Comparison of Linear, Nonlinear, and Feature Selection Methods for EEG Signal Classification

Deon Garrett, David A. Peterson, Charles W. Anderson, and Michael H. Thaut

***Abstract*—The reliable operation of brain–computer interfaces (BCIs) based on spontaneous electroencephalogram (EEG) signals requires accurate classification of multichannel EEG. The design of EEG representations and classifiers for BCI are open research questions whose difficulty stems from the need to extract complex spatial and temporal patterns from noisy multidimensional time series obtained from EEG measurements. The high-dimensional and noisy nature of EEG may limit the advantage of nonlinear classification methods over linear ones. This paper reports the results of a linear (linear discriminant analysis) and two nonlinear classifiers (neural networks and support vector machines) applied to the classification of spontaneous EEG during five mental tasks, showing that nonlinear classifiers produce only slightly better classification results. An approach to feature selection based on genetic algorithms is also presented with preliminary results of application to EEG during finger movement.**

***Index Terms*—Brain–computer interface (BCI) , electroencephalogram (EEG), feature selection, genetic algorithms (GA), neural networks, pattern classification, support vector machines (SVM).**

I. Introduction

Recently, much research has been performed into alternative methods of communication between humans and computers. The standard keyboard/mouse model of computer use is not only unsuitable for many people with disabilities, but also somewhat clumsy for many tasks regardless of the capabilities of the user. Electroencephalogram (EEG) signals provide one possible means of human–computer interaction, which requires very little in terms of physical abilities. By training the computer to recognize and classify EEG signals, users could manipulate the machine by merely thinking about what they want it to do within a limited set of choices.

In this paper, we examine the application of support vector machines (SVMs) to the problem of EEG classification and compare the results to those obtained using neural networks and linear discriminant analysis. Section II provides an overview of classification methods applied here. Section III presents data acquisition procedures and classification results. Section IV summarizes the findings of this article and their implications.

II. Classification Methods

In this section, the classification methods applied in the Section III are summarized. (See Hastie *et al.* [8] for a thorough development of the classification algorithms and Whitley [23] for an introduction to genetic algorithms.)

A. Linear Discriminant Analysis

One way to classify data is to first create models of the probability density functions for data generated from each class. Then, a new data point is classified by determining the probability density function whose value is larger than the others. Linear discriminant analysis (LDA) is an example of such an algorithm. LDA assumes that each of the class probability density functions can be modeled as a normal density, and that the normal density functions for all classes have the same covariance.

Say there are K classes. Let X_k be a $p \times N_k$ matrix of N_k samples, as p-dimensional columns, of data from class k. Define the prior probabilities π_k and means μ_k of each class, and the common covariance matrix Σ, to be

$$\begin{aligned}\pi_k &= \frac{N_k}{\sum_{i=1}^{K} N_k}\\ \mu_k &= \frac{X_k 1_{N_k}}{N_k}\\ \Sigma &= \frac{\sum_{i=1}^{K}\left(X_k - \mu_k 1_{N_k}^T\right)\left(X_k - \mu_k 1_{N_k}^T\right)^T}{N-K}\end{aligned}$$

where 1_m is an $m \times 1$ matrix of 1's. Then, a new data point x is classified by

$$\arg\max_k x^T \Sigma^{-1}\mu_k - \frac{1}{2}\mu_k^T \Sigma^{-1}\mu_k.$$

The resulting LDA decision boundaries between classes of data are linear.

B. Neural Networks

Artificial neural networks are often used to develop nonlinear classification boundaries. Reasons for their common use include their ease of application, their robustness to choices of parameter values, and their similarity to other nonlinear regression methods.

Again, consider K classes. Let X be a $p \times N$ matrix of N samples, as p*-dimensional* columns. Let Y be a $K \times N$ matrix of indicator variables designating the class corresponding to each sample in X. Let α be the $p+1 \times h$ matrix of hidden-layer weights, where h is the number of hidden units, and β be the $h+1 \times K$ matrix of output-layer weights. The hidden layer output Z and the final network output O are calculated as

$$\begin{aligned}Z &= f(\alpha^T X)\\ O &= f(\beta^T Z)\end{aligned}$$

where $f(a) = 1/(1+e^{-a})$.

