Implementing a new method for Discriminant Analysis when Group Covariance Matrices are nearly singular

Winfried Theis[1], Christian Röver[1], and Britta Pouwels[2]

[1] Sonderforschungsbereich 475, Fachbereich Statistik,
Universität Dortmund, D-44221 Dortmund, Germany
[2] Universitäts- und Landesbibliothek Münster,
D-48143 Münster, Germany

Abstract. We consider a unified description of classification rules for nearly singular covariance matrices. When the covariance matrices of the groups or the pooled covariance matrix become nearly singular, bayesian classification rules become seriously unstable. Several procedures have been proposed to tackle this problem, e.g. SIMCA, and Regularized Discriminant Analysis. Næs and Indahl (1998) discovered common properties for all of these procedures and proposed a unified classifier that incorporates the functionality of them all. Since the unified approach needs many parameters, they also proposed an alternative classifier with fewer parameters. We implemented both classifiers and compared them in a simulation study to the procedures RDA, LDA, and QDA. To enhance the comparability of our results we based the simulation study on the study of Friedman (1989).

In the implementation, we used a combination of the Nelder-Mead Simplex-algorithm and Simulated Annealing (Bohachevsky et al. (1986)) to optimize the classification error directly.

Keywords

DISCRIMINATION, MULTICOLLINEAR DATA, CLASSIFICATION ERROR, OPTIMIZATION

1 Introduction

The classical approaches to discriminant analysis are based on estimates of the inverse covariance matrices for each group (Quadratic Discriminant Analysis, QDA) or on a common inverse covariance matrix (Linear Discriminant Analysis, LDA). The estimates of the covariance matrix can become badly conditioned for several reasons. First of all the explanatory variables may be multicollinear. Another situation encountered is an insufficient number of observations per group, which leads to a similar problem. Since these problems are encountered in many studies, a wide variety of methods has been developed to tackle this problem. There are mainly two typical approaches: On the one hand dimension reduction techniques like Principal Component

Analysis (PCA), Partial Least Squares (PLS) or Minimal Error Rate Classifiers (MEC) (Röhl et al. (2002)) are applied. On the other hand there are methods which preserve the dimensionality but manipulate the covariance matrix estimates, like MCA, SIMCA, DASCO or Regularized Discriminant Analysis (RDA) (please refer to Næs and Indahl (1998) for these methods). Næs and Indahl (1998) recognized that the latter methods could be described in a unified way.

In this paper we consider first the unified description of Næs and Indahl in section 2. Next we explain our approach for an implementation of the method (section 3). We tested our algorithm in an extensive simulation study and compared it to RDA, and the classical LDA and QDA (sections 4 and 5). Finally we conclude with some comments and suggestions for further research.

2 Unified Description

Næs and Indahl recognized three "conceptual dimensions" in which the different methods for manipulating the covariance matrix can be organized. We reformulated these "dimensions" in the following way:

1. Manipulation of singular eigenvalues λ_k , $k = 1, \dots, d$.
2. Shrinking of the group-specific covariance matrices S_j , $j = 1, \dots, G$, towards the pooled covariance matrix S_p.
3. Shrinking of the covariance matrices S_j , $j = 1, \dots, G$, towards the identitiy matrix I_d.

The first dimension yields the opportunity to replace some eigenvalues by values which result in more stable inverses or reduce the typical bias in eigenvalue-estimates. The aforementioned bias corresponds to the fact that small eigenvalues are estimated too small and large eigenvalues too large.

The second dimension focuses on stabilizing the estimate by moving towards an estimator which comprises more knowledge from the data, and is therefore more stable.

Finally the third dimension tries to solve the multicollinearity problem by reducing the influence of covariances between the explanatory variables.

