Neural Processing Letters **10:** 243–252, 1999.

Input Feature Extraction for Multilayered Perceptrons Using Supervised Principal Component Analysis

STAVROS J. PERANTONIS and VASSILIS VIRVILIS
Institute of Informatics and Telecommunications, National Center for Scientific Research “Demokritos”, 153 10 Aghia Paraskevi, Athens, Greece, e-mail: sper@iit.demokritos.gr

Abstract. A method is proposed for constructing salient features from a set of features that are given as input to a feedforward neural network used for supervised learning. Combinations of the original features are formed that maximize the sensitivity of the network’s outputs with respect to variations of its inputs. The method exhibits some similarity to Principal Component Analysis, but also takes into account supervised character of the learning task. It is applied to classification problems leading to improved generalization ability originating from the alleviation of the curse of dimensionality problem.

Key words: feature extraction, feature selection, multilayered perceptron, principal components, saliency

1. Introduction

In recent years, extensive theoretical investigations and application dependent case studies by many researchers have helped establish the role of artificial neural networks as robust and efficient information processors. These studies have shown that artificial neural networks are useful for solving a multitude of classification, function approximation, control and optimization problems. In particular, multilayered feedforward neural networks (MFNN) have attracted much attention because of their universal approximation capabilities and successful training algorithms.

The experience with neural networks has shown that in many cases, these systems must be viewed as one component of the overall system used for the solution of a certain problem. In many cases, preprocessing and postprocessing work invariably improves performance. In the preprocessing stage, it is important to select a set of salient features spanning a space of the lowest possible dimension in order to discard features that merely constitute noise and hence alleviate the curse of dimensionality problem. This is particularly important in the case of real world data sets of high dimensionality. To this end, many feature selection or feature extraction methods have been proposed, that originate from the field of conventional statistics or neural network research. These methods concentrate either on selecting from the original set of features a smaller subset of salient features, or on combining the original features in such a way as to produce a new reduced set of salient features.

A feature selection method particularly suited for feedforward networks has been developed by Ruck [1], who defined a saliency metric that depends on the sensitivity of the trained network outputs with respect to its inputs. The feedforward network is preliminarily trained using all available features whose saliencies are subsequently determined. Only the most salient features are then used in the final training process, thus reducing feature space dimensionality. Recently, Ruck's method has been augmented by statistical techniques used to evaluate a saliency threshold in order to determine the exact number of features that should be retained. A drawback of this method is that it just selects features from the original set of available features, but does not consider further dimensionality reduction by forming salient combinations of the original features.

This drawback is not shared by the very popular and widely used feature extraction method known as principal component analysis (PCA) [2, 3]. This well known technique in multivariate statistical analysis [4] produces a potentially small number of salient linear combinations of the original features based on the maximization of the variance of the training samples. Moreover, there are many convenient and fast neural network implementations of this method that add to its attractiveness [5–9]. However, this method does not take into account class membership information available in supervised classification problems. Moreover, it is prone to failure if the data are arranged into many isotropically distributed clusters [10].

In this paper, we propose a method for feature extraction based on the determination of directions in the feature space along which the overall sensitivity of the feedforward network's output with respect to its input takes locally maximum values. Thus, we formulate an extension of Ruck's method to determine salient linear combinations of the original features. The method thus bears considerable similarity to PCA, but takes into account the supervised character of the learning task. It leads to a number of salient features whose number can be smaller than the number of salient features determined by Ruck's method, thus further alleviating the curse of dimensionality problem and leading to better generalization properties in a class of problems. The usefulness of the method and its advantages are demonstrated in some synthetic and real world supervised learning problems.

This paper is organized as follows: In Section 2 we briefly review Ruck's method and its modifications for the determination of an optimal number of salient features. In Section 3 our method is derived and its relation to PCA and Ruck's method is pointed out. In Section 4 the method is applied to a number of classification problems. Finally, Section 5 is an account of our conclusions and future prospects.

