Flexible Discriminant Analysis by Optimal Scoring

Trevor Hastie
Statistics and Data Analysis Research Group
AT&T Bell Laboratories
Murray Hill, New Jersey

Robert Tibshirani
Department of Preventive Medicine and Biostatistics
and Department of Statistics
University of Toronto
Toronto, Ontario

Andreas Buja
Statistics and Data Analysis Research Group
Bellcore
Morristown, New Jersey

Final Draft—10 December, 1993

Abstract

Fisher's linear discriminant analysis is a valuable tool for multi-group classification. With a large number of predictors, one can find a reduced number of discriminant coordinate functions that are "optimal" for separating the groups. With two such functions one can produce a classification map that partitions the reduced space into regions that are identified with group membership, and the decision boundaries are linear. This paper is about richer nonlinear classification schemes.

Linear discriminant analysis is equivalent to multi-response linear regression using optimal scorings to represent the groups. We obtain nonparametric versions of discriminant analysis by replacing linear regression by any nonparametric regression method. In this way, any multi-response regression technique (such as MARS or neural networks) can be post-processed to improve their classification performence.

1 Introduction

Multigroup classification or discrimination is an important problem with applications in many fields. In the generic problem, the outcome of interest G falls into J unordered classes, which for convenience we denote by the set $\mathcal{J} = \{1, 2, 3, \cdots J\}$. We wish to build a rule for predicting the class membership of an item based on p measurements of predictors or features $X \in R^p$. Our training sample consists of the class membership and predictors for N items. Traditional statistical methods for this problem include linear discriminant analysis and multiple logistic regression. Neural network classifiers have become a powerful alternative, with the ability to incorporate a very large number of features in an adaptive nonlinear model. Ripley (1993) gives an informative survey from a statistician's viewpoint. The recent success and popularity of neural networks led us to look for similar methodologies in the statistical literature, but this seems to be a relatively unexplored area. One significant approach is the Classification and Regression Tree (CART) methodology of Breiman *et al* (1984), which is well known to statisticians and is becoming popular in the artificial intelligence community.

There have been a number of recent advances in the nonparametric multiple regression literature. These include projection pursuit regression (Friedman and Stuetzle, 1981), the ACE algorithm (Breiman and Friedman 1985), additive models (Hastie and Tibshirani, 1990), multivariate adaptive regression splines (MARS, Friedman, 1991), Breiman's (1991) Π method, the interaction spline methodology of Wahba (1990), and more recently the hinging-hyperplanes of Breiman (1991a). Neural networks (e.g. Barron and Barron 1988, Lippman, 1989 and Hinton, 1989) can be viewed as yet another approach to nonparametric regression. In this paper we describe methods for multigroup classification that use these tools to generalize linear discriminant analysis.

The foundations for the developments described here can be found in the nonlinear scaling literature, notably Gifi (1981, 1990). Our work was motivated by the unpublished paper by Breiman and Ihaka (1984). Section 6.3 details the connection with their work.

This paper focuses on adaptive classification procedures. A companion paper, "Penalized Discriminant Analysis", gives a more technical basis for some of the procedures described here, and focuses on obtaining smooth, interpretable canonical variates for high dimensional problems such as spectral and image analysis. Both papers rely on the connections between penalized optimal scoring and penalized discriminant analysis. Hereafter we will refer to this companion paper as PDA.

2 Linear discriminant analysis and generalizations

2.1 A review of linear discriminant analysis

Linear discriminant analysis (LDA) is a time-honored, standard tool for classification. A test observation with predictor X_0 is classified to the class with centroid closest to X_0, where distance is measured in the *Mahalanobis* metric using the pooled within group covariance matrix. This procedure can be justified by assuming the predictors have a multivariate Gaussian distribution, with a different mean but common covariance matrix among the classes. The observation is then assigned to the class having the maximum posterior class probability; this results in the rule described above if the class prior probabilities are the same. Variations in the distance thresholds occur if the class prior probabilities are not equal; further generalizations involve using loss functions other than the 0-1 loss implicit here. Two characteristics of this procedure are:

1. all the relevant distance information is contained in the at most $J-1$ dimensional subspace of R^p spanned by the J group centroids;

2. the decision boundaries are linear.

A reduced form of LDA due to Fisher and Rao adds a graphical component to the procedure. One finds the $K < J-1$ dimensional subspace of R^p in which the group centroids are most separated (once again using the Mahalanobis metric confined to this subspace), and then classifies new data to the closest centroid in the reduced space. This leads to a third characteristic:

3. for small K the data can be plotted in the reduced space, giving a graphical representation of the group separation.

Often two or three dimensions are all that is needed, even for large J. Finally, in certain problems, typically having many classes and limited training data, the reduced space can be more stable and yield improved misclassification results on test data.

4. the dimension-reduced model can show better classification performance.

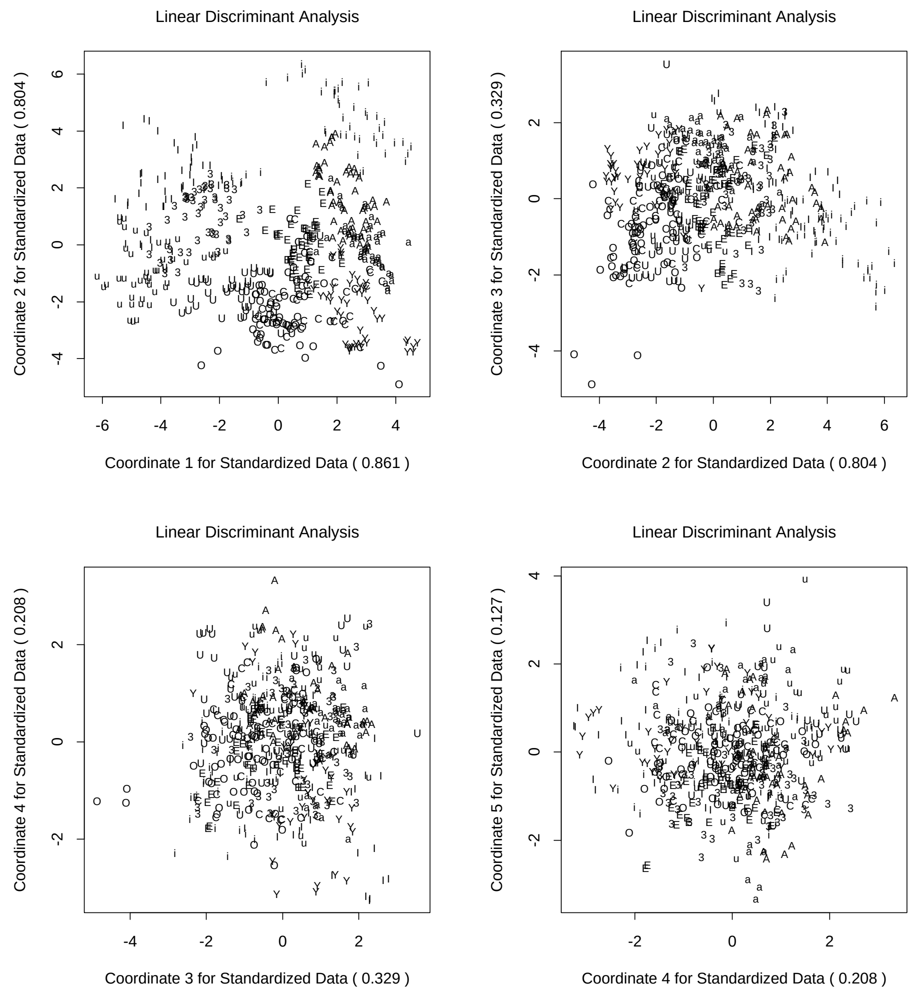

Figure 1: *Some selected 2-dimensional projections of the LDA subspace computed for the vowel data described in section 3.4; the eigenvalues for each coordinate are in parentheses, and indicate the strength of that variable in separating the groups. These plots show visually that most of the discrimination lies in the first few discriminant variables.*

Figure 1 shows several views of the first 5 LDA variates for 10 dimensional speech data, for which there are 11 classes. The 10 predictors are derived from the spectra of the digitized spoken vowels, which fall into 11 classes. The class separation is evident, especially in the plots of the leading LDA variates. These data are described in more detail in section 3.4.

The K dimensional subspace is defined by a $p \times K$ matrix U of LDA vectors u_k, and the derived LDA variables are given by $U^T x$; similarly the transformed centroids are $U^T m_j$ for the jth class. The transformed variables $U^T x$ have identity covariance within groups, and the Mahalanobis distance from x to the jth centroid in this space is simply the Euclidean distance $\delta(x, m_j) = \left\| U^T(x - m_j) \right\|$. In words, we first "sphere" the data using the common within-groups covariance matrix, project these data onto the $J - 1$ dimensional subspace spanned by the J (sphered) centroids (or a subspace thereof), and then classify new observations to the class corresponding to the closest centroid.

In practice the centroids m_j and within-groups covariance matrix Σ_W are unknown and must be estimated from the data. The LDA vectors u_k successively diagonalize the quadratic form $u^T \Sigma_{Bet} u$ with normalization constraint $u^T \Sigma_W u = 1$. Here $\Sigma_{Bet} = \sum_j (N_j/N)(m_j - \bar{m})(m_j - \bar{m})^T$ is the *between groups* covariance matrix. Gnanadesikan and Kettenring (1989) give a recent review of the current literature on discriminant analysis.

2.2 Classification by multivariate linear regression

Multi-response linear regression can also be used for classification. We form an indicator response matrix having J columns, with a value 1 in the jth column if the observation is in class j, and 0 otherwise. Then we carry out a multivariate linear regression of the response matrix on the predictors, and classify each observation by the class having the largest fitted value $\hat{Y}_1, \hat{Y}_2 \dots \hat{Y}_J$. Although $\hat{Y}_j$ can be thought of as an estimate of the class-j probability, there is no guarantee that it lies in $[0, 1]$, nor in general that the J estimates sum to 1. For this reason a modified estimate $\hat{p}_j = \exp(\hat{Y}_j) / \sum_1^J \exp(\hat{Y}_k)$ is often preferred. In the neural network literature, this is known as *softmax* (Bridle, 1990) and we will use that terminology here. In the two group case, with equal sample sizes, *softmax* is essentially equivalent to LDA. They are not equivalent in general, and *softmax* does not seem to work as well as linear discriminant analysis in the examples we have tried. For example, in the vowel example of Figure 1, LDA had test error rate of 56% while *softmax* had test error rate of 67%. In Section 5 we shed some light on the reason for this difference.

