Chapter 9
Statistical Learning in Palaeolimnology

Gavin L. Simpson and H. John B. Birks

Abstract This chapter considers a range of numerical techniques that lie outside the familiar statistical methods of linear regression, analysis of variance, and generalised linear models or data-analytical techniques such as ordination, clustering, and partitioning. The techniques outlined have developed as a result of the spectacular increase in computing power since the 1980s. The methods make fewer distributional assumptions than classical statistical methods and can be applied to more complicated estimators and to huge data-sets. They are part of the ever-increasing array of 'statistical learning' techniques (*sensu* Hastie et al. (2011). The elements of statistical learning, 2nd edn. Springer, New York) that try to make sense of the data at hand, to detect major patterns and trends, to understand 'what the data say', and thus to learn from the data.

A range of tree-based and network-based techniques are presented. These are classification and regression trees, multivariate regression trees, bagged trees, random forests, boosted trees, multivariate adaptive regression splines, artificial neural networks, self-organising maps, Bayesian networks, and genetic algorithms. Principal curves and surfaces are also discussed as they relate to unsupervised self-organising maps. The chapter concludes with a discussion of current developments in shrinkage methods and variable selection in statistical modelling that can help in model selection and can minimise collinearity problems. These include principal components regression, ridge regression, the lasso, and the elastic net.

G.L. Simpson (✉) • H.J.B. Birks
Environmental Change Research Centre, University College London, Pearson Building, Gower Street, London WC1E 6BT, UK

H.J.B. Birks
Department of Biology and Bjerknes Centre for Climate Research, University of Bergen, PO Box 7803, Bergen N-5020, Norway

School of Geography and the Environment, University of Oxford, Oxford OX1 3QY, UK
e-mail: john.birks@bio.uib.no

H.J.B. Birks et al. (eds.), *Tracking Environmental Change Using Lake Sediments*, Developments in Paleoenvironmental Research 5, DOI 10.1007/978-94-007-2745-8_9,

Keywords Artificial neural networks • Bagging trees • Bayesian belief networks • Bayesian decision networks • Bayesian networks • Boosted trees • Classification trees • Data-mining • Decision trees • Genetic algorithms • Genetic programmes • Model selection • Multivariate adaptive regression splines • Multivariate regression trees • Principal curves and surfaces • Random forests • Regression trees • Ridge regression • Self-organising maps • Shrinkage • Statistical learning • Supervised learning • The elastic net • The lasso • Unsupervised learning

Introduction

This chapter considers a range of numerical techniques that lie outside the familiar statistical methods of linear regression, analysis of variance, and maximum-likelihood estimation or data-analytical techniques such as ordination or clustering. The techniques outlined here have developed as a result of the spectacular increase in computational power since the 1980s. They make fewer distributional assumptions than classical statistical methods and can be applied to more complicated estimators and to huge data-sets (Efron and Tibshirani 1991; Raymond et al. 2005; Witten and Frank 2005; Hastie et al. 2011). They allow the exploration and summary of vast data-sets and permit valid statistical inferences to be made without the usual concerns for mathematical tractability (Efron and Tibshirani 1991) because traditional analytical approaches are replaced by specially designed computer algorithms (Hastie et al. 2011).

Many of the techniques discussed in this chapter are part of the ever-increasing battery of techniques that are available for what Hastie et al. (2011) call 'statistical learning'. In this, the aim of the numerical analysis is to make sense of the relevant data, to detect major patterns and trends, to understand 'what the data say', and thus to learn from the data (Hastie et al. 2011). Statistical learning includes prediction, inference, and data-mining (Hastie et al. 2011). Data-mining (Ramakrishnan and Grama 2001; Witten and Frank 2005) usually involves very large data-sets with many objects and many variables. In conventional statistical analyses, the formulation of the hypotheses to be tested usually follows the observation of the phenomena of interest and associated data collection. In statistical learning and data-mining, observations on the numerical properties of previously collected data can also stimulate hypothesis generation (Raymond et al. 2005). Hypotheses generated in this manner can be tested using existing independent data (so-called test-data) or where these are inadequate, by further observations and data-collection. Data-mining within statistical learning is, like exploratory data analysis (Juggins and Telford 2012: Chap. 5), clustering and partitioning (Legendre and Birks 2012a: Chap. 7), and classical ordination (Legendre and Birks 2012b: Chap. 8), a data-driven hypothesis-generation tool as well as a data-summarisation technique. Classical statistical techniques such as regression (Birks 2012a: Chap. 2; Blaauw and Heegaard 2012: Chap. 12), temporal-series analysis (Dutilleul

et al. 2012: Chap. 16), and canonical ordination (Legendre and Birks 2012b: Chap. 8; Lotter and Anderson 2012: Chap. 18) are model-based hypothesis-testing techniques. Statistical learning and data-mining can thus play a critical role, not only in data-analysis but also in the design of future data-collection and research projects.

Statistical learning from large data-sets has provided major theoretical and computational challenges and has led to a major revolution in the statistical sciences (Efron and Tibshirani 1991; Hastie et al. 2011). As a result of this revolution, statistical learning tends now to use the language of machine learning of inputs which are measured or preset (Hastie et al. 2011). These have some influence on one of more outputs. In conventional statistical terminology, inputs are usually called predictors or independent exploratory variables, whereas outputs are called responses or dependent variables. In palaeolimnology, the outputs are usually quantitative variables (e.g., lake-water pH), qualitative (categorical 1/0) variables, (e.g., lake type), or ordered categorical variables (e.g., low, medium, high water-depth). The inputs can also vary in measurement type and are usually quantitative variables. In a typical palaeolimnological study, we have an outcome measurement, usually quantitative (e.g., lake-water pH) or categorical (e.g., fish present/absent) that we want to predict on a set of features (e.g., modern diatom assemblages). We have a training-set of data in which we observe the outcome and feature measurements for a set of objects (e.g., lakes). Using this training-set, we construct a prediction model or learner that will enable us to predict or infer the outcome for new unseen objects with their feature measurements (e.g., fossil diatom assemblages). A good learner is one that accurately predicts such an outcome. The distinction in output type has resulted in the prediction tasks being called regression when predicting quantitative outputs and classification when predicting qualitative outputs (Hastie et al. 2011).

Statistical learning can be roughly grouped into supervised or unsupervised learning. In supervised learning, the aim is to predict the value of an output measure based on a number of input measures. It is called supervised because the presence of the outcome measure(s) can guide the learning process. In unsupervised learning, there is no outcome measure, only input features. The aim is not to predict but to describe how the data are organised or clustered and to discern the associations and patterns among a set of input measures. Table 9.1 summarises the major data-analytical techniques used in palaeolimnology that are discussed by Birks (2012a: Chap. 2), Legendre and Birks (2012a, b: Chaps. 7 and 8), Blaauw and Heegaard (2012: Chap. 12), Juggins and Birks (2012: Chap. 14), Simpson (2012: Chap. 15), and Lotter and Anderson (2012: Chap. 18) in terms of supervised and unsupervised statistical learning.

This chapter outlines several tree-based and network-based data-analytical techniques that permit data-mining and statistical learning from large data-sets (over 500–1000 samples and variables) so as to detect the major patterns of variation within such data-sets, to predict responses to future environmental change, and to summarise the data as simple groups. These techniques are listed in Table 9.2 in relation to whether they are supervised or unsupervised statistical-learning techniques.

Table 9.1 Summary of the major analytical techniques used in palaeolimnology in terms of supervised and unsupervised statistical learning

	Type of statistical learning	
Numerical technique	Unsupervised	Supervised
Clustering (Chap. 7)	+	
K-means partitioning (Chap. 7)	+	
Ordination (e.g. PCA) (Chap. 8)	+	
Canonical ordination (Chaps. 8 and 18)		+
Weighted averaging regression and calibration (Chap. 14)		+
Weighted averaging partial least squares (Chap. 14)		+
Modern analogue technique (Chap. 15)		+
Discriminant analysis (Chap. 2)		+
Regression analysis (Chaps. 2 and 12)		+

Table 9.2 Summary of statistical machine-learning techniques in terms of supervised and unsupervised learning

	Type of statistical learning	
Machine-learning technique	Unsupervised	Supervised
Classification trees		+
Regression trees		+
Multivariate regression trees		+
Bagging trees		+
Boosted trees		+
Random forests	+	+
Multivariate adaptive regression splines		+
Artificial neural networks		+
Self-organising maps (SOMs)	+	
X-Y-fused SOMs, Bi-directional Kohonen networks, and super-organised maps		+
Bayesian belief networks		+
Bayesian decision networks		+
Genetic algorithms		+
Principal curves and surfaces	+	+
Shrinkage methods (ridge regression, the lasso, the elastic net)		+

Classification and Regression Trees

Dichotomous identification keys are common in fields such as biology, medicine, and ecology, where decisions as to the identification of individual specimens or the presence of disease are reduced to a set of simple, hierarchical rules that lead the user through the decision-making process. An example that will be familiar to many readers is the numerous plant identification keys used by field botanists. Computer-generated versions of these keys were first discussed in the social sciences arising

from the need to cope with complex data and scientific questions resulting from questionnaire responses leading to the Automatic Interaction Detection programme of Morgan and Sonquist (1963). Around the same time, similar tree-based methodologies were being developed independently in the machine-learning field (e.g., Quinlan 1993). The seminal work of Breiman et al. (1984) brought the main ideas and concepts behind tree-based models into the statistical arena. De'ath and Fabricius (2000) and Vayssieres et al. (2000) introduced classification and regression trees to the ecological literature. Fielding (2007) provides a simple introduction to tree-based modelling procedures in biology. Witten and Frank (2005) discuss classification and regression trees in the context of data-mining large, heterogeneous data-sets.

The general idea behind tree-based modelling is to identify a set of decision rules that best predicts (i) the 'identities' of a categorical response variable (a classification tree), or (ii) a continuous response variable (a regression tree). By 'best predicts', we mean minimises a loss function such as least-squares errors

$$D_N = \sum_{i=1}^{n} (y_i - \hat{y}_N) \tag{9.1}$$

where D_N is the deviance (impurity) of node N, y_i refers to the i^{th} observation in node N and $\hat{y}_N$ is the mean of y_i in node N. The total deviance (impurity) of a tree (D) consisting of N nodes is the sum of the deviances of the individual N nodes

$$D = \sum_{i=1}^{N} D_i \tag{9.2}$$

Building trees using the recursive binary partitioning method is by far the most commonly used technique. At each stage of fitting a tree, the algorithm identifies a split that best separates the observations in the current node into two groups; hence the binary part of the algorithm's name. The recursive partitioning aspect refers to the fact that each node is in turn split into two child nodes, and those child nodes are subsequently split, and so on in a recursive fashion (see Legendre and Birks 2012a: Chap. 7). We have glossed over many of the details of model fitting in the above description of recursive partitioning. We now expand on the detail of how trees are fitted to data.

The recursive partitioning algorithm starts with all the available data arranged in a single group or node (see also Legendre and Birks (2012a: Chap. 7) and Birks (2012b: Chap. 11) for other partitioning techniques that use this type of recursive algorithm (TWINSPAN, binary partitioning)). The data are a single matrix of n observations on m variables. The response variable y is also known; if y is a categorical variable (e.g., species presence/absence, or different species of pollen or diatom) then a classification tree will be fitted, whereas, if y is a continuous variable (e.g., lake-water pH or temperature) a regression tree is fitted. Each of the

m predictor variables is taken in turn and all possible locations for a split within the variable are assessed in terms of its ability to predict the response. For binary predictor variables, there is a single possible split (0 or 1). Categorical variables present a greater number of potential splits. An unordered categorical variable (e.g., red, green, blue) with number of levels (categories) L has $2(L - 1) - 1$ potential splits, whilst an ordered categorical variable (e.g., dry < moist < wet < very wet) conveys $L - 1$ potential splits. For continuous variables, imagine the observations lain out on a scale in ascending order of values of the variable. A split may be located between any pair of adjacent values. If there are U unique values, then each continuous variable conveys U – 1 potential splits. At each stage in the algorithm all of these potential split locations need to be evaluated to determine how well making each split predicts the response. Once the variable and split location that best predict the response have been identified, the data are separated into two groups on the basis of the split and the algorithm proceeds to split each of the two child groups (or nodes) in turn, using the same procedure as outlined above. Splitting continues until no nodes can be further subdivided or until some stopping criteria have been met, usually the latter. At this point fitting is complete and a full tree has been fitted to the data.

An important question remains; how do we quantify which split location best predicts the response? Splits are chosen on the basis of how much they reduce node impurity. For regression trees, the residual sums-of-squares (RSS, Eq. 9.1) about the child-node means or residual sums of absolute deviations (RSAD) from the child-node medians are used to measure node impurity, although the latter (RSAD) is of lesser utility with ecological data (De'ath and Fabricius 2000). Several alternative measures of node impurity (D_N) are commonly used in classification trees, including

(i) deviance

$$D_N = -2 \sum_k n_{Nk} \log(p_{Nk}) \tag{9.3.1}$$

(ii) entropy

$$D_N = -2 \sum_k p_{Nk} \log(p_{Nk}) \tag{9.3.2}$$

and

(iii) the Gini index

$$D_N = 1 - \sum_k p_{Nk}^2 \tag{9.3.3}$$

where D_N is the node impurity, n_{Nk} is the number of observations of class k in the N^{th} node, and p_{Nk} is the proportion of observations in the N^{th} node that are of type k. The overall node impurity evaluated for all possible splits is the sum of the impurities of the two groups formed by the split.

A final problem we face is how big a tree to grow? Above, we mentioned that the algorithm will continue until either it cannot split any node further (i.e., all nodes have zero impurity) or some stopping criteria are reached (e.g., fewer than five observations in a node). Such an approach will produce a large, complex tree that will tend to over-fit the observed data. Such a tree is unlikely to generalise well and will tend to produce poor out-of-sample predictions. A small tree, on the other hand, will be unlikely to capture important features in the response. Tree-size is a tuning parameter that controls the complexity of the fitted tree-model. The optimal tree-size can be determined from the data using a procedure known as cost-complexity pruning. The cost-complexity of a tree, CC, is computed as $\text{CC} = \text{T}_{\text{impurity}} + \alpha(\text{T}_{\text{complexity}})$, where $\text{T}_{\text{impurity}}$ is the impurity of the current tree over all terminal nodes, $\text{T}_{\text{complexity}}$ is the number of terminal leaves, and α a real number >0. α is the tuning parameter we aim to minimise in cost-complexity pruning, and represents the trade-off between tree-size and goodness-of-fit. Small values of α result in larger trees, whilst large values of α lead to smaller trees. Starting with the full tree, a search is made to identify the terminal node that results in the lowest CC for a given value of α. As the penalty α on tree complexity is increased, the tree that minimises CC will become smaller and smaller until the penalty is so great that a tree with a single node (i.e., the original data) has the lowest CC. This search produces a sequence of progressively smaller trees with associated CC. The solution now is to choose a value of α that is optimal in some sense. κ-fold cross-validation (Birks 2012a: Chap. 2; Juggins and Birks 2012: Chap. 14) is used to choose the value of α that has the minimal root mean squared error (RMSE). An alternative strategy is to select the smallest tree that lies within 1 standard error of the RMSE of the best tree.

Once the final tree is fitted, identified, and pruned, the data used to train the tree are passed down the branches to produce the fitted values for the response. In a regression tree, the predicted value is the mean of the observed values of the response in the terminal node that an observation ends up in. All the observations that are in the same terminal node therefore get the same fitted value. We say that regression trees fit a piece-wise constant model in the terminal nodes of the tree. The fitted values for classification trees are determined using a different procedure; the majority vote. The classes of all the observations in the same terminal node provide votes as to the fitted class for that node. The class that receives the highest number of votes is then the predicted class for all observations in that node.

Palaeolimnological data often contain missing data where, for one reason or another, a particular measurement on one or more samples is not available (Birks 2012a: Chap. 2; Juggins and Birks 2012: Chap. 14; Juggins and Telford 2012: Chap. 5). Deleting missing data reduces the number of samples available for analysis and may also introduce bias into the model if there is a systematic reason for the

'missingness' (Nakagawa and Freckleton 2008). Trees can handle missing data in the predictor variables in a number of ways. The first is to propagate a sample as far down the tree as possible until the variable used to split a node is one for which the data are missing. At that point we assign a fitted value as the average or majority vote of all the samples that pass through that particular node in the tree. The rationale for this is that we have sufficient information to make a partial prediction for a sample with missing data, but we are unable to provide a final prediction because of the missing data.

An alternative strategy is to use *surrogate splits* to decide how to propagate a sample with missing data further down a fitted tree. During the exhaustive search for split locations, a record is made of which alternative split locations provide a similar binary split of the data in the current node to that of the best split. Surrogate splits are those splits that provide the division of the samples in a node that most closely resembles the division made by using the best split location. When a sample with missing data is passed down a tree during prediction, the sample proceeds until it reaches a node where data on the splitting variable is missing. At this point, the best surrogate split is used to attempt to assign the sample to one of the two child nodes. If the variable used in the best surrogate split is also missing, the next best surrogate split is used, and so on until all available surrogate splits have been examined. If it is not possible to assign the sample to one of the two child nodes, then the sample is left in the current node and its predicted value is taken as the average or majority vote of samples passing through the node as previously described.

Surrogate splits are those that produce a similar binary division of a set of samples to that of the best split for a given node. There may also be split variables that reduce node impurity almost as much as the best split but do so using a different predictor variable and result in a different binary partition of a node. Such splits are known as *alternative splits*. Replacing the best split with an alternative split might lead to the fitting of a very different tree simply because of the legacy of having chosen one predictor over another early on in the tree-building process. Examination of the alternative splits can help provide a fuller description of the system under study by highlighting alternative models that explain the training data to a similar degree as the fitted tree.

High temperature combustion of coal and oil produces, amongst other pollutants and emissions, spheroidal carbonaceous particles (SCPs) (Rose 2001). Rose et al. (1994) studied the surface chemistry of a range of SCPs produced by burning coal, oil, and oil-shale fuels, and used linear discriminant analysis to identify linear combinations of surface chemistry variables that best distinguished between particles of the different fuel sources (see Birks 2012a: Chap. 2). To illustrate tree-based models, we re-analyse these data using a classification tree. The data consist of 6000 particles (3000 coal, 1000 oil, and 2000 oil-shale). A full classification tree was fitted using the rpart package (Therneau and Atkinson 2011) for the R statistical language and environment (R Core Development Team 2011). Apparent and ten-fold cross-validation (CV) relative error rates for trees of various size up to the full tree are shown in Fig. 9.1. The tendency for trees to over-fit the training data is illustrated nicely as the apparent relative error rate continues decreasing as

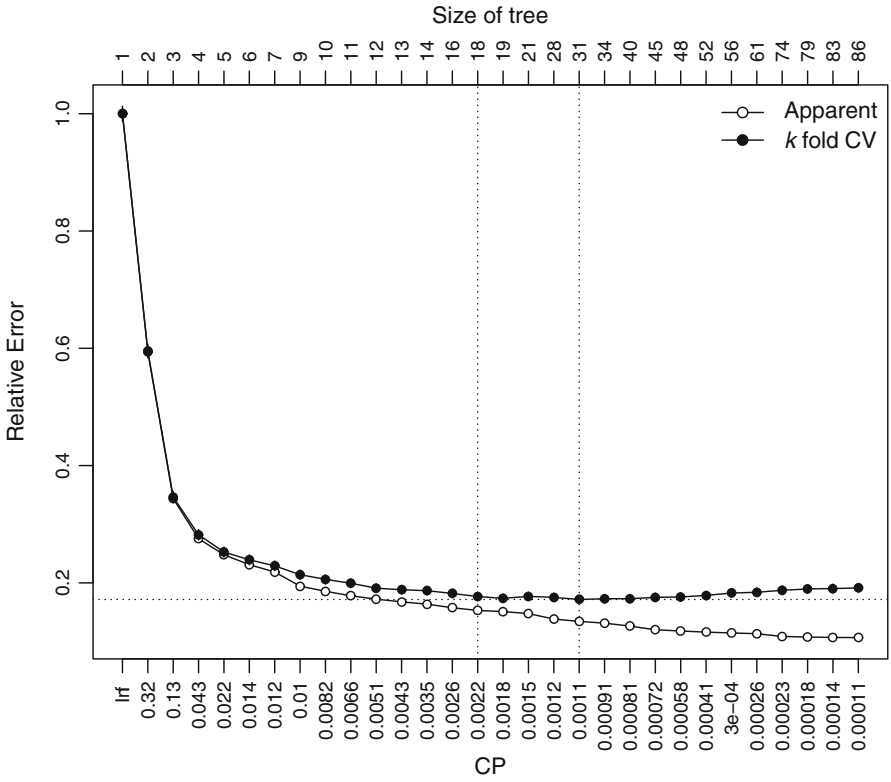

Fig. 9.1 Cost complexity and relative error for various sizes of classification trees fitted to the three-fuel spheroidal carbonaceous particle (SCP) example data. Apparent (*open circles*) and ten-fold cross-validated (CV; *filled circles*) relative error to the simplest tree (size one) are shown. The tree with the smallest CV relative error has 31 leaves, whilst the smallest tree within one standard error of the best tree has 18 leaves

the tree is grown and becomes more complex, whilst the ten-fold CV error rate stabilises after the tree contains 18 nodes or leaves and increases once the size of the tree exceeds 31 nodes. The values on the x-axis of Fig. 9.1 are the values of the cost-complexity parameter to which one must prune in order to achieve a tree of the indicated size. The best sized tree is one consisting of 31 nodes, with a CV relative error of 0.172 (CV standard error 0.007), and is indicated by the right-most vertical line. The smallest tree within one standard error of this best tree, is a model with 18 nodes and a CV relative error of 0.177 (CV standard error 0.007), and is indicated by the left-most vertical line.

Trees between sizes 18 and 48 all do a similar job, but we must guard against over-fitting the training data and producing a model that does not generalise well, so we select a tree size using the one standard-error rule and retain the tree with 18 nodes. This tree is shown in Fig. 9.2. The first split is based on Ca, with SCPs having

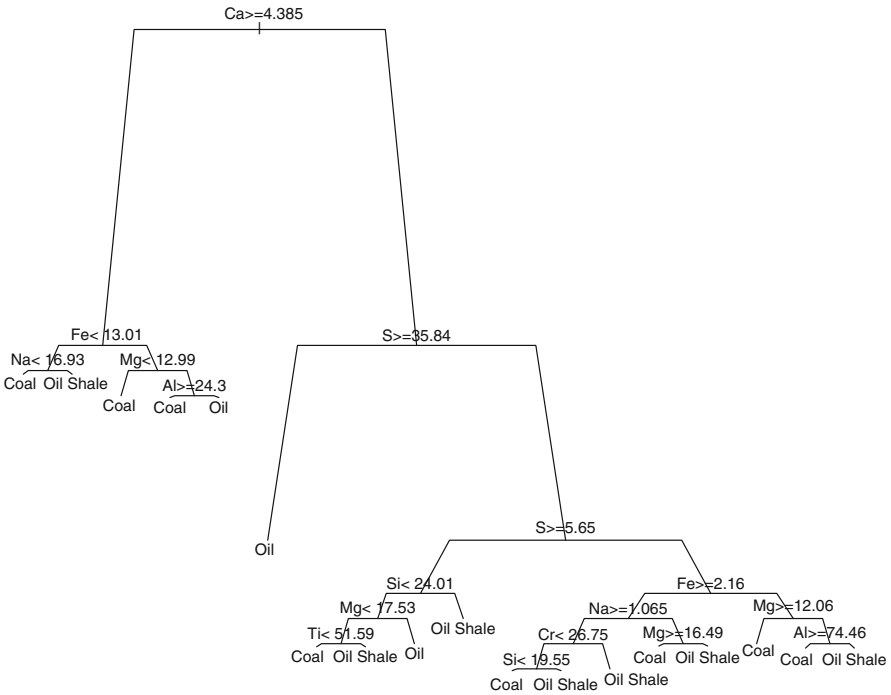

Fig. 9.2 Pruned classification tree fitted to the three-fuel spheroidal carbonaceous particle (SCP) example data. The predicted fuel types for each terminal node are shown, as are the split variables and thresholds that define the prediction rules

low amounts of Ca passing into the right-hand branch of the tree and those particles with Ca ≥ 4.385 passing into the left-hand branch. The right-hand branch is further split on the basis of S, with particles having $\geq$35.84 (and Ca < 4.385) classified as being produced by oil-fired power stations. By convention, the tree is plotted in such a way that the heights of the stems between nodes indicate the degree of importance attached to a split in terms of decreased node impurity. The first split on Ca and the split on S in the right-hand branch of the tree are clearly the most important rules for predicting SCP fuel type. The remaining splits are largely a fine tuning of these two main rules. The tree correctly classifies 5680 of the particles in the training data, giving an apparent error rate of 0.0533. Table 9.3 contains a summary of the predictions from the classification tree in the form of a confusion matrix. Individual error rates for the three fuel-types are also shown. Using ten-fold cross-validation to provide a more reliable estimate of model performance yields an error rate of 0.1 for the classification tree.

Of the machine-learning techniques described in this chapter, with the exception of artificial neural networks, trees are the most widely used method in palaeoecology and palaeolimnology, being employed in a variety of ways. Lindblah et al. (2002) used a classification tree to classify *Picea* pollen grains from three different species;

Table 9.3 Confusion matrix of predicted fuel type for the three-fuel classification tree

	Coal	Oil	Oil-shale	Error rate
Coal	2871	49	118	0.055
Oil	16	938	11	0.028
Oil-shale	113	13	1817	0.063

The rows in the table are the predicted fuel types for the 6000 spheroidal carbonaceous particles (SCPs) based on the majority vote rule. The columns are the known fuel-types. The individual fuel-type error rates of the classification tree are also shown. The overall error rate is 0.053

P. glauca, *P. mariana*, and *P. rubens* in eastern North America. Seven morphological measurements were made on approximately 170 grains of each species' pollen, and were used as predictor variables in the classification tree. An overall classification tree was fitted to assign grains to one of the three species, as well as individual species-specific binary classifications which aimed to predict whether a grain belonged to one of the three pollen taxa or not. Lindblah et al. (2003) used this approach to assign *Picea* pollen grains from a sediment core to one of the three species in late-glacial and Holocene sediments at a number of sites in New England, USA. Barton et al. (2011) employed a similar approach, using a classification tree to differentiate between pollen of red pine (*Pinus resinosa*) and jack pine (*Pinus banksiana*) in eastern North America. The habitat characteristics of sites where terrestrial snails, typical of full-glacial conditions in southern Siberia, are found have been described using a classification tree (Horsak et al. 2010). Other palaeoecological examples include Pelánková et al. (2008). CARTs are widely used in forestry (e.g., Baker 1993; Iverson and Prasad 1998, 2001; Iverson et al. 1999), ecology (e.g., Olden and Jackson 2002; Caley and Kuhnert 2006; Spadavecchia et al. 2008; Keith et al. 2010), biogeography (e.g., Franklin 1998, 2010), species-environment modelling (e.g., Iverson et al. 1999; Cairns 2001; Miller and Franklin 2002; Thuiller et al. 2003; Bourg et al. 2005; Kallimanis et al. 2007; Aho et al. 2011), limnology (e.g., Rejwan et al. 1999; Pyšek et al. 2010), hydrology (e.g., Carlisle et al. 2011), conservation biology (e.g., Ploner and Brandenburg 2003; Chytrý et al. 2008; Pakeman and Torvell 2008; Hejda et al. 2009), analysis of satellite data (e.g., Michaelson et al. 1994; DeFries et al. 2010), and landscape ecology (Scull et al. 2005).

Trees, whilst being inherently simple and interpretable, have a major drawback: the fitted model has high variance. A small change in the data can often lead to large changes in the form of the fitted tree, where a very different series of splits is identified. This makes trees somewhat difficult to interpret reliably; you might get a very different answer if you collected a different sample of data to fit the model. This is the downside of such a simple model structure. Solutions to this problem exist, and they all involve fitting many different trees to the data and averaging the predictions from each tree in some way. Collectively, these approaches are ensemble methods and include bagging, boosting, and random forests. We will discuss each of these techniques in later sections of this chapter.

Multivariate Regression Trees

The trees described in the previous section are univariate, dealing with a single response variable. Their extension to the multivariate response case is reasonably trivial (De'ath 2002; Larsen and Speckman 2004) yet the resulting technique is surprisingly versatile and is a useful counterpart to constrained ordination techniques such as redundancy analysis (RDA) and canonical correspondence analysis (CCA) (De'ath 2002; Legendre and Birks 2012a, b: Chaps. 7 and 8). Typically we have a response matrix of observations on m species for n sites. In addition, observations on p predictor variables (e.g., lake-water chemistry, climate-related variables) for the same n sites are available. In multivariate regression trees (MRT), the aim is to find a set of simple rules from the p predictor variables that best explains variation in the multivariate species-response matrix. Whilst MRT is closely related to constrained ordination, it can also be instructive to view MRT as a constrained clustering technique, where we partition the n observations in k groups or clusters on the basis of similar species composition *and* environment (Legendre and Birks 2012a: Chap. 7).

