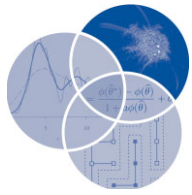
Overview



Decision trees

Barry de Ville

Decision trees trace their origins to the era of the early development of written records. This history illustrates a major strength of trees: exceptionally interpretable results which have an intuitive tree-like display which, in turn, enhances understanding and the dissemination of results. The computational origins of decision trees—sometimes called classification trees or regression trees—are models of biological and cognitive processes. This common heritage drives complementary developments of both statistical decision trees and trees designed for machine learning. The unfolding and progressive elucidation of the various features of trees throughout their early history in the late 20th century is discussed along with the important associated reference points and responsible authors. Statistical approaches, such as a hypothesis testing and various resampling approaches, have coevolved along with machine learning implementations. This had resulted in exceptionally adaptable decision tree tools, appropriate for various statistical and machine learning tasks, across various levels of measurement, with varying levels of data quality. Trees are robust in the presence of missing data and offer multiple ways of incorporating missing data in the resulting models. Although trees are powerful, they are also flexible and easy to use methods. This assures the production of high quality results that require few assumptions to deploy. The treatment ends with a discussion of the most current developments which continue to rely on the synergies and cross-fertilization between statistical and machine learning communities. Current developments with the emergence of multiple trees and the various resampling approaches that are employed are discussed. © 2013 Wiley Periodicals, Inc.

How to cite this article:
WIREs Comput Stat 2013, 5:448–455. doi: 10.1002/wics.1278

Keywords: decision trees; rule induction; predictive models; machine learning; boosting; random forests

INTRODUCTION

Decision trees are general purpose prediction and classification mechanisms that were among the first statistical algorithms to be implemented in electronic form during the adoption of digital circuitry to electronic computations in the later decades of the 20th century. They have evolved to become highly cross-disciplinary, general purpose computationally intensive methods for prediction and classification, artificial intelligence, machine learning, knowledge discovery, and inductive rule-builders that are used in a range of data mining, knowledge discovery, machine learning, and artificial intelligence tasks.

The main characteristic of decision trees is a recursive subsetting of a target field of data according to the values of associated input fields or predictors to create partitions, and associated descendent data subsets (called leaves or nodes), that contain progressively similar intra-leaf (or intra-node) target values and progressively dissimilar inter-leaf (or inter-node values) at any given level of the tree.

The 'Porphyrian tree', a form of decision tree, is the oldest known type of classification tree diagram and was conceived by the Greek philosopher

*Correspondence to: barry.deville@sas.com
SAS Institute Inc., Cary, NC, USA

Conflict of interest: The authors have declared no conflicts of interest for this article.

