[9] G. Pfurtscheller, C. Guger, G. Muller, G. Krausz, and C. Neuper, "Brain oscillations control hand orthosis in a tetraplegic," *Neurosci. Lett.*, vol. 292, no. 3, pp. 211–4, 2000.

[10] M. Moore, J. Mankoff, E. Mynatt, and P. Kennedy, "Nudge and shove: Frequency thresholding for navigation in direct brain-computer interfaces," in *Proc. SIGCHI 2001 Conf. Human Factors in Computing Systems*, Seattle, WA, Mar. 31–Apr. 5, 2001.

[11] A. Worden, N. Walker, K. Bharat, and S. Hudson, "Making computers easier for older adults to use: Area cursors and sticky icons," in *Proc. CHI '97*, Atlanta, GA, pp. 266–271.

[12] (2002) BCI2000. Wadsworth Center. [Online]. Available: http://www.bciresearch.org/BCI2000/bci20000.html

[13] J. Mankoff, M. Moore, and U. Batra, "Web accessibility for low bandwidth input," in *Proc. ASSETS 2002*, Edinburgh, U.K., July 10–12, 2002.

[14] M. Moore and O. Tomori, "The neurally controllable web browser (BrainBrowser)," in *Proc. SIGCHI 2003*, Fort Lauderdale, FL, Apr. 2003.

Linear and Nonlinear Methods for Brain–Computer Interfaces

Klaus-Robert Müller, Charles W. Anderson, and Gary E. Birch

Abstract—At the recent Second International Meeting on Brain–Computer Interfaces (BCIs) held in June 2002 in Rensselaerville, NY, a formal debate was held on the pros and cons of linear and nonlinear methods in BCI research. Specific examples applying EEG data sets to linear and nonlinear methods are given and an overview of the various pros and cons of each approach is summarized. Overall, it was agreed that simplicity is generally best and, therefore, the use of linear methods is recommended wherever possible. It was also agreed that nonlinear methods in some applications can provide better results, particularly with complex and/or other very large data sets.

Index Terms—Feature spaces, Fisher's discriminant, linear methods, mathematical programming machines, support vector machines (SVMs).

I. INTRODUCTION

At the First International Meeting on Brain-Computer Interfaces (BCIs) held in June 1999 in Rensselaerville, NY [26], there was a significant amount of discussion around the relative advantages and disadvantages of using linear and nonlinear methods in the development of BCI systems. Therefore, at the recent Second International

K.-R. Müller is with Fraunhofer FIRST.IDA, 12489 Berlin, Germany, and also with the Department of Computer Science, the University of Potsdam, 14482 Potsdam, Germany (e-mail: klaus@first.fhg.de).

C. W. Anderson is with the Department of Computer Science, Colorado State University, Fort Collins, CO, 80523 USA (e-mail: anderson@cs.colostate.edu).

G. E. Birch is with the Neil Squire Foundation, Burnaby, BC V5M3Z3 Canada and also with the University of British Columbia, Department of Electrical and Computer Engineering, Vancouver, BC V6T1Z4 Canada.
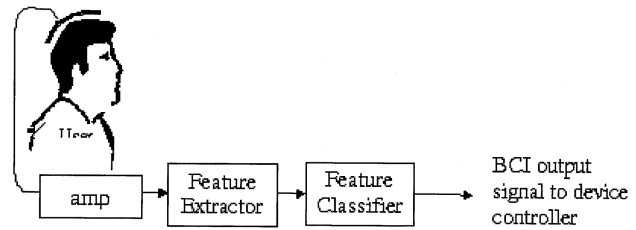
Fig. 1. Simplified functional model of a BCI System adapted from [14].

Meeting on BCIs held in June 2002 in Rensselaerville, NY, a 45-min debate was held on linear versus nonlinear methods in BCI research. The debate format involved a moderator and two discussants. K.-R. Müller from Fraunhofer FIRST.IDA, Berlin, Germany, was the first discussant and he was assigned the task of representing the point of view that linear methods should be used. The other discussant, C. W. Anderson from Colorado State University, Fort Collins, CO, was assigned the counter position that nonlinear approaches should be favored.