The error backpropagation learning algorithm is simply an iterative gradient descent procedure to minimize the squared error $(Y-O)^2$ summed over all outputs and samples. The gradient descent is performed by updating the weights as

$$\begin{aligned}\delta &= (Y-O)\cdot O \cdot (1-O)\\ \Delta\alpha &= \gamma X\left(\beta\delta \cdot Z \cdot (1-Z)\right)^T\\ \Delta\beta &= \gamma Z \delta^T\end{aligned}$$

where γ is a small constant and the $\cdot$ operator denotes component-wise multiplication. These update equations are iterated until the squared error in the network output on a subset of untrained data is minimized. The resulting weights are used to classify new data by picking the output with the largest value.

Manuscript received September 18, 2002; revised May 2, 2003. This work supported in part by National Science Foundation Award 0208958.

D. Garrett (deong@acm.org).

D. Peterson and C. Anderson are with the Department of Computer Science and the Molecular, Cellular, and Integrative Neuroscience Program, Colorado State University, Fort Collins, CO 80523 USA (e-mail: petersod@cs.colostate.edu; anderson@cs.colostate.edu).

M. Thaut is with the Department of Music, Theatre, and Dance, the Center for Biomedical Research, and the Molecular, Cellular, and Integrative Neuroscience Program, Colorado State University, Fort Collins, CO 80523 USA (e-mail: michael.thaut@colostate.edu).

Digital Object Identifier 10.1109/TNSRE.2003.814441

C. SVMs

Conventional neural networks can be difficult to build due to the need to select an appropriate number of hidden units. The network must contain enough hidden units to be able to approximate the function in question to the desired accuracy. However, if the network contains too many hidden units, it may simply memorize the training data, causing very poor generalization. A primary motivation behind SVMs is to directly deal with the objective of good generalization by simultaneously maximizing the performance of the machine while minimizing the complexity of the learned model.

The SVM optimization problem is

$$\min_{\beta,\beta_0} \frac{1}{2}\|\beta\|^2 + \gamma \sum_{i=1}^{N} \xi_i$$

$$\text{subject to } \xi_i \geq 0,\ y_i\left(h(x_i)^T\beta + \beta_0\right) \geq 1 - \xi_i.$$

This is transformed into a convex quadratic programming problem that is solved with standard techniques. The result is a discriminant function $f(x) = h(x)^T\beta + \beta_0$ which, when combined with the optimized value for β, becomes

$$f(x) = \sum_{i=1}^{N} \alpha_i y_i K(x, x_i) + \beta_0$$

where $K(x, x_i)$ is a kernel function [4] that is the same as the dot product of $h(x)$ and $h(x_i)$. This kernel trick allows $h(x)$ to be very high-dimensional, since $h(x)$ need not ever be computed.

Cover's theorem on the separability of patterns [7] essentially says that data cast nonlinearly into a high-dimensional feature space is more likely to be linearly separable there than in a lower-dimensional space. Even though the SVM still produces a linear decision function, the function is now linear in the feature space, rather than the input space. Because of the high dimensionality of the feature space, we can expect the linear decision function to perform well, in accordance with Cover's theorem. Viewed another way, because of the nonlinearity of the mapping to feature space, the SVM is capable of producing arbitrary decision functions in input space, depending on the kernel function. Mercer's theorem [6], [16] provides the theoretical basis for the determination of whether a given kernel function K is equal to a dot product in some space, the requirement for admissibility as an SVM kernel. Two examples of suitable kernel functions are the polynomial kernel $K(x_i, x_j) = (x_i^T x_j + 1)^p$ and the radial basis function (RBF) kernel $K(x_i, x_j) = \exp(-(1/2\sigma^2)\|x_i - x_j\|^2)$.

