J. Neural Eng. **3** (2006) 299–305 doi:10.1088/1741-2560/3/4/007

A comparison of classification techniques for the P300 Speller

Dean J Krusienski[1,4], Eric W Sellers[1], François Cabestaing[2], Sabri Bayoudh[3], Dennis J McFarland[1], Theresa M Vaughan[1] and Jonathan R Wolpaw[1]

[1] Wadsworth Center, New York State Department of Health, Albany, NY 12201, USA
[2] LAGIS Laboratory, UMR-CNRS 8146, Université des Sciences et Technologies de Lille, 59655 Villeneuve d'Ascq, France
[3] IRISA at the Université de Rennes 1, 2 Rue du Thabor, CS 46510, 35065 Rennes Cedex, France

Received 12 July 2006
Accepted for publication 6 October 2006
Published 26 October 2006
Online at stacks.iop.org/JNE/3/299

Abstract

This study assesses the relative performance characteristics of five established classification techniques on data collected using the P300 Speller paradigm, originally described by Farwell and Donchin (1988 *Electroenceph. Clin. Neurophysiol.* **70** 510). Four linear methods: Pearson's correlation method (PCM), Fisher's linear discriminant (FLD), stepwise linear discriminant analysis (SWLDA) and a linear support vector machine (LSVM); and one nonlinear method: Gaussian kernel support vector machine (GSVM), are compared for classifying offline data from eight users. The relative performance of the classifiers is evaluated, along with the practical concerns regarding the implementation of the respective methods. The results indicate that while all methods attained acceptable performance levels, SWLDA and FLD provide the best overall performance and implementation characteristics for practical classification of P300 Speller data.

1. Introduction

A brain–computer interface (BCI) is a device that uses brain signals to provide a non-muscular communication channel [18], particularly for individuals with severe neuromuscular disabilities. The P300 event-related potential, evoked in scalp-recorded electroencephalography (EEG) by external stimuli, has proven to be a reliable response for controlling a BCI [5]. Recent studies have demonstrated that a P300-based BCI trained on a limited amount of data can serve as an effective communication device [1, 13, 14]. In addition, more advanced feature extraction and classification procedures have been implemented, greatly improving the classification performance beyond those reported by Farwell and Donchin on a 6×6 matrix of alphanumeric characters [5]. Several classification techniques have demonstrated notable performance for the P300 Speller, including stepwise linear discriminant analysis [2, 5], support vector machines [8, 10, 11], wavelets [1] and matched filtering [14]. This recent progress has verified the capabilities of P300-based BCI systems and provided the impetus for efforts to improve the speed and accuracy performance of the paradigm.

This study provides a comprehensive comparison of several competitive classification techniques for the P300 Speller: Pearson's correlation method (PCM), Fisher's linear discriminant (FLD), stepwise linear discriminant analysis (SWLDA), linear support vector machine (LSVM) and Gaussian support vector machine (GSVM). PCM and FLD were chosen as simple linear techniques to provide a baseline for comparison. The fundamental difference between these two methods is that PCM only incorporates univariate statistics, while FLD incorporates multivariate statistics. SWLDA is an extension of FLD and was selected because of its successful application to the P300 Speller in earlier work [2, 5]. SVMs were chosen to represent popular modern classifiers that have a theoretical foundation designed to provide several desirable performance characteristics. The LSVM was included for comparison to the linear SWLDA and PCM, and the GSVM was included to evaluate potential gains of nonlinear kernel methods.

[4] Present address: University of North Florida, Jacksonville, FL 32224, USA.

1741-2560/06/040299+07$30.00

1.1. The P300 Speller

The P300 Speller described by Farwell and Donchin presents a 6×6 matrix of characters [5]. Each row and each column are intensified; the intensifications are presented in a random sequence. The user focuses attention on one of the 36 cells of the matrix. The sequence of 12 flashes, 6 rows and 6 columns, constitutes an Oddball Paradigm [4] with the row and the column containing the character to be communicated constituting the rare set, and the other ten intensifications constituting the frequent set. Items that are presented infrequently (the rare set) in a sequential series of randomly presented stimuli will elicit a P300 response if the observer is attending to the stimulus series. Thus, the row and the column containing the target character will elicit a P300 when intensified, because this constitutes a rare event in the context of all other character flashes. Although the P300 response is independent of spatial attention, the relative roles of eye gaze and transient visual responses in the P300 Speller paradigm have yet to be examined.