The Moderator, G. E. Birch from the Neil Squire Foundation, Vancouver, BC, Canada, started the debate by making a few contextual observations. In particular, the discussants were asked to clarify which aspect or component of the BCI system they were referring to when discussing the pros and cons of a particular method. For instance, in the simplified model of a BCI system given in Fig. 1, it should be clear whether a given method was to be used in the feature extractor or the feature classifier. For instance, an autoregressive (AR) modeling method might be used in the process of extracting features from the electroencephalogram (EEG) signal (for example, see [20]). On the other hand, a nearest neighbor classifier method could be applied in the feature classification process (for example, see [15]). Whichever the case, the context in which a given method is being used should be clearly understood.

In the following two sections, a summary of the discussion related to the use of linear and nonlinear methods in BCI systems is provided.

II. LINEAR METHODS FOR CLASSIFICATION

In BCI research, it is very common to use linear classifiers and this section argues in favor of them. Although linear classification already uses a very simple model, things can still go terribly wrong if the underlying assumptions do not hold, e.g. in the presence of outliers or strong noise which are situations very typically encountered in BCI data analysis. We will discuss these pitfalls and point out ways around them.

Let us first fix the notation and introduce the linear hyperplane classification model upon which we will rely mostly in the following (cf. Fig. 2, see e.g. [7]). In a BCI setup, we measure $k = 1 \ldots N$ samples $\mathbf{x}_k$, where $\mathbf{x}$ are some appropriate feature vectors in n dimensional space. In the training data, we have a class label, e.g. $y_k \in \pm 1$ for each sample point $\mathbf{x}_k$. To obtain a linear hyperplane classifier

$$\mathbf{y} = \text{sign}(\mathbf{w} \cdot \mathbf{x} + b) \tag{1}$$

we need to estimate the normal vector of the hyperplane $\mathbf{w}$ and a threshold b from the training data by some optimization technique [7]. On unseen data $\mathbf{x}$, i.e., in a BCI session, we fix the parameters $(\mathbf{w}, b)$ and compute a projection of the new data sample onto the direction of the normal $\mathbf{w} \cdot \mathbf{x}$ via (1), thus determining what class label $\mathbf{y}$ should be given to $\mathbf{x}$ according to our linear model.
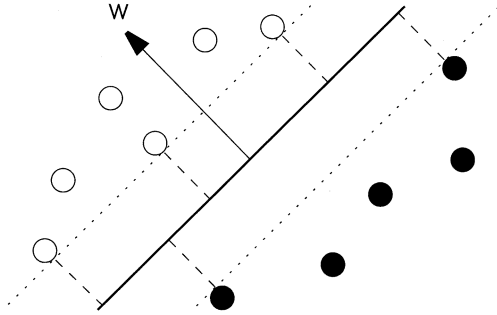
Fig. 2.   Linear classifier and margins. A linear classifier is defined by a hyperplane's normal vector $\mathbf{w}$ and an offset b, i.e. the decision boundary is $\{\mathbf{x}|(\mathbf{w} \cdot \mathbf{x}) + b = 0\}$ (thick line). Each of the two halfspaces defined by this hyperplane corresponds to one class, i.e. $f(\mathbf{x}) = \text{sign}((\mathbf{w} \cdot \mathbf{x}) + b)$. The *margin* of a linear classifier is the minimal distance of any training point to the hyperplane. In this case, it is the distance between the dotted lines and the thick line (from [18]).

A. Optimal Linear Classification: Large Margins Versus Fisher's Discriminant

Linear methods assume a linear separabilty of the data. We will see in the following that the optimal separating hyperplane from last section maximizes the minmal margin (minmax). Fisher's discriminant, that has the stronger assumption of equal Gaussian class covariances, maximizes the average margin.

1) Large Margin Classification: For linearly separable data, there is a vast number of possibilities to determine $(\mathbf{w}, b)$, that all classify correctly on the training set, however, that vary in quality on the unseen data (test set). An advantage of the simple hyperplane classifier (in canonical form cf. [25]) is that literature (see [7] and [25]) tells us how to select the optimal classifier $\mathbf{w}$ on unseen data: it is the classifier with the largest margin $\rho = 1/\|\mathbf{w}\|^2$, i.e. of minimal norm $\|\mathbf{w}\|$ [25] (see also Fig. 2).