The best way to apply SVMs to the multiclass case is an ongoing research problem. The DAGSVM method, proposed by Platt *et al.*, [21], is based on the notion of decision directed acyclic graphs (DDAGs). A given DDAG is evaluated much like a binary decision tree, where each internal node implements a decision between two of the k classes of the classification problem. In the DAGSVM algorithm, each decision node uses a $1 - v - 1$ SVM to determine which class to eliminate from consideration. A separate classifier must be constructed to separate all pairs of classes.

D. Feature Selection With Genetic Algorithms

High-resolution analysis of spatial, temporal, and spectral aspects of the data, and allowing for their interactions, leads to a very high-dimensional feature space. Leveraging a higher percentage of potential features in the measured data requires more powerful signal analysis and classification capabilities. The selection of a subset of features that are most useful to the classification problem often increases classification accuracy on new data. One approach to feature selection that makes no assumptions of relationships among features involves the use of genetic algorithms (GA) to search the space of feature subsets [22], [25].

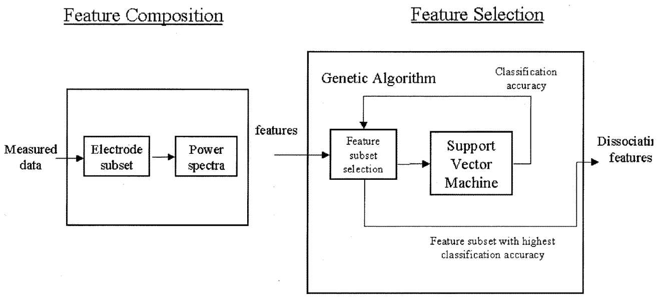

Fig. 1. System architecture for mining the EEG feature space. The space of feature subsets is searched in a "wrapper" fashion, whereby the search is directed by the performance of the classifier, in this case an SVM.

Section III summarizes results using a system consisting of feature composition, feature selection, and classification. A diagram of the system is shown in Fig. 1. The feature selection part includes an SVM for classifying the data and the genetic algorithm. SVMs involve fewer parameters than neural networks, have built-in regularization, are theoretically well-grounded, and, particularly important for ultimate real-time use in a BCI, are extremely fast.

Individuals in the population were binary strings, with 1 indicating that a feature was included, 0 indicating that it was not. We used a GA to search the space of feature subsets for two main reasons. First, exhaustive exploration of search spaces with greater than about 20 features is computationally intractable (i.e., 2^{20} possible subsets). Second, unlike gradient-based search methods, the GA is inherently designed to avoid the pitfall of local optima.

III. Results

A. Linear Versus Nonlinear Classification of Cognitive Tasks

The data used in this study were from the work of Keirn and Aunon [11], [12] and collected using the following procedure. Subjects were placed in a dim, sound controlled room and electrodes were placed at positions C3, C4, P3, P4, O1, and O2 as defined by the 10-20 system of electrode placement [9] and referenced to two electrically linked mastoids at A1 and A2. The impedance of all electrodes was kept below five Kohms. Data were recorded at a sampling rate of 250 Hz with a Lab Master 12-bit A/D converter mounted in an IBM-AT computer. Before each recording session, the system was calibrated with a known voltage. The electrodes were connected through a bank of Grass 7P511 amplifiers with analog bandpass filters from 0.1–100 Hz. Eye blinks were detected by means of a separate channel of data recorded from two electrodes placed above and below the subject's left eye. An eye blink was defined as a change in magnitude greater than 100 μV within a 10-ms period.

