Chapter 11

Discrimination and classification

11.1 Introduction

Principal component, correspondence and cluster analysis are examples of *unsupervised learning* methods. Discriminant analysis is, by contrast, an example of a *supervised learning* method; only linear discriminant analysis (LDA) is considered in any detail here. Unsupervised learning methods are geared towards the discovery of structure in data, often in the form of distinct groups. Information may exist about suspected grouping, but this does not feed into the analysis except that it may be used for labeling graphs to aid interpretation (Chapters 7, 9, 10). By contrast, supervised learning methods include information on (suspected) groups in the data which underpins, and is incorporated into, the mathematics of the methods. The aims of analysis may be to confirm that the groups are genuinely distinct; to display group differences graphically; to identify variables that best discriminate betwen groups, or to allocate cases not in the analysis to an appropriate group. This last aim, often called 'classification', motivates much of the more recent methodological development (Hastie *et al.*, 2009).

Mathematically, PCA, CA and LDA, can be viewed in terms of their increasing order of complexity related to the measure of distance used in analysis. Euclidean distance is used in PCA, chi-squared distance distance in CA, and *Mahalanobis distance* in LDA. Mahalanobis distance is the most complex (Sections 11.2 and C.2.3).

The three methods have, in common, the derivation of linear combinations of the original variables, ordered by importance, the first two of which are used for display purposes. Provided $n > p$ (or $I > J$ for CA) there are p (or J) linear combinations that can be derived. In LDA, where G groups are assumed, $(G-1)$ linear combinations can be defined, so that bivariate graphical display is not available when $G = 2$ and univariate display is needed (e.g., Figure 11.6). A brief methodological account of LDA is provided in (Section C.2.3).

11.2 Mahalanobis distance

Mahalanobis distance (MD) has several uses in archaeology other than for discrimination. Mathematical details are provided in Baxter and Buck (2000), Baxter (2003: 69–72) and Section C.2.3. An important feature is that MD takes account of group structure, in particular allowing for the possibility that groups may have an ellipsoidal shape.

11.2.1 MD and confidence ellipsoids

To illustrate, Figure 11.1 reproduces, with enhancement, Figure 6.1 from Baxter (2003) that was based on the analysis of lead isotope ratio data.

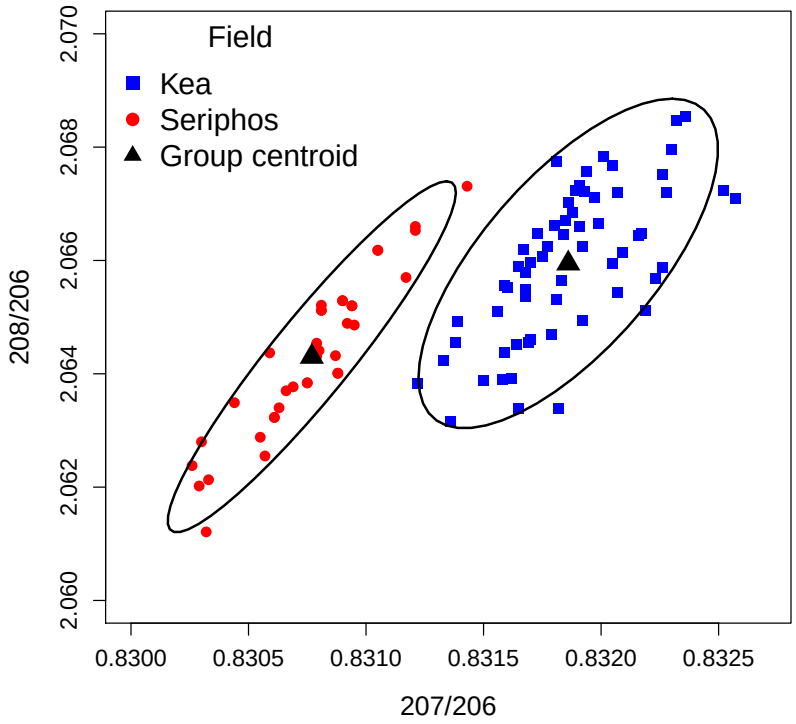

Figure 11.1: *90% confidence ellipsoids for the Kea and Seriphos lead isotope fields using the* $^{208}Pb/^{206}Pb$ *and* $^{207}Pb/^{206}Pb$ *ratios.*

Measures on ore bodies (fields), mined in antiquity for copper, can be characterized by three lead isotope ratios ^{208}Pb/^{206}Pb, ^{207}Pb/^{206}Pb and ^{206}Pb/^{204}Pb. The idea is that different lead isotope fields can be distinguished on the basis of

these ratios. Apart from looking at all pairwise possible plots (e.g., Figure 11.3) they have sometimes been embellished with confidence ellipsoids, as shown. The boundary of an ellipsoid is determined by points equidistant from the centroid in terms of MD (Section C.2.3).

Ellipsoids can be used to delineate groups defined using archaeological criteria, as above, or identified using statistical methods (e.g., PCA or cluster analysis) plotted on pairs of PCs. Figures 6.4 and 6.5 illustrate this for the Pompeiian loomweight data of Tables B.3 and B.4 where this was contrasted with the use of convex hulls in Figure 6.5.

The rationale for using confidence ellipsoids is that the data are samples from populations where the true extent of the field extends beyond that observed. Convex hulls do not allow for this; confidence ellipsoids represent an attempt to estimate the true extent. A potential drawback of their use is that it needs to be assumed that the population has a (multivariate) normal distribution and this is sometimes obviously dubious. The construction of confidence ellipsoids, when the assumption of normality is valid, is discussed in Section C.2.3.

11.2.2 MD, outliers, and allocation to groups

For the Seriphos field there is a case, 33, at the upper extreme of the ellipsoid and lying just outside it. Visually it seems to belong with the Seriphos field, but also seems further away from the centroid of that field than that of Kea. This is confirmed if the Euclidean distance of the case to the two centroids is calculated, after standardization. The distances are 2.33 and 1.20 for Seriphos and Kea.

