Presented at the 2018 ICEAA Professional Development & Training Workshop - www.iceaaonline.com

Using Regression Trees to Model Step Functions

Bopha Seng

Tecolote Research

Outline

- Classification and Regression Trees
- Why use Regression Trees?
- Decision Trees Terminology
- Algorithm
- Cost Toy Example
  - Regression Tree
  - CV & Pruning
  - Cross Validation
- Learning Curve Toy Example
  - Regression Tree
  - CV & Pruning
  - Cross Validation

What is CART?

- CART -> Classification and Regression Trees
  - Developed in the 1980's by Breiman, Friedman, Olshen, Stone
  - Introduced decision tree based modeling into the statistics
  - One of many decision tree based methods
  - Rigorous approach to select the optimal tree (cross-validation embedded within algorithm)
- Involves *stratifying* or *segmenting* the predictor space into a number of simple regions.
- Classification tree: used for modeling ***discrete target variable***
- Regression tree: used for modeling ***continuous* target variable**
  - Possible applications: Cost Methodology; Cost Improvement Curve

Why use Regression Trees?

Cost Estimating Application

- Use to model cost that traditional regressions may not fit well
- Model step-like behavior we often witness in Cost Improvement, Design Life, Technology Nodes, etc

Pros

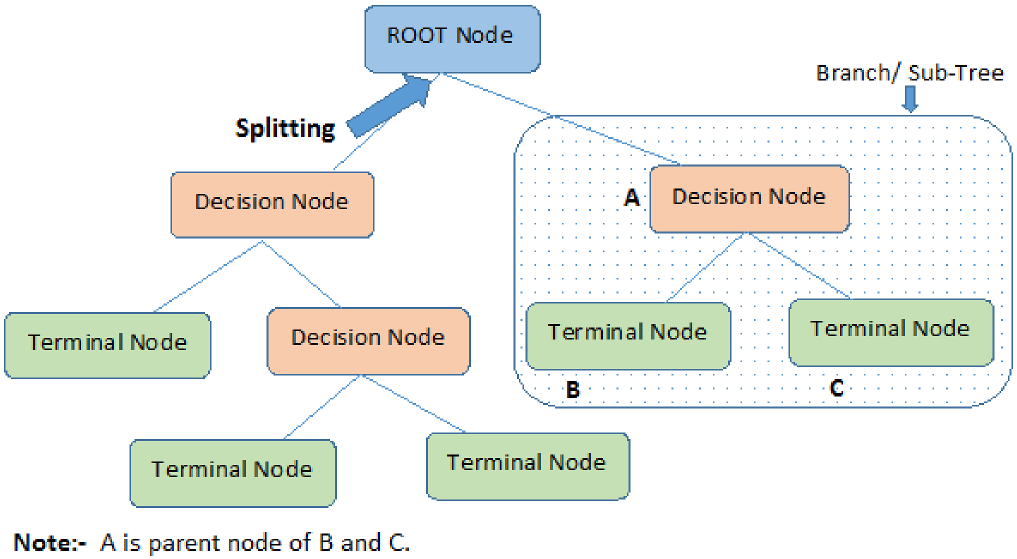
- Easy to Understand
- Useful in Data exploration
- Less data cleaning required
- Data type is not a constraint
- Non Parametric Method

Cons

- Over fitting
- Not fit for continuous variables

.

Decision Trees Terminology



- Root Node: It represents entire population or sample and this further gets divided into two or more homogeneous sets.
- Splitting: It is a process of dividing a node into two or more sub-nodes.
- Decision Node: When a sub-node splits into further sub-nodes, then it is called decision node.
- Leaf/Terminal Node: Nodes that do not split are called Leaf or Terminal node.
- Pruning: When we remove sub-nodes of a decision node, this process is called pruning. You can say opposite process of splitting.
- Branch / Sub-Tree: A sub section of entire tree is called branch or sub-tree.
- Parent and Child Node: A node, which is divided into sub-nodes is called parent node of sub-nodes where as sub-nodes are the child of parent node.