2. Feature Selection Using MFNNs

Consider an MFNN with one layer of input, M layers of hidden and one layer of output units. The units in each layer receive input from all units in the previous layer. Inputs to the first layer of the MFNN are denoted by x_i, $i = 1, \ldots, N$ where N is the total number of features the network is called upon to process. Output units are denoted by $O_i^{(m)}$, where the superscript (m) labels a layer within the structure of the neural network ($m = 1, 2, \ldots, M$ for the hidden layers, $m = M+1$ for the output layer), and i labels a unit within a layer. The synaptic weights are denoted by $w_{i_{m-1}i_m}^{(m)}$, where m, i_m denote respectively the layer and the unit toward which the synapse is directed and i_{m-1} denotes the unit in the previous layer from which the synapse emanates. Biases will be treated as weights emanating from units with constant, pattern-independent output equal to one. The logistic function $f(s) = 1/(1 + \exp(-s))$ is used as the activation function of hidden and output units.

Ruck and collaborators have proposed a method for arranging input features for training the MFNN in descending order of saliency [1]. The method amounts to "pretraining" the MFNN to learn a specific supervised learning task using all available features and computing a saliency metric S_j related to each individual feature. Pretraining may be repeated a number of times, e.g. with different initial weights or different partitions of the training set. Ruck's saliency metric for an input feature is designed to express the sensitivity of the pretrained network's output to perturbing this feature, simultaneously leaving all other features unaffected. Its formal definition is as follows:

$$S_j = \sum_{\{\boldsymbol{x}\}} \sum_i \left| \frac{\partial O_i^{(M+1)}}{\partial x_j} \right| \tag{1}$$

where the first sum denotes inclusion of information from all pretraining sessions and input patterns and the partial derivative is readily calculated using the formula:

$$\frac{\partial O_{i_{M+1}}^{(M+1)}}{\partial x_{i_0}} = \sum_{i_1, i_2, \ldots, i_M} \prod_{m=1}^{M+1} O_{i_m}^{(m)} \left(1 - O_{i_m}^{(m)}\right) w_{i_{m-1}i_m}^{(m)} \tag{2}$$

In effect, the saliency of a feature is a sum over the possible input vectors of a norm of the output vector derivative with respect to this feature. Ruck has employed the 1-norm (absolute value).

Once the saliency metrics have been evaluated, the MFNN is trained again, this time using only features with saliencies exceeding a certain saliency threshold. An interesting method for determining the most appropriate threshold is the "noise injection" method proposed by Belue and Bauer [11]. According to this technique, an additional noise feature is added during the pretraining phase as an extra MFNN input, formed using random samples from a uniform (0, 1) distribution. The MFNN is subsequently trained a number of times with different starting conditions. Assuming that the average saliency of the noise feature is normally distributed, features are declared adequately salient if their average saliency falls outside an upper one-sided confidence interval for the mean value of the saliency of the noise feature. Finally, the MFNN is retrained using only adequately salient features.

3. Supervised PCA Method

The Ruck metric employs the 1-norm of the output vector derivative with respect to input features. However, the order of the norm does not seem to be important for selecting salient features. Indeed, a similar saliency metric proposed by Tarr [12] is more closely related to the 2-norm of this derivative. For the purposes of this work, it is most convenient to adopt the 2-norm.

Considering the vector space $\mathcal{V}$ spanned by all possible feature vectors $\boldsymbol{x}$, we can speak of S_j as the saliency along the direction labeled by j. Let us now consider an arbitrary direction in $\mathcal{V}$, defined by a unit vector $\hat{\boldsymbol{u}}$. Given a vector $\boldsymbol{x}$, let us denote by $x_{\hat{u}}$ its projection along the direction $\hat{u}$, i.e. $x_{\hat{u}} = \boldsymbol{x} \cdot \hat{\boldsymbol{u}}$. Then the saliency along the direction $\hat{\boldsymbol{u}}$ is defined by:

$$S_{\hat{u}} = \sum_{\{\boldsymbol{x}\}} \sum_i \left(\frac{\partial O_i^{(M+1)}}{\partial x_{\hat{u}}} \right)^2 \tag{3}$$

