularized Fisher's LDA (RFLDA) has also been used in the field of BCI [38, 39]. This classifier introduces a regularization parameter C that can allow or penalize classification errors on the training set. The resulting classifier can accommodate outliers and obtain better generalization capabilities. As outliers are common in EEG data, this regularized version of LDA may give better results for BCI than the non-regularized version [38, 39]. Surprisingly, RFLDA is much less used than LDA for BCI applications.
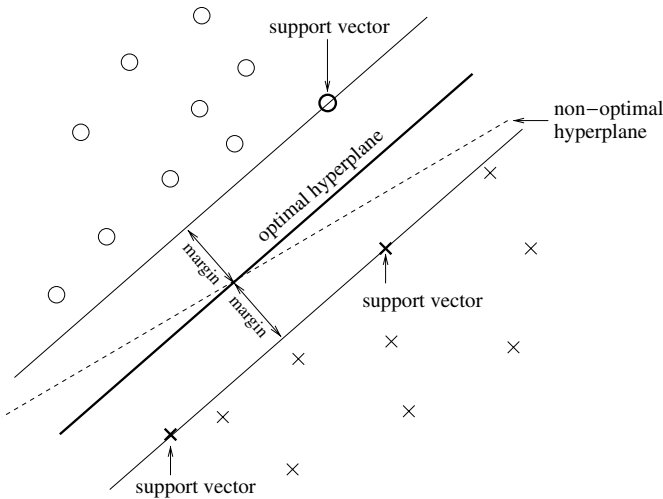
Figure 2. SVM find the optimal hyperplane for generalization.

3.1.2. Support vector machine. An SVM also uses a discriminant hyperplane to identify classes [40, 41]. However, concerning SVM, the selected hyperplane is the one that maximizes the margins, i.e., the distance from the nearest training points (see figure 2). Maximizing the margins is known to increase the generalization capabilities [40, 41]. As RFLDA, an SVM uses a regularization parameter C that enables accommodation to outliers and allows errors on the training set.

Such an SVM enables classification using linear decision boundaries and is known as linear SVM. This classifier has been applied, always with success, to a relatively large number of synchronous BCI problems [19, 35, 38]. However, it is possible to create nonlinear decision boundaries, with only a low increase of the classifier's complexity, by using the 'kernel trick'. It consists in implicitly mapping the data to another space, generally of much higher dimensionality, using a kernel function $K(x, y)$. The kernel generally used in BCI research is the Gaussian or radial basis function (RBF) kernel:

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

The corresponding SVM is known as Gaussian SVM or RBF SVM [40, 41]. RBF SVM have also given very good results for BCI applications [10, 35]. As LDA, SVM has been applied to multiclass BCI problems using the OVR strategy [42].

SVM have several advantages. Actually, thanks to the margin maximization and the regularization term, SVM are known to have good generalization properties [9, 41], to be insensitive to overtraining [9] and to the curse-of-dimensionality [40, 41]. Finally, SVM have a few hyperparameters that need to be defined by hand, namely, the regularization parameter C and the RBF width σ if using kernel (2). These advantages are gained at the expense of a low speed of execution.

3.2. Neural networks

Neural networks (NN) are, together with linear classifiers, the category of classifiers mostly used in BCI research (see, e.g., [43, 44]). Let us recall that an NN is an assembly of several artificial neurons which enables us to produce nonlinear decision boundaries [45].

This section first describes the most widely used NN for BCI, which is the multilayer perceptron (MLP). Then, it briefly presents other architectures of neural network used for BCI applications.

3.2.1. Multilayer perceptron. An MLP is composed of several layers of neurons: an input layer, possibly one or several hidden layers and an output layer [45]. Each neuron's input is connected with the output of the previous layer's neurons whereas the neurons of the output layer determine the class of the input feature vector.

Neural networks and thus MLP are universal approximators, i.e., when composed of enough neurons and layers, they can approximate any continuous function. Added to the fact that they can classify any number of classes, this makes NN very flexible classifiers that can adapt to a great variety of problems. Consequently, MLP, which are the most popular NN used in classification, have been applied to almost all BCI problems such as binary [46] or multiclass [44], synchronous [20] or asynchronous [12] BCI. However, the fact that MLP are universal approximators makes these classifiers sensitive to overtraining, especially with such noisy and non-stationary data as EEG, e.g., [47]. Therefore, careful architecture selection and regularization is required [9].

A multilayer perceptron without hidden layers is known as a perceptron. Interestingly enough, a perceptron is equivalent to LDA and, as such, has been sometimes used for BCI applications [18, 48].

3.2.2. Other neural network architectures. Other types of NN architecture are used in the field of BCI. Among them, one deserves a specific attention as it has been specifically created for BCI: the Gaussian classifier [49, 50]. Each unit of this NN is a Gaussian discriminant function representing a class prototype. According to its authors, this NN outperforms MLP on BCI data and can perform efficient rejection of uncertain samples [49]. As a consequence, this classifier has been applied with success to motor imagery [51] and mental task classification [49], particularly during asynchronous experiments [49, 52].

Besides the Gaussian classifier, several other NN have been applied to BCI purposes, in a more marginal way. They are not described here, due to space limitations:

- learning vector quantization (LVQ) neural network [53, 54];
- fuzzy ARTMAP neural network [55, 56];
- dynamic neural networks such as the finite impulse response neural network (FIRNN) [20], time-delay neural network (TDNN) or gamma dynamic neural network (GDNN) [57];
- RBF neural network [5, 58];

- Bayesian logistic regression neural network (BLRNN) [8];
- adaptive logic network (ALN) [59];
- probability estimating guarded neural classifier (PeGNC) [60].