Subjects were asked to perform five separate mental tasks. These tasks were chosen to invoke hemispheric brainwave asymmetry. The subjects were asked to first relax as much as possible. This task represents the baseline against which other tasks are to be compared. The subjects were also asked to mentally compose a letter to a friend, compute a nontrivial multiplication problem, visualize a sequence of numbers being written on a blackboard, and rotate a three-dimensional solid. For each of these tasks, the subjects were asked to not vocalize nor gesture in any way. Data were recorded for 10 s for each task, and each task was repeated five times. The data from each channel were divided into half-second segments overlapping by one quarter-second. After segments containing eye blinks were discarded, the remaining data contained at most 39 segments. Sixth-order autoregressive (AR) models were formed for each channel independently for the data within each segment. Therefore, data in each segment were reduced

TABLE I
PERCENTAGE OF TEST DATA CORRECTLY CLASSIFIED BROKEN DOWN BY TASK. THE SVM IN THESE EXPERIMENTS USED THE SET OF PARAMETERS WHICH RESULTED IN THE HIGHEST CORRECT RATE OF CLASSIFICATION AMONG ALL SVMS TESTED

Classifier	Rest	Math	Letter	Rotate	Count	Total	Average Over 20 Windows
LDA	47.3	45.1	51.1	38.8	44.5	44.8	66.0
NN	64.3	47.3	54.7	51.1	47.3	52.8	69.4
SVM	59.4	44.5	52.7	57.0	47.9	52.3	72.0

from 750 dimensions (125 samples × 6 channels) to 36 dimensions (6 AR coefficients × 6 channels).

In testing the classification algorithms, five trials from one subject were selected from one day of experiments. Each trial consisted of the subject performing all five mental tasks. Classification experiments were performed with LDA, neural networks, and SVMs. The neural networks consisted of 36 inputs, 20 hidden units, and five output units and were trained using backpropagation with $\gamma = 0.1$. Training was halted after 2000 iterations or when the generalization began to fail, as determined by a small set of validation data chosen without replacement from the training data. SVMs were tested with polynomial kernels of degrees two, three, five, or ten, or RBF kernels with σ =0.5, 1.0, or 2.0. The SVMs were trained and tested using Platt's sequential minimal optimization (SMO) and DAGSVM algorithms [19]–[21]. The best SVM results were obtained using the RBF kernel with $\sigma = 0.5$.

The training data was selected from the full set of five trials as follows. One trial was selected as test data. Of the four remaining trials, one was chosen to be a validation set, which was used to determine when to halt training of the neural networks and which values of the kernel parameters and regularization parameter to use for the SVM tests. Finally, the remaining three trials were compiled into one set of training data. The experiments were repeated for each of the 20 ways to partition the five trials in this manner and the results of the 20 experiments were averaged to produce the results shown in Table I. This choice of training paradigm is based on earlier results [1].

Table I shows that classification of half-second windows independently results in classification accuracies of approximately 45%, 53%, and 52% for LDA, neural networks, and SVM, respectively. When the classifier outputs are averaged over 20 consecutive windows, these accuracies increase to 66%, 69%, and 72%, respectively. There is little difference in the results of the two nonlinear methods. The nonlinear methods do perform better than the linear LDA method, but only by 3%–6% when averaging over consecutive windows. Examining the results by task shows that the largest differences between linear and nonlinear methods occur for the resting and rotation tasks, suggesting that EEG for these tasks are more difficult to distinguish than for other tasks.

B. Feature Selection of Finger Movement Tasks

Blankertz *et al.* [3] collected data during a "self-paced key typing" task, which includes 413 prekey press epochs of EEG recorded from one subject. Six electrodes at F3, F4, C3, C4, CP3, and CP4 were chosen because they overlay bilateral sensorimotor cortex, presumably involved in the premotor aspects of this key pressing task. Each trial was partitioned into 11 500-ms windows shifted by 100 ms over the entire epoch, zero-meaned the signals, zero-padded them to length 1024, and computed their power spectra at 1-Hz frequency resolution. The power spectra were averaged over the standard EEG frequency bands of delta (2–4 Hz), theta (4–8 Hz), alpha (8–13 Hz), beta1 (13–20 Hz), beta2 (20–35 Hz), and gamma (35–46 Hz).

