Tree Structured Data Analysis: AID, CHAID and CART

Leland Wilkinson

SPSS Inc., 233 South Wacker, Chicago, IL 60606

Department of Statistics, Northwestern University, Evanston, IL 60201

email: leland@spss.com

KEY WORDS: recursive partitioning, classification and regression trees, CART, CHAID

Abstract

Classification and regression trees are becoming increasingly popular for partitioning data and identifying local structure in small and large datasets. Classification trees include those models in which the dependent variable (the predicted variable) is categorical. Regression trees include those in which it is continuous. This paper discusses pitfalls in the use of these methods and highlights where they are especially suitable.

Paper presented at the 1992 Sun Valley, ID, Sawtooth/SYSTAT Joint Software Conference.

1 Introduction

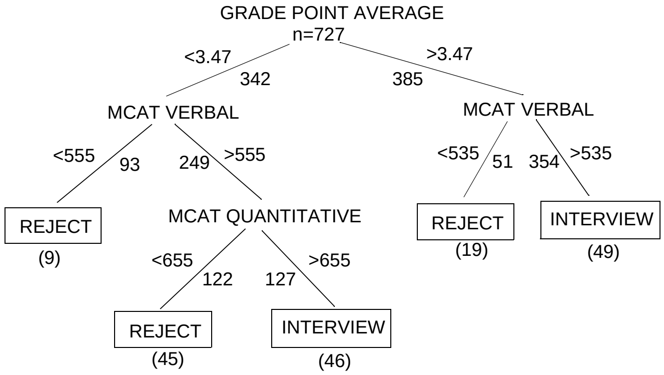
Trees are connected acyclic graphs. They are fundamental to computer science (data structures), biology (classification), psychology (decision theory), and many other fields. Recursive partitioning trees are tree-based models used for prediction. In the last two decades, they have become popular as alternatives to regression, discriminant analysis, and other procedures based on algebraic models. Tree fitting methods have become so popular that several commercial programs now compete for the attention of market researchers and others looking for software. Different commercial programs produce different results with the same data, however. Worse, some programs provide no documentation or supporting materials to explain their algorithms. The result is a marketplace of competing claims, jargon, and misrepresentation. Reviews of these packages (e.g. Levine, 1991; Simon, 1991) have used words like "sorcerer," "magic formula," and "wizardry" to describe the algorithms and have expressed frustration at vendors' scant documentation. Some vendors, in turn, have represented tree programs as state-of-the-art "artificial intelligence" procedures capable of discovering hidden relationships and structures in databases. Despite the marketing hyperbole, many of the now popular tree fitting algorithms have been around for decades. Warnings of abuse of these techniques are not new either (e.g. Einhorn, 1972; Bishop, Fienberg, and Holland, 1975). Originally proposed as automatic procedures for detecting interactions among variables, tree fitting methods are actually closely related to classical cluster analysis (Hartigan, 1975). This paper will attempt to sort out some of the differences between algorithms and illustrate their use on real data. In addition, tree analyses will be compared to discriminant analysis and regression.

2 Tree models

Figure 1 shows a tree for predicting decisions by a medical school admissions committee (Milstein et al., 1975). It was based on data for a sample of 727 applicants. We selected a tree procedure for this analysis because it was easy to present the results to the Yale Medical School admissions committee and because the tree model could serve as a basis for structuring their discussions about admissions policy.

Notice that the values of the predicted variable (admissions decision to reject or interview) are at the bottom of the tree and the predictors (Medical College Admissions Test and College Grades) come into the system at each node of the tree. The top node contains the entire sample. Each of the remaining nodes contains a subset of the sample in the node directly above it. Furthermore, any node contains the sum of the samples in the nodes connected to and directly below it. The tree thus splits samples. Each node can be thought of as a cluster of objects (cases) which is to be split by further branches in the tree. The numbers in parentheses below the terminal nodes show how many cases are incorrectly classified by the tree. A similar tree data structure is used for representing the results of single and complete linkage and other forms of hierarchical cluster analysis (Hartigan, 1975). Tree prediction models add two ingredients: the predictor and predicted variables labeling the nodes and branches.