3.3. Nonlinear Bayesian classifiers

This section introduces two Bayesian classifiers used for BCI: Bayes quadratic and hidden Markov model (HMM). Although Bayesian graphical network (BGN) has been employed for BCI, it is not described here as it is not common and, currently, not fast enough for real-time BCI [61, 62].

All these classifiers produce nonlinear decision boundaries. Furthermore, they are generative, which enables them to perform more efficient rejection of uncertain samples than discriminative classifiers. However, these classifiers are not as widespread as linear classifiers or neural networks in BCI applications.

3.3.1. Bayes quadratic.
Bayesian classification aims at assigning to a feature vector the class it belongs to with the highest probability [5, 32]. The Bayes rule is used to compute the so-called *a posteriori* probability that a feature vector has of belonging to a given class [32]. Using the MAP (maximum *a posteriori*) rule and these probabilities, the class of this feature vector can be estimated.

Bayes quadratic consists in assuming a different normal distribution of data. This leads to quadratic decision boundaries, which explains the name of the classifier. Even though this classifier is not widely used for BCI, it has been applied with success to motor imagery [22, 51] and mental task classification [63, 64].

3.3.2. Hidden Markov model.
Hidden Markov Models (HMM) are popular dynamic classifiers in the field of speech recognition [26]. An HMM is a kind of probabilistic automaton that can provide the probability of observing a given sequence of feature vectors [26]. Each state of the automaton can modelize the probability of observing a given feature vector. For BCI, these probabilities usually are Gaussian mixture models (GMM), e.g., [23].

HMM are perfectly suitable algorithms for the classification of time series [26]. As EEG components used to drive BCI have specific time courses, HMM have been applied to the classification of temporal sequences of BCI features [23, 52, 65] and even to the classification of raw EEG [66]. HMM are not very widespread within the BCI community but these studies revealed that they were promising classifiers for BCI systems.

Another kind of HMM which has been used to design BCI is the input–output HMM (IOHMM) [12]. IOHMM is not a generative classifier but a discriminative one. The main advantage of this classifier is that one IOHMM can discriminate several classes, whereas one HMM per class is needed to achieve the same operation.

3.4. Nearest neighbor classifiers

The classifiers presented in this section are relatively simple. They consist in assigning a feature vector to a class according to its nearest neighbor(s). This neighbor can be a feature vector from the training set as in the case of k nearest neighbors (kNN), or a class prototype as in Mahalanobis distance. They are discriminative nonlinear classifiers.

3.4.1. k nearest neighbors.
The aim of this technique is to assign to an unseen point the dominant class among its k nearest neighbors within the training set [5]. For BCI, these nearest neighbors are usually obtained using a metric distance, e.g., [38]. With a sufficiently high value of k and enough training samples, kNN can approximate any function which enables it to produce nonlinear decision boundaries.

KNN algorithms are not very popular in the BCI community, probably because they are known to be very sensitive to the curse-of-dimensionality [29], which made them fail in several BCI experiments [38, 39, 42]. However, when used in BCI systems with low dimensional feature vectors, kNN may prove to be efficient [67].

3.4.2. Mahalanobis distance.
Mahalanobis distance based classifiers assume a Gaussian distribution $N(\mu_c, M_c)$ for each prototype of the class c. Then, a feature vector x is assigned to the class that corresponds to the nearest prototype, according to the so-called Mahalanobis distance $d_c(x)$ [52]:

$$d_c(x) = \sqrt{(x - \mu_c) M_c^{-1} (x - \mu_c)^T}. \tag{3}$$

This leads to a simple yet robust classifier, which even proved to be suitable for multiclass [42] or asynchronous BCI systems [52]. Despite its good performances, it is still scarcely used in the BCI literature.

3.5. Combinations of classifiers

In most papers related to BCI, the classification is achieved using a single classifier. A recent trend, however, is to use several classifiers, aggregated in different ways. The classifier combination strategies used in BCI applications are the following:

Boosting. Boosting consists in using several classifiers in cascade, each classifier focusing on the errors committed by the previous ones [5]. It can build up a powerful classifier out of several weak ones, and it is unlikely to overtrain. Unfortunately, it is sensitive to mislabels [9] which may explain why it was not successful in one BCI study [68]. To date, in the field of BCI, boosting has been experimented with MLP [68] and ordinary least square (OLS) [69].

Voting. While using voting, several classifiers are being used, each of them assigning the input feature vector to a class. The final class will be that of the majority [9]. Voting is the most popular way of combining classifiers in BCI research, probably because it is simple and efficient. For instance, voting with LVQ NN [54], MLP [70] or SVM [19] has been attempted.

Stacking. Stacking consists in using several classifiers, each of them classifying the input feature vector. These classifiers are

Table 1. Properties of classifiers used in BCI research.

	Linear	Nonlinear	Generative	Discriminant	Dynamic	Static	Regularized	Stable	Unstable	High dimension robust
FLDA	X			X		X		X		
RFLDA	X			X		X	X	X		
Linear-SVM	X			X		X	X	X		X
RBF-SVM		X		X		X	X	X		X
MLP		X		X		X			X	
BLR NN		X		X		X			X	
ALN NN		X		X		X			X	
TDNN		X		X	X				X	
FIRNN		X		X	X				X	
GDNN		X		X	X				X	
Gaussian NN		X		X		X			X	
LVQ NN		X		X		X			X	
Perceptron	X			X		X		X		
RBF-NN		X		X		X			X	
PeGNC		X		X		X	X		X	
Fuzzy ARTMAP NN		X		X			X		X	
HMM		X	X		X				X	
IOHMM		X		X	X				X	
Bayes quadratic		X	X			X			X	
Bayes graphical network		X	X			X			X	
kNN		X		X		X			X	
Mahalanobis distance		X		X		X			X	

called level-0 classifiers. The output of each of these classifiers is then given as input to a so-called meta-classifier (or level-1 classifier) which makes the final decision [71]. Stacking has been used in BCI research using HMM as level-0 classifiers, and an SVM as meta classifier [72].