The GA was implemented with a population of 20, 2-point crossover probability of 0.66, and mutation rate of 0.008. Crossover and mutation were applied uniformly to each generation's selected individuals. Roulette-wheel selection was used, which only probabilistically chooses higher-ranked individuals. We searched over the eleven time windows and six frequency bands, while constantly including all six electrodes in each case. Thus, the dimensionality of the searchable feature space was 66 (11 × 6). The evaluation of each individual (feature subset) in the GA population consisted of training and testing the SVM with $\sigma = 0.2$ using 10 × 10 fold cross validation and averaging classification accuracy as the individual's fitness measure.

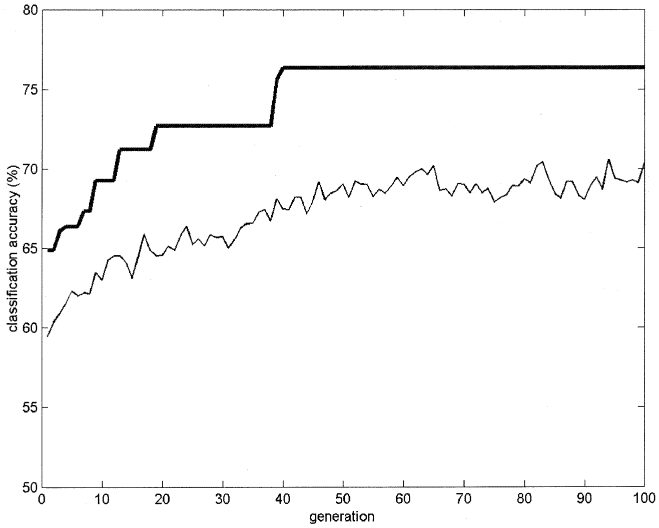

Fig. 2. Classification accuracy (population fitness) evolves over iterations (generations) of the genetic algorithm. Thin line—average fitness of the population. Thick line—fitness of the best individual in the population.

The GA evolves a population of feature subsets whose corresponding fitness (classification accuracy) improves over iterations of the GA (Fig. 2). Note that although both the population average and best individual fitness improve over successive generations, the best individual fitness does so in a monotonic fashion. The best fitness obtained was a classification accuracy of 76%. It was stable for over 50 generations of the GA. The standard deviation of the classification accuracy produced by the SVM was typically about 6%.

Fig. 3 shows the feature subset exhibiting the highest classification accuracy. The feature subset included features from every time window and every frequency band. This suggests that alternative methods that include only a few time windows or frequencies may be missing features that could improve classification accuracy. Furthermore, all frequency bands were included in the third time window, suggesting that early wide-band activity may be a significant feature of the process for deciding finger laterality.

Although the best classification accuracy (76%) was considerably higher than chance, it was much lower than the approximately 95% classification accuracy obtained by Blankertz *et al.* [3]. One possible reason is that we used data from only a small subset of the electrodes recorded (6 of 27) in order to reduce computation time by restraining the dimensionality of the feature vector presented to the SVM.

Optimizing classification accuracy was not, however, our primary goal. Instead, we sought insight into the nature of the features that would provide the best classification accuracy. The feature selection method showed that a diverse subset of spectrotemporal features in the EEG contributed to the best classification accuracy. However, most BCIs that use EEG frequency information in imagined or real movement look only at alpha (mu) and beta bands over one or a few time windows [17], [18], [24]. Furthermore, the system is amenable to online applications. One could use the full system, including the GA, to learn the best dissociating features for a given subject and task, then