We prove in the appendix that the space of linear discriminant fits is the same as the space of multivariate linear regression fits, as defined above. This means that the LDA solution can be obtained from a linear discriminant analysis of the *fitted values* from the multivariate regression fits. This fact is clearly known to some (Breiman and Ihaka 1984 also give a proof, and we discuss the equivalences in detail in PDA), but does not seem to be widely appreciated. Using LDA in this fashion as a post-processor for multivariate linear regression generally improves its classification performance, sometimes dramatically; see section 5 for a detailed discussion of this point

2.3 Generalization of linear discriminant analysis

In practice linear decision boundaries are often too crude, and nonlinear boundaries can be more effective. The Gaussian assumptions are rarely met, and sometimes a group might even be disjoint. Using different class-covariance matrices in the Bayesian procedure results in quadratic decision boundaries. A different approach is to augment the predictor set to include quadratic and bilinear terms, and then perform the LDA in the enlarged space; this also leads to quadratic decision boundaries in the original space, although slightly more restrictive than the former. Besides the fact that a globally-quadratic boundary may not adapt well to the problem at hand, the generalization from linear to quadratic discrimination adds $O(Jp^2)$ parameters to the model; far too many if p is large.

We propose to use nonparametric regression procedures to estimate nonlinear boundaries for classification. To this end, we make use of the well known fact that LDA is equivalent to canonical correlation analysis; the linear predictors define the one set of variables, and a set of dummy variables representing class membership define the other set. Canonical correlation analysis in this context gives the solution to a scoring problem which we now describe.

Suppose $\theta : \mathcal{J} \mapsto R^1$ is a function that assigns scores to the classes, such that the transformed class labels are optimally predicted by linear regression on X. This produces a one dimensional separation between the classes. More generally, we can find K sets of independent scorings for the class labels, $\{\theta_1, \theta_2, \cdots \theta_K\}$, and K corresponding linear maps $\eta_k(X) = X^T\beta_k$, $k = 1, \ldots, K$, chosen to be optimal for multiple regression in R^K. If our training sample has the form (g_i, x_i), $i = 1, 2, \cdots N$, then the scores $\theta_k(g)$

and the maps β_k are chosen to minimize the average squared residual:

$$ASR = \frac{1}{N}\sum_{k=1}^{K}\sum_{i=1}^{N}(\theta_k(g_i) - x_i^T\beta_k)^2 \tag{1}$$

The set of scores are assumed to be mutually orthogonal and normalized with respect to an appropriate inner product to prevent trivial zero solutions.

> It is well known that the sequence of LDA vectors u_k are identical to the sequence β_k up to a constant (Mardia, Kent and Bibby, 1979).

The standard way of carrying out a canonical correlation analysis is by way of a suitable singular value decomposition. Here we outline a somewhat different approach to the solution that lends itself naturally to a nonlinear generalization.

Let Y be the $N \times J$ indicator matrix corresponding to the dummy-variable coding for the classes. That is, the ijth element of Y is 1 if the ith observation falls in class j, and 0 otherwise. Let $\Theta_{J\times K}$, $K \leq J-1$ be a matrix of K score vectors for the J classes. If we let Θ^* be the $N \times K$ matrix of transformed values of the classes with ikth element $\theta_k(g_i)$, then $\Theta^* = Y\Theta$. Looking at (1), it is clear that if the scores were fixed we could minimize ASR by regressing Θ^* on x. If we let P_X project onto the column space of the predictors, this says

$$\begin{aligned} ASR(\Theta) &= \operatorname{tr}\left\{\Theta^{*T}(I-P_X)\Theta^*\right\}/N \\ &= \operatorname{tr}\left\{\Theta^T Y^T(I-P_X)Y\Theta\right\}/N \end{aligned} \tag{2}$$

If we assume the scores have mean zero, unit variance, and are uncorrelated for the N observations ($\Theta^{*T}\Theta^*/N = I_K$), minimizing (2) amounts to finding the K largest eigenvectors Θ of Y^TP_XY with normalization $\Theta^TD_p\Theta = I_K$, where $D_p = Y^TY/N$, a diagonal matrix of the sample class proportions N_j/N. We could do this by constructing the matrix P_X, computing Y^TP_XY, and then calculating its eigenvectors. But a more convenient approach avoids explicit construction of P_X and takes advantage of the fact that P_X computes a linear regression. Here is a summary:

Linear discriminant analysis by optimal scoring: LDA

1. **Initialize:** *form* Y*, the* $N \times J$ *indicator matrix corresponding to the dummy-variable coding for the classes.*

2. **Multivariate regression:** *set* $\hat{Y} = P_X Y$*, and denote the* $p \times J$ *coefficient matrix by* $\mathrm{B}: \hat{Y} = X\mathrm{B}$.

3. **Optimal scores:** *obtain the eigenvector matrix* Θ *of* $Y^T\hat{Y} = Y^T P_X Y$ *with normalization* $\Theta^T D_p \Theta = I$.

4. **Update** *the coefficient matrix in step 2 to reflect the optimal scores:* $\mathrm{B} \leftarrow \mathrm{B}\Theta$*. The final optimally-scaled regression fit is the* $(J-1)$ *vector function* $\eta(x) = \mathrm{B}^T x$.

This is an alternative algorithm for computing the usual canonical variates. The final coefficient matrix B is, up to a diagonal scale matrix, the same as the discriminant analysis coefficient matrix. Specifically, $U^T x = DB^t x = D\eta(x)$ where

$$D_{kk}^2 = \frac{1}{[\alpha_k^2(1-\alpha_k^2)]}, \tag{3}$$

and α_k^2 is the kth largest eigenvalue computed in step 3 above. A derivation is given in the appendix.

Our interest in this procedure is that it provides a starting point for generalizing LDA to a nonparametric version; we replace the linear-projection operator P_X by a nonparametric regression procedure, which we denote by the linear operator S. One simple and effective approach to this end is to expand X into a larger set of basis variables $h(X)$, and then simply use $S = P_{h(X)}$ in place of P_X. The second approach to quadratic discriminant analysis mentioned above is a member of this class, in which case h is a 2nd degree polynomial expansion.

The procedures that we have in mind are more "nonparametric" than this. In this paper we concentrate on two strategies for adaptive nonparametric regression: MARS and BRUTO, both described in detail in section 4. MARS is an adaptive regression technique, able to capture interactions in a hierarchical manner (Friedman, 1991). BRUTO is an adaptive method for estimating an additive model of the form $\alpha + \sum_1^p f_j(X_j)$ using smoothing splines. In terms of basis functions they can be characterized as follows:

- MARS builds up a basis expansion $h(X)$ adaptively, including particular localized interactions only where needed.

- BRUTO automatically generates a very large basis set, and achieves parsimony by shrinking their coefficients in a judicious and structured way.

In both of these procedures, if one conditions on the adaptive choices of terms and/or smoothing parameters, the fitting mechanisms can be represented as the action of a linear operator acting on the response. This is a necessary property for their use here. We denote these choices/smoothing parameters collectively by the *hyper* smoothing parameter λ, and, given λ, the linear smoothing operator by $S(\lambda)$.

The real payoff in expressing the discriminant analysis model in a regression context is the ability to perform model selection and regularization. MARS and BRUTO belong to an ever growing class of adventurous and adaptive regression procedures (including neural networks), which we can simply "plug" into the discriminant analysis problem, and inherit all the adaptivity in the new context.

Before we give details specific to the particular methods, we present our general proposal.

Flexible discriminant analysis by optimal scoring: FDA

1. **Initialize:** *Choose an initial score matrix Θ_0 with $K \leq J$ satisfying the constraints $\Theta^T D_p \Theta = I$; let $\Theta_0^* = Y\Theta_0$.*

2. **Multivariate nonparametric regression:** *Fit a multi-response, adaptive nonparametric regression of Θ_0^* on X, giving fitted values $\hat{\Theta}_0^*$. Let $S(\hat{\lambda})$ be the linear operator that fits the final chosen model, and $\eta(x)$ be the vector of fitted regression functions.*

3. **Optimal scores:** *obtain the eigenvector matrix Φ of ${\Theta_0^*}^T \hat{\Theta}_0^* = {\Theta_0^*}^T S(\hat{\lambda}) \Theta_0^*$, and hence the optimal scores $\Theta = \Theta_0 \Phi$.*

4. **Update** *the final model from step 2 using the optimal scores: $\eta(x) \leftarrow \Phi^T \eta(x)$.*

For a J class problem, the vector of canonical variates or functions $\eta(x)$ has at most $K = J-1$ components. If $\bar{\eta}^j = \sum_{g_i=j} \eta(x_i)/N_j$ denotes the fitted centroid of the jth class in this space of canonical variates, the discrimination rule has the form of a (weighted) nearest centroid rule:

Assign an observation x to the class j that minimizes

$$\delta(x, j) = ||D(\eta(x) - \bar{\eta}^j)||^2 \tag{4}$$

D is once again the diagonal matrix (3) of scale factors that convert optimally scaled fits to discriminant analysis variables.

We have glossed over many details, which we now discuss.

2.4 More details on $S(\lambda)$

$S(\lambda)$ is a linear operator, in this case an $N \times N$ matrix, and λ is the hyperparameter that captures the model selection/smoothing parameter selection required to specify the amount and type of smoothing. $S(\lambda)\Theta^*$ for a $N \times K$ matrix of "responses" Θ^* implies matrix multiplication. In practice we have efficient algorithms for computing $S(\lambda)\Theta^*$ without explicitly computing $S(\lambda)$ itself.

For example, in an additive spline model with p terms, λ_j might be the roughness penalty for the jth term. The "hat" in $S(\hat{\lambda})$ above indicates that we have used adaptive techniques to select the values for λ.

Note that the above description implies that the *same procedure* $S(\lambda)$ is used to fit each of the K models (one for each set of scorings); in particular, the same value for λ is used. Therefore the nonparametric regression procedure must be able to handle a multiple response variable when selecting λ. We discuss the relevant details in the sections on MARS and BRUTO.