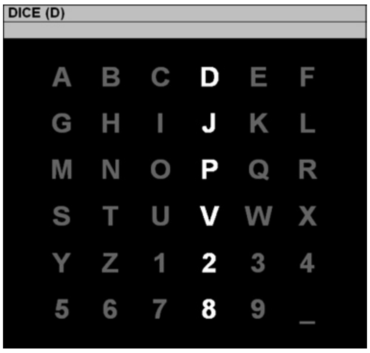

Figure 1. The 6×6 matrix used in the current study. A row or column intensifies for 100 ms every 175 ms. The letter in parentheses at the top of the window is the current target letter 'D'. A P300 should be elicited when the fourth column or first row is intensified. After the intensification sequence for a character epoch, the result is classified and online feedback is provided directly below the character to be copied.

2. Data collection

2.1. Users

Eight people (six men and two women, ages 24–50) were the BCI users in this study. The users varied in their previous BCI experience, but all users had either no or minimal experience with a P300-based BCI system. The study was approved by the New York State Department of Health Institutional Review Board, and each user gave informed consent.

2.2. Task, procedure and design

The user sat upright in front of a video monitor and viewed the matrix display. The task was to focus attention on a specified letter of the matrix and passively count the number of times the target character intensified. All data were collected in the copy speller mode: words were presented on the top left of the video monitor and the character currently specified for attention was listed in parentheses at the end of the letter string (see figure 1). Each session consisted of nine experimental runs; each run was composed of a word or a series of characters prescribed by the investigator. This set of prescribed characters spanned the set of alphanumeric characters in the matrix and was consistent for each user and session. The rows and columns were intensified for 100 ms with 75 ms between intensifications. One character epoch (i.e., one trial) consisted of 15 intensifications of each row and column.

Four of the users were given suboptimal feedback because the classifier was constructed using generic feature weights, not adjusted to match their responses. The other four users used SWLDA feature weights derived from their own previous session's data, and thus were given consistent and accurate online feedback.

2.3. Data acquisition and processing

The EEG was recorded using a cap (Electro-Cap International, Inc.) embedded with 64 electrode locations distributed over the entire scalp, based on the International 10–20 system [15]. All 64 channels were referenced to the right earlobe, and grounded to the right mastoid. The EEG was amplified with an SA Electronics amplifier (20 000×), digitized at a rate of 240 Hz, bandpass filtered 0.1–60 Hz, and stored. All aspects of data collection and experimental control were controlled by the BCI2000 system [12].

2.4. Preprocessing

The channel selection and data preprocessing are based on results found in [9]. In that study, several subsets of 64 channels were systematically evaluated with respect to various data decimation factors and referencing schemes in order to determine the combination that provided maximal classification performance. For each channel in the subset, 800 ms segments of data following each intensification were extracted. The segments were then moving average filtered and decimated by equivalent values. The resulting data segments were concatenated by channel for each intensification, creating a single feature vector for training the classifiers. It was found that the eight-channel ear-referenced subset shown in figure 2, with a moving average window and decimation factor of 12, provided the best general performance. Based on these results, it is presumed that this technique is effective for capturing the essential information of the P300 response for discrimination purposes. Therefore, this channel set and preprocessing technique were adopted for the present study, resulting in a feature vector length of 128 ($192/12$ samples $*$ 8 channels).

3. Classification methods

Determining the presence or absence of a P300 evoked potential from EEG features can be considered a binary classification problem with a discriminant function having a decision hyper-plane defined by:

$$w \cdot f(x) + b = 0 \qquad (1)$$

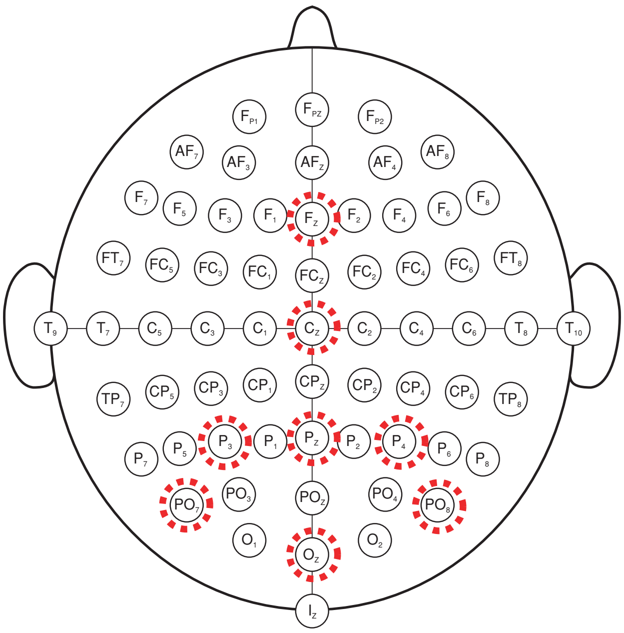

Figure 2. The electrode montage used in the current study [15]. The eight electrodes selected for analysis are indicated by the dotted circles.

where x is the feature vector, $f(\cdot)$ is a transformation function, w is a vector of classification weights and b is the bias term. For nonlinearly separable problems, the $f(\cdot)$ can represent a kernel transformation that maps the features into a higher dimensional space in an attempt to create a linearly separable set. For linear methods, $f(\cdot)$ is simply an identity transformation: $f(x) = x$. All five methods considered are different approaches to solving for w and b. However, because it is assumed that a P300 is elicited for one of the six row/column intensifications, and that the P300 response is invariant to row/column stimuli, the resultant classification is taken as the maximum of the sum of scored feature vectors for the respective rows, as well as for the columns:

$$\text{predicted row} = \underset{\text{rows}}{\arg\max}\left[\sum_{i_{\text{row}}} w \cdot f(x_{i_{\text{row}}})\right] \tag{2}$$

$$\text{predicted column} = \underset{\text{columns}}{\arg\max}\left[\sum_{i_{\text{column}}} w \cdot f(x_{i_{\text{column}}})\right]. \tag{3}$$

By assigning class labels of +1 and -1 to the target and nontarget stimuli, respectively, this design selects the response with the largest positive distance from the trained separating hyper-plane. This is ideally analogous to selecting the response that strongly represents the characteristic P300 as defined by the training data. The predicted character is located at the intersection of the predicted row and column in the matrix. Because equations (2) and (3) are invariant to the constant bias term b, it does not need to be computed. The details of the four linear methods, PCM, FLD, SWLDA, LSVM, and one nonlinear kernel method, GSVM, are described below.

3.1. Pearson's correlation method

Pearson's correlation coefficient [3] is a statistical analysis tool that can be used to test the significance of predictor variables. This coefficient, which measures the correlation between two series $X = (x_i, 1 \leqslant i \leqslant L)$ and $Y = (y_i, 1 \leqslant I \leqslant L)$, is defined by:

$$r = \frac{L\sum x_i y_i - \sum x_i \sum y_i}{\sqrt{\left(L\sum x_i^2 - \left(\sum x_i\right)^2\right)\left(L\sum y_i^2 - \left(\sum y_i\right)^2\right)}} \tag{4}$$

where L is the number of responses in the training set, y_i are the class labels corresponding to each stimulus, and x_i are the values of a single input feature corresponding to each stimulus. It reflects the degree of linear relationship between the two series, and ranges between -1 and +1. If the two series are strictly proportional, r is equal to ± 1. If the two series show no correlation, r is equal to zero. The higher the absolute value of r, the more significant the predictor variable for the model.

To use PCM for discrimination, for each feature in the feature vector, the correlation coefficient between the feature and target observations is computed using equation (4). Rather than selecting only the most significant features for inclusion to the model, all of the respective correlation coefficients are used as the feature weights in equation (1). If a feature is significant, its value will be multiplied by a non-null coefficient and added to the sum. If a feature is not significant, its value will be multiplied by a coefficient near zero and have little impact on the model. This supervised learning scheme, which uses the information contained in each axis of the feature space independently of the others, is very efficient in terms of computational complexity.

3.2. Fisher's linear discriminant

Fisher's linear discriminant [6] is the benchmark method for determining the optimal separating hyper-plane between two classes. FLD is simple to calculate and provides robust classification that is optimal when the two classes are Gaussian with equal covariance. For binary classification tasks such as this, Fisher's linear discriminant and the ordinary least-squares regression solution are equivalent, with the estimated feature weights given as:

$$\hat{w} = (X^T X)^{-1} X^T y \tag{5}$$

where X is the matrix of observed feature vectors and y is the vector of class labels.

