Ecology, 88(1), 2007, pp. 243–251

BOOSTED TREES FOR ECOLOGICAL MODELING AND PREDICTION

GLENN DE'ATH[1]

Australian Institute of Marine Science, PMB 3, Townsville Mail Centre, Qld 4811, Australia

Abstract. Accurate prediction and explanation are fundamental objectives of statistical analysis, yet they seldom coincide. Boosted trees are a statistical learning method that attains both of these objectives for regression and classification analyses. They can deal with many types of response variables (numeric, categorical, and censored), loss functions (Gaussian, binomial, Poisson, and robust), and predictors (numeric, categorical). Interactions between predictors can also be quantified and visualized. The theory underpinning boosted trees is presented, together with interpretive techniques. A new form of boosted trees, namely, "aggregated boosted trees" (ABT), is proposed and, in a simulation study, is shown to reduce prediction error relative to boosted trees. A regression data set is analyzed using ABT to illustrate the technique and to compare it with other methods, including boosted trees, bagged trees, random forests, and generalized additive models. A software package for ABT analysis using the R software environment is included in the Appendices together with worked examples.

Key words: aggregated boosted trees; bagging; boosting; classification; cross-validation; prediction; regression.

INTRODUCTION

Classification and regression trees (Breiman et al. 1984) have been widely used for the exploration, description, and prediction of ecological data (De'ath and Fabricius 2000, Vayssières et al. 2000, De'ath 2002). Trees have many desirable properties, including (1) their ability to handle various types of response including numeric, categorical, censored, multivariate, and dissimilarity matrices; (2) trees are invariant to monotonic transformations of the predictors; (3) complex interactions are modeled simply; and (4) missing values in the predictors are managed with minimal loss of information. However trees have two weaknesses, namely (1) they are poor predictors and (2) large trees can be difficult to interpret. These weaknesses can be overcome through the use of boosted trees; now widely acknowledged as excellent predictors that also render simple graphical and numerical interpretations of complex relationships. Hence, boosted trees are now beginning to appear in ecological studies (Cappo et al. 2005, Leathwick et al. 2006).

Prediction and explanation are fundamental objectives of statistical analysis, yet they seldom coincide. For example, some modern statistical regression methods generate accurate predictions that may prove useful, but they cannot determine relationships between the response and the predictors, and thus they fail to explain the underlying processes. Conversely, traditional statistical models such as linear regression are routinely used to explain data relationships, and they do so in simple ways that are easy to interpret. However, they are often relatively poor predictors, especially if the models are selected on criteria not related to prediction error (accuracy) such as hypothesis tests. Although prediction and explanation seldom coincide, the gap between them is narrowing. For example, prediction error, or measures related to it (e.g., Akaike Information Criterion [AIC], Bayesian Information Criterion [BIC], Network Information Criterion [NIC]), are now routinely used as methods of statistical model selection (Akaike 1974, Schwarz 1978, Breiman et al. 1984, Burnham and Anderson 1998, Ripley 2004). This shift recognizes the deficiencies in hypothesis tests when used for model selection (Draper 1995, Burnham and Anderson 1998, Johnson 1999).

Accurate prediction is now often seen as an important objective of ecological and environmental analyses, and our capacity to predict complex systems has greatly improved through the development of methods based on learning algorithms and intense computation. In recent years, major advances have been made as this has become part of mainstream statistics (Hastie et al. 2001). Examples of such methods include neural networks, support vector machines, bagging, random forests, and boosting.

This work introduces boosted trees to ecologists, both theoretically and also as a practical tool for exploration, analysis, and prediction. First, the concepts of prediction error and its estimation are introduced. Two techniques somewhat similar to boosting, namely bagging and random forests, are then outlined. The core of the paper follows, comprising a detailed presentation of boosted trees, and a refinement that I propose to call aggregated boosted trees. Analyses of an ecological data set follows and includes comparisons of boosted trees with other predictive methods.

Manuscript received 5 May 2006; revised 3 July 2006; accepted 5 July 2006. Corresponding Editor: A. M. Ellison.

[1] E-mail: g.death@aims.gov.au

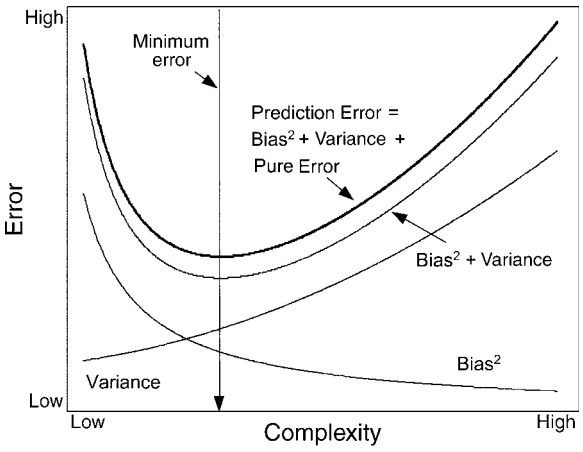

FIG. 1. Illustration of the trade-off between model complexity, bias, and variance. As model complexity increases, bias decreases but variance increases. Prediction error initially improves as complexity increases but typically reaches a minimum before increasing.