General Algorithm

1. Use recursive binary splitting to grow a large tree
    - Stop growing the tree based on stopping criterion (e.g. each leaf node has $r \geq 5$ data points)
2. Apply pruning to obtain best subtrees, as a function of α
3. Use K-fold cross-validation to choose α
4. Return the subtree from Step 2 that corresponds to the chosen value of α

Build the tree

- We divide the predictor space (values for *X*1,*X*2, . . .,*Xp*) into *J* distinct and non-overlapping regions, *R*1,*R*2, . . . , *RJ* .
- For every observation that falls into the region *Rj*, make the same prediction, which is simply the mean of the response values for the training observations in *Rj* .
- Find regions *R*1, . . . , *RJ* that minimize the SSE, given by

$$\sum_{j=1}^{J} \sum_{i \epsilon R_j} \left(y_i - \hat{y}_{R_j} \right)^2$$

- where $\hat{y}_{R_j}$ is the mean response for the training observations within the *j*th box.

Build the tree (continued)

- We take a top-down, greedy approach that is known as *recursive binary splitting*.
  - Begins at the top of the tree and then successively splits the predictor space
- To perform recursive binary splitting, we first select the predictor X_j and the cutpoint s such that splitting the predictor space into the regions $\{X|X_j < s\}$ and $\{X|X_j \geq s\}$ leads to the greatest possible reduction in RSS

We define the pair of half-planes as:

$$R_1(j,s) = \{X|X_j < s\}, \quad R_2(j,s) = \{X|X_j \geq s\}$$

in order to minimize the equation:

$$\sum_{i:x_i \epsilon R_1(j,s)} \left(y_i - \hat{y}_{R_1}\right)^2 + \sum_{i:x_i \epsilon R_2(j,s)} \left(y_i - \hat{y}_{R_2}\right)^2$$

- Continue to build the tree until a stopping criterion is reached
  - e.g. continue until no regions contain more than r observations

Prune

- Most Likely, growing the full tree will have over fitted your data and will lead to poor test set performance
- To obtain best subtrees we apply cost complexity pruning

$$R_\alpha = MC + \alpha L$$

$$MC = misclassification\ rate\ (relative\ to\ \#\ misclassifications\ in\ root\ node)$$

$$L = number\ of\ Leaf\ Nodes$$

- Get credit for lower MC; penalty for more leaves
- Let T_0 be the biggest tree
- Find T_α that minimizes R_α

Cross-Validation

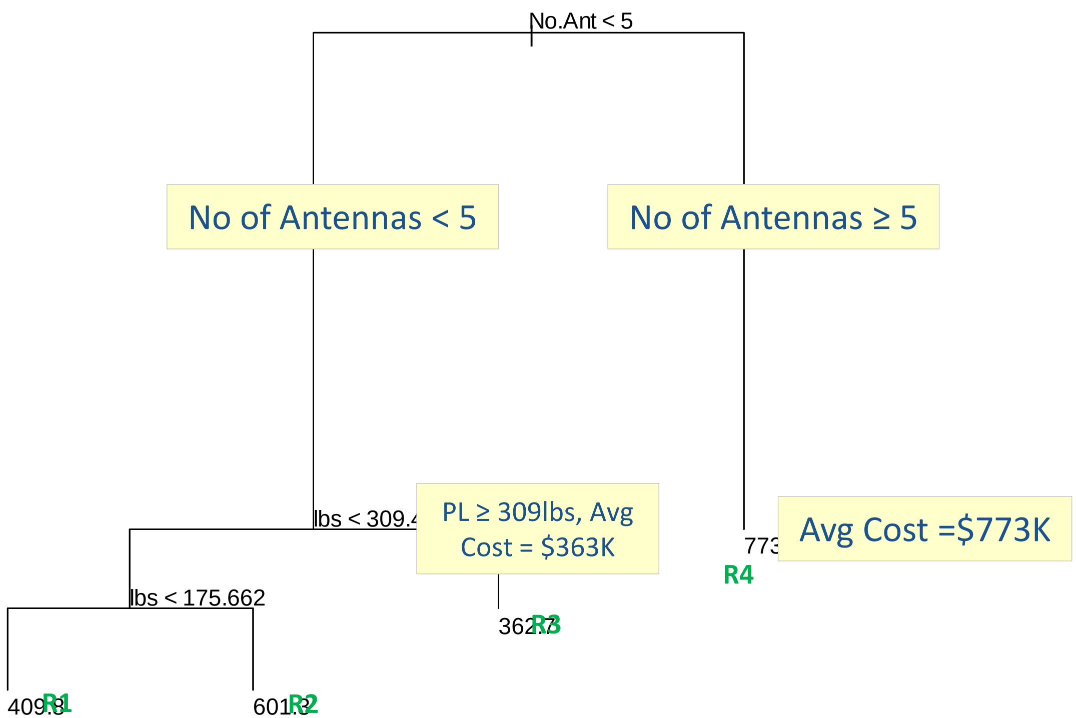
- Grow the tree on all the data: T_0
- Break the data into k equal sizes
- For each dataset (size =k), grow tree on p% of the data
  - Test the remaining (1-p)% of the data on the nested tree to each value of α
  - For each α, add up errors for all test data sets
- Keep track of the α corresponding to lowest test error and the corresponding nested tree

