

# LINEAR AND NON-LINEAR 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 held in Rensselaerville, New York, June 2002, a formal debate was held on the pros and cons of linear and non-linear methods in Brain-Computer Interface research. Specific examples applying EEG data sets to linear and non-linear methods are given and an over view of the various pros and cons of each approach is summarised. Over all it was agreed that simplicity is generally best and therefore, the use of linear methods is recommended wherever possible. It was also agreed that non-linear methods in some applications can provide better results, particularly with complex and/or other very large data sets.

*Keywords*— Linear methods, Fisher’s discriminant, Mathematical Programming Machines, Support Vector Machines, feature spaces.

## I. INTRODUCTION

AT the First International Meeting on Brain-computer Interfaces held in Rensselaerville, New York in June 1999 [25], there was a significant amount of discussion around the relative advantages and disadvantages of using linear and non-linear methods in the development of Brain-Computer Interface systems. Therefore, at the recent Second International Meeting on Brain-Computer Interfaces held in Rensselaerville, New York, a 45-minute debate was held on linear versus non-linear methods in BCI research. The debate format involved a moderator and two discussants. Klaus-Robert Müller from Fraunhofer-FIRST, 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, Charles Anderson from Colorado State University, Colorado, USA was assigned the counter position that non-linear approaches should be favoured.

The Moderator, Gary Birch from the Neil Squire Foundation, Vancouver, Canada, started the debate by making a few contextual observations. In particular, the discussants were asked to make it clear 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 Figure 1, it should be clear if a given method was to

K.-R. Müller is with Fraunhofer FIRST, Kekuléstr. 7, 12489 Berlin, Germany, klaus@first.fhg.de and with University of Potsdam, Department of Computer Science, August-Bebelstr. 89, 14482 Potsdam, Germany. KRM was partially funded by DFG under contract JA 379/9-1, JA 379/7-1 and BMBF under contract FKZ 01IBB02A.

C. Anderson is with the Department of Computer Science, Colorado State University, Fort Collins, CO, 80523, anderson@cs.colostate.edu. CWA was partially funded by NSF Grant 9202100.

Gary Birch is with the Neil Squire Foundation, Vancouver, Canada and with the University of British Columbia, Department of Electrical and Computer Engineering, Vancouver, Canada. GEB was partially funded by NSERC Grant 90278-2002.

be used in the Feature Extractor or the Feature Classifier. For instance, an AR modeling method might be used in the process of extracting features from the EEG signal (for example see [19]). On the other hand, a Nearest Neighbour classifier method could be applied in the feature classification process (for example see [14]). Whichever the case, the context in which a given method is being used should be clearly understood.

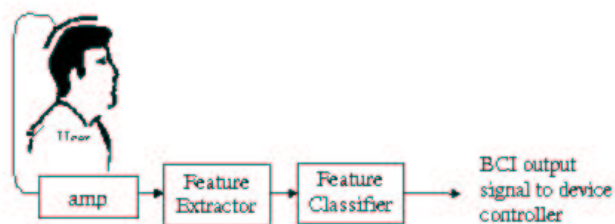


Fig. 1. Simplified functional model of a BCI System adapted from [15].

In the following two sections, a summary of the discussion related to the use of linear and non-linear 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 favour 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 set-up we measure  $k = 1 \dots N$  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

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

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 Eq.(1), thus determining what class label  $\mathbf{y}$  should be given to  $\mathbf{x}$  according to our linear model.

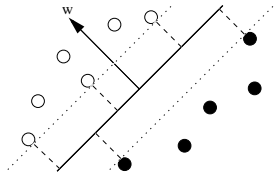


Fig. 2. Linear classifier and margins: A linear classifier is defined by a hyperplane’s normal vector  $\mathbf{w}$  and an offset  $b$ , i.e. the decision boundary is  $\{\mathbf{x} | (\mathbf{w} \cdot \mathbf{x}) + b = 0\}$  (thick line). Each of the two halfspaces defined by this hyperplane corresponds 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 separability of the data. We will see in the following that the optimal separating hyperplane from last section maximizes the *minimal* margin (minmax). Fisher’s discriminant, that has the stronger assumption of Gaussian class covariances, maximizes the *average* margin.

#### A.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. [24]) is that literature (see e.g. [7], [24]) 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  $\|\mathbf{w}\|$  [24] (see also Fig. 2).