3.3. Stepwise linear discriminant analysis

Stepwise linear discriminant analysis [3] is an extension of FLD that performs feature space reduction by selecting suitable features to be included in the discriminant function. This technique was originally introduced for classifying the P300 in [5]. A recent analysis of this method for the P300 Speller [9] has confirmed this relatively simple technique to be effective for online communication.

A combination of forward and backward stepwise analyses was implemented. Here, the input features are weighted using ordinary least-squares regression (equivalent to FLD) to predict the target class labels. Starting with no initial features in the discriminant function, the most statistically significant input feature for predicting the target label (having a p-value <0.1) is added to the discriminant function. After each new entry to the discriminant function, a backward stepwise analysis is performed to remove the least significant input features (having p-values >0.15). This process is repeated until the discriminant function includes a predetermined number of features, or until no additional features satisfy the entry/removal criteria. In this case, the final discriminant function was restricted to contain a maximum of 60 features [9].

3.4. Support vector machines

A support vector machine [17] is designed to determine the hyper-plane that maximizes the separating margin between the two classes of a binary classification. With class labels coded as $y_i \in [\pm 1]$, equation (1) can be reformulated as:

$$y_i(w \cdot f(x_i) + b) + \eta_i \geqslant 1 \tag{6}$$

where $\eta_i > 0$ represents the distance from the misclassified points to the margin. With the margin simply equaling $2/\|w\|$, the maximum margin will minimize:

$$C \sum_{i=1}^{l} \eta_i + \frac{1}{2}\|w\|^2, \tag{7}$$

subject to equation (6), where C is an arbitrary regularization parameter that reflects the penalty for misclassification and l is the number of training examples. This constrained optimization problem can be solved using Lagrangian multipliers, equivalently maximizing:

$$\max_{\alpha} \left\{ \sum_{i=1}^{l} \alpha_i - \frac{1}{2} \sum_{i=1}^{l} \sum_{j=1}^{l} \alpha_i \alpha_j y_i y_j K(x_j, x_i) \right\}$$
$$\text{s.t. } \alpha_i \geqslant 0, \quad i = 1, \ldots, l \tag{8}$$
$$\sum_{i=1}^{l} \alpha_i y_i = 0$$

where the kernel function $K(x_j, x_i) = \Phi(x_j) \cdot \Phi(x_i)$ defines the nonlinear transformation, $\Phi(x) = x$ for the linear case. With the vector of Lagrangian multipliers $\vec{\alpha}$, the classification score of a feature vector x, disregarding the inconsequential bias term, is computed as follows:

$$\text{score}_{\text{SVM}} = \sum_{i=1}^{l} \alpha_i y_i K(x_i, x). \tag{9}$$

The Gaussian kernel used for the nonlinear GSVM is selected because of its universal approximation properties and is given as follows:

$$K(u, v) = \mathrm{e}^{-\frac{\|u-v\|^2}{2\sigma^2}}. \tag{10}$$

A normalization of the input features was performed to improve performance of the SVM algorithm. The SVM parameters were varied for each user, resulting in negligible performance differences. Therefore, the parameters that resulted in the best overall performance on the training data were determined to be $C = 10$ and $\sigma^2 = 10^3$. These values were used for all simulations.

4. Comparison protocol

The previously described feature vectors served as a common input to all five of the classifiers. Parameters for each method were optimized over the set of users and fixed to the values specified in section 3. No restrictions were otherwise placed on the classification schemes and the algorithms were free to use any subset of the prescribed features for classification.

In designing a practical P300-based BCI classifier, performance and implementation are the primary considerations. These factors are outlined below.

4.1. Performance considerations

The performance of a P300 classifier is evaluated by both speed (number of intensification sequences required for accurate classification) and accuracy (per cent correct). An increased communication (bit) rate will result by optimizing one or both of these performance factors.

The performance of the classifiers was validated in two ways using the offline data. First, for each user, the classifiers were trained on the data from the first session only (all 15 intensification sequences, equivalent to 6480 training observations) and tested on all four subsequent sessions. Second, for each user, the classifiers were trained on a single session (again, all 15 intensification sequences) and tested on the subsequent session, for four consecutive sessions. For the test sessions, the feature vectors for each subsequent intensification in the sequence (up to 15) were averaged by corresponding row/columns for each character epoch and classified by the five algorithms.

