# Linear Discriminant Analysis for Two Classes via Recursive Neural Network Reduction of the Class Separation*

Mayer Aladjem
Department of Electrical and Computer Engineering
Ben-Gurion University of the Negev, P.O.B. 653, 84105 Beer-Sheva,
Israel, *e-mail:* aladjem@bguee.bgu.ac.il

**Abstract.** A method for the linear discrimination of two classes is presented. It maximizes the Patrick-Fisher (PF) distance between the projected class-conditional densities. Since the PF distance is a highly nonlinear function, we propose a method, which searches for the directions corresponding to several large local maxima of the PF distance. Its novelty lies in a neural network transformation of the data along a found direction into data with deflated maxima of the PF distance and iteration to obtain the next direction. A simulation study indicates that the method has the potential to find the global maximum of the PF distance.

**Keywords:** Neural networks for classification, auto-associative network, projection pursuit, discriminant analysis, statistical pattern recognition.

## 1 Introduction

We discuss discriminant analysis of two classes which is carried out by a linear mapping of n-dimensional observations, which maximizes the *Patrick-Fisher (PF) distance* [6]. Unfortunately, the PF distance is a highly nonlinear function with respect to the mapping, and has more than one maximum. In most applications, the optimal solution is searched for along the gradient of the PF distance, hoping that with a good starting point the optimization procedure will converge to the global maximum or at least to a practical one. Some known techniques such as principal component analysis, Fisher discriminant analysis and their combination [1] may be used for choosing a starting point for the optimization procedure. Nevertheless, the observed maximum of the PF distance can be merely a local maximum, which is far away from the global one in some data structures. In [3] we proposed a recursive method which searches for several large local maxima of the PF distance. In this work we generalize this method using a neural network implementation, which increases its efficacy.

In Section 2 we describe a normalization of the data, called *sphering* [7] (*or whitening* [5]), which is required by our method. In Section 3 we present our method for linear discriminant analysis by recursive optimization of the PF distance [3] using the terminology of neural networks. The new proposal, called *"Neural Network Reduction of the Class Separation"* (NN_RCS) is described in Section 4. Section 5 contains results and discussions of a simulation study.

* This work has been partially supported by the Paul Ivanier Center for Robotics and Production Management, Ben Gurion University of the Negev, Israel

## 2 Sphered Data

Suppose we are given training data $(\mathbf{z}_1,\mathbf{c}_1)$, $(\mathbf{z}_2,\mathbf{c}_2)$, ... , $(\mathbf{z}_{N_t},\mathbf{c}_{N_t})$ comprising a set $Z_t=\{\mathbf{z}_1, \mathbf{z}_2, \ldots, \mathbf{z}_{N_t}\}$ of $N_t$ training observations in n-dimensional sample space ($\mathbf{z}_j \in R^n$, $n\geq 2$) and their associated class-indicator vectors $\mathbf{c}_j$, $j=1,2,\ldots,N_t$. We discuss a two class problem and we require that $\mathbf{c}_j$ is a two-dimensional vector $\mathbf{c}_j=(c_{1j}, c_{2j})^T$ which shows that $\mathbf{z}_j$ belongs to one of the classes $\omega_1$ or $\omega_2$. The components $c_{1j}$, $c_{2j}$ are defined to be one or zero according to the class-membership of $\mathbf{z}_j$, i.e. $c_{1j}=1$, $c_{2j}=0$ for $\mathbf{z}_j \in \omega_1$ and $c_{1j}=0$, $c_{2j}=1$ for $\mathbf{z}_j \in \omega_2$. The class-indicator vectors $\mathbf{c}_j$ imply decomposition of the set $Z_t$ into two subsets corresponding to the unique classes. We denote by $N_{ti}$ the number of the training observations in class $\omega_i$.

