Direct Minimization of Error Rates in Multivariate Classification

Michael C. Röhl
Fachbereich Statistik
Universität Dortmund

Claus Weihs[*]
Fachbereich Statistik
Universität Dortmund

August 1999

Abstract

We propose a computer intensive method for linear dimension reduction which minimizes the classification error directly. Simulated annealing (Bohachevsky et al. 1986) as a modern optimization technique is used to solve this problem effectively. This approach easily allows to incorporate user requests by means of penalty terms. Simulations demonstrate the superiority of optimal classification to classical discriminant analysis (McLachlan 1992). Special emphasis is put on the case when discriminant analysis collapses.

KEY WORDS: classification, discriminant analysis, error rate, simulated annealing, user requests

1 Introduction

Classification deals with the allocation of objects with feature vectors x to g predetermined groups $G = \{1, 2, \ldots, g\}$, say. The goal is to minimize the misclassification rate over all possible future allocations given the group densities $p_i(x)$ ($i=$ 1 2,$\ldots, g$). The minimal error is the so–called *Bayes error* (McLachlan 1992). Often we want to reduce the dimension of the classification problem to one or two dimensions in order to support human imagination without significantly increasing the misclassification rate. This article deals with linear combinations of the original variables to achieve this goal: Linear Dimension Reduction (LDR).

[*] e-mail: weihs@statistik.uni-dortmund.de

In the literature, this problem is nearly always tackled by procedures using distance or scatter measures which are only surrogates to circumvent the classification error. Among these are the famous linear discriminant analysis (LDA) and quadratic discriminant analysis (QDA) (McLachlan 1992).

In this paper we focus on computer intensive strategies which minimize the misclassification error directly. We call these procedures "Minimal Error (Rate) Classifiers (MEC)" in order to make the difference to the classical approach clear. Figure 1 illustrates that both the classical and the computerintensive procedures are special cases of Linear Dimension Reduction procedures.

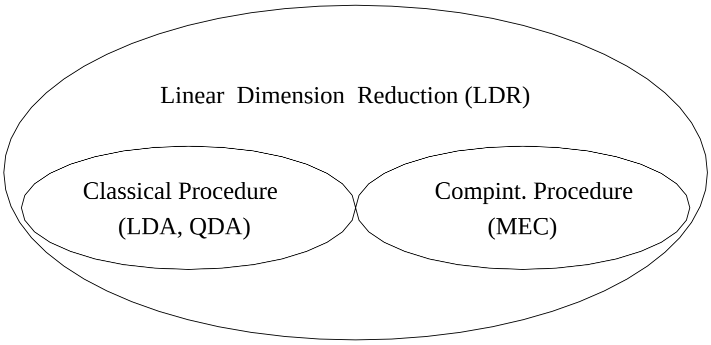

Figure 1: Different classification paradigms.

In the literature, direct error minimization up to now was only tackled by means of projection pursuit algorithms (Posse 1992, Polzehl 1995). In section 3 we will discuss the links between these algorithms and our MECs.

We will proceed as follows: In section 2 we will discuss the simulated annealing algorithm used to compute the MECs introduced in section 3. In addition, section 3 will show how to incorporate user requests (e.g. preference for some variables) in the optimization algorithm. Section 4 will illustrate the foregoing ideas in a simulation study. We will draw our conclusions in section 5.

2 Simulated Annealing

We discuss now the optimization algorithm used in the next section. The algorithm optimizes the entries in the projection matrix A to minimize the error rate. The optimization problem is therefore

$$\begin{aligned} \text{Minimize } f: \quad \mathbb{R}^{d' \times d} \quad &\rightarrow \quad \mathbb{R}^{+} \qquad (1)\\ \text{vectorized projection matrix a} &\mapsto \text{error rate,} \end{aligned}$$

where d' and d denote the (fixed) dimensions of the lower dimensional space and the original feature space, respectively.

We solve this optimization problem by using a simulated annealing algorithm which does not need derivatives, a great advantage compared to gradient methods. It can also be used if the function values are discrete. On the other hand one needs more function evaluations than common gradient algorithms.