Although we have chosen to emphasize MARS and BRUTO in this paper, there are many other possibilities for $S(\lambda)$, including projection pursuit, neural networks and hinging hyperplanes. The theory works for linear operators $S(\lambda)$, and none of the above are strictly linear. Selecting $\hat{\lambda}$ from the training data typically makes $S(\hat{\lambda})$ nonlinear (since $\hat{\lambda}$ is a function of Y). Our theory only applies if we condition on these choices. It does offer some comfort, however, that the technique might be useful as a postprocessor for *any* multivariate regression technique.

2.5 The canonical functions η

The nonparametric procedures produce canonical functions $\eta(x)$, while in the linear case LDA produces discriminant variates $U^T x$. These latter variates are scaled versions of the *linear* canonical functions: $U^T x = D\mathrm{B}^T x$, and hence $D\eta(x)$ are nonparametric versions of the discriminant variates[1]

For both of the nonparametric procedures that we use, the functions $\eta(x)$ are in fact linear combinations of an adaptively chosen set of basis functions $h(x)$:

$$\eta(x) = \mathrm{B}^T h(x).$$

This suggests that given the basis functions (selected in step 2), we can simply perform LDA treating them as variables. When the MARS algorithm or any projection is used for $S(\lambda)$, this is exactly the case: the solutions resulting from step 4 are exactly a linear discriminant analysis using the basis variables $h(x)$. Similarly, we could instead perform LDA using the fitted values $\hat{Y}_j$ from the MARS or projection fit as variables (since $P_{\hat{Y}} Y = P_{h(X)} Y$.) Neither of these equivalences apply when regularized (shrunken) regression procedures such as BRUTO are used; in these cases the FDA procedure solves a form of *penalized discriminant analysis* described next.

2.6 Penalized discriminant analysis

The idea here is to first expand the predictors into a (much) larger set of basis functions $h(x)$, but then ensure by regularization that the linear combinations $\beta^T h(x)$ are in some sense smooth. To focus ideas we use the additive spline model produced by the BRUTO procedure.

Suppose we use an additive model $\sum f_j(X_j)$ for the scores in step 2 of the flexible discriminant analysis procedure. To formalize the estimation of the f_js, we consider a penalized version of the average squared residual (1):

$$\frac{1}{N}\sum_{k=1}^{K}\sum_{i=1}^{N}\left[\theta_k(g_i) - \sum_{j=1}^{p} f_{jk}(x_{ij})\right]^2 + \sum_{k=1}^{K}\sum_{j=1}^{p}\lambda_j \int f''_{jk}(t)^2 dt \tag{5}$$

where x_{ij} is the measurement on the jth variable for the ith observation and λ_j are smoothing parameters that govern the tradeoff between fit and

[1] We avoid calling them "discriminant functions", a phrase reserved for distance functions.

smoothness. The solution is a finite-dimensional additive-spline model and (5) can be written as

$$\frac{1}{N}\sum_{k=1}^{K}\sum_{i=1}^{N}\left[\theta_k(g_i)-\sum_{j=1}^{p}\beta_{jk}^T h_j(x_{ij})\right]^2+\sum_{k=1}^{K}\sum_{j=1}^{p}\lambda_j\beta_{jk}^T\Omega_j\beta_{jk} \qquad (6)$$

where each h_j a vector of up to N natural-spline basis functions defined on the set $\{x_{1j},\ldots,x_{Nj}\}$, and β_{jk}, $k=1,\ldots,K$ are the corresponding coefficients. Thus the entire basis set $h(X)$ can have as many as Np components, in which case the ultimate block-diagonal penalty matrix $\Omega(\lambda)=\text{diag}(\lambda_1\Omega_1,\ldots,\lambda_p\Omega_p)$ will have dimension $Np\times Np$ and the coefficient matrix B dimension $Np\times K$. In this case the regression operator has the form

$$S(\lambda)=H(H^TH+\Omega(\lambda))^{-1}H^T$$

and the partially optimized criterion (6) reduces to

$$ASR_p(\Theta)=\text{tr}\left\{\Theta^TY^T(I-S(\lambda))Y\Theta\right\}/N. \qquad (7)$$

The BRUTO procedure described in section 4.2 is an efficient algorithm for computing the fit $S(\lambda)\Theta^*$ (for fixed θ_k), and selecting the smoothing parameters λ_j adaptively.

The problem that is solved by minimizing equation (6) corresponds to a form of *penalized* discriminant analysis. Let Σ_{Bet} be the between groups covariance matrix for $h(X)$, and $\Sigma_W+\Omega$ the *penalized* within groups covariance. We define:

> A *Penalized Discriminant Analysis* finds a matrix U to maximize $\text{tr}(U^T\Sigma_{Bet}U)$ subject to the penalized constraint $U^T(\Sigma_W+\Omega)U=I$.

The discrimination rule is given by (4). Without Ω, this procedure is just linear discriminant analysis on $h(X)$. The inclusion of Ω forces smoothness of the resulting solution and prevents overfitting. The procedure can also be interpreted in terms of penalized Mahalanobis distances from class centroids in the augmented space of $h(X)$:

$$\begin{aligned}\delta(x,j) &= (h(x)-\bar{h}^j)^T(\Sigma_W+\Omega)^{-1}(h(x)-\bar{h}^j)\\ &= \|D(\eta(x)-\bar{\eta}^j)\|^2 \qquad (8)\end{aligned}$$

where (8) is the same as (4).

Aside: in some applications we have basis functions h that represent the actual input features, rather than being an expanded set of predictors derived from the input features. For example, the h might be the grayscale level of a pixel in an image. In this case we wish the *coefficients themselves* to form a smooth image (neighboring pixels carry similar and highly correlated information.) A procedure similar to that described above leads to a ridge-regression type estimate for the β_js. This is the central topic of the PDA paper and will not be discussed further here.

2.7 Initial values for Θ

In the LDA algorithm on page 8 there was no initialization, while for the nonparametric version we have used an initial Θ_0. We first summarize the issues before we get into details:

- $\Theta_0 = I_J$ is a valid choice, and thus amounts to using $\Theta_0^* = Y$ as in LDA.
- Any rank $J-1$ Θ_0 of dimension $J \times J-1$, which satisfies the normalization constraints and is orthogonal to 1_J (a J-vector of ones), is equivalent to $\Theta_0 = I$. The only advantage is that the multivariate smoother $S(\lambda)$ has one less column to fit.
- If Θ_0 has $K < J-1$ columns, and we seek a K-dimensional solution, then starting values can be critical, as they can alter the search for λ.

The first claim is obvious. To verify the second claim we first note that $Y 1_J = 1_N$, and for most sensible regression procedures, $S(\lambda)1_N = 1_N$ for any λ. This also implies that 1_J is a trivial eigenvector of $Y^T S(\lambda) Y$. The eigendecomposition is equivariant under a rotation $R = (1_J : \Theta_0)$ which maps Y to $YR = (1_N : Y\Theta_0) = (1_N : \Theta_0^*)$. It is expedient to have the $J \times (J-1)$ contrast matrix Θ_0 satisfy the normalization conditions required for the discriminant analysis: $\Theta_0^T 1_J = 0$, $\Theta_0^T D_p \Theta_0 = I_{J-1}$. The eigenvalue problem thus reduces to decomposing the $(J-1)\times(J-1)$ matrix $\Theta_0^{*T} S(\hat{\lambda}) \Theta_0^*$.

All that remains is to show that Θ_0 does not affect the selection of λ. Given Θ_0, the *ASR* criterion is

$$\begin{aligned} ASR &= \frac{1}{N}\mathrm{tr}\left\{\Theta_0^T Y^T (I - S(\lambda))^T (I - S(\lambda)) Y \Theta_0\right\} \\ &= \frac{1}{N}\mathrm{tr}\left\{Y^T (I - S(\lambda))^T (I - S(\lambda)) Y \Theta_0 \Theta_0^T\right\} \end{aligned} \tag{9}$$

As we will see below, ASR is the numerator of the GCV criterion used to select λ. The normalization on Θ_0 implies that $\Theta_0\Theta_0^T = D_p^{-1} - 11^T$ in (9), so any legitimate version of Θ_0 will have the same effect on (9). Another interpretation is that *any* rank $J-1$ legitimate initialization amounts to weighting the jth column of Y by $\sqrt{N/N_j}$ and using a weighted ASR criterion in these original dummy variables. This completes the verification of the second claim.

If we seek a $K < J-1$ dimensional solution, then an initial $J \times K$ Θ_0 could make a difference, since it could steer the search for λ in important directions. This suggests iterating the entire procedure in some form to improve the choices of Θ_0. We discuss iteration further in section 6.1.

2.8 Model selection: choosing λ

Nonparametric regression procedures are typically adaptive and require a criterion for model selection; in our terminology we need to choose a value for the hyperparameter λ. We use the generalized cross-validation (GCV) criterion

$$\mathrm{GCV}(c,\lambda) = \frac{ASR(\lambda)}{[1-\{1+c\cdot df(\lambda)\}/N]^2} \tag{10}$$

Here $df(\lambda)$ is the effective degrees of freedom in the model, not counting the constant term. For the MARS model, $df(\lambda)$ is the number of independent basis functions, while for BRUTO it measures the amount of smoothing. In both cases we use $df(\lambda) = \mathrm{tr}S(\lambda) - 1$.

We have introduced an additional hyper-parameter c, which represents the *cost* per degree-of-freedom; this will typically be greater than 1 to account for the adaptive nature of the fitting in both cases.

The rationale is as follows. The GCV criterion is a regression-based criterion, while our real goal is discrimination or classification. Both MARS and BRUTO (and many other nonparametric regression procedures) are geared up to select λ via GCV, so it is expedient to use GCV here as well. By separating out a single cost parameter c, we can use a regression based GCV to select λ *given* c, and then use a more goal-oriented criterion for selecting c. This is also the basis of the *cost-complexity pruning* of Breiman *et al* (1984).

There are several approaches to selecting c. The simplest is to use a fixed value, and based on Friedman (1991), Owen (1991) and our experience here, it seems that reasonable values are 2 for additive models (BRUTO and degree-1 MARS models), and 3 for higher degree MARS models. A more ambitious approach is to choose c to minimize the misclassification cost estimated by m-fold cross-validation. If there is ample data, such sample reuse techniques are not needed, and we can set aside a portion of the data for tuning the procedure.

