Decision Tree: Introduction

A decision tree is a powerful method for classification and prediction and for facilitating decision making in sequential decision problems. This entry considers three types of decision trees in some detail. The first is an algorithm for a recommended course of action based on a sequence of information nodes; the second is classification and regression trees; and the third is survival trees.

Decision Trees

Often the medical decision maker will be faced with a sequential decision problem involving decisions that lead to different outcomes depending on chance. If the decision process involves many sequential decisions, then the decision problem becomes difficult to visualize and to implement. Decision trees are indispensable graphical tools in such settings. They allow for intuitive understanding of the problem and can aid in decision making.

A decision tree is a graphical model describing decisions and their possible outcomes. Decision trees consist of three types of nodes (see Figure 1):

1. *Decision node:* Often represented by squares showing decisions that can be made. Lines emanating from a square show all distinct options available at a node.
2. *Chance node:* Often represented by circles showing chance outcomes. Chance outcomes are events that can occur but are outside the ability of the decision maker to control.
3. *Terminal node:* Often represented by triangles or by lines having no further decision nodes or chance nodes. Terminal nodes depict the final outcomes of the decision making process.

For example, a hospital performing esophagectomies (surgical removal of all or part of the esophagus) for patients with esophageal cancer wishes to define a protocol for what constitutes an adequate lymphadenectomy in terms of total number of regional lymph nodes removed at surgery. The hospital believes that such a protocol should be guided by pathology (available to the surgeon prior to surgery). This information should include histopathologic cell type (squamous cell carcinoma or adenocarcinoma); histopathologic grade (a crude indicator of tumor biology); and depth of tumor invasion (PT classification). It is believed that number of nodes to be removed should increase with more deeply invasive tumors when histopathologic grade is poorly differentiated and that number of nodes differs by cell type.

The decision tree in this case is composed predominantly of chance outcomes, these being the results from pathology (cell type, grade, and tumor depth). The surgeon's only decision is whether to perform the esophagectomy. If the decision is made to operate, then the surgeon follows this decision line on the graph, moving from left to right, using pathology data to eventually determine the terminal node. The terminal node, or final outcome, is number of lymph nodes to be removed.

Decision trees can in some instances be used to make optimal decisions. To do so, the terminal nodes in the decision tree must be assigned terminal values (sometimes called payoff values or endpoint values). For example, one approach is to assign values to each decision branch and chance branch and define a terminal value as the sum of branch values leading to it. Once terminal values are assigned, tree values are calculated by following terminal values from right to left. To calculate the value of chance outcomes, multiply by their probability. The total for a chance node is the total of these values. To determine the value of a decision node, the cost of each option along each decision line is subtracted from the cost already calculated. This value represents the benefit of the decision.

Classification Trees

In many medical settings, the medical decision maker may not know what the decision rule is. Rather, he or she would like to discover the decision rule by using data. In such settings, decision trees are often referred to as classification trees. Classification trees apply to data where the y-value (outcome) is a classification label, such as the disease status of a patient, and the medical decision maker would like to construct a decision rule that predicts the outcome using x-variables (dependent variables) available in the data. Because the data set available is just one sample of the underlying

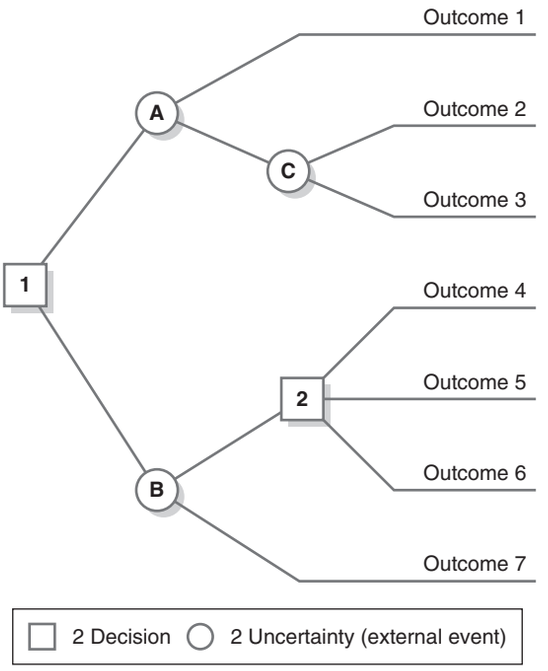

Figure 1 Decision trees are graphical models for describing sequential decision problems.