Formula 1 gives the complete form of the covariance estimator for the jth group, which includes all regularisation possibilities.

$$\hat{\Sigma}_U^j = c_j \left[\delta (U_1^j U_2^j) \begin{pmatrix} m\Lambda_1^j + \alpha_j I_K & 0 \\ 0 & (\alpha_j + \beta_j) I_{d-K} \end{pmatrix} (U_1^j U_2^j)^T + (1-\delta) S_p \right] \tag{1}$$

where $(U_1^j U_2^j)$ is a partition of the $d \times d$ eigenvector matrix of the j-th group covariance matrix, with U_1^j corresponding to the large eigenvalues in the K-dimensional diagonal matrix Λ_1^j and U_2^j corresponding to the $d - K$ small eigenvalues. The other parameters correspond to the different regularisations in the following way:

- $K \in \{0, \dots, d\}$ is the dimension of the space of high variability.
- $\delta \in [0, 1]$ models the shrinking towards the pooled covariance matrix.
- $\alpha_j \in [0, \infty)$ models the shrinking towards the identity matrix.
- m manipulates the large eigenvalues similarly for all groups.
- $\beta_j \in [0, \infty)$ is the replacement for the small eigenvalues.
- $c_j \in [0, \infty)$ is an additional scaling factor.

Thus all mentioned directions of regularisation can be achieved with this estimator. A great problem are the additional $3 + 2G$ parameters which have to be derived from the data to use this estimator. When a covariance estimate for LDA is to be regularized, the pooled covariance matrix is used instead of the group covariance matrices. Thus the group indices j and parameter δ are omitted in formula 1, and the eigenvalue-decomposition is applied directly to the pooled covariance matrix.

To avoid the large number of parameters of the unified approach, Næs and Indahl proposed an alternative estimator by omitting the parameters c_j, α_j and by not using a group-specific choice of β:

$$\hat{\Sigma}_A^j = \delta(U_1^j U_2^j) \begin{pmatrix} m\Lambda_1^j & 0 \\ 0 & \beta I_{d-K} \end{pmatrix} (U_1^j U_2^j)^T + (1 - \delta) S_p \tag{2}$$

This parameter reduction leads mainly to the loss of the possibility of regularization in the third of our dimensions, the shrinking towards the identity matrix. Furthermore the group-specific manipulation of the small eigenvalues is lost.

In our simulations (see section 4) we compare these two approaches which will be called **U**nified **R**egularized **C**lassification method (URC), or **A**lternative **R**egularized **C**lassification method (ARC), resp., to standard LDA and QDA, and as a further competitor we chose Regularized Discriminant Analysis (Friedman (1989)), since RDA especially shrinks towards the identity matrix and so it is of great interest to compare the performance of the alternative estimator to RDA. RDA regularizes in dimensions 2 and 3 and actually has two paramters δ, γ steering the shrinkage. The first step is the shrinkage towards the pooled covariance matrix:

$$\hat{\Sigma}_j(\delta) = \delta S_j + (1 - \delta) S_p\,,$$

and the second step is towards a weighted identity matrix:

$$\hat{\Sigma}_j(\delta, \gamma) = (1 - \gamma)\hat{\Sigma}_j(\delta) + \frac{\gamma}{d} tr(\hat{\Sigma}_j(\delta)) I_d,\; j = 1, \dots, G$$

The unified estimate equals the RDA estimate, if

$$K = d,\; c_j = 1 - \gamma,\; m = 1,\; \alpha_j = \frac{\gamma tr(\delta S_j + (1 - \delta) S_p)}{d(1 - \gamma)\delta}.$$

Before the simulation study will be described, we point out some important issues about the implementation.

3 Algorithm

Since there is no analytical way to estimate the regularization parameters, an optimization has to be performed. We chose as objective for this optimization the minimization of the mean prediction error rate, which is calculated by splitting the observations into equally sized training and test sets four times and predicting the classes of the test data. As optimization algorithm we chose the Nelder-Mead-Simplex Algorithm which is able to cope with a higher dimensional search space and does not need derivatives or other information of the functional form of the objective. We also tested a combination of Nelder-Mead with Simulated Annealing to overcome possible local minima. The standard version of this algorithm makes the assumption of an unrestricted parameter space, which is not true in our case. So the different restrictions on the parameters had to be implemented to ensure the selection of valid parameter values. Furthermore the parameter K is an integer which has to be optimized separately or appropriately included in the optimization process.

