Expanded Overview of CRT

1. Strengths (and Weaknesses) of CRT

Strengths. CRT can display nonlinear relationships among factors. It does not assume a normal distribution; thus, making it easier for researchers in not having to deal with the consequences of violating distributional assumptions. CRT has clear advantages over other statistical methods in terms of a variety of data structures that it can work on, is invariant under some transformations, can be robust against outliers, and has the accessibility of the resulting tree-based output instead of potentially complex equations. CRT can be applied to any data structure with various kinds of predictors (e.g., sparse, skewed), including both categorical and continuous data without a necessity of pre-processing them (Kuhn & Johnson, 2013). In a standard data structure, when the continuous input variables are monotonically transformed, the CRT result is invariant. In other words, the splitting criterion value can also be monotonically transformed, yielding the same classification result. For example, if the CRT splits at a value of an original variable at 3, and the original variable is squared for the CRT analysis, then the CRT will split at a value of 9 with the new transformed variable (i.e., the squared variable) (Breiman et al., 1984).

Tree-based methods resemble human’s decision-making process more than traditional regression and classification approaches (James, Witten, Hastie, & Tibshirani, 2017). When reporting the results, CRT presents the results in an easy-to-interpret form; the tree classifiers illuminate the structure of the data and the steps one should use to decide. It stores the final classification results in a simple form and can be applied to the new data efficiently (Breiman et al., 1984).

Tree-based methods, including ensemble methods—bagging, random forest, and boosting, have the advantage of being able to be applied to scenarios with a small sample size but large number of variables; this is because they consider one predictor at a time, so they can handle a large number of variables sequentially (Strobl et al., 2009). Tree-based methods also have the advantage of approximating any unknown functions, including nonlinear and complex interactions without having to prespecify the shape of the function, the number and the position of the splits, or the relationships between the predictors and the outcome (Strobl et al., 2009).

Weaknesses. Model instability and less-than-optimal predictive performance are widely considered as two weaknesses of CRT (James, Witten, Hastie, & Tibshirani, 2017; Kuhn & Johnson, 2013; Strob et al., 2009). Model instability means that a small change in the learning data can lead to structural changes in the final estimated tree. We can understand the amount of the random variability present in the data by drawing bootstrap samples—smaller samples of the same size repeatedly resampled with replacement—from the original data and see whether the trees constructed from the different samples are different. Less-than-optimal predictive performance can happen when the relationship between the independent variables and the outcome variable cannot be best captured by the rectangular homogenous subspaces defined by the tree partitions. Ensemble methods that combine multiple trees and average over them, such as random forests, have shown great improvement in the prediction accuracy (James, Witten, Hastie, & Tibshirani, 2017; Kuhn & Johnson, 2013; Strob et al., 2009).

Another two potential problems with CRT are the order effect and the “XOR” problem (Strobl et al., 2009). The order effect is what was discussed regarding the similarity between CRT and linear models. When stepwise variable selection is used, variables are selected considering what was entered before and not considering what variables yet to come. The

“XOR” problem is that we may have variables without main effects but with interaction effects. Those variables without main effects may not be selected in CRT, thus neither is their interaction. However, the ensemble method can alleviate both problems. Ensemble methods by constructing a number of parallel tree models will help make the order effects counterbalance and the overall importance measure of each individual predictor more reliable. In addition, some of the ensemble methods, such as random forests, will randomly preselect splitting variables and thus make it possible that variables without main effects may still be selected, thus alleviating the “XOR” problem (Strobl et al., 2009).

2. Impurity Concept and Measurement

CRT uses impurity as a statistical criterion for tree growth for both the classification trees and regression trees. Impurity is a measure of remaining heterogeneity usually quantified in terms of whether members in a group properly belong in the different categories of the outcome variable: an equal number of group members belonging in different categories of the outcome variable is maximally impure, and the case of all members belonging in the same category of the outcome variable is absolutely pure ((Breiman et al., 1984; Ma, 2018). Most scenarios are in-between the extremes, being neither pure nor maximally impure. So a predetermined impurity threshold most often determines whether the partitioning process should continue or stop.

