Tree Based Methods

Lecture

In this week's lecture, you are introduced to the concept of classification and regression trees (CART). In the lecture, we focused on the theory and algorithm. In this tutorial, we will apply CART to a real dataset.

There are a lot of different ways that the *tree* can be built and we will consider a few approaches.

One major draw back of CART is the poor predictive power. Many methods have been proposed to improve the predictive power of CARTS. Next week, we will discuss bagging, random forest and boosted trees, which are algorithm developed to improve the predictability of tree.

Tutorial

Exercise Let's talk trees. In a group of two, discuss the following topics

- What is NP problem?
- Discuss morphological differences between classification and regression trees
- How does CART grow? How do you find the optimal splits?
- Explain what is the greedy algorithm
- Explain what is the cost-complexity criteria and the objective of the criteria
- How does CART build a regression tree?
- What is surrogate split?

Not sure? Read these chapters

Author	Title	Relevant chapters
James et al	An Introduction to Statistical Learning with Applications in R	ch 8.1 and 8.3.1

Independent Learning

Exercises

- Question 1, Chapter 8.4
- Question 3, Chapter 8.4
- Question 4, Chapter 8.4
- Question 6, Chapter 8.4
- Question 8(a) to 8(c), Chapter 8.4
- Question 9, Chapter 8.4

  This problem involves the OJ data set which is part of the `ISLR` package.

  1. What is the objective of this study?

2. Create a training set containing a random sample of 800 observations, and a test set containing the remaining observations.
3. Fit a tree to the training data, with `Purchase` as the response and the other variables except for `Buy` as predictors. Use the `summary()` function to produce summary statistics about the tree, and describe the results obtained. What is the training error rate? How many terminal nodes does the tree have?
4. Type in the name of the tree object in order to get a detailed text output. Pick one of the terminal nodes, and interpret the information displayed.
5. Create a plot of the tree, and interpret the results.
6. Predict the response on the test data, and produce a confusion matrix comparing the test labels to the predicted test labels.
7. What is the test error rate?
8. Apply the `cv.tree()` function to the training set in order to determine the optimal tree size.
9. Produce a plot with tree size on the x-axis and cross-validated classification error rate on the y-axis.
10. Which tree size corresponds to the lowest cross-validated classification error rate?
11. Produce a pruned tree corresponding to the optimal tree size obtained using cross-validation. If cross-validation does not lead to selection of a pruned tree, then create a pruned tree with five terminal nodes.
12. Compare the training error rates between the pruned and unpruned trees. Which is higher?
13. Compare the test error rates between the pruned and unpruned trees. Which is higher?