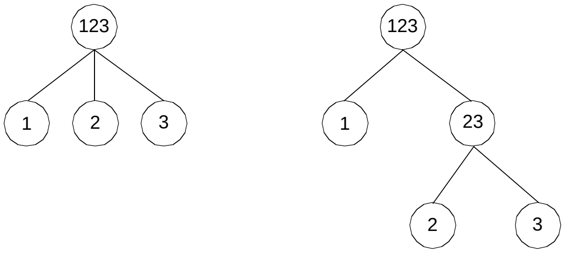
Figure 1. Tree for predicting admissions to medical school.



2.1 Binary vs. general trees

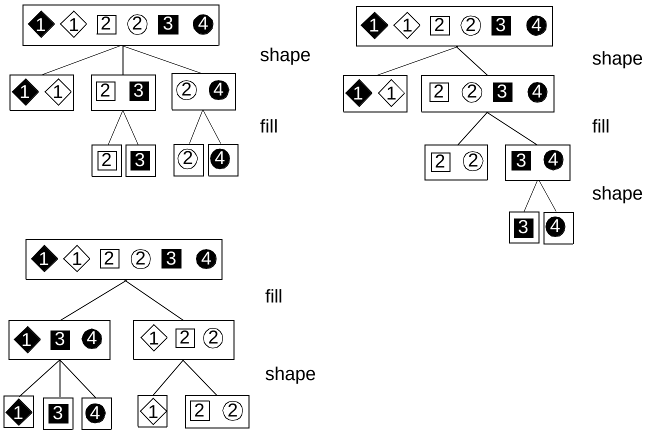
The tree in Figure 1 is binary because each node is split into only two subsamples. Classification or regression trees need not be binary, but most are. Despite the marketing claims of some vendors, nonbinary, or multi-branch trees are not intrinsically superior to binary trees. Each is a permutation of the other. Figure 2 shows this. The tree on the left in Figure 2 is not more parsimonious than that on the right. Both trees have the same number of parameters (split points), and any statistics associated with the tree on the left can be converted trivially to fit the one on the right. A computer program for scoring either tree (IF ... THEN ... ELSE) would look identical. For display purposes, it is often convenient to collapse binary trees into multi-branch trees, but this is not necessary.

Figure 2. Ternary (left) and binary (right) trees.



Some programs which do multi-branch splits do not allow further splitting on a predictor once it has been used. This has an appealing simplicity but it can lead to unparsimonious trees. Figure 3 shows an example of this problem. The upper right tree classifies objects on an attribute by splitting once on shape, then fill, then again on shape. This allows the algorithm to separate the objects into only four terminal nodes having common values. The upper left tree splits on shape, then only on fill. By not allowing any other splits on shape, the tree requires five terminal nodes to classify correctly. This problem cannot be solved by splitting first on fill, as the lower left tree shows. In general, restricting splits to only one branch for each predictor results in more terminal nodes.

Figure 3. Multi-branch trees (left) and binary tree (right)



2.2 Categorical vs. quantitative predictors

The predictor variables in Figure 1 are quantitative, so splits are created by determining cut points on a scale. If predictor variables are categorical as in Figure 3, splits are made between categorical values. It is not necessary to categorize predictors before computing trees. This is as dubious a practice as recoding data well-suited for regression into categories in order to use chi-square tests. Those who recommend this practice are turning silk purses into sows' ears. In fact, if variables are categorized before doing tree computations, then poorer fits are likely to result. Algorithms are available for mixed quantitative and categorical predictors, analogous to analysis of covariance.

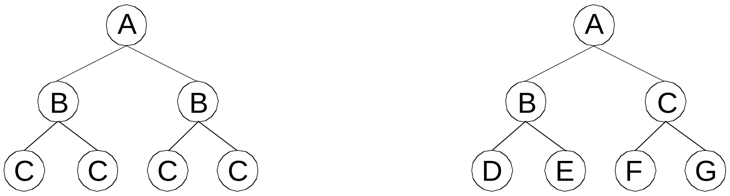
2.3 Regression trees

Morgan and Sonquist (1963) proposed a simple method for fitting trees to predict a quantitative variable. They called the method AID, for Automatic Interaction Detection. The algorithm performs stepwise splitting. It begins with a single cluster of cases and searches a candidate set of predictor variables for a way to split this cluster into two clusters. Each predictor is tested for splitting as follows: sort all the n cases on the predictor and examine all n-1 ways to split the cluster in two. For each possible split, compute the within-cluster sum of squares about the mean of the cluster on the dependent variable. Choose the best of the n-1 splits to represent the predictor's contribution. Now do this for every other predictor. For the actual split, choose the predictor and its cut point which yields the smallest overall within-cluster sum of squares.