We seek to find those directions $\hat{\boldsymbol{u}}$, for which the corresponding saliency $S_{\hat{u}}$ is extremal, subject to the constraint $\hat{\boldsymbol{u}} \cdot \hat{\boldsymbol{u}} = 1$. We shall show that this problem reduces to the eigenvalue problem of a real symmetric matrix, just as in the PCA formalism. Indeed, by employing the well known property of the directional derivative:

$$\frac{\partial O_i^{(M+1)}}{\partial x_{\hat{u}}} = \sum_k \hat{u}_k \frac{\partial O_i^{(M+1)}}{\partial x_k}, \tag{4}$$

we readily obtain the following expression for the saliency S_u:

$$S_{\hat{u}} = \sum_{j,k} R_{jk} \hat{u_j} \hat{u_k} \tag{5}$$

where

$$R_{jk} = \sum_{\{\boldsymbol{x}\}} \sum_i \frac{\partial O_i^{(M+1)}}{\partial x_j} \frac{\partial O_i^{(M+1)}}{\partial x_k} \tag{6}$$

is a symmetric matrix. It is now required to maximize expression (5) with respect to $\hat{u_k}$, subject to the constraint $\sum_k \hat{u_k}\hat{u_k} = 1$. On introducing a Lagrange multiplier μ to take account of the constraint, we form the expression

$$S'_{\hat{u}} = \sum_{j,k} R_{jk} \hat{u_j} \hat{u_k} + \mu(1 - \sum_k \hat{u_k}\hat{u_k}). \tag{7}$$

Constrained extrema of $S_{\hat{u}}$ occur when $\partial S'_{\hat{u}}/\partial \hat{u_j} = 0$, so that

$$\sum_k R_{jk}\hat{u_k} = \mu\hat{u_j}. \tag{8}$$

It follows that the constrained extrema occur when $\hat{\boldsymbol{u}}$ is an eigenvector of $\boldsymbol{R}$. Substituting (7) into (5) and taking account of the constraint, we readily conclude that $S_{\hat{u}} = \mu$, so that maximum saliency is equal to the maximum eigenvalue of $\boldsymbol{R}$ and is found when $\hat{\boldsymbol{u}}$ is the eigenvector of $\boldsymbol{R}$ corresponding to its maximum eigenvalue.

As a result of the above discussion, the following feature extraction method is proposed: The MFNN is "pretrained" using all available features, preferably a number of times using different initial weights. Once pretraining is completed, elements of the matrix $\boldsymbol{R}$ are computed using (6). Given a saliency threshold S_N, let there exist K eigenvalues of $\boldsymbol{R}$ larger than S_N. The eigenvectors $\hat{\boldsymbol{u}}^r$, $r = 1, \ldots K$ of $\boldsymbol{R}$ corresponding to these eigenvalues are evaluated and the K salient features extracted by our method are given by $\boldsymbol{x} \cdot \hat{\boldsymbol{u}}^r$, $r = 1, \ldots K$. Finally, the MFNN is trained using only the newly computed K salient features. Following Belue and Bauer, it is possible to evaluate S_N by including an extra noise input feature in the pretraining stage. The saliency of the extra feature for each pretraining session is evaluated using (1) (2-norm employed) and the saliency threshold S_N can be obtained, assuming that the average saliency of the noise feature is normally distributed, as the infimum of an upper one-sided confidence interval for the mean saliency of the noise feature.

4. Simulations

In order to demonstrate the efficiency of the proposed method, we use one synthetic and three real world examples:

Synthetic Example: Rotated XOR problem. Consider P two-dimensional vectors (x_1, x_2) uniformly sampled from the square defined by $-1 < x_1 < 1$ and $-1 < x_2 < 1$. In the usual XOR problem, there are two classes. Vectors whose components obey $x_1 x_2 > 0$ belong to Class 1, while vectors obeying $x_1 x_2 < 0$ belong to Class 2. We added six distractor features (x_3, x_4, x_5, x_6, x_7 and x_8), all randomly sampled between -1 and 1, and rotated each vector in the eight dimensional space defined by the x_i, $i = 1, \ldots, 8$ by an arbitrary rotation operator $\boldsymbol{A}$. The "rotated XOR problem" is defined as follows: A rotated vector $\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x}$ belongs to Class 1, if $x_1 x_2 > 0$ and to Class 2 if $x_1 x_2 < 0$. The problem is illustrated in Figure 1, where just one distractor variable x_3 is shown for visualization purposes. Note that in the rotated XOR problem all features y_i, $i = 1, \ldots, 8$ play a role in the final classification result, but only two linear combinations of these features are salient. A sample of 200 vectors was used to implement the rotated XOR problem.