3 Examples

In the following examples, we compare the new procedures to a number of existing classification methods. The methods include

1. Linear discriminant analysis—LDA; the classical approach using all the variables.

2. Quadratic discriminant analysis—QDA; observations are classified to the class with closest centroid, using Mahalanobis distance based on the class-specific covariance matrix.

3. CART—the Classification and Regression Tree procedure of Breiman *et al.* (1984). Coordinate splits and default input parameter values are used. In some cases we report as well the results using CART with linear combination splits.

4. FDA/BRUTO— flexible discriminant analysis using additive models with adaptive selection of terms and spline smoothing parameters. This is one of the procedures proposed in this paper.

5. FDA/MARS— flexible discriminant analysis using Friedman's (1991) Multivariate Adaptive Regression Splines. This is another proposal of this paper, with an adaptation of the MARS algorithm to allow for multiple responses. We report the *degree* of interaction used; *degree* = 1 implies an additive model.

Softmax refers to use of the maximum fitted value for classification, rather than discriminant analysis. *Softmax* is not reported in two group examples since it agrees with LDA if the class sizes are equal and they are in close agreement in general.

In some cases, neural network procedures are also compared; details are given in the specific example.

3.1 Example: spherical clusters

In the first of these simulated examples there are 10 predictors and 2 classes. The last 6 predictors are noise variables, with standard normal distributions, independent of each other and the class membership. The first four predictors in class 1 are independent standard normal, conditioned on the radius being greater than 3, while the first four predictors in class 2 are independent standard normal without the restriction. The first class almost completely surrounds the second class in the four dimensional subspace of the first four predictors. We chose 250 observations from each class. The error rates of various procedures are shown in Table 1. The optimal decision boundary is the hull of a sphere with radius 3 in the first four coordinates, and has an error rate $P(\chi^2_4 > 9) = 0.061099$.

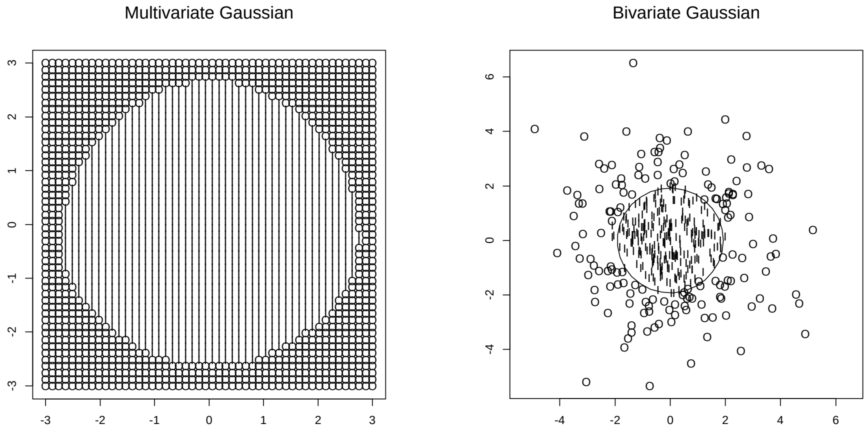

Figure 2: *The left plot shows a view of the decision boundary for the ten-dimensional Gaussian example fit using the BRUTO procedure, obtained by setting coordinates 3-10 to zero. The plot symbols are "I" or "O" for inner and outer, produced over a uniform grid in this two-dimensional slice. The right plot shows the predicted class and the Bayes-optimal decision boundary for the two-dimensional Gaussian example.*

In this case an additive model does the job exactly, since quadratic coordinate functions in the first four variables suffice to describe the spherical hull. Both the BRUTO and MARS (degree = 1) procedure are additive, they both produced terms involving only variables 1–4, the coordinate functions were roughly quadratic, and they both yielded the smallest misclassifica-

Table 1: *Spherical distributions in ten dimensional space with two classes. Six of the variables are noise, while the other four define inner and outer spherical regions that define the classes. The results reported below the horizontal line in the table refer to our flexible discriminant analysis routine using the named nonparametric regression procedure. The values are averages over 10 simulations, with the standard error of the average in parentheses.*

Technique	Error rates	
	Training	Test
LDA	0.439(.006)	0.502(.006)
QDA	0.086(.003)	0.138(.004)
CART	0.056(.002)	0.056(.002)
FDA/BRUTO	0.051(.004)	0.061(.003)
FDA/MARS (degree = 1)	0.052 (.003)	0.065(.004)
FDA/MARS (degree = 2)	0.042 (.003)	0.078(.005)

tion error. The left plot in Figure 2 shows the decision region based on the BRUTO fit for the section of ten-space defined by setting variables 3-10 identically equal to zero.

The right plot shows a decision boundary for a similar but simpler classification problem. Equal sized samples were generated from bivariate gaussian distributions, each with mean zero and diagonal covariance matrices; the one group has variances 4 times that of the other. Bayes classification theory tells us the optimal decision boundary is a circle; the figure shows the optimal boundary, and indicates the classification found using the BRUTO procedure.

3.2 Example: a pure interaction

Here we sampled 200 bivariate uniform variates from $[-1, 1] \times [-1, 1]$. Observations in the south-east and north-west corners were assigned class 1, the others class 2. The results are shown in Table 2.

If CART chooses its first split along $x_1 = 0$ or $x_2 = 0$, it can achieve perfect prediction after its secondary splits. However, in its initial split it is unaware of the nested structure, and hence does not know to split at $x_1 = 0$

Table 2: *Pure interaction example using a sample of 200 observations from a uniform distribution on $[-1,1]\times[-1,1]$, with the two classes defined by the diagonally opposite pairs of quadrants. The values are averages over 10 simulations, with the standard error of the average in parentheses.*

Technique	Error rates	
	Training	Test
LDA	.471(.010)	.485(.013)
QDA	.029(.004)	.038(.005)
CART	.015(.004)	.497(.010)
FDA/BRUTO	.500(.000)	.500(.000)
FDA/MARS (degree = 1)	.491(.009)	.500(.000)
FDA/MARS (degree = 2)	.034(.004)	.050(.007)

or $x_2 = 0$. The FDA procedure using an additive MARS regression (degree = 1) performed poorly, but did very well when interactions were allowed (degree = 2).

3.3 Example: waveform data

This example is taken from Breiman *et al.* (1984 pg 49-55); a concise description is given in Loh and Vanichsetakul (1988). It is a three class problem with 21 variables. The predictors are defined by

$$\begin{aligned} x_i &= uh_1(i) + (1-u)h_2(i) + \epsilon_i \qquad \text{Class 1} \\ x_i &= uh_1(i) + (1-u)h_3(i) + \epsilon_i \qquad \text{Class 2} \\ x_i &= uh_2(i) + (1-u)h_3(i) + \epsilon_i \qquad \text{Class 3} \end{aligned} \tag{11}$$

where $i = 1, 2, \ldots 21$, u is uniform on $(0,1)$, ϵ_i are standard normal variates, and the h_i are the shifted triangular waveforms: $h_1(i) = \max(6-|i-11|, 0)$, $h_2(i) = h_1(i-4)$ and $h_3(i) = h_1(i+4)$.

The training sample has 300 observations, and equal priors were used, so there are roughly 100 observations in each class. We used a test sample of size 500. The results in Table 3 shows that none of the nonlinear methods significantly outperform LDA in terms of misclassification error.

Figure 3 shows the two canonical variates produced in the case of LDA and FDA/BRUTO, evaluated at the test data; the classes are coded in

Table 3: *Results for waveform data. The values are averages over 10 simulations, with the standard error of the average in parentheses*

Technique	Error rates	
	Training	Test
LDA	0.121(.006)	0.191(.006)
Softmax	0.121(.005)	0.194(.007)
QDA	0.039(.004)	0.205(.006)
CART	0.072(.003)	0.289(.004)
FDA/MARS (degree = 1)	0.100(.006)	0.191(.006)
Softmax	0.100(.006)	0.191(.006)
FDA/MARS (degree = 2)	0.068(.004)	0.215(.002)
Softmax	0.071(.003)	0.216(.002)

the plot. The LDA plot suggests that the classes lie on the edges of a triangle. This makes sense, since the $h_j(i)$ are represented by 3 points in 21-space, thereby forming vertices of a triangle. Each class is represented as a convex combination of a pair of vertices, and hence lie on an edge. The clusters in the FDA plot are more rounded than in the former; nonlinear transformations permit the bending of the edges of the triangle, thereby achieving slightly better separation.

The results for CART are in qualitative agreement with those reported by Breiman *et al* (1984)—they report test sample error rates of 0.28 and 0.20 for the univariate-split and linear-combination-split versions respectively. They do not report results for full LDA but report a test error of 0.24 for stepwise LDA.

3.4 Example: vowel recognition data

This example is a popular benchmark for neural network algorithms, and consists of training and test data with 10 predictors and 11 classes. We obtained the data from the benchmark collection maintained by Scott Fahlman at Carnegie Mellon University. The data were contributed by Anthony Robinson (see Robinson, 1989), and he provided the following description.

An ascii approximation to the International Phonetic Association symbol and the word in which the eleven vowel sounds were recorded is given in table

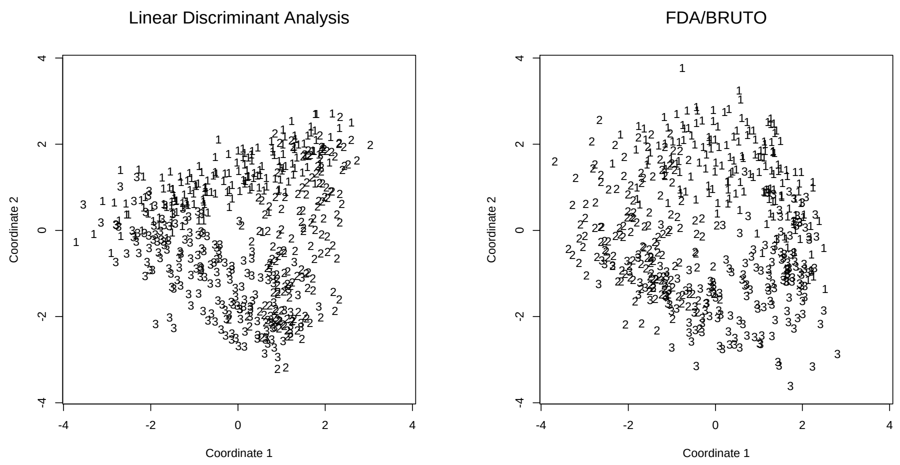

Figure 3: *The left plot shows the canonical variates derived from LDA for the waveform data, evaluated for the test data; the classes are numerically coded. The right plot is the same, but derived from the FDA/BRUTO model.*

(4). The word was uttered once by each of the fifteen speakers. Four male and four female speakers were used to train the networks, and the other four male and three female speakers were used for testing the performance.