PREDICTION

The prediction error (PE; accuracy) of a statistical model is a measure of how close model predictions are to their true values on average. Suppose we have a model $y = f(x) + \varepsilon$; $E(\varepsilon) = 0$; $\mathrm{Var}(\varepsilon) = \sigma^2$ and predictions $\hat{f}(x)$. Then PE is defined by

$$\begin{aligned}\mathrm{PE} &= E\left\{\left[y - \hat{f}(x)\right]^2\right\} \\ &= \left[E\hat{f}(x) - f(x)\right]^2 + E\left[\hat{f}(x) - E\hat{f}(x)\right]^2 + \sigma^2 \\ &= \mathrm{Bias}^2 + \mathrm{Variance} + \mathrm{Pure\ Error}.\end{aligned}$$

Increasing model complexity (e.g., by adding more parameters) decreases bias but increases variance, and vice versa (Fig. 1). This is known as the bias–variance trade-off. In addition to being a measure of model performance, the PE of models is either used directly (based on test data or cross-validation) (Breiman et al. 1984, Ripley 2004) or indirectly (e.g., using information criteria such as AIC and BIC) as a method of model selection.

To obtain accurate estimates of PE, predictions should be made on new data (test data), not the data used to fit the model (training data), since the latter typically gives overoptimistic estimates of accuracy (Fig. 2a). If large data sets are available and many models are being compared, then dividing the data into training, validation, and test sets is ideal. The training set is used to fit the models, the validation data to fine tune them and select the best one, and the test set to estimate its PE. If model comparisons are not required then training and test sets suffice. If data sets are relatively small ($n < 1000$) then the small size of the training and test data sets degrades model PE. For such data sets, estimates of PE can be based on cross-validation, thereby making effective use of all the data.

The relative contribution of bias and variance to PE varies across different types and complexity of statistical models (Fig. 1), and hence many methods are used to reduce PE. For example, model averaging, shrinkage, and bagging are three commonly used techniques that focus on the bias/variance balance in different ways. Model averaging involves the fitting of several models, and then averaging of parameter estimates or predictions across the models (Ripley 2004). The averaging is usually weighted by some measure of predictive accuracy; e.g., information criteria. Model averaging mainly

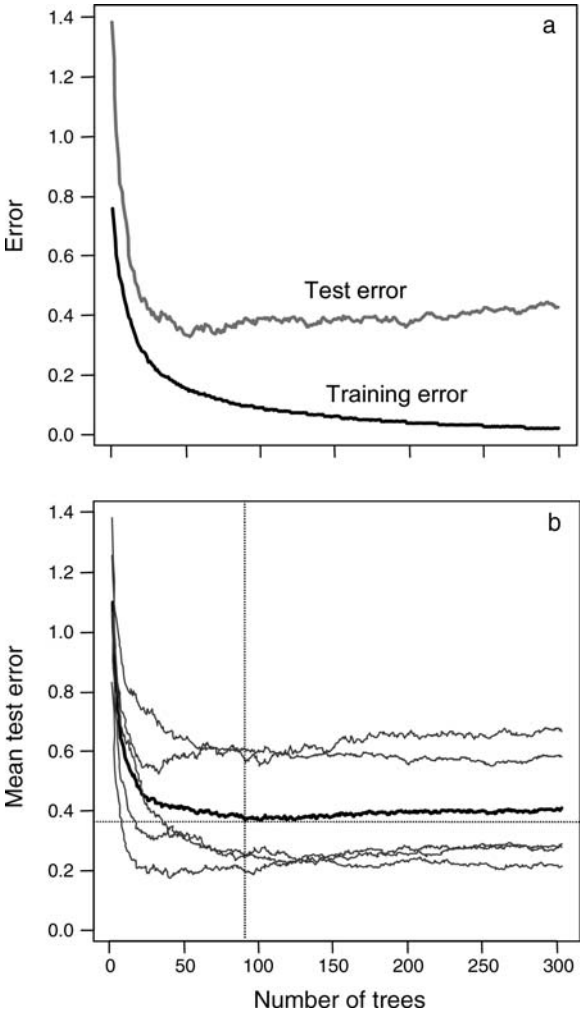

FIG. 2. (a) The relationship between training and test error and the number of trees (a measure of complexity) used for the boosting. Training error can be reduced to very low levels and will not usually increase. Test error will typically reach a minimum before flattening or increasing slightly. (b) The test error as a function of the number of trees for fivefold aggregated boosted trees. The five gray profiles are the test error profiles for the five component boosted trees and are quite variable with evidence of over-learning. Their minima occur between 35 and 262 trees. The averaged profile (black line) is more stable and shows little evidence of over-learning, with a flat section between 80 and 150 trees illustrating the greater stability of aggregated boosted trees.

reduces variance. Shrinkage methods, such as ridge regression and the lasso (Hastie et al. 2001), shrink the parameter estimates toward zero, and thus reduce variance while increasing bias to a lesser degree. Bagging, typically applied to high-variance models that are nonlinear, reduces variance. It works most effectively for high-variance predictors with low bias, and is ineffective for linear models (Breiman 1996).