Real World Examples: We give results concerning four supervised learning examples from the University of California-Irvine machine learning repository [13], namely

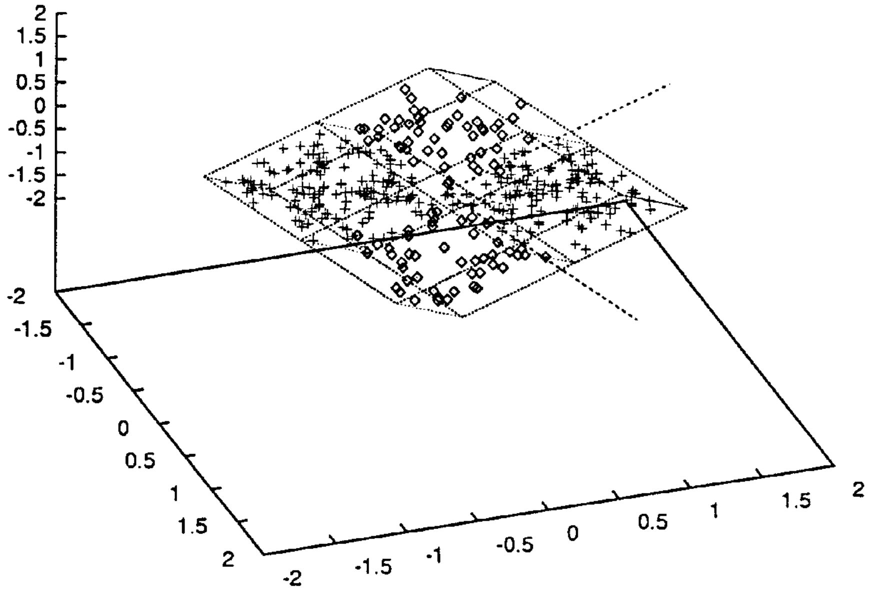

Figure 1. Rotated XOR problem with one extra noisy feature added. Sample data from the two classes are denoted by circles and crosses respectively.

1. The “Ionosphere” data set [14]. Here the task is to distinguish between two sets of radar returns from the ionosphere. This set comprises 351 patterns with 33 features for each pattern.
2. The “BUPA Liver Disorders” set. The task is to distinguish between two categories of patients with possible liver disorders on the basis of 6 attributes originating from blood test results and daily alcohol consumption figures. The set comprises 345 patterns with 6 features for each pattern.
3. The “Pima Indians Diabetes” data set [15]. It comprises 768 patterns taken from patients who may show signs of diabetes. Each sample is described by 8 attributes.
4. The “Sonar Targets” dataset [16]. The task is to distinguish between sonar returns from a metal cylinder and sonar returns from a cylindrically shaped rock. The set comprises 208 patterns with 60 features for each pattern.

For purposes of comparison, apart from the method proposed in this work, we also give results from the application of the following feature selection or feature extraction methods:

1. Ruck’s method
2. Tarr’s method
3. A method based on Student’s t-test for the difference of means of the two categories [17], whereby the significance of each original feature is assessed by finding its t-score and the most significant features are those that correspond to highest t-scores.

4. Principal Component Analysis

For all MFNN pretraining sessions, the original input features were normalized to lie in the interval between zero and one. To assess generalization ability, each dataset was partitioned into a training set consisting of 80% of the available input vectors and a test set consisting of the remaining 20% of the data. Thirty different partitions were chosen at random. Generalization ability results are given as averages over the 30 test sets. Note that for all feature extraction/selection methods, salient features were evaluated separately for each of the 30 training sets. This method was preferred to evaluating salient features using the whole dataset, because this would use information from the test sets for evaluating salient features and would lead to biased results concerning generalization ability.

All pretraining sessions (where relevant) and all final training sessions were performed using an efficient variation of the backpropagation algorithm based on the adaptive use of momentum acceleration [18]. The values $\delta P = 0.3$ and $\xi = 0.5$ were used for the gain δP and the momentum regulator ξ respectively for all problems. For all benchmarks, training was carried on for at most 400 epochs or until the mean squared error dropped below the value $2 \cdot 10^{-3}$. Networks with one hidden layer were used for all problems. For the rotated XOR problem, the hidden layer had 4 units. For all other problems 10 hidden units were used.

