Parametric vs Nonparametric Models for Discrimination & Classification

John Maindonald

August 29, 2010

Contents

The methods described here have the character of regression models where the outcome is categorical, one of g classes. For example, the `fgl` dataset has measurements of each on nine physical properties, for 214 samples of glass that are classified into six different glass types.

Linear Discriminant Analysis (LDA), which will be discussed first, may be contrasted with the strongly non-parametric random forest method that uses an ensemble of trees. See Maindonald & Braun (2010, Section 11.7).

A good strategy for getting started is to fit a linear discriminant model with main effects only, comparing the accuracy with that from a random forest analysis. If the random forest analysis gives little or no improvement, the linear discriminant model may be hard to better. There is much more that can be said, but this is often a good starting strategy.

See Ripley (1996); Venables and Ripley (2002); Maindonald & Braun (2010, Section 12.2).

1 Linear Methods for Discrimination

Notation and types of model

Observations are rows of a matrix $\mathbf{X}$ with p columns. The vector $\mathbf{x}$ is a row of $\mathbf{X}$, but in column vector form. The outcome is categorical, one of g classes.

Methods discussed here will all use as predictors continuous non-linear functions of the columns of $\mathbf{X}$. There are several mechanisms for such modeling that involve the use of spline basis terms.

As before, observations are rows of a matrix $\mathbf{X}$ with p columns. The vector $\mathbf{x}$, is a row of $\mathbf{X}$, but in column vector form.

The outcome is categorical, one of g classes, where now g may be greater than 2. The matrix $\mathbf{W}$ estimates the within class variance-covariance matrix, while $\mathbf{B}$ estimates the between class variance-covariance matrix. Details of the estimators used are not immediately important. Note however that they may differ somewhat between computer programs.

1.1 `lda()` and `qda()`

The functions that will be used are `lda()` and `qda()`, from the *MASS* package. The function `lda()` implements linear discriminant analysis, while `qda()` implements quadratic discriminant analysis. Quadratic discriminant analysis is an adaptation of linear discriminant analysis to handle data where the variance-covariance matrices of the different classes are markedly different. For $g = 2$ the logistic regression model, fitted using R's `glm()` function, is closely analagous to the linear discriminant model that is fitted using `lda()`. The difference can however be important.

An attractive feature of `lda()` is that the search for a discriminant rule leads to a representation of a subspace of the column space of $\mathbf{X}$ in r-dimensional space. Providing that the rank of $\mathbf{X}$ is at least $g-1$, $r = g-1$. Use of a spectral decomposition leads to r sets of scores, where each set of scores explains a successively smaller (or at least, not larger) proportion of the sum of squares of differences of group means from the overall mean. The r sets of scores can be examined using a pairs plot.

With three groups, two dimensions will account for all the variation. A scatterplot is then a geometrically complete representation of what the analysis has achieved. With larger numbers of groups, it will often happen that a two or at most three dimensions will account for most of the variation.

The plots that it yields are a major part of the appeal of `lda()`. Where `lda()` does not work well, they may hint at what type of alternative method might be preferred. They can be useful for identifying subgroups of the orginal g groups, and for identifying points that may be misclassified

1.1.1 `lda()` and `qda()` – theory

The functions `lda()` and `qda()` in the *MASS* package implement a Bayesian decision theory approach. Points to note are:

- The methodology is implemented within a Bayesian framework. By default, the prior probabilities for the various categories are taken to be the relative frequencies for those categories. The classification rule changes if the frequencies are changed from the default.
- For any given classification rule, the overall accuracy (proportion correctly classified) changes if the prior probabilities are changed.
- For estimating the accuracy for a given target population, the prior probabilities should be the proportions in that population, not the proportions in the sample.

More specifically:

- A prior probability π_c is assigned to the cth class ($c = 1, \ldots g$).
- The density $p(\mathbf{x}|c)$ of $\mathbf{x}$, conditional on the class c, is assumed multivariate normal, i.e., rows of $\mathbf{X}$ are sampled independently from a multivariate normal distribution.
- For linear discrimination, classes are assumed to have a common covariance matrix Σ, or more generally a common $p(\mathbf{x}|c)$. For quadratic discrimination, different $p(\mathbf{x}|c)$ are allowed for different classes.
- Use Bayes' formula to derive $p(c|\mathbf{x})$. The allocation rule that gives the largest expected accuracy chooses the class with maximal $p(c|\mathbf{x})$; this is the Bayes' rule.
- More generally, assign cost L_{ij} to allocating a case of class i to class j, and choose c to minimize $\sum_i L_{ic} p(i|\mathbf{x})$.

Note that `lda()` and `qda()` use the prior weights, if specified, as weights in combining the within class variance-covariance matrices.