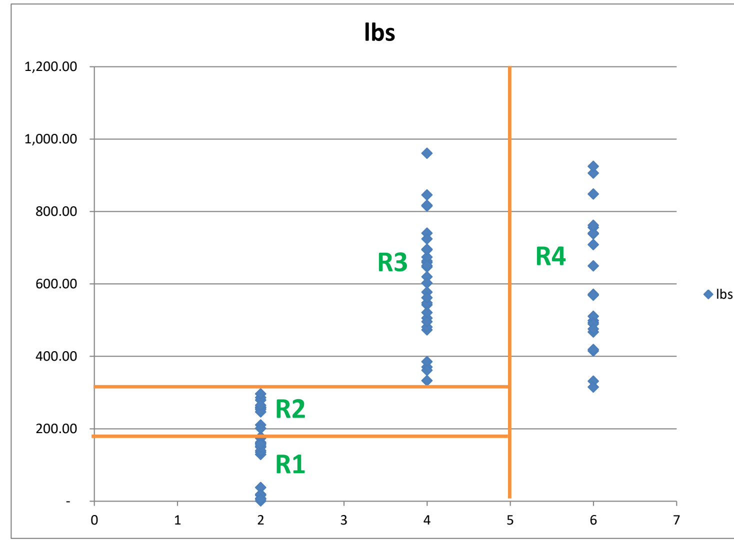
Cost Toy example

- Let's try to estimate the cost of a satellite payload using a Regression Tree
- Toy Example (75 data points)
  - Independent variables: Weight (lbs), Number of Antennas
  - Dependent variables: Nonrecurring Cost Dollars (BY12$K)
  - 50% Training Data Set, 50% Test Data Set

Cost	lbs	No.Ant
259.21	174.09	2
68.88	695.72	4
982.26	279.11	2
517.42	542.07	4
261.06	646.53	4
909.38	18.83	2
553.11	505.52	4
…	…	…