Categorical predictors require a different approach. Since categories are unordered, all possible splits between categories must be considered. For deciding on one split of k categories into two groups, this means that 2^k-1 possible splits must be considered. Once a split is found, its suitability is measured on the same within-cluster sum of squares as for a quantitative predictor. Morgan and Sonquist called their algorithm AID because it naturally incorporates interaction among predictors. Interaction is not correlation. It has to do instead with conditional discrepancies. In the analysis of variance, interaction means that a trend within one level of a variable is not parallel to a trend within another level of the same variable. In the ANOVA model, interaction is represented by cross-products between predictors. In the tree model, it is represented by branches from the same node which have different splitting predictors further down the tree.

Figure 4 shows a tree without interactions on the left and with interactions on the right. Because interaction trees are a natural byproduct of the AID splitting algorithm, Morgan and Sonquist called the procedure “automatic.” In fact, AID trees without interactions are quite rare for real data, so the procedure is indeed automatic. To search for interactions using stepwise regression or ANOVA linear modeling, we would have to generate 2^p interactions among p predictors and compute partial correlations for every one of them in order to decide which ones to include in our formal model.

Figure 4. No interaction (left) and interaction (right) trees.



2.4 Classification trees

Regression trees parallel regression/ANOVA modeling, in which the dependent variable is quantitative. Classification trees parallel discriminant analysis and algebraic classification methods. Kass (1980) proposed a modification to AID called CHAID for categorized dependent and independent variables. His algorithm incorporated a sequential merge and split procedure based on a chi-square test statistic. Kass was concerned about computation time (although this has since proved an unnecessary worry), so he decided to settle for a sub-optimal split on each predictor instead of searching for all possible combinations of the categories. Kass’s algorithm is like sequential cross-tabulation. For each predictor:

1) cross tabulate the m categories of the predictor with the k categories of the dependent variable,

2) find the pair of categories of the predictor whose 2xk sub-table is least significantly different on a chi-square test and merge these two categories;

3) if the chi-square test statistic is not “significant” according to a preset critical value, repeat this merging process for the selected predictor until no non-significant chi-square is found for a sub-table, and

4) pick the predictor variable whose chi-square is largest and split the sample into $m \leq l$ subsets, where l is the number of categories resulting from the merging process on that predictor;

5) continue splitting, as with AID, until no “significant” chi-squares result.

The CHAID algorithm saves some computer time, but it is not guaranteed to find the splits which predict best at a given step. Only by searching all possible category subsets can we do that. CHAID is also limited to categorical predictors, so it cannot be used for quantitative or mixed categorical-quantitative models, as in Figure 1. Nevertheless, it is an effective way to search heuristically through rather large tables quickly.

Within the computer science community there is a categorical splitting literature which often does not cite the statistical work and is, in turn, not frequently cited by statisticians (although this has changed in recent years). Quinlan (1986, 1992), the best known of these researchers, developed a set of algorithms based on information theory. These methods, termed ID3, iteratively build decision trees based on training samples of attributes.

2.5 Stopping rules, pruning, and cross-validation

AID, CHAID, and other forward sequential tree fitting methods share a problem with other tree clustering methods - where do we stop? If we keep splitting, a tree will end up with only one case or object at each terminal node. We need a method for producing a smaller tree than the exhaustive one. One way is to use stepwise statistical tests, as in the *F*-to-enter or alpha-to-enter rule for forward stepwise regression. We compute a test statistic (chi-square, *F*, etc.), choose a critical level for the test (sometimes modifying it with the Bonferroni inequality), and stop splitting any branch which fails to meet the test (see Wilkinson, 1979, for a review of this procedure in forward selection regression).