The main advantage of such techniques is that a combination of similar classifiers is very likely to outperform one of the classifiers on its own. Actually, combining classifiers is known to reduce the variance (see section 2.2.2) and thus the classification error [28, 29].

3.6. Conclusion

A great variety of classifiers have been tried in BCI research. Their properties are summarized in table 1. It should be stressed that some famous kinds of classifiers have not been attempted in BCI research. The two most relevant ones are decision trees [9] and the whole category of fuzzy classifiers [73]. Furthermore, different combination schemes of classifiers have been used, but several other efficient and famous ones can be found in the literature such as Bagging or Arcing [9, 28]. Such algorithms could prove useful as they all succeeded in several other pattern recognition problems. As an example, preliminary results using a fuzzy classifier for BCI purposes are promising [74].

4. Guidelines to choose a classifier

This section assesses the use of the algorithms presented in section 3. It aims at providing the readers with guidelines to help them choose a classifier adapted to a given context. The performances of the BCI using the classifiers described above are gathered in tables, in the appendix. Several measures of performance have been proposed in BCI, such as accuracy of classification, Kappa coefficient [42], mutual information [75], sensitivity and specificity [76]. The most common one is the accuracy of classification, i.e., the percentage of correctly classified feature vectors. Consequently, this review only considers this particular measure.

Two different points of view are proposed. The first identifies the best classifier(s) for a given kind of BCI whereas the second identifies the best classifier(s) for given kinds of features.

4.1. Which classifier goes with which BCI?

Different classifiers were shown to be efficient according to the kind of BCI they were used in. More specifically, different results were observed between synchronous and asynchronous BCI.

4.1.1. The synchronous BCI. The synchronous case is the most widely spread. Three kinds of classification algorithms proved to be particularly efficient in this context, namely, support vector machines, dynamic classifiers and combinations of classifiers. Unfortunately they have not been

Table A1. Accuracy of classifiers in *movement intention* based BCI.

Protocol	Preprocessing	Features	Classification	Accuracy (%)	References
Finger: on BCI competition 2003 data set IV	CSSD	ERD and Bereitschaftspotential based features	Perceptron	84	[48]
	14–26 Hz band pass + CSP	Activity in two brain regions with sLORETA	Perceptron	83	[18]
	bandpass	PCA+CSP+OLS	SVM	90	[77]
Speech muscles	0.5–15 Hz band pass	Raw EEG	MLP	100	[43]
Finger/toe		Raw EEG	TDNN	<90	[57]
			GDNN	≃90	
	0–5 Hz band pass	Amplitude values of smoothed EEG	kNN	78.4	[38]
			Linear SVM	96.8	
			RFLDA	96.7	
			LDA	96.3	
Finger: on different data sets		Amplitude values of smoothed EEG	Gaussian SVM	93.3	[39]
			Linear SVM	92.6	
			RFLDA	93.7	
			LDA	90.7	
			kNN	78	
		BP	Voting with LVQ NN	≃85	[54]
Finger: in asynchronous mode	1–4 Hz band pass	Bi-scale wavelet features	1-NN	up to 81	[76]
	1–4 Hz band pass + PCA	Normalized bi-scale wavelet features		97	[67]

compared with each other yet. Justifications and possible reasons for such an efficiency are given hereafter.

Support vector machines. SVM reached the best results in several synchronous experiments, should it be in its linear [38, 42] or nonlinear form [10, 35], in binary [37, 38, 77] or multiclass [35, 42] BCI (see tables A1, A3, A4 and A5). RFLDA shares several properties with SVM such as being a linear and regularized classifier. Its training algorithm is even very close to the SVM one. Consequently, it also reached very interesting results in some experiments [38, 39].

The first reason for this success may be regularization. Actually, BCI features are often noisy and likely to contain outliers [39]. Regularization may overcome this problem and increase the generalization capabilities of the classifier. As a consequence, regularized classifiers, and more particularly linear SVM, have outperformed unregularized ones of the same kind, i.e., LDA, during several BCI studies [38, 39, 42]. Similarly a nonlinear SVM has outperformed an unregularized nonlinear classifier, namely, an MLP, in another BCI study [35].

The second reason may be the simplicity of SVM. Indeed, the decision rule of SVM is a simple linear function in the kernel space which makes SVM stable and therefore have a low variance. Since BCI features are very unstable over time,

having a low variance may also be a key for low classification error in BCI.

The last reason probably is the robustness of SVM with respect to the curse-of-dimensionality (see section 2.1.1). This has enabled SVM to obtain very good results even with very high dimensional feature vectors and a small training set [19, 42]. However, SVM are not drawback free for BCI as they generally are slower than other classifiers. Luckily, they are fast enough for real-time BCI, e.g., [78].

Dynamic classifiers. Dynamic classifiers almost always outperformed static ones during synchronous BCI experiments [20, 23, 57] (see table A1). Reference [79] is an exception, but the authors admitted that the chosen HMM architecture may not have been suitable. Dynamic classifiers probably are successful in BCI because they can catch the relevant temporal variations present in the extracted features. Furthermore, classifying a sequence of low dimensional feature vectors, instead of a very high dimensional one, in a way, solves the curse-of-dimensionality. Finally, using dynamic classifiers in synchronous BCI also solves the problem of finding the optimal instant for classification as the whole time sequence is classified and not just a particular time window [23].

Combination of classifiers. Combining classifiers was shown efficient [19, 69, 72] and almost always outperformed a

Table A2. Accuracy of classifiers in *mental task imagination* based BCI. These tasks are (t1) visual stimulus driven letter imagination, (t2) auditory stimulus driven letter imagination, (t3) left motor imagery, (t4) right motor imagery, (t5) relax (baseline), (t6) mental mathematics, (t7) mental letter composing, (t8) visual counting, (t9) geometric figure rotation, (t10) mental word generation.

