9 Univariate tree models

9.1 Introduction

A further tool to explore the relationship between a single response variable and multiple explanatory variables is a regression or classification tree (Chambers and Hastie 1992; De'Ath and Fabricus 2000; Fox 2000; Venables and Ripley 2002; Maindonald and Braun 2003). Classification trees are used for the analysis of a nominal response variable, and regression trees for a continuous response variable. Both types of tree models deal better with non-linearity and interaction between explanatory variables than regression, generalised linear models and generalised additive models, and can be used to find interactions missed by other methods. They also indicate the relative importance of different explanatory variables and are useful in analysing residuals from linear regression, GLM or GAM.

Tree models are relatively new in ecology, so we begin with a detailed, but non-technical, explanation of how they work using an artificial example inspired by Maindonald and Braun (2003). Suppose we count the numbers of bees found at seven sites on a particular day. Sampling took place in the morning and the afternoon, and the number of plant species found at each site were also recorded (Table 9.1). The response variable is the number of bees (abundance), and the explanatory variables are the number of plant species at a site and the time of sampling (morning or afternoon). Time of day is a nominal variable, and the number of plant species is considered a continuous variable. The questions of interest in this hypothetical example are whether bee abundance is related to time of day and/or number of different plant species, and which explanatory variable is the most important.

The overall mean of the data is $\bar{y} = (0 + 1 + 2 + 3 + 4 + 5 + 6)/7 = 3$ and we define the deviance D, or total sum of squares, as

$$D = \sum_{j=1}^{7} (y_j - \bar{y})^2$$

where y_j is the number of bees at site j. This gives a value for $D = 28$.

Table 9.1. Artificial bee data. The response variable is the bee abundance and the explanatory variables are number of plants and time of day (M = morning and A = afternoon).

Site	Bees	Number of Plant Species (P)	Time of Day
A	0	1	M
B	1	1	A
C	2	2	M
D	3	2	A
E	4	3	M
F	5	3	A
G	6	3	A

Now suppose that we want to split the seven observed bee data into *two* groups, based on the values of an explanatory variable. We have arbitrarily decided to start with time of day. As this explanatory variable has only two classes, we can readily split the bee data into two groups:

- Group 1: 0, 2 and 4 bees for the morning.
- Group 2: 1, 3, 5 and 6 bees for the afternoon.

Using basic algebra, it can be shown that the deviance can be rewritten as

$$\begin{aligned} D &= \sum_{j=1}^{7}(y_j - \bar{y})^2 \\ &= \sum_{j\in \text{morning}}(y_j - \bar{y})^2 + \sum_{j\in \text{afternoon}}(y_j - \bar{y})^2 \qquad (9.1) \\ &= \\ &\sum_{j\in \text{morning}}(y_j - \bar{y}_M)^2 + \sum_{j\in \text{afternoon}}(y_j - \bar{y}_A)^2 + n_M(\bar{y}_M - \bar{y})^2 + n_A(\bar{y}_A - \bar{y})^2 \qquad (9.2) \\ &= 8 + 14.75 + 3 + 2.25 = 28 \end{aligned}$$

where $\bar{y}_M$ and $\bar{y}_A$ are the averages of the morning and afternoon data, and n_M and n_A are the number of observations in the morning and afternoon, respectively. Note that $\bar{y}_M = (0 + 2 + 4)/3 = 2$ and $\bar{y}_A = (1 + 3 + 5 + 6)/4 = 3.75$. Going from equation (9.1) to (9.2) requires basic algebra. The first component in equation (9.2) measures the variation within the morning observations. The second component determines the variation within the afternoon observations. So, the sum of the first two components represents the *within group variation*. The term $n_M(\bar{y}_M - \bar{y})^2$ measures the deviation of the morning average from the overall average, and the fourth term in equation (9.2) gives the deviation of the afternoon average from the overall average. The sum of the third and fourth components in equation (9.2) therefore represents the *between group variation*. The results are visualised in Figure 9.1, which can be interpreted as follows. If for a particular observation, the statement "Time = morning" is true, then follow the left branch, and if "Time = morning" is false, follow the right branch. The mean value for the

number of bees from the morning observations is 2 ($n = 3$), and for the afternoon observations is 3.75 ($n = 4$).

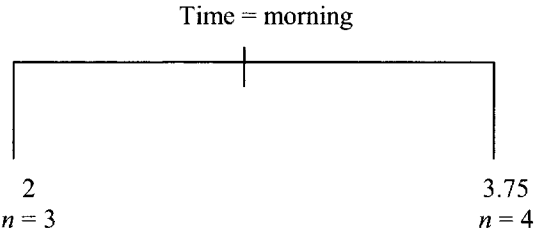

Figure 9.1. Graphical presentation for splitting up the bee data in morning and afternoon data. The mean value in the morning data is 2 ($n = 3$ observations), and in the afternoon 3.75 ($n = 4$ observations).

Instead of splitting the bee data into two groups using the time of day (morning and afternoon), we could equally have chosen the number of different plant species, and this is done next. This is a continuous explanatory variable. We now introduce an additional rule. The options for splitting the response variables are constrained by the need to keep any splits in the *continuous* explanatory variables in their original order. There are two possible options to split the bee data *into two groups* based on the continuous variable numbers of different plant species:

- Option 1: sites A and B versus C, D, E, F, G.
- Option 2: sites A, B, C, D versus E, F, G.