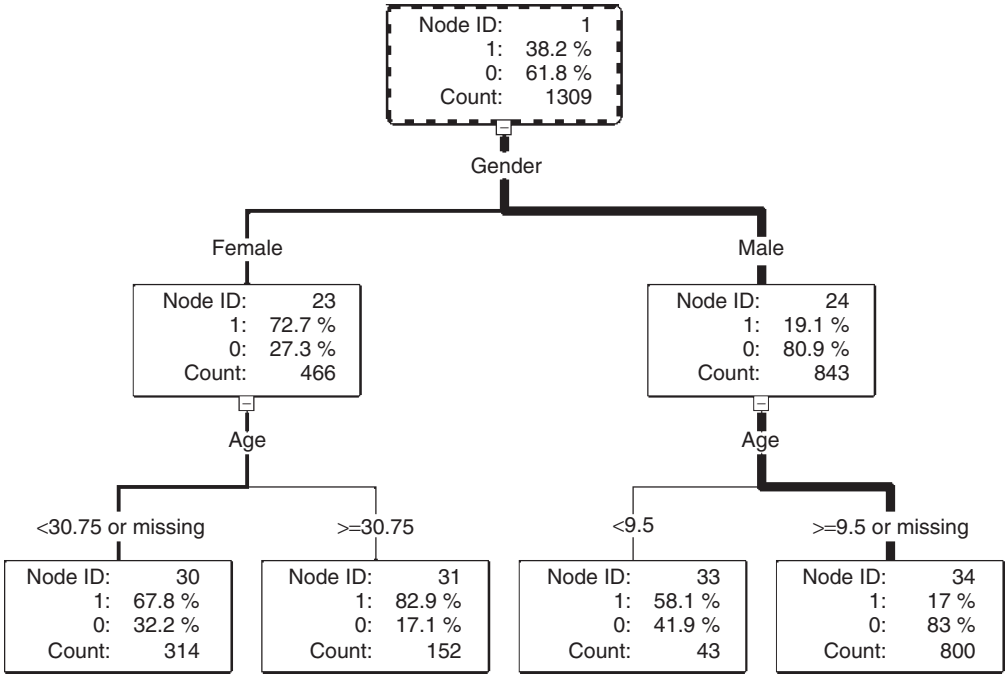


FIGURE 1 | A decision tree illustrating analysis of survival in Titanic sinking

Porphyry in the 3rd century C.E.[1] These early precomputational origins of decision trees confirm a persistently useful, innate capability of decision trees to project and encapsulate contextually revealing visual displays that are both intuitive and powerful visual metaphors. If we fast forward to the 20th century, we see that computational decision trees emerged at the same time as the nascent fields of artificial intelligence[2,a] and statistical computation. As a result their development has benefitted from a rich cross-disciplinary cross-fertilization that has led to a range of new methods—from resampling methods like boosting and bagging—to more recent generalized multiple tree methods such as Random Forests.

OPERATION, FEATURES, AND INTERPRETATION

The characteristic form of decision trees is shown in Figure 1. Here we see a recursive subsetting of a target field of data according to the values of associated fields to create partitions, and associated descendent data subsets (nodes), that contain progressively similar intra-node target values and progressively dissimilar inter-node values at any given level of the tree.

Figure 1 shows a decision tree analysis perfomed on data that are drawn from research conducted on passengers on the ill-fated Titanic.[2] The top-most node of the tree—termed the 'root node'—contains 1309 observations. This top-most root node contains the global distribution of the 'target' field for the analysis: in this case, survival versus nonsurvival. In general, targets may be any level of measurement; e.g. nominal, ordinal, or interval. When nominal targets are used, as in the case shown in Figure 1, the tree is sometimes referred to as a 'classification tree'.

In Figure 1, the overall survival rate—repesented by '1' in the data—is 38%. Marginal counts are sometimes presented alongside the percentages so as to display the actual number of observations that fall into the two respective categories. In Figure 1, only the total number of observations are displayed at the bottom of the node display (labeled as Node ID: 1).

The decision tree unfolds in a stepwise fashion: the tree is formed by first partitioning the root node to form branches that define the descendent leaves (or nodes) that form clusters of observations that are alike within a node yet dissimilar when compared to other nodes at any given level of the tree. The branch partitions are based on a selection that is taken from a search through the data set to discover fields of data that can be input as partitioning fields to best describe the variablility among the target values that are displayed in the root node. Potential partitioning fields are thereby termed 'inputs'. Once an input is selected, the descendent leaves, or nodes, are produced. (terminal nodes are usually called ''leaves''). In Figure 1, the first level

 © 2013 Wiley Periodicals, Inc.

of the decision tree is produced by selecting the 'Gender' field as the best input field from the set of inputs that are available (other inputs in this data set include passenger age, cabin class (first, second, and so on), fare paid, cabin location, boarding location, and destination).