In physics it is well-known that freezing and crystallizing of liquids overcomes local energy minima. This strategy serves as the prototype for a computer program: *Simulated Annealing* (Bohachevsky et al. 1986). To model the natural procedure, we need a configuration space (a discrete or continuous domain), a mechanism which describes how to get from one configuration to another and a cooling schedule describing how to decrease the temperature T ($T_0 \rightarrow T_1 \rightarrow \ldots \rightarrow T_n \rightarrow \ldots$). In a concrete optimization, the temperature T is not a physical quantity but an abstract parameter which controls the optimization.

In our problem the configuration space is $\mathbb{R}^{d' \times d}$, the space of vectorized projection matrices. In our algorithm, the cooling schedule is a simple linear scheme, and at each parameter value T a markov chain based on a stochastic version of the well-known Nelder/Mead search algorithm (Press et al. 1992) serves as the transition mechanism between succeeding configurations.

At each parameter value T – beginning at any configuration a_0 – we start a markov chain. This chain generates random realizations (after some burn–in period) from a density proportional to $\pi(a) = \exp(-f(a)/T)$. A trial point a_p is chosen according to some symmetric probability transition function $q(a_0, a_p)$. The efficiency of the optimization algorithm depends on this transition function and the cooling schedule. The transition function "explores" the configuration space. Information about this space enhances the search. The cooling schedule "encodes" the size of the neighbourhood that can be visited from a point of the markov chain.

The trial point is accepted with probability $\pi(a_p)/\pi(a_0) = \exp(-(f(a_p) - f(a_0))/T)$. In this way, in our problem projections leading to a decrease of misclassification are accepted in any case, but also projections increasing the error rate are accepted with some probability. This is the reason, why simulated annealing is able to overcome local optima and thus avoids the selection of multiple starting points.

After a number of steps in the markov chain, the parameter T_n will be decreased by the simple linear scheme $T_{n+1} = \alpha T_n$ $(0 < \alpha < 1)$, and a new chain will be created (the starting point of the new chain is the end point of the last one, see Figure 2).

We use an implementation of the simulated annealing algorithm based on a routine in NUMERICAL RECIPES IN C (Press et al. 1992). The basis of

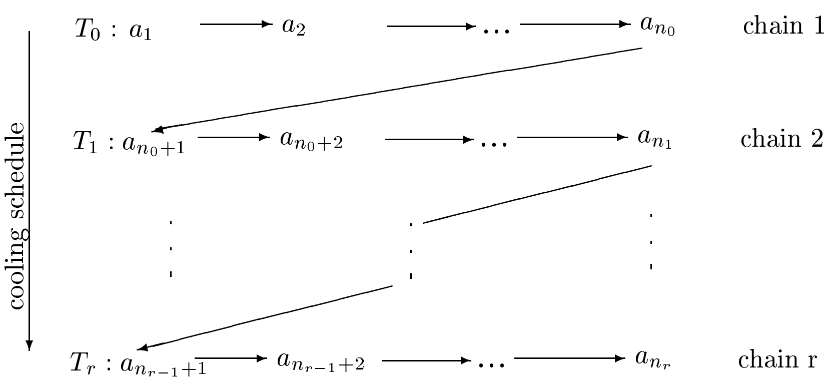

Figure 2: Flow chart of the simulated annealing algorithm.

this routine is a stochastic version of the search algorithm of NELDER AND MEAD (Press et al. 1992). This algorithm encloses the optimum by shrinking simplices. The shrinking is proportional to the parameter T_n. Therefore, as T_n approaches zero, the allowed movements will be more and more local, and the algorithm converges to the next optimum. Because of the bigger parameter values in the beginning of the procedure there is a good chance that this optimum is a global one.