Regression trees use the concept of sum of squared errors as their measure of node impurity. This is inherently univariate, but can be extended to the multivariate case by considering sum of squared errors about the multivariate mean (centroid) of the observations in each node (De'ath 2002). In geometric terms, this amounts to being simply the sum of squared Euclidean distances of sites about the node centroid. In all other respects, the fitting and pruning of multivariate trees is the same as for univariate regression trees. However, the interpretation of multivariate trees requires additional techniques owing to the more complex nature of the response variable being modelled.

The Euclidean distance is often not suitable for use with ecological data as it focuses on absolute values, does not ignore or downweight double zeros, and imposes a linear framework on the analysis (Legendre and Birks 2012b: Chap. 8). MRTs can be adapted to work with any dissimilarity coefficient via direct decomposition of a supplied dissimilarity matrix to derive within-node sum of squared distances between node members. De'ath (2002) calls this method distance-based MRTs (db-MRTs). Note that in db-MRTs the within-node sum-of-squares are not computed with respect to the node centroid but instead with respect to pairs of samples. Minimising the sum of all pair-wise squared distances between samples within nodes is equivalent to computing the within-node sum-of-squares where the response data are species abundances. The response data in a db-MRT are a dissimilarity matrix computed using a chosen dissimilarity or distance coefficient (see Legendre and Birks 2012b: Chap. 8). As such, the raw data are not available during fitting to enable computation of the node centroids. Therefore, db-MRT uses the sum of pair-wise within-node distances as the measure of node impurity.

Univariate trees describe the mean response and a single tree-diagram can be used to convey in a simple fashion a large amount of information about the fitted model and the mean response. In MRTs, the mean response is multivariate, being

the mean abundance of each species for the set of samples defined by the tree nodes. A biplot is a natural means for displaying the mean response. De'ath (2002) suggests that principal component analysis (PCA) (Legendre and Birks 2012b: Chap. 8) be used as the base plot, with PCA being performed on the fitted values of the response (the mean abundance for each species in each of the MRT terminal nodes). The observed data are also shown on the biplot. The samples themselves can thus be located in the biplot about their node centroid. Species loadings can be added to the biplot either as simple PCA loadings (species scores), in which case they are represented as biplot arrows, or as a weighted mean of the node means, in which case the species are represented as points in ordination space. The branching tree structure can also be drawn on the biplot to aid visualisation.

Earlier, we mentioned that MRTs can be viewed as a constrained form of cluster analysis. From the description of the technique we have provided, it should be clear that MRTs find k groups of samples that have the lowest within-group dispersion for the k^{th} partition. If the constraints or predictor variables were not involved in the analysis then MRTs would be a way of fitting a minimum variance-cluster analysis (Legendre and Birks 2012a: Chap. 7). However, because the constraints are included in a MRT analysis, the identification of the group structure in the data is supervised, with groups being formed by partitioning the response variables on the basis of thresholds in the constraints. Chronological or constrained clustering and partitioning have a long tradition in palaeoecology and several numerical approaches to the problem of zoning stratigraphical data have been suggested (e.g., Gordon and Birks 1972, 1974; Gordon 1973; Birks 2012b: Chap. 11; Legendre and Birks 2012a: Chap. 7). One proposed solution to the problem is the binary divisive procedure using the sum-of-squares criterion (SPLITLSQ) method of Gordon and Birks (1972) which fits a sequence of b boundaries to the stratigraphical diagram, where $b \in \{1, 2, \ldots, n-1\}$. The boundaries are placed to minimise the within-group sums-of-squares of the groups formed by the boundaries. The process is sequential or hierarchical; first the entire stratigraphical sequence is split into two groups by the placement of a boundary that most reduces within-group sums of squares. Subsequently, one of the groups formed by positioning the first boundary is split by the placement of a second boundary, and so on until b boundaries have been positioned. The SPLITLSQ approach is exactly equivalent to the MRT when the Euclidean distance is used (see Legendre and Birks 2012b: Chap. 8). The utility of the MRT as a means of zoning stratigraphical diagrams is that the cross-validation procedure provides a simple way to assess the number of groups into which the sequence should be split.

To illustrate MRTs and to emphasise the constrained clustering nature of the technique, we turn to the classic Abernethy Forest pollen data of Birks and Mathewes (1978) (see Birks and Gordon 1985 for details). We fit a MRT to the pollen percentage data without transformation. A plot of the apparent and cross-validated relative error as a function of the cost-complexity parameter (or tree-size) for the MRT-fit to the Abernethy Forest data is shown in Fig. 9.3. Of the tree-sizes considered, the minimum cross-validated relative error is achieved by a tree with

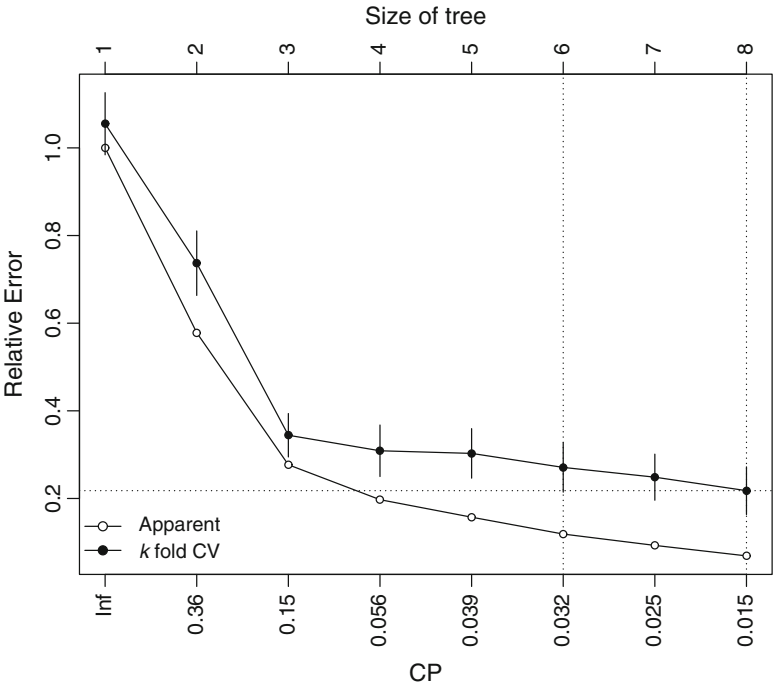

Fig. 9.3 Cost complexity and relative error for various sizes of multivariate regression trees fitted to the late-glacial and early-Holocene Abernethy Forest pollen sequence. Apparent (*open circles*) and ten-fold cross-validated (CV; *filled circles*) relative error to the simplest tree (size one) are shown. The tree with the smallest CV relative error has 8 leaves, whilst the smallest tree within one standard error of the best tree has 6 leaves

eight terminal nodes (seven splits), whilst the one standard-error rule would select the six-node sized tree. We select the latter and show the pruned, fitted MRT in Fig. 9.4. The first split is located at 7226 radiocarbon years BP and the second at 9540 BP. These two splits account for much larger proportions of the variance in the pollen data than the subsequent splits, as shown by the heights of the bars below the splits. The bar charts located at the terminal nodes in Fig. 9.4 provide a representation of the mean abundance for each pollen type over the set of samples located in each terminal node. A better representation of the mean response is given by the tree biplot (Fig. 9.5). The first split separates the samples dominated by *Pinus*, *Quercus*, and *Ulmus* pollen from the other samples, and is aligned with the first principal component (PC) axis. The second PC axis separates a group of samples characterised by *Juniperus*, *Corylus*, and *Betula* pollen.

MRTs have proved a relatively popular machine-learning technique in the palaeoenvironmental sciences. Davidson et al. (2010a) employed MRT to infer simultaneously the densities of zooplanktivorous fish and aquatic macrophytes from cladoceran species composition. The MRT was applied to a training-set of 39 lakes,

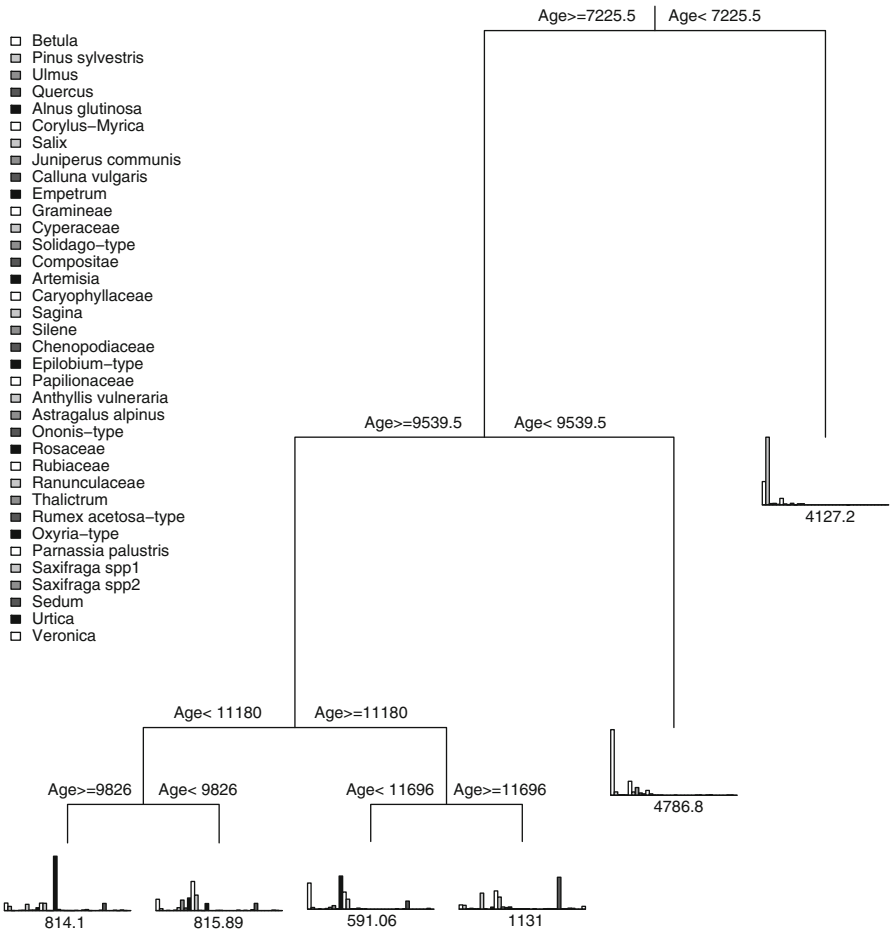

Fig. 9.4 Pruned multivariate regression tree (MRT) fitted to the late-glacial and early-Holocene Abernethy Forest pollen sequence. The major stratigraphic zones in the pollen stratigraphy are identified by the MRT. The bar charts in the terminal nodes describe the abundance of the individual species in each zone. The numbers beneath the bar charts are the within-zone sums of squares

using the cladoceran taxa as response variables and 14 environmental variables as predictors. The resulting pruned MRT had six clusters of samples resulting from splits on zooplanktivorous fish density (ZF) and plant volume infestation (PVI) and explained 67% of the variance in the species data. Davidson et al. (2010b) then applied their MRT model in conjunction with redundancy analysis (Legendre and Birks 2012b: Chap. 8) to cladoceran assemblages from a sediment core from Felbrigg Lake to investigate past changes in fish abundance and macrophyte abundance. Herzschuh and Birks (2010) used MRT in their investigation of the indicator value of Tibetan pollen and spore taxa in relation to modern vegetation

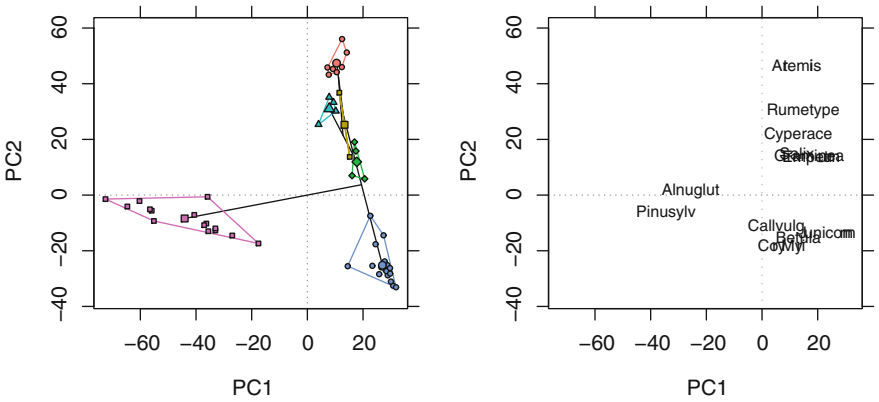

Fig. 9.5 Principal component analysis (PCA) display of the multivariate regression tree (MRT) fitted to the late-glacial and early-Holocene Abernethy Forest pollen sequence (*left*). The terminal nodes of the MRT are shown by large *open circles*, joined by line segments that represent the hierarchy. The samples within each node are differentiated by symbol shape and colour. Species scores (*right*) for the most common taxa in the Abernethy data-set are positioned using weighted averages instead of weighted sums

and climate. Their analysis showed that annual precipitation was the most important climatic variable in grouping the pollen counts in modern assemblages, with a value of ∼390 mm precipitation identified as a critical threshold. Temperature was identified as then playing a role in separating the two groups of pollen assemblages resulting from the 'low' and 'high' precipitation split. The resulting MRT produced four pollen groupings associated with four climate types: dry and warm, dry and cool, wet and warm, and wet and cool. Other palaeolimnological examples include Amsinck et al. (2006) and Bjerring et al. (2009). Surprisingly, MRTs do not appear to have been widely used in ecology or biogeography except in a recent biogeographical study by Chapman and Purse (2011).

Other Types of Tree-Based Machine-Learning Methods (Bagging, Boosted Trees, Random Forests, Multivariate Adaptive Regression Splines)

Earlier, we mentioned the instability problem of single-tree based models, which can be viewed as sampling uncertainty in the model outputs. If we were to take a new sample of observations and fit a model to those and use it to predict for a test-set of observations, we would get a different set of predictions for the test-set samples. If this process were repeated many times for each observation in the test-set, a posterior distribution of predicted values would be produced. The mean of each of these posterior distributions can be used as the predictions for the test-set samples, and in addition, the standard error of the mean or the upper and lower 2.5th

quantiles can be used to form uncertainty estimates on the predictions. In general, however, taking multiple samples of a population is not feasible. Instead, we can use the training-set observations themselves to derive the posterior distributions using bootstrap re-sampling (see Birks 2012a: Chap. 2; Juggins and Birks 2012: Chap. 14; Simpson 2012: Chap. 15). Such approaches are often termed ensemble or committee methods.

This general description applies neatly to bagging and random forests, but less so to the technique of boosting and not at all to multivariate adaptive regression splines (MARS: Friedman 1991). Boosting employs many trees in a manner similar to bagging and random forests, but each additional tree focuses on the hard-to-predict observations in the training-set, thereby learning different features in the data (Schapire 1990; Freund 1995; Friedman et al. 2000; Friedman 2001; Hastie et al. 2011). MARS, on the other hand, relaxes the piece-wise constant models fitted in the nodes of regression trees to allow piece-wise linear functions and in doing so discards the hierarchical nature of the simple tree structure (Friedman 1991). Whilst the switch to piece-wise linear functions is not that dramatic in itself, MARS employs these piece-wise linear functions in a flexible way combining several such functions to fit regression models capable of identifying complex, non-linear relationships between predictor variables and the response (Friedman 1991). Prasad et al. (2006) provide a comprehensive comparison of these newer tree techniques.

Bagging

Bagging, short for bootstrap aggregating, is a general method, proposed by Breiman (1996), for producing ensembles for any type of model, though it has typically been applied to tree-based models. In palaeolimnology, when we perform bootstrapping (Efron and Tibshirani 1993) to estimate calibration-function errors and provide sample-specific errors (Birks et al. 1990; Juggins and Birks 2012: Chap. 14; Simpson 2012: Chap. 15), we are using bagging. The idea is quite simple and draws upon the power of Efron's (1979) bootstrap to produce a set or ensemble of models that replicate the uncertainty in the model arising from sampling variation.

In bagging, a large number of models, b, is produced from a single training-set by drawing a bootstrap sample from the training-set with which to fit each model. Recall that a bootstrap sample is drawn from the training-set with replacement, and that, on average, approximately two thirds of the training-set samples will be included in the bootstrap sample. The remaining samples not selected for the bootstrap sample are set to one side and are known as the out-of-bag (OOB) samples. A tree model without pruning (or any other model) is fitted to this bootstrap sample. The fitted tree is used to generate predictions for the OOB samples, which are recorded, as are the fitted values for the in-bag samples. This procedure is repeated b times to produce a set of b trees. The sets of fitted values for each training-set sample are averaged to give the bagged estimates of the fitted values. In the case of a regression tree the mean is used to average the fitted values, whilst the majority

Table 9.4 Confusion matrix of predicted fuel type for the bagged three-fuel classification tree (number of trees = 500)

	Coal	Oil	Oil-shale	Error rate
Coal	2794	50	116	0.056
Oil	18	930	6	0.025
Oil-shale	188	13	1878	0.100

The rows in the table are the predicted fuel types for the 6000 spheroidal carbonaceous particles (SCPs) based on the majority vote rule over the ensemble of trees. The columns are the known fuel-types. The individual fuel-type error rates of the bagged classification tree are also shown. The overall error rate is 0.066

vote rule is used for classification trees, where each of the b bagged trees supplies a vote as to the fitted class for each observation, and the class with the largest number of votes is selected as the fitted class for that observation. Alternatively, posterior class probabilities can be produced for each observation from the set of bagged classification trees (though not using the relative proportions of votes for each class) and the class with the highest posterior probability is taken as the predicted class. The same procedures are used to provide bagged predictions for new observations not included in the training-set.

Table 9.4 shows the confusion matrix for a bagged classification tree model applied to the three fuel-type SCP data analysed earlier. Error rates for the three fuel-types are also shown. These statistics were computed using the OOB samples and are honest, reliable estimates of the true error rates as opposed to those for the single classification tree we produced earlier. The overall error rate for the bagged model is 0.066 (6.6%), a modest improvement over the single classification tree (k-fold cross-validation error = 0.1). Table 9.4 contains a summary of the predictions from the bagged classification tree. The predictions for the Coal and Oil classes are very similar to the apparent predictions from the classification tree (Table 9.3). The main difference between the bagged tree and the single tree is in their abilities to discriminate between coal- and oil-shale-derived particles, with the single tree being somewhat over-optimistic in its ability to discriminate these two fuel-types. The bagged tree gives a more honest appraisal of its ability to discriminate; the error rate for the oil-shale class is similar to the overall k-fold CV error rate of the classification tree.

Model error for bagged regression trees can be expressed as RMSE

$$\text{RMSE} = \sqrt{\sum_{i=1}^{n} (\hat{y}_i - y_i) / n} \tag{9.4}$$

using the fitted values, but this is an apparent error statistic and is not reflective of the real expected error. Instead, we can compute the equation above for each observation using only the OOB predictions. The OOB predictions are for the samples not used to fit a given tree. As such they provide an independent estimate of the model

error when faced with new observations. A similar quantity can be computed for classification trees and is known as the error rate (number of misclassifications / number of OOB observations). Again, only the OOB samples should be used in generating the error rate of the model to achieve an honest error estimate.

How does bagging help with the tree instability problem? Individual trees are unstable and hence have high variance. Model uncertainty is a combination of bias (model error or mean squared error: MSE) and variance (the variation of model estimates about the mean). Bagging improves over single tree models because averaging over b trees reduces the variance whilst leaving the bias component unchanged, hence the overall model uncertainty is reduced. This does not hold for classification trees, however, where squared loss is not appropriate and 0–1 loss is used instead, as bias and variance are not additive in such cases (Hastie et al. 2011). Bagging a good classification model can make that model better but bagging a bad classification model can make the model worse (Hastie et al. 2011).

The improved performance of bagged trees comes at a cost; the bagged model loses the simple interpretation that is a key feature of a single regression tree or classification tree. There are now b trees to interpret and it is difficult, though not impossible, to interrogate the set of trees to determine the relative importance of predictors. We discuss this in the following section on the related technique of random forests.

Random Forests

With bagged trees, we noted that reduction in model uncertainty is achieved through variance reduction because averaging over many trees retains the same bias as that of a single tree. Each of the b trees is statistically identically distributed, but not necessarily independent because the trees have been fitted to similar data-sets. The degree of pair-wise correlation between the b trees influences the variance of the trees and hence the uncertainty in the model; the larger the pair-wise correlation, the larger the variance. One way to improve upon bagging is to reduce the correlation between the b trees. Random forests (Breiman 2001) is a technique that aims to do just that. Prasad et al. (2006) and Cutler et al. (2007) provide accessible introductions to random forests from an ecological view-point, whilst Chap. 15 of Hastie et al. (2011) provides an authoritative discussion of the method.

The key difference between bagging as described above and random forests is that random forests introduces an additional source of stochasticity into the model-building process (Breiman 2001), which has the effect of de-correlating the set of trees in the ensemble of trees or the forest (Hastie et al. 2011). The tree-growing algorithm, as we saw earlier, employs an exhaustive search over the set of available explanatory variables to find the optimal split criterion to partition a tree node into two new child nodes. In standard trees and bagging, the entire set of explanatory variables is included in this search for splits. In random forests, however, the set of explanatory variables made available to determine each split is a randomly

determined, usually small, subset of the available variables. As a result, each tree in the forest is grown using a bootstrap sample, just as with bagging, and each split in each and every tree is chosen from a random subset of the available predictors.

The number of explanatory variables chosen at random for each split search is one of two tuning parameters in random forests that needs to be chosen by the user. The number of explanatory variables used is referred to as m and is usually small. For classification forests, the recommended value is $\lfloor \sqrt{p} \rfloor$, and $\lfloor p/3 \rfloor$ is suggested for regression forests, where the brackets represent the floor (rounding down to the nearest integer value), and p is the number of explanatory variables (Hastie et al. 2011). The recommended minimum node size, the size in number of observations beyond which the tree growing algorithm will stop splitting a node, is one and five for classification and regression forests, respectively (Hastie et al. 2011). This has the effect of growing large trees to each bootstrap sample with the result that each individual tree has low bias.

The trees are not pruned as part of the random-forest algorithm; the intention is to grow trees until the stopping criteria are met so that each tree in the forest has a low bias. Each of the individual trees is therefore over-fitted to the bootstrap sample used to create it, but averaging over the forest of trees effectively nullifies this over-fitting. It is often claimed that random forests do not over-fit. This is not true, however, and, whilst the details of why this is the case are beyond the scope of this chapter, it is worth noting that as the number of fully grown trees in the forest becomes large, the average of the set of trees can result in too complex a model and consequently suffer from increased variance. Section 15.3.4 of Hastie et al. (2011) explains this phenomenon, but goes on to state that using fully grown trees tends not to increase the variance too much and as such we can simplify our model building by not having to select an appropriate tree depth via cross-validation.

Random forests suffer from the same problem of interpretation as bagged trees owing to the large number of trees grown in the forest. Several mechanisms have been developed to allow a greater level of interpretation for random forests. We will discuss two main techniques: (i) variable importance measures and (ii) proximity measurements.

The importance of individual predictors is easy to identify with a single tree as the relative heights of the branches between nodes represent this, and alternative and surrogate splits can be used to form an idea of which variables are important at predicting the response and which are not. With the many trees of the bagged or random forest ensemble this is not easy to do by hand, but is something that the computer can easily do as it is performing the exhaustive search to identify splits. Two measures of variable importance are commonly used: (i) the total decrease in node impurity averaged over all trees and (ii) a measure of the mean decrease in the model's ability to predict the OOB samples before and after permuting the values of each predictor variable in turn (Prasad et al. 2006). Recall that node impurity can be measured using several different functions. In random forests, the first variable importance measure is computed by summing the total decrease in node impurity for each tree achieved by splitting on a variable and

averaging by the number of trees. Variables that are important will be those that make the largest reductions in node impurity. The accuracy importance measure is generated by recording the prediction error for the OOB samples for each tree, and then repeating the exercise after randomly permuting the values of each predictor variable. The difference between the recorded prediction error and that achieved after permutation is averaged over the set of trees. Important variables are those that lead to a large increase in prediction error when randomly permuted. The mean decrease in node impurity measure tends to be the most useful of the two approaches because there is often a stronger demarcation between important and non-important variables compared with the decrease in accuracy measure, which tends to decline steadily from important to non-important predictors.

A novel feature of random forests is that the technique can produce a proximity matrix that records the dissimilarity between observations in the training-set. The dissimilarity between a pair of observations is based on the proportion of times the pair is found in the same terminal node over the set of trees in the model. Samples that are always found in the same terminal node will have zero dissimilarity and likewise those that are never found in the same node will have dissimilarity of 1. This matrix can be treated as any other dissimilarity matrix and ordinated using principal coordinate analysis (see Legendre and Birks 2012b: Chap. 8) or non-metric multidimensional scaling (see Legendre and Birks 2012b: Chap. 8) or clustered using hierarchical clustering or K-means partitioning (see Legendre and Birks 2012a: Chap. 7).

We continue the three-fuel SCP example by analysing the data using random forests. Five hundred trees were grown using the recommended settings for classification forests; minimum node size of five, $m = \lfloor\sqrt{21}\rfloor = 4$. Figure 9.6 shows the error rate for the OOB samples of the random-forest model as additional trees are added to the forest. The overall OOB error rate and that of each of the three fuel-types is shown. Error rates drop quickly as additional trees are added to the model, and stabilise after 100–200 trees have been grown. Table 9.5 shows the confusion matrix and error rates for the individual fuel-types for the random-forest model. The overall error rate is 6.6%. Figure 9.7 shows the variable importance measures for the overall model, with Ca and S, and, to a lesser extent, Si, having the largest decrease in node impurity as measured by the Gini coefficient. A similar result is indicated by the decrease in the accuracy measure, although it is more difficult to identify clear winners using this index. These same variables are also important for predicting the individual fuel-types, where Fe and Mg also appear as important indicators for the Oil and Oil-shale fuel-types (Fig. 9.8).

Random forests, whilst having recently been used in ecology as a method for broad-scale prediction of species presence/absence or ecological niche modelling (Iverson and Prasad 2001; Benito Garzón et al. 2006, 2008; Lawler et al. 2006; Rehfeldt et al. 2006; Cutler et al. 2007; Peters et al. 2007; Brunelle et al. 2008; Iverson et al. 2008; Williams et al. 2009; Chapman 2010; Chapman et al. 2010; Franklin 2010; Dobrowski et al. 2011; Vincenzi et al. 2011), have been little used in palaeoecology, which is surprising given the accuracy, simplicity, and speed of the method relative to other machine-learning techniques. Brunelle et al. (2008) use

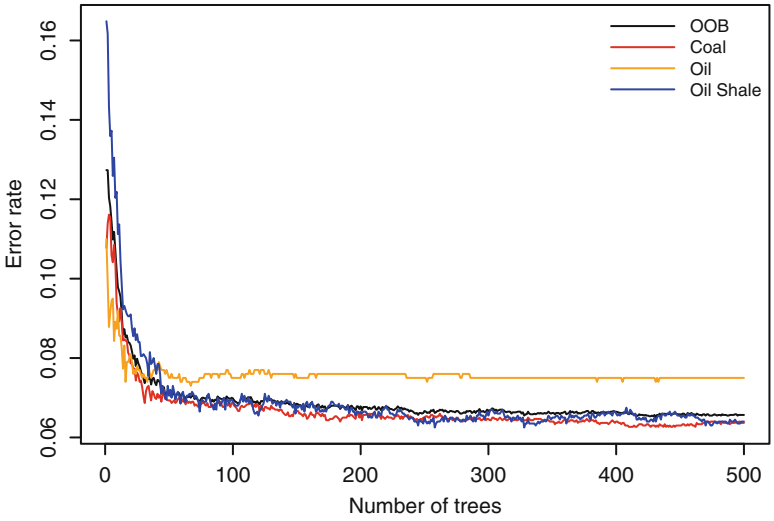

Fig. 9.6 Error rate for the classification random forest fitted to the three-fuel spheroidal carbonaceous particle (SCP) example data as trees are added to the ensemble. The *black line* is the overall error rate for the random forest model. The remaining lines are the error rates for the individual fuel types. The error rates are determined from the out-of-bag (OOB) samples for each tree

Table 9.5 Confusion matrix of predicted fuel type for the three-fuel random forest model (number of trees = 500)

	Coal	Oil	Oil-shale	Error rate
Coal	2809	8	183	0.064
Oil	56	925	19	0.075
Oil-shale	128	0	1872	0.064

The rows in the table are the predicted fuel types for the 6000 spheroidal carbonaceous particles (SCPs) based on the majority vote rule over the ensemble of trees. The columns are the known fuel-types. The individual fuel-type error rates of the random forest classifier are also shown. The overall error rate is 0.066

random forests to investigate the climatic variables associated with the presence, absence, or co-occurrence of lodgepole and whitebark pine in the Holocene, whilst Benito Garzón et al. (2007) employ random forests to predict tree species distribution on the Iberian Peninsula using climate data for the last glacial maximum and for the mid-Holocene. Other palaeoecological examples include Goring et al. (2010) and Roberts and Hamann (2012). Random forests are widely used in genomic and bioinformatical data-analysis (e.g., Cutler and Stevens 2006; van Dijk et al. 2008) and epidemiology (e.g., Furlanello et al. 2003).

