define impurity using the log-rank test. As in CART, growing a tree by reducing impurity ensures that terminal nodes are populated by individuals with similar behavior. In the case of a survival tree, terminal nodes are composed of patients with similar survival. The terminal node value in a survival tree is the survival function and is estimated using those patients within the terminal node. This differs from classification and regression trees, where terminal node values are a single value (the estimated class label or predicted value for the response, respectively). Figure 3 shows an example of a survival tree.

Hemant Ishwaran and J. Sunil Rao

See also Decision Trees, Advanced Techniques in Constructing; Recursive Partitioning

Further Readings

Breiman, L., Friedman, J. H., Olshen, R. A., & Stone, C. J. (1984). *Classification and regression trees*. Belmont, CA: Wadsworth.

LeBlanc, M., & Crowley, J. (1993). Survival trees by goodness of split. *Journal of the American Statistical Association, 88,* 457–467.

Segal, M. R. (1988). Regression trees for censored data. *Biometrics, 44,* 35–47.

Stone, M. (1974). Cross-validatory choice and assessment of statistical predictions. *Journal of the Royal Statistical Society, Series B, 36,* 111–147.

Decision Trees, Advanced Techniques in Constructing

Decision trees such as classification, regression, and survival trees offer the medical decision maker a comprehensive way to calculate predictors and decision rules in a variety of commonly encountered data settings. However, performance of decision trees on external data sets can sometimes be poor. Aggregating decision trees is a simple way to improve performance—and in some instances, aggregated tree predictors can exhibit state-of-the-art performance.

Decision Boundary

Decision trees, by their very nature, are simple and intuitive to understand. For example, a binary classification tree assigns data by dropping a data point (case) down the tree and moving either left or right through nodes depending on the value of a given variable. The nature of a binary tree ensures that each case is assigned to a unique terminal node. The value for the terminal node (the predicted outcome) defines how the case is classified. By following the path as a case moves down the tree to its terminal node, the *decision rule* for that case can be read directly off the tree. Such a rule is simple to understand, as it is nothing more than a sequence of simple rules strung together.

The *decision boundary*, on the other hand, is a more abstract concept. Decision boundaries are estimated by a collection of decision rules for cases taken together—or, in the case of decision trees, the boundary produced in the predictor space between classes by the decision tree. Unlike decision rules, decision boundaries are difficult to visualize and interpret for data involving more than one or two variables. However, when the data involve only a few variables, the decision boundary is a powerful way to visualize a classifier and to study its performance.

Consider Figure 1. On the left-hand side is the classification tree for a prostate data set. Here, the outcome is presence or absence of prostate cancer and the independent variables are prostate-specific antigen (PSA) and tumor volume, both having been transformed on the log scale. Each case in the data is classified uniquely depending on the value of these two variables. For example, the leftmost terminal node in Figure 1 is composed of those patients with tumor volumes less than 7.851 and PSA levels less than 2.549 (on the log scale). Terminal node values are assigned by majority voting (i.e., the predicted outcome is the class label with the largest frequency). For this node, there are 54 nondiseased patients and 16 diseased patients, and thus, the predicted class label is nondiseased.

The right-hand side of Figure 1 displays the decision boundary for the tree. The dark-shaded region is the space of all values for PSA and tumor volume that would be classified as nondiseased, whereas the light-shaded regions are those values classified as diseased. Superimposed on the figure,