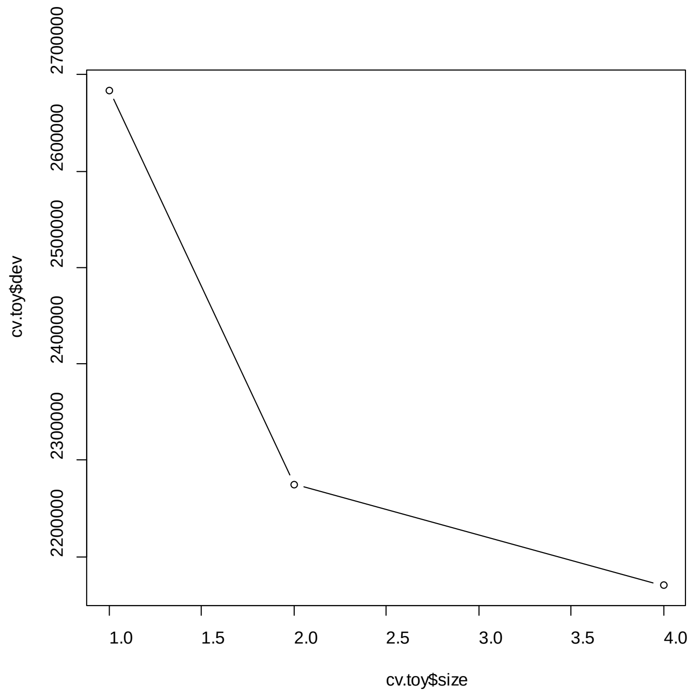
Cost: The Full Tree

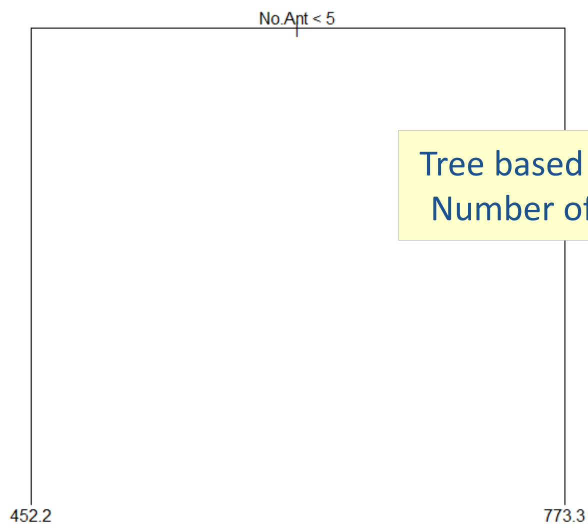





```
Regression tree:
tree(formula = Cost ~ lbs + No.Ant, data = toy, subset = train)
Number of terminal nodes:  4
Residual mean deviance:  40560 = 1339000 / 33
Distribution of residuals:
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
-378.70 -150.60  -19.14    0.00  134.70  552.90
```