Using Bayes' formula

$$\begin{aligned} p(c|\mathbf{x}) &= \frac{\pi_c p(\mathbf{x}|c)}{p(\mathbf{x})} \\ &\propto \pi_c p(\mathbf{x}|c) \end{aligned}$$

The Bayes' rule maximizes $p(c|\mathbf{x})$. For this it is sufficient, for any given $\mathbf{x}$, to maximize

$$\pi_c p(\mathbf{x}|c)$$

or, equivalently, to maximize

$$\log(\pi_c) + \log(p(\mathbf{x}|c))$$

Now assume $p(\mathbf{x}|c)$ is multivariate normal, i.e.,

$$p(\mathbf{x}|c) = (2\pi)^{\frac{p}{2}} \mid \Sigma_c \mid)^{-\frac{1}{2}} exp(-\frac{1}{2} Q_c)$$

where

$$Q_c = (\mathbf{x} - \mu_c)^T \Sigma_c^{-1} (\mathbf{x} - \mu_c)$$

Then

$$\log(\pi_c) + \log(p(\mathbf{x}|c)) = \log(\pi_c) - \frac{1}{2} Q_c + \frac{p}{2} \log(2\pi) - \frac{1}{2} \log(\mid \Sigma_c \mid)$$

Leaving off the $\log(2\pi)$ and multiplying by -2, this is equivalent to minimization of

$$Q_c + \log(\mid \Sigma_c \mid) - 2\log(\pi_c) = (\mathbf{x} - \mu_c)^T \Sigma_c^{-1} (\mathbf{x} - \mu_c) + \log(\mid \Sigma_c \mid) - 2\log(\pi_c)$$

The observation $\mathbf{x}$ is assigned to the group for which

$$(\mathbf{x} - \mu_c)^T \Sigma_c^{-1} (\mathbf{x} - \mu_c) + \log(\mid \Sigma_c \mid) - 2\log(\pi_c)$$

is smallest.

Set $\mu_c = \bar{\mathbf{x}}_c$, and replace $|\,\Sigma_c\,|$ by an estimate $\widehat{\Sigma_c}$.
[Note that the usual estimate of the variance-covariance matrix (or matrices) is positive definite, providing that the same observations are used in calculating all elements in the variance-covariance matrix and $\mathbf{X}$ has no redundant columns.]

Then $\mathbf{x}$ is assigned to the group to which, after adjustments for possible differences in π_c and $|\,\Sigma_c\,|$, the Mahalanobis distance

$$(\mathbf{x} - \bar{\mathbf{x}}_c)^T \widehat{\Sigma}_c^{-1} (\mathbf{x} - \bar{\mathbf{x}}_c)$$

of $\mathbf{x}$ from $\mathbf{x}_c$ is smallest.

If a common variance-covariance matrix $\Sigma_c = \Sigma$ can be assumed, a linear transformation is available to a space in which the Mahalanobis distance becomes a Eucliean distance. Replace $\mathbf{x}$ by

$$\mathbf{z} = (U^T)^{-1}\mathbf{x}$$

and $\bar{\mathbf{x}}_c$ by $\bar{\mathbf{z}}_c = (U^T)^{-1}\bar{\mathbf{x}}_c$ where U is an upper triangular matrix such that $U^T U = \widehat{\Sigma}$. Then

$$(\mathbf{x} - \mu_c)^T \mathbf{W}^{-1} (\mathbf{x} - \mu_c) = (\mathbf{z} - \bar{\mathbf{z}}_c)^T (\mathbf{z} - \bar{\mathbf{z}}_c)$$

which in the new space is the squared Euclidean distance to from $\mathbf{z}$ to $\bar{\mathbf{z}}_c$.

A result of the `lda` calculations is thus to determine, for each observation, a distances from each of the g group means. In general, these means define a hypplane in $g-1$ dimensional space. Three group means define a plane, four group means define a 3-dimensional hyperplane, and so on.

Note: For estimation of the posterior probabilities, the simplest approach is that described above. Thus, replace $p(c|\mathbf{x};\theta)$ by $p(c|\mathbf{x};\hat{\theta})$ for calculation of posterior probabilities (the 'plug-in' rule). Here, θ is the vector of parameters that must be estimated. The functions `predict.lda()` and `predict.qda()` offer the alternative estimate `method="predictive"`, which takes account of uncertainty in $p(c|\mathbf{x};\hat{\theta})$. Note also `method="debiased"`, which may be a reasonable compromise between `method="plugin"` and `method="predictive"`