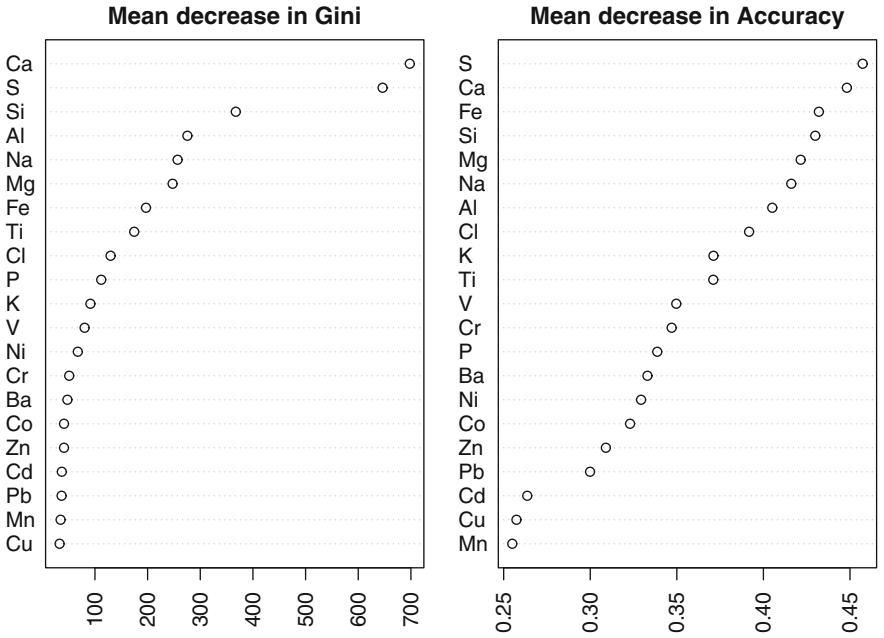

Fig. 9.7 Variable importance measures for the classification forest fitted to the three-fuel spheroidal carbonaceous particle (SCP) example data, showing the mean decrease in the Gini index when each variable is not included in the model (*left*) and the mean decrease in accuracy measure (*right*). See main text for details

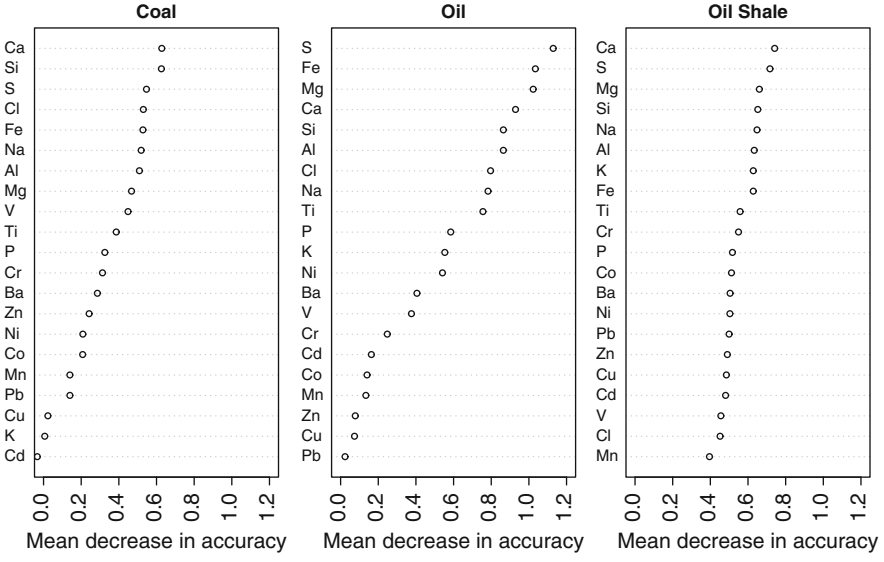

Fig. 9.8 Mean decrease in accuracy for individual fuel types in the spheroidal carbonaceous particle (SCP) example data, determined from the fitted classification forest

Boosting

In the discussion of bagging and random forests, we saw that modelling involves a trade-off between the bias and the variance of the fitted model. Bagging and random forests attempt to reduce the variance of a fitted model through the use of an ensemble of trees in place of the single best tree. These techniques do not reduce the bias of the fitted model. Boosting, a loosely related technique, uses an ensemble of models (in our case trees) to reduce both the bias *and* the variance of a fitted model. Boosting is an incredibly powerful technique that today relates to a whole family of approaches. Here we restrict our discussion to gradient boosting, which also goes by the name multiple additive regression trees (MART), and its variant stochastic gradient boosting. Hastie et al. (2011) contains a lengthy discussion of boosting and is essential reading for anyone attempting to use the technique for modelling data. Elith et al. (2008) is a user friendly, ecologically-related introduction to both the theory and practice of fitting boosting models (see also Witten and Frank 2005; De'ath 2007).

As with bagging and random forests, boosting begins from the realisation that it is easier to identify and average many rough predictors than it is to find one, all encompassing, accurate, single model. The key difference with boosting is that it is sequential; additional models are added to the ensemble with the explicit aim of trying to improve the fit to those observations that are poorly modelled by the previous trees already included in the model. With bagging and random forests each new tree is fitted to a bootstrap sample of the training data with no recourse to how well any of the previous trees did in fitting observations. As such, bagging and random forests do not improve the bias in the fitted model: they just attempt to reduce the variance. Boosting, in contrast, aims to reduce the bias in the fitted model by focussing on the observations in the training data that are difficult to model, or are poorly modelled, by the preceding set of trees. In the terminology of Hastie et al. (2011), boosting is a forward, stage-wise procedure.

Our discussion proceeds from the point of view of regression; this includes models for discrete responses such as logistic or multinomial regression thus encompassing classification models (Birks 2012a: Chap. 2). We have already mentioned loss functions, a function or measure, such as the deviance, that represents the loss in predictive power due to a sub-optimal model (Elith et al. 2008). Boosting is an iterative computational technique for minimising a loss function by adding a new tree to the model that at each stage in the iteration provides the largest reduction in loss. Such a technique is said to descend the gradient of the loss function, something known as functional gradient descent. For boosted regression trees, the algorithm starts by fitting a tree of a known size to the training data. This model, by definition, provides the largest reduction in the loss function. In subsequent iterations, a tree is fitted to the *residuals* of the previously fitted trees, which maximally reduces the loss function. As such, subsequent trees are fitted to the variation that remains unexplained after considering the previous set of trees. Each subsequent tree added to the ensemble has as its focus those poorly modelled

observations that are not well fitted by the combination of previous trees, and as such can have quite different structures incorporating different variables and splits into the tree. Boosting is a stage-wise procedure because the preceding trees in the ensemble are not altered during the current iteration, which contrasts with step-wise procedures where the entire model is updated at each iteration (step-wise regression procedures, for example). Elith et al. (2008) summarise the boosted ensemble as a "linear combination of many trees... that can be thought of as a regression model where each term is a tree."

A further important aspect of boosting is the concept of regularisation. The logical conclusion of the boosting algorithm if no restriction on the learning rate was imposed is that the sequence of trees could be added until the training-set samples were perfectly explained and the model was hopelessly over-fitted to the data. In the standard regression setting, the number of terms in the model is often constrained by dropping out covariates (variables) or functions thereof, via a set of step-wise selection and elimination steps. A better, alternative approach is to fit a model with many terms and then down-weight the contributions of each term using shrinkage, as is done in ridge regression (Hoerl and Kennard 1970) or the lasso (Tibshirani 1996) (see below). With ridge regression or the lasso, the shrinkage that is applied is global, acting on the full model. In boosting, shrinkage is applied incrementally to each new tree as it is added to the ensemble and is controlled via the learning rate, *lr*, which, together with the number of trees in the ensemble, *tr*, and tree complexity, *tc* (the size of the individual trees), form the set of parameters optimised over by boosted trees.

Stochasticity was introduced into bagging and random forests through the use of bootstrap samples, where it introduces randomness that can improve the accuracy and speed of model fitting and help to reduce over-fitting (Friedman 2002) at the expense of increasing the variance of the fitted values. In boosting, stochasticity is introduced through randomly sampling a fraction, *f*, of the training samples at each iteration. This fraction is used to fit each tree. *f* lies between 0 and 1 and is usually set to 0.5 indicating that 50% of the training observations are randomly selected to fit each tree. In contrast to bagging and random forests, the sampling is done without replacement.

Recent work (Elith et al. 2008) on boosting has shown that it works best when learning is slow and the resulting model includes a large ($>1{,}000$) number of trees. This requires a low learning rate, say $lr = 0.001$. We still need a way of being alerted to over-fitting the model so as to guide how many trees should be retained in the ensemble. If using stochastic boosting, each tree has available a set of OOB samples with which we can evaluate the out-of-sample predictive performance for the set of trees up to and including the current tree. A plot of this predictive performance as new trees are added to the ensemble can be used to guide as to when to stop adding new trees to the ensemble. If stochastic boosting is not being used, other methods are required to guide selection of the number of trees. An independent test-set can also be employed, if available, in place of the OOB samples. Alternatively, *k*-fold cross-validation (CV) can be used if computational time and storage are not issues, and there is evidence that this procedure performs best for a wide range of test data-

sets (Ridgeway 2007). In *k*-fold cross-validation, the training data are divided into *k* subsets of (approximately) equal size. A boosting model is fitted to the *k*-l subsets and the subset left out is used as an independent test-set. A large boosting model is fitted and the prediction error for the left-out subset is recorded as the number of trees in the model increases. This process is repeated until each of the *k* subsets has been left out of the model-building process, and the average CV error is computed for a give number of trees. We take as the number of trees to retain in the model as that number of trees with lowest CV error.

Tree complexity, *tc*, is a tuning parameter in boosting; it affects the learning rate required to yield a large ensemble of trees, and also determines the types of interactions that can be fitted by the final model. Earlier, we saw how trees were able to account flexibly for interactions between predictor variables by allowing additional splits within the separate nodes of the tree, namely the interaction that only affects the predicted values for the set of samples in the node that is subsequently split by a further predictor. The more complex the individual trees in the boosted model are, the more quickly the model will learn to predict the response and hence will require fewer trees to be grown before over-fitting, for a fixed learning rate. The complexity of the individual trees should ideally be chosen to reflect the true interaction order in the training data. However, this is often unknown and selection via an independent test-set or optimisation-set will be required.

To illustrate the fitting of boosted regression trees we demonstrate their use in a calibration setting using the European Diatom Database Initiative (EDDI) combined pH-diatom training-set. The combined pH data-set contains diatom counts and associated lake-water pH measurements for 622 lakes throughout Europe with a pH gradient of 4.32–8.40. As an independent test-set, we applied a stratified random sampling strategy to select a set of 100 samples from across the entire pH gradient by breaking the pH gradient into ten sections of equal pH interval and subsequently choosing ten samples at random from within each section of the gradient. The remaining 522 samples formed the training-set to which a boosted regression-tree model is fitted using the gbm package (Ridgeway 2010) for the R statistical software. The squared error loss-function was employed during fitting and we explored various learning rates of 0.1, 0.01, 0.001, and 0.0001 and tree complexities of 2, 5, 10, and 20 to identify the best set of learning parameters to predict lake-water pH from the diatom percentage abundance data. Preliminary exploration suggested that a large number of trees was required before error rates stabilised at their minimum and that a modest degree of tree complexity is required to optimise model fit, so we fitted models containing 20,000 trees. Throughout, we assessed model fit using five-fold cross-validation on-line during model fitting.

Figure 9.9a shows the value of the loss-function as trees are added to the model for a variety of learning rates. A tree complexity of 10 was used to build the models. The two fastest learning rates (0.1 and 0.01) converge quickly to their respective minima and then slowly start to over-fit, as shown by the increasing CV squared error loss. Conversely, the model fitted using the smallest learning rate is slow to fit the features of the training data-set and has still to converge to a minimum squared error loss when 20,000 trees had been fitted. The best fitting of all the models shown

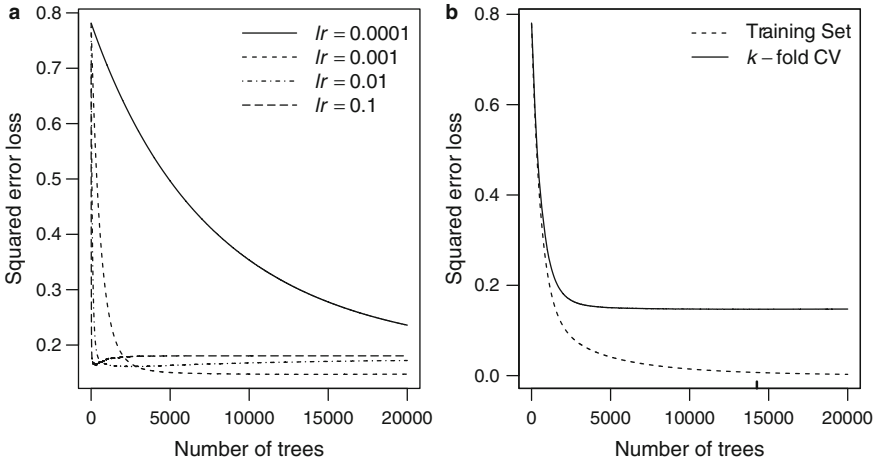

Fig. 9.9 Relationship between squared error loss, number of boosted trees, and learning rate (*lr*) for a boosted regression tree fitted to the European Diatom Database Initiative (EDDI) calibration set predicting lake-water pH from diatom species composition. (**a**) *k*-fold cross-validated and apparent squared error loss for the tuned boosted regression tree fitted to the EDDI data (**b**). The apparent squared error loss is derived using the training data to test the model and continues to decline as additional trees are added to the ensemble, indicating over-fitting. The thick tick mark on the x-axis of panel (**b**) is drawn at the optimal number of trees (14,255)

is the one with a learning rate of 0.001, which reaches a minimum squared error loss after approximately 14,000 trees. Figure 9.9b shows the CV squared error loss for this model alongside the training-set based estimate or error. We can clearly see that the boosted-tree model over-fits the training data converging towards an error of 0 given sufficient trees. This illustrates the need to evaluate model fit using a cross-validation technique, such as *k*-fold CV, or via a hold-out test-set that has not taken part in any of the model building.

The learning rate is only one of the parameters of a boosted regression tree for which optimal values must be sought. Tree complexity, *tc*, controls the size of the individual trees: the more complex the trees, the higher the degree of flexible interactions that can be represented in the model. Models that employ more complex trees also learn more quickly than models using simpler trees. This is illustrated in Fig. 9.10, which shows the effect of tree complexity on the squared error loss as trees are fitted for several values of complexity and for two learning rates ($lr = 0.001$ and 0.0001). The effect of tree complexity on the speed of learning is easier to see in the plot for the slowest learning rate (right hand panel of Fig. 9.10). The simplest trees, using tree complexities of 2 and 5, respectively, converge relatively slowly compared to the boosted trees using trees of complexity 10 or 20. Of the latter two, there is little to choose between the loss functions once tree complexity reaches a value of 10. Figure 9.10 combines the three parameters that users of boosted trees need to set to control the fitting of the model, and illustrates the key feature of

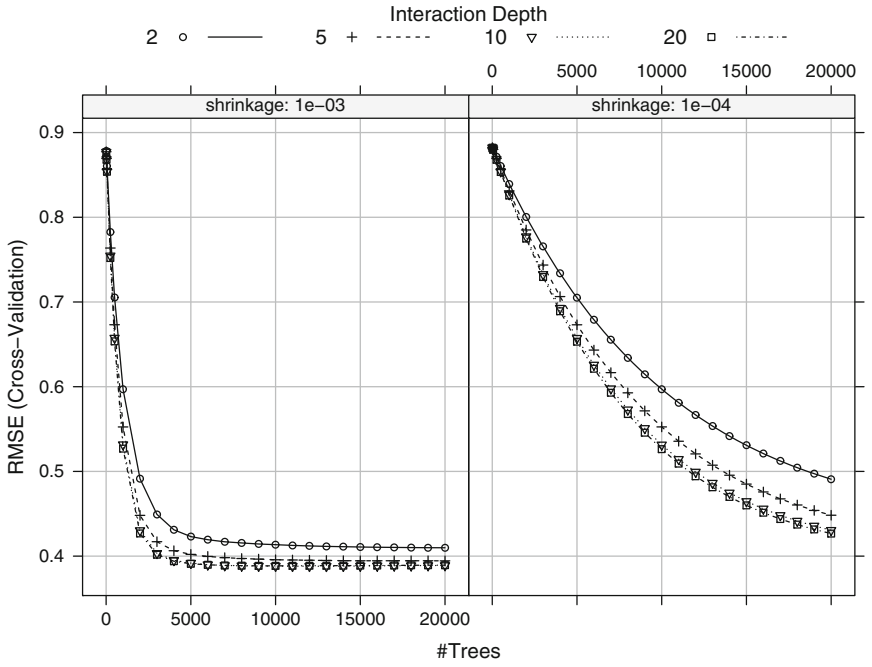

Fig. 9.10 Relationship between cross-validated root mean squared error of prediction (RMSEP) and number of boosted trees for a range of tree interaction depths and two learning rates ($lr = 0.001$, *left*; $lr = 0.0001$, *right*), for the European Diatom Database Initiative (EDDI) diatom-pH boosted regression tree model

requiring a sufficiently slow learning rate to allow averaging over a large number of trees, whilst using trees of sufficient complexity to capture the degree of interaction between predictors in the training data.

We can assess the quality of the boosted-tree calibration model by using the best fitting model ($lr = 0.001$, $tc = 10$, $nt = 13{,}000$). This model was chosen as the one giving the lowest five-fold CV error over a grid of tuning parameters. The RMSEP of the boosted tree for the test-set is 0.463 pH units. On the basis of Fig. 9.9a, one might consider using the model with $lr = 0.01$, $tc = 10$, and $nt = 2500$ instead of the best model as it has a similar, though slightly higher, squared error loss than the best model identified. Using this model gives a RMSEP for the test-set samples of 0.533, which is substantially higher than the best model. For comparison, we fitted weighted averaging (WA) calibration models (Juggins and Birks 2012: Chap. 14) to the EDDI training data using inverse and classical deshrinking and then applied each of these models to the held-out test-set. RMSEP for the WA models was 0.467 and 0.439 using inverse and classical deshrinking, respectively. There is little to choose between these models, with WA with classical deshrinking having the lowest hold-out sample RMSEP. It is always surprising how well the simple heuristic WA performs on such a complex problem of predicting lake-water pH from hundreds

of diatom species. In this example, one of the state-of-the-art machine-learning methods is unable to beat WA in a real-world problem!

Weighted averaging, whilst being very simplistic and powerful, is not a very transparent modelling technique as we do not have any useful variable importance measures that we can use to interrogate a WA calibration model. Bagged trees and random forests employ various variable importance measures to indicate to the user which predictors are important in modelling the response. In boosted trees, Friedman (2001) proposed to use the relative improvement in the model by splitting on a particular variable, as used in single tree models, as a variable importance measure in a boosted tree model but to average this relative importance over all trees produced by the boosting procedure. Figure 9.11 shows a needle plot of the 20 most important predictor variables (diatom species) for the boosted pH calibration model fitted to the EDDI data-set. The most important taxon is *Eunotia incisa* (EU047A), an acid-tolerant diatom, whilst *Achnanthes minutissima* agg. (AC048A) is a diatom that tends to be found in circum-neutral waters. The suite of taxa shown in Fig. 9.11 are often identified as indicator species for waters of different pH, so it is encouraging that the boosted model has identified these taxa as the most important in predicting lake-water pH (see Legendre and Birks 2012a: Chap. 7). Ecological examples of boosted regression trees are given by Elith et al. (2008), De'ath and Fabricius (2010), and Dobrowski et al. (2011).

Multivariate Adaptive Regression Splines

Multivariate adaptive regression splines (MARS) (Friedman 1991; Friedman and Meulman 2003; Leathwick et al. 2005) are an attempt to overcome two perceived problems of the single regression tree. The hierarchical nature of the tree imposes a severe restriction on the types of model that can be handled by such models. A change made early on in growing the tree is very difficult to undo with later splits, even if it would make sense to change the earlier split criteria in light of subsequent splits. Furthermore, as regression trees (as described above) fit piece-wise constant models in the leaves of the tree, they have difficulties fitting smooth functions; instead, the response is approximated via a combination of step functions determined by the split criteria. MARS does away with the hierarchical nature of the tree and uses piece-wise linear basis functions, combined in an elegant and flexible manner, to approximate smooth relationships between the responses and the predictors.

MARS proceeds by forming sets of reflected pairs of simple, piece-wise linear basis functions. These functions are defined by a single knot location, and take the value 0 on one side of the knot, and a linear function on the opposite side. Each such basis function has a reflected partner, where the 0-value region and the linear-function region are reversed. Figure 9.12 shows an example of a reflected pair of basis functions for variable or covariate x, with a single knot located at $t = 0.5$.

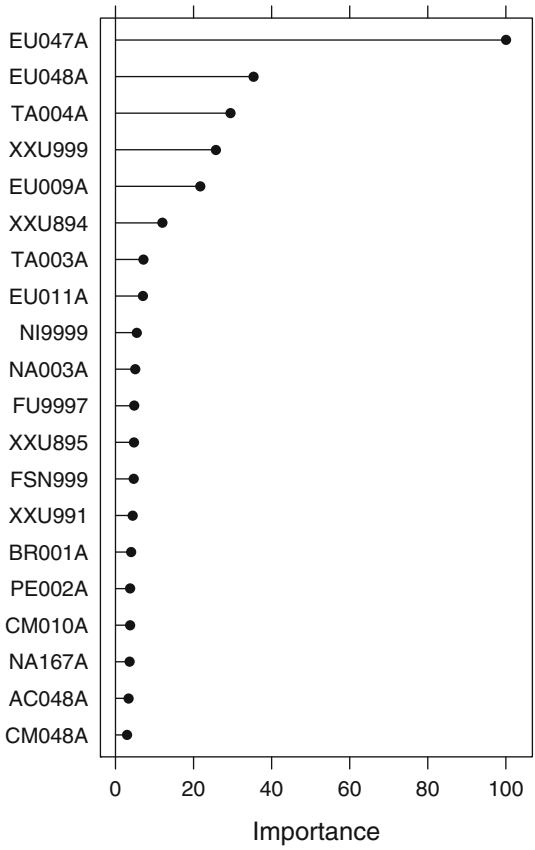

Fig. 9.11 Relative variable importance measure for the 20 most important diatom taxa in the European Diatom Database Initiative (EDDI) diatom training-set

The solid line is denoted $(x-t)_+$, with the subscript $+$ indicating that we take the positive part of the function only, with the negative part set to 0. As a result, the basis function illustrated by the solid line in Fig. 9.12 is zero until the knot location ($x = 0.5$) is exceeded. The reflected partner has the form $(t-x)_+$, and is illustrated by the dashed line in Fig. 9.12. For each quantitative covariate in the training data, a reflected pair of basis functions is formed by setting each t to be a unique value taken by that covariate. Qualitative covariates are handled by forming all possible binary partitions of the levels of a categorical covariate to form two groups. A pair of piece-wise constant functions are formed for each binary partition, which act as indicator functions for the two groups formed by the binary partition, and are treated like any other reflected pair of basis functions during fitting.

Model building in MARS is similar to step-wise linear regression except the entire set of basis functions is used as input variables and not the covariates themselves. The algorithm begins with a constant term, the intercept, and performs an exhaustive search over the set of basis functions to identify the pair that minimises the model residual sum-of-squares. That pair and their least-squares coefficients are added to the model-set of basis functions and the algorithm continues. Technically the

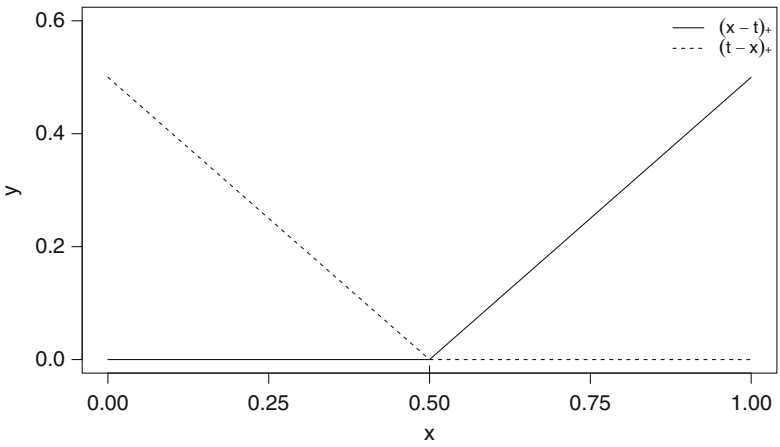

Fig. 9.12 Examples of a reflected pair of basis functions used in multivariate adaptive regression splines. The basis functions are shown for the interval (0,1) with a knot located at $t = 0.5$. See main text for details

algorithm finds the pair that, when multiplied by a term *already* included in the model, results in the lowest residual sum-of-squares, but as the only term in the model at the first iteration is the constant term, this amounts to finding the pair of basis functions that affords the largest improvement in fit. At the second and subsequent steps of the algorithm, the products of each existing model-term with the set of paired basis functions are considered and the basis function that results in the largest decrease in residual sum-of-squares is added to the model along with its partner basis function and their least-squares coefficients. The process continues until some stopping criteria are met; for example, the improvement in residual sum-of-squares falls below a threshold or a pre-specified number of model terms is reached. An additional constraint is that a single basis function pair may only be involved in a single product term in the model. Because products of basis functions are considered, interactions between covariates are handled naturally by the MARS model. The degree of interactions allowed is controlled by the user; if the degree is set to 1, an additive model in the basis functions is fitted.

At the end of this forward stage of model building a large model of basis functions has been produced that will tend to strongly over-fit the training data. A backwards elimination process is used to remove sequentially from the model the term that causes the smallest increase in the residual sum-of-squares. At each stage of the forward model-building phase, we added a basis function *and* its partner to the model during each iteration. The backwards elimination stage will tend to remove one of the pair of basis functions unless both contribute substantially to the model-fit (Hastie et al. 2011). A generalised cross-validation (GCV) procedure is used to determine the optimal model-size as ordinary cross-validation is too computationally expensive to apply to MARS for model-building purposes. The size

of a MARS model is not simply the number of model terms included within it; a penalty must be paid for selecting the knots for each term. The effective degrees of freedom (EDF) used by a MARS model is given by $\text{EDF} = \lambda + c((\lambda - 1)/2)$, where λ is the number of terms in the model, and c is a penalty term on the number of knots $((\lambda - 1)/2)$ and is usually equal to 3, or 2 in the case of an additive MARS model where no interactions are allowed. The EDF term is part of the GCV criterion that is minimised during the backwards elimination phase.

MARS was originally derived using least squares to estimate the coefficients for each basis function included in the model. The technique is not restricted to fitting models via least squares, however. The scope of MARS can be expanded by estimating the basis function coefficients via a generalised linear model (GLM), which allows the error distribution to be one of the exponential family of distributions (see Birks 2012a: Chap. 2).

We illustrate MARS via a data-set of ozone measurements from the Los Angeles Basin, USA, collected in 1976. A number of predictor variables are available; *inter alia*, air temperature, humidity, wind speed, and inversion base height and temperature. The aim is to predict the ozone concentration as a function of the available predictor variables. The variance in ozone concentrations increases with the mean concentration and as negative concentrations are not possible, a sensible fitting procedure for MARS is to estimate the coefficients of the model terms via a gamma GLM and the inverse link function (Birks 2012a: Chap. 2). Only first-order interaction terms were considered during fitting. The MARS model was fitted using the R package earth (Milbarrow 2011). A MARS model comprising ten terms, including the intercept and seven predictor variables, was selected using the GCV procedure. Four model terms involve the main effects of air temperature (two terms), pressure gradient[1] (DPG), and visibility. The remaining terms involve interactions between variables. A summary of the model terms and the estimated coefficients is shown in Table 9.6, whilst Fig. 9.13 displays the model terms graphically. The upper row of Fig. 9.13 shows the main effect terms. A single knot location was selected for air temperature at 58°F, with terms in the model for observations above and below this knot. Both air-temperature terms have different coefficients as illustrated by the differences in slopes of the fitted piece-wise functions. Note that the terms are non-linear on the scale of the response due to fitting the model via a gamma GLM.