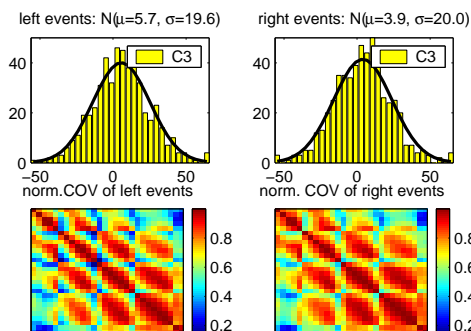


Fig. 3. For EEG channel C3, we show in the upper panels that the projections onto the decision directions are approximately Gaussian for the ‘left’ and the ‘right’ class. In the lower panel we see that also the class covariances coincide. Thus the assumptions for using Fisher’s discriminant are ideally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

#### A.2 Fisher’s discriminant

Fisher’s discriminant computes the projection  $\mathbf{w}$  and the threshold  $b$  differently and under the more restrictive as-

sumption that the class distributions are (identically distributed) Gaussians, 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], [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 non-robust 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. [11], [23], [20], [4]). 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 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! Very useful in practice has been the regular-

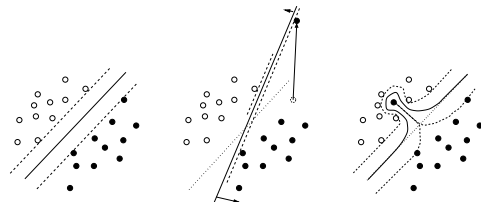


Fig. 4. The problem of finding a maximum margin ‘hyper-plane’ on reliable data (left), data with an outlier (middle) and with a mislabeled pattern (right). The solid line shows the resulting decision line, whereas the dashed line marks the margin area. In the middle and on the left the original decision 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 [21].

ized Fisher Discriminant (cf. [16], [18], [2], [3]). Here  $\mathbf{w}$  is found by solving the mathematical program

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

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

where  $\xi$  denote the slack variables and  $C$  is the regularization strength (a hyperparameter that needs to be deter-

mined by model selection, see e.g. [18]). Clearly, it is in general a good strategy to remove outliers first. In high dimension (as for BCI) the latter is a highly demanding if not impossible statistical mission. In some cases, however, it can be simplified by physiological prior knowledge. A further very useful step towards higher robustness is to train with robust loss functions, e.g.  $\ell_1$ -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<sup>1</sup> manner: a linear algorithm is applied in some appropriate (kernel) feature space. Thus, all beneficial properties (e.g. optimality) of linear classification are maintained<sup>2</sup>, but at the same time the overall classification is nonlinear in input space, since feature- and input space are nonlinearly related. A cartoon of this idea can be found in Fig.5, where the classification in input space requires some complicated non-linear (multi-parameter) 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) [24], [18], Kernel Fisher Discriminant (KFD) [17] or Kernel Principal Component Analysis (KPCA) [22].

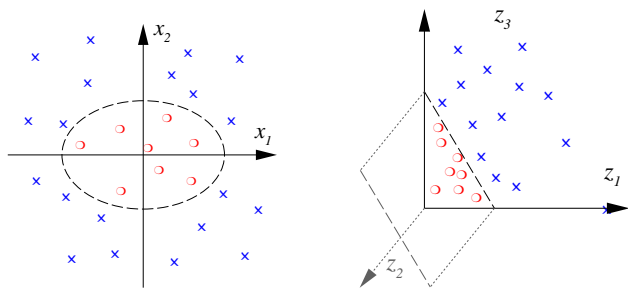


Fig. 5. Two dimensional classification example. Using the second order monomials  $x_1^2$ ,  $\sqrt{2}x_1x_2$  and  $x_2^2$  as features a separation in feature space can be found using a *linear* hyperplane (right). In input space this construction corresponds to a *non-linear* ellipsoidal decision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

### D. Discussion

To wrap up: 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 non-robust behaviour, 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 if a certain linear classifier

works lousy, then there are (at least) two potential reasons for this: (a) either the regularization was not done well or non-robust estimators were used and a *properly chosen* linear classifier would have done well. Alternatively it *could* as well be that (b) the problem is intrinsically nonlinear. Then the recommendation is to try a linear classifier in the appropriate kernel-feature space (e.g. Support Vector Machines) and regularize well.