1.1.2 Canonical discriminant analysis

Here we assume a common variance-covariance matrix. As described above, replace $\mathbf{x}$ by

$$\mathbf{z} = \mathbf{U^{T^{-1}}}\mathbf{x}$$

where $\mathbf{U}$ is an upper triangular matrix such that $\mathbf{U^T U} = \widehat{\Sigma}$. The estimated variance-covariance matrix of $\mathbf{z}$ is then the identity matrix. Observe that

$$\begin{aligned}
\widehat{\mathrm{var}[\mathbf{z}]} &= \mathrm{E}[\mathbf{U^{T^{-1}}}(\mathbf{x}-\boldsymbol{\mu})(\mathbf{x}-\boldsymbol{\mu})^T\mathbf{U}^{-1}] \\
&= \mathbf{U^{T^{-1}}}\mathrm{E}[(\mathbf{x}-\boldsymbol{\mu}(\mathbf{x}-\boldsymbol{\mu})^T]\mathbf{U}^{-1} \\
&= \mathbf{U^{T^{-1}}}\widehat{\Sigma}\mathbf{U}^{-1} \\
&= \mathbf{I}_p, \quad \text{where } \mathbf{I}_p \text{ is the } p \times p \text{ identity matrix.}
\end{aligned}$$

The between classes variance-covariance matrix becomes

$$\tilde{\mathbf{B}} = \mathbf{U^{T^{-1}}}\mathbf{B}\mathbf{U}^{-1}$$

The ratio of between to within class variance of the linear combination $\alpha^T\mathbf{z}$ is then

$$\begin{aligned}
\mathrm{E}[\alpha^T(\mathbf{z}-\boldsymbol{\mu})(\mathbf{z}-\boldsymbol{\mu})^T\alpha &= \frac{\alpha^T\tilde{\mathbf{B}}\alpha}{\tilde{\alpha}^T\tilde{\alpha}} \\
&= \alpha^T\tilde{\mathbf{B}}\alpha, \text{ subject to the constraint } \|\,\alpha\,\| = 1.
\end{aligned}$$

The matrix $\tilde{\mathbf{B}}$ admits the principal components decomposition

$$\tilde{\mathbf{B}} = \lambda_1 \mathbf{u_1} \mathbf{u_1^T} + \lambda_2 \mathbf{u_2} \mathbf{u_2^T} + \ldots + \lambda_r \mathbf{u_r} \mathbf{u_r^T}$$

The choice $\alpha = \mathbf{u_1}$ maximizes the ratio of the between to the within group variance, a fraction λ_1 of the total. The choice $\alpha = \mathbf{u_2}$ accounts for the next largest proportion λ_2, and so on.

The vectors $\mathbf{u}_1, \ldots \mathbf{u}_r$ are known as "linear discriminants" or "canonical variates". Scores, which are conveniently centered about the mean over the data as a whole, are available on each observation for each discriminant. These locate the observations in r-dimensional space, where r is at most $\min(g-1, p)$. A simple rule is to assign observations to the group to which they are nearest, i..e., the distance d_c is smallest in a Euclidean distance sense.

For plotting in two dimensions, one takes the first two sets of discriminant scores. A point $\mathbf{z}_i$ that is represented as

$$\zeta_{i1}\mathbf{u}_1 + \zeta_{i2}\mathbf{u}_2 + ... + \zeta_{ir}\mathbf{u}_r$$

is plotted in two dimensions as (ζ_{i1}, ζ_{i2}), or in three dinsensions as $(\zeta_{i1}, \zeta_{i2}, \zeta_{i3})$. The amounts by which the original columns of $\mathbf{x}_i$ need to be multiplied to give ζ_{i1} are given by the first column of the list element `scaling` in the `lda` object. For ζ_{i2}, the elements are those in the second column, and so on. See the example below.

As variables have been scaled so that within group variance-covariance matrix is the identity, the variance in the transformed space is the same in every direction. An equal scaled plot should therefore be used to plot the scores.

1.1.3 Linear Discriminant Analysis – Fisherian and other

Fisher's linear disciminant analysis was a version of canonical discriminant analysis that used a single discriminant axis. The more general case, where there can be as many as $r = \min(g-1, p)$ discriminant functions, is described here.

The theory underlying `lda()` assigns $\mathbf{x}$ to the class that maximizes the likelihood. This is equivalent to choosing the class c that minimizes $d_c + \log(\pi_c)$, where if the same estimates are used for $\mathbf{W}$ and $\mathbf{B}$, d_c is the distance as defined for Fisherian linear discriminant analysis. Recall that π_c is the prior probability of class c.

The output from `lda()` includes the list element `scaling`, which is a matrix with one row for each column of $\mathbf{X}$ and one column for each discriminant function that is calculated. This gives the discriminant(s) as functions of the values in the matrix $\mathbf{X}$.

There are two ways that one can run `lda()` and/or `qda()`:

- With the argument `CV=TRUE`, leave-one-out cross-validation is used to return a list with components `class` (the class assigned by the cross-validation) and `posterior` (the posterior probabilities).
- For purposes other than leave-one-out cross-validation, use the argument `CV=FALSE`, which is the default.