Variable importance measures are also available to aid in interpreting the model fit, and are shown in Fig. 9.14 for the ozone example. The 'number of subsets' measurement indicates how many models, during the backward elimination stage, included the indicated term. The residual sum-of-squares (RSS) measure indicates the reduction in RSS when a term is included in one of the model subsets considered. The decrease in RSS is summed over all the model subsets in which a term is involved and is expressed relative to the largest summed decrease in RSS (which is notionally given the value 100) to aid interpretation. The GCV measure is

[1] Pressure gradient between Los Angeles airport (LAX) and Daggert in mm Hg.

Table 9.6 MARS model terms and their coefficients

Term	$\hat{\beta}$
Intercept	0.0802293
h(temp-58)	−0.0007115
h(58-temp)	0.0058480
h(2-dpg)	0.0018528
h(200-vis)	−0.0002000
h(wind-7) × h(1069-ibh)	0.0000681
h(55-humidity) × h(temp-58)	0.0000196
h(humidity-44) × h(ibh-1069)	0.0000005
h(temp-58) × h(dpg-54)	0.0000435
h(258-ibt) × h(200-vis)	0.0000010

The h() terms refer to basis functions, the numeric value inside the parentheses is the knot location for the piece-wise linear function, and the name inside the parentheses is the variable associated with the basis function *temp* Air Temperature (°F), *dpg* pressure gradient (mm Hg) from LAX airport to Daggert, *vis* visibility in miles, *wind* wind speed in MPH, *ibh* temperature inversion base height (feet), *humidity* percent humidity at LAX airport, *ibt* inversion base temperature (°F)

computed in the same manner as the RSS measure but involves summing the GCV criterion instead of RSS. A variable might increase the GCV score during fitting, indicating that it makes the model worse. As such, it is possible for the GCV importance measure to become negative. For the ozone model, air temperature is clearly the most influential variable, with the remaining variables included in the model all being of similar importance. The model explains approximately 79% of the variance in the response (76% by the comparable GCV measure). Ecological and biogeographical applications of MARS are relatively few and include Moisen and Frescino (2002), Leathwick et al. (2005, 2006), Prasad et al. (2006), Elith and Leathwick (2007), Balshi et al. (2009), and Franklin (2010).

Artificial Neural Networks and Self-organising Maps

Artificial neural networks (ANNs) and self-organising maps (SOMs) were developed for applications in artificial-intelligence research and are often conflated into a general machine-learning technique that is based on the way biological nervous systems process information or generate self-organising behaviour. However, despite these similarities, ANN and SOM are best considered from very different vantage points. There are also a large number of variations that fall under the ANN or SOM banner – too many to consider here. Instead we focus on the techniques most frequently used in ecological and limnological research (Lek and Guégan 1999).

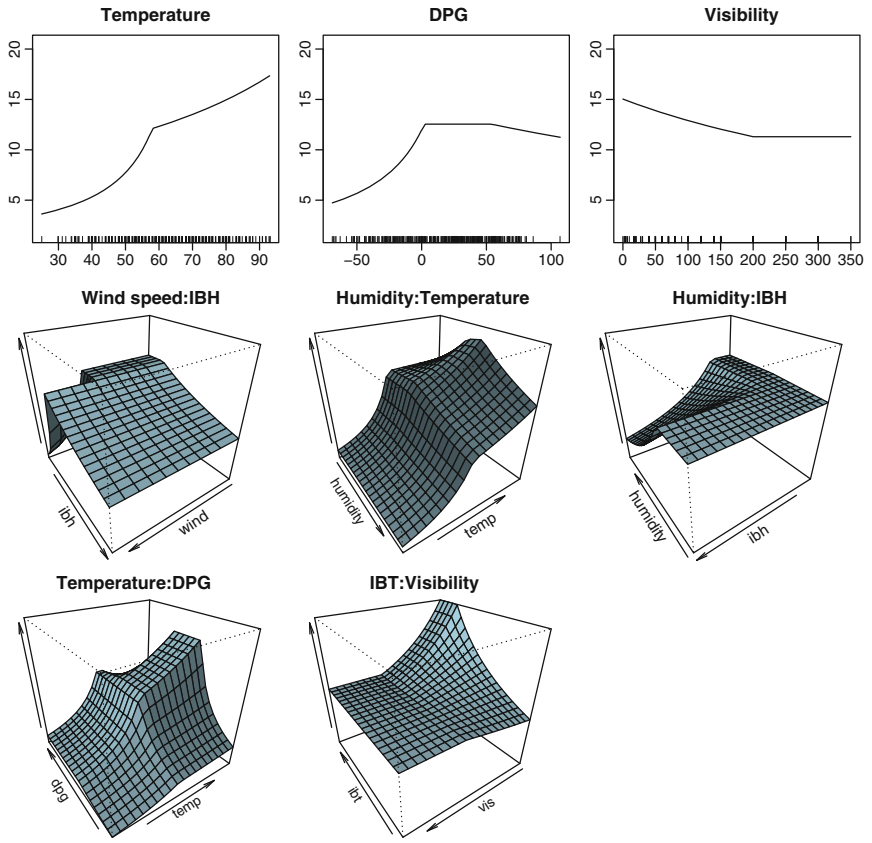

Fig. 9.13 Partial response plots for the multivariate adaptive regression spline (MARS) model fitted to the ozone concentration data from the Los Angeles basin

Artificial Neural Networks

An artificial neural network is a particularly flexible, non-linear modelling technique that is based on the way neurons in human brains are thought to be organised (Chatfield 1993; Warner and Misra 1996; Witten and Frank 2005; Ripley 2008). The term ANN today encompasses a large number of different yet related modelling techniques (Haykin 1999). The most widely used form of ANN is the forward-feed neural network, which is sometimes known as a back-propagation network or multi-layer perceptron. The general form of a forward-feed ANN is shown in Fig. 9.15. Configurations for both regression and classification settings are shown. The main feature of a forward-feed ANN is the arrangement of 'neurons' or units into a series of layers. The input layer contains m units, one per predictor variable in the training data-set, whilst the output layer contains units for the response variable or

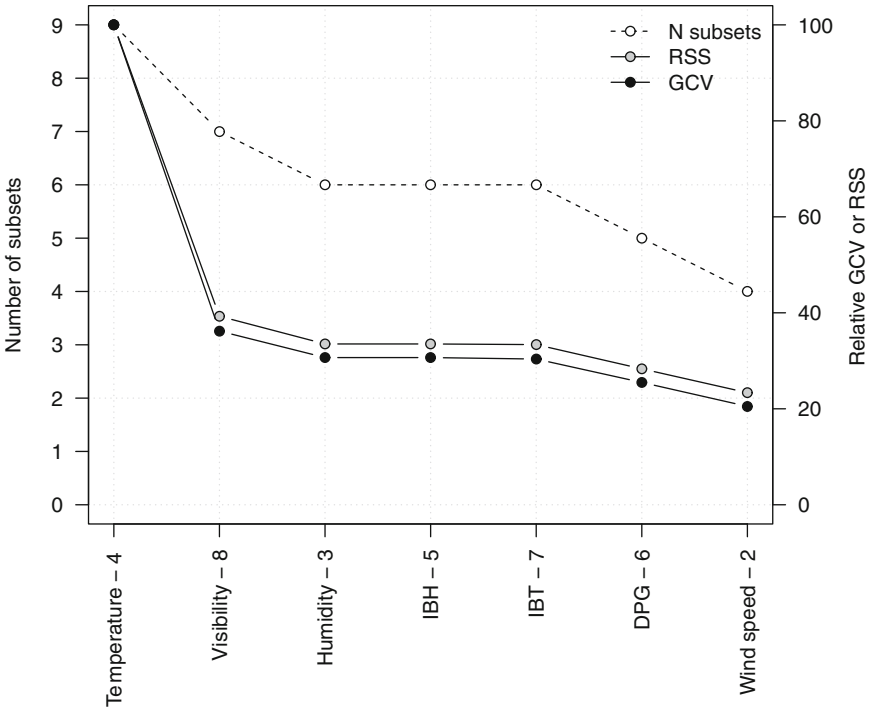

Fig. 9.14 Variable importance measures for the covariates in the multivariate adaptive regression spline (MARS) model fitted to the ozone concentration data from the Los Angeles basin. See main text for details of the various measures. *RSS* residual sum-of-squares, *GCV* generalised cross-validation

variables. In the univariate regression setting, there is a single unit in the output layer (Fig. 9.15a). In a classification setting, where the response takes one of k possible classes, there are k units in the output layer, one per class. The predicted class in a classification ANN is the largest value taken by $\mathbf{Y}_k$ for each input. Between the input and output layers a hidden layer of one or more units is positioned. Units in the input layer each have a connection to each unit in the hidden layer, which in turn have a connection to every unit in the output layer. The number of units in the hidden layer is a tuning parameter to be determined by the user. Additional hidden layers may be accommodated in the model, though these do not necessarily improve model fit. In addition, bias units may be connected to each unit in the hidden and output layers, and play the same role as the constant term in regression analysis.

Each unit in the network receives an input signal, which in the case of the input layer is an observation on m predictor variables, and outputs a transformation of the input signal. Where a unit receives multiple inputs, a transformed sum of these inputs is outputted. Bias units output a value of $+1$. The connections between units are represented as lines in Fig. 9.15 and each is associated with a weight. The output

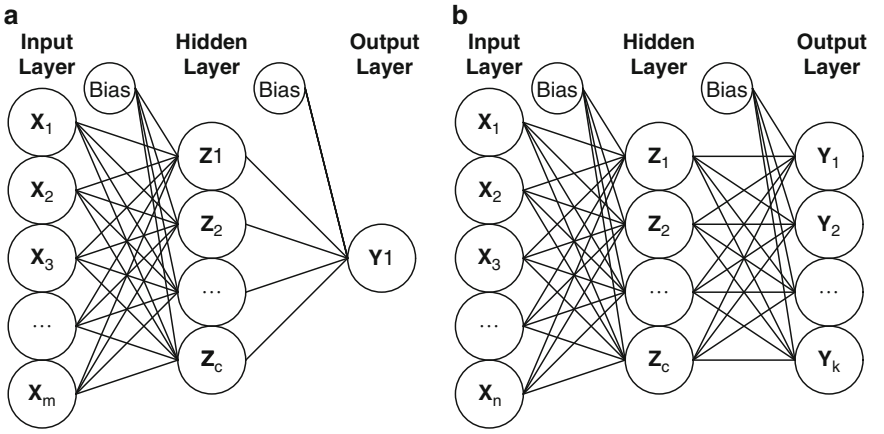

Fig. 9.15 Structure of a forward-feed, back-propagation neural network in a regression (**a**) and a classification (**b**) setting. A single hidden layer (Z_c) is shown. The lines connecting the layers of the network carry weights that are estimated from the data during fitting to minimise the loss of the final model. It is clear that the response is modelled as a function of a series of linear combinations (Z_c) of the input data

signal from an individual unit is multiplied by the connection weight and passed on to the next layer in the network along the connection. The weights are the model coefficients and optimal values for these are sought that best fit the response data provided to the network during training.

We said that the inputs to a unit are transformed. The identity function (Birks 2012a: Chap. 2) is generally used for the input layer, as a result the input data for the i^{th} sample are passed on to the hidden layer units untransformed. The hidden layer generally employs a non-linear transformation, typically a sigmoid function of the form

$$\sigma\,(sv) = 1\,/(1 + e^{-v}) \tag{9.5}$$

where v is the sum of the inputs to the unit and s is a parameter that controls the activation rate. Figure 9.16 shows the sigmoid function for various activation rates. As s becomes large, the function takes the form of a hard activation or threshold once a particular value of the inputs is reached. The origin can be shifted from 0 to v_0 by replacing the terms in the parentheses on the left hand side of Eq. 9.5 with $s(v - v_0)$. If an identity function is used in place of the sigmoid, the entire model becomes a simple linear regression. For the output layer, an identity function is used for regression models, whilst the softmax function, which produces positive outputs that sum to one, is used for classification.

The connection weights are estimated using gradient descent, known as back-propagation in the ANN field. For regression ANNs, sum-of-squares error is used to estimate the lack-of-fit for the current set of weights, whilst cross-entropy is

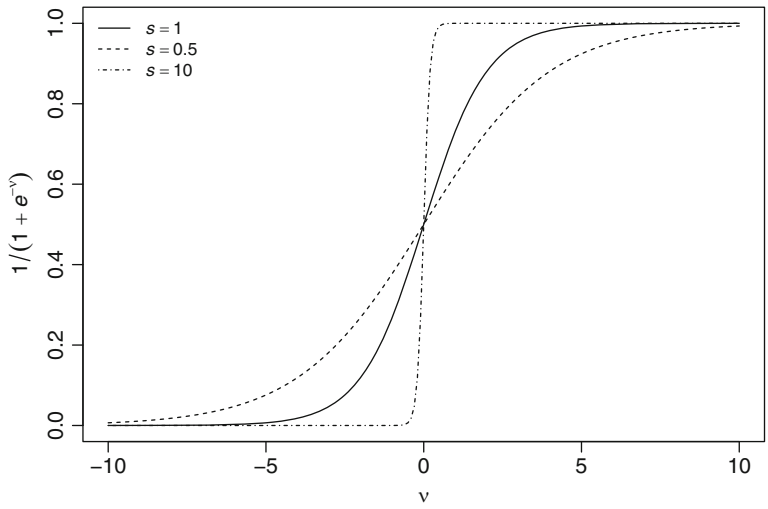

Fig. 9.16 Sigmoid function used to non-linearly transform inputs to hidden layer units in an artificial neural network (ANN) shown using a variety of activation rates *s*. See main text for further details

generally used in classification. The weights are sequentially updated to improve the fit of the model and each pass over the data is termed a training epoch. Generally, a large number of training epochs is performed to optimise the weights and thus improve the accuracy of the model. The set of weights that provides the global minimum of the model error is likely over-fitted to the training data. To alleviate over-fitting, training is often stopped early, before the global minimum is reached (Hastie et al. 2011). A validation data-set is useful in determining the appropriate stopping point, where the prediction error for the validation samples begins to increase. An alternative procedure, called weight-decay, provides a more explicit regularisation of the model weights, and is analogous to that used in ridge regression (see below). Details of the weight-decay procedure are given in Sect. 11.5.2 of Hastie et al. (2011).

It is instructive to consider what the units in the hidden layer represent; they are linear combinations of the input variables with the loading (or weighting) of each input variable in each unit $\mathbf{Z}_c$ given by the connection weight of the relevant unit in the input layer. We can then think of the forward-feed ANN as a general linear model in the linear combinations $\mathbf{Z}_c$ of the inputs (Hastie et al. 2011). A key feature of the forward-feed ANN is that the connection weights that define the linear combinations $\mathbf{Z}_c$ are learnt from the data during training. In other words, a set of optimal linear combinations of the inputs are sought to best predict the response.

ANNs are often considered black-box prediction tools (Olden and Jackson 2002) owing to how ANNs learn patterns from the data and encode this information in the connection weights, which makes it more difficult to extract and interpret than more simple, parametric techniques. To some extent this is a valid criticism;

however the connection weights are available for inspection along with the linear combinations of the inputs reconstructed ($\mathbf{Z}_c$) from these. Several methods for inspecting ANN model structure have been proposed, including the connection weighting approach of Olden et al. (2004) to derive a measure of variable importance, sensitivity analyses (Lek et al. 1996a), and various pruning algorithms (Bishop 1995, 2007; Despagne and Massart 1998; Gevrey et al. 2003). An example of a pruning algorithm applied in a palaeoecological context is the skeletonisation procedure of Racca et al. (2003), which for the Surface Waters Acidification Programme (SWAP) diatom-pH training-set allowed the removal of 85% of the taxa from the training data without drastically affecting model performance. This pruning also improved the robustness of the resulting calibration (Racca et al. 2003) (see Juggins and Birks 2012: Chap. 14).

Several factors can affect optimisation in ANNs which ultimately can determine the quality of the resulting model. We have already mentioned the potential for over-fitting the training data. In addition, the number of hidden layers and units within those layers needs to be decided. In general a single hidden layer will be sufficient, but additional layers can speed up model fitting. The number of units in the hidden layer controls the flexibility of functions of the input data that can be described by the model. Too many hidden units and the model may over-fit the data quickly, whilst too few units will unnecessarily restrict the very flexibility that ANNs afford. The optimal number of units in the hidden layer can be determined analytically (Bishop 1995, 2007; Ripley 2008) but in practice, treating the number of units as a tuning parameter to be optimised using *k*-fold cross-validation is generally used. Özesmi et al. (2006) reviewed other aspects of ANN assessment.

ANNs, when compared to the majority of the machine-learning tools described in this chapter, have been used relatively frequently to model palaeoecological data, particularly as a means of implementing calibration models (Borggaard and Thodberg 1992; Næs et al. 1993; Wehrens 2011). At one time ANNs were becoming a popular means of producing palaeoenvironmental reconstructions as they were seen as highly competitive when compared to modern analogue technique (MAT), weighted averaging (WA), and weighted-averaging partial least squares (WAPLS) because the calibration functions produced using ANNs had comparatively low root mean squared errors of prediction (RMSEP). Malmgren and Nordlund (1997) compared ANNs with Imbrie and Kipp factor analysis (IKFA), MAT, and soft independent modelling of class analogy (SIMCA) on a data-set of planktonic foraminifera and achieved substantially lower RMSEP than the other techniques. Racca et al. (2001) compared ANN, WA, and WAPLS calibration models for a data-set of diatom counts from 76 lakes in the Quebec region of Ontario, Canada. In this study, ANNs gave modest improvements in RMSEP over WA and WAPLS. Other palaeoecological applications of ANNs include Peyron et al. (1998, 2000, 2005), Tarasov et al. (1999a, b), Malmgren et al. (2001), Grieger (2002), Nyberg et al. (2002), Racca et al. (2004), Barrows and Juggins (2005), and Kucera et al. (2005). Limnological, environmental, biogeographical, and ecological examples are numerous, as reviewed by Lek and Guégan (2000). Illustrative examples include Lek et al. (1996a, b), Recknagel et al. (1997), Guégan et al. (1998), Lindstrom et al.

(1998), Brosse et al. (1999), Manel et al. (1999a, b), Spitz and Lek (1999), Olden (2000), Belgrano et al. (2001), Cairns (2001), Černá and Chytrý (2005), Steiner et al. (2008), and Chapman and Purse (2011).

The popularity of ANNs in palaeoecology has waned recently following the discovery that many published ANN-derived calibration functions may have greatly under-estimated model RMSEP by failing to account for spatial autocorrelation in the training data (Birks 2012a: Chap. 2). The autocorrelation problem can be accounted for using appropriate cross-validation techniques, such as the *h*-block approach of Burman et al. (1994) as used by Telford and Birks (2009). Typically, when one accounts for the dependence structure in the input data, the performance of ANNs is comparable to or worse than the best fits produced using WA and WAPLS.

Self-organising Maps

The self-organising map (SOM; also known as a self-organising feature map) is a relatively popular machine-learning tool for mapping and clustering high-dimensional data (Wehrens 2011), which has been used in a wide variety of ecological, environmental, and biological contexts (see e.g., Chon 2011 for a recent overview, and Giraudel and Lek 2001 for a comparison of SOMs and standard ordination techniques used in palaeoecology). The SOM is superficially similar to an artificial neural network, but this analogy only gets one so far and it is simpler to consider SOMs as a constrained version of the *K*-means clustering or partitioning method (Legendre and Birks 2012a: Chap. 7). As we will see, SOMs are also similar to principal curves and surfaces (see below and Hastie et al. 2011) and can be likened to a non-linear form of principal component analysis (PCA).

In a SOM, p prototypes are arranged in a rectangular or hexagonal grid of units of pre-defined dimension (number of rows and columns). The number of prototypes, p, is usually small relative to the dimensionality (number of variables or species) of the input data. A prototype is assigned to each grid unit. The SOM algorithm forces each of the samples in the input data to map onto one of the grid units during an iterative learning process. The goal of the SOM is to preserve the similarities between samples such that similar samples map on to the same or neighbouring units in the grid, whilst dissimilar samples are mapped on to non-neighbouring units.

At the start of the algorithm, the p prototypes are initialised via a random sample of p observations from the input data. Alternatively, the first two principal components of the input data can be taken and a regular grid of points on the principal component plane used as the prototypes (Hastie et al. 2011). Regardless of how the prototypes are initialised, each is characterised by a codebook vector that describes the typical pattern for the unit to which it has been assigned. If the prototypes are initialised using a random sample from the input data, then the codebook vector for an individual prototype will be the values of the species abundances, for example, in the sample assigned to that prototype. The aim of the SOM algorithm is to update these codebook vectors so that the input data are best described by the small number of prototypes.

During training, samples from the input data are presented to the grid of units in random order. The distance between the species abundances in the presented sample and the codebook vectors for each of the units is determined, usually via the Euclidean distance, but other distance measures can be used. The unit whose codebook vector is closest, i.e., most similar, to the presented sample is identified as the winning unit. The winning unit is then made more similar to the presented sample by updating its codebook vector. Geometrically, we can visualise this update as moving the unit in the m-dimensional space towards the location of the presented sample. By how much the codebook vector of the winning unit is updated (moved towards the presented sample) is governed by the learning rate, α, which is typically a small value of the order of 0.05. The learning rate is gradually decreased to 0 during learning to allow the SOM to converge.

Earlier, we noted that the SOM can be considered a constrained form of K-means clustering or partitioning: the constraint is spatial and arises because neighbouring units in the grid are required to have similar codebook vectors. To achieve this, not only is the winning unit updated to become more similar to the presented sample, but those units that neighbour the winning unit are also updated in the same way. Which units are considered neighbours of the winning unit is determined via another tuning parameter, r, which can be thought of as the distance within which a grid unit is said to be a neighbour of the winning unit. This distance, r, is topological, i.e., it is the distance between units on the grid, not the distance between the units in the m-dimensional space defined by the input data. The value of r, and hence the size of the neighbourhood around the winning unit, is also decreased during training; the implication is that as learning progresses, eventually only the winning units are updated. The SOM algorithm proceeds until an *a priori*-defined number of learning iterations, known as epochs, has been performed. The standard reference work for SOM is Kohonen (2001) where further details of the learning algorithm can be found.

As described above, SOM is an unsupervised technique, learning features of the data from the data themselves. However, the simplicity of the SOM algorithm allows scope for significant adaptation. One such adaptation allows SOMs to be used in a supervised fashion. If additional, dependent variables are available then these can be modelled alongside the independent or predictor variables. Such a supervised SOM then allows for predictions of the dependent variable to be made at new values of the predictor variables. One simple means of achieving this is to take an indirect approach and fit the SOM without regard to the dependent (response) variable(s) of interest and then take as the predicted value for each sample in the input the mean of the values of the response for all the samples that end up in the same grid unit as the sample of interest. This approach is very much in the spirit of the indirect ordination approach (Legendre and Birks 2012b: Chap. 8), but cannot be considered truly supervised.

Kohonen (2001) considered a supervised form of SOM and suggested building the map on the concatenation of the response variables ($\mathbf{Y}$) and the predictor variables ($\mathbf{X}$). In practice however, it may be difficult to find a scaling of $\mathbf{X}$ and $\mathbf{Y}$ such that both contribute similarly in the analysis. Indeed, if one of $\mathbf{X}$ or $\mathbf{Y}$ contains

many more variables than the other, it will dominate the distance computations when identifying the winning unit. Melssen et al. (2006) introduce two variations of supervised SOMs that have wide applicability as general techniques for analysing palaeoenvironmental data: (i) the X-Y Fused Kohonen Network (XYF) and (ii) the Bi-directional Kohonen Network (BDK). Both approaches make use of two grids of prototypes, the first providing a mapping of **X**, the second a mapping of **Y**, into low dimensions. The networks are trained in the same manner as described for the unsupervised SOM, but differ in how the two mappings are combined to identify the winning unit during each learning epoch.

XYF networks operate on a fused distance, where the total distance between each observation and the prototypes is a weighted sum of the scaled distance between each observation and the prototypes on the individual maps. The winning unit is the one that has the lowest weighted sum distance to the observation. The relative weighting is given by α, taking values between 0 and 1, with the distances on the **X** map weighted by $\alpha(t)$ and the distances on the **Y** map weighted by $1-\alpha(t)$. The distances between observations and prototypes on the individual maps are normalised by the maximum distance on each map so that the maximal distance on each map is 1. This scaling allows for very different magnitudes of distances on the maps, such as might arise when computing distances where **X** and **Y** are measured in different units or where different dissimilarity coefficients are used for the different maps. This latter point is particularly useful when applying the supervised SOM in a classification setting where the distance used for the response **Y** should consider group membership (0, 1). In such cases, the Jaccard distance (Legendre and Birks 2012b: Chap. 8; often called the Tanimoto distance in the chemometrics literature where the XYF and BDK methods were developed) is generally used. The t in $\alpha(t)$ indexes the learning epoch, allowing α to be decreased linearly during learning. Initially, this results in the determination of the winning unit being dominated by distances to prototypes on the **X** map. As learning proceeds, α is slowly decreased such that at the end of learning, distances to prototypes on both the **X** and **Y** maps contribute equally. It should be noted that a single epoch entails presenting, at random, each observation in the training-set to the maps.

The BDK network is similar to that described for the XYF network, but differs in that the two maps are considered individually during separate passes over the data. First, in the forward pass, the winning unit on the **X** map is identified as a weighted sum of distances on the two maps, as described above, and updated in the usual SOM manner. A reverse pass over the data is then performed, where the winning unit in the **Y** map is determined, again via a weighted sum of distances on the two maps, but this time using the **X** map updated in the forward pass. Learning proceeds in this alternating manner until convergence or an upper limit of epochs is reached. In practice there is generally little difference between the networks learned via the XFY or BDK methods (Melssen et al. 2006).

The XYF supervised SOM can be generalised to any number of maps, where the winning unit is identified as a weighted sum of distances over i maps, each map weighted by α_i, where $\Sigma\alpha_i = 1$, and the distances on each map scaled so the maximal distance is 1. Such a network is known as a super-organised SOM.

One problem with supervised SOMs as presented above is that in a regression setting, the number of possible fitted (or predicted) values of the response **Y** is limited by the number of units in the grid used. The fitted values for each observation are the mean of the response over the set of observations that map to the same unit. The predicted value for new observations is likewise the mean of the response for the training samples mapped to the same unit as each new observation. This is the same problem as identified for regression trees; in the terminology introduced there, a piece-wise constant model is fitted in the units of the trained supervised SOM. Melssen et al. (2007) combine the BDK or XYF networks with partial least squares regression (PLS) (Martens and Næes 1989; Wehrens 2011: Juggins and Birks 2012: Chap. 14) to overcome this deficiency in supervised SOMs.

We illustrate the utility and applicability of SOMs for palaeoecological data analysis using the SWAP-138 diatom calibration data-set, using the R package kohonen (Wehrens and Buydens 2007). Figure 9.17 shows output from a SOM fitted to the standardised, log-transformed (except pH, and conductivity was excluded from this analysis) water-chemistry for the 138-lake training-set. Figure 9.17a shows how the mean distance to the winning unit (per epoch) improves as the network is trained. The SOM appears to have converged after approximately 60 iterations. There is a clear conductivity signal in the data that is captured by the SOM (Fig. 9.17b), with units to the left of the map identified by high values of various ions and high pH and alkalinity. The upper right section of the map is characterised by dilute, low pH waters, whilst very low pH waters with high aluminium concentrations are located in the lower right area of the map. High total organic carbon (TOC) concentrations are found towards the lower left. The average distance of observations to the unit onto which they map is a measure of the quality of the mapping achieved by the SOM, and is shown in Fig. 9.17c for the SWAP water-chemistry SOM. There are few units with high mean distances, which suggests that the low-dimensional mapping closely fits the data. Figure 9.17d shows which unit each of the 138 SWAP sites maps onto and the number of samples within each unit. Given the small numbers of observations within some of the map units, it might be prudent to sequentially refit the SOM with reduced grid sizes until the degree of fit drops appreciably.

A supervised SOM can be fitted to the SWAP-138 diatom and lake-water chemistry data to investigate relationships between chemistry and diatom species composition. Here we use the square-root transformed species data as the response data, **Y**, and the standardised water chemistry data in the predictor role, **X**. Only diatom taxa that were present in at least 5 samples at 2% abundance or greater were included in the analysis. Both maps converged after approximately 60 epochs (Fig. 9.18a) and achieved similar levels of fit. The codebook vectors for the **X** map (chemistry: Fig. 9.18b) are very similar to those produced by the unsupervised SOM (Fig. 9.17b), indicating the strong influence on diatom species composition exerted by the water chemistry. In general, the supervised SOM **X** map is a reflected, about the vertical, version of the unsupervised SOM; higher ionic strength waters are found to the right and the more acid sites to the left. The high aluminium, low pH units are now located to the upper left, with the low pH and low aluminium units to the lower left.