If MD is calculated the conclusions are reversed and conform more closely with the visual assessment of group assignment. The MDs to the centroids of Seriphos and Kea are 5.76 and 10.29; these are squared quantities so their square-roots of 2.40 and 3.21 may be compared more directly with Euclidean distance. The MD calculations depend on both the centroid of a group and its covariance matrix (Section C.2.3). If the latter is diagonal MD reduces to squared Euclidean distance. Where the data exhibit strong covariances/correlations MD and Euclidean distance calculations can produce rather different results, as shown, because MD but not Euclidean distance allows for the elliptical nature of the groups.

A complication is that MD calculations depend on estimates of the centroid and covariance matrix, the calculation of which is influenced by the case whose membership is being assessed. An obvious idea here is to base calculations on what have been called *leave-one-out* (LOO) methods, where calculations omit the case of interest. Applying this idea to case 33 from the Seriphos field, the square-rooted LOO MD value is 2.66 which exceeds the original value of 2.40, as expected, but is still closer than the distance to the Kea centroid of 3.21 (calculations of which

are unaffected). In R, and in the context of LDA, the term leave-one-out *cross-validation* is used.

11.3 Linear discriminant analysis – examples

11.3.1 Lead isotope-ratio data – three groups

Data for three lead isotope fields are given in Table B.18 for three ratios. With $G = 3$ two linear discriminant functions are defined which lends itself to the display of results in two-dimensional plots. Figure 11.2 contrasts the results of applying PCA and LDA to the three groups using all the ratio data. Remember that LDA uses the information about groups to maximize the separation between them.

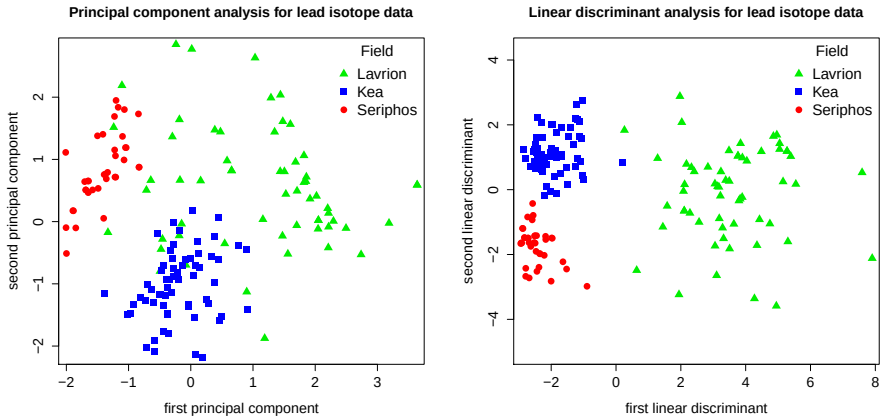

Figure 11.2: *Principal component and linear discriminant analyses of the lead isotope-ratio data of Table B.18 using all three fields.*

From Figure 11.1 it was seen that a plot based on two ratios is more than adequate to establish the fact that the Seriphos and Kea fields are distinct. This is also a feature of the PCA and DA plots. The presence of the Lavrion field complicates matters; it is widely spread out on the PCA plot, showing some overlap with the Seriphos field and more with that for Kea. The LDA, by contrast successfully separates the three fields apart from one case each for the Kea and Lavrion fields whose group membership is in doubt. This is a further clear illustration that PCA and LDA can produce noticeably different results.

In practice, preliminary data inspection is sensible. A pairs plot of the ratios is shown in Figure 11.3. It is clear that all three fields can be separated using just the first two ratios; plots involving the third ratio confuse matters. This is reflected in Figure 11.2 where PCA fails to show the field separation evident from the simple

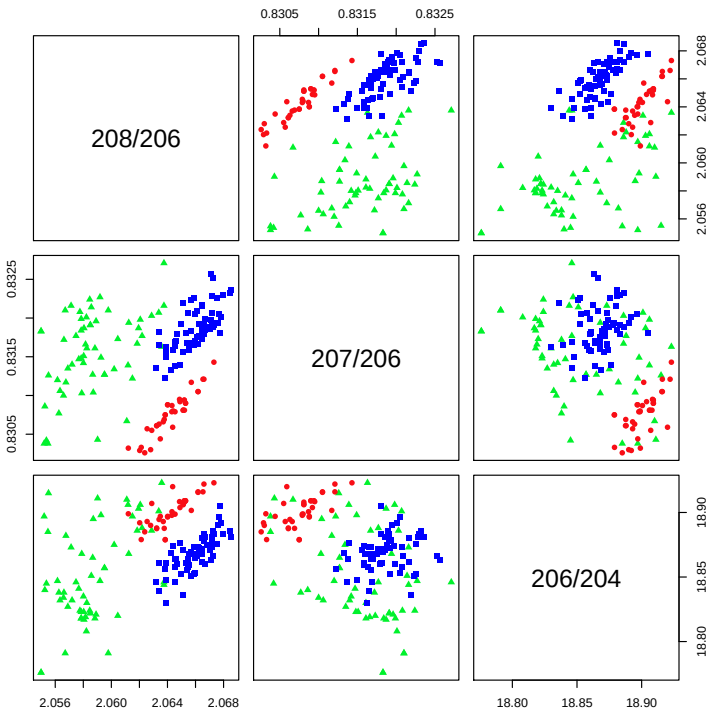

Figure 11.3: *A pairs plot for the lead isotope ratio data. Labeling is as in Figure 11.2, green triangles for Lavrion, blue squares for Kea, red circles for Seriphos.*

analyses. This points to the potential importance of variable selection when using multivariate methods; the problem is that 'non-structure-carrying' variables can obscure the lessons to be learned from variables that are revelatory of structure.