Note that groups C, D versus A, B, E, F, G is not an option as the splitting rule for a continuous variable is based on the order of the value of an explanatory variable. The first option results in:

$$
\begin{aligned}
\mathrm{D} &= \sum_{j=1}^{7}(y_j - \bar{y})^2 \\
&= \sum_{j=A,B}(y_j - \bar{y})^2 + \sum_{j=C,D,E,F,G}(y_j - \bar{y})^2 \\
&= \sum_{j=\mathrm{A,B}}(y_j - \bar{y}_{A,B})^2 + \sum_{j=C,\ldots,G}(y_j - \bar{y}_{C,\ldots,G})^2 + n_{A,B}(\bar{y}_{A,B} - \bar{y})^2 + n_{C,\ldots,G}(\bar{y}_{C,..G} - \bar{y})^2 \\
&= 0.5 + 10 + 12.5 + 5 = 28
\end{aligned}
$$

The mean value at the sites A and B is $\bar{y}_{A,B} = 0.5$ and at sites C,..,G it is $\bar{y}_{C,..,G} = 4$. Just as before, the results can be presented graphically (Figure 9.2). If an observation has only one species of plant or no plants present, then the mean value of bees is 0.5, and if there are two or more plant species present, then the mean value for number of bees is four.

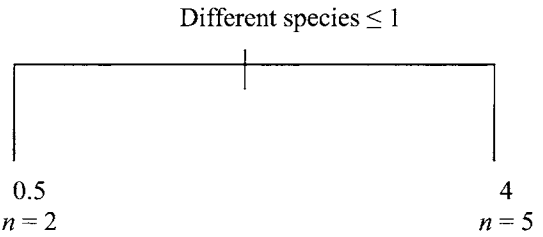

Figure 9.2. Graphical presentation for splitting the bee data based on the values of the number of plant species. The mean value for observations with 1 species of plant or no plants is 0.5 bees, and at observations with more than one plant species present, the mean number of bees is four.

Note that this division results in a within group variation of 10.5 and a between group variation of 17.5. We leave it as an exercise to the reader to check that splitting the data using the second option results in a within variation of 5+2=7 and between variation of 9 + 12 = 21. To summarise, using the explanatory variables we have divided the bee data three times into two groups. The within and between group variation for these splits are:

1. $D_{\text{time of day}} = D_{\text{within}} + D_{\text{between}} = 22.75 + 5.25$
2. $D_{\text{different species} \leq 1} = D_{\text{within}} + D_{\text{between}} = 10.5 + 17.5$
3. $D_{\text{different species} \leq 2} = D_{\text{within}} + D_{\text{between}} = 7 + 21$

The aim is to have groups that have a between variation as large as possible and a within variation as small as possible. Clearly dividing the data into two groups using the time of day is a poor choice as it gives the highest within group variation. In practice the statistical software automatically chooses the optimal grouping; and in this case it is group three, which has the smallest within group variation, and the largest between group variation. For each sub-group, the software will apply the same procedure on the remaining observations and continue until some stopping criteria are met. This process of repeatedly partitioning the observations into two homogenous groups based on the values (order) of an explanatory variable is called a regression (or classification) tree. Note that applying a transformation on the explanatory variables will not change the deviances. Hence tree models are not affected by transformations on the explanatory variables.

The terminal nodes are called leaves, and the regression tree gives a mean value for each leaf allowing the residuals for each observation to be calculated as observed values minus the average leaf value. The sum of squares of the residuals can then be calculated, and adding them together will give the residual sum of squares. Alternatively, the sum of the deviances of the leaves can be calculated (the sum of all D_{within} values).

Example: Bahamas fisheries data

Using the Bahamas fisheries dataset that we used earlier (see Section 7.3), we now look at a real example. In this example, parrotfish densities are used as response variable, with the explanatory variables: algae and coral cover, location, time (month), and fish survey method (method 1: point counts, method 2: transects). There were 402 observations measured at 10 sites, and the regression tree is shown in Figure 9.3. Looking at the regression tree you can see that the 402 parrotfish observations (the response variable) are repeatedly split into groups based on the values of the explanatory variables. In each step this division keeps each group as homogenous as possible. Splitting is conditional on the order of the values of the explanatory variable, and with nominal variables, the division is by (groups of) classes. For the parrotfish, the algorithm splits the data into two groups of 244 observations (left branch) and 158 observations (right branch) and is based on the explanatory variable 'fish survey method', which we called 'Method'. Hence, the most important variable in splitting the data is the survey method: point sampling versus transect sampling. The typical numerical output of SPlus and R is given below, and it shows that the overall deviance and mean are 50188.35 and 10.78, respectively.

```
node), split, n, deviance, yval      * denotes terminal node
 1) root   402  50188.35  10.78
   2) as.factor(Method)=2   244   7044.21   6.45
     4) CoralTotal< 4.955      87   1401.13   3.53 *
     5) CoralTotal>=4.955    157   4488.06   8.07 *
   3) as.factor(Method)=1    158  31494.20  17.47
     6) as.factor(Station)=3,4  25   781.74   3.157 *
     7) as.factor(Station)=1,2,5,6,7,8,9,10  133  24627.58  20.16
       14) as.factor(Month)=5,7,8,10  94  11587.55  17.09*
       15) as.factor(Month)=11  39  10023.52  27.56 *
```

This output looks a little confusing. We will explain the key aspects of interpretation here and look at it in more detail later in Chapter 23. The notation 'as.factor' is a computer notation for nominal variables. The first line shows that there are 402 observations, the total deviance is 50188.35 and the overall mean is 10.78. The lines labelled as 2) and 3) indicate the first split is based on the nominal explanatory variable 'Method'. The left branch is described by the lines labelled as 2), 4) and 5), and the right branch as 3) and all the lines below this one. The layout therefore mirrors the graphical tree representation of the data. Starting with the left branch, the information on the line labelled 2) shows that all the observations in 'Method=2' have a mean of 6.45, but it is not a final leaf as final leaves are labelled by a '*' at the end of the line. The 244 observations can be further split based on coral total and the cut off level is 4.955. All observations smaller than this threshold value for a group with a mean of 3.53, which is one of the smallest group means in the tree! The 157 observations with coral total larger than 4.955 have a mean value of 8.07. Now let us look at the main right-hand branch, starting at the line labelled 3). These observations are split further based

on station and month, making these the second most important variables (for observations measured with method one).

