Delayed Greedy Algorithm for Classification and Regression Trees (CART)

Kyle A. Caudle [*] Brenna Mollett [†]

Abstract

Classification and Regression trees (CART) are non-linear prediction models dating back to Breiman's work in the 1980's. These models are often thought of as decision trees whereby the feature space is divided into a tree-like structure based on based on specific levels of the independent variables. The basic CART methodology cycles through all variables and levels of the variables until it finds a partition of the feature space that minimizes the total impurity of the tree. CART is a greedy algorithm because it just looks for the split point that gives you the largest reduction in impurity. Our approach finds the best split for each of the possible *n* candidate predictor variables. For each of these *n* splits, we then build a tree for each of the two partitions obtained from the first split. This results in *n* regression trees. The resulting trees are then compared to see which tree is best. We describe the "best" tree as either be the tree with maximum R^2 or minimum cross validation error. This paper will show that the best first split doesn't always lead to the best tree.

Key Words: regression trees, greedy algorithms, prediction

1. Introduction

Binary decision trees trace their initial roots back to social scientists in the early 60's [5]. With the development of the Automatic Interaction Detection program (AID) by Morgan and Sonquist [8] binary decision trees started to become a popular tool for organizing and interpreting data. Leo Breiman and Jerome Friedman began working on trees independently in the early 70's and then teamed up with Stone and Olshen to publish what most would consider the bible of the subject "Classification and Regression Trees" in 1984 [1]. Since the 80's CART has become one of the more popular machine learning tools.

Tree based models or decision trees are used to solve two basic problems.

1. **Classification**: Deciding what cases belong in each of several classes (categorical response).
2. **Regression**: Deciding what value to give to a continuous response variable based on the values of several attributes.

The acronym for these types of problems is CART. Classification and Regression Trees. CART does not work very well on small data sets. You need enough classes to get an overall sense of the data. With large datasets you may find structure within the data that is not possible through regular regression models. Decision trees are a type of regression in the sense that you have variables that you use to predict some type of a response which can either be a class or a continuous variable.

Some benefits of regression trees include:

- Can handle missing values.
- No complicated software or user expertise to select variables.
- Can model non-linear relationships and local effects.

[*] South Dakota School of Mines and Technology, Rapid City, SD 57701, USA

[†] South Dakota School of Mines and Technology, Rapid City, SD 57701, USA

- Automatically account for variable interactions.

We conclude our introductory section with an example borrowed from “Classification and Regression trees” [1].

Example: At the University of California, San Diego Medical Center, when a heart attack patient is admitted, lots of data on the patient is collected (19 variables to be exact!). The goal is to determine if they are a “high risk” or a “low risk” patient. Based on the 19 variables, 3 were deemed necessary for determining which class to put patients in. The decision tree is shown below (Yes - Left, No - Right).

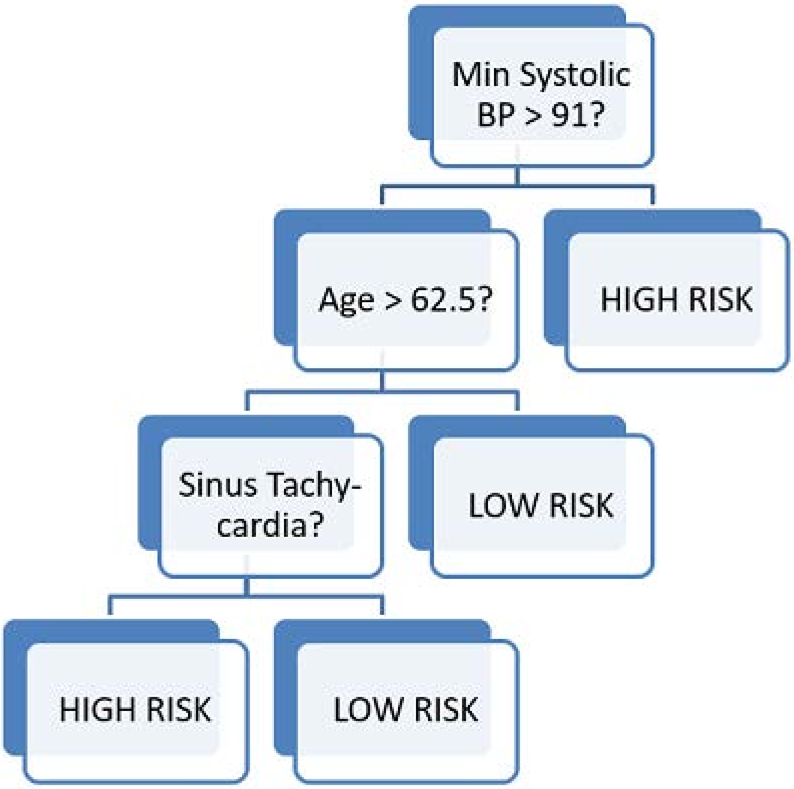

Figure 1: Sample Decision Tree

For each “node” in the decision tree a “yes” answer moves you to the left and a “no” answer moves you to the right. As an example, consider a 50 year old patient who has a systolic blood pressure (BP) that is above 91 and Sinus Tachycardia is present. By inspection of the decision tree, the first node would send us left because their BP is more than than 91. Next, because they are less than 62.5 years of age, we would proceed right which would put them in the “low risk” category.