Bagged Trees and Random Forests

Large classification and regression trees have high variance and low bias and are therefore well suited to bagging. Bagging is simple: (1) take a bootstrap sample from the data set, (2) fit the tree to this data set, (3) repeat steps 1 and 2 a large number of times (typically 50–1000), and (4) make predictions for new data using each of the fitted models and average the predictions. Methods have been developed to visualize and quantify results through dependency plots and measures of influence of the predictors (see *Quantifying and visualizing results*).

Random forests (Breiman 2001) are a modified version of bagged trees. For each tree of the collection, a random subset of predictors is chosen to determine each split. The number of randomly selected predictors is fixed; typically $\sqrt{p}$ or $\log(p)$, where p is the number of predictors. By injecting randomness in this way, correlations between predictions of the individual trees are reduced, and this in turn reduces the variance component of PE. Additionally, by reducing the number of predictors used at each split considerable computational savings are made.

Boosted Trees

The development of boosting

Boosting originated in the machine learning community with the introduction of AdaBoost (Freund and Schapire 1996), an algorithm that classifies binary responses. The basic idea is to combine the predictions from a collection of weak classifiers (high PE) in such a way that the averaged predictions form a strong classifier (low PE). One such weak classifier often used in AdaBoost is a single-split classification tree. AdaBoost grows a sequence of trees, with successive trees grown on reweighted versions of the data. At each stage of the sequence, each data case is classified from the current sequence of trees, and these classifications are used as weights for fitting the next tree of the sequence. Incorrectly classified cases receive more weight than those that are correctly classified, and thus cases that are difficult to classify receive ever-increasing weight, thereby increasing their chance of being correctly classified. The final classification of each case is determined by the weighted majority of classifications across the sequence of trees. It was not clear how AdaBoost worked until boosting was examined from a statistical perspective (Friedman et al. 2000, Hastie et al. 2001). This leads to a series of theoretical and practical advances in the understanding of boosting, and has realized its potential as a general method of function approximation based on additive models. Although boosting is not restricted to trees, boosted trees have proved to be highly accurate predictors, and this work focuses on this form of boosting.

Boosting as an additive model

Suppose we have a response y and predictors x, and we wish to approximate y by the function $f(x)$. Typically, we specify the functional form of $f(x)$ together with a loss function $L(y, f(x))$, and then minimize L to estimate $f(x)$. The most familiar form of f is the linear model with $f(x) = x\beta$ where β is a matrix of parameters, and the loss function is squared error, i.e., $L(y, f(x)) = (y - f(x))^2$. However, other loss function are used; e.g., for robust linear models absolute loss could be used, i.e., $L = |y - f(x)|$. Additive models (Hastie and Tibshirani 1990, Hastie et al. 2001) express $f(x)$ as a sum of basis functions $b(x; \gamma_m)$ as follows:

$$f(x) = \sum_m f_m(x) = \sum_m \beta_m b(x; \gamma_m).$$

This form includes models such as neural networks, generalized linear models, multivariate adaptive regression splines, wavelets, and classification and regression trees (Hastie et al. 2001). The basis functions b vary with the type of method, and include polynomials, smoothers, and regression trees, and the parameters (β_m and γ_m) are estimated by minimizing a specified loss function over the data. For boosted trees, the functions $b(x; \gamma_m)$ represent the individual trees, with γ_m defining the split variables, their values at each node, and the predicted values (Friedman 2001). The β_m represent weights given to the nodes of each tree in the collection and determine how predictions from the individual trees are combined.

Forward stagewise estimation

Estimation of the parameters of additive models depends on the functional form of f and can be difficult. Forward stagewise fitting simplifies the problem by estimating β_m and γ_m sequentially from $m = 1$ to n. The procedure can be summarized as follows:

1) Initialize $f_0(x) = 0$.
2) For $m = 1$ to n:
   a) Get estimates β_m and γ_m by minimizing $L(y, f_{m-1}(x) + \beta b(x; \gamma))$.
   b) Update $f_m(x) = f_{m-1}(x) + \beta_m b(x; \gamma_m)$.
3) Calculate $f(x) = \Sigma_m f_m(x)$.

For squared-error loss the procedure greatly simplifies since

$$\begin{aligned} L(y, f(x)) &= [y - f(x)]^2 \\ L(y, f_{m-1}(x) + \beta b(x; \gamma)) &= [y - f_{m-1}(x) - \beta b(x; \gamma)]^2 \\ &= [r - \beta b(x; \gamma)]^2 \end{aligned}$$

where r are the usual least-squares residuals.

Thus, for boosted trees with squared-error loss, we simply fit a least-squares regression tree to the residuals of the previous iteration, and the values of β are the predicted values of the terminal nodes. Thus, the minimization and update (2a and 2b above) for boosted regression trees based on least-squares loss is relatively simple. For AdaBoost, using classification trees and exponential loss, the minimization of the loss function also simplifies algebraically, again leading to a relatively simple minimization and update (Friedman et al. 2000).