In order to compute saliencies, 10 pretraining sessions with different randomly chosen initial weights were performed for each of the 30 training sets. To evaluate the saliency matrix elements R_{jk}, different ways of forming the first sum of (6) were considered, including the use of random input vectors from the unit hypercube or the use of the specific input vectors of the training set. Different methods gave comparable results. Here we report results with the first summation of (6) formed using the input vectors in each training set for all 10 pretraining sessions. To determine the number of salient features, we tried using as a guide the method of Belue and Bauer. This was found to work reasonably well in conjunction with the methods of Tarr, Ruck and the t-test method. In these cases, the number of salient features determined by the method of Belue and Bauer using the mean saliency of the noise feature plus one standard deviation as a threshold was comparable to the optimal number of features for which maximum classification ability in the test set was obtained. For the proposed method, however, we found that the technique of Belue and Bauer tended to overestimate the optimal number of salient features, so that the mean value plus 4–6 standard deviations had to be used as a saliency threshold. More work is needed to explain this interesting observation.

The results of our simulations are summarized in Tables I and II. Generalization ability results are presented in Table I in the form of average percentages of successfully classified patterns in the test sets. For networks trained using all original features and for the proposed method, standard deviations are also quoted in parentheses, on the basis of which the p-value of the last column is calculated. The p-value is computed by the t-test hypothesis testing method for comparing the means of two normal distributions. It represents the probability that the statistical

Table I. Generalization ability (average classification accuracy in the test sets) achieved by various feature extraction/selection methods in five benchmark problems. For training with all original features and for the method proposed in this paper, the standard deviation of the classification accuracy in the test set is also provided (in parentheses). The p-value shown in the last column is a measure of whether there is a significant increase in the accuracy of the network trained with features selected with the proposed method over the accuracy obtained by training with all available features. Low p-values show a statistically significant increase in generalization ability.

	Proposed	Ruck	Tarr	t-test	PCA	Original features	p-value
Rotated XOR	88.32(3.40)	82.40	82.67	82.60	82.27	82.27(3.21)	$1.66 \cdot 10^{-9}$
Ionosphere	93.14(2.21)	91.37	93.28	92.71	92.27	91.68(2.63)	0.013
BUPA	71.13(5.70)	69.32	69.32	69.32	69.32	69.32(5.44)	0.110
PIMA indians	75.30(2.56)	73.57	73.57	73.57	75.22	73.57(2.23)	0.004
Sonar	79.20(3.93)	79.02	79.02	79.35	83.44	79.02(5.83)	0.445

Table II. Number of salient features selected or extracted by different methods in five benchmark problems.

	Proposed	Ruck	Tarr	t-test	PCA	Original features
Rotated XOR	3	7	7	7	8	8
Ionosphere	4	10	8	12	18	33
BUPA	1	6	6	6	6	6
PIMA indians	3	8	8	8	6	8
Sonar	18	60	60	30	20	60

means of the generalization ability distributions using all features and the features extracted with our method are the same. In Table II, the optimal number of features selected or extracted by the various methods is shown.

In all benchmarks, with the exception of the sonar data problem, the hypothesis that our method gives improved generalization ability over the method of using all original features can be accepted with adequate statistical significance. In three benchmarks (Rotated XOR, BUPA Liver Disorders, PIMA Indians) our method exhibited the best generalization ability of all methods, while in the Ionosphere benchmark it came a close second behind the method of Tarr. Naturally, best results were obtained in the synthetic rotated XOR problem, where it is known that the salient features are indeed linear combinations of a subset of the original features.

We note that the sonar data problem is linearly separable [19, 20], so that in principle only one linear combination of input features is adequate for the data to be completely separated. Indeed, the most salient feature extracted by our method had a great saliency difference from all other features. The eigenvalue of the most salient feature amounted to 85% of the sum of the eigenvalues for all features, whereas the corresponding figure for the second most salient feature was 2.5%. Even with one feature, there was no significant decrease in generalization ability (78.80% was achieved with only the most salient feature), although maximum generalization ability was achieved with 18 features (79.20%).

In all five benchmark cases our method has succeeded in extracting a relatively low number of significant features. As it is evident from Table II, this characteristic is not shared by any of the other methods, since for all other methods there were always cases where the number of selected or extracted salient features was equal to the original number of features, so that reducing the number of input features led to a decrease in generalization ability.