For the not yet specified parameters of the simulated annealing procedure we have chosen the following values:

- initial value $T_0 = 5\ .0\%$
- $\alpha = 0\ .8$ or $0\ .9$
- number of iterations in the markov chain at each temperature: $30 - 50$

In our application of simulated annealing, the function to be optimized (minimized) is the misclassification rate. The next section shows how this error rate is calculated. Note that the initial projection matrix was obtained by means of a classical discriminant procedure (LDA or QDA) by projecting on the first discriminant coordinates.

3 Minimal Error (Rate) Classifiers (MEC)

3.1 Two versions of MEC

Basically, there are two possibilities to calculate the error rate during each optimization step: first estimate the densities in the original space and then find the optimal projection (MEC 2), or first project and then estimate the density in projected space and iterate until the errror rate is optimal (MEC 1).

1. **MEC 1:**
   First project the data, estimate the densities in the projected space and calculate the error rate by a modelfree, computerintensive technique (leave one out or bootstrap). In the following, we concentrate on the bootstrap technique.

   We draw B artificial samples D_i $(1 \leq i \leq B)$ from the given data set and project all the data into a lower dimensional space by a projection matrix. Then we determine the classification rule in the projected space by means of each training sample D_i and apply the rule to the test samples $original\, data\, set \setminus D_i$.

   The classification rule can be either a parametric rule or a nonparametric one. We use the classification rule based on the estimated group densities $p_i(x)$ $(1 \leq i \leq g)$ in the training sample and allocate an observation with feature vector x in the test sample to group i if

   $$\pi_i p_i(x) > \pi_j p_j(x) \quad (\quad j\!\!\!\neq i), \tag{2}$$

   where π_i denotes the group i apriori probability. This rule is optimal (minimal error rate) when the parametric form of the densities is known.

   The mean of the B error rates is the function value to be minimized. Besides, the B values allow us to judge the spread of the error rates, e.g. we can calculate a nonparametric 68% confidence interval.

   Note that this algorithm works in the projected lower dimensional space which might motivate normality of densities and results in reliable density estimates.

2. **MEC 2:**
   Estimate the densities in the original space first and project the estimated densities into a lower dimensional space. Then, the error rate could be estimated by means of the bootstrap technique described in MEC 1. In MEC 2, however, we decided to implement a totally parametric search technique in that, assuming d' is small enough and the projected group densities are not too complicated, we instead calculate the error rate by exact integration using the group densities.

   That is, for each group $i \in G$, we determine the regions where at least one of the other group densities is greater. We integrate $p_i(x)$ over these regions and get the misclassification error conditional on this group i. The total error is calculated as an average over all groups weighted by their apriori probability. Thus, in (1) error rate $= \sum_{i=1}^{g} \pi_i$ $\int_{B_i} p_{A,i}(x)$ dx, where π_i is the apriori probability of group i, $B_i = \{x \mid$

> $\exists j \, : \, p_{A,i}(x) < p_{A,j}(x)\}$, $p_{A,i}$ is the i–th projected group density, and A is the matrix applied to project into lower dimensional space.
>
> Another possibility to determine the error rate is Monte Carlo Simulation which generates random realizations from the group densities and allocates them according to the classification rule (2).
>
> Obviously, MEC 2 has the drawback that densities have to be specified in orginal space. After the optimal projection space is determined, we project the data set into the lower dimensional space and calculate the error rate by a modelfree, computerintensive technique (leave one out or bootstrap).

Both possibilities have their merits and drawbacks as will be discussed in section 4. Figure 3 contrasts the two methods.

Polzehl (Polzehl 1995) addressed the problem of constructing an optimal classification rule by projection pursuit based density estimation. In his algorithm, density estimation is adaptively guided by the error rate. In this respect, his algorithm is more general than MEC 2. But it is not clear whether the lower dimensional space determined in this way is really the one producing the lowest error rate because the error rate of the "whole" density is not minimized, but the error rate of one dimensional slices. The author uses a combination of stochastic search and the already mentioned Nelder and Mead algorithm. We think that simulated annealing is a more powerful minimization tool, especially to avoid getting trapped in a local minimum.