1.2 Example – analysis of the forensic glass data

The data frame `fgl` in the *MASS* gives 10 measured physical characteristics for each of 214 glass fragments that are classified into 6 different types. As noted above, the data frame `fgl` has 10 measured physical characteristics for each of 214 glass fragments that are classified into 6 different types.

First, fit a linear discriminant analysis, and use leave-one-out cross-validation to check the accuracy, thus:

```
> fglCV.lda <- lda(type ~ ., data=fgl, CV=TRUE)
> tab <- table(fgl$type, fglCV.lda$class)
> ## Confusion matrix
> print(round(apply(tab, 1, function(x)x/sum(x)), digits=3))
```

```
        WinF WinNF   Veh   Con  Tabl  Head
  WinF  0.729 0.237 0.647 0.000 0.111 0.034
  WinNF 0.229 0.684 0.353 0.462 0.222 0.069
  Veh   0.043 0.000 0.000 0.000 0.000 0.000
  Con   0.000 0.039 0.000 0.462 0.000 0.034
  Tabl  0.000 0.026 0.000 0.000 0.556 0.000
  Head  0.000 0.013 0.000 0.077 0.111 0.862
```

Now run the function with `CV=FALSE`, and examine the output:

```
> opt <- options(digits=2)
> fgl.lda <- lda(type ~ ., data=fgl)
> fgl.lda

Call:
lda(type ~ ., data = fgl)

Prior probabilities of groups:
 WinF WinNF   Veh   Con  Tabl  Head
0.327 0.355 0.079 0.061 0.042 0.136

Group means:
          RI Na   Mg  Al Si    K   Ca     Ba    Fe
WinF   0.718 13 3.55 1.2 73 0.45  8.8 0.0127 0.057
WinNF  0.619 13 3.00 1.4 73 0.52  9.1 0.0503 0.080
Veh   -0.036 13 3.54 1.2 72 0.41  8.8 0.0088 0.057
Con    0.928 13 0.77 2.0 72 1.47 10.1 0.1877 0.061
Tabl  -0.544 15 1.31 1.4 73 0.00  9.4 0.0000 0.000
Head  -0.884 14 0.54 2.1 73 0.33  8.5 1.0400 0.013

Coefficients of linear discriminants:
     LD1   LD2  LD3    LD4   LD5
RI  0.31 0.029 0.36  0.247 -0.80
Na  2.38 3.165 0.46  6.924  2.40
Mg  0.74 2.986 1.57  6.850  2.80
Al  3.34 1.725 2.20  6.419  0.94
Si  2.45 3.006 1.70  7.542  0.96
K   1.57 1.862 1.29  8.076  2.82
Ca  1.01 2.373 0.65  6.697  3.71
Ba  2.31 3.443 2.60  6.438  4.41
Fe -0.51 0.217 1.20 -0.045 -1.30

Proportion of trace:
  LD1   LD2   LD3   LD4   LD5
0.815 0.117 0.041 0.016 0.011

> options(opt)
```

Observe that 93% of the information, as measured by the trace, is in the first two discriminants. We can plot scores on these discriminants, one against the other, as in Figure 1:

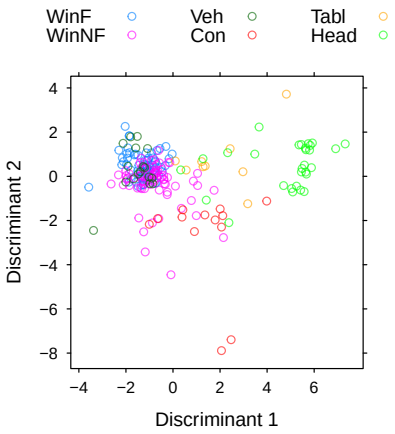

Figure 1: Visual representation of scores derived from *linear discriminant analysis*, for the forensic glass data. A six-dimensional pattern of separation between the categories has been collapsed down to two dimensions. Some categories may therefore be better distinguished than is evident from this figure.

The code for Figure 1 is:

```
> library(lattice)
> scores <- predict(fgl.lda)$x
> gph <- xyplot(scores[,2] ~ scores[,1], groups=fgl$type,
                xlab="Discriminant 1", ylab="Discriminant 2",
                aspect=1, scales=list(tck=0.4), auto.key=list(columns=3),
                par.settings=simpleTheme(alpha=0.6),
                title="Plot shows first two linear discriminant scores")
> print(gph)
```

The discriminant functions

The following demonstrates the use of the information, giving details of the linear discriminant functions, in the component `scaling` of the model object `fgl.lda`:

```
library(MASS)
fgl.lda <- lda(type ~ ., data=fgl)
scores <- predict(fgl.lda, dimen=5)$x  # Default is dimen=2
## Now calculate scores from other output information
checkscores <- model.matrix(fgl.lda)[, -1] %*% fgl.lda$scaling
## Center columns about mean
checkscores <- scale(checkscores, center=TRUE, scale=FALSE)
plot(scores[,1], checkscores[,1])  # Repeat for remaining columns
```

93% of the information, as measured by the trace, is in the first two discriminants.