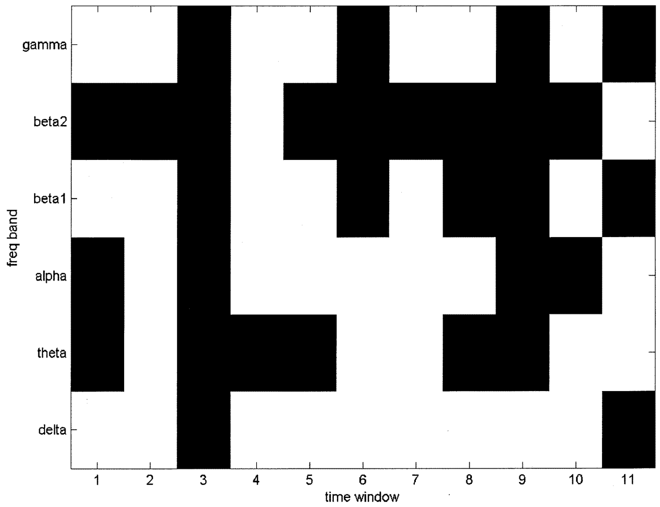

Fig. 3. Features selected for the best individual. Black indicates the feature was included in the subset, white indicates it was not. Time windows correspond to number of 100 ms shifts from epoch onset, i.e., time window 1 is early in the epoch, time window 11 ends 120 ms before the key press.

use the trained SVM with the best dissociating features in real-time. Thus, preliminary results from the research suggests that BCI performance could be improved by leveraging advances in machine learning and artificial intelligence for systematic exploration of the EEG feature space.

IV. Conclusion

SVMs provide a powerful method for data classification. The SVM algorithm has a very solid foundation in statistical learning theory, and guarantees to find the optimal decision function for a set of training data, given a set of parameters determining the operation of the SVM. The empirical evidence presented here shows that the algorithm performs well on the tested EEG classification problems, though LDA and conventional neural networks do not perform much worse.

The genetic algorithm study showed interesting changes in the subset of most significant features during a trial. These preliminary results must be explored further with additional data sets and variations in the parameters of the genetic algorithm, including the population size and the mutation and crossover rates. Population-wide measures, such as number of features chosen in common, should be examined. Comparison with standard, fixed feature selection practices, such as selection of the alpha and beta bands, should be performed. The present study describes results from only one subject. Additional, unpublished results demonstrate very different optimal feature subsets for different subjects doing the same task.

Signal transformations used here consist of either AR coefficients or power spectra. Other transformations show promise in isolating key components in signals. Independent-components analysis (ICA), which has been used to extract artifacts from EEG [10], may produce representations that increase classification accuracy. Recent results show that maximum noise fraction and cascade correlation analysis lead to accurate classification of two mental tasks [2], [13].

Acknowledgment

Implementations of SVMs by Cawley [5] and OSU [15] were used. The GAs were implemented with the commercial FlexTool GA software [14].

References

[1] C. W. Anderson, S. V. Devulapalli, and E. A. Stolz, “Determining mental state from EEG signals using neural networks,” *Sci. Program.*, vol. 4, no. 3, pp. 171–183, 1995.

[2] C. Anderson and M. Kirby, “EEG subspace representations and feature selection for brain-computer interfaces,” in *Proc. 1st IEEE Workshop on Computer Vision and Pattern Recognition for Human-Computer Interaction (CVPRHCI)*, Madison, WI, 2003.

[3] B. Blankertz, G. Curio, and K. R. Müller, “Classifying single trial EEG: toward brain computer interfacing,” in *Advances in Neural Information Processing Systems*, T. G. Dietterich, S. Becker, and Z. Ghahramani, Eds. Cambridge, MA: MIT Press, 2002, vol. 14.

[4] B. E. Boser, I. M. Guyon, and V. Vapnik, “A training algorithm for optimal margin classifiers,” in *Proc. 5th Annu. Workshop on Computational Learning Theory*, Pittsburgh, PA, 1992, pp. 144–152.

[5] G. C. Cawley, *MATLAB Support Vector Machine Toolbox*. Norwich, Norfolk, U.K.: School of Inform. Syst., Univ. East Anglia, 2000.

[6] R. Courant and D. Hilbert, *Methods of Mathematical Physics*. New York: Wiley Interscience, 1970, vol. I and II.

[7] T. M. Cover, “Geometrical and statistical properties of systems of linear inequalities with applications in pattern recognition,” *IEEE Trans. Electron. Comput.*, vol. EC-14, pp. 326–334, 1965.

[8] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Berlin, Germany: Springer-Verlag, 2001.

[9] H. Jasper, “The ten twenty electrode system of the international federation,” *Electroencephalogr. Clin. Neurophysiol.*, vol. 10, pp. 371–375, 1958.