2) Fisher's Discriminant: Fisher's discriminant computes the projection $\mathbf{w}$ and the threshold b differently and under the more restrictive assumption that the class distributions are (identically distributed) Gaussians of equal covariance, it can be shown to be Bayes optimal. The separability of the data is measured by two quantities: How far are the projected class means apart (should be large) and how big is the variance of the data in this direction (should be small). This can be achieved by maximizing the so-called Rayleigh coefficient of between and within class variance with respect to $\mathbf{w}$ [8], [9]. These slightly stronger assumptions have been fulfilled in several of our BCI experiments, e.g. in [2] and [3]: Fig. 3 clearly shows that the covariance structure is very similar for both classes such that we can safely use Fisher's discriminant.

B. Some Remarks About Regularization and Nonrobust Classifiers

Linear classifiers are generally more robust than their nonlinear counterparts, since they have only limited flexibility (less free parameters to tune) and are, thus, less prone to overfitting. Note, however, that in the presence of strong noise and outliers even linear systems can fail. In the cartoon of Fig. 4, one can clearly observe that one outlier or strong noise event can change the decision surface drastically, if the influence of single data points on learning is not limited. Although this effect can yield strongly decreased classification results for linear learning machines, it can be even more devastating for nonlinear methods. A more formal way to control one's mistrust in the available training data, is to use regularization (e.g. [4], [11], [19], [21], and [24]). Regularization helps to limit (a) the influence of outliers or strong noise [e.g., to avoid Fig. 4 (middle)], (b) the complexity of the classifier [e.g. to avoid Fig. 4 (right)], and (c) the raggedness of the
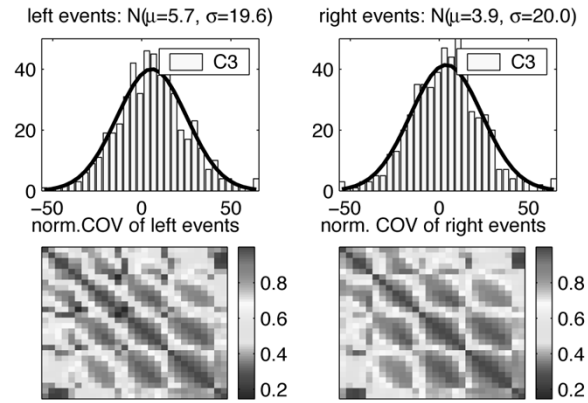

Fig. 3.   For EEG channel C3, we show in the upper panels that the projections onto the decision directions are approximately Gaussian for the "left" and the "right" class. In the lower panel, we see also that the class covariances coincide. Thus, the assumptions for using Fisher's discriminant are ideally fulfilled (from [3]).
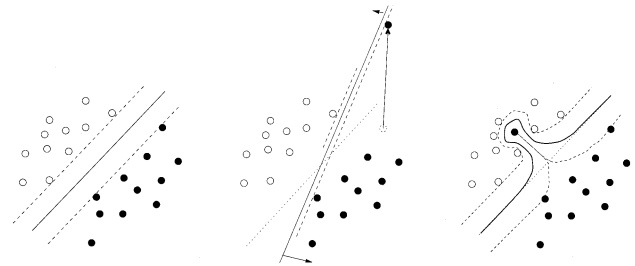

Fig. 4.   Problem of finding a maximum margin "hyper-plane" on (left) reliable data, (middle) data with an outlier and (right) with a mislabeled pattern. The solid line shows the resulting decision line, whereas the dashed line marks the margin area. In the middle and on the left, the original decision line is plotted with dots. Illustrated is the noise sensitivity: only one strong noise/outlier pattern can spoil the whole estimation of the decision line (from [22]).