Generally speaking, linear models are more forgiving and easy to use for ‘naive’ users, but a design where all assumptions are carefully tested whether they are fulfilled or not and prior knowledge is included, will achieve better results with high probability over a naive linear Ansatz.

Finally, note that if ideal model selection could be done then the complexity of the learning algorithm does not matter too much anymore. 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. NON-LINEAR METHODS FOR CLASSIFICATION

It is always desirable to avoid reliance on non-linear classification methods if possible, because they often involve a number of parameters 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 sections describes how autoassociative networks can be used to learn good non-linear transformations.

### A. Fixed Non-linear 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., (citation in this volume) who compare linear and non-linear 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 autoregressive (AR) model of the data [1], [19]. One linear and two non-linear 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 units achieved 69.4% and a support vector machine achieved 72.0%. A purely random classification would result in 20% correct. The non-linear methods do perform slightly better in this experiment, but the difference is not large. The compu-

<sup>1</sup>By virtue of the so-called ‘kernel trick’ [24].

<sup>2</sup>As we do linear classification in this feature space.

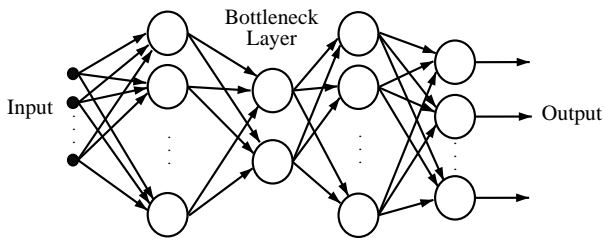


Fig. 6. Bottleneck form of autoassociative neural network for non-linear dimensionality reduction. Two bottleneck units shown.

tation time and memory for the neural network and the support vector machine are much higher than for the linear discriminant method. The neural network used 20 hidden units and the support vector machine resulted in an average of about 200 support vectors.

### B. Learned Non-linear Transformations

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

Autoassociative neural networks are non-linear, 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 number of input components, as shown in Figure 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 non-linear 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 multi-digit 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 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 seconds.

The resulting six time series of data for each task were divided into quarter-second 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 bottle-

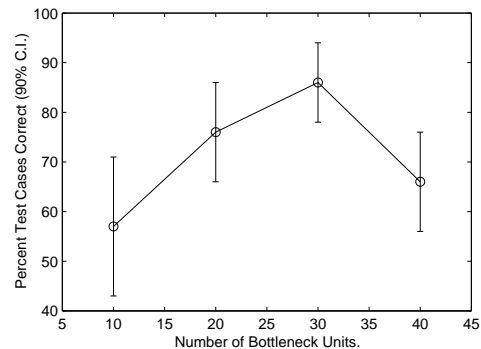


Fig. 7. Percent of test data correctly classified versus number of bottleneck units [6]. The error bars show the extent of the 90% confidence intervals.

neck 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 10 times, once for each trial designated as the test data, and classification results are averaged over the 10 repetitions. Figure 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 1/2 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 non-linear neural networks trained to perform the classification. However, classification may be possible if the dimensionality of the EEG signals is first reduced with a non-linear transformation. Here it is shown that an autoassociative neural network can learn this non-linear 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. CONCLUSIONS

During the debate, most of the discussion focused on the Feature Classifier. It was underscored several times that it is very important to understand the underlying principles of various methods and, in particular, the assumptions that are being made when applying a method in any given application. In addition, it is important to understand the characteristics, as best as possible, of the data set that will be used in a proposed system. It is also very important to use a process to regularise the data and/or use robust methods especially when applying non-linear methods.

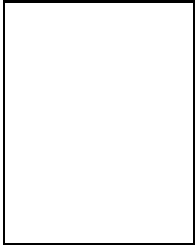
Over all, it was agreed that simplicity is generally best and therefore, use linear methods wherever possible, particularly in cases where there is limited knowledge about the data sets. In many cases when there is limited knowledge of the data sets, after some experience with applying linear methods this can lead to a better understanding of the data, perhaps preparing the way to using an appropriate non-linear method. 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 non-linear transformations of the data. Autoassociative neural networks can be used to determine these transformations. It was also agreed that non-linear methods in some applications can provide better results, particularly with complex and/or very large data sets.