The selection of the 'best' input field is an open subject of active research. Decision trees allow for a variety of computational approaches to input selection. The top-down graphical display also supports the exploration of various effects visually, so that strong branches—or compelling branches—may be selected based on theoretical notions about the interaction of the various model components. In this example, the 'best' field selection is based on partition strength diagnostics produced by the software, coupled with the domain knowledge of the analyst. In the Titanic data, gender, age, and cabin class are all important and predictive inputs with multiple, interweaving interactions. A complete exposition of the various interactions is not possible in the limited space here. Consequently, gender alone is used in the description here so as to present a simple, hopefully compelling, result. This result, and the domain knowledge framework that describes it, is presented below.

The descendent nodes produced by the selection of gender as the first partitioning field in Figure 1 are commonly referred to as the first level of the tree. The leaves in this first level correspond to the male and female passengers. The 'leaf' terminology is often used when the decision 'tree' metaphor for this method is used. The more general term 'node' is used in recognition of the fact that decision trees are a particular form of connected graph. In graph terminology, the partitions are 'edges' and the leaves are 'nodes'.

Using the 'node' terminology, the first level of the tree has two descendent nodes: the 'female' descendent has a survival rate of about 72%, whereas the 'male' descendent node has a survival rate of only 19%. It is normal, as in this case, to select the the input that produces the most dramatic separation in the variability among the descendent nodes. In practice, the analyst may often guide the sequence of the unfolding of branch partitions in order to support a better explanation of a sequence of effects or to support and confirm the conditional relations that are assumed to exist among the various inputs and the component nodes that they produce. In the case of high performance predictive modeling applications there is less emphasis on analyst interaction in the formation of the tree and more emphasis on the selection of high quality partitions that can collectively produce the best overall model. Regardless of the method, once the initial level of the tree is determined the process continues in a recursive fashion until one of more possible stop conditions are met, thus terminating the process. Generally, stopping rules consist of thresholds on diminishing returns (in terms of test statistics) or in a diminishing supply of training cases (minimum acceptable number of observations in a node).

As shown in Figure 1, gender is selected as the first partitioning field below the root node. In this case, we see that the use of gender as the partitioning field forms two descendent nodes for female and male passengers, respectively. One interpretation might be to note that the effect of gender is strong and appears to follow a protocol that calls for 'women and children first' in the lifeboats. Here we see that, while the overall survival rate is 38%, this increases to about 73% among females whereas the overall male survival rate drops to about 19%. The descendent nodes formed by recursively partitioning the female and male nodes, respectively, illustrate one of the most striking and useful features of decision trees: here we see the contextual effect of age on survival rate. In this case, we see that among females, older ages are more likely to survive (83% survival rate among older females vs 68% survival rate among younger females). In the male population, the effect is completely reversed: older males have a substantially lower survival rate (17% vs 58% is older males compared with younger males).

We can interpret these findings as normative behavior in the social dynamics that evolved in this impromptu community that consists of the self-selected passengers of this inaugural voyage across the Atlantic. Our initial sense of the 'women and children first' protocol—displayed in the first partition—is reinforced by normative behavior that demonstrates preferential treatment based on age status. Because the second tier partitions are unique to female and male groups, respectively, we see a contrasting preferential age treatment among females compared with males. This contrast favors older females and younger males. This asymmetry in the descendent nodes on the second level of the tree provides a dramatic illustration of the outstanding ability of decision trees to expose relationships in context.

The enduring legacy of decision trees is that they demonstrate that multiple contributors need to be recruited to effectively explain a relationship. Further, the form of the resulting relationships will reveal multiple contextual effects that will influence the understanding and effective presentation of the results. The utility of decision trees in detecting and presenting contextual effects was a significant driver to the development of one of the earliest and most

 © 2013 Wiley Periodicals, Inc.

influencial computer implementations of decision trees: the Automatic Interaction Detection (AID) program developed at the University of Michigan.[3]

The bottom level of the tree presented in Figure 1 shows some other useful and important features of decision trees: we see that the partitions that form the branches of the respective female and male descendent nodes employ different cut-points: among females the cut-point for the age partition lies at 30.75, whereas among males the cut-point lies at 9.5. Here we see that the decision tree methodology employs similarity search algorithms to find the most discriminating cut-points among the branches of potential partitions. This means that members of a given partition are as much alike as possible. This determination is often made on the basis of a statistical test of differences between values. Although binary partitions are shown here for differences in the age dimension, the technique allows for the use of multiway (k-way) partitions. The multiway partitions were developed as an enhancement to the method of AID.[4] Whether binary or k-way partitions are used depends much upon analyst preference; in any case, methods are usually applied to define cut-points which attempt to identify differences among nodes that are statistically significant and which generalize well to novel data.