Cost: Prune the tree

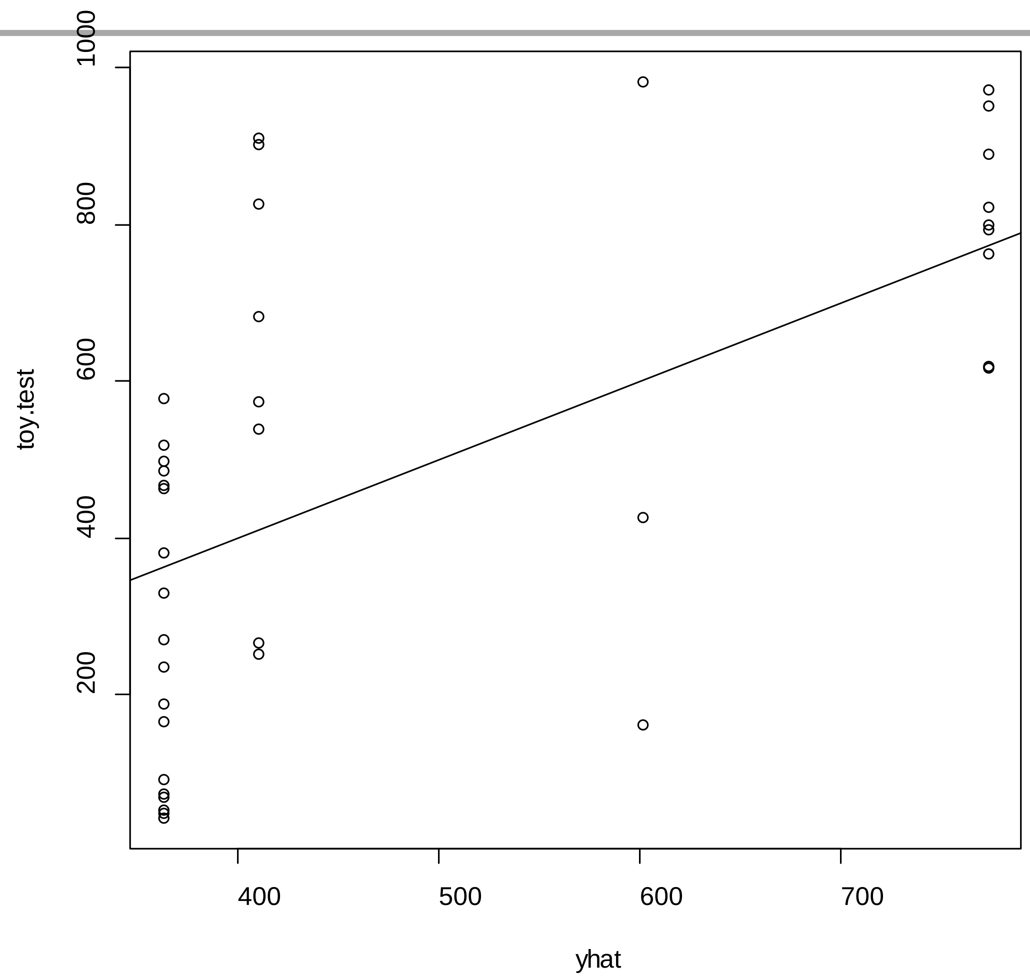




Tree based entirely on Number of Antennas

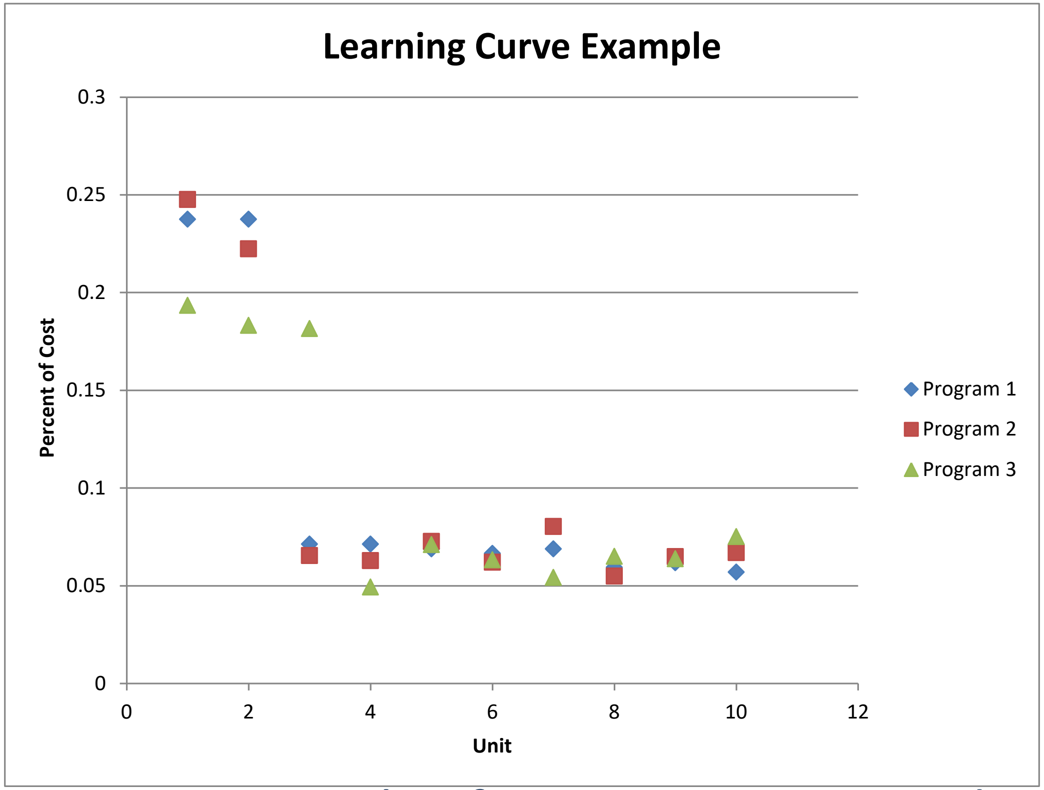
- We can check the CV to see if we can improve performance
- In this case, no additional pruning is needed
- If we had applied pruning we would have 1 spilt based on number of Antennas
  - Would have increased CV and lost valuable insight into cost impact due to PL wt.