To achieve data sphering [5],[7] we perform an eigenvalue-eigenvector decomposition $\mathbf{S}_z=\mathbf{RDR}^T$ of the pooled sample covariance matrix $\mathbf{S}_z$ estimated over training set $Z_t$. Here $\mathbf{R}$ and $\mathbf{D}$ are $n\times n$ matrices; $\mathbf{R}$ is orthonormal and $\mathbf{D}$ diagonal. We then define the normalization matrix $\mathbf{A}=\mathbf{D}^{-1/2}\mathbf{R}^T$. The matrix $\mathbf{S}_z$ is assumed to be non-singular, otherwise only the eigenvectors corresponding to the non-zero eigenvalues must be used in the decomposition [7]. In the remainder of the paper, all operations are performed on the *sphered training data* $X_t=\{\mathbf{x}_j: \mathbf{x}_j=\mathbf{A}(\mathbf{z}_j-\mathbf{m}_z), \mathbf{z}_j \in Z_t\}$ with $\mathbf{m}_z$ the sample mean vector estimated over $Z_t$. For the sphered training data $X_t$ the pooled sample covariance matrix becomes the identity matrix $\mathbf{AS}_z\mathbf{A}^T=\mathbf{I}$.

## 3 Training SL Network for Classification by Recursive Reduction of the Class Separation (RCS)

Here we present our method for linear discriminant analysis by recursive *reduction of the class separation* (RCS) [3] using the terminology of *single-layer* (SL) neural networks for classification [5]. We discuss an SL network with linear activation function of the output. It carries out a linear mapping $y=\mathbf{w}^T\mathbf{x}$, $\mathbf{x}\in R^n$, $y\in R^1$, $n\geq 2$, with $\mathbf{x}$ an arbitrary n-dimensional observation, and $\mathbf{w}$ a vector containing the weights of the network (Fig.1). We require $\mathbf{w}$ to have unit length, and $y=\mathbf{w}^T\mathbf{x}$ can be interpreted geometrically [5, pp.77-79] as the projection of the observation $\mathbf{x}$ onto vector $\mathbf{w}$ in $\mathbf{x}$-space (Fig.2).

We train the network by maximizing the *Patrick-Fisher (PF) distance* [6]

$$PF(\mathbf{w}) = \Big\{ \int_{R^n} \Big[ \frac{N_{t1}}{N_t} \hat{p}(\mathbf{w}^T\mathbf{x}|\omega_1) - \frac{N_{t2}}{N_t} \hat{p}(\mathbf{w}^T\mathbf{x}|\omega_2) \Big]^2 d\mathbf{x} \Big\}^{1/2} \qquad (1)$$

with

$$\hat{p}(\mathbf{w}^T\mathbf{x}|\omega_i) = \frac{1}{h\sqrt{2\pi}N_{ti}} \sum_{j=1}^{N_t} c_{ij} \exp\Big\{ \frac{-1}{2h^2} [\mathbf{w}^T(\mathbf{x}-\mathbf{x}_j)]^2 \Big\} \quad , i=1,2 \qquad (2)$$

the Parzen estimators with Gaussian kernels of the class-conditional densities of the projections $y=\mathbf{w}^T\mathbf{x}$. Here $\mathbf{x}$ is an arbitrary observation ($\mathbf{x}\in R^n$), $c_{ij}$ is the class-indicator

which constrains the summation in (2) on the $\omega_i$-training observations ($\mathbf{x}_j$ corresponding to $c_{ij}=1$), and h is a smoothing parameter.

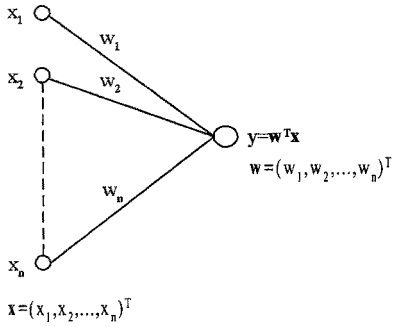


**Fig.1.** Representation of a linear mapping as a neural network diagram.

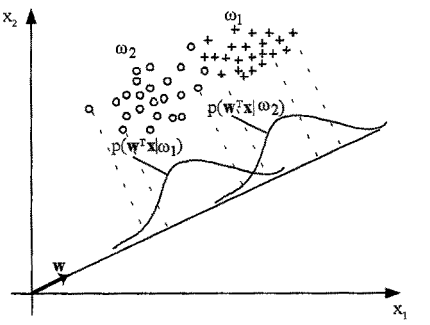


**Fig.2.** Linear mapping ($y=\mathbf{w}^T\mathbf{x}$) in a two-dimensional x-space. Class-conditional densities $p(\mathbf{w}^T\mathbf{x}|\omega_1)$ and $p(\mathbf{w}^T\mathbf{x}|\omega_2)$ along the vector **w**.