2. Delayed Greedy Approach

The CART greedy algorithm determines the locally optimal split choice. Locally optimal splits often provide a sub-optimal tree [4]. Our simplest delayed greedy approach (DGA), splits on all variables initially. After performing the first split, the remaining tree(s) will be built using the CART greedy algorithm.

Suppose there are n candidate predictor variables. DGA would find the best split point for each of the n predictor variables. After partitioning the data based on the best split for a predictor variable, a decision tree is built using the CART greedy algorithm. This results in n separate trees (figure 2).

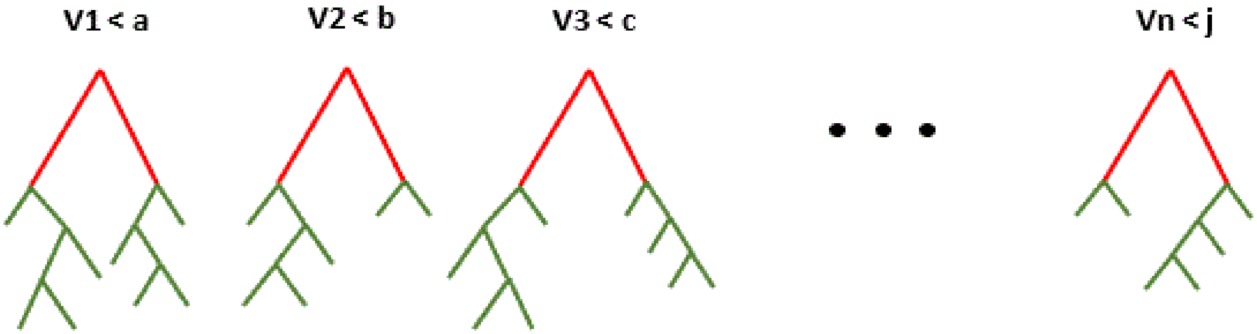

Figure 2: Sample Decision Tree

For each of these n trees, we must decide which tree is "best". The criteria for determining which tree is "best" should be based on size, depth or prediction accuracy. Smaller shallower trees are easier to interpret and also have a smaller computational cost. Larger trees tend to have better predictive capability. Unfortunately, the problem of building an optimal tree is intractable [6]. For obvious reasons, prediction accuracy is probability the most desirable tree characteristic and will hence be the focus of this paper. In our demonstrations, we choose to use leave one out (jackknife) cross-validation and the overall coefficient of determination (R^2) for each tree as the method for determining the "best" overall tree.

3. Demonstration 1 (auto-mpg)

The data from demonstration was taken from the StatLib library which is maintained at Carnegie Melon University. The data set auto-mpg [7] has been used frequently as a decision tree example. We start by summarizing the data.

- **Response**: Miles per gallon
- **Number of Instances**: 398
- **Attributes**:

Table 1: Auto-Mpg Attributes

Variable	Description	Type
mpg	Miles per gallon	continuous
cyl	Cylinders	discrete
disp	Displacement	continuous
hp	Horsepower	continuous
wt	Weight	continuous
ac	Acceleration	continuous

Since the response variable is continuous, this would be an example of a regression tree. The total sum of squares (no partitions) for the dataset would be calculated as:

$$\text{TSS} = \Sigma_i (y_i - \bar{y})^2 = 24,252.58$$
$$(y_i = \text{response = mpg})$$

Now we determine the split value for each variable that partitions the data so that the drop in sum of squares is the largest.

Table 2: SS Drop for Split 1 (auto-mpg)

Variable	SS Drop	Split Value
cyl	13,979.07	5.5
disp	13,982.74	190.5
hp	12,394.07	93.5
wt	13,480.25	2764.5
ac	4,867.43	13.75

The CART greedy algorithm would split on displacement because the reduction in sum of squares is the largest. The CART greedy algorithm would produce the decision tree in figure 3.