This will not be pursued in detail here; some of the issues are discussed in Baxter (1994b). Examples there show that variable selection can improve the success of allocation, though selection of the best discriminating variables is not guaranteed. With two groups LDA can be formulated in terms of linear regression, with methods based on stepwise selection procedures widely available in software packages. Analogous stepwise procedures for LDA with more than two groups are available in software packages such as `SPSS` and `SAS`. Stepwise selection methods have been subjected to considerable criticism (failure to find 'optimal' solutions; lack of generalizability; etc.). More modern methods are discussed in Chapter 3 of Hastie *et al.* (2009) but have had few archaeological applications.

11.3.2 Neolithic pot dimensions

For a second example data from Table 1 of Madsen (1988b: 18), reproduced in Tables B.19 and B.20, are used. They consist of 16 measurements on eight profile points from pottery vessels from the the Early and earlier Middle Neolithic TRB culture in Denmark.

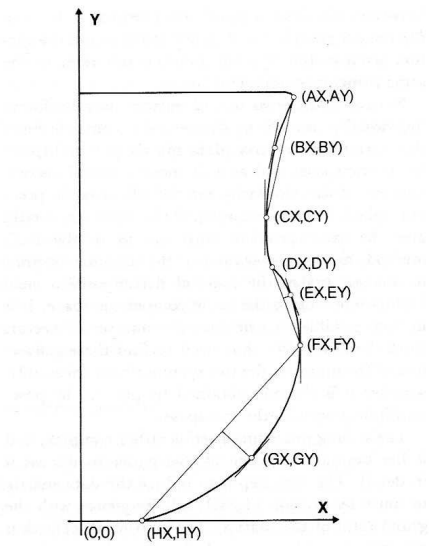

Figure 11.4: *Measurement points of Danish Neolithic pot profiles. (Source: Madsen 1988b: 16, after E.K.Nielsen.)*

Each profile point is represented by vertical and horizontal components, so there are 16 measurements in total (see Figure 11.4). The plots were classified into three vessel forms, funnel beakers, bowls and flasks, with sample sizes of 81, 21 and 16. Classification was based on archaeological criteria and it was of interest to see if this could be reproduced using multivariate methods. It is obvious from simple bivariate plots (e.g., Madsen 1988b: 17) that flasks are dimensionally distinct, and sensible to separate these out at an initial stage of analysis, though they are retained for illustrative purposes in some of the examples to follow.

Madsen (1988b) concentrates on analyses of the funnel-beakers using PCA, whereas we use the data to illustrate aspects of LDA. Initial analysis was dominated by a size component – undesirable in the context of the aims of the analysis. Madsen discusses various ways of removing this and we follow him in scaling vertical measurements to pot height and horizontal measurements to rim width.

Three groups

Figure 11.5 contrasts PCA and LDA analyses of the data for the three types[1]. Both analyses separate out the flasks from the other forms, the separation being much clearer for LDA. Neither analysis separates out the other two forms, though LDA is a bit better. Both analyses suggest a small but distinctive group of six or seven bowls.

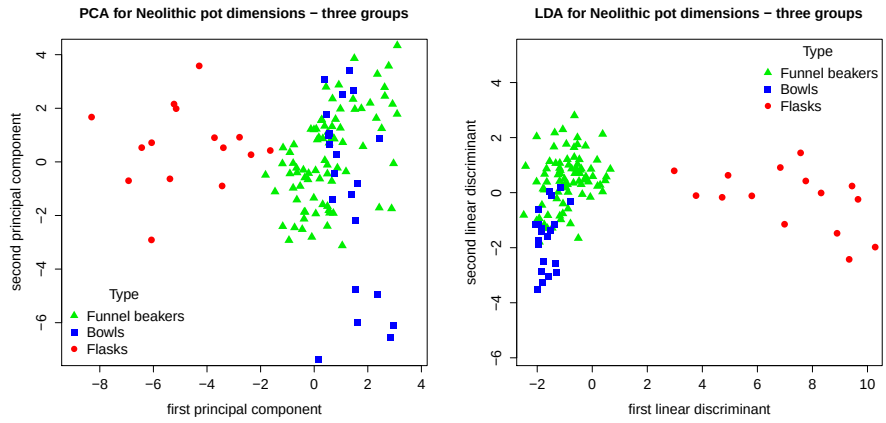

Figure 11.5: *Principal component and linear discriminant analyses of the Neolithic pot dimensions of Tables B.19 and B.20 using all three vessel types.*

Two groups

It is often easier to discern pattern using a small number of groups for any one analysis, and Figure 11.6 repeats the previous analysis after omitting the flasks. There is only one discriminant function, so graphical display must be accomplished using methods other than bivariate plots. Other options than that used, such as boxplots, are available but don't show the individual data points. The message is the same as that derived from the three-group analysis, namely that with the measurements used funnel beakers and bowls are not well discriminated. Using LOO classification 85/102 (83%) of cases are successfully classified. The *resubstitution* method,where statistics are influenced by the case to be classified, produces over-optimistic assessments.

A more informative way of assessing success, though more time-consuming to digest, is to examine the posterior probabilities of group membership, using leave-one-out calculations. These are obtained from the `lda` function in `R` with the

[1]Case 112 (Id. 237 in Table B.20) was found to be a clear outlier in preliminary analysis and omitted from this analysis.

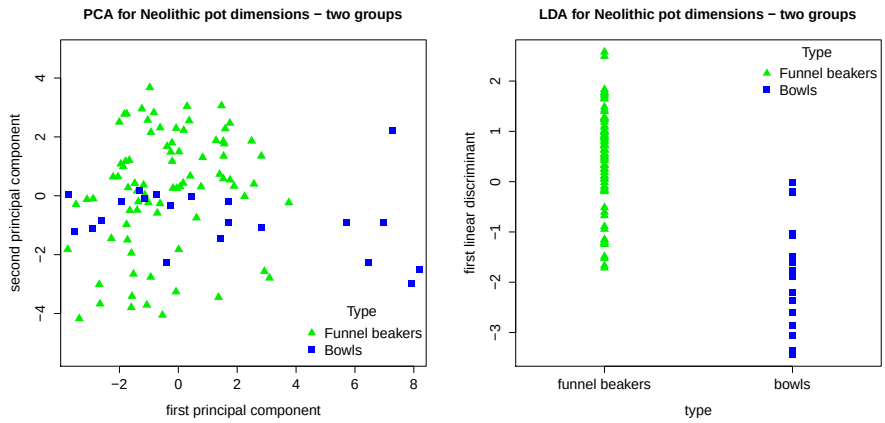

Figure 11.6: *Principal component and linear discriminant analyses of the Neolithic pot dimensions of Tables B.19 and B.20 using the first two types, beakers and bowls.*

argument `CV = TRUE`. The outcome for a subset of the funnel-beaker and bowl data is shown in Table 11.1.