**Acknowledgments:** KRM acknowledges Benjamin Blankertz, Gabriel Curio and Jens Kohlmorgen for inspiring discussions and thanks his co-authors for letting him use the figures and joint results from previous publications [18], [3], [21]. CWA acknowledges Saikumar Devulapalli for his work with auto-associative networks and Deon Garrett for his work with support vector machines.

#### REFERENCES

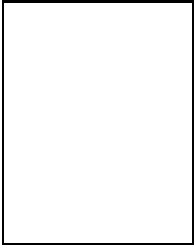
- [1] C. W. Anderson, E. A. Stolz, and S. Shamsunder. Multivariate autoregressive models for classification of spontaneous electroencephalogram during mental tasks. *IEEE Transactions on Biomedical Engineering*, 45(3):277–286, 1998.
- [2] B. Blankertz, G. Curio, and K.-R. Müller. Classifying single trial eeg: Towards brain computer interfacing. In T.G. Dietterich, S. Becker, and Z. Ghahramani, editors, *Advances in Neural Information Processing Systems*, volume 14. MIT Press, 2002.
- [3] B. Blankertz, G. Dornhege, C. Schäfer, R. Krepki, J. Kohlmorgen, K.-R. Müller, V. Kunzmann, F. Losch, and G. Curio. BCI bit rates and error detection for fast-pace motor commands based on single-trial EEG analysis. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 2002. submitted.
- [4] D.D. Cox and F. O’Sullivan. Asymptotic analysis of penalized likelihood and related estimates. *The Annals of Statistics*, 18(4):1676–1695, 1990.
- [5] David DeMers and Garrison Cottrell. Non-linear dimensionality reduction. In S. J. Hanson, J. D. Cowan, and C. L. Giles, editors, *Advances in Neural Information Processing Systems 5*, pages 580–587, San Mateo, CA, 1992. Morgan Kaufmann Publishers, Inc.
- [6] Saikumar Devulapalli. Non-linear principal component analysis and classification of EEG during mental tasks. Master’s thesis, Colorado State University, Department of Computer Science, Fort Collins, CO, 80523, 1996.
- [7] R.O. Duda, P.E.Hart, and D.G.Stork. *Pattern classification*. John Wiley & Sons, second edition, 2001.

- [8] R.A. Fisher. The use of multiple measurements in taxonomic problems. *Annals of Eugenics*, 7:179–188, 1936.
- [9] K. Fukunaga. *Introduction to Statistical Pattern Recognition*. Academic Press, San Diego, 2nd edition, 1990.
- [10] P. J. Huber. *Robust Statistics*. John Wiley and Sons, New York, 1981.
- [11] G.S. Kimeldorf and G. Wahba. Some results on Tchebycheffian spline functions. *J. Math. Anal. Applic.*, 33:82–95, 1971.
- [12] M. A. Kramer. Nonlinear principal component analysis using autoassociative neural networks. *American Institute of Chemical Engineering Journal*, 37(2):233–243, 1991.
- [13] M. A. Kramer. Autoassociative neural networks. *Computers and Chemical Engineering*, 16(4):313–328, 1992.
- [14] S. G. Mason and G. E. Birch. A brain-controlled switch for asynchronous control applications. *IEEE Transactions on Biomedical Engineering*, 47(10):1297–1307, 2000.
- [15] S. G. Mason and G. E. Birch. A general framework for describing brain-computer interface design and evaluation. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 2002.
- [16] S. Mika, G. Rätsch, and K.-R. Müller. A mathematical programming approach to the Kernel Fisher algorithm. In T.K. Leen, T.G. Dietterich, and V. Tresp, editors, *Advances in Neural Information Processing Systems*, volume 13, pages 591–597. MIT Press, 2001.
- [17] S. Mika, G. Rätsch, J. Weston, B. Schölkopf, and K.-R. Müller. Fisher discriminant analysis with kernels. In Y.-H. Hu, J. Larsen, E. Wilson, and S. Douglas, editors, *Neural Networks for Signal Processing IX*, pages 41–48. IEEE, 1999.
- [18] K.-R. Müller, S. Mika, G. Rätsch, K. Tsuda, and B. Schölkopf. An introduction to kernel-based learning algorithms. *IEEE Transactions on Neural Networks*, 12(2):181–201, 2001.
- [19] 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 Transactions on Rehabil. Engineering*, 6:316–325, 1998.
- [20] T. Poggio and F. Girosi. Regularization algorithms for learning that are equivalent to multilayer networks. *Science*, 247:978–982, 1990.
- [21] G. Rätsch, T. Onoda, and K.-R. Müller. Soft margins for AdaBoost. *Machine Learning*, 42(3):287–320, March 2001. also NeuroCOLT Technical Report NC-TR-1998-021.
- [22] B. Schölkopf, A.J. Smola, and K.-R. Müller. Nonlinear component analysis as a kernel eigenvalue problem. *Neural Computation*, 10:1299–1319, 1998.
- [23] A.N. Tikhonov and V.Y. Arsenin. *Solutions of Ill-posed Problems*. W.H. Winston, Washington, D.C., 1977.
- [24] V.N. Vapnik. *The nature of statistical learning theory*. Springer Verlag, New York, 1995.
- [25] 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 Transactions on Rehabilitation Engineering*, 8(2):164–173, June 2000.