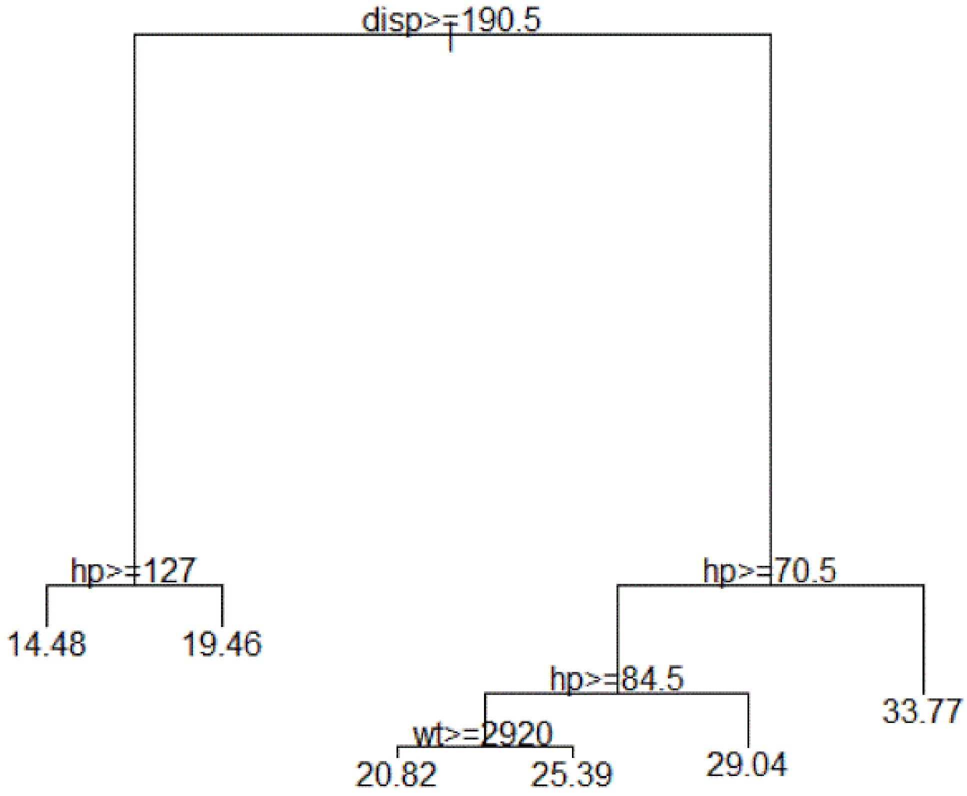

Figure 3: Auto-Mpg Decision Tree

The DGA would instead of building only this tree, it would build the other four remaining trees. Each tree was built using the rpart package in R [9]. We overgrow each tree and then prune it back to it's optimal size. After building the 5trees, we compare them to see which tree was "best".

For each of the trees, we calculate the R^2 value and the CV error. Table 3 presents the results.

Table 3: MPG Results

Variable	SS Drop (S1)	R^2	CV Error
cyl	13,979.07	0.8076	3.076379
disp	13,982.74	0.7893	3.123278
hp	12,394.07	0.7788	2.895039
wt	13,480.25	0.7627	3.010638
ac	4,867.43	0.8492	3.173234

The best fitting tree would be the one that splits on acceleration (ac) first with an R^2 value of 0.8492. The predictive capability of each tree is similar with cross-validation errors between 2.9 and 3.2 mpg. Another measure of quality would be the prediction accuracy on an independent noise free testing dataset.

4. Demonstration 2 (Red Wine)

Our second demonstration is also a regression tree. For this demonstration, we use the red wine dataset from Porto Portugal [2]. We start by summarizing the data.

- **Response**: Quality of red wine (0-10)
- **Number of Instances**: 1599
- **Attributes**: fixed acidity, volatile acidity, citric acid, residual sugar, chlorides, free sulfur dioxide, total sulfur dioxide, density, pH, sulphates, alcohol.

The total sum of squares (no partitions) equals 1,042.67. Table 4

Table 4: SS Drop for Split 1

Variable	SS Drop	Split Value
Fixed Acidity	24.71	9.95
Volatile Acidity	119.25	0.43
Citric Acid	75.74	0.30
Residual Sugar	4.59	1.68
Chlorides	39.42	0.07
Free Sulfur Dioxide	3.98	19.50
Total Sulfur Dioxide	38.03	59.50
Density	67.16	0.98
pH	5.77	3.28
Sulphates	131.38	0.65
Alcohol	186.17	10.53

The CART greedy algorithm would split on alcohol. DGA would instead build 11 trees and compare them. Table 5 summarizes the results for each of the 11 trees. For predictive capability, we again see that all trees are equally competitive. Each error is within 0.5 points on a scale of 0-10. Splitting on raw sugar however, results in the best fitting tree.