If a 'hard' classification is needed a case would be assigned to the type with highest probability. With the exception of vessels 307, 308, 311 and 329 the funnel-beakers would mostly be convincingly classified as such, with probabilities close to 1. Four beakers are classified as bowls, beaker 307 with a probability of 0.45 of being a beaker and 0.55 of being a bowl only marginally so; the others are more convincing. Of the six bowls shown, only two are classified as such, the remaining four being classified as beakers with high or quite high probabilities.

A limitation of this kind of analysis is that results are presented in terms of relative probabilities and assume that cases must belong to a group in the analysis. This need not be the case; for example outliers may occur, or classes not recognized in the analysis may be represented. In these circumstances a case will be assigned to one of the assumed groups, possibly with high probability, even though the reality is that it belongs to none.

Often this kind of issue will be recognized in preliminary data analysis, or from the LDA itself. More formal methods exist; using normality assumptions MDs can be converted to absolute probabilities that allow a more realistic assessment of likely group membership. This is discussed in Section C.2.3.

			Predictions	
Id.	Actual	Predicted	Beaker	Bowl
⋮	⋮	⋮	⋮	⋮
307	Beaker	Bowl	0.45	0.55
308	Beaker	Bowl	0.25	0.75
311	Beaker	Bowl	0.14	0.86
312	Beaker	Beaker	0.98	0.02
320	Beaker	Beaker	0.93	0.07
321	Beaker	Beaker	0.89	0.11
323	Beaker	Beaker	0.98	0.02
324	Beaker	Beaker	1	0
325	Beaker	Beaker	0.92	0.08
326	Beaker	Beaker	0.95	0.05
327	Beaker	Beaker	0.99	0.01
328	Beaker	Beaker	0.99	0.01
329	Beaker	Bowl	0.33	0.67
336	Beaker	Beaker	0.87	0.13
3	Bowl	Bowl	0.03	0.97
33	Bowl	Beaker	0.77	0.23
59	Bowl	Beaker	0.78	0.22
60	Bowl	Beaker	0.97	0.03
61	Bowl	Bowl	0.01	0.99
131	Bowl	Beaker	0.98	0.02
⋮	⋮	⋮	⋮	⋮

Table 11.1: *Cross-validated estimates of the relative probabilities of belonging to a group, after a two-group LDA for a subset of the Danish Neolithic pot data.*

11.3.3 Practicalities

The normality assumption

The examples touch on a number of practical issues. Among them is the question of normality – it is sometimes incorrectly asserted that LDA requires the assumption that groups have a multivariate normal distribution. In fact Fisher's (1936) original derivation of LDA, subsequently developed by Rao (1948), made no such assumption. A sensible measure of group separation is defined and optimized mathematically to determine the discriminant functions. For descriptive, graphical analysis normality is not assumed. If normality can be assumed then LDA has some optimal properties, but the lack of normality does not necessarily compromise its practical utility. Normality does need to be assumed for probability calculations of the kind described in the previous section and Section C.2.3 but, as some of the examples suggest, it can be questionable.

The equal covariance assumption – quadratic discriminant analysis

Applications so far have assumed that groups are sampled from separate populations having equal covariance matrices, allowing their estimates to be pooled and leading to LDA. If the assumption is relaxed this gives rise to quadratic discriminant analysis (QDA) (Venables and Ripley, 2002: 333–334) which, as the name suggests, leads to quadratic boundaries between groups. A practical drawback of QDA is the need to estimate separate covariance matrices within groups, which requires more parameters and may be problematic with a reasonable number of variables to deal with, so demands more data than LDA.

11.3.4 Example - Steatite compositions

The extended example in this section illustrates the kind of output produced by `R`. It is based on compositional data for samples of steatite (soapstone) analyzed by Truncer *et al.* (1998) in order to see if the method of chemical analysis used succeeded in distinguishing between quarry sources more effectively than had previously been the case. The data are given in Tables B.21 to B.23.

There are six quarry sources used here with sample sizes between 24 and 31; there was a considerable number of measurements below the level of detection and only 6 of the 17 variables that were measured, for which complete information was available, are used. The analysis in the original paper is not emulated, the data are useful for illustrating aspects of LDA in this section, and classification trees in Section 11.4.2. Data are logarithmically transformed, to base 10, before analysis.

The transformation `St.log <- log10(steatite)` is applied to the 159×6 data matrix, `steatite`. The vector of quarry identifiers is named `St.type`. With this in place

```
library(MASS)
St.lda <- lda(St.log, St.type, CV = FALSE)
St.ld <- predict(St.lda)$x
```

sets up an object `St.lda` that can be used for interrogating the analysis, and `St.ld` which is used for plotting purposes. The obvious thing to do, and one of the main purposes of the exercise, is to look at the data graphically using a plot based on the first two functions, as in Figure 11.7[2]. This shows that discrimination is far from perfect. Visually Lawrenceville seems most distinct, but there is overlap between the predictions for all the quarries, particularly Susquehanna.

This can be investigated more closely using the 'confusion' table, which shows the predictions for each quarry. In the `lda` function the argument `CV = FALSE`

[2]This is obtained using `eqscplot(St.ld[,1], St.ld[,2])` where the arguments governing the labeling have been omitted.