Another contribution in this field is Posse's paper (Posse 1992). This paper is very close in spirit to our approach in MEC 2 because the error rate of the group densities is directly minimized. However, his reasoning is limited to two groups and the optimization algorithm uses different random starting points to overcome local minima instead of the powerful simulated annealing procedure. In section 4 it is shown that especially in the case of more than two groups there are severe differences between the classical and computer-intensive approach.

Both authors do not discuss an algorithm like MEC 1.

The next subsection shows how users can incorporate different weights (preferences) associated with the features.

3.2 User requests

We will discuss a combination of two kinds of user requests, one concerning preferences for features and one concerning a tolerated error rate. Then the function to be minimized is not the error rate any more, but the linear

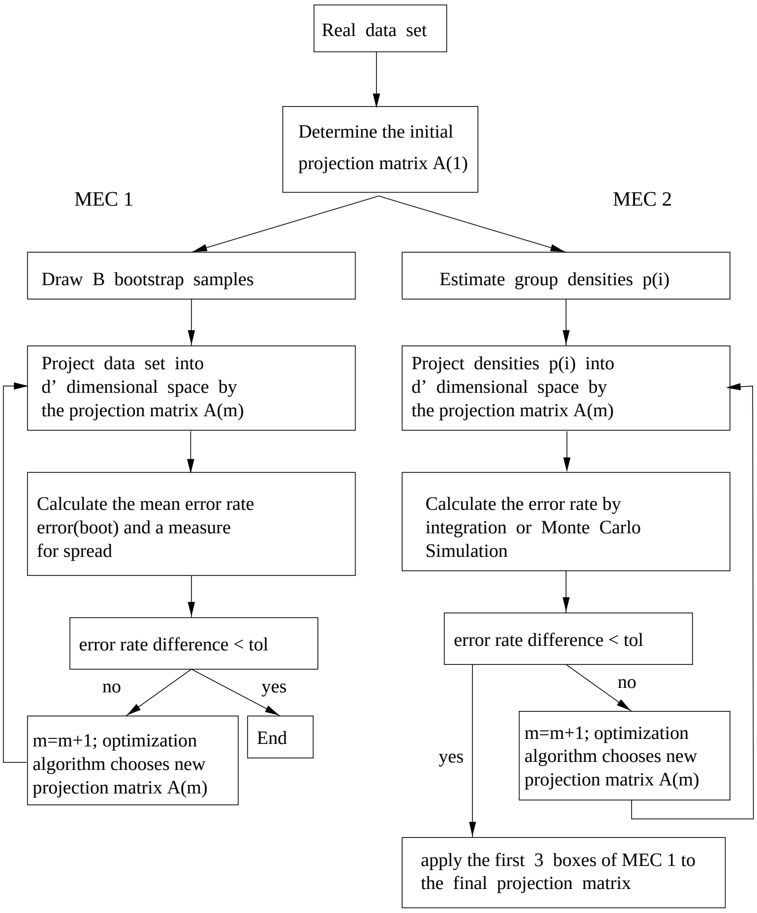
Real data set
Determine the initial projection matrix A(1)
MEC 1
MEC 2
Draw B bootstrap samples
Estimate group densities p(i)
Project data set into d' dimensional space by the projection matrix A(m)
Project densities p(i) into d' dimensional space by the projection matrix A(m)
Calculate the mean error rate error(boot) and a measure for spread
Calculate the error rate by integration or Monte Carlo Simulation
error rate difference < tol
error rate difference < tol
no
yes
no
yes
m=m+1; optimization algorithm chooses new projection matrix A(m)
End
m=m+1; optimization algorithm chooses new projection matrix A(m)
apply the first 3 boxes of MEC 1 to the final projection matrix

Figure 3: Comparison of MEC 1 and MEC 2.