Breiman et al. (1984) showed that this method tends to yield trees with too many branches and can also fail to pursue branches which can add significantly to the overall fit. They advocate, instead, pruning the tree. After computing an exhaustive tree, their program eliminates nodes which do not contribute to the overall prediction. They add another essential ingredient, however: the cost of complexity. This measure is similar to other cost statistics, such as Mallows' C_p (see Neter, Wasserman, and Kutner, 1985), which add a penalty for increasing the number of parameters in a model. Breiman's method is *not* like backward elimination stepwise regression. It resembles instead forward stepwise regression with a cutting back on the final number of steps using a different criterion than the *F*-to-enter. This method still cannot do as well as an exhaustive search, which would be prohibitive for most practical problems.

Regardless of how a tree is pruned, it is important to cross validate it. As with stepwise regression, the prediction error for a tree applied to a new sample can be considerably higher than for the training sample on which it was constructed. Whenever possible, data should be reserved for cross validation.

2.6 Loss functions

Different loss functions, based on the predicted variable, are appropriate for different forms of data. For regression trees, typical loss functions are *least squares* of deviations from mean of a subgroup at a node, *trimmed least squares* of deviations from trimmed mean, and *least absolute deviations* from the mean. Least squares loss yields the classic AID tree. At each split, cases are classified so that the within-group sum of squares about the mean of the group is a small as possible. The trimmed mean loss works the same way, but first trims a percentage of outlying cases (e.g., 10% at each extreme) in a splittable subset before computing the mean and sum of squares. It can be useful when you expect outliers in subgroups and don't want them to influence the split decisions. Least absolute deviations loss computes the sum of absolute deviations about the mean rather than squares. It, too, gives less weight to extreme cases in each potential group.

For classification trees, typical loss functions are the *phi* coefficient, *Gini* index, and *twoing*. The phi coefficient is χ^2/n for a 2 x *k* table formed by the split on *k* categories of the dependent variable. The Gini index is a variance estimate based on all comparisons of possible pairs of values in a subgroup. Finally, twoing is a word coined by Breiman et al. to describe splitting *k* categories as if it were a 2 category splitting problem. For more information on the effects of Gini and twoing on computations, see Breiman et al. (1984).

2.7 Geometry

Most discussions of trees versus other classifiers compare tree graphs and algebraic equations. There is another graphic view of what a tree classifier does, however. If we look at the cases embedded in the space of the predictor variables, we can ask how a linear discriminant analysis partitions the cases and how a tree classifier partitions them.

Figure 5 shows how cases are split by a multiple linear discriminant analysis. There are three predictors (X, Y, Z) and four subgroups of cases (black, shaded, white, hidden) in this example. The fourth group is assumed to be under the bottom plane in the figure. The cutting planes are positioned roughly half-way between each pair of group centroids. Their orientation is determined by the discriminant analysis. With three predictors and four groups, there are six cutting planes, although only four planes show in the figure. In general, if there are *k* groups, the linear discriminant model cuts them with *k*(*k*-1)/2 planes.

Figure 5. Cutting planes for discriminant model.

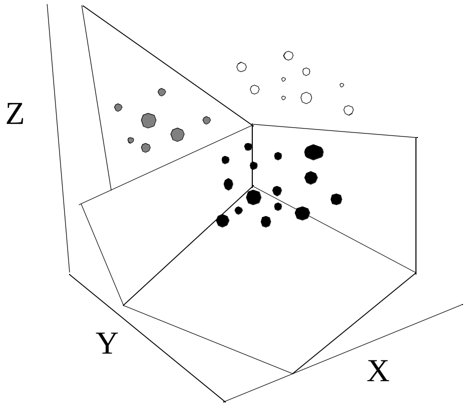
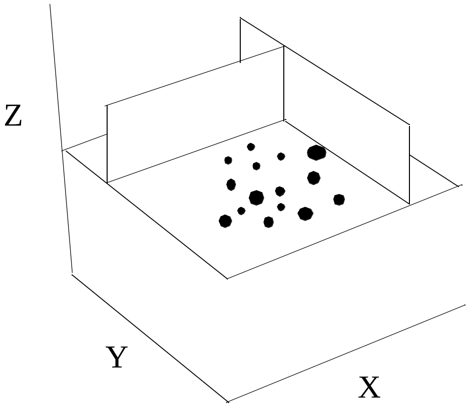