1.2.1 Two groups – comparison with logistic regression

Logistic regression, which can be handled using R's function `glm()`, is a special case of a Generalized Linear Model (GLM). The approach is to model $p(c|\mathbf{x};\hat{\theta})$ using a parametric model that may be the same logistic model as for linear and quadratic discriminant analysis.

In this context it is convenient to change notation slightly, and give $\mathbf{X}$ an initial column of ones. In the linear model and generalized linear model contexts, $\mathbf{X}$ has the name "model matrix".

The vector $\mathbf{x}$ is a row of $\mathbf{X}$, but in column vector form. Then if π is the probability of membershipÂăin the second group, the model assumes that

$$\log(\pi/(1-\pi) = \beta'\mathbf{x}$$

where β is a constant.

Compare logistic regression with linear discriminant analysis:

- Inference is conditional on the observed $\mathbf{x}$. A model for $p(\mathbf{x}|c)$ is not required. Results are therefore more robust against the distribution $p(\mathbf{x}|c)$.
- Parametric models with "links" other than the logit $f(\pi) = \log(\pi/(1-\pi)$ are available. Where there are sufficient data to check whether one of these other links may be more appropriate, this should be done. Or there may be previous experience with comparable data that suggests use of a link other than the logit.
- Observations can be given prior weights.
- There is no provision to adjust predictions to take account of prior probabilities, though this can be done as an add-on to the analysis.
- The fitting procedure minimizes the deviance, which is twice the difference between the log-likelihood for the model that is fitted and the loglikelihood for a 'saturated' model in which predicted values from the model equal observed values. This does not necessarily maximize predictive accuracy.
- Standard errors and Wald statistics (roughly comparable to t-statistics) are provided for parameter estimates. These are based on approximations that may fail if predicted proportions are close to 0 or 1 and/or the sample size is small.

1.2.2 How important are the linearity assumptions?

The linearity assumptions are restrictive, even allowing for the use of regression spline terms to model non-linear effects. It is not obvious how to choose the appropriate degree for each of a number of terms. The attempt to investigate and allow for interaction effects adds further complications. In order to make progress with the analysis, it may be expedient to rule out any but the most obvious interaction effects. These issues affect regression methods (including GLMs) as well as discriminant methods.

1.2.3 Low-dimensional Graphical Representation

In linear discriminant analysis, discriminant scores in as many dimensions as seem necessary are used to classify the points. These scores can be plotted. Each pair of dimensions gives a two-dimensional projection of the data. If there are three groups and at least two explanatory variables, the two-dimensional plot is a complete summary of the analysis. Even where higher numbers of dimensions are required, two dimensions may capture most of the information. This can be checked.

With most other methods, a low-dimensional representation does not arise so directly from the analysis. An approach that will be demonstrated with random forests, can be adapted for use with other methods.

1.3 A further example – cuckoo egg lengths

To illustrate linear and quadratic discriminant analysis, we will use the data set `cuckoos` (*DAAG* package), in the first instance limiting attention to hedge sparrow and wren nests. This dataset provides measurements on the length and breadth of eggs of each of six host species. Because there are just two measurements, a two-dimensional representation provides a complete description of the results of the analysis. Any plot of scores will be a rotated version of the plot of `length` versus `breadth`.

Examining `cuckoos.lda$scaling`, the entries in the column headed `LD1` are the coefficients of `length` and `breadth` that give the first set of discriminant scores. Those in the column headed `LD2` give the second set of discriminant scores. These scores can be obtained directly from the calculation `predict(cuckoos.lda)$x`

The following uses leave-one-out cross-validation to give an assessments of the accuracy for `lda()`

```
> ## Leave-one-out cross-validation
> ## Accuracies for linear discriminant analysis
> cuckooCV.lda <- lda(species ~ length + breadth,
                      data=cuckoos, CV=TRUE)
> confusion(cuckoos$species, cuckooCV.lda$class,
             gpnames=abbreviate(levels(cuckoos$species), 10))

Overall accuracy = 0.433
This assumes the following prior frequencies:
hedg.sprrw meadow.ppt pied.wagtl      robin tree.pipit       wren
     0.117      0.375      0.125      0.133      0.125      0.125

Confusion matrix
            Predicted (cv)
Actual       hedg.sprrw meadow.ppt pied.wagtl robin tree.pipit  wren
  hedg.sprrw      0.000      0.571      0.143 0.071      0.143 0.071
  meadow.ppt      0.000      0.867      0.067 0.000      0.022 0.044
  pied.wagtl      0.067      0.467      0.200 0.067      0.067 0.133
  robin           0.000      0.625      0.188 0.000      0.062 0.125
  tree.pipit      0.067      0.667      0.200 0.067      0.000 0.000
  wren            0.000      0.267      0.000 0.067      0.000 0.667
```