The restrictions were implemented first by testing the validity of each new parameter vector. If one parameter is outside its respective boundaries, it is set to the nearer bound of the parameter space. For example, if δ, which stands for the convex combination of the group covariance with the pooled covariance, becomes greater than 1 by expansion of the simplex, it is set to 1.

In a second step we tested whether smooth transformations of the parameters into the admissable region might lead to better results. For the parameter δ a sigmoidal function of the form $f(x) = \dfrac{1}{1+e^{-x}}$ is used, for the other parameters we used the exponential function as transformation.

The greatest difficulty was the inclusion of the parameter K into the optimization. The first approach was to select a range of interest for K and optimize for each K and then select the set of parameters with the best estimated error rate. But this approach is very time consuming so a second approach was tested. In this second approach the search for an optimal K is included into each optimization step by testing three subsequent values of K lieing next to the best value of K found so far. If nothing else is specified, the starting value is chosen by calculating the mean of the eigenvalues over the groups and choosing the number of large eigenvalues which contribute at least 75% to the sum of all mean eigenvalues.

Finally we implemented our own version of RDA by using the algorithm with the sigmoidal transformation described above to optimize the parameters δ and γ.

4 Simulation Study

Two test data sets are constructed as described in the study of Friedman (1989) to make our results comparable to the findings there. In all settings

Friedman considered three groups in the data. Here we restrict ourselves to two settings from this study, namely Setting 1 where the mean vectors are $\mu_1 = \mathbf{0}$, $\mu_2 = (3, 0, \ldots, 0)^T$, $\mu_3 = (0, 3, 0, \ldots, 0)^T$ and covariance $\Sigma_j = diag(1, \ldots, 1)$ and Setting 4, where the eigenvalues of the covariance matrices are equal and determined by $e_i = (9(i-1)/(d-1)+1)^2$, $i = 1, \ldots, d$ the means are $\mu_1 = \mathbf{0}$, $\mu_2 = (2.5\sqrt{e_i/d}\frac{d-i}{d/2-1})_{i=1}^d$, $\mu_3 = \left((-1)^i \mu_{2i}\right)_{i=1}^d$. So in the latter situation the mean differences are concentrated in the high-variance subspace, which makes it difficult for RDA.

As a further test data set we constructed extremely multicollinear data by simulating a one-dimensional normal variable which is projected into a six dimensional space and adding noise of different magnitude for each dimension. The space of high variability consists of three dimensions. Again we defined three different group distributions. We considered three different situations of this type:

1. Equal covariance structure with different spreads, mean differences in high variance space.
2. Different covariance structures with highest spread in orthogonal directions, mean differences in high variance space.
3. Covariance structures as in 2., but higher total variance, and mean differences in low variance subspace.

For all situations described above 100 replications were sampled and evaluated. Like Friedman we trained the procedures on samples of size 40 for the settings taken from his study, which were randomly distributed into the three groups, but used equal priors of 1/3 in all procedures. For the multicollinear data sets we used training samples of size 100. The test data sets consisted each of 100 observations. We applied LDA and QDA as implemented in the MASS library (Venables and Ripley (1999)) for **R** and our new routines named URC and ARC are implemented in **R** (Ihaka and Gentleman (1996)) as well. We applied ARC with and without simulated annealing. Beside the results for our implementation of RDA we report the results from the orginal paper by Friedman.

5 Results of the Simulation

Tables 1 and 2 give an overview of the results for the two settings from Friedman. The "# no results" lines give the number of test sets where singular covariance matrices appeared in QDA. It is obvious, that the Næs procedures can not compete with RDA. But they are better than QDA in all situations and the unified approach is equally good or even better than LDA.

