Decision Trees

Sravani Vadlamani

Agenda

- Regression Trees
- Empirical Example
- Bagging, Boosting, Random Forests

Variable Selection

- We have discussed various approaches to variable selection
  - Forward, Backward and Stepwise Selection
  - Lasso Regression
  - Regression Splines
- The first two approaches focus on variable selection and identify variables with strong linear associations with the outcome. These approaches do not consider interactions unless they are explicitly input in the data

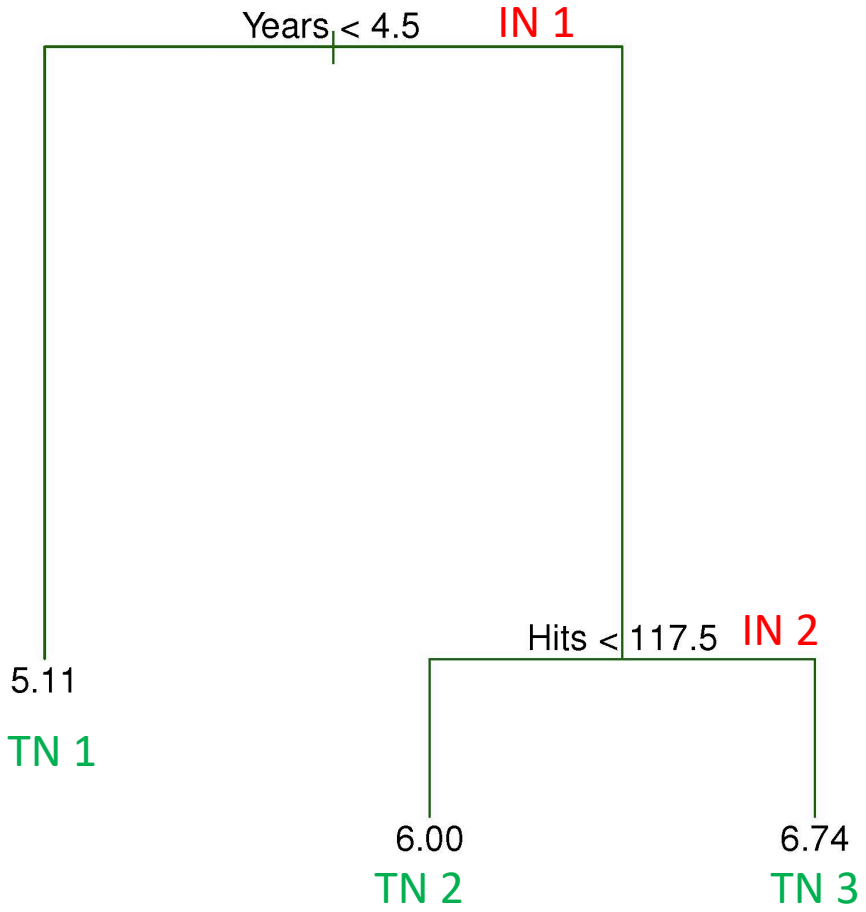
Regression Trees

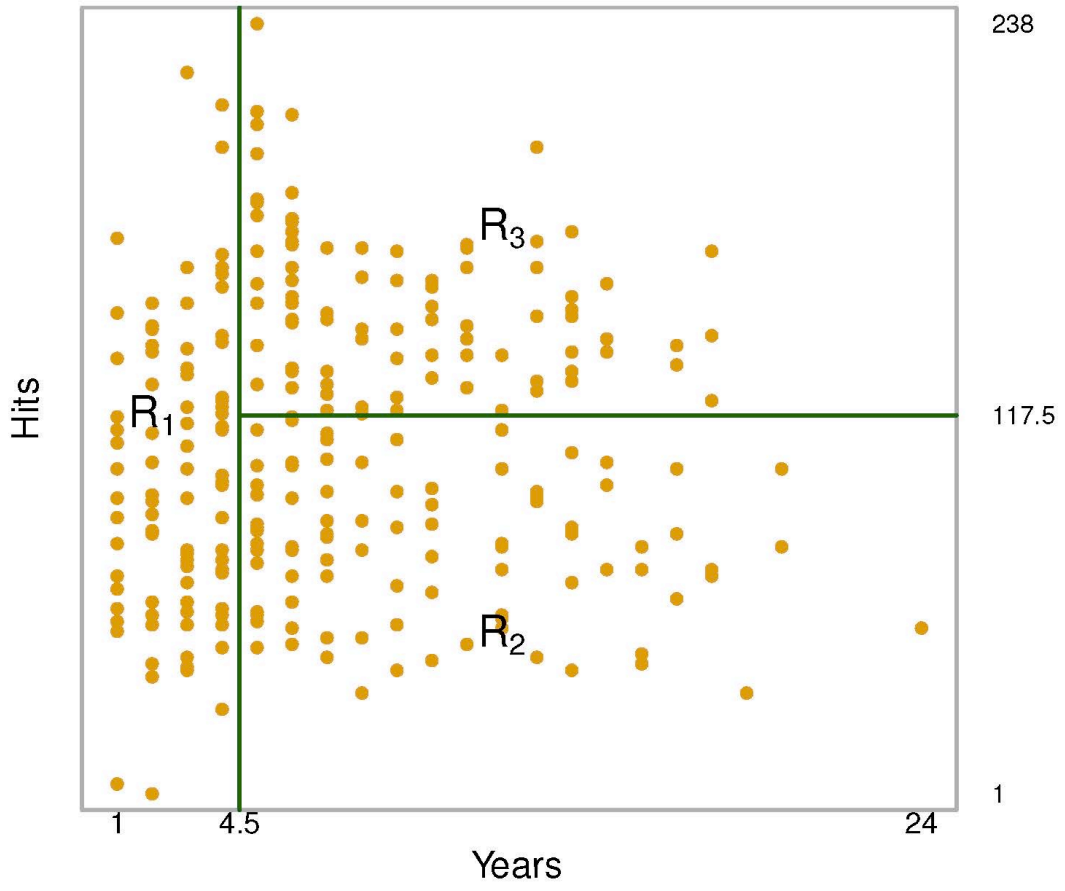
- Regression trees identify variables that are important for prediction in a different way
  - Stratifying the prediction space into several simple regions
  - Identifies variable and cut point on the variable to partition the data and does this repeatedly
  - Allows for non-linear associations and interaction effects
  - Simple methods useful for interpretation but not competitive in terms of accuracy
  - Can be applied to both regression and classification problems

Regression Trees - Example

- Predict Salaries of Baseball players using numbers of years of experience (Years) and number of hits made in the previous year (Hits)

Regression Trees - Example





Two Internal Nodes (denoted as IN 1 and IN 2)

Three Terminal Nodes (denoted as TN 1, TN2 and TN 3)

Regression Trees - Interpretation

- Years is the most important factor in determining salary.
- Players with less experience earn lower salaries in comparison to more experienced players
- When a player is less experienced, number of hits made in the previous year has little role in determining his salary
- In players with five or more years of experience, number of hits made in the last year does affect salary. Players with more hits tend to have higher salaries
- This tree is likely an oversimplification of the true relation between the variables. However, it is easy to display, interpret and explain