The following uses leave-one-out cross-validation to give assessments of the accuracy for `qda()`:

```
> ## Accuracies for quadratic discriminant analysis
> cuckooCV.qda <- qda(species ~ length + breadth,
                      data=cuckoos, CV=TRUE)
> acctab <-confusion(cuckoos$species, cuckooCV.qda$class,
                     gpnames=abbreviate(levels(cuckoos$species), 10),
                     printit=FALSE)
> tab <- table(cuckoos$species)
> ##
> ## Overall accuracy
> sum(diag(acctab)*tab)/sum(tab)

[1] 0.425

> ## Confusion matrix
> round(acctab, 3)

            Predicted (cv)
Actual       hedg.sprrw meadow.ppt pied.wagtl robin tree.pipit  wren
  hedg.sprrw      0.214      0.429      0.143 0.071      0.000 0.143
  meadow.ppt      0.000      0.822      0.044 0.000      0.044 0.089
  pied.wagtl      0.067      0.533      0.067 0.067      0.133 0.133
  robin           0.000      0.688      0.188 0.000      0.000 0.125
  tree.pipit      0.200      0.600      0.133 0.067      0.000 0.000
  wren            0.067      0.133      0.000 0.133      0.000 0.667
```

The calculations that follow will require `lda()` and `qda()` fits with `CV=FALSE`, which is the default:

```
> cuckoos.lda <- lda(species ~ length + breadth, data=cuckoos)
> cuckoos.qda <- qda(species ~ length + breadth, data=cuckoos)
```

Figure **??**A plots `length` versus `breadth`, with the axes for the discriminant scores added.

Figure **??**B shows 50% contours for distinguishing `wren` from not-wren, both for the `lda()` analysis (solid line) and for the `qda()` analysis (gray line). The countours are very different. These different

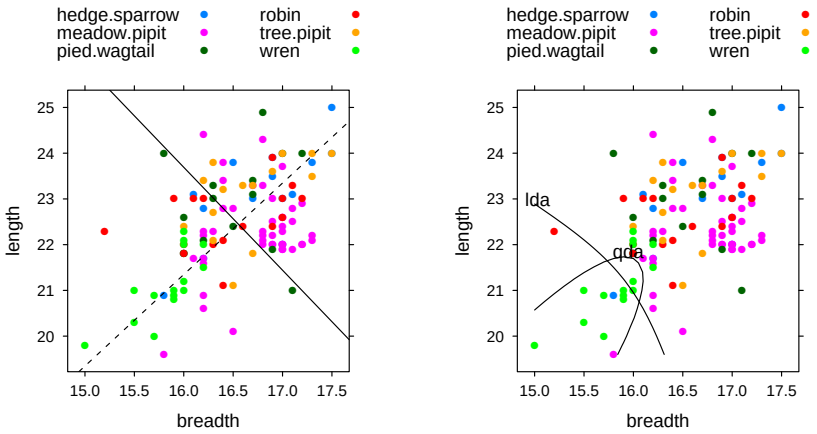

Figure 2: Length versus breadth, compared between cuckoo eggs laid in hedge sparrow and those laid in wren nests. Panel A overlays the axes for the scores. Panel B shows the estimated 50% boundary line for distinguishing `wren` from not-wren.

contours lead in each case to a cross-validated accuracy of 66.7% for correctly predicting wren eggs as wren – a close agreement that may seem surprising.

The following creates a graphics object that plots the points:

```
> gph <- xyplot(length ~ breadth, groups=species, data=cuckoos,
                type=c("p"), auto.key=list(columns=2), aspect=1,
                scales=list(tck=0.5), par.settings=simpleTheme(pch=16))
```

The code for Figure 2A is then:

```
> library(latticeExtra)  # This package has the function layer()
> LDmat <- cuckoos.lda$scaling
> ld1 <- LDmat[,1]
> ld2 <- LDmat[,2]
> library(DAAGxtras)
> gm <- sapply(cuckoos[, c("length", "breadth")], mean)
> av1 <- gm[1] + ld1[2]/ld1[1]*gm[2]
> av2 <- gm[1] + ld2[2]/ld2[1]*gm[2]
> gphA <- gph + layer(panel.abline(av1, -ld1[2]/ld1[1], lty=1),
                      panel.abline(av2, -ld2[2]/ld2[1], lty=2))
```