decision surface [e.g. to avoid Fig. 4 (right)]. No matter whether linear or nonlinear methods are used, one should always regularize—in particular for BCI data. The regularized Fisher discriminant has been very useful in practice (cf. [2], [3], [16], and [18]). Here, $\mathbf{w}$ is found by solving the mathematical program

$$\min_{\mathbf{w},b,\xi} \frac{1}{2}\|\mathbf{w}\|^2 + \frac{C}{N}\|\xi\|^2$$

subject to

$$y_k(\mathbf{w} \cdot \mathbf{x}_k + b) = 1 - \xi_k \quad \text{for } k = 1, \dots, N$$

where ξ denote the slack variables and C is the regularization strength (a hyperparameter that needs to be determined by model selection, see e.g., [18]). Clearly, it is in general a good strategy to remove outliers first. In high dimensions (as for BCI), the latter is a very demanding if not impossible statistical mission. In some cases, however, it can be simplified by physiological prior knowledge. A further very useful step toward higher robustness is to train with robust loss functions, e.g. ℓ_1-norm or Huber-loss (e.g. [10]).

C. Beyond Linear Classifiers

Kernel based learning has taken the step from linear to nonlinear classification in a particularly interesting and efficient[1] manner: a linear

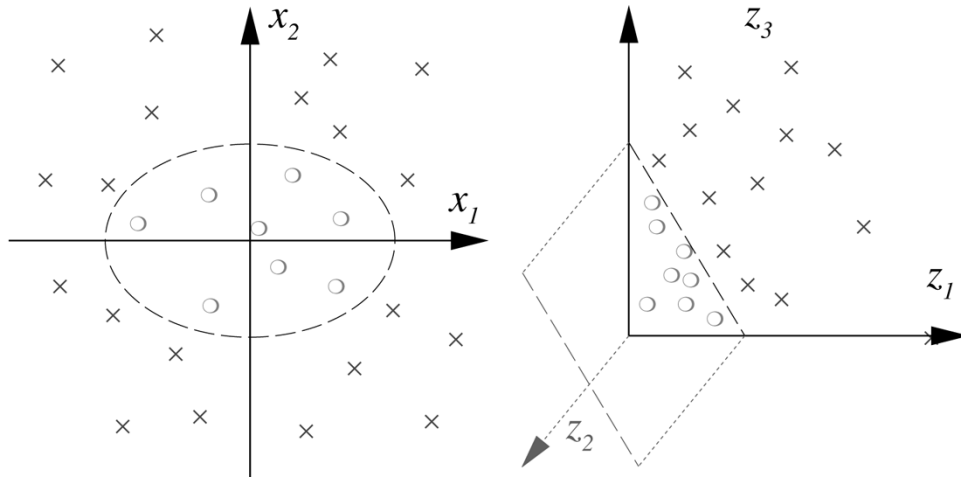[1]By virtue of the so-called "kernel trick" [25].

Fig. 5. Two dimensional classification example. Using the second-order monomials x_1^2, $\sqrt{2}x_1x_2$ and x_2^2 as features a separation in feature space can be found using (right) a linear hyperplane. In input space this construction corresponds to a (left) *nonlinear* ellipsoidal decision boundary (from [18]).

algorithm is applied in some appropriate (kernel) feature space. Thus, all beneficial properties (e.g. optimality) of linear classification are maintained,[2] but at the same time, the overall classification is non-linear in input space, since feature- and input space are nonlinearly related. This idea can be found in Fig. 5, where the classification in input space requires some complicated nonlinear (multiparameter) ellipsoid classifier. An appropriate feature space representation, in this case polynomials of second order, supply a convenient basis in which the problem can be most easily solved by a linear classifier. Examples of such kernel-based learning machines are among others, e.g. support vector machines (SVMs) [18], [25], kernel Fisher discriminant (KFD) [17] or kernel principal component analysis (KPCA) [23].

D. Discussion

In summary, a small error on unseen data cannot be obtained by simply minimizing the training error, on the contrary, this will, in general, lead to overfitting and nonrobust behavior, even for linear methods (cf. Fig. 4). One way to avoid the overfitting dilemma is to restrict the complexity of the function class, i.e. a "simple" (e.g. linear) function that explains most of the data is preferable over a complex one (Occam's razor). This still leaves the outlier problem which can only be alleviated by an outlier removal step and regularization. Note that whenever a certain linear classifier does not work well, then there are (at least) two potential reasons for this: 1) either the regularization was not done well or nonrobust estimators were used and a properly chosen linear classifier would have done well. Alternatively, it could also be that 2) the problem is intrinsically nonlinear. Then, the recommandation is to try a linear classifier in the appropriate kernel-feature space (e.g. SVMs) and regularize well.