We can also see that the labels for the age partitions in Figure 1 indicate that missing values are being used in the determination of partition cut-points. Decision trees can explicitly include missing values as valid values in the determination of branch partitions. When included in the analysis, missing values for a given input are often allowed to group with other values that they most closely resemble in terms of relation to the target. Missing values may also be included as a separate value and so may form a distinct partition on their own. Of course, missing values can also be excluded from the analysis.

This brief illustration displays some of the distinctive features of the decision tree method:

- The form of the tree results from a stepwise and recursive partitioning of a target field according to significant discriminating features of one or more associated input fields.
- All levels of measurement—nominal, ordinal, and interval—are automatically accommodated in either target or input position in the tree formation.
- Successive partitioning results in the presentation of a tree-like visual display with a top node and descendent branches.
- The automatic treatment of various levels of measurement and the associated visual display contribute to model flexibility, ease of use and ease of presentation and interpretation. While not shown in the figure, the approach can be calibrated to either automatically include or exclude missing values. This represents a further contribution to flexibility and ease of use.
- There is a potential unfolding of asymmetric trees with different subpartitions in descendent nodes.
- Descendent nodes result in the identification of local effects that are conditional on the fields that form the partition sequence that are used to identify the effect. These local effects are conditional on the interactions among the partitioning fields and are sometimes referred to as 'interaction effects'.
- Partitioning fields may be at nominal, ordinal, or interval measurement levels. In the case of ordinal or interval measures, the partitioned values are grouped together so as to maximally discriminate among high and low percentages in the resulting target field proportions in the descendent nodes.
- While not shown in the illustration, we note that branch partitions may be two-way or multiway branches.

OTHER FEATURES

As with all quantitative models, the form of the tree has to be limited by considerations of reproducibility and generalization. This has led to the development of various stopping criteria to limit the growth of the tree on one hand and various test and validation approaches that increase the likely accuracy and reliability of tree models in post-training applications. A concise summary is presented in an early paper by Kass.[4] The flexibility of the single decision tree approach described here has proven to be well adapted to multitree models, based on a variety of resampling and multisample methods, a development that has radically improved the practicality and overall observed performance of decision trees in a wide variety of model settings.

ORIGINS

All modern versions of decision trees trace their origins to work carried out by Belson in the 1950s, especially his work in the analysis of nation-wide audience surveys on behalf of the British Broadcasting Corporation.[5] This work was originally undertaken prior to the introduction of digital computers and

 © 2013 Wiley Periodicals, Inc.

exploited the then most current technology of mechanical calculators.

Decision trees turn out to be well adapted to mechanical calculators using Hollerith punch cards because of the sorting and selection characteristics of the algorithm and the avoidance of, e.g., any matrix-based computations. For each predictive field that could be considered as an input for use in the characterization of a target field it was possible to sort subclasses formed for each target-predictor combination and then to identify imbalances between the expected frequency of the subclass and the observed frequency of the subclass. This step-by-step recursive process is simple enough for both mechanical calculators and unassisted humans. Unbalanced distributions—which we would now identify as distributions with high chi-squared values—could be easily identified with the tabulating machines available at this time. This method—so useful in the era prior to digital computers—survives to the current day as the underpinning for all decision tree implementations.