Protocol	Preprocessing	Features	Classification	Accuracy (%)	References
t1+t2	PCA + ICA	LPC spectra	RBF NN	77.8	[58]
(t3 or t4)+t6		Lagged AR	BLRNN	80 ± 12	[8, 21]
t5+t6		Multivariate AR	MLP	91.4	[83]
t5 + best of {t6,t7,t8,t9}		BP	MLP	95	[46]
All couples between {t5,t6,t7,t8,t9} on Keirn and Aunon EEG data		AR	BGN	90.63	[61]
			MLP	88.97	
		AR	Bayes quadratic	84.6	[63]
		PSD		81.0	
		AR	Bayes quadratic	90.51 ± 3.2	[79]
			BGN	88.57 ± 3.0	
			MLP	87.38 ± 3.4	
			LDA	86.63 ± 3.3	
			HMM	65.33 ± 8.5	
Best triplet between {t5,t6,t7,t8,t9}		WK-Parzen PSD	Fuzzy ARTMAP NN	94.43	[56]
t3+t4+t5		Cross ambiguity functions	Mahalanobis distance + MLP	≃71	[84]
t5+t6+t7+t8+t9 on Keirn and Aunon EEG data		PSD with Welch's periodogram	MLP	90.6–98.6	[64]
			Bayes quadratic	89.2–98.6	
		AR	MLP	up to 71	[44]
	0.1–100 Hz band pass	AR	LDA	66	[35]
			MLP	69.4	
			Gaussian SVM	72	
t3+t4+t5 in asynchronous mode	SL + 4–45 Hz band pass	PSD with Welch's periodogram	Gaussian classifier	84	[49]
t3+t4+t10 in asynchronous mode	SL	PSD	HMM	77.35	[12]
			GMM	76.5	
			IOHMM	81.6	
			MLP	79.4	

single one [19, 70, 72] (see table A3). Similarly, on data set IIb of BCI competition 2003, the best results, i.e., best accuracy and smallest number of repetitions, were obtained with combinations of classifiers, namely, boosting of OLS [69] and voting of SVM [19] (see table A5). The study in [68] is an exception as a boosting of MLP was outperformed by a single LDA. This may be explained by the sensitivity of boosting to mislabels [9] and the fact that these mislabels are likely to occur in such noisy and uncertain data as EEG signals. Therefore, combinations such as voting or stacking may be preferred for BCI applications.

As seen in section 3.5, the combination of classifiers helps reduce the variance component of the classification error which generally makes combinations of classifiers more efficient than their single counterparts [28]. Furthermore, this variance reflects the sensitivity toward the training set used. In BCI experiments, variance can be due to time variability [21, 54], session-to-session variability [19] or subject-to-subject variability. Therefore, variance probably is an important source of error. Combining classifiers may be a way of solving this variability/non-stationarity problem [19], which may explain its success.

4.1.2. The asynchronous BCI. Few asynchronous BCI experiments have been carried out yet, therefore, no optimal classifier can be identified for sure. In this context, it seems that dynamic classifiers do not perform better than static ones [12, 52]. Actually, it is very difficult to identify the beginning of each mental task in asynchronous experiments. Therefore dynamic classifiers cannot use their temporal skills efficiently [12, 52]. Surprisingly, SVM or

Table A3. Accuracy of classifiers in *pure motor imagery* based BCI: *two-class* and *synchronous*. The two classes are left and right imagined hand movements.

Protocol	Preprocessing	Features	Classification	Accuracy (%)	References
On different EEG data		Correlative time-frequency representation	Gaussian SVM	86	[37]
			LDA	61	
		BP	LDA	83.6	[68]
			Boosting with MLPs	76.4	
		Fractal dimension	LDA	80.6	
			Boosting with MLPs	80.4	
		Hjorth parameters	HMM	81.4 ± 12.8	[23]
		AAR	LDA	72.4 ± 8.6	
		AAR parameters	MLP	85.97	[20]
			FIR NN	87.4	
		PCA	HMM	75.70	[72]
			HMM+SVM	78.15	
On BCI competition 2003 data set III		Morlet wavelet	Bayes quadratic integrated over time	89.3	[22]
		AAR	BGN	83.57	[79]
			MLP	84.29	
			Bayes quadratic	82.86	
		Raw EEG	HMM	up to 77.5	[66]
	SL + 4–45 Hz band pass	PSD	Gaussian classifier	65.4	[51]
			LDA	65.6	
			Bayes quadratic	63.4	
			Mahalanobis distance	63.1	

combinations of classifiers have not been used in asynchronous BCI yet.

4.1.3. Partial conclusion. Concerning synchronous BCI, SVM seem to be very efficient regardless of the number of classes. This success may be explained by its good properties, namely, regularization, simplicity and immunity to the curse-of-dimensionality. Besides SVM, combination of classifiers and dynamic classifiers also seem to be very efficient and promising for synchronous BCI. Concerning the asynchronous experiments, due to the current lack of published results, no classifier seems better than the other. The only information that can be deduced is that dynamic classifiers seem to lose their superiority in such experiments.

4.2. Which classifier goes with which kinds of features?

To propose another point of view, this section compares classifiers considering their ability to cope with specific problems of BCI features (see section 2.1.1):

- *noise and outliers:* regularized classifiers, such as SVM, seem appropriate to deal with outliers. Muller *et al* even recommended to systematically regularize the classifiers used in BCI systems in order to cope with outliers [39]. It is also argued that discriminative classifiers perform better than generative ones in the presence of noise or outliers [12];

- *high dimensionality:* SVM probably are the most appropriate classifiers to deal with feature vectors of high dimensionality. If the high dimensionality is due to the use of a large number of time segments, dynamic classifiers can also solve the problem by considering sequences of feature vectors instead of a single vector of very high dimensionality. For instance, SVM [10, 19] and dynamic classifiers such as HMM [66] or TDNN [57] are perfectly able to classify raw EEG. The kNN should not be used in such a case as they are very sensitive to the curse-of-dimensionality. Nevertheless, it is always preferable

Table A4. Accuracy of classifiers in *pure motor imagery* based BCI: *multiclass* and/or *asynchronous* case. The classes are (c1) left imagined hand movements, (c2) right imagined hand movements, (c3) imagined foot movements, (c4) imagined tongue movements, (c5) relax (baseline).