So, what does this all mean? The group of observations with the highest mean value is obtained by method one, at stations 1, 2, 5, 6, 7, 8, 9 and 10, and is equal to 27.56 (39 observations). In fact, most groups in Method 1 have higher mean densities, which means that there is a structural difference between the observations due to sampling technique!

If a particularly explanatory variable appears repeatedly along the branches in a tree, this indicates a non-linear relationship between the response variable and the explanatory variable.

The results indicate that for the parrotfish, the type of sampling method is important and it might be useful to apply the same procedure on the other fish groups sampled. If similar results are found with other fish groups, then there is a strong argument for analysing the data from point and survey transects separately. Instead of applying a univariate tree model on each individual fish family, it is also possible to apply a multivariate regression tree (De'Ath 2002) on all the fish groups. But this method is not discussed in this book.

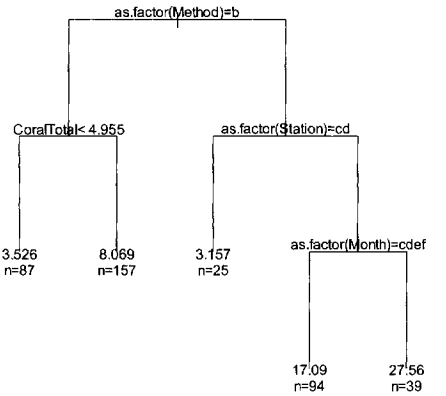

Figure 9.3. Regression tree for parrotfish of the Bahamas fisheries dataset. The observations are repeatedly split into two groups. The first and most important split is based on the nominal explanatory variable 'Method', where 'a' stands for method 1 and 'b' for method 2. If a statement is true, follow the left side of a branch, and if false follow the right side. Numbers at the bottom of a terminal leaf represent the mean value and the number of observations in that particular group.

9.2 Pruning the tree

A major task in linear regression is selecting the optimal subset of explanatory variables. Using too many explanatory variables results in a model that overfits the data and is difficult to interpret. Using only a few explanatory variables can lead to a poor model fit. Tools like the AIC (Chapter 5) can be used to judge which selection of explanatory variables is optimal in terms of model fit and model simplicity. In tree models, we have a similar problem deciding on the size of the tree: defined as the number of splits in the tree. In Figure 9.3 the size of the tree is five, because it has four splits. The number of splits is always equal to the number of leaves (terminal nodes at the bottom) minus 1. If a large tree size is used, we end up with very small groups but lots of terminal nodes, and therefore lots of information to interpret. On the other hand, using a small tree might result in a poor fit. The mean value of a group of observations is given at the end of a branch. The fitted value of a group is its mean value. Hence, if we have lots of terminal nodes with only a few values in it, it is likely that the mean value is close to the observations in the group. On the other hand, if there are only a few splits, then we have groups with lots of observations and the mean value may not represent all observations in that group (some points may have a large deviation from the group mean, causing large residuals).

Hence, the algorithm for repeatedly splitting up the data into two groups needs to stop at a certain stage. One option is to stop the algorithm if the 'to be split subset of data' contains less than a certain threshold value. This is called the minimum split value. However, this is like saying in linear regression: 'use only the best 5 explanatory variables' and a more sophisticated method is available. Define D_{cp} as

$$D_{cp} = D + cp \times \text{size-of-tree}$$

Recall that the deviance D measures the lack of fit, and cp is the so-called complexity parameter (always positive). Suppose that $cp = 0.001$. The criteria D_{cp} is similar to the AIC; if the size of the tree is small the term $0.001 \times$ size-of-tree is small, but D is relatively large. If the size of the tree is large, D is small. Hence, $D_{0.001}$ can be used to find the optimal size in a similar way as the AIC was used to find the optimal regression model. However, the choice $cp = 0.001$ is rather arbitrary, and any other choice will result in a different optimal tree size. To deal with this, the tree algorithm calculates the full tree, and then prunes the tree (pruning means cutting) back to the root. The graphical output of the pruning process is given in Figure 9.4 and is explained next. It is also useful to have the numerical output available, see Table 9.2. This numerical output was obtained by using the default cp value of 0.001. The column Rel-error gives the error of the tree as a fraction of the root node error (=deviance divided by the number of observations). For example, the tree of size 3 has a relative error of 0.59. This means that the sum of all leaf deviances is $0.59 \times 50188 = 29610.92$, and the error is $0.59 \times 125 = 73.75$. Recall that the total deviance was 50188.35. The more leaves used in a tree, the smaller the relative error. Note that there is a large decrease in the relative er-

ror for one, two and three splits, but thereafter differences become rather small. In this case choosing a tree with four splits seems to be a safe choice.