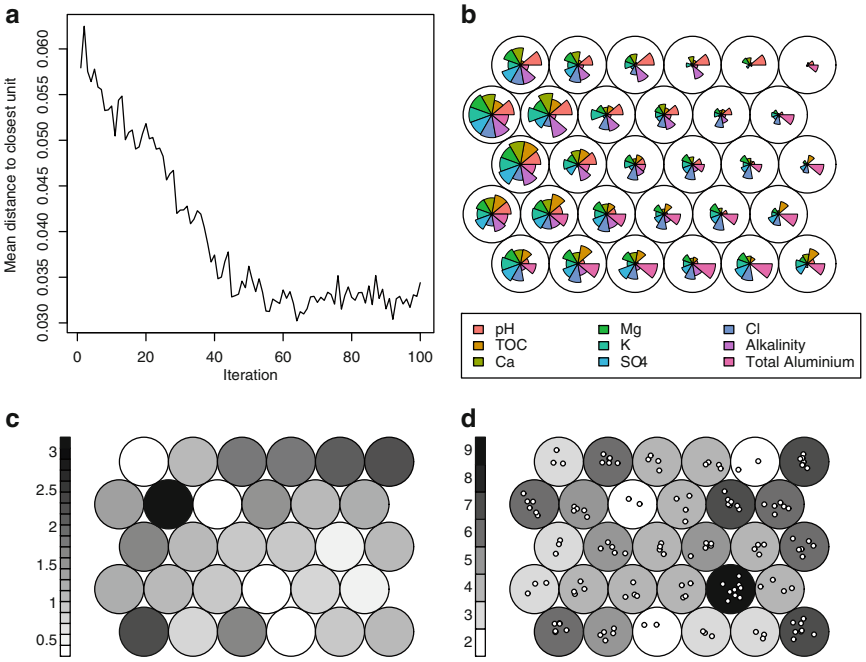

Fig. 9.17 Graphical summary of the self-organising map (SOM) fitted to the Surface Waters Acidification Programme (SWAP) water chemistry data-set: (**a**) shows how the mean distance to the closest map unit falls steadily as the SOM is trained and learns the features of the data, stabilising after 60 iterations or training epochs. The codebook vectors for the trained SOM map units are shown in (**b**) where each segment represents one of the nine water chemistry determinands and the radius of each segment represents the 'abundance' of the determinand (*large radii* indicate large values and *small radii* indicate small values). The degree of heterogeneity in the water chemistry of samples within each map unit is shown in panel (**c**) with higher values indicating units with samples of more heterogeneous chemistry. The number of samples in the SWAP training-set mapped to each unit in the SOM grid is shown in (**d**); the background shading refers to the number of samples and each map unit on the panel contains that number of samples (*circles*) plotted using a small amount of jitter

Due to the large number of taxa, the codebook vectors for the **Y** map are best visualised on a per taxon basis. Figure 9.19 shows the XYF SOM-predicted abundances (back-transformed) for four taxa with differing water chemistry responses. *Achnanthes minutissima* is restricted to the high pH, high alkalinity units to the right of the map. Predicted abundances for *Brachysira brebissonii* are positive for many units indicating the wide tolerance of this taxon, however it is most abundant in the slightly more-acidic units on the map. *Tabellaria binalis*, an acid-loving species, is found most abundantly in the very acid, high aluminium map units towards the upper left of the map, whilst *Eunotia incisa*, an acid-tolerant species common in nutrient-poor, acid waters, is most abundant in a range of the low pH units but particularly in those with lower aluminium concentrations.

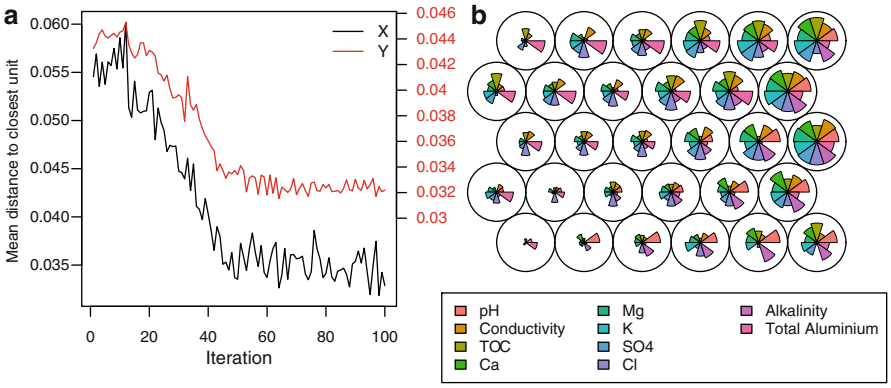

Fig. 9.18 Graphical summary of the X-Y fused Kohonen network self-organising map (XYF-SOM) fitted to the Surface Waters Acidification Programme (SWAP) diatom training-set. The square-root transformed diatom data were used as the response map Y with the water chemistry data used as predictor map X. (**a**) Shows how the mean distance to the closest unit for both X and Y maps decreases steadily as the XYF-SOM is trained, apparently converging after 50 iterations. The codebook vectors for the X map (water chemistry) are shown in (**b**). See Fig. 9.17 for details on interpretation

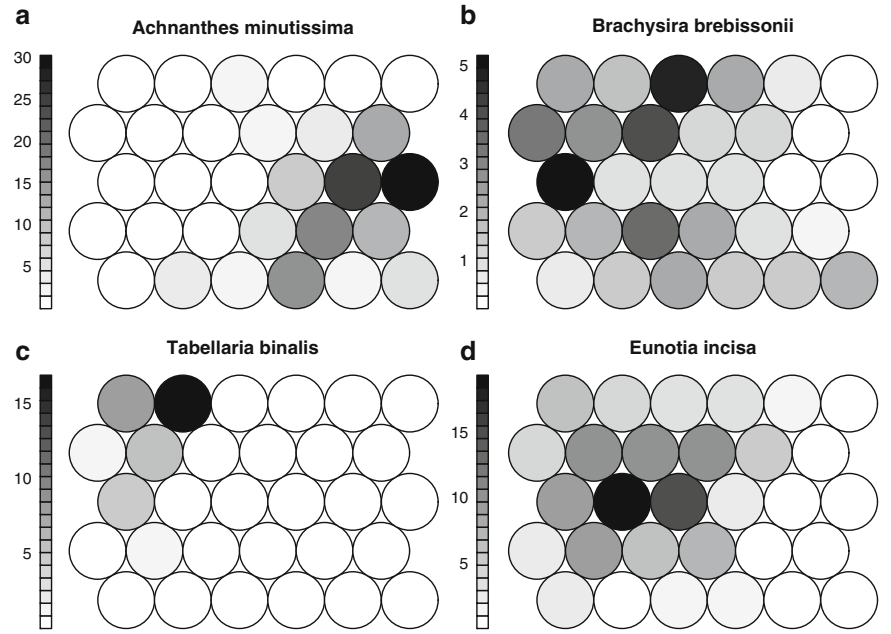

Fig. 9.19 Predicted percentage abundance for four diatom taxa using a X-Y fused Kohonen network self-organising map (XYF-SOM) fitted to the Surface Waters Acidification Programme (SWAP) training-set data

A supervised SOM can also be used as a multivariate calibration tool; here the species data play the role of the predictor variables (**X** map), whilst the variable(s) of interest to be predicted are now used in the response role (**Y** map). Here we build a supervised SOM to predict lake-water pH from the diatom data, using the same data as for the previous example except in reverse roles. We also only include pH as the sole **Y** map variable, although, where appropriate, two or more response variables may be included in a calibration SOM. The fitted model has an apparent root mean squared error (RMSE) of 0.215 pH units when assessed using the training-set data. Further analysis of the fitted codebook vectors of the species (**X** map) can be performed, to identify those taxa most influential for predicting pH and also the species composition of the SOM map unit. We use the fitted XYF SOM to predict lake-water pH values for the Holocene core from The Round Loch of Glenhead (Birks and Jones 2012: Chap. 3). Only those taxa used to fit the XYF SOM were selected from the fossil data. The pH reconstruction is shown in the upper panel of Fig. 9.20, whilst the pH codebook vector is shown for each map unit in the lower panel with the fossil samples projected on to the map. Whilst the general form of the reconstruction is similar to previously published reconstructions (e.g., Birks et al. 1990) and the recent acidification period is captured by the reconstruction, a major deficiency in the reconstruction is immediately apparent; the predicted values for the core samples only take on one of nine possible values. This is due to the predicted pH for each fossil sample being the fitted pH value from the map unit onto which each fossil sample is projected. As the fossil samples project onto only nine map units, only nine possible values can be predicted for the reconstruction. This deficiency is addressed by Melssen et al. (2007) by combining supervised SOMs with PLS. Although we will not consider this technique further here, the general idea is that a BDK SOM is trained on the input data and the similarities between the objects and the codebook vectors of the trained SOM are computed to form a similarity matrix. The elements of this matrix are weighted by a kernel function to form a so-called kernel matrix. The columns of this kernel matrix are then used as predictor variables in a PLS model to predict the response (Melssen et al. 2007). In this way, the information contained in the trained SOM is used to predict the response, but continuous predictions can now be produced because of the use of PLS. Examples of the use of SOMs in limnology and palaeoecology include Malmgren and Winter (1999), Céréghino et al. (2001), Holmqvist (2005), and Weller et al. (2006).

Bayesian Networks

Bayesian networks (also known as belief networks or Bayesian belief networks) are a powerful modelling technique that describes a means by which reasoning in the face of uncertainty about a particular outcome can be performed (Witten and Frank 2005; Bishop 2007; Jensen and Nielsen 2007; Ripley 2008). A Bayesian network can be viewed as a graphical description of the system under study, where

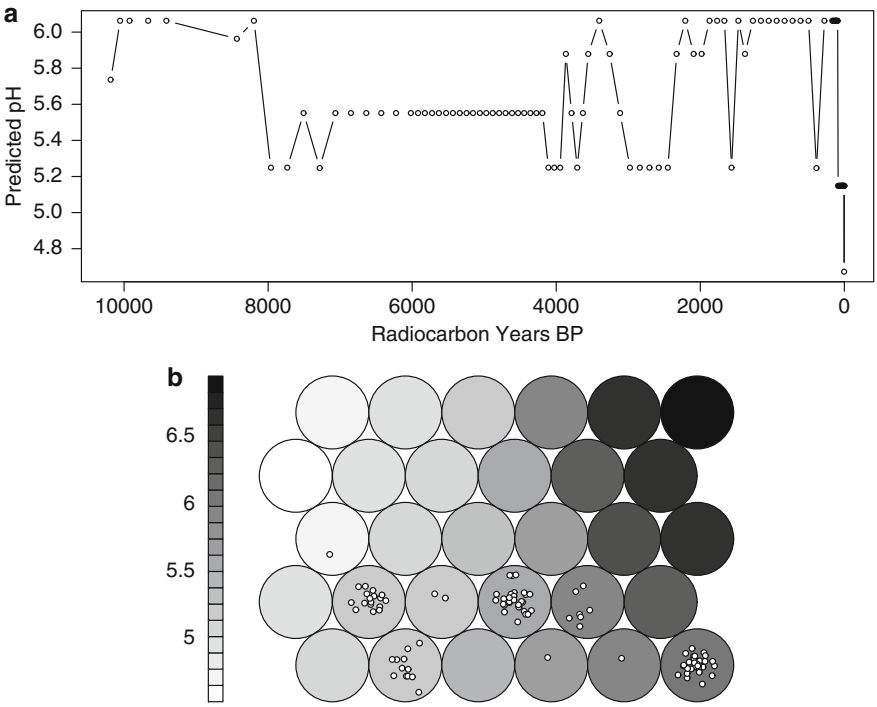

Fig. 9.20 Graphical summary of a X-Y fused Kohonen network self-organising map (XYF-SOM) fitted to the Surface Waters Acidification Programme (SWAP) training-set in calibration mode, with lake-water pH used as the response data Y and square-root transformed diatom abundance data used as prediction data X, and applied to a Holocene diatom sequence from The Round Loch of Glenhead, Scotland, UK. (**a**) Reconstructed lake-water pH history for the loch. The predicted pH for each map unit is shown in (**b**) with The Round Loch of Glenhead sediment core samples mapped on it

key features of the system are represented by nodes that are linked together in some fashion so that the cause-and-effect relationships between the nodes are described. Bayesian networks are more formally known as directed acyclic graphs (DAGs), where the nodes represent random variables and the linkages between nodes represent the conditional dependencies between the joined nodes. The graph is acyclic, meaning that there are no loops or feedbacks in the network structure, and is directed because the relationships between nodes have stated directions; A causes B (Ripley 2008).

Consider a simple system with two nodes, A and B, which are the nodes in the network. A and B are linked by a directional arrow from A to B indicating that A influences B. In this network, A is the parent of B, and B is the child of A. A has no parents and thus is also known as a root node, and plays the role of an input variable in the network. A node that does not have any children is known as a leaf node and

plays the role of an output variable. Each node in the network is associated with a set of states, that may be discrete or continuous, which represent the set of possible conditions that the node may take. A conditional probability table is produced for each node, which states the probability with which a node will take each of its states conditional upon the states (or values) of the parent nodes. As such, root nodes are not initialised with conditional probability tables and instead are provided unconditional probabilities: the probability that the input variable (root node) is in a particular state. Conditional independence is a key property of Bayesian networks: two events X and Y given a third event Z are said to be conditionally independent if, given knowledge about the state of Z, knowledge of X conveys no information about the state of Y or vice versa. Independent and interactive (conditional) effects of variables on the modelled response (output nodes) can be examined. Bayesian networks also assume the Markov property, namely that the conditional probability tables can be completed only by considering the immediate parents of a particular node. If we know the probabilities of the states for the parents of a particular node, given the conditional probability table for that node, the probabilities for the child nodes can be computed using Bayes Theorem

$$P\ (y|x) = \frac{P\ (x|y)\ P(y)}{P(x)} \tag{9.6}$$

where $P(y)$ is the prior probability of the child node, $P(x \mid y)$ is the likelihood or the conditional probability of x given y, $P(x)$ is the probability of the parent node and is a normalising constant in the equation, and $P(y \mid x)$ is the posterior probability of the child node given the state of the parent x. The posterior probability $P(y \mid x)$ is the probability of a particular state of the child node conditional upon the probabilities of the states of the parent. The prior probabilities and the conditional probability tables for the nodes may be specified using expert judgement and knowledge of the system under study or learned from the training data via one of several Bayesian learning algorithms.

Bayesian networks can be operated bottom-up or top-down. Consider again our system with two nodes, A and B. In bottom-up mode, we might observe a particular state for B, thus setting the probability for that state to 1, and then propagate this information back up the network to A to determine the most likely state of A, given that we have observed the state of B. Conversely, we might be interested in determining the effect on B of altering the state of A, therefore we set the probability for one of the A states to 1 and then propagate this information down the network to see the most likely response of B to the state of A.

As an example, consider a study relating nutrient loadings, through trophic levels, to provide an estimate of water quality (Castelletti and Soncini-Sessa 2007a). Nitrogen and phosphorus loadings influence the trophic level of a water body, stimulating primary production when nutrient levels are elevated, and thus the trophic level is an influence on the perceived water quality. The network associated with this hypothetical system/problem is shown in Fig. 9.21. In this simplified illustration, each of the nodes is characterised by two states; low and high. Table 9.7

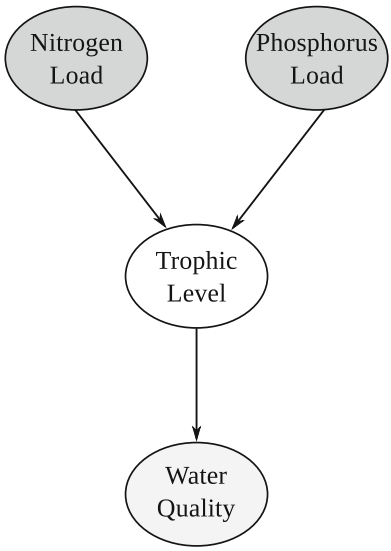

Fig. 9.21 Example of a Bayesian Network discussed in the text, showing the directional relationship of the effects of nutrient loadings on trophic level and consequently upon water quality. Input/root nodes are shown in dark grey, whilst leaf/output nodes are shown in light grey (Modified from Castelletti and Soncini-Sessa 2007a)

Table 9.7 Conditional probability tables for the Trophic Level (a) and Water Quality (b) nodes in Fig. 9.21

(a)					
Nitrogen loading		L		H	
Phosphorus loading		L	H	L	H
Trophic level	**L**	1.0	0.3	0.5	0.0
	H	0.0	0.7	0.5	1.0

(b)			
Trophic level		L	H
Water quality	**L**	0.0	0.8
	H	1.0	0.2

L Low, *H* High

shows the conditional probability tables for the trophic level and water quality nodes for this illustrative example. If the prior beliefs of the states for the phosphorus and nitrogen loading nodes are set to the values shown in the left-hand section of Table 9.8, the posterior probabilities computed using the conditional probability tables (Table 9.7) of the trophic level and water quality states would be those shown in the right-hand section of Table 9.8. If our prior beliefs about the probabilities of the nutrient-loading states were to change or be updated, then the conditional probabilities of the states for trophic levels and water quality would likewise be updated in light of the new prior beliefs.

Bayesian networks can be used to inform the decision-making process via the inclusion of a decision node into the network (Korb and Nicholson 2004; Bishop 2007). Returning to our simple two-node network example (A and B), we could turn this network into a Bayesian decision network (BDN) by assigning a decision parent node to A. This decision node might also be associated with a cost function describing the cost of enacting the decision. The decision node describes the states

Table 9.8 Prior beliefs for the states of nitrogen and phosphorus loading, which when combined with the conditional probability tables in Table 9.7, yield the posterior probabilities for the states of trophic level and water quality. Arrows show the directional relationships of the effects of the nutrient loadings on trophic level and hence water quality (see Fig. 9.21)

	Nitrogen loading	Phosphorus loading		Trophic level		Water quality
L	0.1	0.3	→	0.1	→	0.7
H	0.9	0.7		0.9		0.3

L Low, *H* High

of possible management actions, for example restoration strategies or water-quality limits or standards, whilst the cost function describes the cost of enacting a particular restoration strategy or setting a particular water-quality standard. The output node in our example, B, is linked to a utility node, which describes the desirability (utility) of particular states of the outcome node. Node A now needs to be assigned a conditional probability table to describe the probabilities of the states of A conditional upon the different states of the decision node. The utility output from the network is the sum of the individual utilities of the output state in node B, weighted by the probabilities of each of the output states. Management decisions can then be based on selecting the intervention that maximises the output utility of the network relative to the cost of intervention. As with the simpler Bayesian networks, the prior and conditional probabilities of the BDN nodes can be set *a priori* using expert judgement or learned from available training data or a combination of the above; probabilities for decision nodes and utility values for outcome states are set by the user.

Bayesian networks have seen little use in palaeoecology, but have had some limited use in conservation management in freshwater ecology. Stewart-Koster et al. (2010), for example, use Bayesian networks to investigate the cost effectiveness of flow and catchment restoration for impacted river ecosystem, the output of which would be used to guide investments in different types of restoration. Other examples include the use of Bayesian networks in water-resource management (Castelletti and Soncini-Sessa 2007b; Allan et al. 2011), the evaluation of management alternatives on fish and wildlife population viability (Marcot et al. 2001), and the effects of land-management practices on salmonids in the Columbia River basin (Rieman et al. 2001), whilst Newton et al. (2006, 2007), Aalders (2008), Kragt et al. (2009), Murphy et al. (2010), and Ticehurst et al. (2011) employ Bayesian networks in vegetation conservation and management. Pourret et al. (2008) present a wide range of case studies from many disciplines that have found Bayesian networks useful.

Genetic Algorithms

Genetic algorithms are one of a number of stochastic optimisation tools that fall under the heading of evolutionary computing. Numerical optimisation is a general catch-all term for algorithms that given a cost (or loss) function aim to find a globally

optimal solution to a modelling problem, for example a set of model coefficients that minimises the lack of fit of a model to a set of training samples. Numerical optimisation techniques that use derivatives of the loss function proceed towards an optimal solution in an iterative fashion but which may not, however, converge to a globally optimal solution, instead they find a locally optimal solution. This is akin to always walking downhill to find the lowest point in a landscape; eventually you will not be able to proceed further because to do so would involve moving uphill. A much lower valley just over a small rise from the one you are currently in would be out of reach if you could only walk downhill. Evolutionary computing introduces ideas from natural selection and evolution to add elements of stochasticity to the optimisation search in an attempt to avoid becoming trapped in sub-optimal local solutions.

Of the various evolutionary computing techniques, genetic algorithms have been most frequently used in ecology, especially the Genetic Algorithm for Rule-set Prediction (GARP) procedure, which has seen extensive use in modelling spatial distributions of species (Anderson et al. 2003; Jeschke and Strayer 2008; Franklin 2010). Here we describe genetic algorithms in a general sense, and then we briefly discuss genetic programmes and GARP.

Genetic algorithms consider a population of solutions to a modelling problem rather than a single solution (D'heygere et al. 2003). Each of the solutions is described by a string of numbers, each number representing a gene and the set of numbers an individual chromosome in the terminology of genetic algorithms. The strings represent terms in the model. If we consider a simple least-squares regression, then we could use a string of length m zeroes and ones indicating which of the m predictor variables is in the model (Wehrens 2011). Alternatively, we could just record the index of the variables included in the model, where the string of values would be of length M (the number of variables in the model, its complexity) and the individual values in the string would be in the set $(1, 2, \ldots, m)$ (Wehrens 2011). The size of the population of chromosomes (the number of solutions) considered by the genetic algorithm needs to be set by the user; with too small a population the algorithm will take a long time to reach a solution, whilst too large a population entails fitting many models to evaluate each of the chromosomes in every generation. The initial population of chromosomes is generally seeded by assigning a small random selection of the available predictor variables to each of the C chromosomes.

Offspring solutions (chromosomes) are produced via a sexual reproduction procedure whereby genes from two parent solutions are mixed. The fitness of the offspring determines which of them persist to produce offspring of their own, with fitness being defined using a loss function, such as least-squares error. Offspring with low fitness have a low probability of reproducing, whilst the fittest offspring have the highest chance of reproducing. This process of sexual selection is repeated a large number of times with the result that subsequent generations will tend to consist of better solutions to the modelling problem. The sexual reproduction step consists of two random processes termed crossover or sharing of parents' genes, and mutation. These processes are random and as such are not influenced by the fitness

of individual parents. Sexual reproduction mixes the genes from two parents in a random fashion to produce an offspring that contains a combination of the genes from the two parents. Mutation introduces a stochastic component to the genetic algorithm, and allows predictor variables not selected in the initialisation of the population of chromosomes a chance to enter the genetic code of the population. Mutation is a low-probability event; say 0.01 indicating that one time in a hundred a mutation will take place during reproduction. Mutations can involve the addition of a new variable to the chromosome, the removal of an existing variable, or both addition and removal of variables. Mutation allows the genetic diversity of the population to be maintained.

Each iteration of a genetic algorithm produces a new generation of offspring by sexual reproduction of the fittest members of the current population. The candidates for reproduction are chosen at random from those models that reach a minimum fitness threshold. The selection of two candidates for reproduction may be done at random from within this set of fittest chromosomes or at random with the probability of selection weighted by the fitness of each chromosome. The latter gives greater weight to the best of the best solutions in the current population.

The genetic algorithm is run for a large number of iterations (generations) and the fittest solution at the end of the evolutionary sequence is taken as the solution to the modelling problem. It is possible that the population of solutions will converge to the same, identical solution before the stated number of generations has been produced. Likewise, there is no guarantee of convergence to the best solution in the stated number of iterations. As such, it is important that the evolutionary process is monitored during iteration, say by recording the fitness of the best solution and the median fitness over the population of solutions for each generation (Wehrens 2011). If the fitness of the best solution is still rising and not reached an asymptote by the end of the generations then it is unlikely that the algorithm has converged.

Genetic algorithms are a general purpose optimisation tool, and as such they require far more user interaction than many of the other statistical machine-learning methods described in this chapter. The size of the population of solutions, the minimum and maximum number of variables included in a single solution, the number of iterations or generations to evolve, the mutation rate, the fitness threshold required to select candidates for sexual reproduction, and the loss function all need to be specified by the user. The flexibility of the genetic algorithm thus comes with a price. However, the algorithm can be applied to a wide range of problems, simply by introducing a new loss function that is most appropriate to the modelling problem to hand. The loss function can be any statistical modelling function, such as least-squares, linear discriminants, principal components regression, or partial least squares, for example, and as such a wide range of problems can be tackled. Genetic algorithms can also be slow to converge to an optimal solution, especially when faced with a complex modelling problem consisting of many observations and predictor variables.

Genetic programmes are related to genetic algorithms, but now each chromosome in the population is a computer program that uses combinations of simple arithmetic rules (using $+$, $-$, $\times$, etc.) and mathematical functions or operators. The

various rules and functions are combined into a syntax tree to combine numeric values with mathematical operators and functions that form a solution to a problem. Reproduction now takes the form of randomly swapping sub-trees in the syntax trees of two parents to produce new offspring that include aspects of both parents' genetic programme. Mutation is performed by randomly selecting a sub-tree in the syntax tree of an individual and replacing that sub-tree with a randomly generated sub-tree. Which programmes are allowed to reproduce is controlled by a fitness criterion in the same way as described for genetic algorithms. The key difference between a genetic algorithm and a genetic programme is that genetic algorithms optimise an *a priori* specified model by evolving solutions to the modelling problem (regression coefficients for example) that give the best fit of the model to the training data, whereas genetic programmes aim to find an optimal solution to an unspecified modelling problem by combining simple mathematical steps to best fit or explain the training data.

GARP (Stockwell and Noble 1992; Stockwell and Peters 1999) is a genetic algorithm where the genes do not represent inclusion or exclusion of particular predictor variables, but instead are simple rules that are very much akin to the rules produced by the tree models we described earlier. In GARP, each of the rules follows a similar form: *if* 'something' is true, *then* 'this' follows, where 'something' is a simple rule and 'this' is a predicted value say. For example, a rule might be *if* pH is less than Y and aluminium is greater than X, *then* the abundance of the diatom *Tabellaria binalis* is Z%. The set of possible rules using combinations of predictor variables is impossibly large for most problems for an exhaustive search to be made. Instead, genetic algorithms are used to evolve the rules into a set of optimal combinations that best predict the response. The algorithm starts by identifying all rules consisting of a single predictor; at this point, the algorithm is very much similar to the exhaustive search used in tree models to identify the first split. A predefined number, r, of these rules is then chosen as the initial set of rules upon which the genetic algorithm will operate. The r best rules are chosen as the initial set. Each of several predefined operators is then applied to the initial set of rules to evolve a new generation of rules. These operators include a random operator which creates a rule with a random number of conditions (*if* 'something's) and values (*then* 'this's), a mutation operation which randomly changes the values used in a condition, and a concatenation operation which combines two randomly chosen rules from the existing set. Having applied these operators to the current set of rules, the rules are ordered in terms of fitness, and the least fit rules are discarded. The remaining set of rules then undergo another round of operator application to evolve a new generation of rules and the least fit rules again are discarded. This process is repeated a large number of times in order to evolve a set of rules that best predicts the response. GARP is most useful in situations where the user has little reliable background knowledge to guide model choice and in situations where rules are sought in noisy, high dimensional, discontinuous data with many local optima. However, GARP is considered computer intensive relative to the many of the statistical machine-learning tools described here.

Genetic algorithms and programmes and GARP are very flexible, general optimisation tools. However, they are not well suited to all problems. More-specific statistical machine-learning tools, such as regression or classification trees and related methods will tend to perform as well or better than the evolutionary computing approaches for general regression or classification problems (D'heygere et al. 2003; Olden et al. 2008), and as we have seen, bagging, random forests, and boosting can all improve upon single tree models by combining information from several weak learners. In addition, Elith et al. (2006) and Lawler et al. (2006) both observed that GARP tended to over-fit species distributions compared with other modelling techniques. As such, and given the availability of powerful alternative techniques plus the additional effort required by the user to use evolutionary computing techniques, we cannot recommend their use over the other statistical machine-learning techniques described earlier. GARP is, however, widely used in species-climate modelling in biogeography and climate-change predictive biology (e.g., Elith and Burgman 2002; Stockwell and Peterson 2002; Pearson et al. 2006; Tsaor et al. 2007; Jeshcke and Strayer 2008).

