1 CaRT

1.1 Overview

CaRT, a recursive partitioning method, builds classification and regression trees for predicting continuous dependent variables (regression) and categorical predictor variables (classification). The classic C&RT algorithm was popularized by Breiman et al.

1.2 Classification and Regression Problems

- There are numerous algorithms for predicting continuous variables or categorical variables from a set of continuous predictors and/or categorical factor effects. For example, in GLM (General Linear Models) and GRM (General Regression Models), we can specify a linear combination (design) of continuous predictors and categorical factor effects (e.g., with two-way and three-way interaction effects) to predict a continuous dependent variable.
- In GDA (General Discriminant Function Analysis), we can specify such designs for predicting categorical variables, i.e., to solve classification problems.

1.3 Regression-type problems

- Regression-type problems are generally those where we attempt to predict the values of a continuous variable from one or more continuous and/or categorical predictor variables. For example, we may want to predict the selling prices of single family homes (a continuous dependent variable) from various other continuous predictors (e.g., square footage) as well as categorical predictors (e.g., style of home, such as ranch, two-story, etc.; zip code or telephone area code where the property is located, etc.; note that this latter variable would be categorical in nature, even though it would contain numeric values or codes).
- If we used simple multiple regression, or some general linear model (GLM) to predict the selling prices of single family homes, we would determine a linear equation for these variables that can be used to compute predicted selling prices.

- There are many different analytic procedures for fitting linear models (GLM, GRM, Regression), various types of nonlinear models (e.g., Generalized Linear/Nonlinear Models (GLZ), Generalized Additive Models (GAM), etc.), or completely custom-defined nonlinear models (see Nonlinear Estimation), where we can type in an arbitrary equation containing parameters to be estimated. CHAID also analyzes regression-type problems, and produces results that are similar (in nature) to those computed by C&RT. Note that various neural network architectures are also applicable to solve regression-type problems.

1.4 Classification-type problem

- Classification-type problems are generally those where we attempt to predict values of a categorical dependent variable (class, group membership, etc.) from one or more continuous and/or categorical predictor variables. For example, we may be interested in predicting who will or will not graduate from college, or who will or will not renew a subscription.

- These would be examples of simple binary classification problems, where the categorical dependent variable can only assume two distinct and mutually exclusive values. In other cases, we might be interested in predicting which one of multiple different alternative consumer products (e.g., makes of cars) a person decides to purchase, or which type of failure occurs with different types of engines.

- In those cases there are multiple categories or classes for the categorical dependent variable. There are a number of methods for analyzing classification-type problems and to compute predicted classifications, either from simple continuous predictors (e.g., binomial or multinomial logit regression in GLZ), from categorical predictors (e.g., Log-Linear analysis of multi-way frequency tables), or both (e.g., via ANCOVA-like designs in GLZ or GDA).

- The CHAID also analyzes classification-type problems, and produces results that are similar (in nature) to those computed by C&RT. Note that various neural network architectures are also applicable to solve classification-type problems.

1.5 Classification and Regression Trees (CaRT)

In most general terms, the purpose of the analyses via tree-building algorithms is to determine a set of if-then logical (split) conditions that permit accurate prediction or classification of cases.

1.5.1 Classification Trees

- For example, consider the widely referenced Iris data classification problem introduced by Fisher [1936; see also Discriminant Function Analysis and General Discriminant Analysis (GDA)]. The data file Irisdat reports the lengths and widths of sepals and petals of three types of irises (Setosa, Versicol, and Virginic).

- The purpose of the analysis is to learn how we can discriminate between the three types of flowers, based on the four measures of width and length of petals and sepals. Discriminant function analysis will estimate several linear combinations of predictor variables for computing classification scores (or probabilities) that allow the user to determine the predicted classification for each observation.

- A classification tree will determine a set of logical if-then conditions (instead of linear equations) for predicting or classifying cases instead:

- The interpretation of this tree is straightforward: If the petal width is less than or equal to 0.8, the respective flower would be classified as Setosa; if the petal width is greater than 0.8 and less than or equal to 1.75, then the respective flower would be classified as Virginic; else, it belongs to class Versicol.