A better option for tree size selection is cross-validation. The tree algorithm applies a cross-validation, which means that the data are split into k (typically $k = 10$) subsets. Each of these k subsets is left out in turn, and a tree is calculated for the remaining 90% (if $k = 10$) of the data. Once the optimal tree size is calculated for a given *cp* value using the 90% subset, it is easy to determine in which leaves the observations of the remaining 10% belong by using the tree structure and the values of the explanatory variables. We already have the mean values per leaf so we can calculate a residual (observed value minus group mean) and prediction errors (sum of all squared difference between observed values and mean values) for each observation in the 10% group. This process is applied for each of the $k = 10$ cross-validations, giving 10 replicate values for the prediction error. Using those 10 error values, we can calculate an average and standard deviation. This entire process is then repeated for different tree sizes (and *cp* values) in a 'back-ward selection type' approach. This is illustrated in Figure 9.4. The average (dots) and the standard deviation (vertical bars) are plotted versus the complexity parameter *cp* and the tree size. Along the lower x-axis, the complexity parameter *cp* is printed, and the size of the tree runs along the upper x-axis. The y-axis is the relative error in the predictions, obtained by cross-validation. The vertical lines represent the variation within the cross-validations (standard deviation). This graph is used to select the closest to optimal *cp* value. A good choice of *cp* is the leftmost value for which the mean (dot) of the cross-validations lies below the horizontal line. This rule is called the one standard deviation rule (1-SE). The dotted line is obtained by the mean value of the errors (x-error) of the cross-validations plus the standard deviation (x-std) of the cross-validations upon convergence. Hence, this is $0.62 + 0.07 = 0.69$. The optimal tree is the first tree where the x-error is smaller than 0.69, which is either a tree of size 3 (Nsplit = 3) or 4 (Nsplit = 4) (Table 9.2). Figure 9.4 also suggests that a tree of size 3 or 4 is the optimal one. We decided to present the tree of size 4 (Figure 9.3), as the 1-SE rule should only be used as a *general* guidance. Figure 9.3 shows the tree with Nsplit = 4.

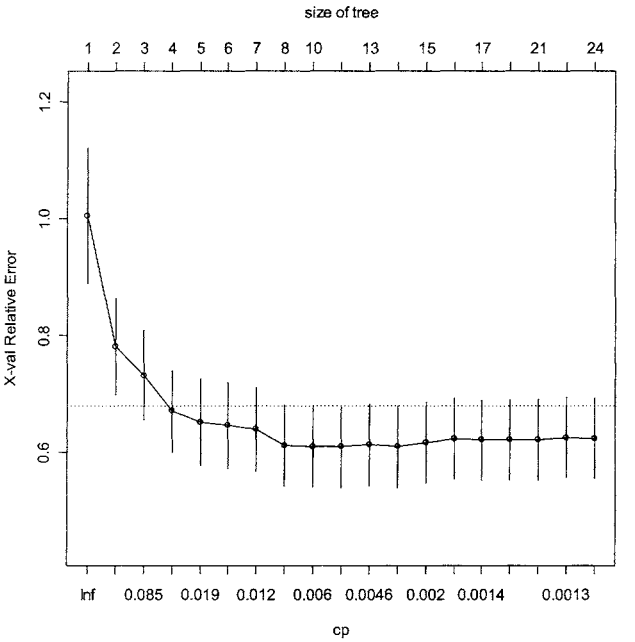

Figure 9.4. Pruning the tree. The lower horizontal axis shows the *cp* values and the upper horizontal axis the corresponding tree sizes. The vertical axis is the relative error in the predictions, obtained by cross-validation. The dots are the averages of the 10 cross-validations, and the vertical lines around the dots represent the variation within the cross-validations (standard deviation).

Table 9.2. Numerical output of the pruning process. The root node error is 50188/402 = 125, and there are 402 observations. X-error is the mean value of the k cross-validations (and is expressed as a percentage of the root node error), and x-std is the standard deviation. The column labelled "Rel-error" is the percentage of the root deviance explained by all terminal leaves of the tree.

	cp	Nsplit	Rel-error	x-error	x-std
1	0.2321	0	1.00	1.00	0.116
2	0.1212	1	0.77	0.78	0.083
3	0.0601	2	0.65	0.73	0.078
4	0.0230	3	0.59	0.67	0.071
5	0.0152	4	0.56	0.65	0.075
6	0.0132	5	0.55	0.65	0.074
7	0.0108	6	0.54	0.64	0.072
8	0.0062	7	0.52	0.61	0.070
..	...	..	...	...	...
19	0.0001	23	0.48	0.62	0.070

9.3 Classification trees

Classification trees work in the same way as regression trees except that (i) the response variable is a nominal variable, and (ii) the deviance is defined slightly differently. For 0–1 data, the deviance at a particularly leaf j is defined as

$$D_j = -2[n_{1j} \log \mu_j + n_{0j} \log(1 - \mu_j)]$$

where n_{1j} is the number of observation in leaf j for which $y = 1$ and n_{0j} is the number of observations for which $y = 0$. The fitted value at leaf j, μ_j, is the proportion $n_{1j}/(n_{1j} + n_{0j})$. The overall deviance of a tree is the sum of the deviances over all leaves. If the response variable has more than two classes, the deviance at leaf j is defined as

$$D_j = -2\sum_{i=1}^{n} n_{ij} \log \mu_{ij}$$

For example, if the response variable has the classes 1, 2 and 3, then the deviance at leaf j is defined as:

$$D_j = -2[n_{1j} \log \mu_{1j} + n_{2j} \log \mu_2 + n_{3j} \log \mu_{3j}]$$

where n_{1j} is the number of observations at leaf j for which $y = 1$, and $\mu_{1j} = n_{1j}/(n_{1j} + n_{2j} + n_{3j})$. The classification tree shows misclassification errors (errors/number of observations per leaf) instead of mean values. Further details and examples of classification trees are discussed in Chapter 24.

9.4 A detailed example: Ditch data

These data are from an annual ditch monitoring programme at a waste management and soil recycling facility in the UK. The soil recycling facility has been storing and blending soils on this site for over many years, but it has become more intensive during the last 8 to 10. Soils are bought into the stockpile area from a range of different sites locations, for example from derelict building sites, and are often saturated when they arrive. They are transferred from the stockpile area to nearby fields and spread out in layers approximately 300 mm deep. As the soils dry, stones are removed, and they are fertilised with farm manure and seeded with agricultural grasses. These processes recreate the original soil structure, and after about 18 months, the soil is stripped and stockpiled before being taken off-site to be sold as topsoil.