population, it is desirable to construct a decision rule that is accurate not only for the data at hand but over external data as well (i.e., the decision rule should have good prediction performance). At the same time, it is helpful to have a decision rule that is understandable. That is, it should not be so complex that the decision maker is left with a black box. Decision trees offer a reasonable way to resolve these two conflicting needs.

Background

The use of tree methods for classification has a history that dates back at least 40 years. Much of the early work emanated from the area of social sciences, starting in the late 1960s, and computational algorithms for automatic construction of classification trees began as early as the 1970s. Algorithms such as the THAID program developed at the Institute for Social Research, University of Michigan, laid the groundwork for recursive partitioning algorithms, the predominate algorithm used by modern-day tree classifiers, such as Classification and Regression Tree (CART).

An Example

Classification trees are decision trees derived using recursive partitioning data algorithms that classify each incoming x-data point (case) into one of the class labels for the outcome. A classification tree consists of three types of nodes (see Figure 2):

1. *Root node:* The top node of the tree comprising all the data.
2. *Splitting node:* A node that assigns data to a subgroup.
3. *Terminal node:* Final decision (outcome).

Figure 2 is a CART tree constructed using the breast cancer databases obtained from the University of Wisconsin Hospitals, Madison (available from http://archive.ics.uci.edu/ml). In total, the data comprise 699 patients classified as having either benign or malignant breast cancer. The goal here is to predict true disease status based on nine different variables collected from biopsy.