PF(**w**) (1) measures the separation of the class-conditional densities along vector **w** (see Fig.2). The theoretical motivation of PF(**w**) is its resultant upper bound on the Bayes error along **w**. It is known that PF(**w**) induces an upper bound which is larger than those of other probabilistic class separability measures [9]. Nevertheless, PF(**w**) is more practical, because of the existence of an analytical expression of its gradient used in the training [6, pp.277-280].

PF(**w**) is a nonlinear function with respect to **w**. We train the SL network using a local optimizer for maximizing PF(**w**). We choose the starting point of the optimizer by means of an extended Fisher discriminant analysis [1,2] which has proved to be suitable in the practical applications. Nevertheless the observed maximum of PF(**w**) can be merely a local maximum in some data structures.

In order to search for several large local maxima of PF(**w**) we have proposed a method for recursive maximization of PF(**w**) [3]. We obtain a vector of weights **w*** related to a local maximum of PF(**w***) and then we transform the data along **w*** into data with greater overlap of the class-conditional densities (deflated maximum of PF(**w**) at the solution **w***), and iterate to obtain a new vector of weights.

The main point of the method is the procedure for deflating the local maximum of PF(**w**) called *"reduction of the class separation"* (RCS). In order to deflate PF(**w**) at **w*** (to increase class overlap along **w***), we transform class-conditional densities along **w*** to normal densities. For this purpose, we rotate the data applying the linear transformation

$$\mathbf{r}=\mathbf{U}\mathbf{x} \tag{3}$$

with **U** an orthonormal (n×n) matrix. We denote the new coordinates as $r_1, r_2, \ldots, r_n$ ($\mathbf{r}=(r_1, r_2, \ldots, r_n)^T$). We require that the first row of **U** is **w***, which results in a rotation such that the new first coordinate of an observation **x** is the output of the SL

network having weight vector $\mathbf{w}^*$ ( $r_1=y=(\mathbf{w}^*)^T\mathbf{x}$ ). Assume that $p(y|\omega_i)$, i=1,2 are the class-conditional densities of $y=(\mathbf{w}^*)^T\mathbf{x}$ and, $m_{y|\omega_i}$, $\sigma^2_{y|\omega_i}$ their means and variances. We transform $p(y|\omega_i)$ to normal densities and leave the coordinates $r_2, r_3, \ldots, r_n$ unchanged. Let $\mathbf{q}$ be a vector function with components $q_1,q_2,\ldots,q_n$ that carries out this transformation: $r_1'=q_1(y)$ with $r_1'$ having normal class-conditional distributions and $r_i'=q_i(r_i)$, i=2,3,...,n each given by the identity transformations. The function $q_1$ is obtained by the percentile transformation method [2,3,7]:

- for observations $\mathbf{x}$ from class $\omega_1$:

$$q_1(y) = [\Phi^{-1}(F(y|\omega_1))](\sigma^2_{y|\omega_1} \pm \Delta\sigma^2)^{1/2} + (m_{y|\omega_1} - \Delta m_1); \quad (4)$$

- for observations $\mathbf{x}$ from class $\omega_2$:

$$q_1(y) = [\Phi^{-1}(F(y|\omega_2))](\sigma^2_{y|\omega_2} \pm \Delta\sigma^2)^{1/2} + (m_{y|\omega_2} - \Delta m_2). \quad (5)$$

Here, $\Delta\sigma^2$ ($0 \le \Delta\sigma^2 \le 1$), $\Delta m_1$, $\Delta m_2$ are user-supplied parameters, $F(y|\omega_i)$ is the class-conditional (cumulative) distribution function of $y=(\mathbf{w}^*)^T\mathbf{x}$ for i=1,2 and $\Phi^{-1}$ is the inverse of the standard normal distribution function $\Phi$. Finally,

$$\mathbf{x}' = \mathbf{U}^T\mathbf{q}(\mathbf{U}\mathbf{x}) \quad (6)$$

transforms the class-conditional densities of the output of the SL network to give normal densities

$$p(r_1'|\omega_i)=N(m_{y|\omega_i} - \Delta m_i,\ \sigma^2_{y|\omega_i} \pm \Delta\sigma^2) \quad (7)$$

leaving all directions orthogonal to $\mathbf{w}^*$ unchanged.