combination

$$S_1 + S_2 := \kappa_1 P_1 + \kappa_2 P_2, \tag{3}$$

where $0 \leq \kappa_1, \kappa_2$ and P_1, P_2 are two penalty terms. The first penalty term is

$$P_1 = \sum_{i=1}^{d'} \sum_{j=1}^{d} K_j(|a_{ij}|) \quad , K_j(|a_{ij}|) \geq 0, \tag{4}$$

where d' denotes the dimension of the lower dimensional space and $K_j(|a_{ij}|)$ are the weights (costs) of the different features. These costs depend on the entries a_{ij} of the projection matrix A. Only the relative magnitude of these costs compared to each other is relevant, not the absolute value. In this paper we use the natural choice

$$K_j(|a_{ij}|) = |a_{ij}|K_j \quad , K_j \geq 0. \tag{5}$$

Another choice of the cost function would be the more sophisticated logistic function

$$K_j(|a_{ij}|) = \frac{\exp((|a_{ij}| - t_j)/s_j)}{1 + \exp((|a_{ij}| - t_j)/s_j)} \quad \forall\, i = 1\,, \dots, d \tag{6}$$

with target weight $t_j > 0$ and smoothness $s_j > 0$. The bigger s_j, the smoother $K_j(|a_{ij}|)$ varies around the target weight.

The second penalty term is taking into account the deviation of the error rate from a tolerated error rate t :

$$P_2 = \frac{\exp((error\,rate - t)/s)}{1 + \exp((error\,rate - t)/s)} \quad ; t, s > 0, \tag{7}$$

with, again, s being a smoothness parameter. The tolerated error rate specifies how much bigger than the optimal error rate a realized *error rate* is allowed to be with the suboptimal choices of the projection space caused by unequal costs for the different features. In order to judge how big t should be, one should calculate a reference error rate with $K_j = 0\ \forall\, j$. An application will be given in subsection 4.3.

4 Simulations

4.1 Comparison of MEC 1 and MEC 2

To compare MEC 1 with MEC 2, we consider the following two group case with normal densities $p_i(x)$ (i= 1 ,2) and parameters

$$\begin{aligned} \mu_1 &= (0\,,0, \dots, 0)' \quad \Sigma_1 = I_d \quad \text{and} \\ \mu_2 &= (2\,,0, \dots, 0)' \quad \Sigma_2 = I_d. \end{aligned} \tag{8}$$

The dimension is varied ($2 \leq d \leq 10$) and the sample size is fixed at $n_1 = n_2 = 100$. We want to find the best one dimensional projection direction. The following procedure is repeated 100 times in each dimension:

1. Generate $n_1 = n_2 = 100$ random realizations according to the group densities $p_i(x)$ (i= 1 ,2).

2. Apply MEC 2 to these samples.

3. Draw $B = 200$ bootstrap samples from each sample and apply MEC 1.

This procedure results in 100 error rates for MEC 1 and MEC 2. The mean and standard deviation of these error rates are used to characterize the goodness of both methods.

Table 1 lists the results. The columns entitled with the corresponding method report the mean ± the standard deviation. The columns with the title angle$_{mean}$ report the mean of the angles between the true optimal projection direction $(1, 0, \ldots, 0)'$ and the calculated one.

dimension	MEC 1	angle$_{mean}$	MEC 2	angle$_{mean}$
$d = 2$	$15.0 \pm 2.6\%$	$8.2°$	$15.5 \pm 2.6\%$	$4.3°$
$d = 4$	$14.2 \pm 2.3\%$	$15.3°$	$15.8 \pm 2.9\%$	$9.4°$
$d = 6$	$13.7 \pm 2.2\%$	$19.0°$	$15.6 \pm 2.8\%$	$13.2°$
$d = 8$	$13.0 \pm 2.2\%$	$21.2°$	$15.1 \pm 2.4\%$	$14.3°$
$d = 10$	$11.8 \pm 1.8\%$	$24.0°$	$15.3 \pm 2.6\%$	$16.6°$

Table 1: MEC 1 and MEC 2 results ordered by dimension.

One may argue that MEC 1 can not be recommended if the quotient (sample size)/dimension is small:

- The small sample is not representative for the underlying group densities. The optimization leads to an optimistic bias because there are many possibilities to project into an one or two dimensional space separating the few points in an optimal way.
- I.e., the optimal projection space should not be expected to be identifiable because there will probably be many possibilities to separate, e.g., 7 points in two groups in two dimensions with an error of one or two points. Then, which projection space should we choose?