4.2. Implementation considerations

The implementation of a P300 classifier is evaluated by the training requirements for the algorithm to arrive at a suitable solution and the online classification requirements of applying the resultant solution. An evaluation of the algorithm training encompasses the model selection and parameterization, the amount of training data required to construct the model, the computational complexity, and the convergence properties. These issues are confounded in characterizing the fundamental practicality of the algorithm: the amount of time and computational resources required for successful training. For instance, algorithms requiring more training data, having slower convergence properties, having increased computational complexity, or requiring multiple model/parameter/data evaluations for optimization all result in increased training time and/or more required computational resources. Because data and parameter dependences are involved, it is difficult to quantify and provide a definitive comparison of the implementation properties of the classification algorithms. The practical aspects regarding training of each algorithm are discussed in section 6.

For the five algorithms considered, online classification requirements are of less concern because all of the models are relatively simple and static. Additionally, feedback is given at the end of trial, so moderate processing delay can be tolerated. All algorithms considered merely involve simple transforms

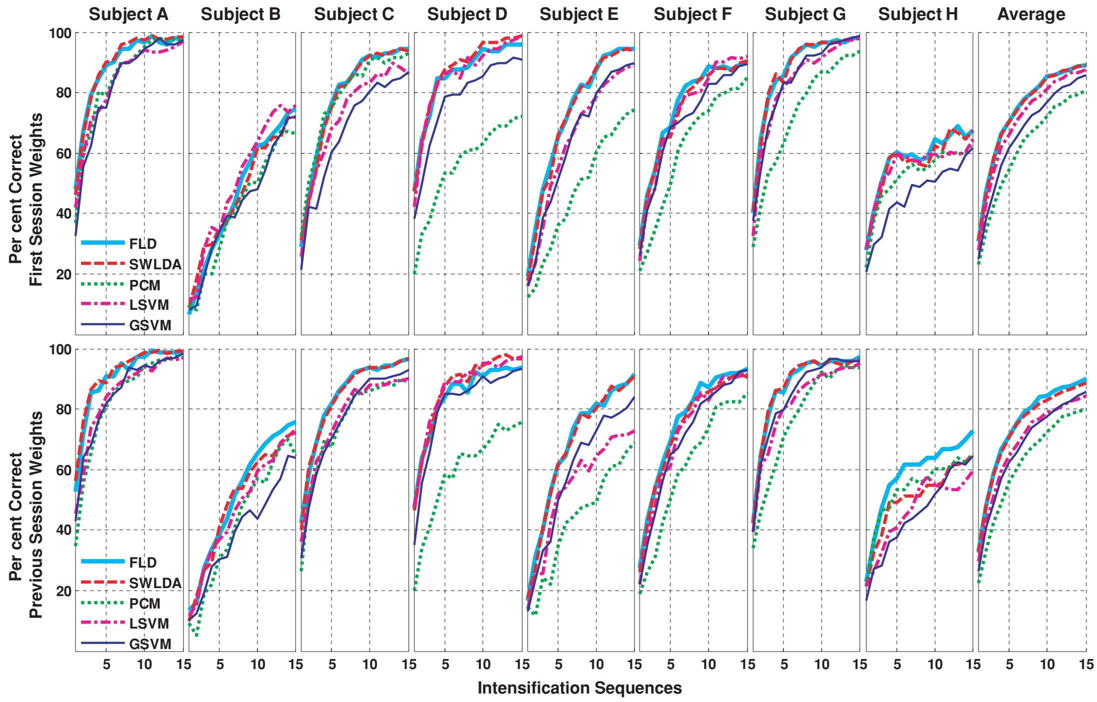

Figure 3. Performance curves for each method of training. The top row of plots shows the classification results using weights derived from the first session for each user applied to all subsequent sessions. The bottom row of plots shows the classification results using weights derived from the previous session for each user applied to each subsequent session. Legend acronyms: FLD (Fisher's linear discriminant), SWLDA (stepwise linear discriminant analysis), PCM (Pearson's correlation method), LSVM (linear support vector machine), GSVM (Gaussian support vector machine). Note: the comparatively low performance for user B is presumably the result of suboptimal feedback (see section 2.2). The comparatively low performance for user H is presumed to be due to concentration and attentional issues reported by the user during the online sessions.