In [3] we proposed a procedure for defining the values of the control parameters $\Delta\sigma^2$, $\Delta m_1$, $\Delta m_2$ and the sign (+ or -) of the change $\pm\Delta\sigma^2$ in order to direct the local optimizer to a new maximum of PF($\mathbf{w}$), and to keep the class-conditional densities of $\mathbf{x}'$ (6) as close to the densities of the original data $\mathbf{x}$ as is possible.

We presented the method in its abstract version based on probability distributions. The application to observed data is accomplished by substituting an estimate of the distributions over the training set $X_t$ [3,7].

## 4 Neural Network Reduction of the Class Separation (NN_RCS)

Here we propose a neural network implementation of the procedure for "reduction of the class separation", called NN_RCS. We use an auto-associative multi-layer network having non-linear activation functions in the hidden units (Fig.3). The targets used to train the network are the input vectors themselves, so that the network is attempting to map each input vector onto itself.

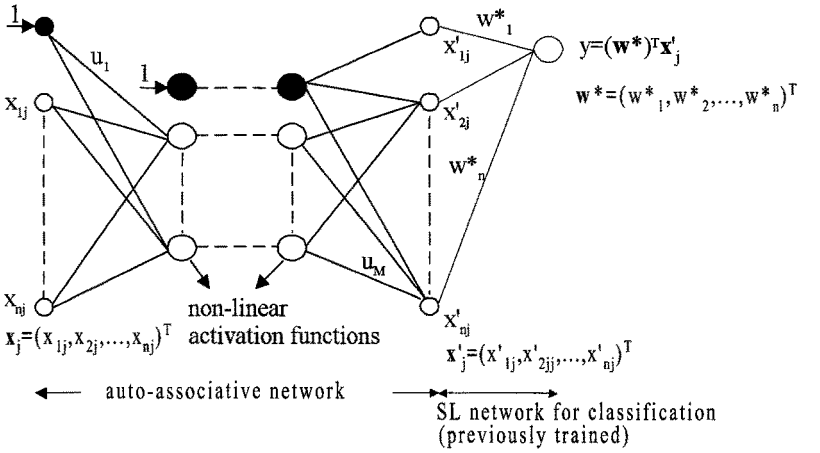


**Fig.3.** Auto-associative network. It is trained to map input vectors into themselves in such a way that the data along **w*** have normal class-conditional densities.

We train the network by minimizing an error function of the form

$$E(\mathbf{u}) = (1-\nu)E^{AA}(\mathbf{u}) + \nu\Omega(\mathbf{u}), \quad 0\le\nu\le 1, \tag{8}$$

with

$$E^{AA}(\mathbf{u}) = \frac{1}{N_t}\sum_{j=1}^{N_t}[\mathbf{r}(\mathbf{x}_j;\mathbf{u}) - \mathbf{x}_j]^T[\mathbf{r}(\mathbf{x}_j;\mathbf{u}) - \mathbf{x}_j], \tag{9}$$

$$\Omega(\mathbf{u}) = \frac{1}{N_t}\sum_{j=1}^{N_t}\left[(\mathbf{w}^*)^T\mathbf{r}(\mathbf{x}_j;\mathbf{u}) - q_1\left((\mathbf{w}^*)^T\mathbf{x}_j\right)\right]^2. \tag{10}$$

Here, $E^{AA}(\mathbf{u})$ is the standard mean-square error of the auto-associative network, $\Omega(\mathbf{u})$ is the penalty function and $\nu$ is the parameter controlling the extent to which the penalty term $\Omega(\mathbf{u})$ influences the form of the solution. In (9) and (10) $\mathbf{r}(\mathbf{x}_j;\mathbf{u})$ represents the output vector $\mathbf{x}_j'=\mathbf{r}(\mathbf{x}_j;\mathbf{u})$ of the auto-associative network (Fig.3) as a function of the input training vectors $\mathbf{x}_j$, j=1, 2, ...,$N_t$ and vector **u** comprising the adjustable weights of the network; $q_1((\mathbf{w}^*)^T\mathbf{x}_j)$ is the function which transforms the class-conditional densities of $y=(\mathbf{w}^*)^T\mathbf{x}_j$ to the normal densities (see (4) and (5)).