Table 1 clearly indicates the optimistic bias. From dimension eight on, the true Bayes error of 15.9% does not even lie in the one standard deviation interval. Moreover, the mean deviation angle to the optimal direction is bigger with MEC 1 than with MEC 2.

Both methods use normal densities as group densities and estimate the parameters from the sample. This means that both methods use the same model, i.e. there is no information bias in the model choice and the differences in table 1 are only due to the different procedures.

MEC 1 might be more useful when the densities are complicated, so that we have to resort to nonparametric density estimation. Indeed, when we then do not have enough data to estimate the group densities in the original space, MEC 1 might be the better alternative, especially when the sample size is large enough to reduce the optimistic bias. This can be judged by small simulations: approximate the group densities by some simple group densities, fixing at least the group means and spread, and perform the same simulation as above. Then you can decide whether the bias is significant for your problem at hand.

Note that the projection space is determined in a model dependent way in MEC 2, but finally, the error rate is calculated modelfree by leave one out or bootstrap with the projected data set.

4.2 Comparison of MEC 2 and discriminant analysis

MEC 2 with known densities is now compared to the classical approach. The classical procedures do not provide a direct link to the misclassification rate (that is, from a small perturbation of the direction a, you can not analytically derive the corresponding variation in the misclassification rate). In fact, in some special cases (depending on constellation of the groups and the form of the covariance matrices), a significant difference between the two procedures can be detected. Apart from the pure comparison, emphasis is put on the question when classical methods collapse. In these cases only MEC 2 supplies valid results.

The classical procedure LDA proceeds by maximizing

$$\Delta_1 := \frac{a' S_B a}{a' \Sigma a}, \quad \text{where} \quad ||a|| = 1 \qquad (\ McLachlan 1992). \tag{9}$$

In this formula $S_B = \sum_{i=1}^{g} \pi_i (\mu_i - \bar{\mu})(\mu_i - \bar{\mu})'$, where μ_i = mean in the i-th group, $\bar{\mu}$ = overall mean and $\pi_i = \frac{1}{g}$ = apriori group probabilities, is the between-groups covariance matrix and Σ is the common covariance matrix inside the groups.

To compare MEC 2 and LDA, we conduct some simulations with three groups assuming normality. First, we transform the common covariance matrix Σ by the transformation $x_{new} := \Sigma^{-1} x_{old}$ to the identity matrix I_d. This does not change the misclassification rate. Because of the symmetry induced by three groups, it suffices to take $d = 2$. Therefore we set

$$\mu_1 = (0\ ,0)', \quad \mu_2 = (2\ ,0)' \quad \text{and} \quad \mu_3 = (\ x, y)'. \tag{10}$$

Mean μ_1 only determines the origin and μ_2 is somewhat arbitrary. A variation of μ_2 would only alter the misclassification level, not the qualitative conclusion. The third mean contains two variables x and y.

We are looking for the optimal 1D-projection. The two dimensional surface of the dependency of the corresponding error rate on x, y can be conveniently plotted. Because of the symmetry of the constellation, it is enough to regard the positive quadrant. We take the range $0 \leq x \leq 2.5$ and $0 \leq y \leq 2.5$.

Figures 4 and 5 show the estimated misclassification rates (given the means and the covariance matrix) of the classical and the optimized procedure, respectively. Note the different scales of the two graphs.