The main objective of the monitoring was to maintain a long-term surveillance of the surrounding ditch water quality and identify any changes in water quality that may be associated with the works and require remedial action. The ecological interest of the site relates mainly to the ditches: in particular the large diversity of aquatic invertebrates and plants, several of which are either nationally rare or

scarce. Water quality data were collected four times a year in Spring, Summer, Autumn and Winter, and was analysed for the following parameters: pH, electrical conductivity (μS/cm), biochemical oxygen demand (mgl^{-1}), ammoniacal nitrogen (mgl^{-1}), total oxidised nitrogen, nitrate, nitrite (mgl^{-1}), total suspended solids (mgl^{-1}), chloride (mgl^{-1}), sulphate (mgl^{-1}), total calcium (mgl^{-1}), total zinc (μgl^{-1}), total cadmium (μgl^{-1}), dissolved lead (μgl^{-1}), dissolved nickel (μgl^{-1}), orthophosphate (mgl^{-1}) and total petroleum hydrocarbons (μgl^{-1}). In addition to water quality observations, ditch depth was measured during every visit. The data analysed here was collected from five sampling stations between 1997 and 2001. Vegetation and invertebrate data were also collected, but not used in this example.

The underlying question now is whether we can make a distinction between observations from different ditches based on the measured variables, and whether a classification tree can help with this.

A classification tree works in a similar way as a regression tree, except that the response variable is now a nominal variable with two or more classes. Here, the response variable is the ditch number with classes one, two, three, four and five. The explanatory variables are the chemical variables plus depth, month (nominal) and year (nominal). Tree models are not affected by a data transformation on the explanatory variables, and therefore we used the untransformed data. A detailed discussion of regression trees was given earlier in this chapter. Here, a short summary is given and we spend more time looking at the numerical output produced by software like Splus and R, as it can be rather cryptic.

As in linear regression (Chapter 5), we need to find the optimal regression or classification tree. A model with all variables is likely to overfit the data but using too few variables might give a poor fit. An AIC type criteria (Chapter 5) is used to determine how good or bad a particular tree is, and is of the form:

$$RSS_{cp} = RSS + cp * \text{size of tree} \tag{9.3}$$

For regression trees, *RSS* stands for residual sum of squares and is a measure of the error. For 0−1 data, the *RSS*, or deviance at a particular leaf j was defined in Section 9.3. If the response variable has more than two classes, say five ditches, the deviance at leaf j is defined as

$$D_j = -2\sum_{i=1}^{5} n_{ij} \log \mu_{ij}$$

$$D_j = -2[n_{1j} \log \mu_{1j} + n_{2j} \log \mu_2 + n_{3j} \log \mu_{3j} + n_{4j} \log \mu_{4j} + n_{5j} \log \mu_{5j}]$$

where n_{1j} is the number of observations at leaf j for $y = 1$, and $\mu_{1j} = n_{1j}/(n_{1j} + n_{2j} + n_{3j} + n_{4j} + n_{5j})$. The parameter cp is a constant. For a given value, the optimal tree size can easily be determined in a similar way to choosing the optimal number of regression parameters in a regression model. Setting $cp = 0$ in equation (9.3) results in a very large tree as there is no penalty for its size and setting $cp = 1$ results in a tree with no leaves. To choose the optimal cp value, cross-validation can be applied. The underlying principle of this approach is simple: leave out a certain percentage of the data and calculate the tree. Once the tree is available, its struc-

ture is used to predict in which group the omitted data falls. As we know to which groups the omitted data belong, the actual and the predicted values can then be compared, and a measure of the error (the prediction error) can be calculated. This process is then repeated a couple of times, omitting different sets of observations. In more detail, the data are divided in 10 parts and 1 part is omitted. The tree is then estimated using 90% of the data. Once the tree has been estimated, the omitted 10% can be used to obtain a prediction error. This process is then repeated by leaving out each of the 10 datasets in turn. This gives 10 prediction errors. The mean values of these 10 cross-validation prediction errors are represented by dots in Figure 9.5. The vertical bars represent the variation in the 10 cross-validation errors. To choose the optimal *cp* value, the 1-SE rule can be used. This rules suggests choosing the *cp* value for the first mean value (dot) that falls below the dotted horizontal line (Figure 9.5). The dotted line is obtained from the average cross-validation mean multiplied with the standard deviation of the 10 mean values for the largest tree. In this case, the optimal tree size is four (Figure 9.5), and the corresponding *cp* value is slightly smaller than 0.1. Instead of the graph in Figure 9.5, the numerical output produced by most programmes can be used to choose the most optimal tree size (see below). The cross-validation mean value for the largest tree is 0.76 (this is a percentage of the root node error), and the standard deviation is 0.089. The sum of these two is 0.849. The smallest tree that has a smaller mean cross-validation error (x-error = 0.84) has three splits, and therefore has a tree size of four.

	cp	Nsplit	rel-error	x-error	x-std
1	0.237	0	1.00	1.18	0.044
2	0.184	1	0.76	1.16	0.050
3	0.132	2	0.58	1.00	0.074
4	0.079	3	0.45	0.84	0.086
5	0.053	4	0.37	0.82	0.087
6	0.026	7	0.21	0.76	0.089
7	0.001	8	0.18	0.76	0.089
Root node error: 38/48 = 0.79. n = 48					

The root node error for a regression tree is the total sum of squares. For a classification tree it is the classification error. Because most observations in the dataset were from group one (10 observations), the algorithm classified the entire dataset as group one (in fact it was a tie because two other groups that also had 10 observations). Therefore, observations of all other groups, 38 in total, are wrongly classified, and the root node error is 38 out of 48 (=total number of observations). Using eight splits, which corresponds to a tree of size nine, gives an error of 18% (0.18) of the root node error. A tree of size four (Nsplit = 3) has an error of 45% of the root error. Further output produced by tree software is given below. An explanation is given in the next paragraph.