Protocol	Preprocessing	Features	Classification	Accuracy (%)	References
c1+c2+c3 on different data		BP	LVQ NN	$\simeq$60	[85]
		BP	HMM	$\simeq$77.5	[65]
c1+c2+c3+c4 on different data			Linear SVM	63	[42]
			LDA	54.46	
		AAR	kNN	41.74	
			Mahalanobis distance	53.5	
		BP	HMM	$\simeq$63	[65]
c1+c2+c3+c4+c5		BP	HMM	$\simeq$52.6	[65]
c1+c2 in asynchronous mode		Welch power spectrum	Mahalanobis distance	90	[52]
			Gaussian classifier	80	
			HMM	65	
c1+c2+c3 in asynchronous mode		BP	LDA	95	[36]

Table A5. Accuracy of classifiers in *P300 speller* BCI.

Protocol	Preprocessing	Features	Classification	Accuracy (%)	References
BCI competition data set IIb	0.1–20 Hz band pass	Raw EEG	Voting with linear SVMs	100 Four repetitions	[19]
			Linear SVM	100 Ten repetitions	
	0–9 Hz band pass	Raw EEG	Boosting with OLS	100 Four repetitions	[69]
	0.5–30 Hz band	Raw EEG	Gaussian SVM	100 Five repetitions	[10]
		t-Continuous Wavelet Transform	LDA	100 Six repetitions	[34]
Other data	Low pass + PCA	Raw EEG + EEG time derivatives	Gaussian SVM	$\simeq$95 Ten repetitions	[78]

Table A6. Accuracy of classifiers in μ *and* β *rhythm based cursor control* BCI.

Protocol	Preprocessing	Features	Classification	Accuracy (%)	References
Four-targets on BCI competition data set IIa	ICA, CAR μ band pass	BP + CSP features	Voting with MLP	65.18	[70]
			MLP	60.08	
	μ and β band pass + CSP	Fourier power coefficient	RFLDA	74.7	[86]

to have a small number of features [31]. Therefore, it is highly recommended to use dimensionality reduction techniques and/or features selection [39];

- *time information:* for synchronous experiments, dynamic classifiers seem to be the most efficient method to exploit the temporal information contained in features. Similarly, integrating classifiers over time can efficiently utilize the time information [22]. For asynchronous experiments, no clear superiority could be observed (see the previous section);

- *non-stationarity:* a combination of classifiers may solve this problem as it reduces the variance. Stable classifiers such as LDA or SVM can also be used but would probably be outperformed by combinations of LDA or SVM;

Table A7. Accuracy of classifiers in *other* BCI.

Protocol	Preprocessing	Features	Classification	Accuracy (%)	References
Open or close eyes		FFT components for 5–15 Hz	PeGNC	87 but one error	[60]
SCP modulation on different data		t-Continuous Wavelet Transform	LDA	54.4	[34]
		SCP mean + multitaper γ BP	LDA	88.7	[87]

- *small training sets:* if the training set is small, simple techniques with few parameters should be used, such as LDA [30].

5. Conclusion

This review has surveyed classification algorithms used to design brain–computer interfaces (BCI). These algorithms were divided into five categories: linear classifiers, neural networks, nonlinear Bayesian classifiers, nearest neighbor classifiers and combinations of classifiers. The results they obtained, in a BCI context, have been analyzed and compared in order to provide the readers with guidelines to choose or design a classifier for a BCI system. In a nutshell, it seems that SVM are particularly efficient for synchronous BCI. This probably is due to their regularization property and their immunity to the curse-of-dimensionality. Furthermore, combinations of classifiers and dynamic classifiers also seem very efficient in synchronous experiments.

This paper focused on reviewing classifiers used in BCI research, i.e., related to published online or offline studies. However, other existing classification techniques, not currently used for BCI purposes, could be explored and may prove to be rewarding. Furthermore, it should be noted that once BCI are more widely used in clinical practice, new properties will have to be taken into consideration, such as the availability of large data sets or long term variability of EEG signals.

One difficulty encountered in such a study concerns the lack of published objective comparisons between classifiers. Ideally, classifiers should be tested within the same context, i.e., with the same users, using the same feature extraction method and the same protocol. Currently, this is a crucial problem for BCI research. For this reason some researchers have proposed general purpose BCI systems such as the BCI2000 toolkit [80]. This toolkit is a modular framework which makes it possible to easily change the classification, preprocessing or feature extraction modules. With such a system it becomes possible to test several classifiers with the same features and preprocessings. With similar objectives of modularity, the Open-ViBE platform [81] proposes a framework to experiment BCI on various protocols using, for instance, neuro-feedback and virtual reality. An extensive use of such platforms could lead to interesting findings in the future.

Appendix. Classifier performances

Numerous BCI studies using the classifiers described so far have been carried out. The classifier performances are summarized in tables A1–A7. Each table corresponds to a particular protocol and displays the preprocessing and the feature extraction techniques employed. Two kinds of studies have been chosen to appear in these tables. The first one corresponds to studies for which it is possible to objectively compare the algorithms since the EEG data used are benchmark data, such as data from the BCI competition 2003 [82] or personal data on which several classifiers are compared. The second corresponds to studies that assess the usability and/or the efficiency of a classifier for a BCI problem, e.g., [56, 60].