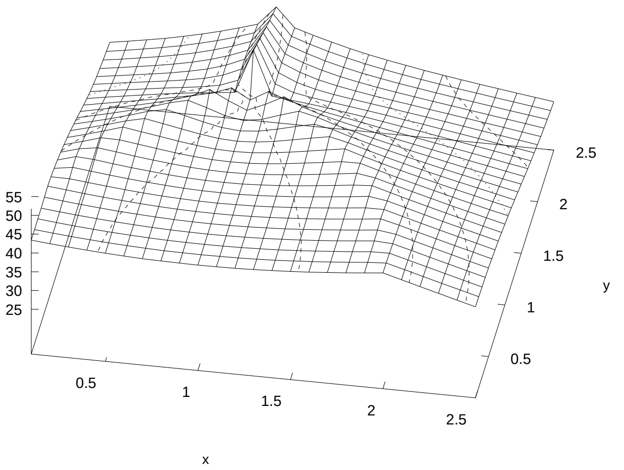

Figure 4: Misclassification rate with the classical procedure.

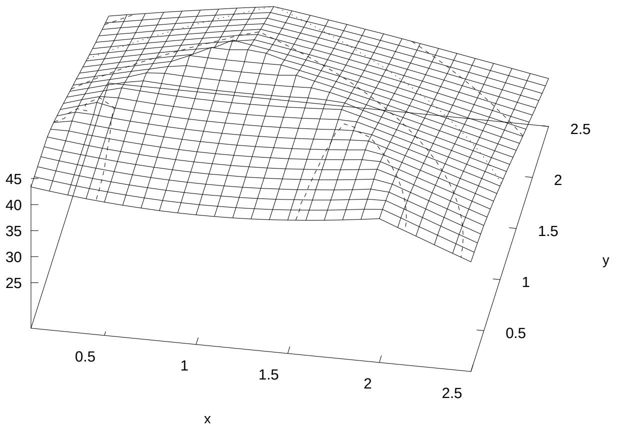

Figure 5: Misclassification rate with the optimized procedure.

The results of the classical procedure are qualitatively similiar in the "front" range ($0 \leq x \leq 2.5$ and $0 \leq y \leq 1.7$), whereas there is a significant difference in the "back".

We now analyze the reason of the "mountain ridge" in the classical case in more detail. A special situation arises, when the means of the three groups constitute a regular triangle. For that reason, we reparametrize the third mean: $\mu_3 = (1 + \delta x, \sqrt{3} + \delta y)$. In the special case $\delta x = 0$ one can then show that

$$\Delta_1 = \frac{2}{9} \left\{ (\delta y + \sqrt{3})^2 - a_1^2 \left((\delta y + \sqrt{3})^2 - 3 \right) \right\}, \tag{11}$$

and after maximization we get the following distinction of cases:

$$\begin{array}{lllll} (\delta y + \sqrt{3})^2 > 3 & \Leftrightarrow & \delta y > 0 \quad \text{or} \quad \delta y < -2\sqrt{3} & \Rightarrow & a_1 = 0 \,, a_2 = 1 \quad (12) \\ (\delta y + \sqrt{3})^2 < 3 & \Leftrightarrow & - \ \sqrt{3} < \delta y < 0 & \Rightarrow & a_1 = 1 \,, a_2 = 0 \\ (\delta y + \sqrt{3})^2 = 3 & \Leftrightarrow & \delta y = 0 & \Rightarrow & a_1, a_2 \text{ arbitrary.} \end{array}$$

The mean $\mu_3 = (1 \ \sqrt{3})'$ results in a singularity (projection vector $a = (a_1, a_2)'$ not defined). But this mean is realized with probability zero by the empirical mean value and is therefore unimportant. More important is the fact that the projection behaviour "turns over" at this value. As long as $\delta y < 0$, the projection is onto the x–axis (as by the optimized procedure), then onto the y–axis. This causes a higher misclassification rate compared to the optimized procedure, because the projected first group coincides with the second one, while the optimized method still projects onto the x–axis.

The classical approach even more often fails for more than $g = 3$ groups, because there are more critical constellations. A more detailed discussion and a comparison with quadratic discriminant analysis in the case of unequal covariance matrices can be found in (Röhl and Weihs 1999).