(This figure is in colour only in the electronic version)

and inner products using static feature weights: PCM, FLD and LSVM apply a single weight for each input feature; SWLDA has a maximum number of weights set to the predefined model order (likely less because of the termination heuristic); and the GSVM requires the Gaussian kernel operation for each support vector, which may become a computational burden with a large number of support vectors. However, for online application of the P300 Speller, none of the methods considered will impose an impractical feedback delay.

5. Results

The performance results are provided in figure 3. The top row of plots shows the classification results using weights derived from the first session for each user applied to all subsequent sessions. The bottom row of plots shows the classification results using weights derived from the previous session for each user applied to each subsequent session.

A repeated measures ANOVA on the performance results revealed a significant difference ($F(4, 28) = 19.94, p < 0.0001$) between the five classification algorithms. Using a post hoc Tukey–Kramer [7] test, FLD and SWLDA were significantly better than PCM ($p < 0.01$) and GSVM ($p < 0.05$), and the SVM methods were also significantly better than PCM ($p < 0.05$). There is no statistically significant difference between classification using weights derived from the first session versus weights derived from the previous session ($F(1, 7) < 1$, ns), or in performance across sessions ($F(3, 21) < 1$, ns). Also, there is no statistically significant difference ($F(1, 7) < 1$, ns) between the users whose feedback was based on suboptimal generic feature weights and the users whose feedback was based on SWLDA weights optimized to their unique P300 response.

6. Discussion

With sufficient and discriminable input features, poor performance characteristics are commonly the result of inadequate modeling and/or over-fitting of the data. For this study, the input features were selected to presumably contain the essence of the P300 Speller response for discrimination [9]. In general, all of the algorithms were capable of adequately classifying the data. However, the statistical analysis suggests that linear classifiers are sufficient for P300 data and that the added complexity of nonlinear methods is not necessary. Additionally, the statistical analysis further suggests that the P300 response appears to be stable across sessions, which is consistent with [13]. Furthermore, the statistical analysis indicates that the users' offline performance was unaffected by the accuracy of the online feedback over a limited number of sessions. This invariance to feedback verifies the innate nature of the P300 response.

PCM is extremely simple: it optimizes the classifier based on the univariate statistics of a fixed model including all features. Because PCM solely relies on univariate statistics, the method is not limited by the number of training observations when the dimension of the input feature space becomes large. The training required for PCM is solely dependent on equation (4) and does not require any parameterization in this basic form. This results in rapid training, implementation and good performance. However, PCM exhibits the lowest average performance because it does not utilize the covariance between features and because it can be unnecessarily redundant.

FLD is similar in conception to PCM, but more advantageous because it accounts for the covariance between features. FLD is also extremely simple: it optimizes the classifier by optimizing the weights of all the features in a least-squares sense. As with PCM, the training required for FLD is solely dependent on equation (5) and does not require any parameterization in this basic form. This again results in rapid training, implementation and performance superior to PCM.

With increasingly large input feature spaces, the results produced by FLD and similar methods could begin to deteriorate if there is an insufficient number of training observations. SWLDA offers a solution to this problem by selectively limiting the size of the input feature space. The SWLDA algorithm is reasonably efficient because the terminating heuristic is implemented in such a way that suitable features are selected in a non-exhaustive manner. The only required parameters, the maximum model order and the termination heuristic, are intuitive and can be easily gauged based on the expected characteristics of the data. In a sense, SWLDA has the advantage of having automatic feature extraction because insignificant terms are removed from the model (i.e. weights are set to zero). Although SWLDA can be tuned to provide faster convergence by limiting the model order or termination heuristic, it is not guaranteed to be convergent and will not provide a model if the heuristic cannot be satisfied. However, this typically occurs only if the model is inadequate or if there is not discriminable information contained within the features. When properly configured, this result can be used to conclude that P300 evoked potentials are not present in the session.