node) split	n	loss	yval	(yprob) *=terminal node
1) root	48	38	1	(0.21 0.19 0.21 0.21 0.19)
2) Total.Calcium>=118	25	16	5	(0.32 0.28 0 0.04 0.36)
4) Conductivity.< 1505	11	6	1	(0.45 0.45 0 0.091 0) *
5) Conductivity.>=1505	14	5	5	(0.21 0.14 0 0 0.64) *
3) Total.Calcium< 118	23	13	3	(0.087 0.087 0.43 0.39 0)
6) Depth>=0.505	8	0	3	(0 0 1 0 0) *
7) Depth< 0.505	15	6	4	(0.13 0.13 0.13 0.6 0) *

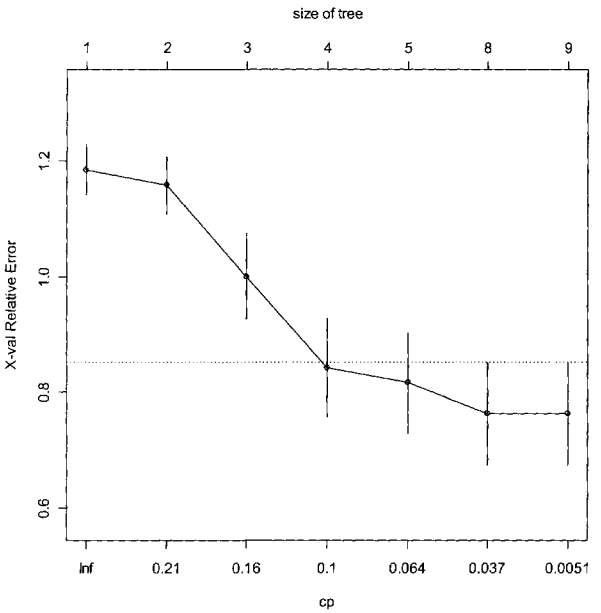

Figure 9.5. Cross-validation results for ditch data. The optimal tree size is four.

The optimal classification tree is presented in Figure 9.6. The most important variable to classify the 48 observations is total calcium. All observations with total calcium values, equal to, or larger than 118 are in the left branch. There are 25 such observations. The other 23 observations are in the right branch. These are not terminal nodes, but the relevant numerical output is:

node) split	n	loss	yval	(yprob)
2) Total.Calcium>=118	25	16	5	(0.32 0.28 0 0.04 0.36)
3) Total.Calcium< 118	23	13	3	(0.087 0.087 0.43 0.39 0)

The proportions per group (as a fraction of 25 observations) are 0.32/0.28/0/0.04/0.36. These proportions correspond to the ditches (groups) one, two, three, four and five. Hence from the 25 observations, 8 ($0.32 \times 25 = 8$) were

from group 1. Expressed as real numbers per group, this is: 8/7/0/1/9. Most observations (nine) are from group five, and therefore, this group is classified as group five. However, it is not a terminal node, and therefore further splitting is applied. Sixteen observations are incorrectly classified (this is called the loss). Both splits can be further split. For example, the 25 observations in the left branch can be divided further on conductivity. Observations with conductivity values smaller than 1505 are classified as from ditch one, and those with larger conductivity as ditch five. These are terminal nodes and the relevant output is:

```
4) Conductivity < 1505   11  6  1  (0.45 0.45 0 0.091 0) *
5) Conductivity >=1505   14  5  5  (0.21 0.14 0 0 0.64) *
```

There are 11 observations in this group, and they were from the following ditches: 5/5/0/1/0. These can either be inferred from the numerical output or from the numbers at the end of each leaf. The number of wrongly classified observations in this branch is 6. Note that we actually have a tie, and the algorithm chooses for the first ditch. For observations with total calcium larger, or equal to 118 and conductivity larger than or equal to 1505, the predicted ditch is five. There are 14 such observations, and the observations per group are as follows: 3/2/0/0/9. Hence, this is clearly a ditch five group. The right branch makes a clear distinction between observations from ditch three and four, and involves depth.

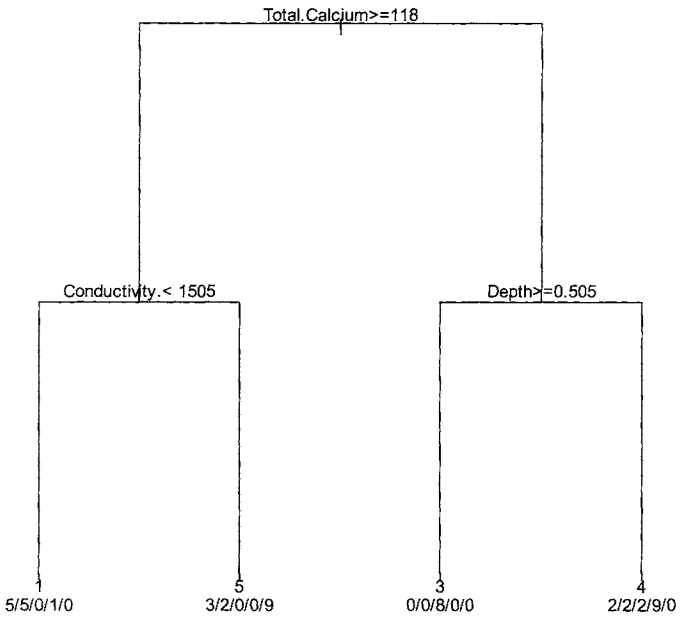

Figure 9.6. Optimal tree for ditch data. The numbers below terminal nodes represent groups (ditches), and the notation '5/5/0/1/0' in the far left terminal leaf means that in this group there were 5 observations from ditch one, 5 from ditch two, 0 from ditch three, 1 from ditch four and 0 from ditch five. The ditch with the highest number of observation wins and determines the name of the group. In this case it is a tie between ditch one and two and one is chosen arbitrarily. The tree shows that a classification can be made among ditches one, five, three and four based on total cadmium, conductivity and depth, but there is not enough information to make a distinction between ditch two and four.