Gradient boosting

Squared-error loss and exponential loss can be efficient for regression and classification, but are not always so; e.g., they perform poorly for non-robust data or censored data. Thus other loss functions are required (Friedman 2001, Hastie et al. 2001), and for all such losses the minimization step (2a above) is difficult. To overcome this, Friedman (2001) devised gradient boosting, an approximation technique that applies the method of steepest descent to forward stagewise estimation. This involves a two-step approximation of the loss function; the first step estimates γ_m using a least squares regression tree, and the second estimates β_m. Step 2a of forward stagewise estimation (see above) is replaced by three new steps and the algorithm for gradient boosting becomes:

1) Initialize $f_0(x) = 0$.
2) For $m = 1$ to n:
   a) Calculate the residuals, $r = -([\partial L(y, f(x))]/[\partial f(x)])_{f(x)=f_{m-1}(x)}$.
   b) Fit a least-squares regression tree to r to get the estimate γ_m of $\beta b(x; \gamma)$.
   c) Get the estimate β_m by minimizing $L(y, f_{m-1}(x) + \beta b(x; \gamma_m))$.
   d) Update $f_m(x) = f_{m-1}(x) + \beta_m b(x; \gamma_m)$.
3) Calculate $f(x) = \Sigma_m f_m(x)$.

The new step 2a calculates the residuals as the negative of the first derivative of the loss function evaluated for the current value of $f(x)$. For squared-error loss this is the usual residuals, and for least absolute deviations it is the sign of the residuals. Step 2b always uses a least-squares regression tree to estimate γ_m. Least-squares trees are used irrespective of the chosen loss function, and are computationally very efficient. Step 2c then estimates the values β_m assigned to the nodes of the tree to minimize the overall loss; this is called the "line-search."

Reducing the learning rate

Boosting is subject to over-learning with PE reaching a minimum before increasing (Fig. 2a). This over-learning can be controlled by slowing the learning process by modifying 2d above as follows:

$$f_m(x) = f_{m-1}(x) + \varepsilon\beta_m b(x; \gamma_m) \qquad 0 < \varepsilon \leq 1.$$

Smaller values of ε (typically 0.1–0.001) result in slower learning that needs to be compensated for by more iterations (trees) in the boosting sequence. A 10-fold reduction in learning rate requires an approximately 10-fold increase in iterations, but also makes it is easier to identify the number of iterations that minimizes loss. Slowing the learning in this way is a form of shrinkage that penalizes the sum of the absolute size of the estimates of β_m; a technique known as the lasso (Hastie et al. 2001). The shrunken estimates are biased and thus boosted trees trade-off a small increase in bias for a larger reduction in variance, thereby improving PE. The effectiveness of slow learning for stagewise boosting is now better understood through a new method of model selection known as least angle regression (Efron et al. 2004).

Stochastic gradient boosting

The performance of gradient boosting is also improved by injecting randomness into the sequential fitting (Friedman 2002). This involves taking subsamples of the training data (typically 40–60%) for each iteration. It has three important benefits:

1) PE improves substantially.
2) Computation is reduced; e.g., half-samples (i.e., taking 50% of the training data without replacement) reduce the computational demand by almost 50%.
3) Over-learning is further reduced, thus further aiding identification of the optimum number of trees (Fig. 2a). Note however that over-learning is not completely eliminated, particularly for regression models.

These benefits have been widely accepted, and reference to "stochastic gradient boosting" is often simplified to "gradient boosting" or more simply "boosted trees." Heuristics other than trees can also be boosted, and hence "boosting" should be used to refer the general case rather than boosted trees specifically. Henceforth in this paper, the term "boosted trees" (BT) will be used to denote stochastic gradient boosting using least squares regression trees.

Growing boosted trees

Selecting the meta-parameters.—The first step in fitting a boosted tree is to select the response, the predictors, and the loss function. The latter is determined by the type of response (e.g., numeric or categorical) and the expected variation in the response. Next, the number of trees to be grown in the sequence, the shrinkage rate, the size of the individual trees, and the fraction of the training data sampled are chosen. These are meta-parameters, i.e., they are fixed in advance by the user, as opposed to parameters estimated in the fitting process. Shrinking rates of 0.1 to 0.001 are typically used and smaller values give lower PE but require proportionally more computation. The fraction of training data sampled is typically set in the range (0.4–0.6) and is seldom varied.

Selecting the number of trees.—Although stochastic boosting greatly reduces over-learning, we still need to

determine the number of trees of the BT that minimizes PE, and two methods can be used depending on the amount of data available. For large data sets, we divide the data into training and test data sets, and then grow the BT (using m trees say) on the training data. Next, we estimate PE from the test data for BTs comprising 1 to m trees (Fig. 2a), then select the number of trees, m^*, corresponding to the minimum PE. Finally we reduce the number of trees in the BT to m^*. For small data sets however, the reduction in size of the training data relative to the full data degrades estimates of PE (Hastie et al. 2001), and thus the second approach is to determine the number of trees using cross-validation (Bühlmann and Yu 2003). The procedure is as follows:

1) Divide the data into N (typically 5–10) subsets and construct N training data sets each of which omits one of the N subsets (the "out-of-bag" data).

2) Grow N BTs; one for each of the N training sets.

3) Calculate the PE for each BT for tree sizes 1 to m from the corresponding out-of-bag data and pool across the N boosted trees (Fig. 2b). The minimum PE estimates the optimum number of trees m^* for the BT.

4) Grow a BT of m^* trees from the whole data set.

This BT can then be used to assess relationships between the response and predictors and to make predictions based on new data.