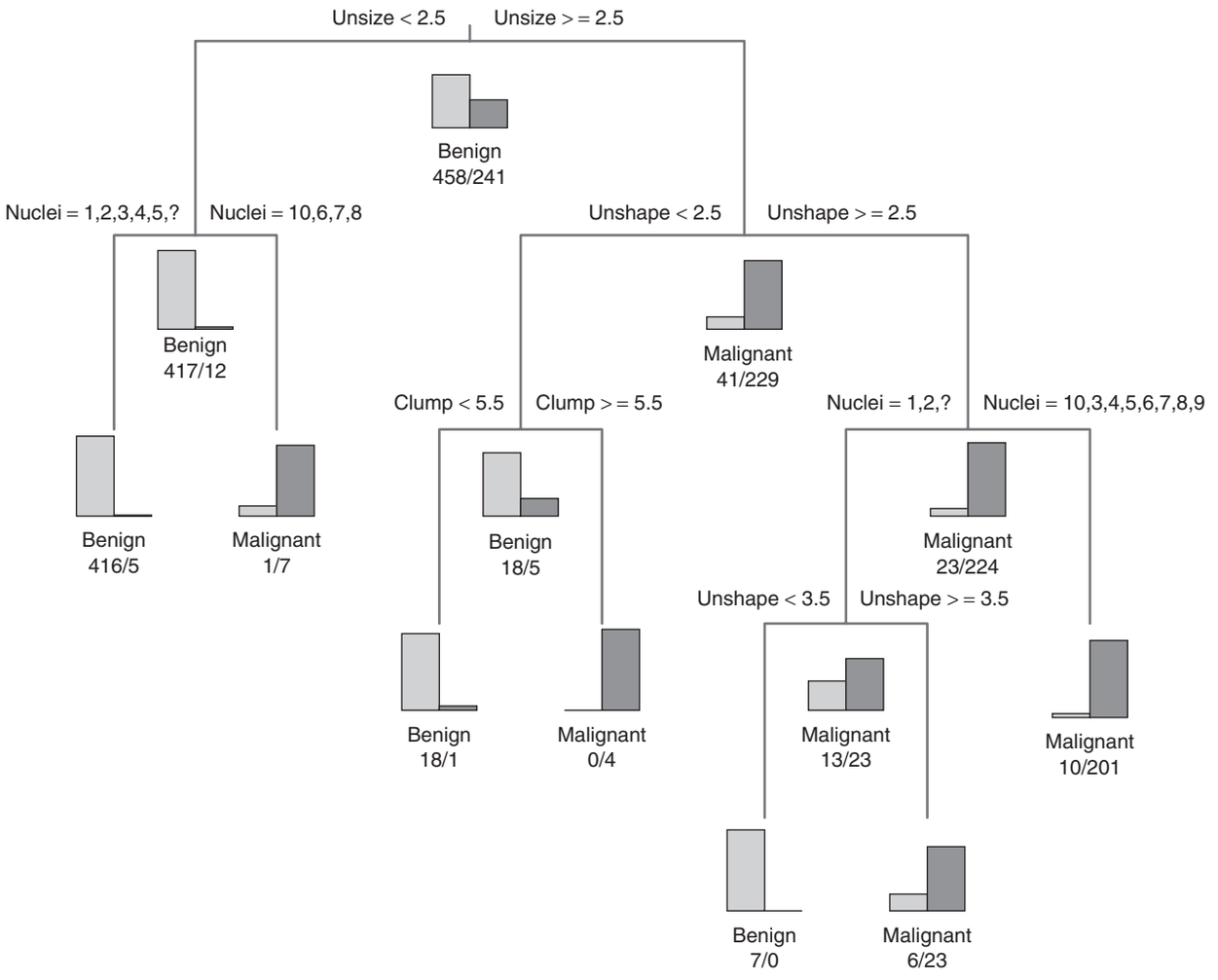

Figure 2 Classification tree for Wisconsin breast cancer data

Note: Light-shaded and dark-shaded barplots show frequency of data at each node for the two classes: benign (light shaded); malignant (dark shaded). Terminal nodes are classified by majority voting (i.e., assignment is made to the class label having the largest frequency). Labels in black given above a splitting node show how data are split depending on a given variable. In some cases, there are missing data, which are indicated by a question mark.

The first split of the tree (at the root node) is on the variable "unsize," measuring uniformity of cell size. All patients having values less than 2.5 for this variable are assigned to the left node (the left daughter node); otherwise they are assigned to the right node (right daughter node). The left and right daughter nodes are then split (in this case, on the variable "unshape" for the right daughter node and on the variable "nuclei" for the left daughter node), and patients are assigned to subgroups defined by these splits. These nodes are then split, and the process is repeated recursively in a procedure called recursive partitioning. When the tree construction is completed, terminal nodes are assigned class labels by majority voting (the class label with the largest frequency). Each patient in a given terminal node is assigned the predicted class label for that terminal node. For example, the leftmost terminal node in Figure 2 is assigned the class label "benign" because 416 of the 421 cases in the node have that label. Looking at Figure 2, one can see that voting heavily favors one class over the other for all terminal nodes, showing that the decision tree is accurately classifying the data. However, it is important to assess accuracy using external data sets or by using cross-validation as well.

Recursive Partitioning

In general, recursive partitioning works as follows. The classification tree is grown starting at the root node, which is the top node of the tree, comprising all the data. The root node is split into two daughter nodes: a left and a right daughter node. In turn, each daughter node is split, with each split giving rise to left and right daughters. The process is repeated in a recursive fashion until the tree cannot be partitioned further due to lack of data or some stopping criterion is reached, resulting in a collection of terminal nodes. The terminal nodes represent a partition of the predictor space into a collection of rectangular regions that do not overlap. It should be noted, though, that this partition may be quite different than what might be found by exhaustively searching over all partitions corresponding to the same number of terminal nodes. However, for many problems, exhaustive searches for globally optimal partitions (in the sense of producing the most homogeneous leaves) are not computationally feasible, and recursive partitioning represents an effective way of undertaking this task by using a one-step procedure instead.