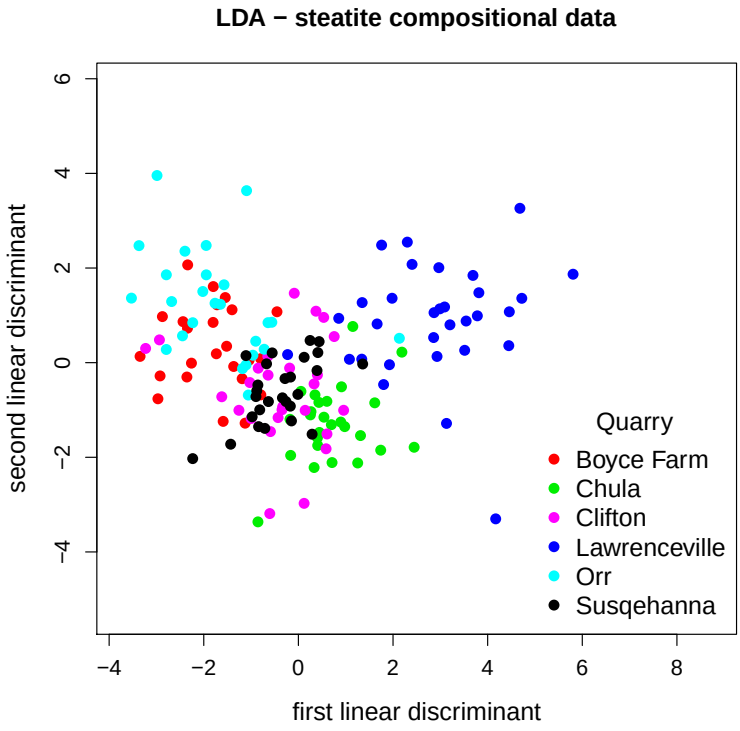

Figure 11.7: *LDA of the steatite compositional data of Tables B.21 to B.23.*

was used. This is the default; if `CV = TRUE` is used LOO cross-validation is implemented. This provides output in a different form; in particular `St.lda$class` provides the LOO predictions and this can be cross-tabulated with `St.type` to get the 'confusion' table (Table 11.2). The overall 'success' of classification is 113/159 = 71%, where 113 is the sum of the diagonal cells. If `St.lda$posterior` is accessed this provides posterior probabilities for the classes, of the kind shown in Table 11.1. The success of classification is summarized in percentage terms for each quarry in the third column of Table 11.3. This suggests that, for example, the classification for Boyce Farm and Orr is not very impressive.

In fact these assessments may be rather pessimistic. In attempting to discriminate between all six groups simultaneously the discriminant functions can be thought of as 'averaging' over all the groups, producing results that fail to show that good discrimination between pairs of groups may be possible. An alternative approach is to undertake LDAs on pairs of quarries. This is illustrated in Table 11.3 where the LOO and resubstitution success rates for each possible pair are given.

For the pairs the success rates are mostly noticeably greater than the global

Quarry	Predicted quarry						
	B	Ch	Cl	L	O	S	n
B	13	1	2	0	4	4	24
Ch	0	23	0	2	0	1	26
Cl	2	0	19	0	0	5	26
L	0	5	0	24	0	2	31
O	1	0	2	1	15	6	25
S	2	2	4	0	0	19	27

Table 11.2: *The 'confusion' table for LDA of the steatite data.*

			Pairwise comparisons					
Quarry	n	CV (%)	B	Ch	Cl	L	O	S
Boyce Farm	24	54		98	90	96	82	88
Chula	26	88	98		90	84	94	89
Clifton	26	73	92	94		93	88	85
Lawrenceville	31	77	100	90	95		96	90
Orr	25	60	86	98	92	96		90
Susquehanna	27	70	90	96	87	93	94	

Table 11.3: *Classifications from LDAs of the steatite compositional data. The CV is the percentage correctly classified using LOO methodology. The second part of the table shows the success rate for pairwise comparisons; the upper triangle is for LOO calculations, the lower triangle for the resubstitution approach.*

rates and in excess of 90%. Although not pursued in detail here, this is because the variables are weighted rather differently in the discriminant functions for pairs, both from each other and from the global analysis.

11.4 Classification trees

11.4.1 Basic ideas

Classification trees have been used intermittently in archaeological applications (Baxter, 2003: 116–118). They are an attractive alternative supervised learning method for problems often tackled using (LDA), if sample sizes are large enough. Venables and Ripley (2002: 331) state that 'classical methods of multivariate analysis [including LDA] have largely been superseded by methods from pattern recognition', and classification trees are one such 'modern' alternative.

Many of these modern methods are computationally complex, rather 'black-boxy', and difficult for the non-specialist to understand. Classification trees, by contrast, are conceptually simple and result in economical and elegant displays of

the results that – with some caveats to be entered – can be readily understood. Output is typically in the form of a tree diagram. The data are initially treated as a single group that is successively sub-divided until some stopping criterion is satisfied. The aim is to identify those variables that best separate the groups. This is best illustrated by example in Section 11.4.2. Technical aspects of the method are discussed in Section 11.5 with a further example in Section 11.6.

11.4.2 Example – Steatite compositions (continued)

The data on steatite compositions from Tables B.21 to B.23, used to illustrate LDA in Section 11.3.4, are used. Figure 11.8, the same as Figure 9.2 in Baxter (2003), shows the outcome of a classification tree analysis of the data.