The code for Figure 2B is:

```
> x <- pretty(cuckoos$breadth, 20)
> y <- pretty(cuckoos$length, 20)
> Xcon <- expand.grid(breadth=x, length=y)
> cucklda.pr <- predict(cuckoos.lda, Xcon)$posterior
> cuckqda.pr <- predict(cuckoos.qda, Xcon)$posterior
> gphB <- gph + as.layer(contourplot(cucklda.pr["wren"] ~ breadth*length,
                                     at=c(0,.5,1), labels=c("", "lda",""),
                                     label.style="flat",
                                     data=Xcon),
                         axes=FALSE) +
                as.layer(contourplot(cuckqda.pr["wren"] ~ breadth*length,
                                     at=c(0,.5,1), labels=c("", "qda",""),
```

```
                                        label.style="flat",
                                        data=Xcon),
                              axes=FALSE)
> gphB
```

For quadratic discriminant analysis, use `qda()` in place of `lda()`.

2 Accuracy comparisons

The function `compareModels()` (*DAAGxtras*) can be used to compare the accuracies of alternative model fits, checking for consistency over the data as a whole. Three model fits will be compared – the `lda()` fit above, the `qda()` fit above, and a variation on the `lda()` fit that includes terms in `length`$\hat{2}$, `breadth`$\hat{2}$ and `length*breadth`

```
> cucklda.pr <- cuckooCV.lda$posterior
> cuckqda.pr <- cuckooCV.qda$posterior
> cucklda.pr2 <- lda(species ~ length + breadth + I(length^2)
                       + I(breadth^2) + I(length*breadth), CV=TRUE,
                       data=cuckoos)$posterior
> compareModels(groups=cuckoos$species,
                  estprobs=list(lda=cucklda.pr, qda=cuckqda.pr,
                                "lda plus"=cucklda.pr2))
```

```
[1] "Average accuracies for groups:"
  WinF  WinNF    Veh    Con   Tabl   Head
0.1703 0.5113 0.1467 0.1497 0.1574 0.5780
Approx sed
    0.0271
[1] "Average accuracies for methods:"
     lda      qda lda plus
  0.3342   0.3402   0.3510
Approx sed
    0.0049
```

3 Tree-based methods and random forests

On a scale in which highly parametric methods lie at one end and highly non-parametric methods at the other, linear discriminant methods lie at the parametric end, and tree-based methods and random forests at the non-parametric extreme. An attraction of tree-based methods and random forests is that model choice can be pretty much automated.

Figure 3 is a visual summary of results from the use of tree-based classification. The three classes are from a clinical classification of Diabetes – `overt` (overt diabetic), `chemical` (chemical diabetic), and `normal` (normal).

The clinical measures (explanatory variables) are `relwt` (relative weight), `fpg` (fasting plasma glucose), `glucArea` (glucose area), `Insulin` (insulin area), and `SSPG` (steady state plasma glucose).

Tree-based classification proceeds by constructing a sequence of decision steps. At each node, the split is used that best separates the data into two groups. Here (Figure 3) tree-based regression does unusually well (CV accuracy = 97.2%), perhaps because it is well designed to reproduce a simple form of sequential decision rule that has been used by the clinicians.

How is 'best' defined? Splits are chosen so that the Gini index of "impurity" is minimized. Other criteria are possible, but this is how `randomForest()` constructs its trees.

3.0.1 Random forests

The random forest methodology will usually improve (but not here), sometimes quite dramatically, on tree-based classification. Figure 4 shows trees that have been fitted to different bootstrap samples of the diabetes data. Typically 500 or more trees are fitted, without a stopping rule. Individual trees

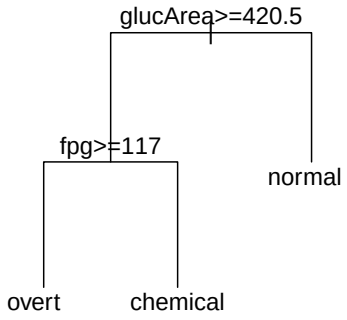

Figure 3: Can the clinical diagnosis be derived directly from the five available clinical measures? The graph shows the classification rule that is given by a tree-based classification.

are likely to overfit. As each tree is for a different random sample of the data, there is no overfitting overall.

Figure 5 is a visual summary of the random forest classification result. The proportion of trees in which any pair of points appear together at the same node may be used as a measure of the "proximity" between that pair of points. Then, subtracting proximity from one to obtain a measure of distance, an ordination method is used to find a representation of those points in a low-dimensional space.