Principal Curves and Surfaces

Principal component analysis (PCA) (Jolliffe 2002; Legendre and Birks 2012b: Chap. 8) is used in a large number of fields as a means of dimension reduction by expressing on the first few principal components orthogonal linear combinations of the input data that explain the data best in a statistical sense. These first few principal component axes are often used as synthesisers of the patterns of change found in stratigraphical data for example (Birks 2012b: Chap. 11). PCA is also the basis of the linear, multivariate calibration technique principal components regression (Juggins and Birks 2012: Chap. 14), where the input data are reduced to $p \ll m$ components, which are then used in a multiple regression to predict the known response variable. In the high-dimensional space of the input data, the principal components represent lines, planes, or manifolds (where manifold is the generic term for these surfaces in m dimensions). These principal components are inherently linear, and where data do not follow linear patterns, PCA may be sub-optimal at capturing this non-linear variation. This is why correspondence analysis, principal coordinates, and non-metric multidimensional scaling (Legendre and Birks 2012b: Chap. 8) are popular in ecology where the input data are assumed to be inherently non-linear.

SOMs can be viewed as a non-linear two-dimensional manifold, one that is best fitted to the data in m dimensions. One of the options for choosing the starting points of a SOM grid is to select points on the two-dimensional principal component plane, which are then bent towards the data to improve the quality of fit. A number of other techniques have been developed in the last 20 years or so that generalise the problem of fitting non-linear manifolds in high dimensions. Here we discuss one particular technique – that of principal curves and surfaces.

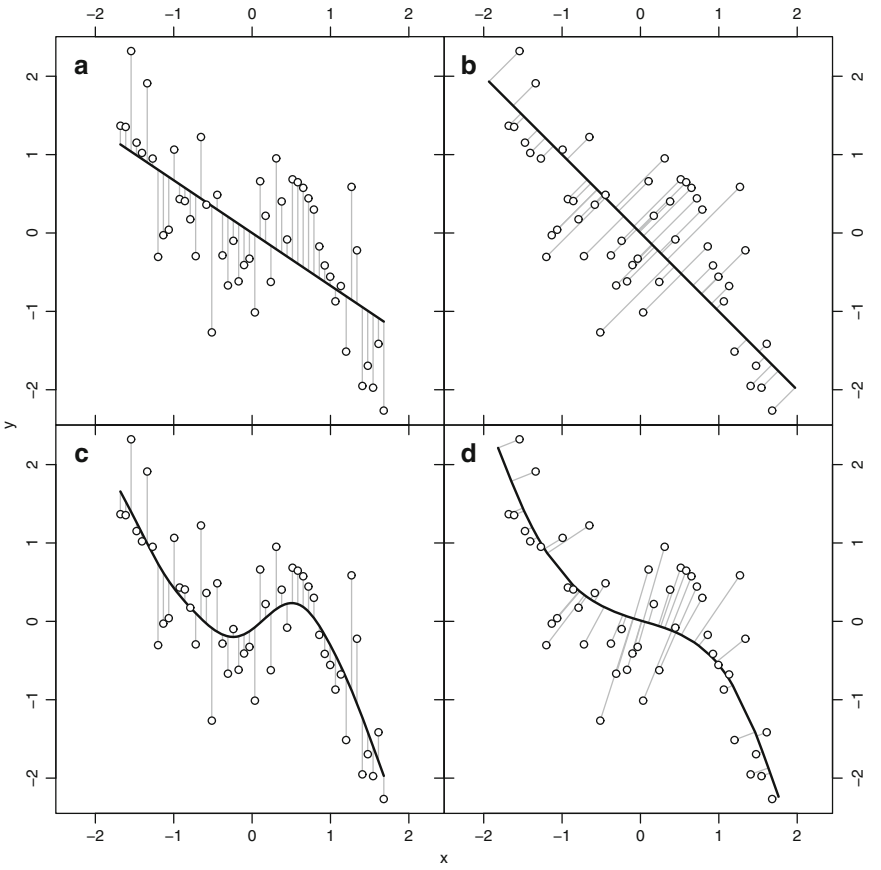

Fig. 9.22 Fitted relationship between x and y (*solid black line*) and the minimised errors (*grey line segments*) for least-squares regression (**a**), principal component analysis (**b**), cubic smoothing spline (**c**), and a principal curve (**d**). Where relevant, y is treated as the response variable and x as the predictor variable

Principal curves (PCs: Hastie and Stuetzle 1989) are a generalisation of the first principal component line, being a smooth, one-dimensional curve fitted through the input data in m dimensions such that the curve fits the data best, i.e., the distances of the samples to the PC are in some sense minimised (Hastie et al. 2011). In least-squares regression, the model lack-of-fit is computed as the sum of squared distances between the fitted values and the observations for the response variable. These errors are shown as vertical lines in Fig. 9.22a for the function

$$y = -0.9x + 2x^2 + -1.4x^3 + \varepsilon \quad \varepsilon \sim N\left(\mu = 0, \sigma = 0.05\right) \tag{9.7}$$

In PCA, the first principal component is fitted such that it minimises the lack-of-fit in terms of both the 'response' variable and the 'predictor' variable. These errors are shown in Fig. 9.22b for the function in Eq. 9.7 and are the orthogonal distances of the observations to the principal component line. We can generalise the simple least-squares regression to a smooth function of the covariates (= variables) using smoothing splines (or, for example, in a generalised additive model; Birks 2012a: Chap. 2). A smoothing spline fit to the data generated from Eq. 9.7 is shown in Fig. 9.22c. As with the least-squares regression, the lack-of-fit is measured in terms of the sum of squared distances in the response between the fitted values and the observations. Principal curves generalise the first principal component line by combining the orthogonal errors aspect of PCA with the concept of a smooth function of the covariates. A PC fitted to the data generated from Eq. 9.7 is shown in Fig. 9.22d with the errors shown as orthogonal distances between the observations and the points on the PC onto which they project. The degree of smoothness of the fitted PC is constrained by a penalty term, just as with smoothing splines (Birks 2012a: Chap. 2), and the optimal degree of smoothing is identified using a generalised cross-validation (GCV) procedure. The point on the PC to which an observation projects is the point on the curve that is closest to the observation in m dimensions.

Principal curves are fitted to data using a two-stage iterative algorithm. Initially, a starting point for each observation is determined, usually from the sample scores on the first principal component or correspondence analysis axis. These starting points define a smooth curve in the data. The first stage of the algorithm then proceeds by projecting each point in m dimensions onto a point on the initial curve to which they are closest. The distances of the projection points along the curve from one arbitrarily selected end are determined. This is known as the projection step. In the second stage of the algorithm, the local averaging step, the curve is bent towards the data such that the sum of orthogonal distances between the projection points and the observed data are reduced. This local averaging is achieved by fitting a smoothing spline to each species' abundance using distance along the curve as the single predictor variable. The fitted values of these individual smoothing splines combine to describe a new smooth curve that more closely fits the data. At this point, a self-consistency check is performed such that if the new curve is sufficiently close to the previous curve, convergence is declared to have been reached and the algorithm terminates. If the new curve is not sufficiently similar to the previous curve, the projection and local averaging steps are iterated until convergence, each time bending the curve closer to the data.

The algorithm used to fit a PC is remarkably simple, yet several choices need to be made by the user that can affect the quality of the fitted curve and ultimately the interpretation of the fitted curve. The first choice is the selection of suitable starting points for the algorithm. A logical starting point is the first principal component line, however De'ath (1999) found that better results were achieved using the first correspondence analysis (CA) axis. The second choice involves the fitting of smooth functions to the individual species during the local averaging step. Above we used the general term *smoothing splines* to describe the functions used.

Here we use a cubic smoothing spline (Birks 2012a: Chap. 2) for the example, but LOESS or kernel smoothers may also be used, as could generalised additive models (GAMs). GAMs (Birks 2012a: Chap. 2) are particularly useful when the individual species responses are not thought to be normally distributed; for example, for count abundances, a Poisson GAM may provide a better fit to each species. Whichever type of smoother is used, it is effectively a plug-in component used by the algorithm to perform the local averaging.

Having chosen a type of smoother, the degree of smoothness for the fitted PC needs to be determined. De'ath (1999) suggests that an initial smoother is fitted to each species in the data using GCV to determine, separately for each species, the degree of smoothness required for each curve. The median degree of smoothness (span or degrees of freedom) over the set of fitted smoothers is then chosen for the degree of smoothness used to fit the PC. Alternatively, the complexity of the individual smoothers fitted during the local averaging step can be allowed to vary between the different species, with GCV used to select an appropriate degree of smoothness for each species during each of the averaging steps (GL Simpson unpublished). This allows the individual smoothers to adapt to the varying degrees of response along the PC exhibited by each species; some species will respond linearly along the curve whilst others will show unimodal or skew-unimodal responses, and it seems overly restrictive to impose the same degree of smoothing to each species in such situations.

It is essential that the algorithm is monitored during fitting and that the resulting PC is explored to identify lack-of-fit. Choosing good starting locations can help with over-fitting, but overly complex, over-fitted PCs are most easily identified via examination of the final smoothers for each species, which tend to show complex fitted responses along the curve. The PC can be visualised by projecting it into a PCA of the input data. De'ath (1999) contains further advice on fitting, evaluating, and interpreting PCs.

One use of PCs is in summarising patterns of species compositional change in a stratigraphical sequence. PCA, CA, and DCA axes one and two scores are often used in palaeoecological studies to illustrate where the major changes in species composition occur (Birks 2012b: Chap. 11). Given the additional flexibility of a PC, it is likely to explain similar, or even greater, amounts of temporal compositional change in a single variable (distance along the PC) than that explained by two or more ordination axes. We illustrate the use of PCs in this setting by describing temporal compositional change in a sequence of pollen counts from Abernethy Forest for the period 12,150–5515 radiocarbon years BP (Birks and Mathewes 1978).

As the starting curve we used sample scores on the first CA axis, and fitted the PC to the data using cubic smoothing splines allowing the complexity of the individual smoothers used in the local averaging step to vary between pollen taxa, using GCV to choose the optimal degree of smoothing for each taxon. A penalty term of 1.4 was used to increase the cost of degrees of freedom in the GCV calculations. The PC converged after six iterations and is shown in Fig. 9.23, as projected onto a PCA of the pollen data. The configuration of the samples in PCA

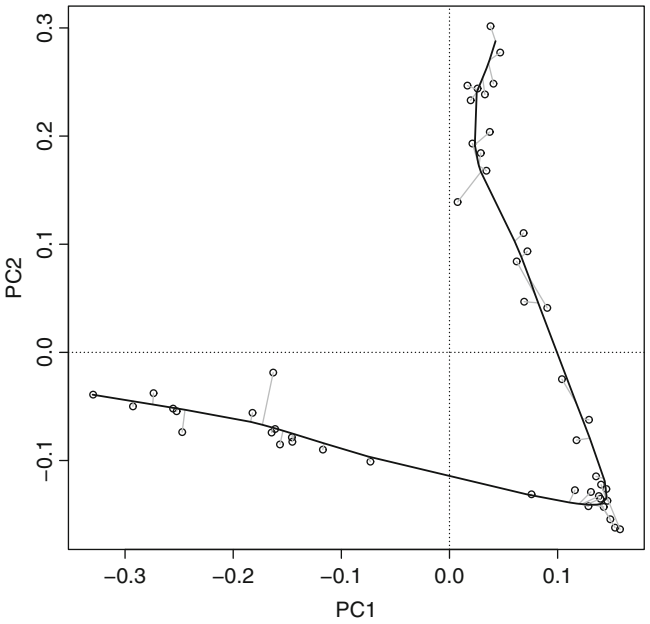

Fig. 9.23 Principal component analysis (PCA) plot of the Abernethy Forest late-glacial and early-Holocene pollen data with the fitted principal curve superimposed (*thick line*). The *thin*, *grey lines* join each observation with the point on the principal curve on to which they project, and are the distances minimised during fitting. *PC* principal component

space shows a marked horseshoe-like shape that is commonly encountered when a single, dominant gradient is projected onto 2 dimensions. The fitted PC is shown by the thick curved line in Fig. 9.21 with the orthogonal errors represented by thin segments drawn between the sample points and the curve. The PC explains 95.8% of the variation in the Abernethy Forest pollen sequence, compared with 46.5% and 30.9% for the first principal component axis and the first correspondence analysis axis, respectively. The PC accounts for substantially more of the variation in species composition than two PCA or CA axes (80.2% and 52.3%, respectively), which might conventionally be used. Figure 9.24a shows the distance along the PC between adjacent samples in the sequence expressed as a rate of change per 1000 years, clearly illustrating four periods of substantial compositional change in the pollen taxa. The actual distances along the PC are shown in Fig. 9.22b, alongside similar measures for the first PCA and CA axis scores. The total gradient described by each method has been normalised to the range (0,1) to allow a direct comparison between the three methods. Although the changes in PCA and CA axis 1 scores appear more marked, exhibiting apparently greater variation during periods of change, the PC adequately captures these periods of change but also places them within the context of overall compositional change as ∼96% of the variation in the pollen taxa is described by the PC.

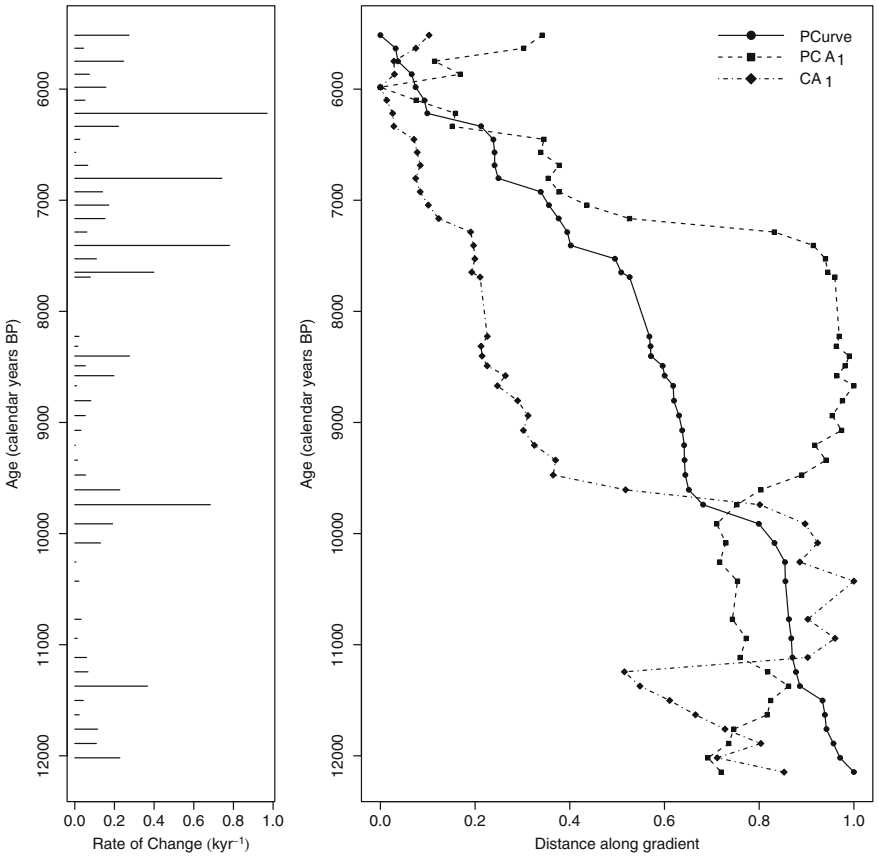

Fig. 9.24 (*left*) Distance along the principal curve expressed as a rate of change per kyr between samples for the Abernethy Forest pollen data-set. Several periods of rapid compositional change are detected. (*right*) Distance along the gradient expressed as a proportion of the total gradient for the fitted principal curve and the first ordination axes respectively of a principal component analysis (PCA) and a correspondence analysis (CA) fitted to the Abernethy Forest data

Figure 9.25 shows cubic smoothing splines fitted to the nine most abundant pollen taxa in the Abernethy Forest sequence. Each smoothing spline models the proportional abundance of the taxon as a function of the distance along the PC (expressed in temporal units). The degrees of freedom associated with each smoothing spline was taken from the smoother fitted to each taxon during the final local averaging step at convergence. As expected, given the amount of variation explained, the PC clearly captures the dynamics present in the pollen data and further illustrates that the data represent a single gradient of successive temporal change in pollen composition.

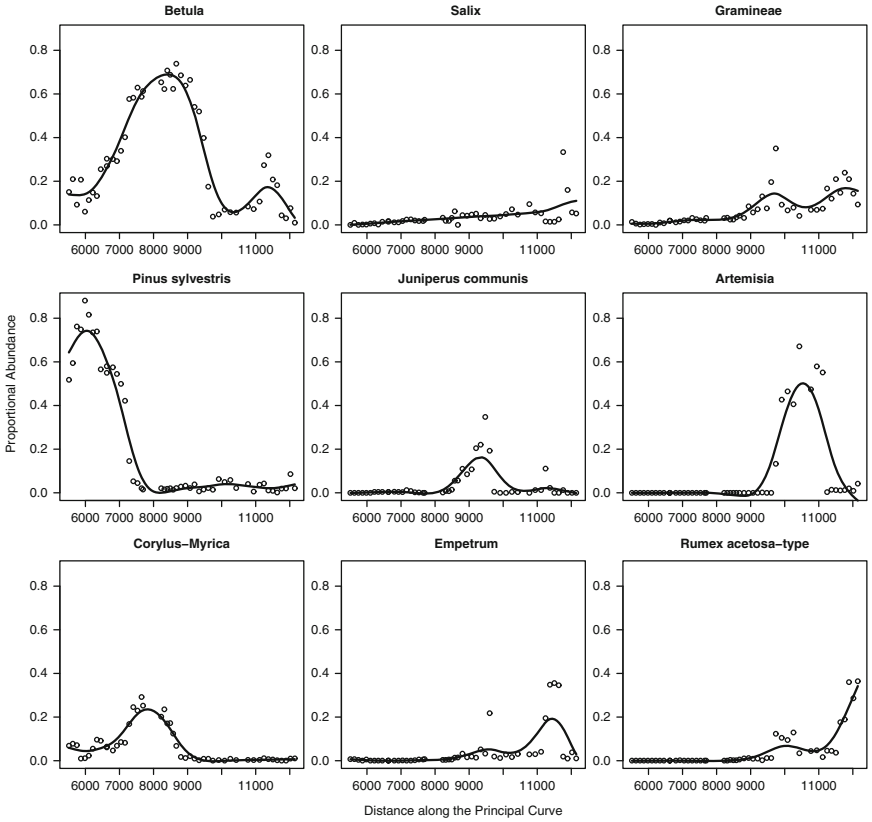

Fig. 9.25 Fitted response curves for the nine most abundant pollen taxa in the Abernethy Forest data as estimated using a principal curve. *Open circles* are the observed proportional abundance and the *solid line* is the optimised smoother from the final iteration of the principal curve. The distance along the principal curve is expressed here in radiocarbon years BP

When combined with the rate-of-change along the curve, the PC approach is far better at describing compositional change than either PCA or CA. This is particularly apparent when the stratigraphical data are best described by a single dominant, though not necessarily long, gradient. The PC degrades to the first principal component solution when all taxa are described by 1 degree-of-freedom linear functions; as a result the method can perform no worse than PCA and can, in the right circumstances, perform substantially better.

Principal curves can be generalised to principal surfaces, analogous to a plane described by the first two principal components. The algorithm described above is adapted in this case to use two-dimensional smoothers for the individual species and the projection points on the curve now become projection points on the principal surface. Higher dimensional principal surfaces can, in theory, be fitted but their

use is infrequent owing not least to problems in visualising such curves and in performing the smoothing in multiple dimensions. An unsupervised SOM is very similar to a two-dimensional principal surface, although motivated from a very different view point. Both principal surfaces and SOMs fit a manifold that is progressively warped towards the response data in order to achieve a closer fit to the data points. Geological examples of PCs include Banfield and Raftery (1992) and medical examples include Jacob et al. (1997).

Shrinkage Methods and Variable Selection

A fundamental problem in the statistical analysis of a data-set is in finding a minimal set of model terms or parameters that fit the data well (Murtaugh 2009; Birks 2012a: Chap. 2). By removing terms or parameters from the model that do not improve the fit of the model to the data we aim to produce a more easily interpretable model that is not over-fitted to the training data. The assumption that there is a single 'best' model is, in general, wrong. A more likely situation is that there will be a number of candidate models that all do a similar job in terms of explaining the training data without being over-fitted to them. Without further external criteria it may be wrong to assume that the 'best' of the candidate models is the one that describes the relationship between predictors and response for the population from which the sample of data used to fit the model was collected.

The information theoretic approach advocated by a number of authors (Burnham and Anderson 2002; Whittingham et al. 2006) proceeds by ranking candidate models in terms of the Akaike Information Criterion (AIC) and combining the terms in the various models by averaging over the set of models, and weighting each model in proportion to a likelihood function that describes the probability that each model is the best model in terms of AIC if the training data were collected again under the same circumstances (Whittingham et al. 2006). Often, AIC is used to select between nested models and the model averaging step skipped, to identify the 'best' model. In such cases, selection via AIC (or Bayesian Information Criterion (BIC), etc.) suffers from the same problems as forward-selection or backward-elimination and step-wise selection procedures, in particular, selection bias in the estimates of the model parameters. Anderson (2008) provides a gentle introduction to model-based inference.

Forward-selection and backward-elimination techniques are routinely used in ecology and palaeolimnology to prune models of unimportant terms. Starting from a model containing only an intercept term, forward selection proceeds by adding to the model that predictor variable that affords the largest reduction in model residual sum-of-squares (RSS). The procedure continues by identifying the predictor that provides the largest reduction in RSS conditional upon the previously selected terms included in the model. When the reduction in RSS afforded by inclusion of an additional predictor in the model is insignificant (usually assessed using an F-ratio test between models including and excluding the predictor, or an

information statistic such as AIC), selection stops. Backward elimination operates in a similar manner, except in reverse, starting with a model containing all the available predictor variables. The predictor whose removal from the current model would result in the smallest increase in RSS is eliminated from the model if doing so does not result in a significantly worse model. Backward elimination proceeds until either all predictors are removed from the model or no terms can be removed from the model without significantly affecting the fit to the response. An important difference is that forward selection can be performed on a model fitted to any data-set consisting of two or more predictors, whereas backward selection can only be performed on data-sets where there are $n-1$ predictors.

Step-wise selection combines both forward selection and backward elimination; at each step in the selection procedure, all single-term additions or deletions are considered and the change that results in the most parsimonious model is made subject to the condition that the added term significantly improves, or the deleted term does not significantly harm, the model fit. An alternative approach to step-wise selection is best-subsets selection, in which models using all possible combinations of predictor variables are generated and the best model of a given size, or the best model over all subsets, is selected from the set of models. The feasibility of this exhaustive search depends on the number of available predictor variables and becomes computationally difficult when only a modest number are available. The branch and bound algorithm (Miller 2002), however, allows an exhaustive search to be performed in a feasible amount of time.

There are several problems with the sequential selection and best-subsets methods, most notably (i) selection bias in the estimates of the model parameters, (ii) increased variability of the selected model, and (iii) bias in the standard errors of model parameters and its effect on the interpretation of p-values. Selection bias arises because the selection techniques described above amount to the imposition of a hard threshold on the size of the model coefficients; the estimate for a coefficient is either zero when the term is not included in the model, or some value $\hat{\beta}_i$ when included in the model. An extreme example, adapted from Whittingham et al. (2006), is shown in Fig. 9.26, where 5000 data-sets of size 10 were drawn from the model

$$y_i = 1 + 0.8x_i + \varepsilon_i \tag{9.8}$$

where x_i are the values $\{1, 2, \ldots, 10\}$ and ε_i are model errors consisting of independent Gaussian random variables with mean 0 and σ_i equal to 1. The subscripts i index the 10 observations in each data-set. In the above model, the coefficient is known ($\beta = 0.8$). Given values for x_i and y_i, we can fit a linear regression to estimate β for each of the 5000 data-sets. The distribution of the estimates for β is shown in the upper panel of Fig. 9.26 with the known value superimposed. If we set the estimates of β to zero for models where the estimate is not statistically different from 0 at the $\alpha = 0.95$ level (i.e., with a p-value >0.05) and retain those estimates that are statistically significant (i.e., those with a p-value ≤ 0.05), a process which amounts to selecting whether to include the term in the model or not, we observe

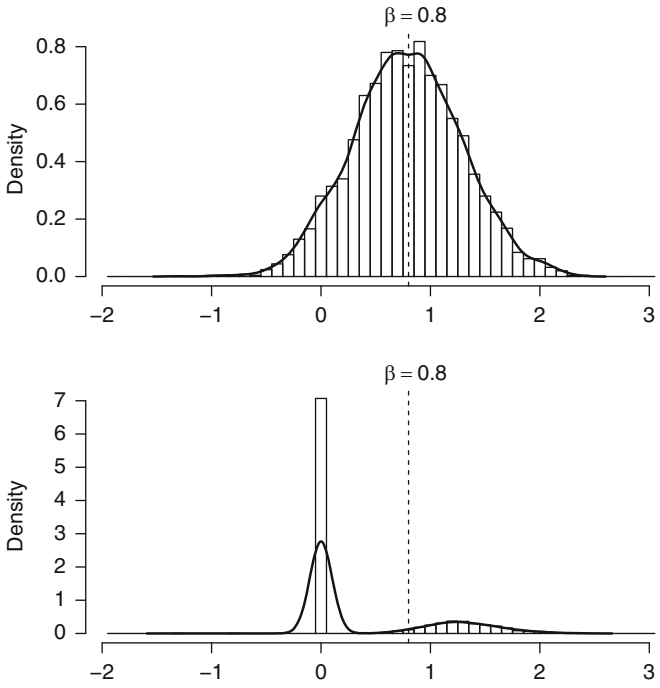

Fig. 9.26 An illustration of selection bias of regression coefficients. The *upper panel* shows the distribution of estimates of a single regression coefficient from models fitted to random samples from a model with known coefficient $\beta = 0.8$. The estimates from 5000 random draws are centred about the known value of β. If we retain the estimates of β from the 5000 random draws that are significant at the $\alpha = 0.95$ (95%) level and set the insignificant coefficients to 0, equivalent to a hard selection threshold, we observe the distribution shown in the lower panel, which contains coefficient estimates that are very different from the known value of β

the distribution of the estimates of β for the 5000 models shown in the lower panel of Fig. 9.26. Note that the retained values are all substantially different from the known population value of β; they are biased low when the term is not selected or biased high when the term is retained. No such bias occurs in the set of unselected parameter estimates (Fig. 9.26); it is the act of selection that introduces the bias and arises because the term is either included in the model or not. This bias occurs whether terms are selected using *p*-values or via some other statistic, such as AIC.

Models resulting from forward selection and/or backward elimination are prone to increased variance, and hence, ultimately higher model error (Mundry and Nunn 2009). The argument behind this statement is the same as that used to explain the instability of single tree-based models (see above). Small changes in the sample data may lead to a different variable entering the model in the early stages of selection, especially if there are two or more predictors that have similar predictive ability as in the case of collinear predictors. The resultant model may be over-fitted to the training sample and generalise poorly when making predictions for other

observations from the population. Such models are said to have high variability; the uncertainty in the predicted values is large.

An often overlooked issue with model selection is that the standard errors of the estimated coefficients in a selected model are biased and too small, suggesting apparent precision in their estimation; their construction knows nothing of the previous, often convoluted, selection process that led to the selected model. Consequently, test statistics and their p-values are too optimistic and the possibility of making a Type I error is increased. It is not clear how this bias can be corrected for in a practical sense (Hastie et al. 2011). This problem affects best-subsets selection as well as forward selection/backward elimination.

Model selection often results in models that contain too many parameters unless steps are taking during selection to manage the entry of variables to the model. Consider the situation where a p-value threshold of 0.05 is used to decide whether to include a variable in a model at each stage of a forward-selection procedure. Each of the tests performed to decide whether to include the predictor or not is subject to a Type I error-rate of 0.05, and as such the final model has a much larger Type I error-rate. A correction to the p-value used in each test may be made, to guard against this inflated Type I error-rate. For example, a Bonferroni-type correction can be made of p/t, where p is the user-selected p-value threshold (0.05 in the above discussion) and t is the number of tests conducted thus far. In deciding whether to include the first predictor variable, using 0.05 as the threshold for inclusion, the variable is included if it achieves a p-value of $0.05/1 = 0.05$ or lower. For the second variable to enter the model it must achieve a p-value of $0.05/2 = 0.025$ or lower to be selected, and so on for the subsequent rounds of selection. Using BIC instead of AIC to decide on inclusion or elimination penalises more-complex models to a stronger degree and thus may help to guard against selecting overly complex models.

Correlated predictors, as well as making model selection more difficult, cause additional problems in estimating model coefficients; they are poorly determined and have high variance (large standard errors). Consider two correlated predictors; a large positive value as the estimate for the model coefficient for one of the predictors can be counteracted by a large negative coefficient for the other predictor (Hastie et al. 2011). If the interest in fitting the model is to interpret the coefficients to shed light on ecological or environmental mechanisms, spurious inflation of effects due to multicollinearity, if undetected, may lead to erroneous statements about the mechanisms under study.