The auto-associative network is trained by minimizing the total error function E(**u**) (8) with respect to **u**. A function $\mathbf{r}(\mathbf{x}_j;\mathbf{u})$ which provides a good fit to the training data $\mathbf{x}_j$ , j=1,2, ..., $N_t$ will give a small value for $E^{AA}(\mathbf{u})$ (9), while one which produces data with the normal densities along **w*** will give a small value for $\Omega(\mathbf{u})$ (10). Minimizing E(**u**) (8) we obtain the network mapping $\mathbf{r}(\mathbf{x}_j;\mathbf{u})$ which is a compromise between fitting the training data $\mathbf{x}_j$ and reducing the class separation along **w*** (deflating PF(**w**) (1) at **w*** for suitable $\nu$, $\Delta\sigma^2$, $\Delta m_1$ and $\Delta m_2$). We can view this network as a neural network implementation of the data transformation (6). Here the auto-associative network (Fig.3) performs principal component analysis [5,p.314]

constrained on the class-conditional densities of $\mathbf{x}_j$ along $\mathbf{w}^*$ to be normal densities. Since the penalty term $\Omega^{SL}(\mathbf{u})$ (10) is a highly non-linear function (see Exprs. (4) and (5)) we use nonlinear activation functions in the hidden units of the network despite the fact that the standard network for principal component analysis has linear activation functions in the hidden units [5, p.314].

The computational complexity of the NN_RCS is higher than that of our method proposed in [2,3]. Here we have to use an intensive non-linear optimization technique for training the auto-associative network, and also the values of the parameters $\nu$, $\Delta\sigma^2$, $\Delta m_1$, $\Delta m_2$ must be specified in advance of training the network, so that it is necessary to train and compare several auto-associative networks for different values of $\nu$, $\Delta\sigma^2$, $\Delta m_1$, $\Delta m_2$. This high computational complexity is the price that we pay in order to gain the following advantages:

*1. NN_RCS improves the preservation of the training data:* Our method for reduction of the class separation (RCS), explained in Section 3, exactly preserves the data in the subspace orthogonal to the vector $\mathbf{w}^*$. In order to preserve the data as much as possible in the entire space, we have proposed in [3] a procedure which searches for the smallest values of the parameters $\Delta\sigma^2$, $\Delta m_1$ and $\Delta m_2$, which direct the local optimizer to a new maximum of PF($\mathbf{w}$). Nevertheless, in some applications [3] our procedure causes large changes of the class-conditional distributions of the transformed data $\mathbf{x}'$ (6), which is undesirable. The NN_RCS by performing highly non-linear data transformation increases the range of data preservation, which is demonstrated by the experiments explained in the next Section 5.

*2. NN_RCS can be applied for reduction of the class separation of the non-linear classification functions:* Actually, by using the NN_RCS, we overcome the use of the orthonormal matrix $\mathbf{U}$ in the transformation (3). This makes it possible to apply NN_RCS for the non-linear classification functions $y(\mathbf{x})$. We have just to obtain function $q_1(y)$ which transforms the class-conditional densities of $y(\mathbf{x})$ to normal densities and to use the non-linear mapping $y(\mathbf{x}_j')$ instead of the linear one $(\mathbf{w}^*)^T\mathbf{x}_j'$, for $\mathbf{x}_j'=\mathbf{r}(\mathbf{x}_j; \mathbf{u})$ in the penalty function (10). In [4], using NN_RCS, we have proposed a method for recursive training of a *multi-layer* (ML) neural network by reduction of the class separation for the non-linear classification functions obtained by the ML network.