This paragraph is technical, and describes how the analog speech signals were transformed into a ten-dimensional feature vector. The speech signals were low-pass filtered at 4.7kHz and then digitized to 12 bits with a 10kHz sampling rate. Twelfth order linear predictive analysis was carried out on six 512 sample Hamming windowed segments from the steady part of the vowel. The reflection coefficients were used to calculate 10 log-area parameters, giving a 10 dimensional input space. For a general introduction to speech processing and an explanation of this technique see Rabiner and Schafer (1978). Each speaker thus yielded six frames of speech from eleven vowels. This gave 528 frames from the eight speakers used to train the networks and 462 frames from the seven speakers used to test the networks.

The results of several classification procedures are shown in table 5. Lines (5)–(8) were taken from Robinson (1989); the first 3 performed the best among the neural network classifiers reported by Robinson. Lines (9)–(11) show the results of various FDA fits, and (12) using the original Breiman & Ihaka code. The *best reduced dimension* was obtained by evaluating models of all dimensions on the test data. The FDA/MARS procedure with *degree* =

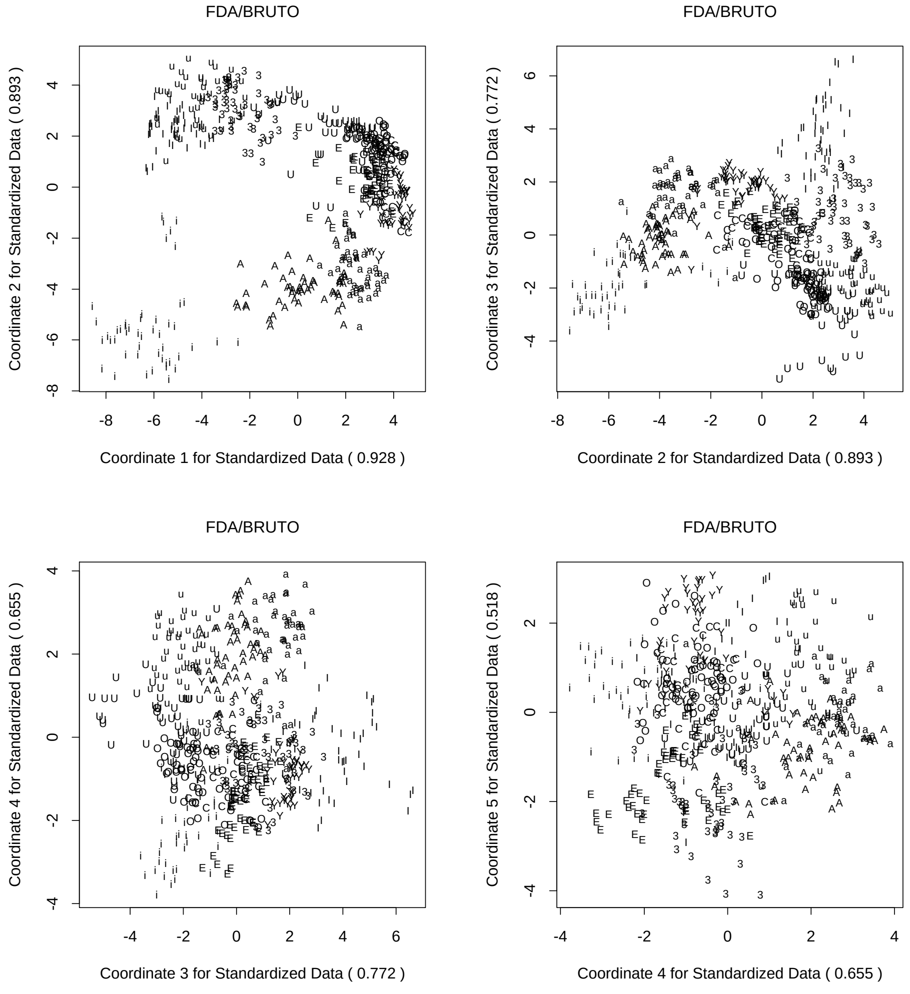

Figure 4: *Some selected 2-dimensional projections of the canonical variates derived from the FDA/BRUTO model fit to the vowel data; the eigenvalues for each coordinate are in parentheses, and indicate the strength of that variable.*

Table 4: *Words used in Recording the Vowels*

vowel	word	vowel	word
i	heed	O	hod
I	hid	C:	hoard
E	head	U	hood
A	had	u:	who'd
a:	hard	3:	heard
Y	hud		

2 and dimension 6 outperforms all of the others. In practice test data might not be available to us, and we would need to use cross-validation or a similar technique to select this parameter.

Plotting the original data reveals that there is a strong person effect, reflected both in mean and variances irrespective of the vowel spoken. We standardized the data for each person so that each variable had mean zero and variance unity, and redid the analysis. The lower part of table 5 reports these improved results. Figure 4 shows some chosen projections of the 10 dimensional space of canonical variates found by the FDA/BRUTO procedure. Some classes stand out more distinctly than the corresponding plots for LDA shown in figure 1.

4 Multi-response adaptive regression procedures

In this section we define the class of multi-response regression procedures that can be used and interpreted in the flexible discriminant analysis proposal of section 2, and give a number of examples. Following that, we provide details on the two particular choices that we focus on in this paper, the MARS and BRUTO procedures.

Suppose we have response variables $Y_1, Y_2, \ldots Y_K$ and a vector of predictors X. The class of regression models for Y that we consider are of the form

$$\eta_1(X) = \beta_{01} + \sum_{m=1}^{M} \beta_{m1} h_m(X)$$

Table 5: *vowel recognition data*

	Technique	Error rates	
		Training	Test
(1)	LDA	0.32	0.56
	Softmax	0.48	0.67
(2)	QDA	0.01	0.53
(3)	CART	0.05	0.56
(4)	CART (linear combination splits)	0.05	0.54
(5)	Single-layer Perceptron		0.67
(6)	Multi-layer Perceptron (88 hidden units)		0.49
(7)	Gaussian Node Network (528 hidden units)		0.45
(8)	Nearest Neighbor		0.44
(9)	FDA/BRUTO	0.06	0.44
	Softmax	0.11	0.50
(10)	FDA/MARS (degree = 1)	0.09	0.45
	Best reduced dimension (=2)	0.18	0.42
	Softmax	0.14	0.48
(11)	FDA/MARS (degree = 2)	0.02	0.42
	Best reduced dimension (=6)	0.13	0.39
	Softmax	0.10	0.50
(12)	Breiman & Ihaka	0.16	0.47
	Using standardized features		
(13)	LDA	0.25	0.36
	Softmax	0.41	0.60
(14)	QDA	0.02	0.53
(15)	CART	0.04	0.68
(16)	Nearest Neighbor		0.46
(17)	FDA/BRUTO	0.10	0.29
(18)	FDA/MARS (degree = 1)	0.11	0.30
	Softmax	0.14	0.38
(19)	FDA/MARS (degree = 2)	0.10	0.39
	Softmax	.09	0.47

$$
\begin{aligned}
\eta_2(X) &= \beta_{02} + \sum_{m=1}^{M} \beta_{m2} h_m(X) \\
\vdots \quad & \quad \vdots \\
\eta_K(X) &= \beta_{0K} + \sum_{m=1}^{M} \beta_{mK} h_m(X)
\end{aligned} \tag{12}
$$

The models for the different responses share a common set of basis functions, but are allowed different coefficients β_{mk}. These coefficients may be found by least squares, or a penalized least squares procedure, or by some other method. Finally, the basis functions $h_m(X)$ will often be chosen adaptively; to obtain the linearity required in the flexible discriminant analysis procedure, the basis functions are treated as fixed after they are selected.

Besides the MARS and BRUTO procedures, this class includes most well known multi-response regression methods. Here are some examples:

- *Multi-response projection pursuit regression* ("SMART"—Friedman, 1984). In this model the K response variables share a common set of ridge functions: $h_m(X) = h_m(\alpha_m^T X)$.

- *Regression trees* (CART, Breiman et al. 1984). A fitted regression tree can be represented by (12), with each h_m an indicator function of a rectangular region of predictor space.

- *Neural Networks.* Consider a multilayered neural network with output functions linear in each of M hidden units in the top layer. The implicit model in such a network is of the form (12) where $h_m(X)$ is the output of the mth unit.

- *Hinge functions.* Breiman (1991a) describes a multiresponse version of his hinge function method that uses a common set of hinge functions for each response.

Operationally any linear smoother $S(\lambda)$ could be used in the FDA procedure, not just penalized expansions onto basis functions as described here. However the latter allows a crisp interpretation as a penalized discriminant analysis, while the former does not. In the remainder of this section we gives details of MARS and BRUTO.

4.1 MARS: multivariate adaptive regression splines

The MARS proposal of Friedman (1991) is a procedure for adaptive, non-parametric regression. It approximates the regression function by a model of the form

$$f(x) = \beta_0 + \sum_{m=1}^{M} \beta_m \prod_{k=1}^{L_m} h_{\ell m}(x_{v(\ell,m)}) \tag{13}$$

Here $x_1, x_2, \ldots x_p$ are predictors, and $v(\ell, m)$ is the index of the predictor used in the ℓth term of the mth product. The basis functions $h_{\ell m}$ are defined in pairs:

$$\begin{aligned} h_{\ell m}(x) &= \ [x - t_{\ell m}]_+ \\ h_{\ell,m+1}(x) &= \ [t_{\ell m} - x]_+ \end{aligned} \tag{14}$$

for m an odd integer, where the knot value $t_{\ell m}$ is one of the unique values of $x_{v(\ell,m)}$. The model (13) is a sum of products of piecewise linear basis functions of the form (14). Denoting a typical model term by $H_j(x)$, the model is built up in a forward stepwise manner as follows:

a) We start with the constant basis function in the model, and define the class of candidate basis functions to be all functions of the form $[x - t]_+$ where x is any predictor and t is a knot placed at any observed value of that predictor.

b) At each stage we consider adding to the model two terms defined by the products $H_j(x)h_{\ell m}(x)$ and $H_j(x)h_{\ell,m+1}(x)$ (m odd). If $H_j(x)h_{\ell m}(x)$ and $H_j(x)h_{\ell,m+1}(x)$ cause the greatest decrease in residual sum of squares, then they are added to the model. The forward stepwise process is stopped when some maximum model size is reached.

c) A backward "pruning" procedure is applied to the model, removing the least important terms one at a time. The best fitting model in the stepwise sequence is chosen, with the fit measured by the generalized cross-validation (GCV) criterion.

