DECISION TREES

Paul Speaker

Decision Trees

- Decision trees are a popular machine learning algorithm used for both classification and regression tasks.
- They work by recursively splitting the data into subsets based on the features that are most informative for the task at hand.
  - "Most informative" → creating the most separation (see next slide)
  - One variable at each split
- The resulting tree is a series of decisions that lead to the classification, based on the population of that part
- Each of the final nodes is commonly called a ***leaf***

Quantifying the Best Splits

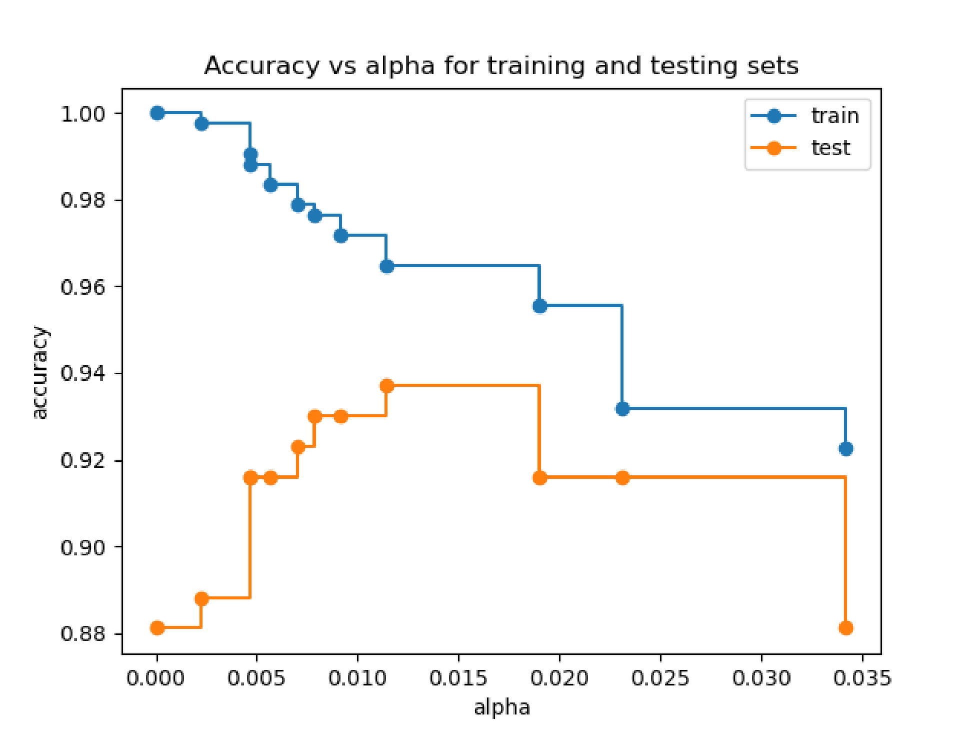
- Suppose you have a classification problem with 2 inputs
  - Prior proportions are 50/50
- The best split is one that adds the most certainty to the two classes
  - Best split would be between 2 leaves, where all of 1 class are in one leaf, and all in another are in the other
  - Have 2 things to decide
    - For each input determine a optimal value with which to split
    - Have to decide which input is the most useful (comparing the best for each input)
- With a real example, it is unlikely to get such a perfect split.
  - How to quantify?
    - Gini Coefficient
    - Entropy

Gini Coefficient and Entropy

- The Gini Coefficient and Entropy are 2 closely related measures of how good a split is for a decision tree
  - G = Sum(p*(1 – p)) where the Sum is taken over all the target classes
  - Entropy = Sum(-p*log(p)) where the Sum is taken over all the target classes
- Method for creating the split
  - Calculate the Gini coefficient or Entropy for current state
  - For each X calculate the split to which reduces the Gini Coefficient or Entropy the most
  - Among all X's see which X will reduce G or Entropy the most
  - That is our new split

Growing and Pruning Decision Trees

- Taken to its logical conclusion you could have almost as many splits as data points
  - This would clearly lead to overfitting
- Where to stop?
- What if you go too far?
  - Pruning in the process of removing leaves which contribute to overfitting
  - Based on cost-complexity
- Overall process
  - Build tree down until all leaves are either 100% classified or some size threshold
  - Hyperparameter α which controls pruning
    - When it's zero, no pruning Higher α is, more is pruned
    - Coefficient which penalizes for number of nodes
      - Very similar in concept to lasso/ridge regression



Advantages and Limitations of Decision Trees

- Big advantage is that a decision tree is intuitive and easy to explain
- Works well with categorical inputs
- Also easy to productionize in new environments (if-then-else statements)
- Disadvantages
  - Only one variable at a time → not great cuts of the data
  - Small changes in data can affect entropy calculations greatly
  - Pruning to avoid overfitting not easy to do
- Fortunately, there are solutions to these problems that will make decision trees great—we will cover these methods over the next couple of weeks
  - Resampling to have many trees, take "average"
    - Bagging
    - **Random forests**
  - Adaptive growing of trees so new trees "learn" from past trees
    - Boosting
    - **Xgboost**
  - Allow for splits based on combinations of X's
    - **Support Vector Machines**

Decision Trees in R

- We will use the tree package for making decision trees
- Standard format for command: tree(y ~ <sum of x's>, data = <df>)
- Character/string target
- Output lists out the tree structure with leaf populations
- Output can be plotted for tree structure
  - plot only creates the splitting
  - Separate "text" command to show structure
- predict, confusionMatrix works as usual
  - Predict values can be class or probabilities
    - Assign to leaf, give leaf proportions as probabilities
- Many options, but we won't use, since we typically won't use them in isolation