A further refinement introduced by Belson involved the differential assessment of nested subclass predictors.[6] Belson recognized that descendent nodes of a tree could be examined recursively, just as the top node had been. Belson further recognized that descendent nodes could be subset by either the same predictor or another predictor such that the descendent nodes of the tree could be balanced and symmetrical—employing a matching set of predictors with each level of the subtree—or could be unbalanced in that subnode partitions could be based on the most powerful predictor at a given level of the subtree. This innovation exploits the power of decision trees to explore and discover a host of subregion effects in data and, like the use of predictors identified on the basis of deviation from expected values, forms the basis of modern decision trees.

Morgan and Sonquist[7] built on Belson's early work and saw decision trees as a complement and alternative to regression to analyze survey data. Initially, Morgan and Sonquist began with the notion of employing trees in order to identify interaction terms that would be useful in forming the most effective regression solution for their data modeling tasks. In tests run by Morgan and Sonquist, they observed a decision tree which partitioned data into 21 groups that accounted for two-thirds of variance of the response variable. A similar regression with 30 terms, including interaction terms, was only able to account for 36% of the variance in the response. The authors reached three conclusions: (1) that interactions among inputs are inevitable; (2) that regression requires the analyst to specify interactions in advance; and (3) that decision trees were better tools because they find the interactions as they grow the tree.

Many observers at the time were resistant to employ the relatively new and lightly tested approach advocated by Morgan and Sonquist. Regression practitioners then—and now—develop results on the basis of well-informed theory and widely tested results in a broad, active, and well-informed community. The theoretical underpinnings—coupled with a rich history of fielded results—enable regression practitioners to develop time-tested, effective metrics and diagnostics in a wide range of circumstances.

Decision trees were demonstrated to have shortcomings of their own: how to go about selecting appropriate variables to form the tree partitions (input vetting and selection) and how many partitions, of what complexity, to build. These latter two problems served as the 'grist for the mill' of the next steps in the development of statistical decision trees carried out by Kass and Hawkins[8] and Breiman et al.,[9] respectively. Over time, this body of work has provided substantial credibility and a rich legacy of fielded applications that help establish trees as a useful, viable, and trustworthy technique.

RULE INDUCTION, MACHINE LEARNING, AND DECISION TREES

During the 1950s, as Belson was developing his approach, a kind of computation which he described as based '... on the principal of biological classification', other researchers in experimental psychology were attempting to encode human approaches to concept formation tasks. Both approaches naturally fed into the nascent field of artificial intelligence and machine learning. In this way Belson's work serves as a precursor to a new line of decision tree development that employs machine algorithms to produce executable rules.

The work in experimental psychology led to the development of a computer implementation, entitled 'CLS' (for Concept Learning System) developed by Hunt et al.[10] As in the earlier approaches of Belson and Morgan and Sonquist, CLS works through the successive application of partitions in the data based on highly discriminating variables or inputs. J. Ross Quinlan entered this field from a machine learning perspective. He formalized the development of this approach to concept formation as a method of knowledge acquisition. This resulted in the development of 'Interactive Dichotomizer 3' (ID3).[11]

Follow-ons to Quinlan's initial work have led to the development of a number of rule generation

 © 2013 Wiley Periodicals, Inc.

approaches for knowledge acquisition, commonly referred to as 'rule induction'.

BOX 1

Donald Michie served as the editor of a set of findings that featured Quinlan's initial work on ID3. Michie was a colleague of Alan Turing during the World War II Enigma Project and is a founding father of the field of artificial intelligence. He later employed inductive rules to the adaptive control of robotic devices and spacecraft.[12] This rule method serves as a template for self-learning robotic systems up to the present day.