The inclusion of each basis function and its conjugate pair in step (b) may cause redundancy in the basis, but allows both members of the pair to be used in subsequent forward steps. The *degree* is a parameter of the procedure, and limits the number of different terms in the product terms

H_j and hence the order of interactions allowed. First degree models are additive, second degree allow pairwise interactions, and so on. Full details of MARS may be found in Friedman (1991). He demonstrates the ability of MARS to capture structure in high-dimensional data. The effectiveness of MARS stems mainly from:

- the piecewise linear basis functions which permit fast updating of the least-squares fit as the knot position is changed, and
- the forward stepwise product strategy, which builds up the surface in small pieces. It takes advantage of any low order structure that may exist and spends its degrees of freedom in a parsimonious manner.

We have generalized MARS for incorporation into the adaptive discriminant procedure. We coded an implementation of MARS from scratch, allowing for multiple response variables. In particular, for K response variables, K simultaneous models are estimated, having the same set of basis functions but different coefficients. The notation in (13) easily accommodates this, if each of the coefficients β_m is a K vector. The models are built and pruned exactly as in Friedman's MARS, the only difference being that the residual sum-of-squares and GCV criterion involve sums over the K response variables.

This generalization did not significantly complicate the implementation. In particular, Friedman's updating formulae still apply, which allow a rapid search for the knot locations.

When MARS is used in the flexible discriminant analysis procedure, the relevant scoring criterion is:

$$ASR = \frac{1}{N}\sum_{k=1}^{K}\sum_{i=1}^{N}(\theta_k(g_i) - \sum_{m=1}^{M}\beta_{mk}\prod_{k=1}^{K_m}h_{\ell m}(x_{v(\ell,m)}))^2$$

4.2 BRUTO: adaptive additive modelling

The model considered here is additive, and therefore more restrictive than the MARS model. On the other hand, it deals with the predictors in a smoother fashion, and can be effective in situations with a large number of predictors.

We introduced the additive model in section 2.6, and for simplicity here we will work with a single response version of (6):

$$\sum_{i=1}^{N}(y_i - \sum_{j=1}^{p}\beta_j^T h_j(x_{ij}))^2 + \sum_{j=1}^{p}\lambda_j\beta_j^T\Omega_j\beta_j. \tag{15}$$

Fortunately (15) can be simplified, and the solution for fixed λ_j can be characterized by a simple set of estimating equations:

$$f_j = S_j(\lambda_j)\{y_i - \sum_{k\neq j} f_k(x_{ik})\} \tag{16}$$

Here the S_j denote smoothing spline operators for smoothing against the jth predictor. These equations lend themselves naturally to an iterative Gauss-Seidel algorithm, also known as backfitting; see Buja *et al* (1989) for further details.

We still need to select which terms to include in the model, the values of λ for those terms selected, and modify the entire procedure to accommodate multiple responses.

For a given set of p variables, we could optimize a criterion such as GCV over all p smoothing parameters λ_j in the p-term additive model. Gu and Wahba (1988) describe an efficient algorithm for doing that; however, their algorithm requires $O(N^3)$ computations, and so a cheaper approximation is attractive. The $O(N)$ algorithm outlined here is described in Hastie (1989); named BRUTO, it combines both backfitting and smoothing parameter selection.

The GCV for an additive model is defined as

$$GCV(\lambda) = \frac{\sum_{i=1}^{N}\{y_i - \sum_j \hat{f}_{j,\lambda_j}(x_{ij})\}^2}{N\{1 - \operatorname{tr} S(\lambda)/N\}^2}. \tag{17}$$

$S(\lambda)$ is the *additive* operator for the given values of the smoothing parameters, and the terms $\hat{f}_{j,\lambda_j}$ denote the fitted functions corresponding to these parameters. Computation of the $\operatorname{tr}(S(\lambda))$ in the denominator requires $O(N^3)$ operations. BRUTO attempts to minimize the GCV-like statistic:

$$GCV^B(c,\lambda) = \frac{\sum_{i=1}^{N}\{y_i - \sum_j \hat{f}_{j,\lambda_j}(x_{ij})\}^2}{N\left(1 - [1 + c\cdot\sum_1^p df_j]/N\right)^2} \tag{18}$$

where $df_j = \operatorname{tr} S_j(\lambda_j) - 1$ measures of the approximate df used in the jth smooth term and c the cost per degree of freedom as described in section 2.8.

So $GCV^B(1,\lambda)$ approximates $\operatorname{tr} S(\lambda)$ by $1+\sum_j\{\operatorname{tr} S_j(\lambda_j)-1\}$, which requires only $O(N)$ computations.

We can write the criterion (18) in a form that focuses on the jth term:

$$GCV^B(1,\lambda) = \frac{\sum_{i=1}^N \{r_i^{(j)} - f_{j,\lambda_j}(x_{ij})\}^2}{N\left(1 - [df^{(j)} + \operatorname{tr} S_j(\lambda_j)]/N\right)^2} \tag{19}$$

where $r^{(j)} = y_i - \sum_{k\neq j} \hat{f}_k(x_{ij})$ is the partial residual in (16), fixing all the other terms in the model, and likewise $df^{(j)}$ are the degrees of freedom for the other terms, an *offset* in the GCV criterion. This suggests that we can minimize GCV^B iteratively, by focusing on one parameter at a time. Each step simply requires a univariate GCV optimizer, slightly modified to include the degrees of freedom for other terms in the model. Including the cost c adds no additional computational burden.

An important modification to the smoothers allows the null fit (mean) or the linear fit to be candidates. Fitting a term by a constant effectively *removes* it from the model, which already includes a constant. In this way the BRUTO algorithm includes variable selection in its model selection procedure.

A final modification accommodates multiple responses. As shown in (6) multiple responses are accommodated in an efficient manner, using the same basis functions and penalty matrices. Instead of a coefficient vector, we have a coefficient matrix for each term. Similarly the backfitting procedure generalizes, and with an omnibus GCV criterion, the entire procedure generalizes.

5 Linear discriminant analysis versus multiple linear regression

Given a multi-response regression procedure there is a simpler way to perform classification, as mentioned is section 2.2. The regression procedure is applied to an indicator response matrix Y representing the classes, and a new observation is assigned to the class with largest fitted value. We will refer to this procedure as *softmax*, as is done in the neural-network literature. Notice that steps 1 and 2 of the LDA algorithm in section 2 compute $\hat{Y}$, so if we stopped there, the classification rule would be the same as that obtained from the *softmax* rule. Instead we apply steps 3 and 4, and then assign to the closest centroid in the η space. These remaining steps may be regarded as a post-processor for converting the regression procedure into a discriminant analysis, and have the following advantages:

1. Accuracy: we demonstrate below that for 3 or more classes *softmax* can produce ridiculous decision boundaries. It seems the fault lies in the fact that it uses the wrong metric in computing distances, as well as the wrong threshold.

2. Dimension reduction: training data may produce some dimensions with marginal and most likely spurious class separations. Discrimination on test data can benefit from weeding out these weak dimensions (see the vowel data test results in table 5.) Furthermore, the dimension reduction provides an important graphical tool for looking at the classification procedure when the number of classes is large.

3. Normal heuristics: the discriminant classifier is Bayes optimal for normally distributed data (assuming that class centers and shared class covariances are known). In practice the nonparametric canonical variates are linear combinations of many basis functions, and sums of random variables suggest normality.

On the other hand, Bayesian heuristics drive the *softmax* procedure: if the fitted values from the regression approximate the posterior probabilities well, then *softmax* will approximate the optimal Bayes procedure. Similar arguments can be used asymptotically for a consistent nonparametric regression model. It therefore seems that the success of FDA rests on the fact that the posterior probabilities are estimated with error by the regression method. We are currently studying this phenomenon in more detail.

In the preceding data examples, discriminant analysis almost always outperformed *softmax*, sometimes by a large margin. For the remainder of this section we try to gain some insight into the relationship between discriminant analysis and *softmax* after multiple regression. First we view *softmax* as a minimum distance method, then we turn things around and view discriminant analysis as a regression method.

For the remainder of this section, by X we mean either the original predictor matrix, or else some adaptively chosen basis expansion of them. Without loss of generality we assume the columns of X have mean zero. As before m_j is the mean vector of class j, and S_T, S_{Bet} and $S_W = S_T - S_{Bet}$ the total, between- and within-class sample covariance matrices respectively. The prior probability of class j will be denoted by π_j.

5.1 Multiple regression as a distance method

When the class priors are equal, Fisher's discriminant rule assigns to the class closest in Mahalanobis distance:

$$\delta_{LDA}(x, j) = (x - m_j)^T S_W^{-1} (x - m_j) \tag{20}$$

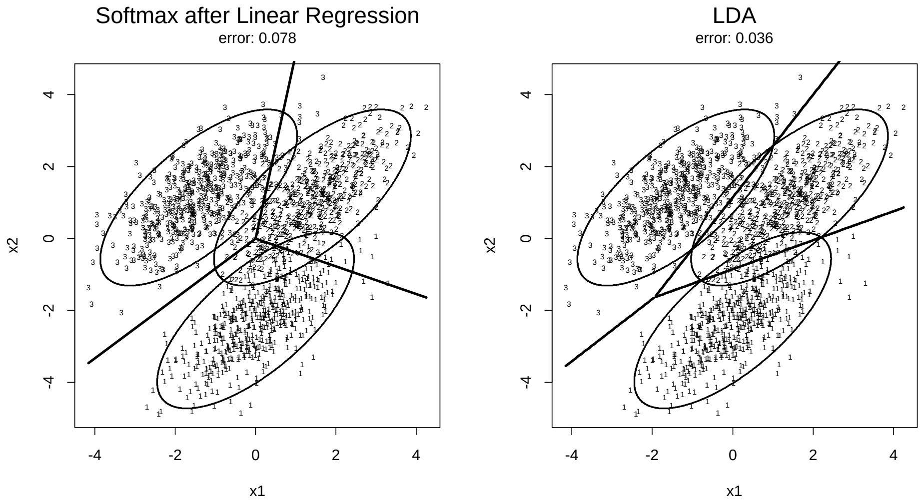

Figure 5: *Softmax uses the wrong metric. The data consist of 500 samples each from 3 bivariate Gaussian distributions, differing in location, but with the same covariance structure. The location of their centroids distorts S_T relative to S_W, and results in a biased decision boundary for* softmax *on the left, while not for LDA on the right.*

We show below that the corresponding *softmax* rule uses the distance:

$$\delta_{Soft}(x, j) = (x - m_j)^T S_T^{-1} (x - m_j) - m_j^T S_T^{-1} m_j \tag{21}$$

They differ in the distance metric and *softmax* has an additional threshold term.