Aggregated Boosted Trees

In this section, I propose a new variety of BTs called "aggregated boosted trees" (ABTs) that comprise a collection of BTs (component BTs) each of which is grown on a cross-validation subset of the data. As is the case with bagged trees, the PE of ABTs can be assessed from the out-of-bag data, and ABT predictions for new data are made by first predicting using each of the component trees and then aggregating the predictions (e.g., by averaging).

To grow ABTs we simply follow steps 1–3 for selecting the number of trees using cross-validation. Then, instead of growing a single BT to the optimum number of trees m^* on the whole data set, we simply reduce the number of trees for each BT to m^*. This collection of BTs forms the ABT. The estimated PE of the single BT obtained by cross-validation can thus also be used to estimate PE for the ABT (Appendix A). ABTs thus require minimal additional computation beyond estimation of m^*.

ABTs with m^* trees per component consistently predict more accurately than a single BT comprising m^* trees. This property holds over variable size data sets of differing complexity, and results from reduced variance due to the averaging of predictions over the component BTs. Simulations (Appendix A) showed that for regression, ABTs were more accurate than BT for 94% of data simulations, with a mean improvement in PE (of true values) of 9.9% due to a substantial improvement in precision (12.0%) and small improvement in bias (3.5%). For classification, ABTs and BTs were indistinguishable in PE. However the misclassification error of ABT was marginally, but consistently, better than BT. Typically, the performance of BT approaches that of ABT when the number of trees used in a BT is N times that used in an ABT (Appendix A). However this can be expensive because the size of the BT is also based on N-fold CV, so ABT not only performs better on average than BT, but is also computationally more efficient.

Interpreting and Selecting Boosted Trees

Quantifying and visualizing results

As noted earlier, not all predictive methods enable us to explore the relationships between the response and predictors. For methods based on trees however, we can look "inside the box" to quantify and visualize such relationships, and this in turn can lead to the identification of simpler models; e.g., including only main effects, or imposing monotonic constraints. These techniques have largely been developed for boosted trees (Hastie et al. 2001), though some, but not all, can also be applied to bagged and single trees, and random forests.

Interactions between predictors

Varying the numbers of splits (size) of the individual trees can be used to determine the degree to which predictors interact in determining the response. For trees comprising a single split, the estimated response depends only on main effects. Trees with two splits include first-order interactions, trees with three splits include up to second-order interactions, and so on. Thus comparison of PE based on different-sized trees can determine the level of interactions between the predictors. For example, a large increase in the PE from trees of size 2 to size 3, but then relatively constant PE for larger trees would indicate strong first-order interactions but no higher-order interactions of importance.

Although this method can determine the level of interactions, it does not identify which predictors are involved. To do this we need to quantify the partial dependencies between the response and subsets of different predictors. For two predictors, $x1$ and $x2$, their dependency can be quantified as follows (Friedman 2001). Using the chosen BT, predict the response based on $x1$ and $x2$ together (denote the predictions by $p12$ and so on), $x1$ alone ($p1$), and $x2$ alone ($p2$). Regress $p12$ on $p1$ and $p2$ and calculate the coefficient of determination R^2. The expression $\sqrt{1-R^2}$ gives a measure of the interaction of $x1$ and $x2$, with larger values indicating stronger dependencies. This method of assessing dependencies can be extended to any two subsets of the predictors.

Relative variable influence

For single regression trees the relative influence (importance) of a predictor can be quantified by the sum of squared improvements at all splits determined by

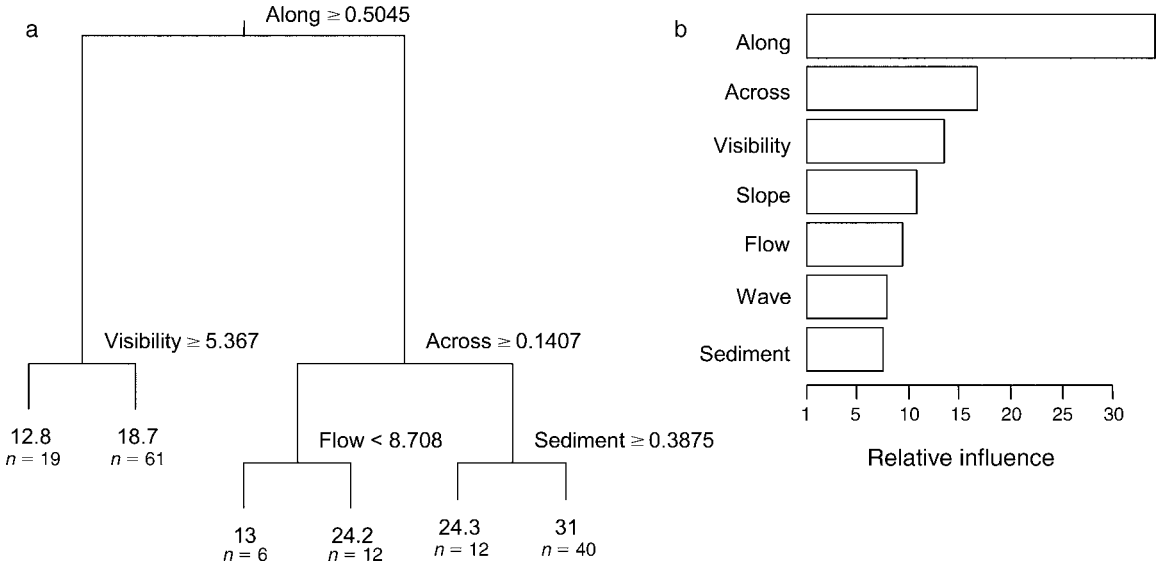

FIG. 3. (a) A single regression tree analysis of soft coral richness data shows five different variables used for the splits. Relative distances across and along account for substantial spatial variation in richness. The training error of the tree was 46.7%, and estimated prediction error was 66.4% (averaged over 10 fivefold cross-validations). (b) The variable importance plot showed that along accounted for most variation, but there was substantial variation due to all other predictors.