4.3 User requests

We use the $I - \Lambda$ group densities introduced in Fukunaga (1990). The parameters of the 8-dimensional normal densities are

$$\begin{aligned} &\mu_1 = (0 \,, \ldots \,,0)' \quad \text{with} \quad \Sigma_1 = \mathrm{diag}(1, \ldots, 1) \quad \text{and} \\ &\mu_2 = (3 \ .86,3.10, 0.84, 0.84, 1.64, 1.08, 0.26, 0.01)' \quad \text{with} \\ &\Sigma_2 = \mathrm{diag}(8.41, 12.06, 0.12, 0.22, 1.49, 1.77, 0.35, 2.73). \end{aligned} \tag{13}$$

There are no symmetries and the constellation is therefore quite general. The true Bayes error is 1.9% and the best one-dimensional projection gives an error slightly bigger than 5%.

The penalty terms (4) and (7) are combined with weights $\kappa_1 = 1$, $\kappa_2 = 16$ and $s \approx 0$, i.e. P_2 is a hard constraint. The cost vector is

$$K = (0 \ .0,0.0, 2.0, 2.0, 2.0, 2.0, 2.0, 2.0). \tag{14}$$

t (tolerance)	minimum $S_1 + S_2$	projection direction
6%	2.18	(0.68,0.41,0.19,0.18,0.51,0.22,0.00,0.00)'
8%	0.60	(0.81,0.51,0.00,0.00,0.27,0.03,0.00,0.00)'
10%	0.00	(0.66,0.75,0.00,0.00,0.00,0.00,0.00,0.00)'

Table 2: Results ordered by the tolerance t.

These costs study the importance of the first two features.

With a tolerance level of 6%, the influence of the fifth feature on the best one-dimensional projection can not be removed (cp. table 2). Only the last two features have no influence. With 8%, the influence of feature five is reduced, all the other features, except for the first two, are rather unimportant. With 10%, the first two features are enough to separate the groups. Now we can vary the costs and prefer other variables (costs=0). This gives an impression of the importance of the features when the other ones are present.

5 Conclusions

The computerintensive methods MEC 1 and MEC 2 based on the powerful simulated annealing optimization algorithm minimize the classification error rate directly and are flexible tools for incorporating constraints, e.g. user requests. For more than two groups, it has been shown that there exist some group constellations which induce a significant difference between this approach and the classical procedure. In future work, the optimistic bias of MEC 1 should be reduced.

6 Computation

The simulation study was done on a 200MHz PC with 64MB main memory using the programming language C. The operating system was LINUX. With MEC 2 the identification of the optimal one dimensional projection needs a few seconds, the identification of the optimal 2D-plane from some minutes to half an hour depending on the grid size in the integration routine computing the error rate.

Acknowledgment

This work has been supported by the Collaborative Research Centre ”Reduction of Complexity in Multivariate Data Structures” (SFB 475) of the German Research Foundation (DFG). We particularly thank Dipl. Math. W. Theis for his constructive criticism.

References

I. O. Bohachevsky, M. E. Johnson, M. L. Stein, *Generalized Simulated Annealing for Function Optimization*, Technometrics, **28**, 3, 209-217 (1986).

G. J. McLachlan, *Discriminant Analysis and Statistical Pattern Recognition*, John Wiley & Sons (1992).

C. Posse, *Projection Pursuit Discriminant Analysis for two Groups*, Commun. Statist.- Theory and Methods, **21**, 1, 1-19 (1992).

J. Polzehl, *Projection pursuit discriminant analysis*, Comp. Stat. and Data Analysis, **20**, 141-157 (1995).

W. H. Press, B. P. Flannery, S. A. Teukolsky and W. T. Vetterling, *Numerical Recipes in C*, 2nd edition, 444-455, Cambridge University Press, Cambridge (1992).

M. C. Röhl and C. Weihs, *Optimal vs. Classical Linear Dimension Reduction*,in: W. Gaul, H. Locarek-Junge (eds.), Classification in the Information Age, Studies in Classification, Data Analysis, and Knowledge Organization, Springer, 252-259 (1999)

K. Fukunaga, *Introduction to Statistical Pattern Recognition*, 2. Auflage, New York: Academic Press (1990).