The classification tree indicates that if observations have a total calcium smaller than 118 and depth values larger than 0.505, then they are likely to be from ditch three. If the depth is smaller than 0.505, then they are from ditch four. On the other hand, if total calcium is larger than 118, then conductivity can be used to classify observations in either group five or in groups one or two. Note that there is not enough information available in the explanatory variables to discriminate

ditch two. It is the closest to group one (see the proportions in the leftmost terminal leaf). These results indicate that total calcium, depth and conductivity are the most important variables to discriminate among the observations from the five ditches.

To clarify the results obtained by the classification tree, a Cleveland dotplot of total calcium was made (Figure 9.7). The tree identified the value of 118. If one imagines a vertical line at 118, then most observations of ditches three and four have calcium values lower than 118 and most observations from ditches one, two and five have calcium values higher than 118. Similar Cleveland dotplots using depth and total conductivity can be made for the sub-groups.

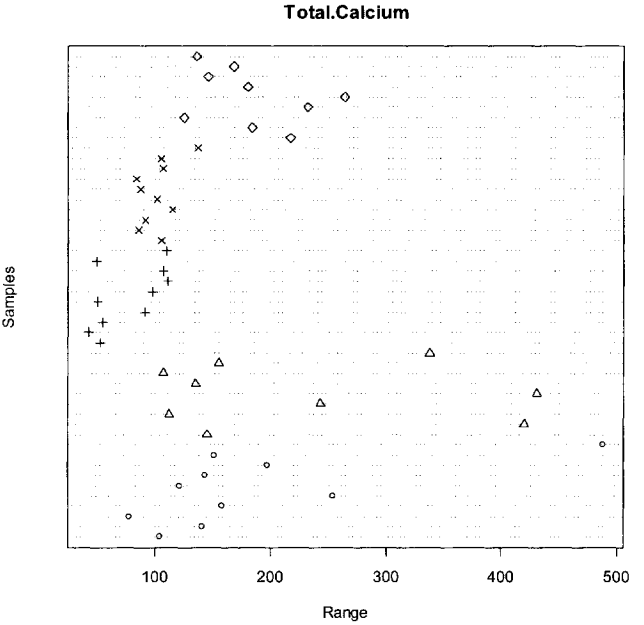

Figure 9.7. Cleveland dotplot for total calcium. Observations from the same ditch are represented by the same symbols. The horizontal axis shows the value of total calcium, and the vertical axis shows each observation in the same order as in the spreadsheet; the 10 observations at the top are from ditch five, and the first 10 at the bottom are from ditch one. Hence, the total calcium in the first observation from ditch one is slightly larger than 100.

Analysing the data in a different way: multinomial logistic regression

One of the confusing aspects of statistics is that the same data can be analysed using different techniques. For the ditch data, we could use classification trees, multinomial logistic regression and discriminant analysis (and neural networks). Discriminant analysis and neural networks will be discussed later, and the case

study chapters give examples of using several statistical methods applied to the same data. However, none of the case study chapters use multinomial logistic regression, an extension of logistic regression. Although this technique should probably be explained in Chapter 6 after the logistic regression section, we decided to present it here. The reason for this is that data suitable for classification techniques, such as the ditch data, are also suitable for multinomial logistic regression.

We assume that the reader is familiar with logistic regression (Chapter 6). Suppose that the data consist of observations from two ditches (e.g., one and two). A possible logistic regression model is

$$\ln(\frac{P_i}{1-P_i}) = g(x_i)$$

P_i is the probability that an observation is from ditch one and $1 - P_i$ is the probability that it is not from ditch one. The function $g(x)$ can be of the form:

$$g(x) = \alpha + \beta_1 \text{ Total calcium} + \beta_2 \text{ Conductivity} + \beta_3 \text{ Depth}$$

where α is the population intercept and β_1, β_2 and β_3 the population slopes. Hence, in logistic regression the probability of an observation coming from ditch one divided by the probability that it is not from ditch one, is modelled as an exponential function of explanatory variables, such as total calcium, etc. In multinomial logistic regression, a similar model is used except that the response variable is allowed to have more than two classes. For the ditch data, we have five ditches so the response variable has five classes: 1, 2, 3, 4 and 5. In multinomial logistic regression, one of the classes is chosen as baseline, and by default most software chooses the first class as the baseline, which in this instance is ditch one. If there are only three explanatory variables, say (total) calcium, conductivity and depth, then the model is written as

$$\ln(\frac{P_{i2}}{P_{i1}}) = \alpha_2 + \beta_{12}Calcium_i + \beta_{22}Conductivity_i + \beta_{32}Depth_i$$
$$\ln(\frac{P_{i3}}{P_{i1}}) = \alpha_3 + \beta_{13}Calcium_i + \beta_{23}Conductivity_i + \beta_{33}Depth_i$$
$$\ln(\frac{P_{i4}}{P_{i1}}) = \alpha_4 + \beta_{14}Calcium_i + \beta_{24}Conductivity_i + \beta_{34}Depth_i$$
$$\ln(\frac{P_{i5}}{P_{i1}}) = \alpha_5 + \beta_{15}Calcium_i + \beta_{25}Conductivity_i + \beta_{35}Depth_i$$