the predictor (Breiman et al. 1984). For boosted trees, we simply average the relative influence of each predictor variable over the collection of trees.

Partial dependency plots

Averaging over a collection of trees by bagging or boosting can significantly improve predictions, but the simple interpretation of a single tree is lost. For many types of statistical model, partial dependency plots (Friedman 2001) can be used to visualize dependencies between the response and one or more predictors. The dependency of the response on the remaining predictors is conditioned out. This can be done by choosing a set of values of the predictor(s), predicting the response for each of those values for all cases of the remaining predictors, and then averaging the predictions across the cases. Computationally, this is potentially expensive, but an efficient method—weighted traversal—is available for trees (see Friedman 2001 for details). Plots of these predictions can illustrate joint dependencies of up to three, or even four, predictors.

A Regression Example of Boosting

This analysis has two objectives. First, boosting methods are illustrated, including graphical representations of effects, empirical measures of the model structure, identification of interactions, and the use of monotonic effects. Second, the predictive performance of ABT is compared with alternative methods; single regression trees, bagged regression trees, random forests, and generalized additive regression. Comparisons were based on 10 runs of fivefold cross-validation fits for all methods, and all predictors were included in all the models. However, for boosted trees we additionally fit simpler models based on changes in PE and identified a single best model.

Soft coral richness data

Data on richness of soft coral genera were collected during surveys of 150 reefs on the Great Barrier Reef (GBR), Australia (De'ath and Fabricius 2000). Five physical variables (sedimentation thickness, visibility, wave exposure, slope angle, and flow rate of ambient water) and two spatial variables (relative distances across and along the GBR) were also recorded.

A single regression tree (SRT) was fitted to the richness data with the size of tree, six nodes, selected by cross-validation and the 1-SE rule (Breiman et al. 1984, De'ath and Fabricius 2000) (Fig. 3a). The five splits were based on different variables and the strong splits for across and along the reef indicated strong spatial variation in richness. The importance plot is dominated by along but also shows substantial variation for all predictors (Fig. 3b).

A series of ABTs were then fitted to the data. The first ABT, comprising trees with four splits and using all seven predictors, had PE of 39.9%; a dramatic improvement on the SRT (PE = 64.4%). The variable importance plots shows strong effects for along, moderate effects for across, visibility, wave and slope, and weak effects for flow and sediment (Fig. 4); not greatly different to the importance plot of the SRT. The partial dependency plots of the single predictors show a predominantly linear trend for along, linear trend then a plateau for visibility, modal effects for across, wave and slope, and weak effects for flow and sediment.