Generally speaking, linear models are more forgiving and easy to use for inexperienced users. Furthermore, they can be substantially robustified by incorporating prior knowledge and regularization.

Finally, note that if ideal model selection can be done, then the complexity of the learning algorithm is less important. In other words, the model selection process can chose the best method, be it linear or nonlinear. In practice, k-fold cross validation is quite a useful (although not optimal) approximation to such an ideal model selection strategy.

III. NONLINEAR METHODS FOR CLASSIFICATION

It is always desirable to avoid reliance on nonlinear classification methods if possible, because they often involve a number of parame-

[2]As we do linear classification in this feature space.

ters whose values must be chosen in an informed way. If the process underlying the generation of the sampled data that is to be classified is well understood, then the user of a classification method should use this knowledge to design transformations that extract the information that is key to good classification. The extent to which this is possible determines whether or not a linear classifier will suffice. This is demonstrated in the following two sections. First, examples are discussed for which useful transformations are known. The second subsection describes how autoassociative networks can be used to learn good nonlinear transformations.

A. Fixed Nonlinear Transformations

In Section II, an example of EEG classification is shown in which the user has selected a single channel of EEG and a particular frequency band that is assumed to be very relevant to the discrimination task. With this representation, the linear classifier performed well.

A second example is described by Garrett *et al.* [27] who compare linear and nonlinear classifiers for the discrimination of EEG recorded while subjects perform one of five mental tasks. Previous work showed that a useful representation of multichannel windowed EEG signals consists of the parameters of an AR model of the data [1], [20]. One linear and two nonlinear classifiers were applied to EEG data represented as AR models. The linear method, Fisher's linear discriminant, achieved a classification accuracy on test data of 66.0%. An artificial neural network achieved 69.4% and an SVM achieved 72.0%. A purely random classification would result in 20% correct. So, the nonlinear methods do perform slightly better in this experiment, but the difference is not large. However, the computation time and memory for the neural network and the SVM are much higher than for the linear discriminant method. The neural network used 20 hidden units and the SVM resulted in an average of about 200 support vectors.

B. Learned Nonlinear Transformations

When the source of the data to be classified is not well understood, methods for finding good nonlinear transformations of the data are required. In this section, the use of autoassociative neural networks to learn such transformations is illustrated for an EEG discrimination problem.

Autoassociative neural networks are nonlinear, feedforward networks trained using the standard error backpropagation algorithm to minimize the squared error between the output and the input to the network [12], [13]. Dimensionality reduction is achieved by restricting an interior layer of the network to a number of units less than the
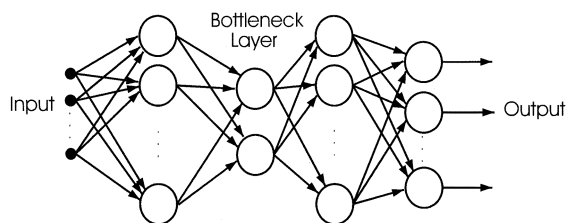
Fig. 6. Bottleneck form of autoassociative neural network for nonlinear dimensionality reduction. Two bottleneck units are shown.

number of input components, as shown in Fig. 6. This configuration is sometimes referred to as a "bottleneck" network. If the input to the network is closely approximated by the output of the network, then the information contained in the input has been compactly represented by the outputs of the bottleneck units. The nonlinear mapping from the input to the bottleneck unit outputs is formed by the two layers of units in the left half of the network.

Devulapalli [6] applied autoassociative networks to a classification problem involving spontaneous EEG. Six channels of EEG were recorded from subjects while they performed two mental tasks while minimizing voluntary muscle movement. For one task, subjects were asked to multiply two multidigit numbers. For the second task, they were asked to compose a letter to a friend and imagine writing the letter. Eye blinks were determined by a separate electrooculogram (EOG) channel and data collected during eye blinks was discarded. Data was recorded in two sessions on two different days. On each day, five trials for each task were recorded with each trial lasting for 10 s.