## 5 Simulation Studies

Here we compare the preservation of the data after deflating a local maximum of the PF distance by RCS (Section 3) and by our new proposal NN_RCS (Section 4). We ran experiments with samples for two classes of the sample sizes $N_{t1}=N_{t2}=150$, which were drawn from two-dimensional normal mixtures:

for class $\omega_1$ :

$$p(x_1,x_2\,|\omega_1)=1/3N([-1\ 0]^T, 0.1\mathbf{I})+1/3N([0.5\ 3]^T, 0.1\mathbf{I})+1/3N([-0.5\ -3]^T, 0.1\mathbf{I}), \quad (11)$$

for class $\omega_2$:

$$p(x_1,x_2|\omega_2) = 1/3N([-0.5\ 3]^T, 0.1\mathbf{I})+1/3N([3\ 0]^T, 0.1\mathbf{I})+1/3N([0.5\ -3]^T, 0.1\mathbf{I}) . \quad (12)$$

Here, $N([\mu_1\ \mu_2]^T,\mathbf{I})$ denotes bivariate normal density with a mean vector $[\mu_1\ \mu_2]^T$ and a unit covariance matrix. Fig.4 presents the sphered data (see Section 2). For this data we computed the PF distances for 91 equally angled directions into the $(x_1,x_2)$-plane. The solid path "——" in Fig.5 presents the PF distances for the vectors $\mathbf{w}$ directed under different angles with respect to $x_1$-axis. We observe local maxima of PF($\mathbf{w}$) at angles 15°, 49°, 64°, 109°, 124°, 135° and 162°. The global maximum (PF distance 0.7838) is at 15°.

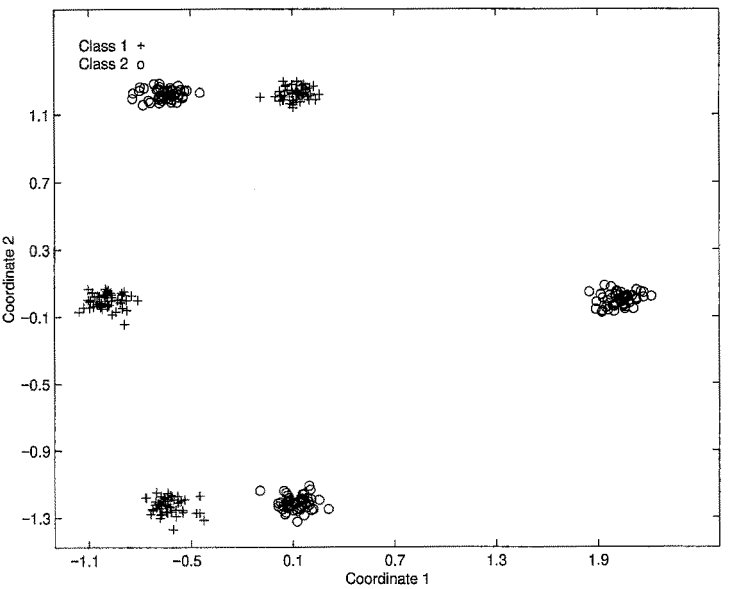


**Fig.4.** Sphered data set

We reduced the class separation along the direction under 64°. Here we study a situation which is highly unfavorable to our procedures: we use data with significantly nonnormal class-conditional distributions (see Fig.4), and we try to deflate the local maximum at 64° which has a large value (PF distance 0.7003) and which is located close to the global maximum at 15°.

We applied RCS along the direction under 64°: we set $\Delta m_1=0$, $\Delta m_2=0$, $\Delta\sigma^2=0$ in (4) and (5) and computed the transformed data $\mathbf{x}'$ (6). Then we calculated the PF distances for $\mathbf{x}'$ (dotted path "...." of Fig.5). Theoretically [7, p.254] and [8, p.456], the setting $\Delta\sigma^2=0$ implies minimal changes of the data after the "reduction of the class separation". Nevertheless, we observed a strong destructuring of the classification structure of the data in the result obtained (dotted path "...." of Fig.5). RCS deflates the PF distance in the range 0°-90° including the location of the global maximum at 15° which is undesirable for our recursive optimization procedure [2,3].

