Lecture 18 Tree-based Methods II: Classification Tree

STAT 441/505: Applied Statistical Methods in Data Mining

Linglong Kong

Department of Mathematical and Statistical Sciences
University of Alberta

Winter, 2016

Outline

Pruning a Tree

Classification Trees

Summary and Remark

Pruning a Tree

- The process described above may produce good predictions on the training set, but is likely to overfit the data, leading to poor test set performance.Why?
- A smaller tree with fewer splits (that is, fewer regions $R_1, \cdots, R_J$) might lead to lower variance and better interpretation at the cost of a little bias.
- One possible alternative to the process described above is to grow the tree only so long as the decrease in the RSS due to each split exceeds some (high) threshold.
- This strategy will result in smaller trees, but is too short-sighted: a seemingly worthless split early on in the tree might be followed by a very good split | that is, a split that leads to a large reduction in RSS later on.

Pruning a Tree

- A better strategy is to grow a very large tree T_0, and then prune it back in order to obtain a subtree.
- Cost complexity pruning — also known as weakest link pruning — is used to do this.
- we consider a sequence of trees indexed by a nonnegative tuning parameter α. For each value of α there corresponds a subtree $T \subset T_0$ such that

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

  is as small as possible.
- Here $|T|$ indicates the number of terminal nodes of the tree T, R_m is the rectangle (i.e. the subset of predictor space) corresponding to the mth terminal node, and $\hat{y}_{R_m}$ is the mean of the training observations in R_m.

Pruning a Tree

- There are two terms in the loss function, one is its fidelity and the other one is its penalty.
- The tuning parameter controls a trade off between the subtree's complexity and its fit to the training data.
- We select an optimal value $\hat{\alpha}$ using cross-validation.
- We then return to the full data set and obtain the subtree corresponding to $\hat{\alpha}$.

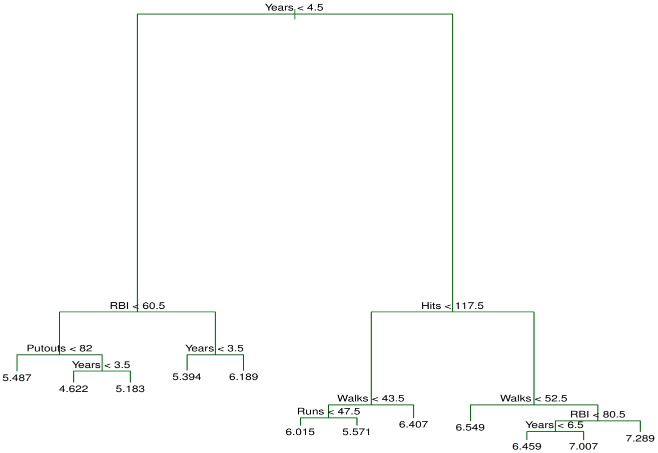
Pruning a Tree

1. Use recursive binary splitting to grow a large tree on the training data, stopping only when each terminal node has fewer than some minimum number of observations.
2. Apply cost complexity pruning to the large tree in order to obtain a sequence of best subtrees, as a function of α.
3. Use K-fold cross-validation to choose α. For each $k = 1, \cdots, K$:
    - Repeat Steps 1 and 2 on the $(K-1)/K$th fraction of the training data, excluding the kth fold.
    - Evaluate the mean squared prediction error on the data in the left-out kth fold, as a function of α. Average the results, and pick α to minimize the average error.
4. Return the subtree from Step 2 that corresponds to the chosen value of α.

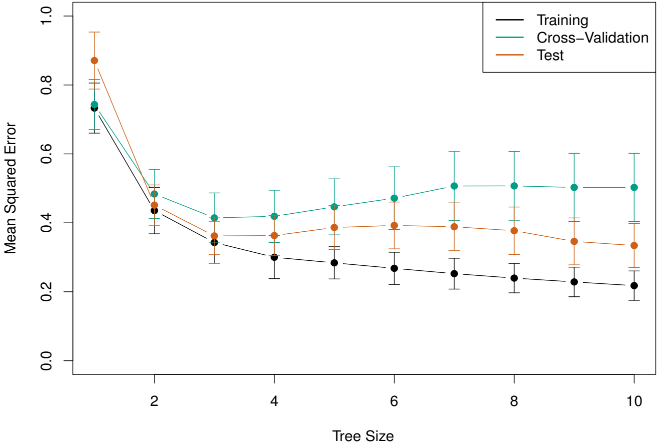
Baseball salary data

- First, we randomly divided the data set in half, yielding 132 observations in the training set and 131 observations in the test set.
- We then built a large regression tree on the training data and varied α in in order to create subtrees with different numbers of terminal nodes.
- Finally, we performed six-fold cross-validation in order to estimate the cross-validated MSE of the trees as a function of α.