A classification tree as described above is referred to as a *binary recursive partitioned tree*. Another type of recursively partitioned tree is multiway recursive partitioned tree. Rather than splitting the parent node into two daughter nodes, such trees use multiway splits that define multiple daughter nodes. However, there is little evidence that multiway splits produce better classifiers, and for this reason, as well as for their simplicity, binary recursive partitioned trees are often favored.

Splitting Rules

The success of CART as a classifier can be largely attributed to the manner in which splits are formed in the tree construction. To define a good split, CART uses an impurity function to measure the decrease in tree impurity for a split. The purity of a tree is a measure of how similar observations in the leaves are to one another. The best split for a node is found by searching over all possible variables and all possible split values and choosing that variable and split that reduces impurity the most. Reduction of tree impurity is a good principle because it encourages the tree to push dissimilar cases apart. Eventually, as the number of nodes increases, and dissimilar cases become separated into daughter nodes, each node in the tree becomes homogeneous and is populated by cases with similar outcomes (recall Figure 2).

There are several impurity functions used. These include the twoing criterion, the entropy criterion, and the gini index. The gini index is arguably the most popular. When the outcome has two class labels (the so-called two-class problem), the gini index corresponds to the variance of the outcome if the class labels are recoded as being 0 and 1.

Stopping Rules

The size of the tree is crucial to the accuracy of the classifier. If the tree is too shallow, terminal nodes will not be pure (outcomes will be heterogeneous), and the accuracy of the classifier will suffer. If the tree is too deep (too many splits), then the number of cases within a terminal node will be small, and the predicted class label will have high variance—again undermining the accuracy of the classifier.

To strike a proper balance, pruning is employed in methodologies such as CART. To determine the optimal size of a tree, the tree is grown to full size (i.e., until all data are spent) and then pruned back. The optimal size is determined using a complexity measure that balances the accuracy of the tree as measured by cost complexity and by the size of the tree.

Regression Trees

Decision trees can also be used to analyze data when the *y*-outcome is a continuous measurement (such as age, blood pressure, ejection fraction for the heart, etc.). Such trees are called regression trees. Regression trees can be constructed using recursive partitioning similar to classification trees. Impurity is measured using mean-square error. The terminal node values in a regression tree are defined as the mean value (average) of outcomes for patients within the terminal node. This is the predicted value for the outcome.

Survival Trees

Time-to-event data are often encountered in the medical sciences. For such data, the analysis

focuses on understanding how time-to-event varies in terms of different variables that might be collected for a patient. Time-to-event can be time to death from a certain disease, time until recurrence (for cancer), time until first occurrence of a symptom, or simple all-cause mortality.

The analysis of time-to-event data is often complicated by the presence of censoring. Generally speaking, this means that the event times for some individuals in a study are not observed exactly and are only known to fall within certain time intervals. Right censoring is one of the most common types of censoring encountered. This occurs when the event of interest is observed only if it occurs prior to some prespecified time. For example, a patient might be monitored for 2 weeks without occurrence of a symptom and then released from a hospital. Such a patient is said to be right censored because the time-to-event must exceed 2 weeks, but the exact event time is unknown. Another example of right censoring occurs when patients enter a study at different times and the study is predetermined to end by a certain time. Then, all patients who do not experience an event within the study period are right censored.