Tree Construction

- Two basic steps:
  - Divide the predictor space
  - For every observation that falls into a specific region, we make the same prediction, which is typically the mean of the outcome variable

Dividing the predictor space

- How is the division done? Theoretically, the region can be of any shape.
- Divide the predictor space into high-dimensional rectangles (2-dimensional space) or boxes (3 – dimensional space) for simplicity and ease of interpretation
- Goal is to find regions that minimize the residual sum of squares

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

- $\hat{y}_{R_j}$ is the mean response for region j

How to divide the predictor space

- It is infeasible to consider every partition of the predictor space
- We use a top-down (greedy) approach
  - Recursive binary splitting
- Algorithm
  - Choose the best split (that minimizes RSS) of the predictor space
  - Divide the data into partitions based on the split
  - Choose the best split of the predictor space for one of the two partitions
    - Now you get three partitions
  - Repeat until a stopping criterion is reached (for e.g. no region contains more than 5 observations)

Issues with greedy approach

- Results in a tree optimized to the dataset
  - At each step, a cut off score on a variable is chosen that minimizes the residual sum of squares the most
- Can lead to overfitting
  - Fitting a model too closely to the training data set may make it difficult to replicate the model on test data (i.e., model is too complex)
  - This can be avoided by using a smaller tree
- Secondary issue with regression trees is the heavy dependence on the first split

Pruning

- Process of reducing the size of your tree
- A smaller tree (few splits) can have less variance in the predicted values at the cost of slight bias
- Why not just grow smaller trees?
  - Too short sighted
  - Better to grow a large tree and then prune it back to create a subtree

How to prune?