The resulting six time series of data for each task were divided into 0.25 s windows. The sampling rate was 250 Hz, so each window consisted of $6 \times 250/4$, or 372, values. Thus, the associative network applied to this data has 372 input and output components. The best number of hidden units, including bottleneck units, is usually determined experimentally—the usual practice is to train autoassociative networks with different numbers of bottleneck networks to determine the minimum number below which the network's input is not accurately approximated by the output. Here, the outputs of the bottleneck units are taken as a new compact representation of the windowed EEG data and classified by a second, two-layer feedforward neural network trained to output a low value for the first mental task and a high value for the second task. For this application, the classification accuracy for different numbers of bottleneck units can be used to choose the best number.

Both networks were trained on nine of the trials of each task and tested on the remaining trial. This is repeated ten times, once for each trial designated as the test data, and classification results are averaged over the ten repetitions. Fig. 7 shows the results in terms of the percent of test data correctly classified versus the number of bottleneck units. For these experiments, the number of units in the layers before and after the bottleneck layer were approximately 1.5 times the number of bottleneck units.

The best result is for 30 bottleneck units with a classification accuracy of about 85%. This is over a 12-times reduction in dimensionality, from 372 to 30. With only 10 bottleneck units, the accuracy is about 57%, not much better than the 50% level that would result from a random classification choice. Accuracy also decreases quickly as the number of bottleneck units increases. It is also known that simply training the classification network with the original representation of 372 values results in an accuracy not significantly higher than 50%.

These experiments show that the classification of untransformed EEG signals is very difficult, even with nonlinear neural networks trained to perform the classification. However, classification may be possible if the dimensionality of the EEG signals is first reduced with
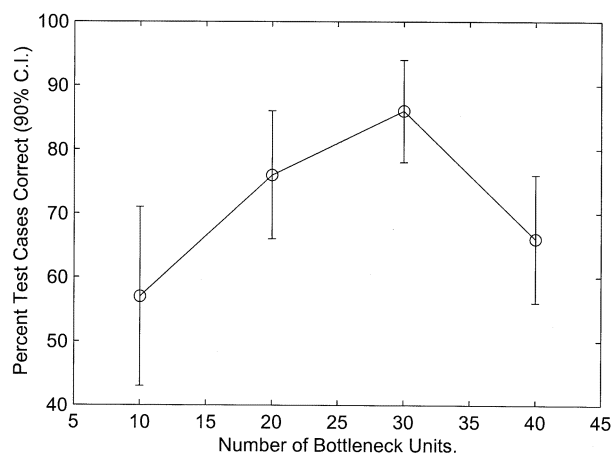

Fig. 7. Percent of test data correctly classified versus number of bottleneck units [6]. The error bars show the extent of the 90% confidence intervals.

a nonlinear transformation. Here, it is shown that an autoassociative neural network can learn this nonlinear dimensionality-reducing transformation.

Clearly, the number of bottleneck units, and, thus, the size of the reduced-dimension space, has a critical effect on the results. Ideally, we would like a method for determining the intrinsic dimension of the data. An example of automatically determining the best number of bottleneck units is the pruning algorithm demonstrated by DeMers and Cottrell [5].

IV. CONCLUSION

During the debate, most of the discussion focused on the feature classifier. The importance to understand the underlying principles and tacit assumptions of the linear and nonlinear data analysis methods was emphazised several times.

Overall, it was agreed that simplicity should be prefered. Thus, linear methods seem ideal when limited data and limited knowledge about the data is available. If there are large amounts of data, nonlinear methods are suitable to find potentially more complex structure in the data. In particular, it is suggested that when the source of the data to be classified is not well understood, to use methods that are good at finding nonlinear transformations of the data. Autoassociative neural networks or kernel methods can be used to determine these transformations.