1.6 Regression Trees

The general approach to derive predictions from few simple if-then conditions can be applied to regression problems as well. This example is based on the data file Poverty, which contains 1960 and 1970 Census figures for a random selection of 30 counties. The research question (for that example) was to determine the correlates of poverty, that is, the variables that best predict the percent of families below the poverty line in a county. A reanalysis of those data, using the regression tree analysis [and v-fold cross-validation, yields the following results:

Again, the interpretation of these results is rather straightforward: Counties where the percent of households with a phone is greater than 72% have generally a lower poverty rate. The greatest poverty rate is evident in those counties that show less than (or equal to) 72% of households with a phone, and where the population change (from the 1960 census to the 170 census) is less than -8.3 (minus 8.3). These results are straightforward, easily presented, and intuitively clear as well: There are some affluent counties (where most households have a telephone), and those generally have little poverty. Then there are counties that are generally less affluent, and among those the ones that shrunk most showed the greatest poverty rate. A quick review of the scatterplot of observed vs. predicted values shows how the discrimination between the latter two groups is particularly well "explained" by the tree model.

1.7 Advantages of Classification and Regression Trees (CaRT) Methods

- As mentioned earlier, there are a large number of methods that an analyst can choose from when analyzing classification or regression problems. Tree classification techniques, when they "work" and produce accurate predictions or predicted classifications based on few logical if-then conditions, have a number of advantages over many of those alternative techniques.

- Simplicity of results. In most cases, the interpretation of results summarized in a tree is very simple. This simplicity is useful not only for purposes of rapid classification of new observations (it is much easier to evaluate just one or two logical conditions, than to compute classification scores for each possible group, or predicted values, based on all predictors and using possibly some complex nonlinear model equations), but can also often yield a much simpler "model" for explaining why observations are classified or predicted in a particular manner (e.g., when analyzing business problems, it is much easier to present a few simple if-then statements to management, than some elaborate equations).

- Tree methods are nonparametric and nonlinear. The final results of using tree methods for classification or regression can be summarized in a series of (usually few) logical if-then conditions (tree nodes). Therefore, there is no implicit assumption that the underlying relationships between the predictor variables and the dependent variable are linear, follow some specific non-linear link function , or that they are even monotonic in nature.

- For example, some continuous outcome variable of interest could be positively related to a variable Income if the income is less than some certain amount, but negatively related if it is more than that amount (i.e., the tree could reveal multiple splits based on the same variable Income, revealing such a non-monotonic relationship between the variables).

- Thus, tree methods are particularly well suited for data mining tasks, where there is often little a priori knowledge nor any coherent set of theories or predictions regarding which variables are related and how. In those types of data analyses, tree methods can often reveal simple relationships between just a few variables that could have easily gone unnoticed using other analytic techniques.

1.8 General Computation Issues and Unique Solutions of CaRT

The computational details involved in determining the best split conditions to construct a simple yet useful and informative tree are quite complex. Refer to Breiman et al. (1984) for a discussion of their CART algorithm to learn more about the general theory of and specific computational solutions for constructing classification and regression trees. An excellent general discussion of tree classification and regression methods, and comparisons with other approaches to pattern recognition and neural networks, is provided in Ripley (1996).

1.9 Avoiding Over-Fitting: Pruning, Crossvalidation, and V-fold Crossvalidation

- A major issue that arises when applying regression or classification trees to ”real” data with much random error noise concerns the decision when to stop splitting. For example, if we had a data set with 10 cases, and performed 9 splits (determined 9 if-then conditions), we could perfectly predict every single case. In general, if we only split a sufficient number of times, eventually we will be able to "predict" ("reproduce" would be the more appropriate term here) our original data (from which we determined the splits). Of course, it is far from clear whether such complex results (with many splits) will replicate in a sample of new observations; most likely they will not.