Table 5: Wine Results

Variable	SS Drop (S1)	R^2	CV Error
Facid	1017.89	0.4021	0.5289
Vacid	923.35	0.4652	0.5492
Cacid	966.86	0.3237	0.5253
Rsugar	1038.01	0.6062	0.5300
Chlorides	1003.18	0.5176	0.5393
FreeSO2	1038.62	0.3652	0.5269
TotalSO2	1004.57	0.4073	0.5292
Density	975.44	0.5949	0.5282
pH	1036.83	0.5070	0.5322
Sulphates	911.22	0.3476	0.5174
Alcohol	856.43	0.3814	0.5241

5. Demonstration 3 (Email Spam)

Our last demonstration involves a classification tree. George Foreman at Hewlett-Packard laboratories collected 4601 emails of which, 1813 were positively identified as SPAM [3].

- **Response**: SPAM (yes/no)
- **Number of Instances**: 4601
- **Attributes**:

Table 6: SPAM Attributes

Variable	Description
make	The frequency of the word make
n000	The frequency of the 000 symbol
money	The frequency of the word money
bang	The frequency of the symbol
dollar	The frequency of the $ symbol
crl.tot	Total length of words in capitals

For classification trees, the standard method for measuring impurity is deviance. Deviance is defined as the negative log of the sum of the Gini impurity over all nodes. The Gini impurity measures of how often an element of a subset would be mislabeled if the element was randomly chosen according to the distribution of labels within the subset.

Suppose we have J classes, with $i = 1, 2, ..., J$. Let p_i be the fraction of items in that set are in class i. The Gini impurity is defined as,

$$I(p) = \Sigma_{i=1}^{J} p_i(1-p_i) = 1 - \Sigma_{i=1}^{J} p_i^2$$

Next, if we take the log of the Gini impurity we get,

$$\log[I(p)] = -2\Sigma_{i=1}^{J} \log(p_i)$$

Finally, If we sum over all K nodes in the tree and weight by the number of elements in the node we get the deviance.

$$\text{Deviance} = -2\Sigma_{j=1}^{K}\Sigma_{i=1}^{J} n_j \log(p_i)$$

At each step, the CART greedy algorithm proceeds by checking all split values for every variable and select the variable/split point that arrives at the largest drop in deviance. Table 7 shows the drop in deviance for the first split for each of the 6 variables in the SPAM data set.

Table 7: SPAM Deviance Drops

Variable	Drop in Deviance	Split Value
make	142.07	0.075
n000	398.25	0.125
money	496.05	0.01
bang	711.96	0.080
dollar	714.17	0.056
crl.tot	347.12	1499

The CART greedy algorithm would split on dollar for the first split. DGA would instead build 6 trees and compare them. Table 8 summarizes the results for each of the 6 trees. The cross validation error shown in 8 is the mis-classification error multiplied by the root node error (i.e. the initial frequency of SPAM in the database 1813/4601).

Table 8: SPAM Results

Variable	Deviance of Tree	CV Error
make	599	0.1296
n000	579	0.1218
money	595	0.1449
bang	549	0.1353
dollar	555	0.1374
crl.tot	662	0.1421

The tree that resulted in the smallest overall deviance was the tree that first split on bang. All of the cross-validation errors were similar. The tree with the largest deviance drop initially resulted in the second worst cross-validation error.

6. Final Remarks

As shown from the 3 demonstrations, the CART greedy method does not consistently produce the most optimal tree and because of this DGA is something to be pursued further. The main difference between the CART greedy algorithm and DGA is determining if the best split choice is decided locally or globally, respectively. More research would be necessary to discover if there are specific characteristics of a data set that could indicate whether DGA would work better on that data set than on other data sets that do not possess those characteristics. Additionally, the DGA could be developed into different algorithms that would decrease the computational cost while increasing the predictive capability.

References

[1] Leo Breiman, Jerome Friedman, Charles J Stone, and Richard A Olshen. *Classification and regression trees*. CRC press, 1984.

[2] Paulo Cortez, António Cerdeira, Fernando Almeida, Telmo Matos, and José Reis. Modeling wine preferences by data mining from physicochemical properties. *Decision Support Systems*, 47(4):547–553, 2009.

[3] George Forman. An extensive empirical study of feature selection metrics for text classification. *Journal of machine learning research*, 3(Mar):1289–1305, 2003.

[4] Rodney M Goodman and Padhraic Smyth. Decision tree design from a communication theory standpoint. *IEEE Transactions on Information Theory*, 34(5):979–994, 1988.

[5] Earl B Hunt. Concept learning: An information processing problem. 1962.

[6] OJ Murphy, , and RL McCraw. Designing storage efficient decision trees. *IEEE Transactions on Computers*, 40(3):315–320, 1991.

[7] J Ross Quinlan. Combining instance-based and model-based learning. In *Proceedings of the Tenth International Conference on Machine Learning*, pages 236–243, 1993.

[8] John A Sonquist. Finding variables that work. *Public Opinion Quarterly*, 33(1):83–95, 1969.

[9] Terry M Therneau, Elizabeth J Atkinson, et al. An introduction to recursive partitioning using the rpart routines. Technical report, Technical report Mayo Foundation, 1997.