The results give rise to the question, why the results of URC and ARC are so similar to LDA. A possible explanation are high values of the shrinkage parameter towards the pooled covariance. Figure 1 shows indeed, that δ is nearer to 1 in dimensions $d = 20$ and $d = 40$ and with $d = 6$ and $d = 10$ the median still lies above 0.5. Therefore, it is not surprising, that the unified approach can not be more effective than LDA, since with a high δ the effect of most of the other parameters is diminished.

	Number of variables			
	6	10	20	40
Friedman RDA	0.11 (0.03)	0.12 (0.04)	0.16 (0.05)	0.19 (0.05)
δ	0.77 (0.037)	0.79 (0.035)	0.75 (0.037)	0.78 (0.034)
γ	0.74 (0.034)	0.72 (0.032)	0.74 (0.028)	0.80 (0.022)
own RDA	0.12 (0.04)	0.13 (0.04)	0.16 (0.05)	0.20 (0.05)
RDA with Sim. Ann.	0.11 (0.04)	0.13 (0.04)	0.15 (0.05)	0.19 (0.04)
LDA	0.13 (0.04)	0.16 (0.05)	0.26 (0.06)	0.48 (0.08)
QDA	0.25 (0.07)	0.46 (0.08)	- (-)	- (-)
# no result	2	53	100	100
URC	0.13 (0.04)	0.17 (0.07)	0.30 (0.16)	0.39 (0.13)
Trafo'-URC	0.16 (0.10)	0.21 (0.14)	0.27 (0.17)	0.49 (0.15)
ARC	0.12 (0.03)	0.16 (0.04)	0.26 (0.07)	- (-)
Trafo'-ARC	0.13 (0.04)	0.17 (0.07)	0.26 (0.07)	- (-)
ARC + SimAnn.	0.14 (0.06)	0.16 (0.04)	0.26 (0.08)	- (-)

Table 1. Means of prediction error rates from Setting 1 by Friedman; the numbers in parantheses are the standard deviations; Trafo'-*RC means that transformations have been used to build in the restrictions on the parameters

A second possible question is if ARC or URC favour parameter settings similar to RDA in these Settings 1 and 4. This would mean that parameters c_j and α_j, $j \in \{1, \ldots, G\}$, would behave like the outer convex combination in RDA realized by γ. This is not the case as can be seen from Figure 2, where the realized values are depicted in the boxplot on the left hand side of each column and on the right hand side the corresponding values for equivalence to RDA are shown. The latter values are calculated from the values realized in our own implementation of RDA. Obviously α_1 is estimated too low for an equal effect as in RDA and c_1 is estimated too high.

The parameters $\alpha_j, j \in \{1, \ldots, G\}$ show an interesting behaviour: They are similar to the constant weight of the identity matrix in RDA, $\frac{1}{d} tr(S_j(\delta))$. We approximate the trace by the sum of the true eigenvalues of the covariance matrices, and plot these points as horizontal lines into the boxplots of the results from URC for α_1, which is taken as an example because the other α_j, $j = 2, 3$, do not differ much. This shows, that URC does select values for α_j as optimal, which are near to a value considered sensible from a theoretical point of view.