Finally, practice shows that large gain can be achieved when incorporating e.g. neurophysiological prior knowledge into learning machines. It was furthermore stressed that regularization and/or robust methods techniques are mandatory to apply even when using linear methods. This holds even more so for nonlinear methods.

REFERENCES

[1] C. W. Anderson, E. A. Stolz, and S. Shamsunder, "Multivariate autoregressive models for classification of spontaneous electroencephalogram during mental tasks," *IEEE Trans. Biomed. Eng.*, vol. 45, pp. 277–286, Mar. 1998.

[2] 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, pp. 157–164.

[3] B. Blankertz, G. Dornhege, C. Schäfer, R. Krepki, J. Kohlmorgen, K.-R. Müller, V. Kunzmann, F. Losch, and G. Curio, "Boosting bit rates and error detection for the classification of fast-pace motor commands based on single-trial EEG analysis," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 11, pp. 127–131, June 2003.

[4] D. D. Cox and F. O'Sullivan, "Asymptotic analysis of penalized likelihood and related estimates," *Annal. Stat.*, vol. 18, no. 4, pp. 1676–1695, 1990.

[5] D. DeMers and G. Cottrell, "Non-linear dimensionality reduction," in *Advances in Neural Information Processing Systems 5*, S. J. Hanson, J. D. Cowan, and C. L. Giles, Eds.   San Mateo, CA: Kaufmann, 1992, pp. 580–587.

[6] S. Devulapalli, "Nonlinear principal component analysis and classification of EEG during mental tasks," M.S. Thesis, Department of Computer Science, Colorado State Univ., Fort Collins, CO, 1996.

[7] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern Classification*, 2nd ed.   New York: Wiley, 2001.

[8] R. A. Fisher, "The use of multiple measurements in taxonomic problems," *Annal. Eugenics*, vol. 7, pp. 179–188, 1936.

[9] K. Fukunaga, *Introduction to Statistical Pattern Recognition*, 2nd ed.   San Diego, CA: Academic, 1990.

[10] P. J. Huber, *Robust Statistics*.   New York: Wiley, 1981.

[11] G. S. Kimeldorf and G. Wahba, "Some results on Tchebycheffian spline functions," *J. Math. Anal. Applicat.*, vol. 33, pp. 82–95, 1971.

[12] M. A. Kramer, "Nonlinear principal component analysis using autoassociative neural networks," *Amer. Inst. Chem. Eng. J.*, vol. 37, no. 2, pp. 233–243, 1991.

[13] ——, "Autoassociative neural networks," *Comput. Chem. Eng.*, vol. 16, no. 4, pp. 313–328, 1992.

[14] S. G. Mason and G. E. Birch, "A general framework for brain-computer interface design," *IEEE Trans. Neural Syst. Rehab. Eng.*, vol. 11, pp. 72–87, Mar. 2003.

[15] ——, "A brain-controlled switch for asynchronous control applications," *IEEE Trans. Biomed. Eng.*, vol. 47, pp. 1297–1307, Oct. 2000.

[16] S. Mika, G. Rätsch, and K.-R. Müller, "A mathematical programming approach to the Kernel Fisher algorithm," in *Advances in Neural Information Processing Systems*, T. K. Leen, T. G. Dietterich, and V. Tresp, Eds.   Cambridge, MA: MIT Press, 2001, vol. 13, pp. 591–597.

[17] S. Mika, G. Rätsch, J. Weston, B. Schölkopf, and K.-R. Müller, "Fisher discriminant analysis with kernels," in *Neural Networks for Signal Processing IX*, Y.-H. Hu, J. Larsen, E. Wilson, and S. Douglas, Eds.   Piscataway, NJ: IEEE Press, 1999, pp. 41–48.

[18] K.-R. Müller, S. Mika, G. Rätsch, K. Tsuda, and B. Schölkopf, "An introduction to kernel-based learning algorithms," *IEEE Trans. Neural Networks*, vol. 12, pp. 181–201, Mar. 2001.

[19] G. Orr and K.-R. Müller, Eds., *Neural Networks: Tricks of the Trade*.   Berlin, Germany: Springer-Verlag, 1998, vol. 1524.