References

[1] Wolpaw J R, Birbaumer N, McFarland D J, Pfurtscheller G and Vaughan T M 2002 Brain–computer interfaces for communication and control *Clin. Neurophysiol.* **113** 767–91
[2] Vaughan T M *et al* 2003 Brain–computer interface technology: a review of the second international meeting *IEEE Trans. Neural Syst. Rehabil. Eng.* **11** 94–109
[3] Kübler A, Kotchoubey B, Kaiser J, Wolpaw J R and Birbaumer N 2001 Brain–computer communication: unlocking the locked in *Psychol. Bull.* **127** 358–75
[4] McFarland D J, Anderson C W, Muller K-R, Schlogl A and Krusienski D J 2006 BCI meeting 2005-workshop on BCI signal processing: feature extraction and translation *IEEE Trans. Neural Syst. Rehabil. Eng.* **14** 135–8
[5] Duda R O, Hart P E and Stork D G 2001 *Pattern Recognition* 2nd edn (New York: Wiley-Interscience)
[6] Vallabhaneni A, Wang T and He B 2005 Brain computer interface *Neural Engineering* ed B He (Dordrecht: Kluwer) pp 85–122
[7] McFarland D J and Wolpaw J R 2005 Sensorimotor rhythm-based brain–computer interface (BCI): feature selection by regression improves performance *IEEE Trans. Neural Syst. Rehabil. Eng.* **13** 372–9

[8] Penny W D, Roberts S J, Curran E A and Stokes M J 2000 EEG-based communication: a pattern recognition approach *IEEE Trans. Rehabil. Eng.* **8** 214–5

[9] Jain A K, Duin R P W and Mao J 2000 Statistical pattern recognition: a review *IEEE Trans. Pattern Anal. Mach. Intell.* **22** 4–37

[10] Kaper M, Meinicke P, Grossekathoefer U, Lingner T and Ritter H 2004 BCI competition 2003–data set iib: support vector machines for the p300 speller paradigm *IEEE Trans. Biomed. Eng.* **51** 1073–6

[11] Pfurtscheller G, Neuper C, Flotzinger D and Pregenzer M 1997 EEG-based discrimination between imagination of right and left hand movement *Electroencephalogr. Clin. Neurophysiol.* **103** 642–51

[12] Chiappa S and Bengio S 2004 HMM and IOHMM modeling of EEG rhythms for asynchronous BCI systems *European Symposium on Artificial Neural Networks ESANN*

[13] Millán J R and Mouriño J 2003 Asynchronous BCI and local neural classifiers: an overview of the adaptive brain interface project *IEEE Trans. Neural Syst. Rehabil. Eng.* **11** 159–61

[14] Pfurtscheller G, Neuper C, Schlogl A and Lugger K 1998 Separability of EEG signals recorded during right and left motor imagery using adaptive autoregressive parameters *IEEE Trans. Rehabil. Eng.* **6** 316–25

[15] Wang T, Deng J and He B 2004 Classifying EEG-based motor imagery tasks by means of time-frequency synthesized spatial patterns *Clin. Neurophysiol.* **115** 2744–53

[16] Qin L, Ding L and He B 2004 Motor imagery classification by means of source analysis for brain–computer interface applications *J. Neural Eng.* **1** 135–141

[17] Kamousi B, Liu Z and He B 2005 Classification of motor imagery tasks for brain–computer interface applications by means of two equivalent dipoles analysis *IEEE Trans. Neural Syst. Rehabil. Eng.* **13** 166–71

[18] Congedo M, Lotte F and Lécuyer A 2006 Classification of movement intention by spatially filtered electromagnetic inverse solutions *Phys. Med. Biol.* **51** 1971–89

[19] Rakotomamonjy A, Guigue V, Mallet G and Alvarado V 2005 Ensemble of SVMs for improving brain computer interface p300 speller performances *Int. Conf. on Artificial Neural Networks*

[20] Haselsteiner E and Pfurtscheller G 2000 Using time-dependant neural networks for EEG classification *IEEE Trans. Rehabil. Eng.* **8** 457–63

[21] Penny W D and Roberts S J 1999 EEG-based communication via dynamic neural network models *Proc. Int. Joint Conf. on Neural Networks*

[22] Lemm S, Schafer C and Curio G 2004 BCI competition 2003–data set iii: probabilistic modeling of sensorimotor mu rhythms for classification of imaginary hand movements *IEEE Trans. Biomed. Eng.* **51** 1077–80

[23] Obermeier B, Guger C, Neuper C and Pfurtscheller G 2001 Hidden Markov models for online classification of single trial EEG *Pattern Recognit. Lett.* 1299–309

[24] Ng A and Jordan M 2002 On generative versus discriminative classifiers: a comparison of logistic regression and naive Bayes *Proc. Advances in Neural Information Processing*

[25] Rubinstein Y D and Hastie T 1997 Discriminative versus informative learning *Proc. 3rd Int. Conf. on Knowledge Discovery and Data Mining*

[26] Rabiner L R 1989 A tutorial on hidden Markov models and selected applications in speech recognition *Proc. IEEE* **77** 257–86

[27] Vapnik V N 1999 An overview of statistical learning theory *IEEE Trans. Neural Netw.* **10** 988–99

[28] Breiman L 1998 Arcing classifiers *Ann. Stat.* **26** 801–49

[29] Friedman J H K 1997 On bias, variance, 0/1-loss, and the curse-of-dimensionality *Data Min. Knowl. Discov.* **1** 55–77

[30] Raudys S J and Jain A K 1991 Small sample size effects in statistical pattern recognition: recommendations for practitioners *IEEE Trans. Pattern Anal. Mach. Intell.* **13** 252–64