Subsequent work by Quinlan led to the development of C4.5.[13]

Rule induction is an active area of development and has led to a range of rule induction approaches, for example, W Cohen's 'RIPPER'.[14] RIPPER incorporates a multitree approach often described as 'sequential covering'. In these approaches the tree is first grown so that a pure node is found. A pure node is a node that results from the identification of a rule that predicts 100% of the target values. The preconditions of the rule 'covers' the training observations that correspond to this rule. The observations that are covered by the rule are then removed from the training data (i.e. are 'ripped' out). Successive trees are run, at each step looking for a rule that produces a 'pure node'. Multiple trees may be grown until no more pure nodes are found. Overall, the predictive space is 'covered' through the layering of these successively grown predictive rules. The RIPPER algorithm is a greedy algorithm; i.e. it produces excessively overoptimistic results that do not generalize well. Alternative multitree approaches, discussed below, are less greedy and offer superior generalization performance. Another innovation suggested by Cohen was to form rules based on *both* the presence and absence of attributes (allow Boolean NOTs to form part of the selection expression). This approach has more recently been implemented as part of a text mining solution to generate automatic text classification rules based on inductive rule learning.[15]

CURRENT DEVELOPMENTS (MULTIPLE TREES)

The bootstrap method, described by Efron,[16] is a prominent example of the utility of resampling in statistical computation. The single tree approach—one of selecting the best single predictor at any one stage in the growth of the decision tree—can be extended by resampling the available training data. This random element has many benefits: the most obvious benefit is the smoothing properties. While a single decision tree bisects the space of training data into a number of hard-edge rectangles, multitrees form many overlapping bisections so that the fitted space more closely approximates such methods as neural networks and multiple regression. With multiple trees we can derive multiple, overlapping viewpoints that are different but complementary. When taken together, the overlapping views reduce both variance and bias.

The resampling approach has led to a number of methods to 'boost' the predictive power of the host training set. Multiple trees are always grown, regardless of the specific method that is employed. In addition to the introduction of random components in multiple trees, these approaches also offer the opportunity to reweight computations in successive iterations of tree growth. Unlike the 'sequential covering' approach, described above, where successive samples are drawn from the training corpus in unaltered form, boosting approaches reweight cases in successive iterations. The coverage offered by these approaches is less structured and deterministic than sequential covering. In this approach, the reweighting goal is to alter successive training samples with the view to improving the predictive performance of successive rule sets. These approaches have been explored and advocated by Schapire;[17] notably in Adaboost developed by Freund and Schapire;[18] Arcing by Breiman;[19] and Gradient Boosting by Friedman.[20] The Adaboost method (from 'adaptive boosting') employs an approach that reweights individual observations in subsequent samples. In Gradient Boosting, the target value is adjusted by a function of the residual of the training value minus the predicted value.

Various group-voting or aggregation methods are possible in the production of a final group-voting metric: including numeric averaging with continuous outcomes and majority votes or polling with categorical outcomes.

The interaction between the fields of statistical decision trees and machine learning continued throughout these adaptations of bootstrapping applications to multiple trees. One innovation included sampling and randomization across both rows and columns of the training data. This technique entered the machine learning field in the application of multiple decision trees to digit recognition as described in Amit and Geman.[21] Much of this cross-fertilization is due to substantial cross-disciplinary work carried

 © 2013 Wiley Periodicals, Inc.

out by Breiman. He described this general row and column sampling approach as 'Random Forests';[22] these are currently the leading benchmark implementation of decision trees across a variety of statistical and machine learning applications.

CONCLUSIONS

There are many variations of multitree themes: autonomous vs serial samples; row vs column reweighting schemes; replacement samples vs no replacement; and so on. Improvements over best-guess, single decision trees are shown in most multitree methods. As training data continues to increase in size there are now obvious benefits in the approach of multiple autonomous trees as these trees can be calculated independently, in parallel, prior to the production of an aggregate effect. As the size of initial training data has increased, so too has the corresponding emphasis on sampling *without* replacement. With larger training data, sampling without replacement tends to reinforce the adoption of differences in the model results. This is now recognized as a potential strength of multitree methods.

To date, most multitree methods demonstrate strengths in various circumstances. As this field evolves it may become clear which method is best in which set of circumstances. Given the pace of innovation in this area it is likely that improved methods and new paradigms will continue to emerge.