[20] G. Pfurtscheller, C. Neuper, A. Schlögl, and K. Lugger, "Separability of EEG signals recorded during right and left motor imagery using adaptive autoregressive parameters," *IEEE Trans. Rehab. Eng.*, vol. 6, pp. 316–325, Sept. 1998.

[21] T. Poggio and F. Girosi, "Regularization algorithms for learning that are equivalent to multilayer networks," *Science*, vol. 247, pp. 978–982, 1990.

[22] G. Rätsch, T. Onoda, and K.-R. Müller, "Soft margins for AdaBoost," *Machine Learning*, vol. 42, no. 3, pp. 287–320, Mar. 2001.

[23] B. Schölkopf, A. J. Smola, and K.-R. Müller, "Nonlinear component analysis as a kernel eigenvalue problem," *Neural Computat.*, vol. 10, pp. 1299–1319, 1998.

[24] A. N. Tikhonov and V. Y. Arsenin, *Solutions of Ill-posed Problems*.   Washington, DC: Winston, 1977.

[25] V. N. Vapnik, *The Nature of Statistical Learning Theory*.   New York: Springer-Verlag, 1995.

[26] J. R. Wolpaw, N. Birbaumer, W. J. Heetderks, D. J. McFarland, P. H. Peckham, G. Schalk, E. Donchin, L. A. Quatrano, C. J. Robinson, and T. M. Vaughan, "Brain-computer interface technology: a review of the first international meeting," *IEEE Trans. Rehab. Eng.*, vol. 8, pp. 164–173, June 2000.

[27] D. Garrett, D. A. Peterson, C. W. Anderson, and M. H. Thaut, "Comparison of linear, nonlinear, and feature selection methods for EEG signal classification," *IEEE Trans. Neural Syst. Rehab. Eng*, vol. 11, pp. 141–144, June 2003.

Training Locked-in Patients: A Challenge for the Use of Brain–Computer Interfaces

Nicola Neumann and Andrea Kübler

Abstract—Training severely paralyzed patients to use a brain–computer interface (BCI) for communication poses a number of issues and problems. Over the past six years, we have trained 11 patients to self-regulate their slow cortical brain potentials and to use this skill to move a cursor on a computer screen. This paper describes our experiences with this patient group including the problems of accepting and rejecting patients, communicating and interacting with patients, how training may be affected by social, familial, and institutional circumstances, and the importance of motivation and available reinforcers.

Index Terms—Biofeedback, brain-computer interfaces (BCIs), locked-in patients, man–machine communication.

I. INTRODUCTION

Brain–computer interfaces (BCIs) are devices that translate brain signals into operational commands for technical devices. While multiple methods have been developed to extract and classify the electrical activity of the brain, the application of BCIs to the target group, for example, patients with severe physical impairment and brain damage, has rarely been considered. Training patients who are diagnosed with intractable neurological diseases to use a BCI (e.g., by self-regulation of brain potentials) entails a number of problems. For the past six years, we have trained 11 severely or totally paralyzed patients to self-regulate their slow cortical potentials (SCP [1], see also Birbaumer *et al.* [24]) in order to control a computer cursor and operate a communication device [2]–[5]. Most of these patients were diagnosed with amyotrophic lateral sclerosis (ALS), a progessive neurodegenerative disease that causes widespread loss of central and peripheral motor neurons [6]. ALS generally progresses to nearly total paralysis and is fatal unless the patient is artificially ventilated and fed. In the end stage of the disease, patients may develop the "locked-in-syndrome," a state of complete motor paralysis in which sensory and cognitive functions remain intact. In the past, training these patients to control a BCI by means of self-regulation of SCPs has lasted from several months up to years and was conducted two to three days per week for 3–4 h, or, as in cases where the patient lived in distant locations, in blocks of several weeks. During training, we have encountered a number of important issues that are discussed in the following.

II. PREDICTORS OF SUCCESSFUL SELF-REGULATION AND COMMUNICATION: ACCEPTANCE AND REJECTION OF PATIENTS

Predictors of successful self-control and BCI use are important because BCIs are still in an early stage of development. To achieve the criterion level of performance in BCI use such that they can control an application (switches, spelling program, etc.), locked-in patients need