[31] Jain A K and Chandrasekaran B 1982 Dimensionality and sample size considerations in pattern recognition practice *Handbook of Statistics* vol 2, ed P R Krishnaiah and L N Kanal (Amsterdam: North-Holland)

[32] Fukunaga K 1990 *Statistical Pattern Recognition* 2nd edn (New York: Academic)

[33] Pfurtscheller G 1999 EEG event-related desynchronization (ERD) and event-related synchronization (ERS) *Electroencephalography: Basic Principles, Clinical Applications and Related Fields* 4th edn, ed E Niedermeyer and F H Lopes da Silva (Baltimore, MD: Williams and Wilkins) pp 958–67

[34] Bostanov V 2004 BCI competition 2003–data sets ib and iib: feature extraction from event-related brain potentials with the continuous wavelet transform and the t-value scalogram *IEEE Trans. Biomed. Eng.* **51** 1057–61

[35] Garrett D, Peterson D A, Anderson C W and Thaut M H 2003 Comparison of linear, nonlinear, and feature selection methods for EEG signal classification *IEEE Trans. Neural Syst. Rehabil. Eng.* **11** 141–4

[36] Scherer R, Muller G R, Neuper C, Graimann B and Pfurtscheller G 2004 An asynchronously controlled EEG-based virtual keyboard: improvement of the spelling rate *IEEE Trans. Biomed. Eng.* **51** 979–84

[37] Garcia G N, Ebrahimi T and Vesin J-M 2003 Support vector EEG classification in the Fourier and time-frequency correlation domains *Conference Proc. 1st Int. IEEE EMBS Conf. on Neural Engineering*

[38] Blankertz B, Curio G and Müller K R 2002 Classifying single trial EEG: towards brain computer interfacing *Adv. Neural Inf. Process. Syst. (NIPS 01)* **14** 157–64

[39] Müller K R, Krauledat M, Dornhege G, Curio G and Blankertz B 2004 Machine learning techniques for brain–computer interfaces *Biomed. Technol.* **49** 11–22

[40] Burges C J C 1998 A tutorial on support vector machines for pattern recognition *Knowl. Discov. Data Min.* **2** 121–67

[41] Bennett K P and Campbell C 2000 Support vector machines: hype or hallelujah? *ACM SIGKDD Explor. Newslett.* **2** 1–13

[42] Schlögl A, Lee F, Bischof H and Pfurtscheller G 2005 Characterization of four-class motor imagery EEG data for the BCI-competition 2005 *J. Neural Eng.* **2** L14–22

[43] Hiraiwa A, Shimohara K and Tokunaga Y 1990 EEG topography recognition by neural networks *IEEE Eng. Med. Biol. Mag.* **9** 39–42

[44] Anderson C W and Sijercic Z 1996 Classification of EEG signals from four subjects during five mental tasks *Solving Engineering Problems with Neural Networks: Proc. Int. Conf. on Engineering Applications of Neural Networks (EANN'96)*

[45] Bishop C M 1996 *Neural Networks for Pattern Recognition* (Oxford: Oxford University Press)

[46] Palaniappan R 2005 Brain computer interface design using band powers extracted during mental tasks *Proc. 2nd Int. IEEE EMBS Conf. on Neural Engineering*

[47] Balakrishnan D and Puthusserypady S 2005 Multilayer perceptrons for the classification of brain computer interface data *Proc. IEEE 31st Annual Northeast Bioengineering Conference*

[48] Wang Y, Zhang Z, Li Y, Gao X, Gao S and Yang F 2004 BCI competition 2003–data set iv: an algorithm based on CSSD and FDA for classifying single-trial EEG *IEEE Trans. Biomed. Eng.* **51** 1081–6

[49] Millán J R, Mouriño J, Cincotti F, Babiloni F, Varsta M and Heikkonen J 2000 Local neural classifier for EEG-based

recognition of mental tasks *IEEE-INNS-ENNS Int. Joint Conf. on Neural Networks*

[50] Millán J R, Renkens F, Mourino J and Gerstner W 2004 Noninvasive brain-actuated control of a mobile robot by human EEG *IEEE Trans. Biomed. Eng.* **51** 1026–33

[51] Solhjoo S and Moradi M H 2004 Mental task recognition: a comparison between some of classification methods *BIOSIGNAL 2004 Int. EURASIP Conference*

[52] Cincotti F, Scipione A, Tiniperi A, Mattia D, Marciani M G, Millán J del R, Salinari S, Bianchi L and Babiloni F 2003 Comparison of different feature classifiers for brain computer interfaces *Proc. 1st Int. IEEE EMBS Conf. on Neural Engineering*

[53] Kohonen T 1990 The self-organizing maps *Proc. IEEE* **78** 1464–80

[54] Pfurtscheller G, Flotzinger D and Kalcher J 1993 Brain–computer interface-a new communication device for handicapped persons *J. Microcomput. Appl.* **16** 293–9

[55] Carpenter G A, Grossberg S, Markuzon N, Reynolds J H and Rosen D B 1992 Fuzzy artmap: a neural network architecture for incrementalsupervised learning of analog multidimensional maps *IEEE Trans. Neural Netw.* **3** 698–713

[56] Palaniappan R, Paramesran R, Nishida S and Saiwaki N 2002 A new brain–computer interface design using fuzzy artmap. *IEEE Trans. Neural Syst. Rehabil. Eng.* **10** 140–8

[57] Barreto A B, Taberner A M and Vicente L M 1996 Classification of spatio-temporal EEG readiness potentials towards the development of a brain–computer interface *Southeastcon '96. 'Bringing Together Education, Science and Technology', Proc. IEEE*

[58] Hoya T, Hori G, Bakardjian H, Nishimura T, Suzuki T, Miyawaki Y, Funase A and Cao J 2003 Classification of single trial EEG signals by a combined principal + independent component analysis and probabilistic neural network approach *Proc. ICA2003* pp 197–202