Figure 6 shows how a tree fitting algorithm cuts the same data. Only the nearest subgroup (dark spots) shows; the other three groups are hidden behind the rear and bottom cutting planes. Notice that the cutting planes are parallel to the axes. While this would seem to restrict the discrimination compared to the more flexible angles allowed the discriminant planes, the tree model allows interactions between variables, which do not appear in the ordinary linear discriminant model. Notice, for example, that one plane splits on the X variable, but the second plane that splits on the Y variable cuts only the values to the left of the X partition. The tree model can continue to cut any of these sub-regions separately, unlike the discriminant model, which may cut only globally and only with $k(k-1)/2$ planes. This is a mixed blessing, however, since tree methods, as we have seen, can over-fit the data. It is critical to test them on new samples.

Tree models are not usually accompanied by variable-space plots of this sort, but it is helpful to see that they have a geometric interpretation. Alternatively, we can construct algebraic expressions for trees. They would require dummy variables for any categorical predictors and interaction (product) terms for every split whose descendants (lower nodes) did not involve the same variables on both sides.

Figure 6. Cutting planes for tree model.



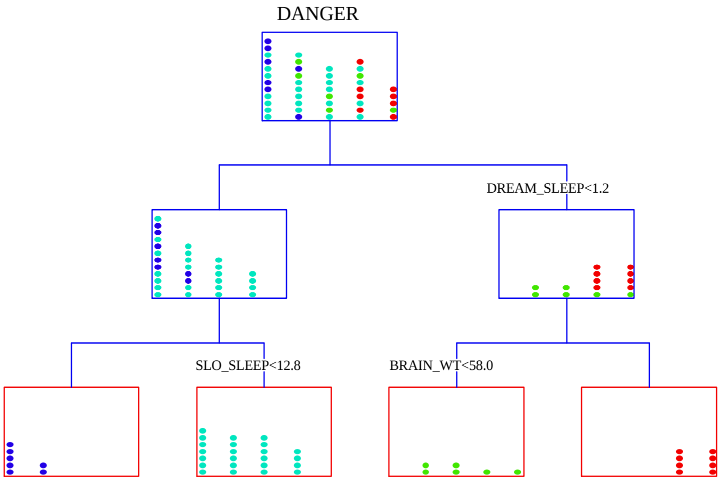
3 Tree displays

There are numerous ways to present the results of a classification or regression tree analysis. Graphical tree displays are among the most useful, because they allow navigation through the entire tree as well as drill-down to individual nodes.

Figure 7 shows a classification tree analysis to predict the danger of a mammal being eaten by predators. The data are from Allison and Cicchetti, 1976. The predictors are hours of dreaming and non-dreaming sleep, gestational age, and body and brain weight. Although the danger index has only five values, we are treating it as a quantitative variable with meaningful numerical values. The tree predicts this danger index (1=unlikely to be killed by a predator, 5=likely to be killed) from type of sleep (slow wave sleep and dreaming sleep) and body and brain weight. In each frame node of the tree is a dot density. The advantage of dot densities in this context is that they work well for both continuous and categorical variables. Unlike ordinary histograms, dot densities have one stack of dots per category because they bin only where the data are.