SVMs are designed to have the desirable theoretical property and advantage of maximizing the margin between classes in order to provide good generalization, and thus can provide reasonable results using a minimum amount of data for training. This has been examined for P300 Speller classification in [10]. However, in practice for P300 classification, SVMs do not necessarily provide an evident performance advantage over other methods. Although the LSVM performed well, the GSVM's inferior performance is likely attributed to over-fitting the training data. Over-fitting can be a common dilemma with nonlinear classifiers because they are often able to model the training data very accurately, but can fail if the training data are not totally representative of independent test data. Over-fitting may be resolved by tuning the classification algorithm to generalize to independent test data. This leads to another drawback of SVMs, the onerous process of attaining suitable model and training parameters. Because SVM parameters such as the regularization parameter and kernel bandwidth cannot be intuitively generated, it may be necessary to examine many combinations to achieve optimal performance. In addition, although SVMs can perform well with little training data, the algorithm is very complex and training is significantly slower than with the other methods considered.

Algorithm training time and resources are of utmost practical importance. Ultimately, when P300-based BCIs are made available to disabled people, it will initially be necessary to test the efficacy of the P300 paradigm for each individual user in his or her home where the testing time may be limited to an hour or so and computing resources are commonly restricted to those of a standard laptop computer. Thus, efficient and effective algorithm training is necessary for prompt calibration, configuration of the classifier and commencement of the experiment. Nevertheless, the ultimate goal is to maximize performance and therefore communication rate. When time, data and computational resources are available, classifier performance should not be forsaken for modest improvements in the convenience of algorithm training.

Ultimately, it is conceivable that, with enough effort, any of the methods examined could likely be tuned to improve performance. However, the required effort to precisely tune each algorithm may vary greatly, which is a major consideration for practical application. Out of the five methods examined, FLD and SWLDA provide the best overall combination of training and performance characteristics for practical P300 Speller classification, with SWLDA providing potential advantages because of its capability to eliminate insignificant features for large, unknown feature spaces.

References

[1] 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

[2] Donchin E, Spencer K M and Wijesinghe R 2000 The mental prosthesis: assessing the speed of a P300-based brain-computer interface *IEEE Trans. Rehabil. Eng.* **8** 174–9

[3] Draper N and Smith H 1981 *Applied Regression Analysis* 2nd edn (New York: Wiley) pp 307–12

[4] Fabiani M, Gratton G, Karis D and Donchin E 1987 Definition, identification, and reliability of measurement of the P300 component of the event-related brain potential *Adv. Psychophysiol.* **2** 1–78

[5] Farwell L A and Donchin E 1988 Talking off the top of your head: toward a mental prosthesis utilizing event-related brain potentials *Electroenceph. Clin. Neurophysiol.* **70** 510–23

[6] Fisher R A 1936 The use of multiple measurements in taxonomic problems *Ann. Eugenics* **7** 179–88

[7] Hochberg Y and Tamhane A C 1987 *Multiple Comparison Procedures* (New York: Wiley)

[8] 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** 1073–76
[9] Krusienski D J, Sellers E W, McFarland D J, Vaughan T V and Wolpaw J R Toward enhanced P300 Speller performance *J. Neurosci. Methods* submitted
[10] Meinicke P, Kaper M, Hoppe F, Huemann M and Ritter H 2002 Improving transfer rates in brain computer interface: a case study *NIPS* 1107–14
[11] Blankertz B, Müller K R, Krusienski D J, Schalk G, Wolpaw J R, Schlögl A, Pfurtscheller G, Millán J R, Schröder M and Birbaumer N 2006 The BCI competition III: validating alternative approaches to actual BCI problems *IEEE Trans. Neural Syst. Rehabil. Eng.* **14** 153–9
[12] 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
[13] Sellers E W and Donchin E 2006 A P300-based brain-computer interface: initial tests by ALS patients *Clin. Neurophysiol.* **117** 538–48
[14] Serby H, Yom-Tov E and Inbar G F 2005 An improved P300-based brain-computer interface *IEEE Trans. Neur. Syst. Rehabil. Eng.* **13** 89–98
[15] Sharbrough F, Chatrian C E, Lesser R P, Luders H, Nuwer M and Picton T W 1991 American Electroencephalographic Society guidelines for standard electrode position nomenclature *J. Clin. Neurophysiol.* **8** 200–2
[16] Spencer K M, Dien J and Donchin E 1999 A componential analysis of the ERP elicited by novel events using a dense electrode array *Psychophysiol.* **36** 409–14
[17] Vapnik V 1998 *Statistical Learning Theory* (New York: Wiley-Interscience)
[18] 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