- **Overfitting** - This general issue is relevant to tree classification, as well as neural networksand regression methods, If not stopped, the tree algorithm will ultimately "extract" all information from the data, including information that is not and cannot be predicted in the population with the current set of predictors, i.e., random or noise variation.

- The general approach to addressing this issue is first to stop generating new split nodes when subsequent splits only result in very little overall improvement of the prediction. For example, if we can predict 90% of all cases correctly from 10 splits, and 90.1% of all cases from 11 splits, then it obviously makes little sense to add that 11th split to the tree. There are many such criteria for automatically stopping the splitting (tree-building) process.

- **Pruning** - Once the tree building algorithm has stopped, it is always useful to further evaluate the quality of the prediction of the current tree in samples of observations that did not participate in the original computations. These methods are used to "prune back" the tree, i.e., to eventually (and ideally) select a simpler tree than the one obtained when the tree building algorithm stopped, but one that is equally as accurate for predicting or classifying "new" observations.

- **Cross-validation** - One approach is to apply the tree computed from one set of observations (learning sample) to another completely independent set of observations (testing sample). If most or all of the splits determined by the analysis of the learning sample are essentially based on "random noise," then the prediction for the testing sample will be very poor. Hence, we can infer that the selected tree is not very good (useful), and not of the "right size."

- **V-fold crossvalidation** Continuing further along this line of reasoning (described in the context of crossvalidation above), why not repeat the analysis many times over with different randomly drawn samples from the data, for every tree size starting at the root of the tree, and applying it to the prediction of observations from randomly selected testing samples. Then use (interpret, or accept as our final result) the tree that shows the best average accuracy for cross-validated predicted classifications or predicted values.

- In most cases, this tree will not be the one with the most terminal nodes, i.e., the most complex tree. This method for pruning a tree, and for selecting a smaller tree from a sequence of trees, can be very powerful, and is particularly useful for smaller data sets. It is an essential step for generating useful (for prediction) tree models, and because it can be computationally difficult to do, this method is often not found in tree classification or regression software.

1.10 Reviewing Large Trees: Unique Analysis Management Tools

Another general issue that arises when applying tree classification or regression methods is that the final trees can become very large. In practice, when the input data are complex and, for example, contain many different categories for classification problems and many possible predictors for performing the classification, then the resulting trees can become very large. This is not so much a computational problem as it is a problem of presenting the trees in a manner that is easily accessible to the data analyst, or for presentation to the "consumers" of the research.

1.11 Analyzing ANCOVA-like Designs

The classic (Breiman et. al., 1984) classification and regression trees algorithms can accommodate both continuous and categorical predictor. However, in practice, it is not uncommon to combine such variables into analysis of variance/covariance (ANCOVA) like predictor designs with main effects or interaction effects for categorical and continuous predictors. This method of analyzing coded ANCOVA-like designs is relatively new and. However, it is easy to see how the use of coded predictor designs expands these powerful classification and regression techniques to the analysis of data from experimental designs (e.g., see for example the detailed discussion of experimental design methods for quality improvement in the context of the Experimental Design module of Industrial Statistics).

1.12 Computational Details

The process of computing classification and regression trees can be characterized as involving four basic steps:

1. Specifying the criteria for predictive accuracy
2. Selecting splits
3. Determining when to stop splitting
4. Selecting the "right-sized" tree.

These steps are very similar to those discussed in the context of Classification Trees Analysis (see also Breiman et al., 1984, for more details).

1.13 Specifying the Criteria for Predictive Accuracy

The classification and regression trees (C&RT) algorithms are generally aimed at achieving the best possible predictive accuracy. Operationally, the most accurate prediction is defined as the prediction with the minimum costs. The notion of costs was developed as a way to generalize, to a broader range of prediction situations, the idea that the best prediction has the lowest misclassification rate. In most applications, the cost is measured in terms of proportion of misclassified cases, or variance. In this context, it follows, therefore, that a prediction would be considered best if it has the lowest misclassification rate or the smallest variance. The need for minimizing costs, rather than just the proportion of misclassified cases, arises when some predictions that fail are more catastrophic than others, or when some predictions that fail occur more frequently than others.