There are a number of approaches that can be applied to help with model selection and collinearity problems. These approaches are known as shrinkage methods. Two shrinkage techniques familiar to palaeolimnologists are principal components regression (PCR) and partial least squares (PLS) (Martens and Næes 1989; Birks 1995; Næs et al. 2002; Juggins and Birks 2012: Chap. 14). In both approaches, the aim is to identify a small number of orthogonal (uncorrelated) components that explain maximal amounts of variance in the predictors (PCR) or maximal amounts of the covariance between the response and predictors (PLS). Predictors that exhibit low variance (PCR) or are unrelated to the response (PLS) will have low weights in the components retained for modelling; in a sense, the

coefficients for these variables have been shrunk from their least-squares estimates (Hastie et al. 2011). PCR and PLS are also useful simplification techniques in situations where there are many more predictor variables than observations, as in chemometrics (Wehrens 2011). However, these techniques suffer in terms of model interpretation; the regression coefficients no longer apply to individual predictors but to linear combinations of the predictors. If the aim of modelling is prediction, and not explanation, then the aim of selecting a minimal adequate model is to achieve lower prediction error, and PCR or PLS are useful techniques.

PCR and PLS impose a size constraint on the coefficients of predictors in the model by retaining a small number of orthogonal components as predictors in the model. Information on those variables that are useful in predicting the response or have high variance is retained, whilst those variables unrelated to the response or have low variance are discarded – their coefficients are effectively, or close to, 0 (Hastie et al. 2011). A number of other techniques have been proposed that also impose size restrictions on model coefficients, namely ridge regression (Hoerl and Kennard 1970; Copas 1983, Hastie et al. 2011), the lasso (Tibshirani 1996; Hastie et al. 2011), and a technique known as the elastic net which combines ridge-like and lasso-like constraints (Zou and Hastie 2005; Hastie et al. 2011).

Ridge regression was proposed as a means to handle collinearity in the set of available predictors. Earlier we saw that two correlated variables may have large coefficients but of opposite sign. Imposing a constraint on the size of the model coefficients helps to alleviate this problem. Ridge regression imposes a quadratic constraint on the size of the coefficients, but can also be seen to shrink components of the predictors that have low variance, in other words, that explain low amounts of the variance in the set of predictors available (Hastie et al. 2011). Ridge regression coefficients β_{ridge} are chosen to minimise a penalised RSS criterion.

$$\beta_{\text{ridge}} = \arg\min_{\beta} \left\{ \sum_{i=1}^{n} \left(y_i - \beta_0 - \sum_{j=1}^{p} x_{ij} \beta_j \right)^2 + \lambda \sum_{j=1}^{p} \beta_j^2 \right\} \tag{9.9}$$

The first term in the braces is the RSS and the second term is the quadratic penalty imposed on the ridge coefficients. Equivalently, in ridge regression, the estimated coefficients minimise the RSS subject to the constraint that $\sum_{j=1}^{p} \beta_j^2 \leq t$ where t is a threshold limiting the size of the coefficients. There is a one-to-one relationship between λ and t; as λ is increased, indicating greater penalty, t is reduced, indicating a lower threshold on the size of the coefficients (Hastie et al. 2011). Software used to fit ridge regression solves the penalised RSS criterion for a range of values of either λ or t and cross-validation is used to identify the value of λ or t that has the lowest prediction error. Note that the model intercept (β_0) is not included in the penalty and that the predictor variables are standardised to zero mean and unit variance before estimation of the ridge coefficients. Where $\lambda = 0$, the ridge coefficients are equivalent to the usual least-squares estimates of the model coefficients.

It is important to note that ridge regression does not perform variable selection; all available predictor variables remain in the model, it is just their coefficients that are shrunk away from the least-squares estimates. The lasso (Tibshirani 1996) is related to ridge regression but can also perform variable selection because it employs a different penalty on the coefficients to that of the ridge penalty. The lasso (least absolute shrinkage and selection operator) imposes a restriction on the size of the absolute values of the coefficients instead of a restriction on the squared values of the coefficients used in ridge regression. The lasso finds coefficients β_{lasso} that minimise the following penalised RSS criterion

$$\beta_{\text{lasso}} = \arg\min_{\beta} \left\{ \frac{1}{2} \sum_{i=1}^{n} \left(y_i - \beta_0 - \sum_{j=1}^{p} x_{ij} \beta_j \right)^2 + \lambda \sum_{j=1}^{p} |\beta_j| \right\} \tag{9.10}$$

which is equivalent to minimising the RSS subject to the constraint that $\sum_{j=1}^{p} |\beta_j| \leq t$ (Hastie et al. 2011). This penalty allows variables whose coefficients are shrunk to zero to be removed from the model. As before, cross-validation is used to identify the value of λ or t with the lowest prediction error. It can be shown that ridge regression shrinks all coefficients proportionally, and the lasso shrinks each coefficient by a constant factor λ and truncates at zero (e.g., a positive coefficient that would otherwise go negative when shrunk by the factor λ is removed from the model). The lasso is a general technique and has been successfully applied to generalised linear models (Tibshirani 1996) and is used as a form of shrinkage in boosted trees (De'ath 2007). A fast computer algorithm, least angle regression (LARS) was developed by Efron et al. (2004) that can compute the entire lasso path from no predictors in the model to the full least-squares solution for the same computational cost as the least-squares solution. Park and Hastie (2007) have developed similar path algorithms for the lasso in a GLM setting.

Ridge regression shrinks the coefficients of correlated predictors and the lasso selects predictors via shrinkage. Ideally, these two characteristics would be combined into a single technique that handles correlated predictors and could perform model selection. This is exactly what the elastic-net penalty does, via a weighted combination of ridge-like and lasso-like penalties to form the elastic-net penalty

$$\lambda \sum_{j=1}^{p} \left(\alpha \beta_j^2 + (1 - \alpha) \left| \beta_j \right| \right) \tag{9.11}$$

where α controls the relative weighting of ridge-like and lasso-like penalties (Zou and Hastie 2005). Where there are correlated predictors, the elastic net will tend to shrink the coefficients for those predictors rather than necessarily dropping one of the predictors giving full weight in the model to the other predictor, which is how

the lasso operates with collinear variables. Friedman et al. (2010) demonstrate an efficient path algorithm for fitting the elastic net regularisation path for GLMs.

Figure 9.27 shows ridge regression (Fig. 9.27a), lasso (Fig. 9.27b), and elastic net (Fig. 9.27c) regularisation paths for the ozone data considered in the MARS example earlier. The models were fitted to the log-transformed ozone concentration because gamma GLMs are not supported in the glmnet R package (version 1.6: Friedman et al. 2010) used here. We consider only the main effects of the nine predictor variables, and for the elastic net we use $\alpha = 0.5$, indicating equal amounts of ridge-like and lasso-like penalties. The left-hand panels of each figure show the regularisation path with the full least-squares solutions on the right of these plots; the y-axis represents the values of the coefficients for each predictor, whilst the lines on the plots describe how the values of the coefficients vary from total shrinkage to the their least-squares values. The right-hand panels show *k*-fold cross-validated mean squared error (MSE) for each regularisation path, here expressed on the $\log(\lambda)$ scale. The numbers on the top of each plot indicate the complexity of the models along the regularisation path or as a function of $\log(\lambda)$. For ridge regression, we note that all nine predictor variables remain in the model throughout the path, whereas for the lasso and elastic-net paths predictors are selected out of the model as an increasing amount of regularisation is applied.

An interesting feature of the ridge-regression path is the coefficient value for wind speed, which is negative in the least-squares solution but becomes positive after a small amount of shrinkage, before being shrunk back to zero as a stronger penalty is applied to the size of the coefficients. The coefficient value for wind speed does not show this pattern in either the lasso or the elastic-net regularisation paths because of the property that both these penalties share, whereby coefficients are truncated at zero and not allowed to change their sign. The elastic-net regularisation path is intermediate between those of the ridge and lasso, although it is most similar to the lasso path. The effect of the lower lasso-like penalty in the elastic-net path for the ozone model is for predictor variables to persist in the model until a higher overall penalty is applied than under the lasso path. However, whilst the nine predictors persist in the path for longer, the ridge part of the penalty is shrinking the size of the coefficients.

The right-hand panels in Fig. 9.27 indicate the optimal degree of shrinkage by identifying the value of λ that affords the lowest CV MSE (the left vertical line) or that is within one standard error of the minimum (the right vertical line). On these plots, model complexity *increases* from left to right. The optimal amount of shrinkage indicates that nine, five, and seven predictors should be included in the model for the ridge regression, lasso, and elastic-net penalties, respectively. Temperature is the most important variable in predicting the log ozone concentration, followed by humidity. At larger penalties in the lasso and elastic-net paths, pressure gradient replaces humidity as the second predictor, after temperature, to be selected in the model. We do not interpret these models further here.

This is an area of considerable research activity, much of which is of direct relevance to ecologists and palaeolimnologists but whose importance is poorly known (e.g., Dahlgren 2010). For example, ter Braak (2009) has developed a new

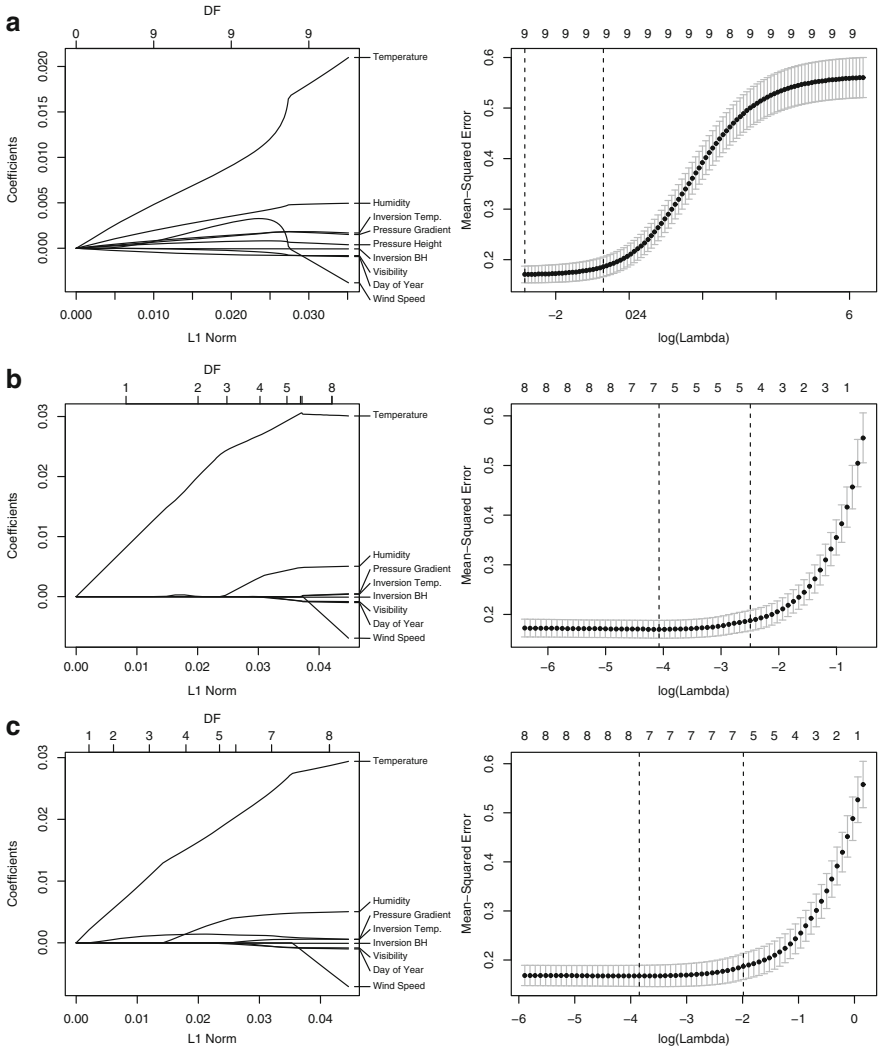

Fig. 9.27 Illustration of three shrinkage methods fitted to the ozone concentration data; (**a**) ridge regression, (**b**) the lasso, (**c**) the elastic net with $\alpha = 0.5$. The *left-hand panels* show the estimates of the regression coefficients for the entire regularisation path estimated, with the least complex model to the left. Estimates of the degrees of freedom associated with various values of the penalty are show on the upper axis of each panel. The *right-hand panels* show k-fold cross-validated model error for increasing (*left to right*) penalty. *Error bars* show the range of model errors across the k folds for each value of the penalty. The best model, with lowest mean squared error is highlighted by the *left-most dashed vertical line* in each panel, whilst the simplest model within one standard error of the best model is shown by the *right-most vertical line*. The values on the upper axis of each panel indicate the number of covariates included in the model for the value of the penalty

regression method, regularisation of smart contrasts and sums (ROSCAS), that outperforms the lasso, elastic net, ridge regression, and PLS when there are groups of predictors with each group representing an independent feature that influences the response and when the groups differ in size.

Discussion and Conclusions

This chapter has described several statistical machine-learning techniques, which can be loosely categorised into supervised and unsupervised learning techniques. The discussion for individual methods was intentionally brief, with the aim of introducing palaeolimnologists to the key features of the main machine-learning methods and illustrating their use. The references cited in each section should provide access to additional sources of information on each technique, and wherever possible we have referred to relevant palaeoecological or ecological papers.

A recurring theme in this chapter has been the reduction of bias, variance, or both in order to identify a model that has low prediction error. Given a model, $y = f(x) + \varepsilon$, that relates a response y to covariate x, we define the prediction error of a model as the expected difference between the true, unknown value of the response (y_0) and the predicted value for the response from the model, $\hat{f}(x)$. This prediction error can be decomposed into three components; (i) bias^2, (ii) variance, and (iii) ε, the irreducible error present even if we knew the true $f(x)$. We are unable to do anything about ε, so we must concern ourselves with trying to reduce bias, variance, or both in order to reduce prediction error. The bias^2 and variance together yield the mean squared error of the model (MSE).

To understand what each of these components is, consider a simple regression model fitted to a response y and covariate x. The relationship is quadratic and we have five observations. A simple least-squares model using one degree of freedom fitted to the data will yield predictions that follow a straight line. This model is very simple, but the straight line does not fit the data well; the model under-fits the data. Such a model will have high bias; over large parts of the observed data, the model systematically fails to capture the true relationship between x and y. Alternatively, we could fit a high degree polynomial that interpolates the training data perfectly, thus having zero bias. This is a more complex model but it over-fits the training data and is unlikely to generalise well to new observations for which we want to predict y. Such a model has high variance; each coefficient in the model has a high degree of uncertainty because we have used all the data to fit a large number of coefficients. In between these extremes is a model that has higher bias than the over-fitted model and lower bias than the simple model and the opposite features for model variance. Figure 9.28 illustrates this bias–variance tradeoff.

Several methods that we have introduced focus on reducing the variance part of MSE, such as bagged trees, random forests, and model averaging in an information theoretic framework. Shrinkage methods, introduced towards the end of the chapter,

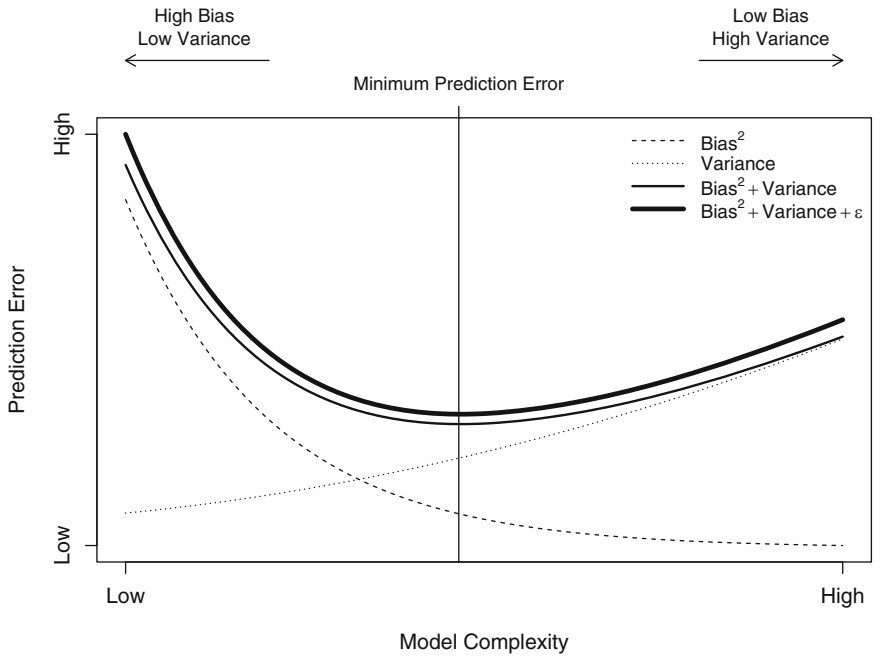

Fig. 9.28 Illustration of the bias–variance trade-off. At low complexity, models under-fit the observations and consequently have high bias which dominates the prediction error. At high complexity, models over-fit the data and as a result have low bias but high variance and the variance component dominates prediction error. Often the aim in statistical machine-learning is to fit a model that has minimal prediction error. Identifying such a model will require trading off bias against variance to achieve an overall lower prediction error. $\text{Bias}^2 + \text{Variance} = \text{MSE}$ (mean squared error). ε is the irreducible error that is present in the model even if one knew the true relationship between the response and the predictors rather than having to estimate it

sacrifice a small increase in model bias (the estimates of regression coefficients using the methods are biased) for a larger reduction in model variance by shrinking coefficient estimates to zero. Of the methods discussed, only boosting has the potential to reduce both the bias and the variance of the fitted model. Bias is reduced by focussing on those observations that are poorly fitted by previous trees in the ensemble, whilst variance is reduced by averaging predictions over a large ensemble of trees.

Understanding the bias–variance trade-off is key to the successful use of statistical machine-learning where the focus is on producing a model for prediction that has the lowest possible prediction error given the available training data.

One feature of all of the techniques discussed is that they use the power of modern computers to learn aspects of the training data that allows the model to make accurate predictions. How well one of these algorithms or methods performs tends to be evaluated on the basis of its ability to predict the response variable on an independent test-set of samples. However, many, if not the majority of the

techniques we describe do now have a thorough statistical underpinning (Hastie et al. 2011). This is especially so for the tree-based methods and boosting in particular.

What we have not been able to do here is illustrate *how* to go about fitting these sorts of models to data. Clearly, the availability of suitable software environments and code that implements these modern machine-learning methods is a prerequisite. All of the detailed examples have been performed by the authors with the R statistical software (version 2.13.1 patched r56332: R Core Development Team 2011) using a variety of add-on packages available on the Comprehensive R Archive Network (CRAN). A series of R scripts are available from the book website which replicate the examples used in this chapter and demonstrate how to use R and the add-on packages to fit the various models. We have used R because it is free and open source, and because of the availability of high-quality packages that implement all the machine-learning methods we have discussed. Other computational statistical software packages, such as MATLAB®, should also be able to fit most if not all the methods described here.

The technical and practical learning curves are far steeper for software such as R and the statistical approaches we discuss than the usual suspects of ordination, clustering, and calibration most commonly employed by palaeolimnologists. Machine-learning methods tend to place far higher demands on the user to get the best out of the techniques. One might reasonably ask if this additional effort is worthwhile? Ecological and palaeoecological data are inevitably noisy, complex, and high-dimensional. The sorts of machine-learning tools we have introduced here were specifically designed to handle such data and are likely to perform as well if not better than the traditional techniques most commonly used in the palaeolimnological realm. Furthermore, if all one knows is how to use CANOCO or C2 there will be a tendency to view all problems as ordination, calibration, or something else that cannot be handled. This situation is succinctly described as Maslow's Hammer; "it is tempting, if the only tool you have is a hammer, to treat every problem as if it were a nail" (Maslow 1966: p.15).

This chapter aims to provide an introduction to the statistical machine-learning techniques that have been shown to perform well in a variety of settings. We hope that it will suitably arm palaeolimnologists with the rudimentary knowledge required to know when to put down the hammer and view a particular problem as something other than a nail.

Acknowledgements We are indebted to Richard Telford, Steve Juggins, and John Smol for helpful comments and/or discussion. Whilst writing this chapter, GLS was supported by the European Union Seventh Framework Programme projects REFRESH (Contract N. 244121) and BioFresh (Contract No. 226874), and by the UK Natural Environment Research Council (grant NE/G020027/1). We are particularly grateful to Cathy Jenks for her editorial help. This is publication A359 from the Bjerknes Centre for Climate Research.

References

Aalders I (2008) Modeling land-use decision behavior with Bayesian belief networks. Ecol Soc 13:16

Aho K, Weaver T, Regele S (2011) Identification and siting of native vegetation types on disturbed land: demonstration of statistical methods. Appl Veg Sci 14:277–290

Allan JD, Yuan LL, Black P, Stockton T, Davies PE, Magierowski RH, Read SM (2011) Investigating the relationships between environmental stressors and stream conditions using Bayesian belief networks. Freshw Biol. doi:10.1111/j.1365-2427.2011.02683.x

Amsinck SL, Strzelczak A, Bjerring R, Landkildehus F, Lauridsen TL, Christoffersen K, Jeppesen E (2006) Lake depth rather than fish planktivory determines cladoceran community structure in Faroese lakes – evidence from contemporary data and sediments. Freshw Biol 51:2124–2142

Anderson DR (2008) Model based inference in the life sciences: a primer on evidence. Springer, New York

Anderson RP, Lew D, Peterson AT (2003) Evaluating predictive models of species' distributions: criteria for selecting optimal models. Ecol Model 162:211–232

Baker FA (1993) Classification and regression tree analysis for assessing hazard of pine mortality caused by *Heterobasidion annosum*. Plant Dis 77:136–139

Balshi MS, McGuire AD, Duffy P, Flannigan M, Walsh J, Melillo J (2009) Assessing the response of area burned to changing climate in western boreal North America using a Multivariate Adaptive Regression Splines (MARS) approach. Global Change Biol 15:578–600

Banfield JD, Raftery AE (1992) Ice floe identification in satellite images using mathematical morphology and clustering about principal curves. J Am Stat Assoc 87:7–16

Barrows TT, Juggins S (2005) Sea-surface temperatures around the Australian margin and Indian Ocean during the Last Glacial Maximum. Quaternary Sci Rev 24:1017–1047

Barton AM, Nurse AM, Michaud K, Hardy SW (2011) Use of CART analysis to differentiate pollen of red pine *(Pinus resinosa*) and jack pine (*P. banksiana*) in New England. Quaternary Res 75:18–23

Belgrano A, Malmgren BA, Lindahl O (2001) Application of artificial neural networks (ANN) to primary production time-series data. J Plankton Res 23:651–658

Benito Garzón M, Blazek R, Neteler M, Sánchez de Dios R, Sainz Ollero H, Furlanello C (2006) Predicting habitat suitability with machine learning models: the potential area of *Pinus sylvestris* L. in the Iberian Peninsula. Ecol Model 197:383–393

Benito Garzón M, Sánchez de Dios R, Sainz Ollero H (2007) Predictive modelling of tree species distributions on the Iberian Peninsula during the Last Glacial Maximum and Mid-Holocene. Ecography 30:120–134

Benito Garzón M, Sánchez de Dios R, Sainz Ollero H (2008) Effects of climate change on the distribution of Iberian tree species. Appl Veg Sci 11:169–178

Birks HH, Mathewes RW (1978) Studies in the vegetational history of Scotland. V. Late Devensian and early Flandrian pollen and macrofossil stratigraphy at Abernethy Forest, Inverness-shire. New Phytol 80:455–484

Birks HJB (1995) Quantitative palaeoenvironmental reconstructions. In: Maddy D, Brew J (eds) Statistical modelling of Quaternary science data. Volume 5: Technical guide. Quaternary Research Association, Cambridge, pp 161–254

Birks HJB (2012a) Chapter 2: Overview of numerical methods in palaeolimnology. In: Birks HJB, Lotter AF, Juggins S, Smol JP (eds) 2012. Tracking environmental change using lake sediments. Volume 5: Data handling and numerical techniques. Springer, Dordrecht

Birks HJB (2012b) Chapter 11: Stratigraphical data analysis. In: Birks HJB, Lotter AF, Juggins S, Smol JP (eds) Tracking environmental change using lake sediments. Volume 5: Data handling and numerical techniques. Springer, Dordrecht

Birks HJB, Gordon AD (1985) Numerical methods in Quaternary pollen analysis. Academic Press, London

Birks HJB, Jones VJ (2012) Chapter 3: Data-sets. In: Birks HJB, Lotter AF, Juggins S, Smol JP (eds) Tracking environmental change using lake sediments. Volume 5: Data handling and numerical techniques. Springer, Dordrecht

Birks HJB, Line JM, Juggins S, Stevenson AC, ter Braak CJF (1990) Diatoms and pH reconstruction. Philos Trans R Soc Lond B 327:263–278

Bishop CM (1995) Neural networks for pattern recognition. Clarendon, Oxford

Bishop CM (2007) Pattern recognition and machine learning. Springer, Dordrecht

Bjerring R, Becares E, Declerck S et al. (2009) Subfossil Cladocera in relation to contemporary environmental variables in 54 pan-European lakes. Freshw Biol 54:2401–2417

Blaauw M, Heegaard E (2012) Chapter 12: Estimation of age-depth relationships. In: Birks HJB, Lotter AF, Juggins S, Smol JP (eds) Tracking environmental change using lake sediments. Volume 5: Data handling and numerical techniques. Springer, Dordrecht

Borggaard C, Thodberg HH (1992) Optimal minimal neural interpretation of spectra. Anal Chem 64:545–551

Bourg NA, McShea WJ, Gill DE (2005) Putting a CART before the search: successful habitat prediction for a rare forest herb. Ecology 86:2793–2804

Breiman L (1996) Bagging predictors. Mach Learn 24:123–140

Breiman L (2001) Random forests. Mach Learn 45:5–32

Breiman L, Friedman JH, Olshen RA, Stone CJ (1984) Classification and regression trees. Wadsworth, Belmont

Brosse S, Guégan J-F, Tourenq J-N, Lek S (1999) The use of artificial neural networks to assess fish abundance and spatial occupancy in the littoral zone of a mesotrophic lake. Ecol Model 120:299–311

Brunelle A, Rehfeldt GE, Bentz B, Munson AS (2008) Holocene records of *Dendroctonus* bark beetles in high elevation pine forests of Idaho and Montana, USA. Forest Ecol Manage 255:836–846

Burman P, Chow E, Nolan D (1994) A cross-validatory method for dependent data. Biometrika 81:351–358

Burnham KP, Anderson DR (2002) Model selection and multimodel inference: a practical information-theoretic approach, 2nd edn. Springer, New York

Cairns DM (2001) A comparison of methods for predicting vegetation type. Plant Ecol 156:3–18

Caley P, Kuhnert PM (2006) Application and evaluation of classification trees for screening unwanted plants. Austral Ecol 31:647–655

Carlisle DM, Wolock DM, Meador MR (2011) Alteration of streamflow magnitudes and potential ecological consequences: a multiregional assessment. Front Ecol Environ 9:264–270

Castelletti A, Soncini-Sessa R (2007a) Bayesian networks and participatory modelling in water resource management. Environ Model Softw 22:1075–1088

Castelletti A, Soncini-Sessa R (2007b) Coupling real-time and control and socio-economic issues in participatory river basin planning. Environ Model Softw 22:1114–1128

Céréghino R, Giraudel JL, Compin A (2001) Spatial analysis of stream invertebrates distribution in the Adour-Garonne drainage basin (France), using Kohonen self-organizing maps. Ecol Model 146:167–180

Černá L, Chytrý M (2005) Supervised classification of plant communities with artificial neural networks. J Veg Sci 16:407–414

Chapman DS (2010) Weak climatic associations among British plant distributions. Global Ecol Biogeogr 19:831–841

Chapman DS, Purse BV (2011) Community versus single-species distribution models for British plants. J Biogeogr 38:1524–1535

Chapman DS, Bonn A, Kunin WE, Cornell SJ (2010) Random forest characterization of upland vegetation and management burning from aerial imagery. J Biogeogr 37:37–46

Chatfield C (1993) Neural networks: forecasting breakthrough or passing fad? Int J Forecast 9:1–3

Chon T-S (2011) Self-organising maps applied to ecological sciences. Ecol Inform 6:50–61

Chytrý M, Jarošik V, Pyšek P, Hájek O, Knollová I, Tichý L, Danihelka J (2008) Separating habitat invasibility by alien plants from the actual level of invasion. Ecology 89:1541–1553