Decision trees can be used to analyze right-censored survival data. Such trees are referred to as survival trees. Survival trees can be constructed using recursive partitioning. The measure of impurity plays a key role, as in CART, and this can be defined in many ways. One popular approach is to

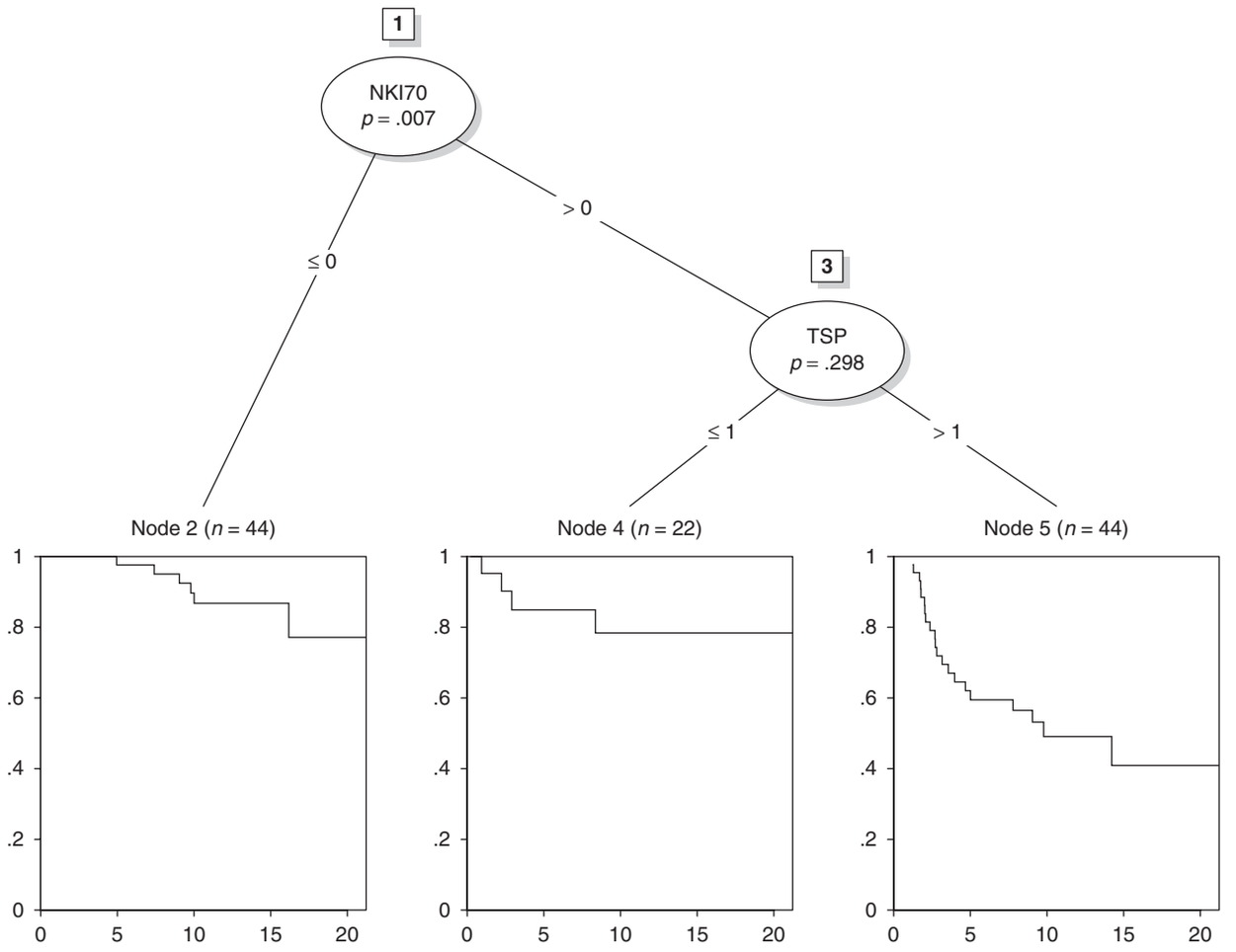

Figure 3 Binary survival tree for breast cancer patients

Note: Dependent variables NKI70 and TSP are gene signatures. For example, extreme right terminal node (Node 5) corresponds to presence of both the NKI70 and TSP gene signatures. Underneath each terminal node are Kaplan-Meier survival curves for patients within that node.

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,