	Number of variables			
	6	10	20	40
Friedman RDA	0.06 (0.03)	0.05 (0.02)	0.14 (0.04)	0.18 (0.05)
δ	0.92 (0.024)	0.86 (0.030)	0.72 (0.038)	0.76 (0.036)
γ	0.71 (0.036)	0.66 (0.036)	0.70 (0.029)	0.79 (0.023)
own RDA	0.06 (0.02)	0.10 (0.03)	0.14 (0.04)	0.16 (0.04)
RDA with Sim. Ann.	0.06 (0.03)	0.10 (0.03)	0.14 (0.04)	0.17 (0.05)
LDA	0.07 (0.03)	0.13 (0.04)	0.23 (0.06)	0.48 (0.08)
QDA	0.18 (0.08)	0.41 (0.10)	- (-)	- (-)
# no result	0	56	100	100
URC	0.07 (0.04)	0.13 (0.07)	0.24 (0.15)	0.33 (0.12)
Trafo'-URC	0.16 (0.10)	0.21 (0.14)	0.27 (0.17)	0.49 (0.15)
ARC	0.12 (0.12)	0.29 (0.21)	0.31 (0.20)	0.44 (0.17)
Trafo'-ARC	0.08 (0.07)	0.13 (0.06)	0.26 (0.09)	- (-)
ARC + Sim. Ann.	0.07 (0.03)	0.14 (0.05)	0.27 (0.06)	- (-)

Table 2. Means of prediction error rates from Setting 4 by Friedman; the numbers in parantheses are the standard deviations Trafo'-*RC means, that transformations have been used to built in the restrictions on the parameters

		Number of variables			
		6	10	20	40
1	URC	51 / 83 (90)	55 / 92 (88)	95 / 133 (79)	300 / 249 (23)
	Trafo'-URC	47 / 87 (85)	58 / 96 (86)	90 / 147 (70)	300 / 220 (39)
	ARC	0 / 0 (100)	0 / 0 (100)	184 / 160 (72)	0 / 40 (0)
	Trafo'-ARC	0 / 0 (100)	0 / 0 (100)	152 / 129 (87)	0 / 28 (0)
4	URC	46 / 79 (89)	65 / 102 (85)	83 / 121 (83)	300 / 252 (22)
	Trafo'-URC	48 / 81 (90)	70 / 83 (96)	89 / 130 (81)	300 / 218 (43)
	ARC	0 / 0 (100)	0 / 0 (100)	102 / 109 (72)	0 / 33 (0)
	Trafo'-ARC	0 / 0 (100)	0 / 0 (100)	105 / 107 (79)	0 / 36 (0)

Table 3. Means/medians of iterations for the different procedures and % successful optimizations in parentheses; "*success*"=convergence within 300 iterations

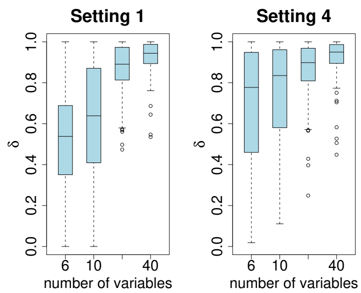

Fig. 1. Shrinkage parameter δ in the Friedman settings for URC

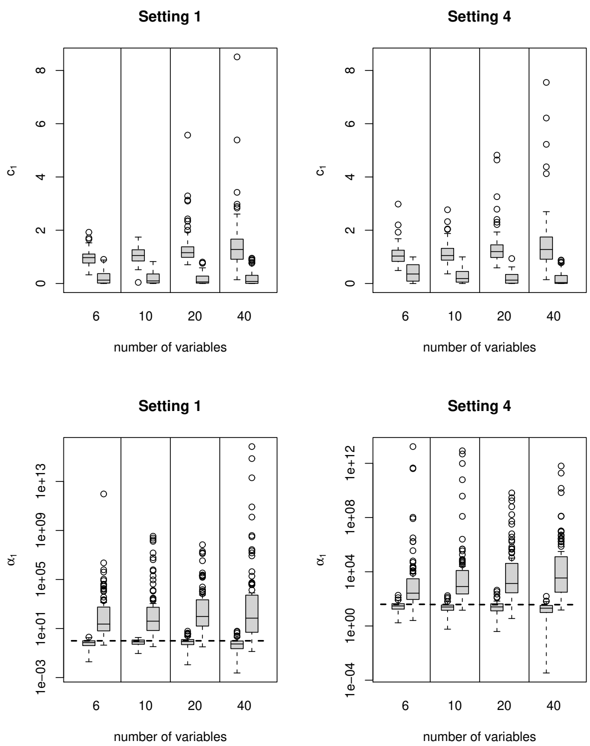

Fig. 2. Parameter c_1 and α_1 in the Friedman settings for URC, depicted in the boxplots on the left hand side of each column; the right hand sides give the values of these parameters for eqivalence to RDA