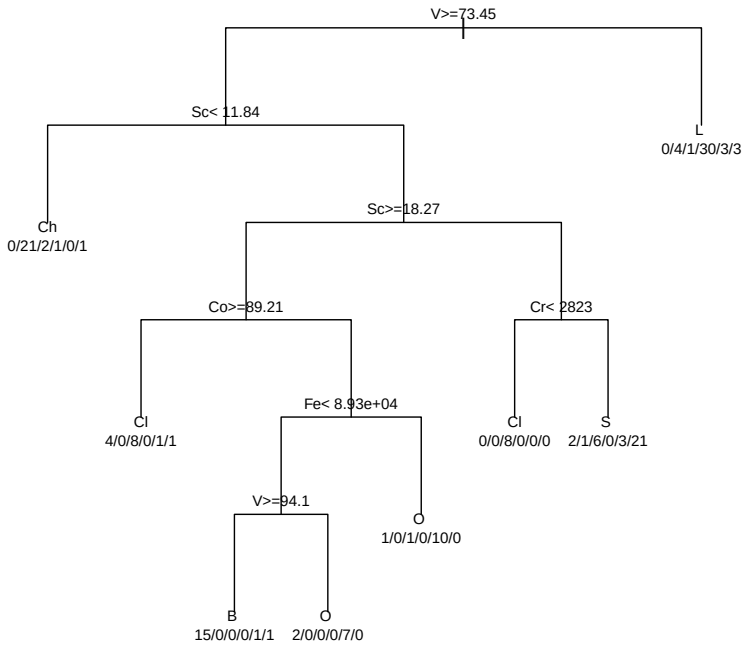

Figure 11.8: *A classification tree for the steatite compositional data.*

The starting point, the *root node*, based on all the data before group separation,

is associated with a measure of *purity* (Section 11.5). The root node is split into two groups/nodes on the basis of one variable, in order to increase the purity of the tree by as much as possible. This (binary) splitting continues until some stopping criterion is met.

Using the variable V (Vanadium) cases having V > 73.45 go to the left in the first split. It happens that the node to the right, according to the criterion used, is a *terminal node* and no further attempt is made to split it. The numbers below a terminal node show how many cases from each group are assigned to that node; 30/41 come from the Lawrenceville quarry which is the dominant group identified by the label L for the node. A *pure* terminal node is one where all the cases come from a single group, and the ideal is that all terminal nodes are pure.

To the left, the second split is based on values of Scandium (Sc) that are less than 11.844 and produces a second terminal node dominated by the Chula (Ch) quarry. The third split is also based on Scandium, this time using the higher values, Sc > 18.2695, without producing any further terminal nodes. As well as the fact that a variable can be used more than once in the splitting process, note that there are more terminal nodes (eight) than there are groups in the data. The quarries Clifton (Cl) and Orr (O) each dominate two terminal nodes; this can indicate either that there are distinct sub-compositions within the groups, or that they are not compositionally well-defined.

Most of the terminal nodes are not pure, so that the classification is less than perfect. The classification success can be assessed at 78% which compares favorably with that for LDA, using leave-one-out classification, of 72%.

11.5 Methodology

The preceding section describes the bare outlines of the methodology, which is conceptually straightforward. Implementation is computationally intensive, requiring a number of choices to be made on the way. A brief, but more technical, discussion of further aspects is provided in this section.

If the sample of size n is partitioned into G classes, with n_i cases in class i, the root node is defined by G probabilities $(p_1, p_2, \ldots, p_G)$ where $p_i = n_i/n$. A node is labeled according to the most dominant class. The misclassification rate at a node is the number of cases not in the dominant class. For the root node call this R_0. Splitting occurs as discussed in the previous section and criteria for what constitutes a terminal node need to be defined. In the example no attempt has been made to split a node with fewer than 20 cases and a terminal node must contain at least 7 cases.

The *impurity* of a node can be defined in different ways; in the example the

Gini index

$$1 - \sum_i p_i^2$$

was used. For a pure node this takes the value zero, and the smaller the value of the index the purer the node. Assume continuous variables are used. Any one of these, X say, can be ordered as $(x_{(1)}\ x_{(2)}\ \ldots\ x_{(n)})$. For $i = 2, \ldots, n$, splits may be contemplated such that cases with values less than $x_{(i)}$ are separated out from cases with values greater than or equal to $x_{(i)}$. New nodes are defined by the two subsets of cases thus defined, and the impurity of these, and hence their average impurity, measured. All possible splits for all variables are examined and the split that most reduces the average impurity is chosen. Each new node is treated in a similar way and the process continues until terminal nodes are reached.

Clearly the appearance of the final tree depends on a variety of choices, including that of the smallest node for which a split is allowed and the minimum size of the terminal node. An understanding of this, and other aspects of the final appearance that can be controlled by the user, is aided by looking at the code used to obtain Figure 11.8.

```
library(rpart)
z.rp <- rpart(St.quarry ~ Co + Cr + Fe + Mn + Sc + V,
data = St.data, cp = .03, minsplit = 20, minbucket = 7)

plot(z.rp, uniform = T, margin = .05)
text(z.rp, use.n = T, cex = .7)
```

The `rpart` package needs to be loaded. In the arguments to the `rpart` function `minsplit` and `minbucket` specify the minimum node size that can be split, and the minimal terminal node size. The values given are the defaults; varying them will change the appearance of the tree. The argument `cp` is a *complexity parameter* (of which more below) for which the default is 0.01; reducing this will result in a larger tree relative to the default, and increasing it will produce a smaller tree. For more detailed information see the `R` help on `rpart.control`. The `plot` function prints the tree. For large trees and/or those with many groups obtaining a satisfactory appearance usually requires some experimentation. The last two lines of code use arguments that affect the appearance; see the `R` help on `plot.rpart` for more detail.

It is possible to get satisfactory results by 'playing around' with the arguments in the `rpart` function, but different analysts may arrive at different trees. A more 'principled' approach to determining tree size is available in the `rpart` package. If `cp` is too small the data may be 'over-fitted' and give an over-optimistic assessment of the success of the classification. The tree in Figure 11.8 originally had 11

terminal nodes using `cp = .001` rather than the eight terminal nodes using `cp = 0.03`.

Cost-complexity pruning can be used to reduce the size of the tree. Let $R(T)$ be the number of misclassifications in a tree of size T as measured by the number of terminal nodes, and let C be the cost-complexity parameter defined by `cp`. A cost-complexity measure of the form

$$R_C = R(T) + CTR_0$$

can be defined. As C increases, and leaves are pruned, T will decrease and $R(T)$ will increase. The degree of pruning is chosen to minimize R_C.