The response variable has five classes, and therefore, the multinomial logistic regression model has four logistic regression equations. Each equation models the probability of an observation belonging to a particular ditch divided by the probability that it is from the baseline ditch (ditch one). This probability is modelled as an exponential function of the three explanatory variables. For each class, the regression parameters are estimated. This is done with maximum likelihood estimation, and all parameters are estimated simultaneously. To find the optimal model,

identical tools as used in logistic regression are available, for example deviance testing, AIC values, backward selection, *t*-values, etc. A backward selection, starting with a model containing all explanatory variables, was used to find the most optimal model. It contained the variables depth, ammoniacal nitrogen, total oxidised nitrogen, total calcium and total zinc. Hence, the optimal model is:

$$\ln(\frac{P_{i2}}{P_{i1}}) = \alpha_2 + \beta_{12}Depth_i + \beta_{22}AN_i + \beta_{32}ON_i + \beta_{42}Calcium_i + \beta_{52}Zinc_i$$

$$\ln(\frac{P_{i3}}{P_{i1}}) = \alpha_3 + \beta_{13}Depth_i + \beta_{23}AN_i + \beta_{33}ON_i + \beta_{43}Calcium_i + \beta_{53}Zinc_i$$

$$\ln(\frac{P_{i4}}{P_{i1}}) = \alpha_4 + \beta_{14}Depth_i + \beta_{24}AN_i + \beta_{34}ON_i + \beta_{44}Calcium_i + \beta_{54}Zinc_i$$

$$\ln(\frac{P_{i5}}{P_{i1}}) = \alpha_5 + \beta_{15}Depth_i + \beta_{25}AN_i + \beta_{35}ON_i + \beta_{45}Calcium_i + \beta_{55}Zinc_i$$

where *AN* and *ON* stand for ammoniacal nitrogen and total oxidised nitrogen, respectively. The estimated regression parameters are:

Class	Intercept	Depth	AN	ON	Calcium	Zinc
2	16.93	−7.11	5.62	−3.25	−7.98	14.93
3	245.62	278.71	34.11	−66.85	−145.48	−40.18
4	212.82	4.92	37.26	−31.76	−102.88	−76.54
5	62.96	20.43	34.13	−11.49	−32.35	−163.51

These estimated regression parameters indicate that depth is important to discriminate between observations from ditches three and one. This regression parameter is also relatively large for ditch five. Ammoniacal nitrogen has relatively large values for ditches three, four and five, but it is small for ditch two. Hence, the probability that an observation is in ditch two, divided by the probability that it is in ditch one, is not influenced by ammoniacal nitrogen. The same holds for oxidised nitrogen. Calcium is important for ditches three and four, and zinc for ditches four and five (relative to ditch one).

The magnitude of most estimated regression parameters indicate that the multinomial logistic regression model cannot discriminate the observations from ditches one and two, but it is able to do this for ditch one versus three, four and five. The important variables for this are depth, ammoniacal nitrogen, total oxidised nitrogen, total calcium and total zinc.

The significance of the regression parameters can be determined by individual Wald statistics. These are obtained by dividing the estimated regression parameters by their standard errors, and can be compared with a *t*-distribution; Wald values larger than 1.96 in an absolute sense indicate non-significance at the 5% level. Often, an explanatory variable is significant for one specific (in this case) ditch, but not for the other ditches. However, such a variable must either be included or excluded in the model. Setting an individual regression parameter β_{ij} to null is not possible. Therefore, it is perhaps better to look at the overall significance of a particular explanatory variable. Just as in logistic regression, this can be done by

comparing deviances of nested models with each other by using the Chi-square statistics. The AIC of the optimal model (presented above) is 77.16. Leaving out depth gives a deviance that is 42.15 larger compared with the optimal model. The difference of 42.14 is highly significant according to Chi-square test with 4 degrees of freedom. There are 4 degrees of freedom because 4 regression parameters are estimated for each explanatory variable. Results in Table 9.3 indicate that all explanatory variables are highly significant. Leaving out total calcium or depth causes the highest changes in deviance, and the highest AIC indicating that these are the two most important variables.

However, there are some concerns over this model and its conclusions. We started the analysis with 17 explanatory variables and used the AIC to decide on the optimal model presented above (the lower the AIC, the better the model). Month and year were the first variables to be removed. We then ended up where all the alternative models had nearly identical AICs: Differences were in the third digit. This indicates serious collinearity of the explanatory variables. To explain this, suppose we have a linear regression model with four explanatory variables x_1, x_2, x_3 and x_4. If x_4 is highly correlated with x_2 and x_3, then leaving out x_4 will give a model with a nearly identical model fit and AIC.

Only when working with 10 or fewer variables was this problem removed for the ditch data. To avoid these problems, a selection based on biological knowledge is required. If this is not done, one has to accept that total calcium might represent another gradient.

The results for the ditch data obtained by multinomial logistic regression (MLR) are similar to the classification trees. Yet, it provides a simpler alternative to discriminant analysis, classification trees or neural networks. MLR is a parametric approach (it is basically an extension of linear regression), and therefore more powerful. But if there are many explanatory variables, a pre-selection of the explanatory variables may be required to avoid problems with the numerical optimisation routines as many explanatory variables have to be estimated.

Table 9.3. Comparing deviances for the optimal multinomial logistic regression model. Log transformed data were used.

Leave out	df	AIC	LRT	Pr(Chi)
<none>		77.16		
Depth	4	111.31	42.14	<0.001
Ammoniacal Nitrogen	4	93.92	24.76	<0.001
Total Oxidised Nitrogen	4	92.76	23.60	<0.001
Total Calcium	4	125.58	56.42	<0.001
Total Zinc	4	89.06	19.90	<0.001