Cost: Cross-Validation



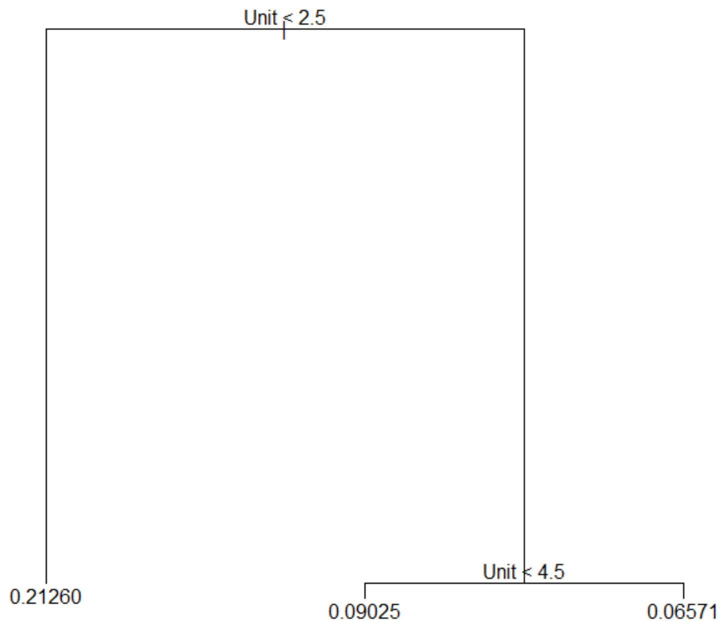
- MSE = 55049.83, Square Root MSE = $235K
  - Model estimates test within $235K of the Mean

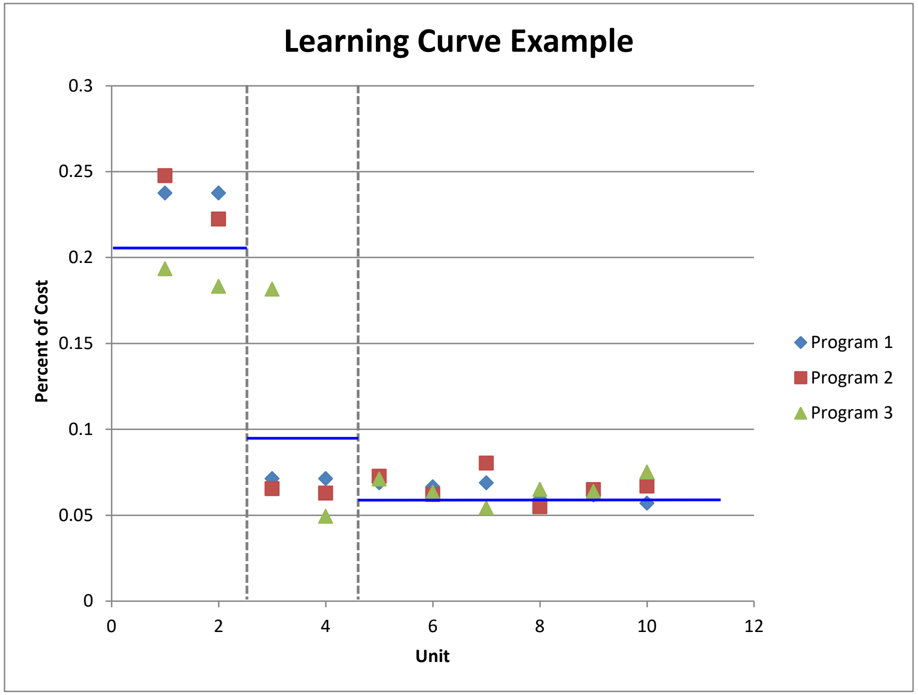
Cost Improvement Toy Example



- Above is a common example of Cost Improvement data we experience
  - Where SV1, 2, and/or 3 (the development vehicles) are much more expensive than the production vehicles
  - We want to use Regression Trees to model the data in a step-like function

Cost Improvement - Tree





- Using this technique allows us to model the step like behavior of Cost Improvement curves
- Able to identify the major shift in cost between the development and production costs

Conclusion

- Regression Trees (and Classification Trees) can be a valuable tool in the cost estimating field
  - Can be used to estimate cost for elements that don’t follow traditional regressions
  - Can identify where splits in the technical/programmatic data have the most impact of the cost
  - Can eliminate variables that have no impact to tree
  - Can model step-like behavior
- Future work
  - Quantify uncertainty for the individual leaves

References

- Breiman, Leo; Friedman, J. H.; Olshen, R. A.; Stone, C. J. (1984). *Classification and regression trees*. Monterey, CA: Wadsworth & Brooks/Cole Advanced Books & Software. ISBN 978-0-412-04841-8.
- Gareth, James; Witten, Daniela; Hastie, Trevor; Tibshirani, Robert (2015). *An Introduction to Statistical Learning*. New York: Springer. p. 315. ISBN 978-1-4614-7137-0.