NOTE

[a] This data table is based on the Titanic Passenger List edited by Michael A. Findlay, originally published in Ref[23], and expanded with the help of the internet community. The original HTML files were obtained by Philip Hind (1999).

REFERENCES

1. Lima M. *Visual Complexity: Mapping Patterns of Information*. New York: Princeton Architectural Press; 2011, 28.
2. http://lib.stat.cmu.edu/S/Harrell/data/descriptions/titanic.html. (Accessed September 23, 2013).
3. Sonquist JA, Baker EL, Morgan JN. *Searching for Structure*. Ann Arbor, MI: Institute for Social Research; 1973.
4. Kass GV. An exploratory technique for investigating large quantities of categorical data. *J R Stat Soc* 1980, 29:119–127.
5. Belson WA. A technique for studying the effects of television broadcast. *J R Stat Soc* 1956, 5:195.
6. Belson WA. Matching and prediction on the principle of biological classification. *J R Stat Soc* 1959, 8:65–75.
7. Morgan JN, Sonquist JA. Problems in the analysis of survey data, and a proposal. *J Am Stat Assoc* 1963, 58:415–435.
8. Hawkins DM, Kass GV. Automatic interaction detection. In: Hawkins DM, ed. *Topics in Applied Multivariate Analysis*. Cambridge: Cambridge University Press; 1982.
9. Breiman L, Friedman JH, Olshen RA, Stone CJ. *Classification and Regression Trees*. London: Chapman and Hall; 1984.
10. Hunt E, Marin J, Stone P. *Experiments in Induction*. New York: Academic Press; 1966.
11. Quinlan JR. Discovering rules by induction from large collections of examples. In: Michie D, ed. *Expert Systems in the Micro-electronic Age*. Edinburgh: Edinburgh University Press; 1979, 168–201.
12. Michie D, Sammut C. Controlling a black-box simulation of a spacecraft. *AI Mag* 1991, 12:56–63.
13. Quinlan JR. *C4.5: Programs for Machine Learning*. New York: Morgan Kaufmann; 1988.
14. Cohen, WW. Fast effective rule induction. Proceedings of the Twelfth International Conference on Machine Learning; 1995, 115–123.
15. Automatic Boolean rule generation. Available at: http://www.sas.com/text-analytics/text-miner/index.html. (Accessed March 25, 2013).
16. Efron B. Bootstrap methods: another look at the Jackknife. *Ann Stat* 1979, 7:1–26.
17. Schapire RE. The strength of weak learnability. *Mach Learn* 1990, 5:197–227.
18. Freund Y. Schapire RE. Experiments with a new boosting algorithm. Proceedings of the Thirteenth International Conference on Machine Learning, Bari, Italy; 1996, 148–156.
19. Brieman L. Arcing classifiers. *Ann Stat* 1998, 26:801–849.
20. Friedman, HJ. Stochastic gradient boosting. 1999. Available at: http://www-stat.stanford.edu/~jhf/ftp/stobst.ps.
21. Amit Y, Geman D. Shape quantization and recognition with randomized trees. *Neural Comput* 1997, 9:1545–1588.

 © 2013 Wiley Periodicals, Inc.

22. Breiman L. Random forests, 2001. Available at http://oz.berkeley.edu/~breiman/randomforest2001.pdf. (Accessed September 23, 2013).

23. Eaton JP, Haas CA. *Titanic: Triumph and Tragedy, Second Edition*. New York: W.W. Norton & Company Inc; 1995.

FURTHER READING

Hawkins DM. Recursive partitioning. *WIREs Comput Stat* 2009, 1:290–295.

Loh WY. Classification and regression trees. *WIREs Data Mining Knowl Discov* 2011, 1:14–23.

de Ville B, Neville P. *Decision Trees for Analytics Using SAS Enterprise Miner*. Cary, NC: SAS Press; 2013.

 © 2013 Wiley Periodicals, Inc.