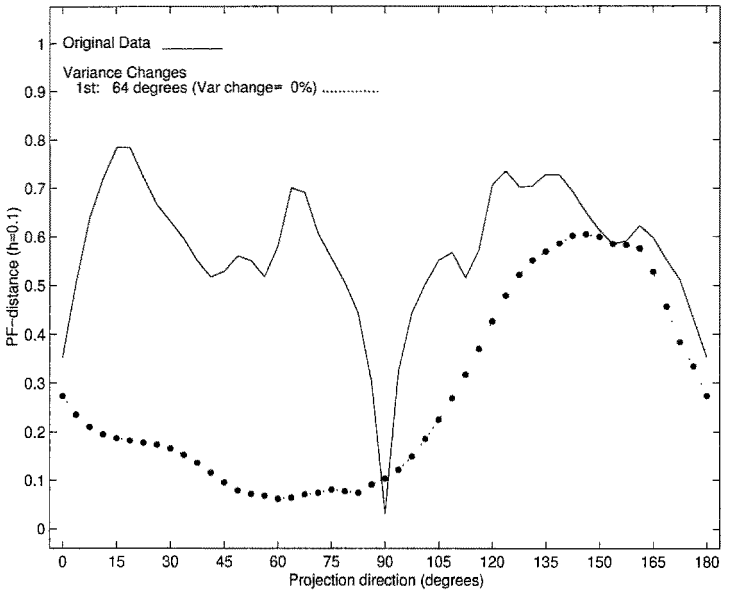


**Fig.5.** PF distance for various directions into $(x_1,x_2)$-plane: "___" original sphered data; "...." transformed data after RCS at 64°.

We ran NN_RCS setting $\Delta m_1=0$, $\Delta m_2=0$, $\Delta\sigma^2=0$ in (4) and (5), and $\nu=0.3$ in (8) using a two layer auto-associative network with 10 hidden units having sigmoid activation functions. We trained the network minimizing error function $E(\mathbf{u})$ (8) by a sequential quadratic programming method (routine E04UCF in the NAG Mathematical Library). We set the number of the batch (major) iterations of the optimization routine E04UCF to 150. We trained the network (Fig.3) using the original training data (Fig.4) and a penalty function (10) for the direction $\mathbf{w}^*$ under 64°. Then we propagated the original data through the trained auto-associative network and obtained a transformed data set (Fig.6). For this data set we calculated the PF distance for different directions (dotted path "...." in Fig.8). We gained some decrease of the class separation along the direction under 64° but we didn't manage to deflate the PF distance at 64°. We iterated the NN_RCS, i.e. we re-trained the auto-associative network using the transformed data (Fig.6) as a training set and the penalty function (10) computed for the direction under 71°, which is the modified location of the maximum which we try to deflate. Finally, we propagated the transformed data (Fig.6) through the re-trained auto-associative network and obtained a new data set shown in Fig.7. For the latter data set we computed the PF distances and observe that after two successive reductions of the class separation, NN_RCS deflates the maximum at 64° and preserves the location of the maximum at 15° (path "_._._" in Fig.8). NN_RCS decreases the value of the maximum at 15°. We view this as desirable because in our recursive optimization procedure [3] we can restore the actual value of the maximum found.