[10] T.-P. Jung, S. Makeig, C. Humphries, T.-W. Lee, M. J. McKeown, V. Iragui, and T. J. Sejnowski, “Removing electroencephalographic artifacts by blind source separation,” *Psychophysiol.*, vol. 37, pp. 168–178, 2000.

[11] Z. A. Keirn, “Alternative modes of communication between man and machine,” M.S. thesis, Purdue Univ., West Lafayette, IN, 1988.

[12] Z. A. Keirn and J. I. Aunon, “A new mode of communication between man and his surroundings,” *IEEE Trans. Biomed. Eng.*, vol. 37, pp. 1209–1214, Dec. 1990.

[13] M. Kirby and C.W. Anderson, “Geometric analysis for the characterization of nonstationary time-series,” in *Springer Applied Mathematical Sciences Series Celebratory Volume for the Occasion of the 70th Birthday of Larry Sirovich*, E. Kaplan, J. Marsden, and K. R. Sreenivasan, Eds. Berlin, Germany: Springer-Verlag, 2003, ch. 8, pp. 263–292.

[14] CynapSys LLC, Flexga, www.cynapsys.com (2002). . [Online]

[15] J. Ma, Y. Zhao, and S. Ahalt. (2002) *OSU SVM Classifier Matlab Toolbox* [Online]http://eewww.eng.ohio-state.edu/~maj/osu_svm/

[16] J. Mercer, “Functions of positive and negative type, and their connection with the theory of integral equations,” *Trans. London Philosoph. Soc.*, vol. A 209, pp. 415–446, 1909.

[17] G. Pfurtscheller, C. Neuper, C. Guger, W. Harkam, H. Ramoser, A. Schlogl, B. Obermaier, and M. Pregenzer, “Current trends in Graz brain-computer interface (BCI) research,” *IEEE Trans. Rehab. Eng.*, vol. 8, pp. 216–219, June 2000.

[18] J. A. Pineda, B. Z. Allison, and A. Vankov, “The effects of self-movement, observation, and imagination on mu rhythms and readiness potentials: toward a brain-computer interface,” *IEEE Trans. Rehab. Eng.*, vol. 8, pp. 219–222, June 2000.

[19] J. C. Platt, “Fast training of support vector machines using sequential minimal optimization,” in *Advances in Kernel Methods—Support Vector Learning*, B. Scholkopf, C. Burges, and A. Smola, Eds. Cambridge, MA: MIT Press, 1998, pp. 185–208.

[20] ——, “Using analytic QP and sparseness to speed training of support vector machines,” in *Advances in Neural Information Processing Systems 11*, M. S. Kearns, S. A. Solla, and D. A. Cohn, Eds. Cambridge, MA: MIT Press, 1999.

[21] J. C. Platt, N.Nello Cristianini, and J. Shawe-Taylor, “Large margin DAG’s for multiclass classification,” in *Advances in Neural Information Processing Systems 12*, S. A. Solla, T. K. Leen, and K.-R. Müller, Eds. Cambridge, MA: MIT Press, 2000, pp. 547–553.

[22] D. Whitley, R. Beveridge, C. Guerra, and C. Graves, “Messy genetic algorithms for subset feature selection,” in *Proc. Int. Conf. on Genetic Algorithms*, T. Baeck, Ed.. Boston, MA, 1997, pp. 568–575.

[23] D. Whitley, “A genetic algorithm tutorial,” *Statist. Comput.*, vol. 4, pp. 65–85, 1994.

[24] J. R. Wolpaw, D. J. McFarland, and T. M. Vaughan, “Brain-computer interface research at the wadsworth center,” *IEEE Trans. Rehab. Eng.*, vol. 8, pp. 222–226, June 2000.

[25] J. Yang and V. Honavar, “Feature subset selection using a genetic algorithm,” in *Feature Extraction, Construction and Subset Selection: A Data Mining Perspective*, H. Liu and H. Motoda, Eds. Boston, MA: Kluwer Academic, 1998, pp. 117–136.