5. Conclusion

In this paper, a new method was proposed for the extraction of features from a set of patterns used for supervised learning purposes. Following a pretraining stage of a MFNN with the original features, linear combinations of these features are extracted, which locally maximize the response of the network's outputs to small perturbations of the inputs. The proposed method exhibits some similarity to the method of principal components analysis, but also takes into account the supervised character of the learning process. The method was applied to a number of synthetic and real world supervised learning problems and generally provided a significant increase in generalization ability with considerable reduction in the number of required input features. Results were also compared with other feature selection or feature extraction methods. Future work includes testing of the method on a larger pool of benchmarks in order to further test its consistency in producing good generalization performances. The extension of the method to other types of paradigms used for supervised learning (e.g. radial basis functions and nearest neighbor classifiers) may also lead to gains in dimensionality reduction and generalization ability.

References

1. Ruck, D. W., Rogers, S. K. and Kabrisky, M.: Feature selection using a multilayer perceptron, *Neural Network Comput.* **2** (1990), 40–48.
2. Karhunen, K.: Ueber lineare Methoden in der Wahrscheinlichkeitsrechnung, *Annales Academiae Scientiarum Fennicae, Series A1: Mathematica-Physica* **37** (1947), 3–79.
3. Loéve, M.: *Probability Theory*, 3rd ed., Van Nostrand, New York, 1963.
4. Preisendorfer, R. W.: *Principal Component Analysis in Meteorology and Oceanography*, Elsevier, New York, 1988.
5. Oja, E.: A simplified neuron model as a principal component analyser, *Journal of Mathematical Biology* **15** (1982), 267–273.
6. Sanger, T. D.: Optimal unsupervised learning in a single-layer linear feedforward neural network, *Neural Networks* **12** (1989), 459–473.

7. Földiak, P.: Adaptive network for optimal linear feature extractors, In: *Proc. International Joint Conference on Neural Networks* **1**, Washington, DC, 1989, pp. 401–405.
8. Kung, S. Y. and Diamantaras, K. I.: A neural network learning algorithm for adaptive principal component extraction (APEX), In: *Proc. International Conference on Acoustics, Speech and Signal Processing* **2**, Albuquerque, NM, 1990, pp. 861–864.
9. Chen, H. and Liu, R.-W.: Adaptive distributed orthogonalization processing for principal components analysis, In: *Proc. International Conference on Acoustics, Speech and Signal Processing* **2**, San Francisco, CA, 1992, pp. 293–296.
10. Huber, P. J.: Projection Pursuit, *Annals of Statistics* **13**, 435–475.
11. Belue, L. M. and Bauer, Jr., K. W.: Determining input features for multilayered perceptrons, *Neurocomputing* **7** (1995), 111–121.
12. Tarr, G.: Multilayered Feedforward Networks for Image Segmentation, Ph.D. Thesis, Air Force Institute of Technology, 1991.
13. Murphy, P. M. and Aha, D. W.: UCI repository of machine learning databases, Machine readable data repository, Irvine, CA, University of California, Department of Information and Computer Science, 1992.
14. Sigillito, V. G., Wing, S. P., Hutton, L. V. and Baker, K. B.: Classification of radar returns from the ionosphere using neural networks, *Johns Hopkins APL Technical Digest* **10** (1989), 262–266.
15. Smith, J. W., Everhart, J. E., Dickson, W. C., Knowler, W. C. and Johannes, R. S.: Using the ADAP learning algorithm to forecast the onset of diabetes mellitus, In: *Proc. Symposium on Computer Applications and Medical Care*, IEEE Computer Society Press, pp. 261–265, 1988.
16. Gorman, R. P. and Sejnowski, T. J.: Analysis of hidden units in a layered network trained to classify sonar target, *Neural Networks* **1** (1988), 75–89.
17. Kendall, M. and Stuart, A.: *The Advanced Theory of Statistics*, 4th ed, London: Griffin, 1977.
18. Perantonis, S. J. and Karras, D. A.: An efficient constrained learning algorithm with momentum acceleration, *Neural Networks* **8**(2) (1995), 237–239.
19. Moreno, J. M. T. and Gordon, M. B.: Characterization of the sonar signals benchmark, *Neural Processing Letters* **7**(1) (1998), 1–4.
20. Perantonis, S. J. and Virvilis, V.: Efficient linear discriminant analysis using a fast quadratic programming algorithm, In: *Proc. International Workshop on Advanced Black-Box Techniques for Nonlinear Modeling*, pp. 164–169, Leuven, Belgium, 1998.