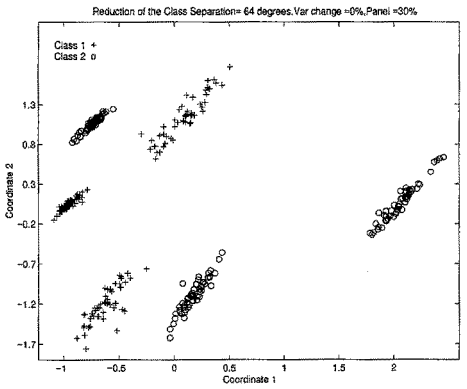


**Fig.6.** Transformed data after the NN_RCS at 64°.

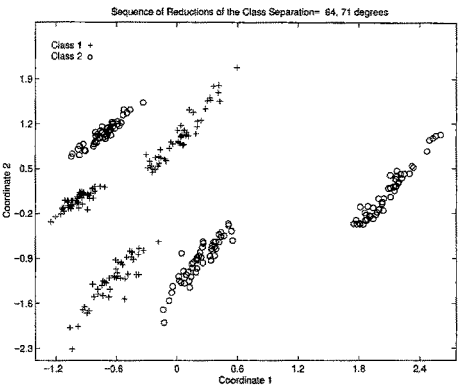


**Fig.7.** Transformed data after two successive runs of the NN_RCS at 64° and 71°.

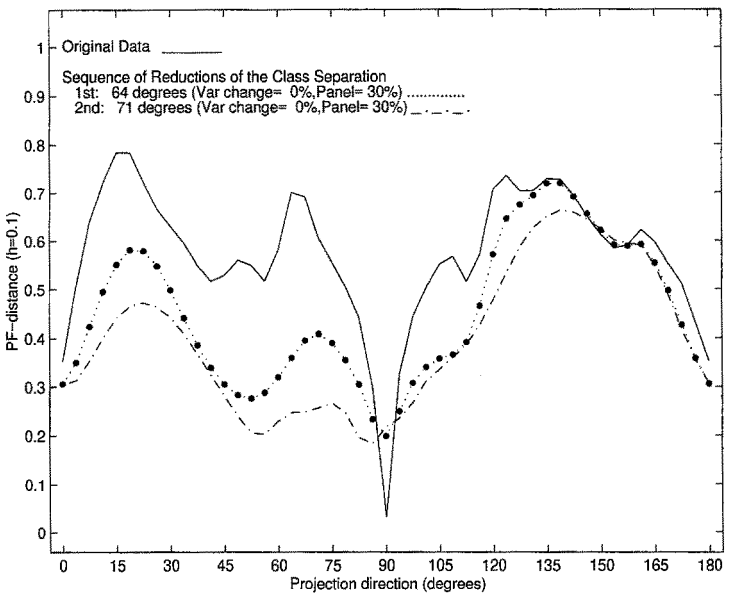


**Fig.8.** PF distance for various directions into $(x_1,x_2)$-plane:
"____" original sphered data; "...." transformed data after NN_RCS at 64°;
"_._._" transformed data after two successive runs of the NN_RCS at 64° and 71°.

## 6 Summary and Conclusion

We have discussed a method for the linear discrimination of two classes proposed by us in [3] previously. It searches for the discriminant direction which maximizes the Patrick-Fisher (PF) distance between the projected class-conditional densities. Since the PF distance is a highly nonlinear function, a sequential search for the directions corresponding to several large local maxima of the PF distance has been used. In order to ensure that a maximum already found will not be chosen again at a later stage we transform the data along a found direction into data with deflated maxima of the PF distance and iterate to obtain the next direction. For the success of this procedure it is important to preserve the location of the large local maxima which were not found in the previous stages.

In this paper we proposed a neural network implementation of our procedure for deflating a local maximum of the PF distance. The neural network by performing a highly non-linear data transformation, increases the efficacy of the procedure. By means of a simulation we demonstrated that the neural network succeeds in a situation which was highly unfavorable to our method. It managed to preserve the location of the global maximum of the PF distance after deflating a large local maximum which was located close to the global one.

The proposed neural network implementation can be used to reduce the class separation of the non-linear classification functions. In [4] we applied it for training an ML neural network by successive reductions of the class separation for the non-linear classification functions obtained by the ML network. It was proved that this training was more successful than conventional training with random initialization of the weights.

## References


1. M.E. Aladjem, "Multiclass discriminant mappings", *Signal Processing*, vol.35, pp.1-18, 1994.
2. M.E.Aladjem, "Linear discriminant analysis for two-classes via removal of classification structure", *IEEE Trans. Pattern Anal. Mach. Intell.*, vol.19, pp.187-192, 1997.
3. M.E.Aladjem, "Nonparametric discriminant analysis via recursive optimization of Patrick-Fisher distance", *IEEE Trans. on Syst., Man, Cybern.*, vol.28B, pp.292-299, 1998.
4. M.E.Aladjem, "Training of an ML neural network for classification via recursive reduction of the class separation", *14th Int. Conf. on Pattern Recognition*, Brisbane, Queensland, Australia, 17-20 August, 1998 (in press).
5. C.M. Bishop, *Neural Networks for Pattern Recognition*, Oxford University Press Inc., New York, 1995.
6. P.A. Devijver and J. Kittler, *Pattern Recognition: A Statistical Approach*, Prentice-Hall International, Inc., London, 1982.
7. J.H. Friedman, "Exploratory projection pursuit", *Journal of the American Statistical Association*, vol. 82, pp.249-266, 1987.
8. P.J. Huber, "Projected pursuit", including Discussions, *The Annals of Statistics*, vol. 13, pp. 435-525, 1985.
9. T.Lissack and K.Fu, "Error estimation in pattern recognition via $L^{\alpha}$-distance between posterior density functions", *IEEE Trans. Inf. Theory*, vol.IT-22, pp.34-45, 1976.