[59] Kostov A and Polak M 2000 Parallel man-machine training in development of EEG-based cursor control *IEEE Trans. Rehabil. Eng.* **8** 203–5

[60] Felzer T and Freisieben B 2003 Analyzing EEG signals using the probability estimating guarded neural classifier *IEEE Trans. Neural Syst. Rehabil. Eng.* **11** 361–71

[61] Tavakolian K and Rezaei S 2004 Classification of mental tasks using Gaussian mixture Bayesian network classifiers *Proc. IEEE Int. Workshop on Biomedical Circuits and Systems*

[62] Rezaei S, Tavakolian K, Nasrabadi A M and Setarehdan S K 2006 Different classification techniques considering brain computer interface applications *J. Neural. Eng.* **3** 139–44

[63] Keirn Z A and Aunon J I 1990 A new mode of communication between man and his surroundings *IEEE Trans. Biomed. Eng.* **37** 1209–14

[64] Barreto G A, Frota R A and de Medeiros F N S 2004 On the classification of mental tasks: a performance comparison of neural and statistical approaches *Proc. IEEE Workshop on Machine Learning for Signal Processing*

[65] Obermaier B, Neuper C, Guger C and Pfurtscheller G 2000 Information transfer rate in a five-classes brain–computer interface *IEEE Trans. Neural Syst. Rehabil. Eng.* **9** 283–8

[66] Solhjoo S, Nasrabadi A M and Golpayegani M R H 2005 Classification of chaotic signals using HMM classifiers: EEG-based mental task classification *Proc. European Signal Processing Conference*

[67] Borisoff J F, Mason S G, Bashashati A and Birch G E 2004 Brain–computer interface design for asynchronous control applications: improvements to the lf-asd asynchronous brain switch *IEEE Trans. Biomed. Eng.* **51** 985–92

[68] Boostani R and Moradi M H 2004 A new approach in the BCI research based on fractal dimension as feature and Adaboost as classifier *J. Neural Eng.* **1** 212–7

[69] Hoffmann U, Garcia G, Vesin J-M, Diserens K and Ebrahimi T 2005 A boosting approach to p300 detection with application to brain–computer interfaces *Conference Proc. 2nd Int. IEEE EMBS Conf. Neural Engineering*

[70] Qin J, Li Y and Cichocki A 2005 ICA and committee machine-based algorithm for cursor control in a BCI system *Lecture Notes in Computer Science*

[71] Wolpert D H 1992 Stacked generalization *Neural Netw.* **5** 241–59

[72] Lee H and Choi S 2003 PCA+HMM+SVM for EEG pattern classification *Proc. 7th Int. Symp. on Signal Processing and its Applications*

[73] Bezdec J C and Pal S K 1992 *Fuzzy Models For Pattern Recognition* (New York: IEEE)

[74] Lotte F 2006 The use of fuzzy inference systems for classification in EEG-based brain–computer interfaces *Proc. 3rd Int. Brain–Computer Interface Workshop and Training Course* pp 12–13

[75] Schlogl A, Neuper C and Pfurtscheller G 2002 Estimating the mutual information of an EEG-based brain–computer interface *Biomed. Tech.* **47** 3–8

[76] Mason S G and Birch G E 2000 A brain-controlled switch for asynchronous control applications *IEEE Trans. Biomed. Eng.* **47** 1297–307

[77] Xu W, Guan C, Siong C E, Ranganatha S, Thulasidas M and Wu J 2004 High accuracy classification of EEG signal *17th Int. Conf. on Pattern Recognition (ICPR'04)* vol 2 pp 391–4

[78] Thulasidas M, Guan C and Wu J 2006 Robust classification of EEG signal for brain–computer interface *IEEE Trans. Neural Syst. Rehabil. Eng.* **14** 24–9

[79] Tavakolian K, Vasefi F, Naziripour K and Rezaei S 2006 Mental task classification for brain computer interface applications *Proc. Canadian Student Conf. on Biomedical Computing*

[80] Schalk G, McFarland D J, Hinterberger T, Birbaumer N and Wolpaw J R 2004 BCI2000: a general-purpose brain–computer interface (BCI) system *IEEE Trans. Biomed. Eng.* **51** 1034–43

[81] Arrouët C, Congedo M, Marvie J E, Lamarche F, Lecuyer A and Arnaldi B 2004 Open-vibe : a 3D platform for real-time neuroscience *J. Neurother.*

[82] Blankertz B *et al* 2004 The BCI competition 2003: Progress and perspectives in detection and discrimination of EEG single trials *IEEE Trans. Biomed. Eng.* **51** 1044–51

[83] Anderson C W, Stolz E A and Shamsunder S 1998 Multivariate autoregressive models for classification of spontaneous electroencephalographic signals during mental tasks *IEEE Trans. Biomed. Eng.* **45** 277–86

[84] Garcia G, Ebrahimi T and Vesin J-M 2002 Classification of EEG signals in the ambiguity domain for brain computer interface applications *14th Int. Conf. on Digital Signal Processing (DSP2002)*

[85] Kalcher J, Flotzinger D, Neuper C, Golly S and Pfurtscheller G 1996 Graz brain–computer interface ii: towards communication between humans and computers based on online classification of three different EEG patterns *Med. Biol. Eng. Comput.* **34** 383–8

[86] Blanchard G and Blankertz B 2004 BCI competition 2003–data set iia: spatial patterns of self-controlled brain rhythm modulations *IEEE Trans. Biomed. Eng.* **51** 1062–6

[87] Mensh B D, Werfel J and Seung H S 2004 BCI competition 2003–data set ia: combining gamma-band power with slow cortical potentials to improve single-trial classification of electroencephalographic signals *IEEE Trans. Biomed. Eng.* **51** 1052–6