ARC did not optimize very much in most cases. As can be seen from Figure 3(a) and Table 3 it stops in most cases after the starting step. This means, that all error rates in the start simplex were equal. This was the reason to test whether simulated annealing could help to overcome this minimum. This was not the case as can be seen from Figure 3 (b), where the first 100 steps are due to the simulated annealing. So we will not discuss ARC in further detail here.

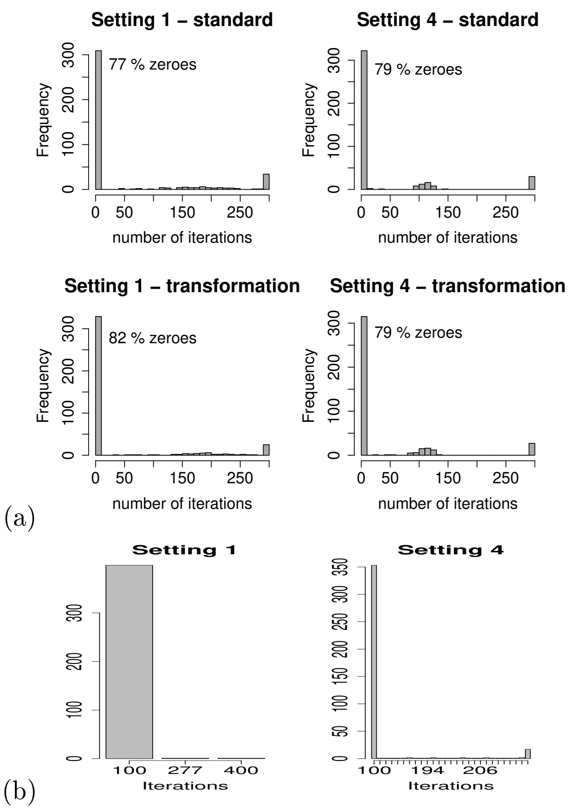

Fig. 3. Iterations for ARC,(a) in the upper row using the restriction, in the lower row using the transformation for the parameters, (b) with simulated annealing

The test of the transformations to implement the restrictions on the parameter space did not improve the results as we had hoped for. The results, where an improvement was encountered could as well be due to chance of a good choice of starting values in ARC, because even in this changed procedure it did select in most cases just one arbitrary setting from the starting simplex, as can be seen from Table 3.

The reason for the difficulties to find an optimal set of parameters can be due to the small number of 40 observations in the learning sets, because this means that different parameter settings will not lead to a change in the estimated error rate.

Finally let us consider our example of truly multi-collinear data. As in the Friedmann-settings our methods are not able to find better classifications than RDA, or LDA, QDA, respectively. It seems that different parameter settings in ARC do not change the misclassification rate significantly. All in all the situations seem to be handled by all procedures similarly.

	Setting		
	1	2	3
LDA	0.25 (0.04)	0.23 (0.04)	0.29 (0.05)
QDA	0.27 (0.05)	0.13 (0.03)	0.33 (0.05)
URC	0.26 (0.06)	0.20 (0.05)	0.54 (0.13)
Trafo'-URC	0.26 (0.06)	0.20 (0.06)	0.50 (0.13)
ARC	0.24 (0.04)	0.23 (0.04)	0.29 (0.04)
Trafo'-ARC	0.25 (0.04)	0.24 (0.05)	0.28 (0.04)
RDA	0.26 (0.04)	0.17 (0.06)	0.35 (0.10)
RDA with Sim. Ann.	0.25 (0.05)	0.15 (0.05)	0.30 (0.05)

Table 4. Means of prediction error rates from simulations with highly collinear data; the numbers in parantheses are the standard deviations Trafo'-*RC means, that transformations have been used to built in the restrictions on the parameters

Again ARC gains good results without any optimization (cp. Table 5), which is rather surprising. One possible interpretation could be, that any shrinkage towards the pooled covariance matrix can improve matters.