To choose C cross-validation is used. The data are split into 10 groups of roughly equal size. Nine of the groups are used to grow a tree and it is tested out on the remaining group to obtain a measure of the error involved. This can be done in 10 ways and the results averaged to get an estimate of the error and its standard deviation. Results may be summarized graphically, as in Figure 11.9 based on the unpruned tree which, after pruning, led to Figure 11.8, obtained using `plotcp(z.rp)`.

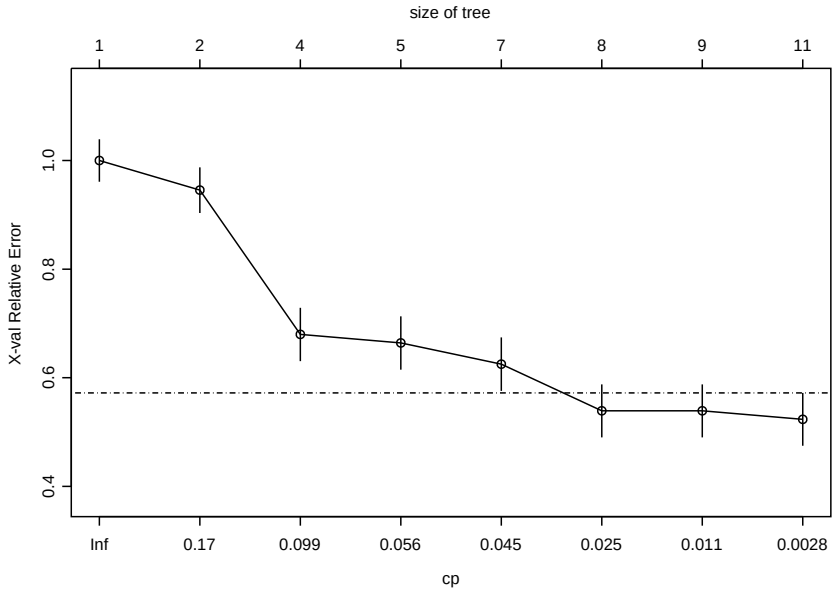

Figure 11.9: *A plot of error-rate against tree size as determined by* `cp`*, based on 10-fold cross-validation. It leads to a pruned tree of size 8 which is that shown in Figure 11.8.*

Plotted points show the mean number of errors across the 10 analyses divided by R_0 (i.e. the vertical axis is scaled to lie between 0 and 1), with 'one standard

error' bars for each mean. The horizontal reference line is located at the mean error for the largest tree plus one standard error. The tree selected is the smallest one whose mean error lies below and within one standard error of the line. In this case this rule results in the unpruned tree of size 11 being pruned to one of size 8. Incidentally, the choice of `cp` in the original call to `rpart` can be thought of as controlling pruning of the tree that takes place as the tree is grown; larger values of `cp` have the effect of preventing the growth of branches likely to be 'lopped-off' when using the above selection procedure.

To the best of my knowledge this 'rule' is not justified theoretically. Therneau and Atkinson (1997: 13), the originators of the `rpart` package, observe that this method has proved good at screening out 'pure noise' variables. That is, the justification for its use is empirical. Classification trees can be viewed as an alternative to variable selection methods in LDA, with several potential advantages. One is the transparency of the method. Others are the fact that the method can handle missing data (illustrated in Baxter and Jackson, 2001) or mixed data, and that it is not necessary to worry about data transformation because of the monotonic relationship between raw and log-transformed data.

11.6 Example 2 - North Apulian pottery

This analysis is based on that shown (with presentational modifications) in Figure 5 of Gliozzo *et al.* (2013). Chemical compositional data were available for pottery from four late antique/early medieval sites (4th to early 7th centuries AD) in northern Apulia, Italy. The main interest lay in seeing if two of the sites, *Herdonia* and *Canusium*, produced pottery that was chemically distinguishable, but similar data from two other sites, *Posta Crusta* and San Giusto, had previously been investigated and the opportunity was taken to compare these with the newer data sets. Data on two kinds of pottery were available, coarse table wares and fine painted wares, and a 65×36 table of results for the latter type – split into groups defined by the four sites – is used here for further illustration. The data are similar in nature to the previous example but far more variables are involved (Tables B.24 and B.25).

A classification tree is shown in Figure 11.10. Sample sizes for some of the sites are quite small so splits of nodes with more than 10 cases, as opposed to the default of 20, were allowed. This had the effect of making it much clearer that the productions of the sites could largely be distinguished in terms of their chemistry.

What is worth noting about this is that only five splits are needed, involving five of the 36 variables, a considerable 'saving' in terms of economy of description and comprehension. There are six terminal nodes of which four are pure; two sites, *Herdonia* and San Giusto split into two mostly quite small groups which are,

however, almost completely distinct from other sites. Of the 65 cases just 3 are misclassified, so the success of the classification is 95%.

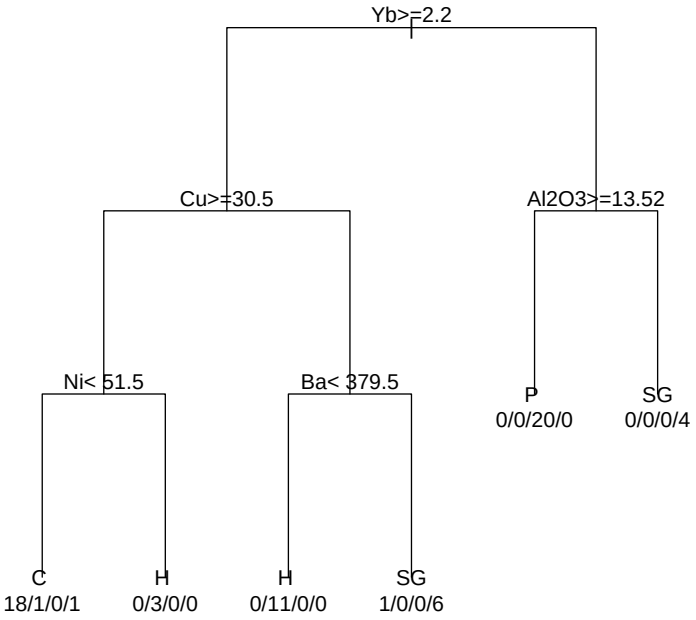

Figure 11.10: *A classification tree for the northern Apulian fine ware pottery data.*