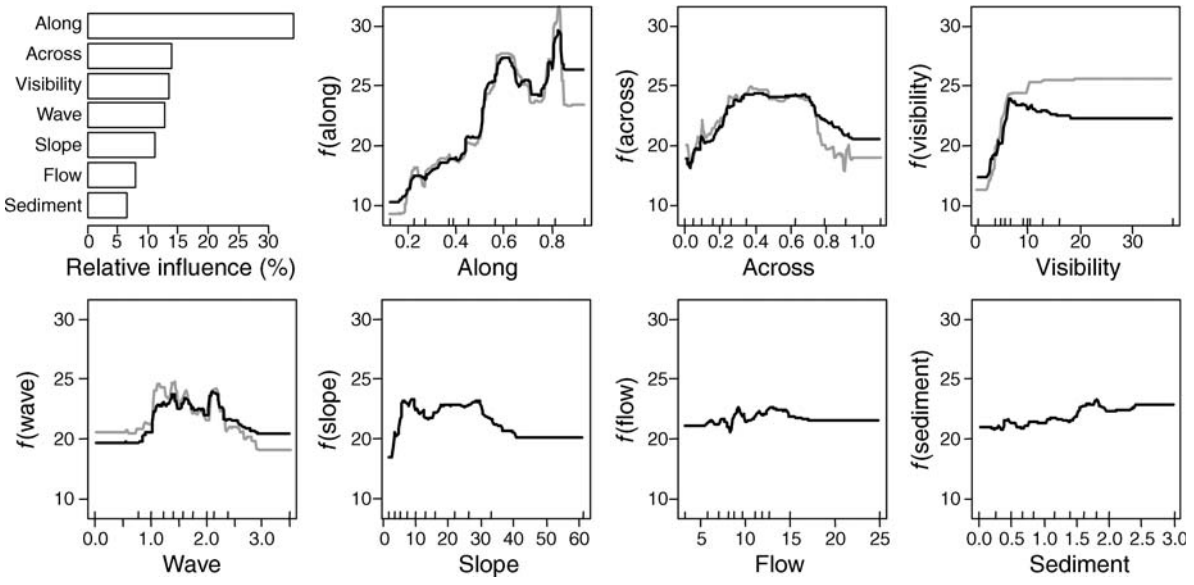

FIG. 4. Relative variable importance plot and partial dependency plots for boosted tree analyses of the soft coral richness data. Two analyses were undertaken. First, all seven predictors were used. The importance plot shows their relative percentage contributions to predicting soft coral richness, and the seven partial plots (black lines) show the dependencies of richness on each of the predictors. A second analysis, dropping slope, flow, and sediment, and restricting the dependence on visibility to be monotonic, is shown in gray lines. The deciles of the distribution of the predictors are indicated by tick marks.

Further ABTs were fitted as follows: (1) sediment, flow, and slope were dropped from the model and reduced PE to 38.1%; (2) fitting the ABT comprising single splits (no interactions) increased PE to 42.4%, indicating that interactions accounted for 4.3% of PE; (3) the ABT with trees of size 2 (first-order interactions) gave a PE of 37.8%, indicating that interactions higher than first-order were negligible; (4) imposing a monotonic increase on the effects of visibility gave a small increase of 0.4% in PE (38.5%) compared to the model without this constraint. This indicates the best model should include the predictors along, across, visibility (possibly monotonic increasing), and sediment, and all first-order interactions. The importance, partial influence, and interactions for the four predictors were used to identify which of the six possible first-order interactions are important. They showed that wave was involved in interactions with along, across, and visibility (Fig. 5).

In the comparison between methods, ABT gave a PE of 39.9%, whereas the six-node single regression tree (66.4%), bagged trees (49.1%), random forests (47.6%), and GAMs (45.6%) all had considerably higher PE. Typical standard deviations of PE were ~0.01% except for SRT (2.1%).

DISCUSSION

Boosted trees (BTs) are clearly a useful tool for analysis of ecological data. They retain the positive

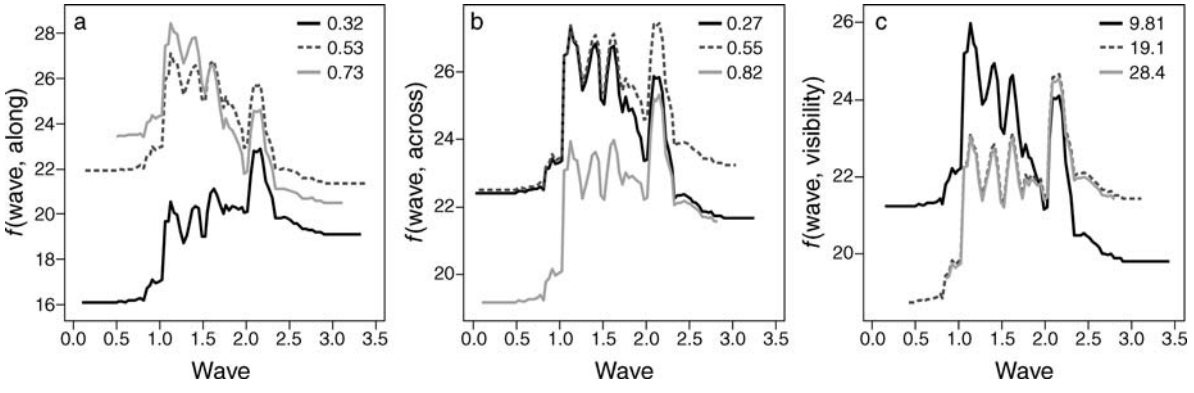

FIG. 5. Partial dependency plots of soft coral richness show the interaction effects of wave with along, across, and visibility. The strong modal effects of wave on richness are clear in all the plots. However, richness increased with wave action for low values of along (0.32) but decreased for high values (0.73). Richness also increased for high values of across (0.82) and increased for moderate to high values of visibility (>19.1) but fell for low values of visibility (9.81).

characteristics of single classification and regression trees, yet also overcome the relatively poor predictive performance and difficulty of interpreting large trees. They consistently outperform other computational intensive methods such as bagged trees and random forests, particularly for regression, and are more interpretable than neural networks and support vector machines. The performance of BTs can also be improved by using aggregated boosted trees (ABTs).

Boosted trees represent an interesting challenge in the context of model selection and prediction. For both model selection and prediction, a widely adopted approach is to generate a collection of plausible models, fit each to the data, then either select the best model according to some criteria (e.g., AIC, BIC), or average over all models weighted by some measure of model performance. Typically these models are parametric and include only linear and/or smooth terms, and main effects. In practice however, the presence of both nonlinearities and interactions is the norm, often diminishing at higher orders to produce tapering effects. Unless the number of predictors is very small, it is implausible that all potential nonlinearities and interactions will be included, and thus the model selection and/or predictions are likely to be suboptimal. For boosted trees however, nonlinearities and interactions are automatically catered for since trees are invariant to transformations of the predictors and interactions are automatically included according to the size of trees used.