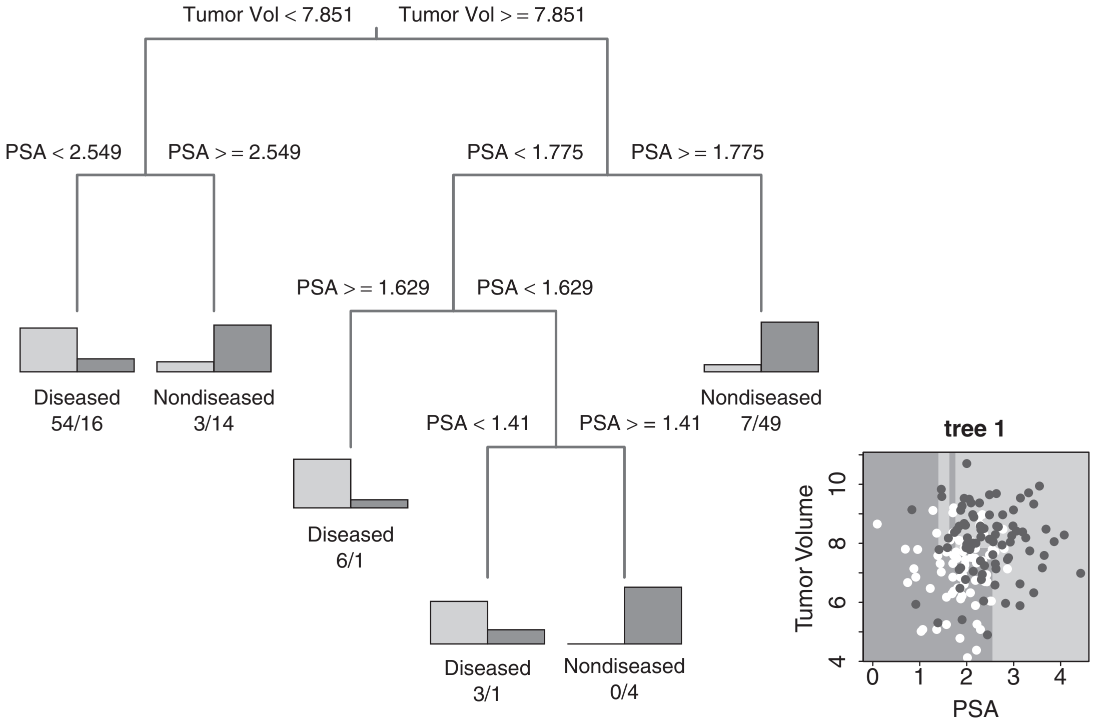

Figure 1 Decision tree (left-hand side) and decision boundary (right-hand side) for prostate cancer data with prostate-specific antigen (PSA) and tumor volume as independent variables (both transformed on the log scale)

Note: Barplots under terminal nodes of the decision tree indicate proportion of cases classified as diseased or nondiseased, with the predicted class label determined by majority voting. Decision boundary shows how the tree classifies a new patient based on PSA and tumor volume. Gray-shaded points identify diseased patients, and white points identify nondiseased patients from the data.

using white and light-gray dots, are the observed data points from the original data. Light-gray points are truly diseased patients, whereas white points are truly nondiseased patients. Most of the light-gray points fall in the light-shaded region of the decision space and, likewise, most of the white points fall in the dark-shaded region of the decision space, thus showing that the classifier is classifying a large fraction of the data correctly. Some data points are misclassified, though. For example, there are several light-gray points in the center of the plot falling in the dark-shaded region. As well, there are four light-gray points with small tumor volumes and PSA values falling in the dark-shaded region. The misclassified data points in the center of the decision space are especially troublesome. These points are being misclassified because the decision space for the tree is rectangular. If the decision boundary were smoother, then these points would not be misclassified. The nonsmooth nature of the decision boundary is a well-known deficiency of classification trees and can seriously degrade performance, especially in complex decision problems involving many variables.

Instability of Decision Trees

Decision trees, such as classification trees, are known to be unstable. That is, if the original data set is changed (perturbed) in some way, then the classifier constructed from the altered data can be surprisingly different from the original classifier. This is an undesirable property, especially if small perturbations to the data lead to substantial differences.

This property can be demonstrated using the prostate data set of Figure 1. However, to show this, it is important to first agree on a method for perturbing the data. One technique that can be used is to employ bootstrap resampling. A bootstrap sample is a special type of resampling procedure. A data point is randomly selected from the data and then returned. This process is repeated n times, where n is the sample size. The resulting bootstrap sample consists of n data points but will contain replicated data. On average, a bootstrap sample draws only approximately 63% of the original data.

A total of 1,000 different bootstrap samples of the prostate data were drawn. A classification tree was calculated for each of these 1,000 samples. The top panel of plots in Figure 2 shows decision boundaries for four of these trees (bootstrap samples 2, 5, 25, and 1,000; note that Tree 1 is the classification tree from Figure 1 based on the original data). One can see clearly that the decision spaces differ quite substantially—thus providing clear evidence of the instability.

It is also interesting to note how some of the trees have better decision spaces than the original tree (recall Figure 1; also see Tree 1 in Figure 2). For example, Trees 2, 5, 25, and 1,000 identify some or all of the four problematic light-gray points appearing within the lower quadrant of the dark-shaded region of the original decision space. As well, Trees 5, 25, and, 1,000 identify some of the problematic green points appearing within the center of the original decision space.

An important lesson that emerges from this example is not only that decision trees can be unstable but also that trees constructed from different perturbations of the original data can produce decision boundaries that in some instances have better behavior than the original decision space (over certain regions). Thus, it stands to reason that, if one could combine many such trees, the classifier formed by aggregating the trees might have better overall performance. In other words, *the whole may be greater than the sum of the parts* and one may be able to capitalize on the inherent instability using aggregation to produce more accurate classifiers.