Copas JB (1983) Regression, prediction and shrinkage. J R Stat Soc Ser B 45:311–354
Cutler A, Stevens JR (2006) Random forests for microarrays. Methods Enzymol 411:422–432
Cutler DR, Edwards TC, Beard KH, Cutler A, Hess KT, Gibson J, Lawler JJ (2007) Random forests for classification in ecology. Ecology 88:2783–2792
Dahlgren JP (2010) Alternative regression methods are not considered in Murtaugh (2009) or by ecologists in general. Ecol Lett 13:E7–E9
Davidson TA, Sayer CD, Perrow M, Bramm M, Jeppesen E (2010a) The simultaneous inference of zooplanktivorous fish and macrophyte density from sub-fossil cladoceran assemblages: a multivariate regression tree approach. Freshw Biol 55:546–564
Davidson TA, Sayer CD, Langdon PG, Burgess A, Jackson MJ (2010b) Inferring past zooplanktivorous fish and macrophyte density in a shallow lake: application of a new regression tree model. Freshw Biol 55:584–599
De'ath G (1999) Principal curves: a new technique for indirect and direct gradient analysis. Ecology 80:2237–2253
De'ath G (2002) Multivariate regression trees: a new technique for modeling species-environment relationships. Ecology 83:1108–1117
De'ath G (2007) Boosted trees for ecological modeling and prediction. Ecology 88:243–251
De'ath G, Fabricius KE (2000) Classification and regression trees: a powerful yet simple technique for ecological data analysis. Ecology 81:3178–3192
De'ath G, Fabricius KE (2010) Water quality as a regional driver of coral biodiversity and macroalgae on the Great Barrier Reef. Ecol Appl 20:840–850
DeFries RS, Rudel T, Uriarte M, Hansen M (2010) Deforestation driven by urban population growth and agricultural trade in the twenty-first century. Nat Geosci 3:178–181
Despagne F, Massart D-L (1998) Variable selection for neural networks in multivariate calibration. Chemometr Intell Lab Syst 40:145–163
D'heygere T, Goethals PLM, de Pauw N (2003) Use of genetic algorithms to select input variables in decision tree models for the prediction of benthic macroinvertebrates. Ecol Model 160:291–300
Dobrowski SZ, Thorne JH, Greenberg JA, Safford HD, Mynsberge AR, Crimins SM, Swanson AK (2011) Modeling plant ranges over 75 years of climate change in California, USA: temporal transferability and species traits. Ecol Monogr 81:241–257
Dutilleul P, Cumming BF, Lontoc-Roy M (2012) Chapter 16: Autocorrelogram and periodogram analyses of palaeolimnological temporal series from lakes in central and western North America to assess shifts in drought conditions. In: Birks HJB, Lotter AF, Juggins S, Smol JP (eds) Tracking environmental change using lake sediments. Volume 5: Data handling and numerical techniques. Springer, Dordrecht
Efron B (1979) Bootstrap methods: another look at the jackknife. Ann Stat 7:1–26
Efron B, Tibshirani R (1991) Statistical data analysis in the computer age. Science 253:390–395
Efron B, Tibshirani R (1993) An introduction to the bootstrap. Chapman & Hall, London
Efron B, Hastie T, Johnstone I, Tibshirani R (2004) Least angle regression. Ann Stat 32:407–499
Elith J, Burgman M (2002) Predictions and their validation: rare plants in the Central Highlands, Victoria, Australia. In: Scott JM, Heglund P, Morrison ML, Raven PH (eds) Predicting species occurrences: issues of accuracy and scale. Island Press, Washington, DC
Elith J, Leathwick JR (2007) Predicting species distributions from museum and herbarium records using multiresponse models fitted with multivariate adaptive regression splines. Divers Distrib 13:265–275
Elith J, Graham CH, Anderson RP, Dudík M, Ferrier S, Guisan A et al. (2006) Novel methods improve prediction of species' distributions from occurrence data. Ecography 29:129–151
Elith J, Leathwick JR, Hastie T (2008) A working guide to boosted regression trees. J Anim Ecol 77:802–813
Fielding AH (2007) Cluster and classification techniques for the biosciences. Cambridge University Press, Cambridge
Franklin J (1998) Predicting the distribution of shrub species in southern California from climate and terrain-derived variables. J Veg Sci 9:733–748

Franklin J (2010) Mapping species distributions — spatial inference and prediction. Cambridge University Press, Cambridge
Freund Y (1995) Boosting a weak learning algorithm by majority. Inf Comput 121:256–285
Friedman JH (1991) Multivariate adaptive regression splines. Ann Stat 19:1–67
Friedman JH (2001) Greedy function approximation: a gradient boosting machine. Ann Stat 29:1189–1232
Friedman JH (2002) Stochastic gradient boosting. Comput Stat Data Anal 38:367–378
Friedman G, Meulman JJ (2003) Multivariate adaptive regression trees with application in epidemiology. Stat Med 22:1365–1381
Friedman JH, Hastie T, Tibshirani R (2000) Additive logistic regression: a statistical view of boosting. Ann Stat 28:337–407
Friedman J, Hastie T, Tibshirani R (2010) Regularization paths for generalized linear models via coordinate descent. J Stat Software 33:1–22
Furlanello C, Neteler M, Merler S, Menegon S, Fontanari S, Donini A, Rizzoli A, Chemini C (2003) GIS and the random forests predictor: integration in R for tick-borne disease risk. In: Hornik K, Leitch F, Zeileis A (eds) Proceedings of the third international workshop on distributed statistical computings, Vienna, Austria. pp 1–11
Gevrey M, Dimopoulos I, Lek S (2003) Review and comparison of methods to study the contribution of variables in artificial neural network models. Ecol Model 160:249–264
Giraudel JL, Lek S (2001) A comparison of self-organising map algorithm and some conventional statistical methods for ecological community ordination. Ecol Model 146:329–339
Gordon AD (1973) Classifications in the presence of constraints. Biometrics 29:821–827
Gordon AD, Birks HJB (1972) Numerical methods in Quaternary palaeoecology. I. Zonation of pollen diagrams. New Phytol 71:961–979
Gordon AD, Birks HJB (1974) Numerical methods in Quaternary palaeoecology. II. Comparison of pollen diagrams. New Phytol 73:221–249
Goring S, Lacourse T, Pellatt MG, Walker IR, Matthewes RW (2010) Are pollen-based climate models improved by combining surface samples from soil and lacustrine substrates? Rev Palaeobot Palynol 162:203–212
Grieger B (2002) Interpolating paleovegetation data with an artificial neural network approach. Global Planet Change 34:199–208
Guégan J-F, Lek S, Oberdorff T (1998) Energy availability and habitat heterogeneity predict global riverine fish diversity. Nature 391:382–384
Hastie T, Stuetzle W (1989) Principal curves. J Am Stat Assoc 84:502–516
Hastie T, Tibshirani R, Friedman J (2011) The elements of statistical learning, 2nd edn. Springer, New York
Haykin S (1999) Neural networks, 2nd edn. Prentice-Hall, Upper Saddle River
Hejda M, Pyšek P, Jarošik V (2009) Impact of invasive plants on the species richness, diversity and composition of invaded communities. J Ecol 97:393–403
Herzschuh U, Birks HJB (2010) Evaluating the indicator value of Tibetan pollen taxa for modern vegetation and climate. Rev Palaeobot Palynol 160:197–208
Hoerl AE, Kennard R (1970) Ridge regression: biased estimation for nonorthogonal problems. Technometrics 12:55–67
Holmqvist BH (2005) Classification of large pollen datasets using neural networks with application to mapping and modelling pollen data. LUNDQUA report 39, Lund University
Horsak M, Chytrý M, Pokryszko BM, Danihelka J, Ermakov N, Hajek M, Hajkova P, Kintrova K, Koci M, Kubesova S, Lustyk P, Otypkova Z, Pelánková B, Valachovic M (2010) Habitats of relict terrestrial snails in southern Siberia: lessons for the reconstruction of palaeoenvironments of full-glacial Europe. J Biogeogr 37:1450–1462
Iverson LR, Prasad AM (1998) Predicting abundance of 80 tree species following climate change in the eastern United States. Ecol Mongr 68:465–485
Iverson LR, Prasad AM (2001) Potential changes in tree species richness and forest community types following climate change. Ecosystems 4:186–199

Iverson LR, Prasad AM, Schwartz MW (1999) Modeling potential future individual tree-species distributions in the eastern United States under a climate change scenario: a case study with *Pinus virginiana*. Ecol Model 115:77–93

Iverson LR, Prasad AM, Matthews SN, Peters M (2008) Estimating potential habitat for 134 eastern US tree species under six climate scenarios. Forest Ecol Manage 254:390–406

Jacob G, Marriott FHC, Robbins PA (1997) Fitting curves to human respiratory data. Appl Stat 46:235–243

Jensen FV, Nielsen TD (2007) Bayesian networks and decision graphs, 2nd edn. Springer, New York

Jeschke JM, Strayer DL (2008) Usefulness of bioclimatic models for studying climate change and invasive species. Ann NY Acad Sci 1134:1–24

Jolliffe IT (2002) Principal component analysis, 2nd edn. Springer, New York

Juggins S, Birks HJB (2012) Chapter 14: Quantitative environmental reconstructions from biological data. In: Birks HJB, Lotter AF, Juggins S, Smol JP (eds) Tracking environmental change using lake sediments. Volume 5: Data handling and numerical techniques. Springer, Dordrecht

Juggins S, Telford RJ (2012) Chapter 5: Exploratory data analysis and data display. In: Birks HJB, Lotter AF, Juggins S, Smol JP (eds) Tracking environmental change using lake sediments. Volume 5: Data handling and numerical techniques. Springer, Dordrecht

Kallimanis AS, Ragia V, Sgardelis SP, Pantis JD (2007) Using regression trees to predict alpha diversity based upon geographical and habitat characteristics. Biodivers Conserv 16:3863–3876

Keith RP, Veblen TT, Schoennagel TL, Sherriff RL (2010) Understory vegetation indicates historic fire regimes in ponderosa pine-dominated ecosystems in the Colorado Front Range. J Veg Sci 21:488–499

Kohonen T (2001) Self-organising maps, 3rd edn. Springer, Berlin

Korb KB, Nicholson AE (2004) Bayesian artificial intelligence. Chapman & Hall, Boca Raton

Kragt ME, Newham LTH, Jakeman AJ (2009) A Bayesian network approach to integrating economic and biophysical modelling. In: Anderssen RS, Braddock RD, Newham LTH (eds) 18th world IMACS congress and MODSIM09 international congress on modelling and simulation. Modelling and Simulation Society of Australia and New Zealand and International Association for Mathematics and Computers in Simulation, Cairns, Australia. pp 2377–2383

Kucera M, Weinelt M, Kiefer T, Pflaumann U, Hayes A, Chen MT, Mix AC, Barrows TT, Cortijo E, Duprat J, Juggins S, Waelbroeck C (2005) Reconstruction of sea-surface temperatures from assemblages of planktonic foraminifera: multi-technique approach based on geographically constrained calibration data sets and its application to glacial Atlantic and Pacific Oceans. Quaternary Sci Rev 24:951–998

Larsen DR, Speckman PL (2004) Multivariate regression trees for analysis of abundance data. Biometrics 60:543–549

Lawler JJ, White D, Neilson RP, Blaustein AR (2006) Predicting climate-induced range shifts: model differences and model reliability. Global Change Biol 12:1568–1584

Leathwick JR, Rowe D, Richardson J, Elith J, Hastie T (2005) Using multivariate adaptive regression splines to predict the distributions of New Zealand's freshwater diadromous fish. Freshw Biol 50:2034–2052

Leathwick JR, Elith J, Hastie T (2006) Comparative performance of generalized additive models and multivariate adaptive regression splines for statistical modelling of species distributions. Ecol Model 199:188–196

Legendre P, Birks HJB (2012a) Chapter 7: Clustering and partitioning. In: Birks HJB, Lotter AF, Juggins S, Smol JP (eds) Tracking environmental change using lake sediments. Volume 5: Data handling and numerical techniqlues. Springer, Dordrecht

Legendre P, Birks HJB (2012a) Chapter 8: From classical to canonical ordination. In: Birks HJB, Lotter AF, Juggins S, Smol JP (eds) Tracking environmental change using lake sediments. Volume 5: Data handling and numerical techniques. Springer, Dordrecht

Lek S, Guégan JF (1999) Artificial neural networks as a tool in ecological modelling, an introduction. Ecol Model 120:65–73

Lek S, Guégan J-F (2000) Artificial neuronal networks: application to ecology and evolution. Springer, Berlin
Lek S, Delacoste M, Baran P, Dimopoulos I, Lauga J, Aulagnier S (1996a) Application of neural networks to modelling nonlinear relationships in ecology. Ecol Model 90:39–52
Lek S, Dimopoulos I, Fabre A (1996b) Predicting phosphorus concentration and phosphorus load from watershed characteristics using backpropagation neural networks. Acta Oecol 17:43–53
Lindblah M, O'Connor R, Jacobson GL Jr (2002) Morphometric analysis of pollen grains for palaeoecological studies: classification of *Picea* from eastern North America. Am J Bot 89:1459–1467
Lindblah M, Jacobson GL Jr, Schauffler M (2003) The postglacial history of three *Picea* species in New England, USA. Quaternary Res 59:61–69
Lindström J, Kokko H, Ranta E, Lindén H (1998) Predicting population fluctuations with artificial neural networks. Wildl Biol 4:47–53
Lotter AF, Anderson NJ (2012) Chapter 18: Limnological responses to environmental changes at inter-annual to decadal time-scales. In: Birks HJB, Lotter AF, Juggins S, Smol JP (eds) Tracking environmental change using lake sediments. Volume 5: Data handling and numerical techniques. Springer, Dordrecht
Malmgren BA, Nordlund U (1997) Application of artificial neural networks to paleoceanographic data. Palaeogeogr Palaeoclim Palaeoecol 136:359–373
Malmgren BA, Winter A (1999) Climate zonation in Puerto Rico based on principal component analysis and an artificial neural network. J Climate 12:977–985
Malmgren BA, Kucera M, Nyberg J, Waelbroeck C (2001) Comparison of statistical and artificial neural network techniques for estimating past sea surface temperatures from planktonic foraminfer census data. Paleoceanography 16:520–530
Manel S, Dias JM, Buckton ST, Ormerord SJ (1999a) Alternative methods for predicting species distribution: an illustration with Himalayan river birds. J Appl Ecol 36:734–747
Manel S, Dias JM, Ormerord SJ (1999b) Comparing discriminant analysis, neural networks and logistic regression for predicting species distributions: a case study with a Himalayan river bird. Ecol Model 120:337–347
Marcot BG, Holthausen RS, Raphael MG, Rowland MG, Wisdom MJ (2001) Using Bayesian belief networks to evaluate fish and wildlife population viability under land management alternatives from an environmental impact statement. Forest Ecol Manage 153:29–42
Martens H, Næes T (1989) Multivariate calibration. Wiley, Chichester
Maslow AH (1966) The psychology of science: a reconnaissance. Harper & Row, New York
Melssen W, Wehrens R, Buydens L (2006) Supervised Kohonen networks for classification problems. Chemometr Intell Lab Syst 83:99–113
Melssen W, Bulent U, Buydens L (2007) SOMPLS: a supervised self-organising map-partial least squares algorithm for multivariate regression problems. Chemometr Intell Lab Syst 86:102–120
Michaelson J, Schimel DS, Friedl MA, Davis FW, Dubayah RC (1994) Regression tree analysis of satellite and terrain data to guide vegetation sampling and surveys. J Veg Sci 5:673–686
Milbarrow S (2011) earth. R package version 3.2-0. http://cran.r-project.org/packages=earth
Miller AJ (2002) Subset selection in regression, 2nd edn. Chapman & Hall/CRC, Boca Raton
Miller J, Franklin J (2002) Modeling the distribution of four vegetation alliances using generalized linear models and classification trees with spatial dependence. Ecol Model 157:227–247
Moisen GG, Frescino TS (2002) Comparing five modelling techniques for predicting forest characteristics. Ecol Model 157:209–225
Morgan JN, Sonquist JA (1963) Problems in the analysis of survey data, and a proposal. J Am Stat Assoc 58:415–434
Mundry R, Nunn CL (2009) Stepwise model fitting and statistical inference: turning noise into signal pollution. Am Nat 173:119–123
Murphy B, Jansen C, Murray J, de Barro P (2010) Risk analysis on the Australian release of *Aedes aegypti* (L.) (Diptera: Culicidae) containing *Wolbachia*. CSIRO, Canberra
Murtaugh PA (2009) Performance of several variable-selection methods applied to real ecological data. Ecol Lett 12:1061–1068

Nakagawa S, Freckleton RP (2008) Missing inaction: the danger of ignoring missing data. Trends Ecol Evol 23:592–596

Newton AC, Marshall E, Schreckenberg K, Golicher D, te Velde DW, Edouard F, Arancibia E (2006) Use of a Bayesian belief network to predict the impacts of commercializing non-timber forest products on livelihoods. Ecol Soc 11:24

Newton AC, Stewart GB, Diaz A, Golicher D, Pullin AS (2007) Bayesian belief networks as a tool for evidence-based conservation management. J Nat Conserv 15:144–160

Nyberg H, Malmgren BA, Kuijpers A, Winter A (2002) A centennial-scale variability of tropical North Atlantic surface hydrology during the late Holocene. Palaeogeogr Palaeoclim Palaeoecol 183:25–41

Næs T, Kvaal K, Isaksson T, Miller C (1993) Artificial neural networks in multivariate calibration. J NIR Spectrosc 1:1–11

Næs T, Isaksson T, Fearn T, Davies T (2002) A user-friendly guide to multivariate calibration and classification. NIR Publications, Chichester

Olden JD (2000) An artificial neural network approach for studying phytoplankton succession. Hydrobiologia 436:131–143

Olden JD, Jackson DA (2002) Illuminating the 'black box': a randomization approach for understanding variable contributions in artificial neural networks. Ecol Model 154:135–150

Olden JD, Joy MK, Death RG (2004) An accurate comparison on methods for quantifying variable importance in artificial neural networks using simulated data. Ecol Model 178:389–397

Olden JD, Lawler JJ, Poff NL (2008) Machine learning methods without tears: a paper for ecologists. Quaternary Rev Biol 83:171–193

Ôzesmi SL, Tan CO, Özesmi U (2006) Methodological issues in building, training, and testing artificial neural networks in ecological applications. Ecol Model 195:83–93

Pakeman RJ, Torvell L (2008) Identifying suitable restoration sites for a scarce subarctic willow (*Salix arbuscula*) using different information sources and methods. Plant Ecol Divers 1:105–114

Park MY, Hastie T (2007) *l*1-Regularization path algorithm for generalised linear models. J R Stat Soc Ser B 69:659–677

Pearson RG, Thuiller W, Araújo MB, Martinez-Meyer E, Brotons L, McClean C, Miles L, Segurado P, DawsonTP Lees DC (2006) Model-based uncertainty in species range prediction. J Biogeogr 33:1704–1711

Pelánková B, Kuneš P, Chytrý M, Jankovská V, Ermakov N, Svobodová-Svitavaská H (2008) The relationships of modern pollen spectra to vegetation and climate along a steppe-forest-tundra transition in southern Siberia, explored by decision trees. Holocene 18:1259–1271

Peters J, De Baets B, Verhoest NEC, Samson R, Degroeve S, de Becker P, Huybrechts W (2007) Random forests as a tool for predictive ecohydrological modelling. Ecol Model 207:304–318

Peyron O, Guiot J, Cheddadi R, Tarasov P, Reille M, de Beaulieu J-L, Bottema S, Andrieu V (1998) Climatic reconstruction of Europe for 18,000 yr BP from pollen data. Quaternary Res 49:183–196

Peyron O, Jolly D, Bonnefille R, Vincens A, Guiot J (2000) Climate of East Africa 6000 ^{14}C yr BP as inferred from pollen data. Quaternary Res 54:90–101

Peyron O, Bégeot C, Brewer S, Heiri O, Magny M, Millet L, Ruffaldi P, van Campo E, Yu G (2005) Lateglacial climatic changes in eastern France (Lake Lautrey) from pollen, lake-levels, and chironomids. Quaternary Res 64:197–211

Ploner A, Brandenburg C (2003) Modelling visitor attendance levels subject to day of the week and weather: a comparison between linear regression models and regression trees. J Nat Conserv 11:297–308

Pourret O, Naïm P, Marcot B (eds) (2008) Bayesian networks. A practical guide to applications. Wiley, Chichester

Prasad AM, Iverson LR, Liaw A (2006) Newer classification and regression tree techniques: bagging and random forests for ecological prediction. Ecosystems 9:181–199

Pysek P, Bacher S, Chytrý M, Jarosik V, Wild J, Celesti-Grapow L, Gassó N, Kenis M, Lambdon PW, Nentwig W, Pergl J, Roques A, Sádlo J, Solarz W, Vilà M, Hiulme PE (2010) Contrasting patterns in the invasions of European terrestrial and freshwater habitats by alien plants, insects and vertebrates. Global Ecol Biogeogr 19:317–331

Quinlan J (1993) C4.5: programs for machine learning. Morgan Kaufman, San Mateo

R Development Core Team (2011) R: a language and environment for statistical computing. R foundation for statistical computing. Vienna, Austria. http://www.r-project.org

Racca JMJ, Philibert A, Racca R, Prairie YT (2001) A comparison between diatom-pH-inference models using artificial neural networks (ANN), weighted averaging (WA) and weighted averaging partial least square (WA-PLS) regressions. J Paleolimnol 26:411–422

Racca JMJ, Wild M, Birks HJB, Prairie YT (2003) Separating wheat from chaff: diatom taxon selection using an artificial neural network pruning algorithm. J Paleolimnol 29:123–133

Racca JMJ, Gregory-Eaves I, Pienitz R, Prairie YT (2004) Tailoring palaeolimnological diatom-based transfer functions. Can J Fish Aquat Sci 61:2440–2454

Ramakrishnan N, Grama A (2001) Mining scientific data. Adv Comput 55:119–169

Raymond B, Watts DJ, Burton H, Bonnice J (2005) Data mining and scientific data. Arct Antarct Alp Res 37:348–357

Recknagel F, French M, Harkonen P, Yabunaka K-I (1997) Artificial neural network approach for modelling and prediction of algal blooms. Ecol Model 96:11–28

Rehfeldt GE, Crookston NL, Warwell MV, Evans JS (2006) Empirical analyses of plant-climate relationships for the western United States. Int J Plant Sci 167:1123–1150

Rejwan C, Collins NC, Brunner LJ, Shuter BJ, Ridgway MS (1999) Tree regression analysis on the nesting habitat of smallmouth bass. Ecology 80:341–348

Ridgeway G (2007) Generalized boosted models: a guide to the gbm package. http://cran.r-project.org/web/packages/gbm/vignettes/gbm.pdf. Accessed 20 July 2011

Ridgeway G (2010) gbm. R package version 1.6-3.1. http://cran.r-project.org/web/packages/gbm/

Rieman B, Peterson JT, Clayton J, Howell P, Thurow R, Thompson W, Lee D (2001) Evaluation of potential effects of federal land management alternatives on trends of salmonids and their habitats in the interior Columbia River basin. Forest Ecol Manage 153:43–62

Ripley BD (2008) Pattern recognition and neural networks. Cambridge University Press, Cambridge

Roberts DR, Hamann A (2012) Predicting potential climate change impacts with bioclimate envelope models: a palaeoecological perspective. Global Ecol Biogeogr 21:121–133

Rose NL (2001) Fly-ash particles. In: Last WM, Smol JP (eds) Tracking environmental change using lake sediments. Volume 2: Physical and geochemical methods. Kluwer Academic Publishers, Dordrecht, pp 319–349

Rose NL, Juggins S, Watt J, Battarbee RW (1994) Fuel-type characterization of spheroidal carbonaceous particles using surface chemistry. Ambio 23:296–299

Schapire RE (1990) The strength of weak learnability. Mach Learn 5:197–227

Scull P, Franklin J, Chadwick OA (2005) The application of classification tree analysis to soil type prediction in a desert landscape. Ecol Model 181:1–15

Simpson GL (2012) Chapter 15: Modern analogue techniques. In: Birks HJB, Lotter AF, Juggins S, Smol JP (eds) Tracking environmental change using lake sediments. Volume 5: Data handling and numerical techniques. Springer, Dordrecht

Spadavecchia L, Williams M, Bell R, Stoy PC, Huntley B, van Wijk MT (2008) Topographic controls on the leaf area index and plant functional type of a tundra ecosystem. J Ecol 96:1238–1251

Spitz F, Lek S (1999) Environmental impact prediction using neural network modelling. An example in wildlife damage. J Appl Ecol 36:317–326

Steiner D, Pauling A, Nussbaumer SU, Nesje A, Luterbacher J, Wanner H, Zumbühl HJ (2008) Sensitivity of European glaciers to precipitation and temperature – two case studies. Clim Change 90:413–441

Stewart-Koster B, Bunn SE, Mackay SJ, Poff NL, Naiman RJ, Lake PS (2010) The use of Bayesian networks to guide investments in flow and catchment restoration for impaired river ecosystems. Freshw Biol 55:243–260

Stockwell DRB, Noble IR (1992) Induction of sets of rules from animal distribution data: a robust and informative method of data analysis. Math Comput Sims 33:385–390

Stockwell DRB, Peters D (1999) The GARP modelling system: problems and solutions to automated spatial prediction. Int J Geogr Info Sci 13:143–158

Stockwell DRB, Peterson AT (2002) Effects of sample size on accuracy of species distribution models. Ecol Model 148:1–13

Tarasov P, Peyron O, Guiot J, Brewer S, Volkova VS, Bezusko LG, Dorofeyuk NI, Kvavadze EV, Osipova IM, Panova NK (1999a) Late glacial maximum climate of the former Soviet Union and Mongolia reconstructed from pollen and plant macrofossil data. Clim Dyn 15:227–240

Tarasov P, Guiot J, Cheddadi R, Andreev AA, Bezusko LG, Blyakharchuk TA, Dorofeyuk NI, Filimonova LV, Volkova VS, Zernitskayo VP (1999b) Climate in northern Eurasia 6000 years ago reconstructed from pollen data. Earth Planet Sci Lett 171:635–645

Telford RJ, Birks HJB (2009) Design and evaluation of transfer functions in spatially structured environments. Quaternary Sci Rev 28:1309–1316

ter Braak CJF (2009) Regression by L_1 regularization of smart contrasts and sums (ROSCAS) beats PLS and elastic net in latent variable model. J Chemometr 23:217–228

Therneau TM, Atkinson B [R port by Ripley B] (2011) rpart: recursive partitioning. R package version 3.1-50. http://cran.r-project.org/package/rpart

Thuiller W, Araújo MB, Lavorel S (2003) Generalized models vs, classification tree analysis: predicting spatial distributions of plant species at different scales. J Veg Sci 14:669–680

Tibshirani R (1996) Regression shrinkage and selection via the lasso. J R Stat Soc Ser B 58:267–288

Ticehurst JL, Curtis A, Merritt WS (2011) Using Bayesian networks to complement conventional analyses to explore landholder management of native vegetation. Environ Model Softw 26:52–65

Tsaor A, Allouche O, Steinitz O, Rotem D, Kadmon R (2007) A comparative evaluation of presence-only methods for modelling species distribution. Divers Distrib 13:397–405

van Dijk ADJ, ter Braak CJF, Immink RG, Angenent GC, van Ham RCHJ (2008) Predicting and understanding transcription factor interactions based on sequence level determinants of combinatorial control. Bioinformatics 24:26–33

Vayssieres MP, Plant RE, Allen-Diaz BH (2000) Classification trees: an alternative non-parametric approach for predicting species distributions. J Veg Sci 11:679–694

Vincenzi S, Zucchetta M, Franzoi P, Pellizzato M, Pranovi F, de Leo GA, Torricelli P (2011) Application of a random forest algorithm to predict spatial distribution of the potential yield of *Ruditapes philippinarum* in the Venice lagoon, Italy. Ecol Model 222:1471–1478

Warner B, Misra M (1996) Understanding neural networks as statistical tools. Am Stat 50:284–293

Wehrens R (2011) Chemometrics with R: multivariate analysis in the natural sciences and life sciences. Springer, New York

Wehrens R, Buydens LMC (2007) Self- and super-organising maps in R: the Kohonen package. J Stat Softw 21:1–19

Weller AF, Harris AJ, Ware JA (2006) Artificial neural networks as potential classification tools for dinoflagellate cyst images: a case using the self-organizing map clustering algorithm. Rev Palaeobot Palynol 141:287–302

Whittingham MJ, Stephens PA, Bradbury RB, Freckleton RP (2006) Why do we still use step-wise modelling in ecology and behaviour? J Anim Ecol 75:1182–1189

Williams JN, Seo C, Thorne J, Nelson JK, Erwin S, O'Brien JM, Schwartz MW (2009) Using species distribution models to predict new occurrences for rare plants. Divers Distrib 15:565–576

Witten IH, Frank E (2005) Data mining: practical machine learning tools and techniques. Morgan Kaufmann/Elsevier, Amsterdam

Zou H, Hastie T (2005) Regularization and variable selection via the elastic net. J R Stat Soc Ser B 67:301–320