BTs typically assume the data are independent, however, BTs and ABTs can also to be adapted to deal with multilevel errors, as is frequently encountered when subsampling or stratified sampling is used. Such data can be accommodated by careful choice of the cross-validation sampling. For example, in the case of a simple nested design, the cross-validation groups should include all or none of the cases in a subsample. A simulated example is shown in Appendix C.

In summary, boosted trees and aggregated boosted trees offer both opportunities and challenges to ecological analysis. They are excellent predictors and also quantify and illustrate the relationships between predictors and the response. They can be used for exploration, explanation and prediction and are simple to use. Software implementations and examples are presented for the R (R Development Core Team 2006) software environment for statistical computation and graphics in Appendices C and D and the Supplement.

Acknowledgments

The concepts and development of classification and regression trees have been inspired and driven by many creative individuals. However two of these individuals, namely, Leo Breiman and Jerome Friedman, have been outstanding in both theoretical development and software implementation. I wish to acknowledge their immense contributions. Thanks are due to Greg Ridgeway for the development of the R package gbm, and to Steve Delean, Katharina Fabricius, and three anonymous reviewers for constructive comments and encouragement. This research was funded by the Australian Institute of Marine Science and Reef CRC, Townsville, Australia.

Literature Cited

Akaike, H. 1974. A new look at the statistical model identification. IEEE Transactions on Automatic Control 19(6):716–723

Breiman, L. 1996. Bagging predictors. Machine Learning 24(2): 123–140.

Breiman, L. 2001. Random forests. Machine Learning 45(1):5–32.

Breiman, L., J. H. Friedman, R. A. Olshen, and C. G. Stone. 1984. Classification and regression trees. Wadsworth International Group, Belmont, California, USA.

Bühlmann, P., and B. Yu. 2003. Boosting with the L2 loss: regression and classification. Journal of the American Statistical Association 98:324–339.

Burnham, K. P., and D. R. Anderson. 1998. Model selection and inference: a practical information theoretic approach. Springer Verlag, New York, New York, USA.

Cappo, M., G. De'ath, S. Boyle, J. Aumend, R. Olbrich, F. Hoedt, C. Colton, P. Perna, and G. Brunskill. 2005. Development of a robust classifier of freshwater residence in barramundi (*Lates calcarifer*) life histories using elemental ratios in scales and boosted regression trees. Marine and Freshwater Research 56:1–11.

De'ath, G. 2002. Multivariate regression trees: a new technique for constrained classification analysis. Ecology 83:1103–1117.

De'ath, G., and K. E. Fabricius. 2000. Classification and regression trees: a powerful yet simple technique for the analysis of complex ecological data. Ecology 81:3178–3192.

Draper, D. 1995. Assessment and propagation of model uncertainty (with discussion). Journal of the Royal Society Series B 57:45–97.

Efron, B., T. Hastie, I. Johnstone, and R. Tibshirani. 2004. Least angle regression. Annals of Statistics 32(2):407–499

Freund, Y., and R. E. Schapire. 1996. Experiments with a new boosting algorithm. Pages 148–156 *in* Machine learning: proceedings of the thirteenth international conference. Morgan Kaufmann, San Francisco, California, USA.

Friedman, J. H. 2001. Greedy function approximation: a gradient boosting machine. Annals of Statistics 29(5):1189–1232.

Friedman, J. H. 2002. Stochastic gradient boosting. Computational Statistics and Data Analysis 38(4):367–378.

Friedman, J. H., T. Hastie, and R. Tibshirani. 2000. Additive logistic regression: a statistical view of boosting. Annals of Statistics 28(2):337–374.

Hastie, T. J., and R. J. Tibshirani. 1990. Generalized additive models. Chapman and Hall, London, UK.

Hastie, T. J., R. J. Tibshirani, and J. H. Friedman. 2001. The elements of statistical learning. Springer-Verlag, New York, New York, USA.

Johnson, D. H. 1999. The insignificance of hypothesis testing. Journal of Wildlife Management 63(3):763–772.

Leathwick, J. R., J. Elith, M. P. Francis, T. Hastie, and P. Taylor. 2006. Variation in demersal fish species richness in the oceans surrounding New Zealand: an analysis using boosted regression trees. Marine Ecology Progress Series 321:267–281.

R Development Core Team. 2006. R: a language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria.

Ripley, B. D. 2004. Selecting amongst large classes of models. Pages 155–170 *in* N. Adams, M. Crowder, D. J. Hand, and D. Stephens, editors. Methods and models in statistics. Imperial College Press, London, UK.

Schwarz, G. 1978. Estimating the dimension of a model. Annals of Statistics 6:461–464.

Vayssières, M. P., R. E. Plant, and B. H. Allen-Diaz. 2000. Classification trees: an alternative non-parametric approach for predicting species distributions. Journal of Vegetation Science 11:679–694.

APPENDIX A

Simulations comparing aggregated boosted trees and boosted trees (*Ecological Archives* E088-015-A1.

APPENDIX B

A classification example of boosting using fish scale data (*Ecological Archives* E088-015-A2).

APPENDIX C

Boosted tree examples in R (*Ecological Archives* E088-015-A3).

APPENDIX D

Fitting boosted trees to data with multi-level errors (*Ecological Archives* E088-015-A4).

SUPPLEMENT

R software packages for boosted trees (*Ecological Archives* E088-015-S1).