	Setting		
	1	2	3
URC	72 / 86 (96)	83 / 127 (80)	80 / 118 (84)
Trafo'-URC	71 / 107 (90)	73 / 107 (87)	66 / 107 (86)
ARC	0 / 0 (100)	0 / 0 (27)	0 / 0 (100)
Trafo'-ARC	0 / 0 (100)	0 / 0 (22)	0 / 0 (100)

Table 5. Means/medians of iterations for successful optimizations for collinear data; "*success*"=convergence within 300 iterations

6 Conclusion

We implemented the unified estimator and the proposed alternative estimator from Næs and Indahl with an enhanced Nelder Mead Simplex algorithm. The problems of restrictions on the parameter space and an integer valued parameter were solved.

In the test situation, which we took from Friedman (1989), it turned out that the procedures tend to regularize in direction of the pooled covariance matrix and therefore behave similar to LDA. Even though the number of unknown parameters is quite high, URC is quite stable in the test setting and chooses parameters which can be compared to theoretical selections of these parameter values.

ARC turned out to be questionable in this implementation, because it returned in most cases some arbitrary values for its parameters. But even these arbitrary parameters seem to improve the error rate compared to QDA in most cases. In the highly collinear data sets it is even better than RDA. In the Diploma-thesis of Pouwels (2001) a slightly different implementation of ARC proved to be better in most cases, but had severe drawbacks in terms of computation time.

For future work on this implementation it is of interest whether a different optimization procedure could improve the selection of the parameters. Another idea would be to introduce a linkage between especially the α's and δ to ensure that an effect of α is not canceled out directly by a high δ. Furthermore, one could estimate $\frac{1}{d}tr(S_j)$ and replace α_j by $\frac{1}{d}tr(S_j(\delta))\alpha_j^*$ and and start with normally distributed α_j^*. Thereby the search algorithm could be enabled to move towards similar values as RDA.

Computations

All computations were carried out on Athlon 1800+ personal computers under Linux using **R**. A package for **R** including ARC and URC is available from www.statistik.uni-dortmund.de/leute.html?name=wtheis_en. The median of the run-times of URC was 11 minutes, for ARC 10 seconds.

Acknowledgements

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 like to thank Claus Weihs for helpful comments and corrections.

References

BOHACHEVSKY, I. O., JOHNSON,M. E. and STEIN, M. L. (1986) Generalized Simulated Annealing for Function Optimization. *Technometrics, 28, 3, 209-217.*

FRIEDMAN, J.H. (1989): Regularized Discriminant Analysis. *Journal of the American Statistical Association, 84, 165–175*

IHAKA, R. and GENTLEMAN, R. (1996): **R**: A Language for Data Analysis and Graphics. *Journal of Computational and Graphical Statistics, 5, number 3, 299–314*

NÆS, T. and INDAHL, U. (1998): A Unified Description of Classical Classification Methods for Multicollinear Data. *Journal of Chemometrics, 12, 205–220.*

POUWELS, B. (2001): Diskriminanzanalyse bei fast-singulären Kovarianzmatrizen. Diplomarbeit, Fachbereich Statistik, Universität Dortmund

RÖHL, M.C., WEIHS, C., and THEIS, W. (2002): Direct Minimization of Error Rates in Multivariate Classification. *Computational Statistics, 17, 29–46*

VENABLES, W.N. and RIPLEY, B.D. (1999): Modern Applied Statistics with S-PLUS. Third Edition. *Springer*