Softmax uses the total covariance matrix to define the metric in computing distances, while LDA uses the within covariance matrix. Since $S_T =$

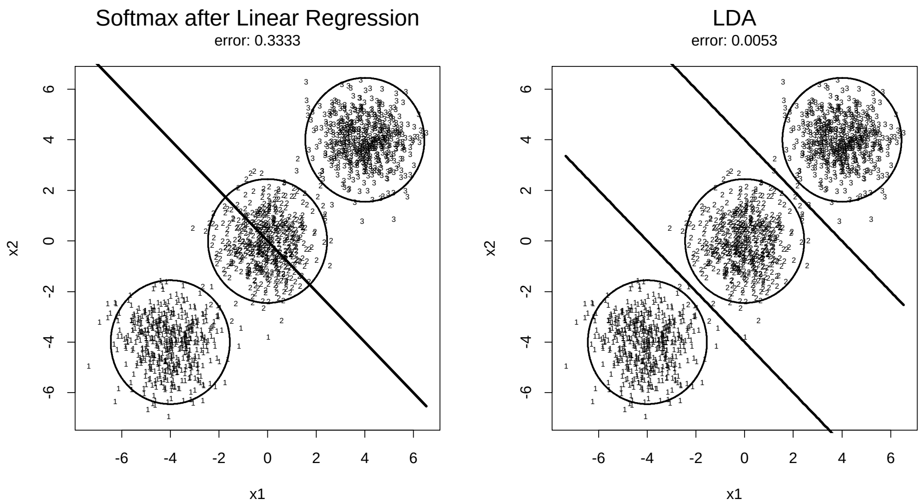

Figure 6: *Softmax uses the wrong threshold. The data consist of 500 samples each from 3 spherical bivariate Gaussian distributions, whose centroids line up along a line. The center class is completely masked by the outside two when* softmax *is used, while LDA has no such problem.*

$S_W + S_{Bet}$, the configuration of the centroids can distort this metric. Figure 5 illustrates this phenomenon with an artificial example, and we see the biased decision boundaries compared to those produced by LDA. This distortion can only occur when there are 3 or more groups. For two groups it can be shown that the metric is the same in both cases, and that only the threshold is different (by an amount $O((\pi_1 - 1/2)^3)$, Ripley, 1993, personal correspondence; see also PDA.)

Softmax has an additional threshold term that also depends on the positions of the class centroids. If the classes are all symmetrically placed about the origin, this term plays no role. A worst-case situation for three classes is illustrated in figure 6, where the class centroids are collinear. The middle class will have centroid zero, while the outer classes not, and each benefit from the threshold correction in (21). (Think of the three dummy-variable regressions for more intuition in this case.)

An opposing argument in favor of *softmax* suggests these phenomena are a result of the constraints imposed by the linear regression model. If the linear regression procedure is consistent for the underlying conditional expectations $p_j(x) = E(Y_j|x)$, then *softmax* is the Bayes-optimal classifier. However, this is unlikely to be the case unless we have sufficient data and thus sufficient basis functions.

We now derive (21). *Softmax* assigns to the class with the largest fitted value

$$R_{Soft}(j,x) = x^T(X^TX)^{-1}X^TY_j + \frac{N_j}{N} \tag{22}$$

This corresponds to minimizing the distance function

$$\delta_{Soft}(x,j) = \frac{N_j}{2N}\left\{(x-m_j)^TS_T^{-1}(x-m_j) - x^TS_T^{-1}x - m_j^TS_T^{-1}m_j - 2\right\} \tag{23}$$

When all the N_j are equal, (23) is equivalent to (21).

5.2 Anti-penalization

Here we view LDA as a regression method. One form of Fisher's linear discriminant rule is to assign to the class with the largest value of

$$R_{LDA}(j,x) = x^TS_W^{-1}m_j - \frac{1}{2}m_j^TS_W^{-1}m_j + \log\pi_j \tag{24}$$

The corresponding multiple regression rule (*softmax*) is to assign to the class with the largest fitted value

$$R_{MR}(j,x) \quad = \quad x^T(X^TX)^{-1}X^TY_j + N_j/N$$

$$= \quad N_j/N(x^T S_T^{-1} m_j + 1) \tag{25}$$

Again to simplify the comparison, we consider the equal N_j situation, and focus on the *class coefficient vectors.* These are up to a constant

$$\begin{aligned} b_j^{LDA} &= S_W^{-1} m_j \ \ \text{(LDA)} \\ b_j^{Soft} &= S_T^{-1} m_j \ \ (\textit{softmax}) \end{aligned}$$

It is not hard to show that these solve the minimization problems

$$\frac{1}{N}||y_j - Xb_j||^2 - b^T S_{Bet} b \qquad \text{(LDA)} \tag{26}$$

$$\frac{1}{N}||y_j - Xb_j||^2 \qquad (\textit{softmax}) \tag{27}$$

Since the penalty term in (26) is subtracted rather than added, we call this *anti-penalization.* These equations offer some further insight into the comparison of LDA and *softmax.* In *softmax*, the direction b_j is chosen to separate out class j from the rest. In contrast, the extra term in the anti-penalization criterion (26) rewards directions b_j that also separate out all of the J classes.

This comparison has some caveats:

- When the N_j are not equal, this interpretation gets a bit murky.
- We have ignored the constant terms, which could be included in both criteria (without penalty in (26).) The *anti-penalized* criterion produces a constant different from that in (24)—in fact, different essentially by the amount $\frac{1}{2} m_j^T S_W^{-1} m_j$. A similar term showed up as crucial in the second example in the previous section.

5.3 Other postprocessors

There are other "postprocessors" that have been proposed, such as nearest neighbor classification in the J dimensional space. For this and other approaches in the context of neural networks see Denker and LeCun (1990). Another approach is to use the basis functions from step 2 in a multinomial regression. Results in standard (linear) logistic and multinomial regression (Efron 1975, Bull and Donner 1987) suggest that this might be more robust to non-normality and might improve classification performance. However since the normality refers here to the space of adaptively chosen basis functions (not the original variables), it is not clear how the methods would compare.

6 Further details and discussion

6.1 Iterative FDA

We can see at least two situations where iterating the FDA algorithm could lead to improvements:

- For $K < J-1$ dimensional models, iteration can lead to better choices of Θ_0;
- Ideally we would like that the criterion for choosing λ reflect misclassification errors; in iterative procedure emerges from such considerations.

Low rank models

In section 2.7 we saw that the ASR and hence the GCV criterion are invariant under different initial scores Θ_0 of rank $J-1$, but not necessarily so for lower rank. Sometimes lower rank models, being more parsimonious and hence less variable, can lead to better classification results (e.g. see vowel data results.) One approach is to use a full-rank Θ_0, and then extract a K dimensional subspace (which is what we did for the vowel example.) It is conceivable, however, that the search for a suitable λ might benefit if a more focused ASR criterion is used—one that only depends on the scorings in question. Since these are not known in advance, an iteration is suggested. We are currently exploring such iterative approaches, especially for problems with large J.

Classification criterion

In section 2.8 we saw that our criterion for selecting λ is most conveniently a regression criterion, while our real focus is on classification errors and group separation. The following identity suggests a way of bringing the two goals closer together. The (penalized) Mahalanobis distance from an observation x to the jth group is given by

$$\delta_\lambda(x,j) = \sum_{k=1}^{J} \frac{(\eta_k(x) - \theta_{jk})^2}{1-\alpha_k^2} - 1/p_j + C(h(x)) \tag{28}$$

where θ^j is the jth row of Θ. This identity is due to Breiman and Ihaka (1984); in PDA we derive the more general version above that covers penalized distances.

Mahalanobis distance is one step away from classification, since we assign classes to the closest group. This suggests using a weighted sum-of-squares or *Discriminant Distance* criterion

$$\mathrm{DD}(\lambda) = \sum_{i=1}^{N} \sum_{k=1}^{J-1} \frac{(\eta_k(x_i;\lambda) - \theta_k(g_i;\lambda))^2}{1-\alpha_k^2(\lambda)} \tag{29}$$

as the basis for finding λ. This weighted sum-of-squares criterion does depend on Θ (and α), and suggests an obvious alternating algorithm.

We can only report empirical evaluations of this approach. On all the examples tried the relative class distance was decreased on average over the training data. This resulted in a modest decrease in the average misclassification errors over the training data, although not always for the test data.

6.2 Priors and Posteriors

So far we have ignored the role of class prior probabilities. Priors can play a role in two places in our procedure:

1. the Bayes classification procedure makes use of the priors in minimum cost assignment;
2. priors can be used in estimation to weight the contributions of observations.

With a 0/1 loss function, the Bayes optimal procedure is to classify an observation x to the class with highest posterior probability. For the Gaussian model, the posterior for the jth class is given by

$$\begin{aligned} p_j(x) &\propto \pi_j \exp\left[-(x-m_j)^T \Sigma_W^{-1}(x-m_j)/2\right] \\ &\propto \exp\left[\{-\delta(x,j) - 2\log\pi_j\}/2\right] \end{aligned} \tag{30}$$

where π_j is the prior for class j. So classifying to the class with the minimum discriminant score $\delta(x,j)$ assumes the priors are equal; if the priors are known and unequal, the posteriors can be computed as in (30), or else the distances corrected by subtracting $2\log(\pi_j)$. If the training sample is random, it is natural to estimate the π_j by the empirical priors N_j/N.