3.1 The `randomForests()` Function

A good first check on the adequacy of "linear" methods in the style of `lda()` and `qda()` adequate is comparison with the highly nonparametric analysis of the function `randomForest()` (*randomForest* package). Random Forests may do well when complex interactions are required to explain the dependence.

The `randomForest()` function can be used in a manner that is highly automatic. There is relatively limited scope for tuning. Such tuning as is possible will often make a very limited improvement.

The random forests methodology takes many (the `randomForest()` default is 500) different bootstrap random samples from the data, each with the same number of observations as the original data. For each such random sample, it takes a random sample of variables, and builds a tree. Splitting of trees usually to the fullest possible extent. The class to which an observation will be assigned is determined by taking a vote between trees.

For each bootstrap sample, predictions can be made for the observations that were not included – i.e., for the out-of-bag data. This is done for each bootsrap sample. Comparison with the actual group assignments then provides an unbiased estimate of accuracy.

In the randomForest() implementation, there is no direct provision for varying prior probabilities from the relative group frequencies. The same effect can however be achieved by varying the sample size (`sampsize` between groups.

4 References

References

Maindonald, J. H. and Braun, W.J. 2010. *Data Analysis and Graphics Using R – An Example-Based Approach.* 3rd edition, Cambridge University Press.
<URL:http://www.maths.anu.edu.au/~johnm/r-book.html>

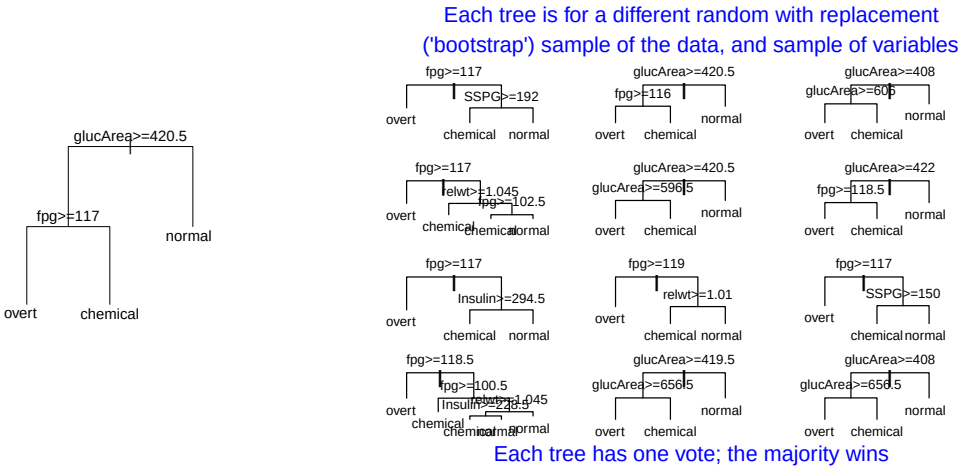

Figure 4: The left panel is a classification tree that was derived from tree-based classification. Each tree in the right panel is for a different bootstrap sample of the diabetes data. Additionally, a different random sample of variables is used for each different tree. The final classification is determined by a random vote over all trees.

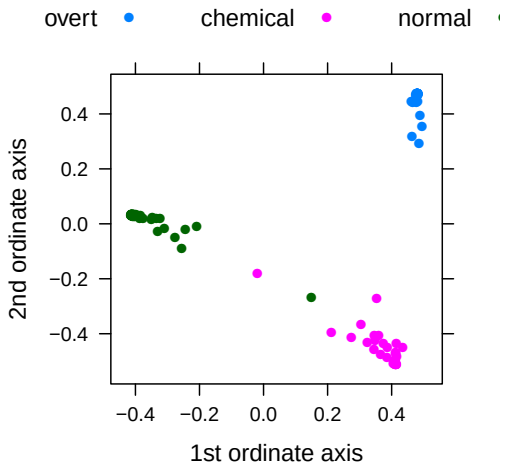

Figure 5: The plot is an attempt to represent, in two dimensions, the random forest result. This plot tries hard to reflect probabilities of group membership assigned by the analysis. It does not result from a 'scaling' of the feature space.

[This is aimed at practicing scientists who have some modest statistical sophistication, and at statistical practitioners. It demonstrates the use of the R system for data analysis and for graphics.]

Ripley, B. D. 1996. *Pattern Recognition and Neural Networks*. Cambridge University Press.

Venables, W. N. and Ripley, B. D. 2002. *Modern Applied Statistics with* S. Springer-Verlag, 4 edition. See also R Complements to Modern Applied Statistics with S.
`http://www.stats.ox.ac.uk/pub/MASS4/`
[This is a wide-ranging account of statistical methods, including statistical learning methods, with details of the S-PLUS and R code required to carry out the computations. Note especially pp.331–341 (lda and qda) and pp.187–198 (logistic and other GLMs).]