Bagging

This idea in fact is the basis for a powerful method referred to as "bootstrap aggregation," or simply "bagging." Bagging can be used for many kinds of predictors, not just decision trees. The basic premise for bagging is that, if the underlying predictor is unstable, then aggregating the predictor over multiple bootstrap samples will produce a more accurate, and more stable, procedure.

To bag a classification tree, the procedure is as follows (bagging can be applied to regression trees and survival trees in a similar fashion):

1. Draw a bootstrap sample of the original data.
2. Construct a classification tree using data from Step 1.
3. Repeat Steps 1 and 2 many times, independently.
4. Calculate an aggregated classifier using the trees formed in Steps 1 to 3. Use majority voting to classify a case. Thus, to determine the predicted outcome for a case, take the majority vote over the predicted outcomes from each tree in Steps 1 to 3.

The bottom panel of plots in Figure 2 shows the decision boundary for the bagged classifier as a function of number of trees (based on the same prostate data as before). The first plot is the original classifier based on all the data (Tree 1). The second plot is the bagged classifier composed of Tree 1 and the bootstrap tree derived using the first bootstrap sample. The third plot is the bagged classifier using Tree 1 and the first four bootstrapped trees, and so forth. As number of trees increases, the bagged classifier becomes more refined. Even the decision boundary for the bagged classifier using only five trees (third plot) is substantially smoother than the original classifier and is able to better classify problematic cases. By 1,000 trees (last plot), the bagged classifier's decision boundary is fully defined. The accuracy of the bagged classifier is substantially better than any single bootstrapped tree. Table 1 records the misclassification (error) rate for the bagged predictor against the averaged error rate for the 1,000 bootstrapped trees. The first column is the overall error rate, the second column is the error rate for diseased patients, and the third column is the error rate for nondiseased patients. Error rates were calculated using out-of-bag data. Recall that each bootstrap sample uses on average 67% of the original data. The remaining 33% of the data is called out-of-bag and serves as test data, as it is not used in constructing the tree. Table 1 shows that

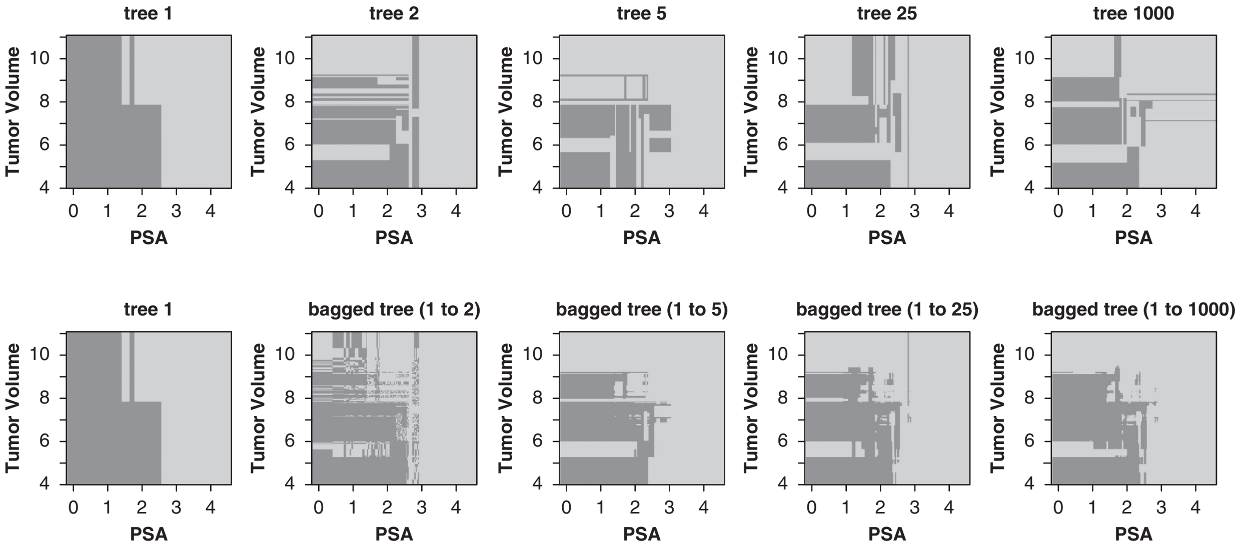

Figure 2 Top row shows decision boundary for a specific bootstrapped tree (1,000 trees used in total), and the bottom plot shows different aggregated (bagged) decision trees

Note: Bagged trees are more robust to noise (stable) because they utilize information from more than one tree. The most stable bagged tree is the one on the extreme right-hand side and shows decision boundary using 1,000 trees.

the bagged classifier is substantially more accurate than any given tree.