- Our goal is to select a subtree that leads to the lowest test error rate
- There may exist many subtrees such that cross-validation of all possible subtrees becomes infeasible
- Instead, we need a way select subtrees from the possible realm of subtrees

Cost complexity pruning

- Also known as weakest link pruning is used to select a small set of subtrees
- Consider a sequence of trees indexed by a non-negative tuning parameter α
- For each value of α there corresponds a subtree T such that

$$\sum_{m=1}^{|T|} \sum_{i:x_i \in R_m} \left(y_i - \hat{y}_{R_m}\right)^2 + \alpha|T|$$

$|T|$ is the number of terminal nodes of tree T

R_m is the subset of predictor space corresponding to the m^{th} terminal node

α balances the trade-off between the subtree's complexity and its fit

Cost complexity pruning

- $As\ \alpha$ increases from 0, branches get pruned from the tree in a nested and predictable fashion
  - Makes it easy to obtain a sequence of subtrees as a function of α
- Optimal value of α is selected using cross-validation
- We will then return to the full data set and obtain the subtree corresponding to α

Building a Regression Tree - Algorithm

1. Use recursive binary splitting to grow a large tree on the training data, stopping when each terminal node has fewer than some minimum number of observations
2. Apply cost complexity pruning to obtain a sequence of best subtrees, as a function of α
3. Use K-fold cross validation to choose α i.e., divide the training observations into k-folds. For each fold:
    1. Repeat steps 1 and 2 on all but the kth fold on the training data
    2. Evaluate the mean squared error on the left-out kth fold as a function of α
    3. Average the results for each value of α and pick α to minimize the average error
4. Return the subtree from step 2 that corresponds to the chosen value of α

Classification Trees

- Like regression trees except used to predict a qualitative variable
- We predict that each observation belongs to the most commonly occurring class of training observations in the region to which it belongs
- We are interested in the class prediction corresponding to a particular terminal node region and the class proportions among the training observations that fall into the region

Classification Trees

- Combines search procedures and cut scores, which can create a simple rule-based approach to making decisions with multiple predictors
  - Enables the detection of higher order interactive effects and non-linear associations

Classification Trees - Construction

1. Divide the predictor space based on a decision rule (cut-off score on the best predictor)
2. Partition the data based on the decision rule
3. For each partition, repeat steps 1 and 2 until a stopping criterion is reached
4. Observations that fall into each terminal node (leaf) are given the same prediction, which is the model response

Classification Trees – Dividing Predictor Space

- Goal is to find predictors that minimize misclassification (false positive and false negatives)
- We cannot use RSS as a criterion for the binary splits
- We used classification error rate which is the fraction of the training observations in that region that do not belong to the most common class

$$E = 1 - max_k(\hat{p}_{mk})$$

- $\hat{p}_{mk}$ is the proportion of training observations in the mth region that are from the kth class
- E is not sufficiently sensitive for tree growing and hence Gini Index and entropy are preferred measures

Classification Trees – Gini Index

- Measure of total variance across the K classes.

$$G = \sum_{k=1}^{K} \hat{p}_{mk}(1 - \hat{p}_{mk})$$

- Takes on small value if all of the $\hat{p}_{mk}$ are close to zero or 1
- Also referred to as a measure of node purity – small value indicates that a node predominantly contains observations from a single class

Classification Trees – Entropy

- An alternative to Gini Index.

$$D = -\sum_{k=1}^{K} \hat{p}_{mk} log \hat{p}_{mk}$$

- Takes on value near zero if all of the $\hat{p}_{mk}$ are close to zero or 1
- Also referred to as a measure of node purity – small value indicates that a node predominantly contains observations from a single class
- Any of the three methods Classification rate, Gini Index or Entropy can be used when pruning. However, classification error rate is preferable if prediction accuracy of the final pruned tree is the goal

Trees vs Linear Models

- Which one is better?
- It depends
  - If the relation between the response and predictors is well approximated by a linear model, linear regression will outperform trees
  - If there is a highly non-linear and complex relation between the response and predictor variables, decision trees will outperform classical methods

Advantages and Disadvantages of Trees

- Advantages
  - Easy to explain (than linear regression!)
  - More closely mirrors human decision-making than the previously seen regression and classification methods
  - Easy to display and interpret for a non-technical audience
  - Handle qualitative predictors without creating dummy variables
- Disadvantages
  - Do not have the same level of predictive accuracy as other regression and classification approaches
  - Not very robust i.e., small changes in data and lead to drastic changes in the final estimated tree