Our estimation procedure implicitly weights each class by the sample prior, since each class is represented by that many observations. This is made more explicit in our eigendecomposition, when we use D_p as the normalization metric for the scores. If the sample is stratified and the true priors are known, it might be more appropriate to weight the observations differently to reflect this knowledge, using weights $N\pi_j/N_j$ for observations in the jth class, and replacing D_p by D_π.

6.3 Discussion

Breiman and Ihaka (1984) suggest a procedure like the one described in this paper, fitting an additive model to the columns of Θ_0^* for a specific choice of Θ_0. In their public domain software (available from the `statlib` archive at Carnegie-Mellon University), they fit a separate additive model to each set of scores, using an adaptive procedure based on "supersmoother" (Friedman and Stuetzle 1982). In addition, they apply the adaptive fitting procedure in step 4 as well as step 2. We have chosen to fit a common model to the set of $J-1$ scores simultaneously, and for our procedure refitting in step 4 is not necessary. Our reasons for doing this are:

1. Our procedure has a clear interpretation as a linear (or penalized) linear discriminant analysis on a set of adaptively chosen functions of the predictors; this interpretation is not valid for their procedure.

2. Fitting one model rather than K models requires roughly $1/K$ times as much computer time. While speed considerations should not be of primary concern, it is a factor especially when full cross-validation is used.

3. Fitting K separate models uses up too many degrees of freedom, too quickly

On the other hand, some problems might benefit from the extra flexibility from having a separate model per dimension. Our experience with the separate approach is that it is generally (slightly) inferior in performance (e.g. table 5), and hence does not warrant the extra complexity.

Software

We have written a general procedure in the S language (Becker and Chambers 1988) called `fda` for flexible discriminant analysis. A `method` argument allows the user to specify the multi-response regression method to be used; the default is linear regression and thus Fisher's LDA. The regression method must be provided in a separate S function. For example `method`

`= mars` or `method = bruto` use the MARS and BRUTO methods respectively. The software will be publicly available in the statistics archive at Carnegie-Mellon University (`statlib@lib.stat.cmu.edu`)

Acknowledgements

The authors thank Leo Breiman and Ross Ihaka for a copy of their technical report, and Leo, Jerry Friedman, and Brian Ripley for helpful discussion. We also thank an associate editor and two referees for their helpful and constructive comments. The second author was supported by the Natural Sciences and Engineering Research Council of Canada.

Appendix: Equivalence of Penalized Discriminant Analysis and Penalized Optimal Scoring

Here we present a terse proof of the equivalence; a more thorough treatment is given in PDA. Our proof includes the equivalence of LDA and optimal scoring as a special case, simply be setting the penalty Ω to zero. Let Σ be the sample covariance matrix of $(Y : H)$, with Σ_{ij}, $i, j \in \{1, 2\}$ the obvious partitions. As before we assume the columns of H are centered, and that H represents a basis expansion of the original variables (or in the unpenalized case X itself.) The following connections exist with our previous notation

- $\Sigma_{11} = D_p$, the diagonal matrix of class proportions.
- $\Sigma_{11}^{-1}\Sigma_{12} = M$, the matrix of class centroids.
- $\Sigma_{22} = \Sigma_T$
- $\Sigma_{Bet} = \Sigma_{21}\Sigma_{11}^{-1}\Sigma_{12}$

Consider the *Generalized, Penalized, Singular Value Decomposition* of Σ_{12}:

$$\Sigma_{11}^{-1}\Sigma_{12}(\Sigma_{22} + \Omega/N)^{-1} = \Theta_* D_\alpha B_*^T \tag{31}$$

with normalizations

$$\Theta_*^T \Sigma_{11} \Theta_* = I = B_*^T(\Sigma_{22} + \Omega/N)B_* \tag{32}$$

We now show that modulo constants, these same left and right singular matrices diagonalize both the penalized optimal scoring criterion and the PDA criterion, and hence are optimal for both.

Penalized Optimal Scoring

We start with the partially optimized criterion (7):

$$ASR_p(\Theta) = \mathrm{tr}\left\{\Theta^T Y^T (I - S) Y \Theta\right\}/N.$$

Substituting the appropriate components of Σ, we see that

$$\begin{aligned} ASR_p(\Theta_*) &= \mathrm{tr}\left\{\Theta_*^T(\Sigma_{11} - \Sigma_{12}(\Sigma_{22} + \Omega/N)^{-1}\Sigma_{21})\Theta_*\right\} \\ &= \mathrm{tr}(I - D_\alpha^2) \end{aligned}$$

The penalized regression coefficients are given by

$$B = (\Sigma_{22} + \Omega/N)^{-1}\Sigma_{21}\Theta_* = B_* D_\alpha.$$

Penalized Discriminant Analysis

One statement of the PDA problem is to maximize $\mathrm{tr}\{U^T \Sigma_{Bet} U\}$ subject to the penalized constraint $U^T(\Sigma_W + \Omega/N)U = I$. Now

$$\begin{aligned} B_*^T \Sigma_{Bet} B_* &= B_*^T \Sigma_{21}\Sigma_{11}^{-1}\Sigma_{12} B_* \\ &= D_\alpha^2 \end{aligned}$$

and

$$\begin{aligned} B_*(\Sigma_W + \Omega/N)B_* &= B_*^T(\Sigma_{22} + \Omega/N - \Sigma_{Bet})B_* \\ &= I - D_\alpha^2. \end{aligned}$$

Thus $U = B_*(I - D_\alpha^2)^{-\frac{1}{2}}$ satisfies the constraint and diagonalizes the criterion, and hence is the optimal discriminant coefficient matrix.

Connecting these two results, we see that $U = BD$ where

$$D^{-1} = D_\alpha(I - D_\alpha^2)^{\frac{1}{2}}$$

References

Barron, A.R. and Barron, R.L. (1988). Statistical learning networks: a unifying view. *Computer Sci and Statist: Proc of the 21st interface.*

Becker, R., Chambers, J., and Wilks, A. (1988). *The New S Language.* Wadsworth International Group.

Breiman, L. and Ihaka, R. (1984). Nonlinear discriminant analysis via scaling and ACE. Technical report, Univ. of California, Berkeley.

Breiman, L. (1991a). Hinging hyperplanes for regression, classification and function approximation. Technical Report 324, Univ. of Calif.- Berkeley.

Breiman, L. (1991b). The π-method for estimating multivariate functions from noisy data. *Technometrics*, 33(2):125–160.

Breiman, L. and Friedman, J. (1985). Estimating optimal transformation for multiple regression and correlation (with discussion). *Journal of the American Statistical Association*, 80:580–619.

Breiman, L., Friedman, J.H., Olshen, R., and Stone, C.J. (1984). *Classification and Regression Trees*. Wadsworth.

Bridle, J.S. (1989). Training stochastic model recognition algorithms as networks can lead to maximum mutual information estimation of parameters. *Advances in Neural Information Processing Systems, Touretzky, D. ed.*, 2.

Buja, A., Hastie, T., and Tibshirani, R. (1989). Linear smoothers and additive models (with discussion). *Annals of Statistics*, 17:453–555.

Bull, S. and Donner, A. (1987). The efficiency of multinmial logistic regression compared with multigroup normal discriminant analysis. *J. Amer. Statist. Assoc.*, 82:1118–1122.

Denker, J. S. and Le Cun, Y. (1991). Transforming neural-net output levels to probability distributions. In Lippmann, R., Moody, J., and Touretzky, D., editors, *Advances in Neural Information Processing Systems*, volume 3, Denver, CO. Morgan Kaufman.

Efron, B. (1975). The efficiency of logistic regression compared to normal discriminant analysis. *J. Amer. Statist. Assoc.*, 70:892–898.

Friedman, J. (1987). Exploratory projection pursuit. *Journal of the American Statistical Association*, 82:249–266.

Friedman, J. (1991). Multivariate adaptive regression splines (with discussion). *Annals of Statistics*, 19(1):1–141.

Friedman, J. and Stuetzle, W. (1981). Projection pursuit regression. *Journal of the American Statistical Association*, 76:817–823.

Friedman, J. and Stuetzle, W. (1982). Supersmoother. Technical report, Dept. of Stat., Stanford Univ.

Gifi, A. (1981). Nonlinear multivariate analysis. unpublished manuscript.

Gifi, A. (1990). *Nonlinear Multivariate Analysis*. Wiley, Chichester.

Gnanadesikan, R. and Kettenring, J. (editors) (1989). Discriminant analysis and clustering. *Statistical Science*, 4:34–69.

Gorman, R. and Sejnowski, T. (1988). Analysis of hidden units in a layered network trained to classify sonar targets. *Neural Networks*, 1:75–89.

Gu., C. and Wahba, G. (1984). Minimizing GCV/GML scores with mulitple smoothing parameters via newton's method. Technical Report 847, University of Wisconsin-Madison.

Hastie, T. (1989). Discussion of "Flexible parsimonious smoothing and additive modelling" by Friedman and Silverman. *Technometrics*, 31:3–39.

Hastie, T. and Tibshirani, R. (1985). Discussion of "Projection pursuit" by Huber. *Annals of Statistics*, 13:502–507.

Hastie, T. and Tibshirani, R. (1990). *Generalized Additive Models*. Chapman and Hall.

Hinton, G. (1989). Connectionist learning procedures. *Artificial intelligence*, 40:185–234.

Lippman, R. (1989). Pattern classification using neural networks. *IEEE communications magazine*, 11:47–64.

Loh, W. and Vanichsetakul, N. (1988). Tree structured classification via generalized discriminant analysis. *Journal of the American Statistical Association*, 83:715–728.

Mardia, K., Kent, J., and Bibby, J. (1979). *Multivariate Analysis*. Academic Press.

Owen, A., (1991). Discussion of "Multivariate Adaptive Regression Splines" by Friedman. In (Friedman, 1991), pages 102–112.

Rabiner, L.R. and Schafer, R.W. (1978). *Digital Processing of Speech Signals*. Prentice Hall.

Ripley, B. (1994). Neural networks and related methods for classification. to appear (with discussion).

Robinson, A. (1989). Dynamic error propagation networks. Technical report, PhD. thesis, Elect. Eng., Cambridge University.

Wahba, G. (1990). *Spline Models for Observational Data.* SIAM, Philadelphia.