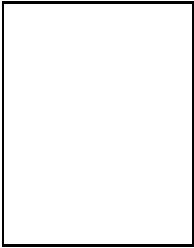
**Klaus-Robert Müller** received the Diplom degree in mathematical physics 1989 and the Ph.D. in theoretical computer science in 1992, both from University of Karlsruhe, Germany. From 1992 to 1994 he worked as a Postdoctoral fellow at GMD FIRST, in Berlin where he started to build up the intelligent data analysis (IDA) group. From 1994 to 1995 he was a European Community STP Research Fellow at University of Tokyo in Prof. Amari’s Lab. From 1995 on he is department head of the IDA group at GMD FIRST (since 2001 Fraunhofer FIRST) in Berlin and since 1999 he holds additionally a joint Professor position of GMD/FHG and University of Potsdam. He has been lecturing at Humboldt University, Technical University Berlin and University of Potsdam. In 1999 he received the annual national prize for pattern recognition (Olympus Prize) awarded by the German pattern recognition society DAGM. He serves in the editorial board of Computational Statistics, IEEE Transactions on Biomedical Engineering and in program and organization committees of various international conferences. His research areas include statistical physics and statistical

learning theory for neural networks, support vector machines and ensemble learning techniques. His present interests are expanded to time-series analysis, blind source separation techniques and to statistical denoising methods for the analysis of biomedical data.



**Charles W. Anderson** received the B.S. degree in computer science in 1978 from the University of Nebraska, and the M.S. and Ph.D. degrees in computer science in 1982 and 1986 from the University of Massachusetts, Amherst. From 1986 to 1990 he worked as a senior member of the technical staff at GTE Laboratories, Waltham, MA. Since 1991 he has been a member of the computer science faculty at Colorado State University, Fort Collins, where he is currently an associate professor.

His research areas include statistical learning algorithms for classification and control with applications to EEG signal analysis for brain-computer interfaces and reinforcement learning for robust control applications, including the control of heating and cooling in buildings.



**Gary Birch** (S'81-M'88) received the B.A. Sc. degree in electrical engineering, and the Ph.D. degree in electrical engineering (biomedical signal processing), both from the University of British Columbia, Vancouver, BC, Canada in 1983 and 1988, respectively. He was appointed Director of Research and Development at the Neil Squire Foundation in August 1988 and then in May 1994 was appointed Executive Director. He is responsible for the on-going operations at the Neil Squire Foundation including

the supervision of a Research and Development team; the preparation of contract proposals and budgets for government sponsored service delivery and research and development projects; negotiating collaborative research and development projects with private sector companies, the future direction and development of the Neil Squire Foundation and is involved in the process of transferring research and development projects ready for commercial manufacturing. His recent and current professional contributions include: Adjunct Professor at UBC, Department of Electrical Engineering since July 1989; Adjunct Professor, SFU, Gerontology Research Program since July 1990; Chair of the Minister's National Advisory Committee for Industry Canada on Assistive Devices since 1996; Member, Reference Group for the Federal Task Force on Disabilities Issues, 1996 - present; Member of the Research Advisory and Review Committee for G.F. Strong Rehabilitation Centre, 1998 - present; Member of the Executive Technical Committee on Assistive Technologies for Persons with Disabilities for the Canadian Standards Association since 1996; Member of the Premier's Advisory Council on Science and Technology, 1993 - 2000. His specific fields of expertise are robotic control systems, EEG signal processing, digital signal processing, human-machine interface systems, and biological systems.