Most data partitioning methods for regression trees such as AID and CART use “the node mean of Y as predicted value and the sum of squared deviations as node impurity function” to build “piecewise constant regression trees” (Loh, 2014, p. 337). The statistical technique behind the Regression Tree is to choose variables that can maximally reduce the variance in the dependent variables. More specifically, the within-node variance is used as a criterion to measure the reduction of the impurity between a parent node and a child node (Ma, 2018). Another way to envision the process of building regression trees is to think of it as ways to partition the predictor space. The predictors and the exact splitting values are chosen to minimize the sum squares of residuals (RSS) between the observed and predicted outcome variables; as a result, the predictor space is partitioned into various regions as a combination of different predictors and their values. The observations that fall into the region of the predictor space are predicted to have the same value (James, Witten, Hastie, & Tibshirani, 2017).

3. Building CRT

3.1 Grow Tree

CRT grows trees by using a reduction in impurity as a measure that is calculated as the difference between the impurity measured in the parent node and the impurity measured in its child nodes weighted by the proportion of cases in each child node. CRT selects the independent variable that leads to the largest reduction in impurity to partition the parent node. After the described partitioning of the parent node is completed, the same analyses procedure will be carried out in every child node, building the CRT tree.

One challenge in CRT is arriving at the right-sized tree; similar to increased R^2 in regression with more variables added, more splits in trees lead to lower values of impurity, thus, encouraging a larger tree to be built in general, in the most extreme case, when each node only contains one case, the prediction error becomes zero; however, such a tree has no predictive power on an independent dataset. The method can be reframed slightly to deal with this and is discussed in the later “prune tree” section.

3.2 Stopping Rules

A careful selection of stopping rules is needed to stop the partitioning because a relaxed stopping rule, which allows much error in the structure of the tree, will lead to a relatively small tree that fails to capture the complex relationship among factors; in contrast, a strict stopping rule which allows little error in the structure of tree will lead to a very large tree that represents the current dataset well but may not be meaningful or typical for other datasets. Types of rules/techniques used to stop trees include hypothesis testing, cross-validation, reduction in impurity, and common rules, such as the number of cases in a terminal node (Ma, 2018).

Hypothesis testing. Hypothesis testing was traditionally used to test whether the selected predictor with the appropriate categorizations that produce the largest reduction in the impurity in a node makes a statistically significantly different partitioning from a random partition (Ma, 2018). A chi-square statistic may be used to measure the degree of deviation but can be conservative.

Cross-Validation. When the sample size is large, the cross-validation approach divides data into subsets—one set (a training set) that builds a tree and produces classifiers that are used to predict the underlying structure of the data, and a test set that validates the tree by calculating the impurity using the classifiers. The tree partitioning stops once the impurity gets to its smallest possible value (Breiman et al., 1984). In order to make full use of the sample, especially when the original sample is small, an alternative validation method is k-fold cross-validation. This method divides the original sample into k mutually exclusive subsets containing an (approximately) equal number of cases. Each time one subset serves as the test sample on which the prediction error is calculated, while the other subsets serve as a large training sample that is almost as large as the original sample size. The same procedure is carried out k times so that each subset gets to serve as a test sample once. Then the prediction error will become the average of the prediction error across k times. Because each case is used to construct the classifiers and only used once in a test sample, cross-validation makes full use of the data and is parsimonious with data (Breiman et al., 1984).

Reduction in impurity. When at least one predictor can contribute to the amount of reduction in impurity more than a selected threshold of the reduction in impurity, the tree keeps growing. The tree stops growing when none of the predictors can reduce the impurity more than the threshold. However, because the threshold is an arbitrary number, it is fairly common to set the number of cases in the terminal node as a threshold instead (Ma, 2018).

3.3 Tree Pruning

Cross-validation mentioned above is often used for pruning by measuring the error on the testing sets (Breiman et al., 1984). A tree is let fully grown until minimum impurity measures are met. Child nodes that contribute marginally to the decrease of the impurity are trimmed according to a nonnegative cost complexity pruning parameter, which indicates the "trade-off between the complexity of the tree and the fit to the training data"; in other words, RSS is adjusted to pay the price for a more complex tree (i.e., a tree of more depth) than a smaller tree (James, Witten, Hastie, & Tibshirani, 2017, p. 307). Their parent nodes become terminal nodes.