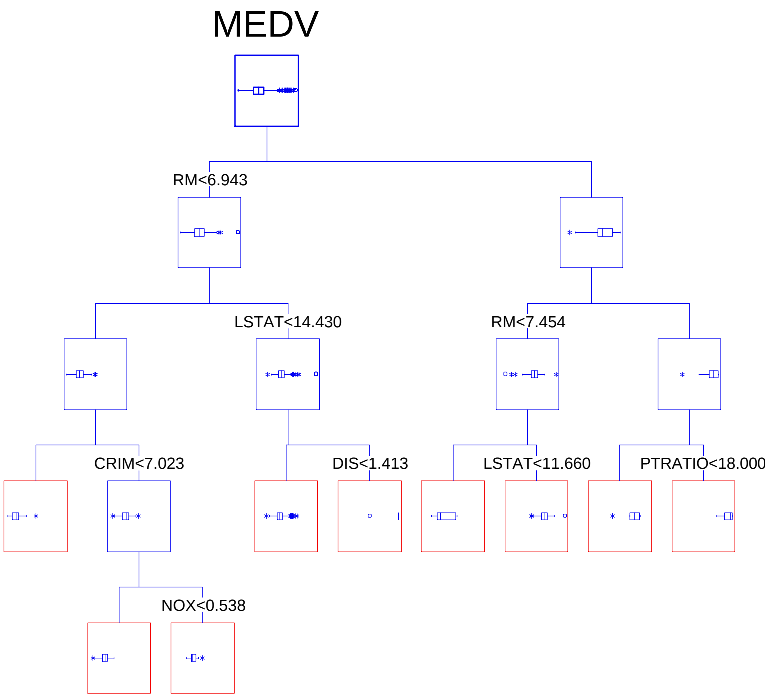
This tree is called a *mobile*. This display format gets its name from the hanging sculptures created by Calder and other artists. If the squares were boxes, the dots marbles, the horizontal branches metal rods, and the vertical lines wires, the physical model would hang in a plane as shown in the figure. This graphical balancing format helps identify outlying splits in which only a few cases are separated from numerous others. Each box contains a dot density based on a proper subset of its parent's collection of dots. The scale at the bottom of each box is the danger index running from 1 (left) to 5 (right). Each dot is colored according to its terminal node at the bottom of the tree so that the distribution of predicted values can be recognized in the mixtures higher up in the tree.

Figure 7. Classification tree with dot histograms.



Other graphics can be inserted into the nodes of a tree. Figure 8 shows a mobile containing Tukey box plots based on a simple AID model. The dataset are Boston housing prices, cited in Belsley, Kuh and Welsch (1980) and used in Breiman et al. (1984). We are predicting median home values (MEDV) from a set of demographic variables. The scale at the bottom of each rectangle is median home value on a standardized range.

Figure 8. Predicting median housing prices for Boston housing data.



4 Conclusion

Classification and regression trees offer a non-algebraic method for partitioning data that lends itself to graphical displays. As with any statistical method, there are pitfalls involved in their use. Commercial assurances that trees are robust, automatic engines for finding patterns in small and large datasets should be distrusted. Those who understand the basics of recursive partitioning trees are in a better position to recognize when they are useful and when they are not.

References

Allison, T. and Cicchetti, D. (1976). Sleep in mammals: Ecological and constitutional correlates. *Science, 194,* 732-734.

Belsley, D.A., Kuh, E., and Welsch, R.E. (1980). *Regression diagnostics: Identifying influential data and sources of collinearity.* New York: John Wiley & Sons.

Bishop, Y.M., Fienberg, S.E., and Holland, P. W. (1975). *Discrete Multivariate Analysis*. Cambridge, MA: MIT Press.

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

Einhorn, H. (1972). Alchemy in the behavioral sciences. *Public Opinion Quarterly, 3,* 367-378.

Hartigan, J.A. (1975). *Clustering Algorithms*. New York: John Wiley & Sons.

Kass, G.V. (1980). An exploratory technique for investigatin large quantities of categorical data. *Applied Statistics, 29,* 119-127.

Levine, M. (1991). Statistical analysis for the executive. *Byte*, *17*, 183-184.

Milstein, R.M., Burrow, G.N., Wilkinson, L., and Kessen, W. (1975). Prediction of Screening Decisions in a medical school admission process. *Journal of Medical Education, 51,* 626-633.

Morgan, J.N. and Sonquist, J.A. (1963). Problems in the analysis of survey data, and a proposal. *Journal of the American Statistical Association*, *58*, 415-434.

Neter, J., Wasserman, W., and Kutner, M. (1985). *Applied Linear Statistical Models,* 2nd Ed. Homewood, IL: Richard E. Irwin, Inc.

Quinlan, J.R. (1986). Induction of decision trees. *Machine Learning*, *1*, 81-106.

Quinlan, J.R. (1992). *C4.5: Programs for Machine Learning*. New York: Morgan Kaufmann.

Simon, B. (1991). Knowledge Seeker: Statistics for decision makers. *PC Magazine*, January 29, 50.

Wilkinson, L. (1979). Tests of significance in stepwise regression. *Psychological Bulletin, 86,* 168-174