Given the success of classification there is little point in comparing it with the results from pairwise comparisons. Results reported in Gliozzo *et al.* (2013) for the coarse ware data noted that the success rate for classification with all four sites in the analysis was 87%. For pairwise analyses the success rate varied between 90% and 96%, with only one or two variables needed to achieve this, except for the *Posta Crusta* and San Giusto comparison, which needed three. As with the fine wares, therefore, chemical separation between the sites for the coarse wares wss good; that in the latter case pairwise comparisons improved on the global analysis mirrors what was observed for the LDA of the steatite compositional data, and for analogous reasons.

11.7 R notes

Figure 11.1

The data sets `Kea`, `Seriphos` and `Lavrion` need to have been created; the last of these is not used for this figure, but is used in later examples.

```
library(car)
K <- Kea; S <- Seriphos; L <- Lavrion

plot(K[, 2], K[, 1], type = "n")
points(K[, 2], K[, 1]); points(S[, 2], S[, 1])

lines(dataEllipse(K[, 2], K[, 1], levels = c(0.9), plot.points = F,
      center.pch = 17, center.cex = 2))

lines(dataEllipse(S[, 2], S[, 1], levels = c(0.9), plot.points = F,
      col = "black", center.pch = 17, center.cex = 2))
```

If the `dataEllipse` function is used separately from the `lines` function it may be necessary to include other arguments to get the desired effect: see the help facility. The `levels` argument specifies that a 90% confidence ellipsoid is required; other arguments dictate the color, plotting character and size of the group centroids and are optional.

Figure 11.2

The three data sets are combined into `LKS` using the `rbind` function. The object `SymLKS` needs to be created and defines the plotting symbols used in the `pch` argument in `eqsc` but also doubles as the group identifier needed for the second argument of `lda` (presentational arguments are not shown). In the `predict` function the argument `dim = 2` extracts two functions (the maximum possible with three groups). This allows a single argument, `LKS.ld`, to define the plotting positions; in general, if more than two functions are extracted, the two columns to be used for plotting need to be listed separately.

```
library(MASS)
LKS <- rbind(L, K, S)

LKS.lda <- lda(LKS, SymLKS, CV = F)
LKS.ld <- predict(LKS.lda, dim = 2)$x
eqscplot(LKS.ld)
```

```
LKS.pca <- prcomp(scale(LKS))$x
x1 <- LKS.pca[, 1]; x2 <- LKS.pca[, 2]
eqscplot(x1, x2)
```

Figure 11.5

The data of Tables B.19 and B.20, excluding `Id.` and `Type`, are held in `TRB.data`. They need to be transformed in the manner described in the text, as follows, with the result in `TRB.trans`.

```
z1 <- TRB.data[, c(1, 3, 5, 7, 9, 11, 13, 15)]
z1 <- z1/z1[, 1]
z2 <- TRB.data[, c(2, 4, 6, 8, 10, 12, 14, 16)]
z2 <- z2/z2[, 1]
TRB.trans <- cbind(z1[, -1], z2[, -1])
```

Nothing really new is illustrated in the PCA below. An obvious outlier, case 112, was identified in preliminary analysis, and is omitted as shown. This needs to be done, also, for the previously created colors (`ColTRB`) and plotting characters (`SymTRB`) and this is shown in the call to `eqsc` for emphasis; other presentational arguments and the legend are omitted.

```
TRB.pca <- prcomp(scale(TRB.trans[-112, ]))$x
x1 <- TRB.pca[, 1], x2 <- TRB.pca[, 2]
eqscplot(x1, x2, col = ColTRB[-112], pch = SymTRB[-112])
```

The commands for LDA are shown for completeness, but introduce nothing new.

```
TRB.lda <- lda(TRB.trans[-112, ], SymTRB[-112], CV = F)
TRB.ld <- predict(TRB.lda, dim = 2)$x
eqscplot(TRB.ld, col = ColTRB[-112], pch = SymTRB[-112])
```

Figure 11.6

The left-hand graph introduces nothing new; all that needed is to select the first two types (rows 1 to 102) from the original data and labeling variables, so analysis is based on `TRB.trans[c(1:102), ]` for the PCA. The first two lines in the following code set up the results for the LDA plot, but the usual bivariate plots are not available because there are only two groups and one discriminant function. The remaining code produces the plot shown in the right-hand side of the figure.

```
library(MASS)
TRB.lda <- lda(TRB.trans[c(1:102), ], SymTRB[c(1:102)], CV = F)
TRB.ld <- predict(TRB.lda, dim = 1)$x

id <- ifelse(SymTRB2 == 15, 2, 1)
plot(id, TRB.ld, xaxt = "n")
axis(1, at = c(1:2), labels = c("funnel beakers", "bowls"))
```

The variable `id` provides a convenient label for the two groups using the `ifelse` function. Most of the plotting arguments are omitted. The `xaxt = "n"` suppresses printing of the x-axis since its appearance is not as needed. It is replaced using the `axis` function. The first argument, 1, specifies the side at which the new axis is to be placed (bottom – see the help for `axis` for other placements); the `at` argument provides the locations of new labels (replacing 1 and 2 in the original plot) whose names are given in the `labels` argument.

Table 11.1

Replace the argument `CV = F` with `CV = T` in the above code; then

```
posteriors <- round(cbind(id, TRB.lda$class, TRB.lda$posterior), 2)
```

will produce the original and predicted groupings (`id` and `TRB.lda$class`), with the posterior probabilities, `TRB.lda$posterior`, rounded to two decimal places. This is for all the data; a certain amount of (straightforward) editing is needed to get it in the format presented in the text.

The analyses for the steatite data use more groups but involve nothing new in what is needed for the plot.