Random Forests

“Random forests” is a refinement of bagging that can yield even more accurate predictors. The method works like bagging by using bootstrapping and aggregation but includes an additional step that is designed to encourage independence of trees. This effect is often most pronounced when the data contain many variables.

To create a random forest classifier, the procedure is as follows (regression forests and random survival forests can be constructed using the same principle):

1. Draw a bootstrap sample of the original data.
2. Construct a classification tree using data from Step 1. For each node in the tree, determine the optimal split for the node using *M* randomly selected dependent variables.
3. Repeat Steps 1 and 2 many times, independently.
4. Calculate an aggregated classifier using the trees formed in Steps 1 to 3. Use majority voting to classify a case. Thus, to determine the predicted outcome for a case, take the majority vote over the predicted outcomes from each tree in Steps 1 to 3.

Step 2 is the crucial step distinguishing forests from bagging. Unlike bagging, each bootstrapped tree is constructed using different variables, and not all variables are used (at most *M* are used at each node in the tree growing process). Considerable empirical evidence has shown that forests can be substantially more accurate because of this feature.

Boosting

Boosting is another related technique that has some similarities to bagging although its connection is not as direct. It too can produce accurate

Table 1 Misclassification error rate (in percentage) for bagged classifier (1,000 trees) and single tree classifier

Classifier	*All*	*Diseased*	*Nondiseased*
Bagged tree	27.2	28.8	25.9
Single tree	34.9	36.7	33.0

classifiers through a combination of reweighting and aggregation. To create a boosted tree classifier, the following procedure can be used (although other methods are also available in the literature):

1. Draw a bootstrap sample from the original data giving each observation equal chance (i.e., weight) of appearing in the sample.
2. Build a classification tree using the bootstrap data and classify each of the observations, keeping track of which ones are classified incorrectly or correctly.
3. For those observations that were incorrectly classified, increase their weight and correspondingly decrease the weight assigned to observations that were correctly classified.
4. Draw another bootstrap sample using the newly updated observation weights (i.e., those observations that were previously incorrectly classified will have a greater chance of appearing in the next bootstrap sample).
5. Repeat Steps 2 to 4 many times.
6. Calculate an aggregated classifier using the trees formed in Steps 1 to 5. Use majority voting to classify a case. Thus, to determine the predicted outcome for a case, take the majority vote over the predicted outcomes from each tree in Steps 1 to 5.

The idea of reweighting observations adaptively is a key to boosting's performance gains. In a sense, the algorithm tends to focus more and more on observations that are difficult to classify. There has been much work in the literature on studying the operating characteristics of boosting, primarily motivated by the fact that the approach can produce significant gains in prediction accuracy over a single tree classifier. Again, as with bagging, boosting is a general algorithm that can be applied to more than tree-based classifiers. While these aggregation algorithms were initially thought to destroy the simple interpretable structure (topology) produced by a single tree classifier, recent work has shown that, in fact, treelike structures (with respect to the decision boundary) are often maintained, and interpretable structure about how the predictors interact with one another can still be gleaned.

Hemant Ishwaran and J. Sunil Rao

See also Decision Tree: Introduction; Recursive Partitioning

Further Readings

Breiman, L. (1996). Bagging predictors. *Machine Learning, 26,* 123–140.

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

Breiman, L., Friedman, J. H., Olshen, R. A., & Stone, C. J. (1984). *Classification and regression trees.* Belmont, CA: Wadsworth.

Efron, B. (1982). *The jackknife, the bootstrap and other resampling plans* (Society for Industrial and Applied Mathematics CBMS-NSF Monographs, No. 38). Philadelphia: SIAM.

Freund, Y., & Shapire, R. E. (1996). Experiments with a new boosting algorithm. In *Machine Learning: Proceedings of the 13th International Conference* (pp. 148–156). San Francisco: Morgan Kaufman.

Ishwaran, H., Kogalur, U. B., Blackstone, E. H., & Lauer, M. S. (2008). Random survival forests. *Annals of Applied Statistics, 2*(3), 841–860.

Rao, J. S., & Potts, W. J. E. (1997). Visualizing bagged decision trees. In *Proceedings of the 3rd International Conference on Knowledge Discovery and Data Mining* (pp. 243–246). Newport Beach, CA: AAAI Press.

Decision Trees, Construction

A decision model is a mathematical formulation of a decision problem that compares alternative choices in a formal process by calculating their expected outcome. The decision tree is a graphical representation of a decision model that represents the basic elements of the model. The key elements of the model are the possible *choices*, *information* about chance events, and *preferences* of the decision maker. The choices are the alternatives being compared in the decision model. The information consists of an enumeration of the events that may occur consequent to the choice and the probabilities of each of their outcomes. Preferences are