Baseball salary data



Baseball salary data



Classification Trees

- Very similar to a regression tree, except that it is used to predict a qualitative response rather than a quantitative one.
- For a classification tree, we predict that each observation belongs to the most commonly occurring class of training observations in the region to which it belongs.
- Just as in the regression setting, we use recursive binary splitting to grow a classification tree.

Classification Trees

- In the classification setting, RSS cannot be used as a criterion for making the binary splits.
- A natural alternative to RSS is the classification error rate. this is simply the fraction of the training observations in that region that do not belong to the most common class:

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

 Here $\hat{p}_{mk}$ represents the proportion of training observations in the mth region that are from the kth class.

- However classification error is not sufficiently sensitive for tree-growing, and in practice two other measures are preferable.

Gini Index and Deviance

- The Gini index is denoted by

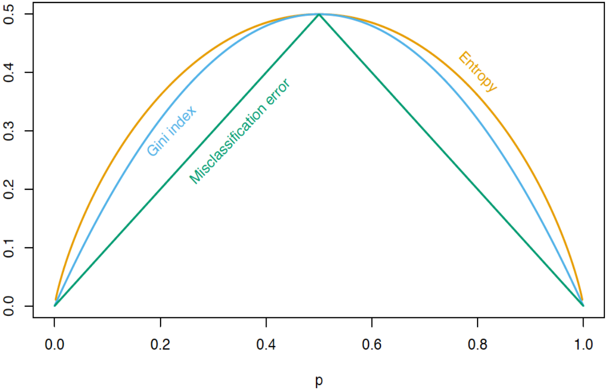
$$G = \sum_{k=1}^{K} \hat{p}_{mk}(1 - \hat{p}_{mk})$$

  a measure of total variance across the K classes. The Gini index takes on a small value if all of the $\hat{p}_{mk}$'s are close to zero or one.
- For this reason the Gini index is referred to as a measure of node purity — a small value indicates that a node contains predominantly observations from a single class.
- An alternative to the Gini index is cross-entropy, given by

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

- It turns out that the Gini index and the cross-entropy are very similar numerically.

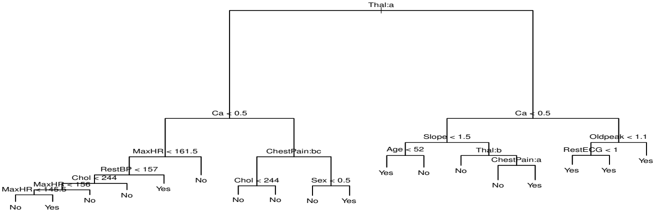
Gini Index and Deviance

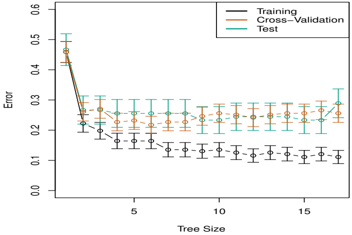


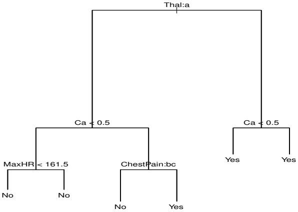
Heart Data

- These data contain a binary outcome HD or 303 patients who presented with chest pain.
- An outcome value of `Yes` indicates the presence of heart disease based on an angiographic test, while `No` means no heart disease.
- There are 13 predictors including `Age`, `Sex`, `Chol` (a cholesterol measurement), and other heart and lung function measurements.
- Cross-validation yields a tree with six terminal nodes. See next figure.

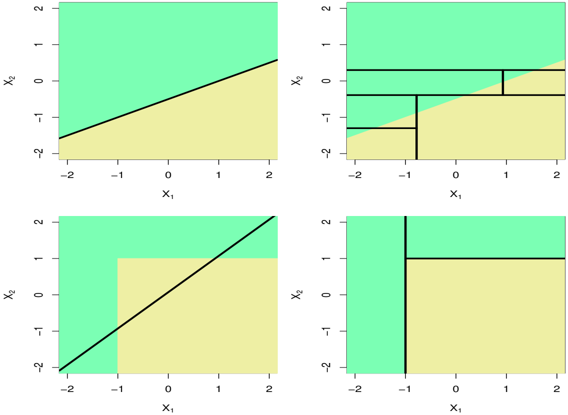
Heart Data


Thal:a
Ca < 0.5
Ca < 0.5
MaxHR < 161.5
ChestPain:bc
Slope < 1.5
Oldpeak < 1.1
RestBP < 157
Chol < 244
MaxHR < 156
MaxHR < 145.5
No
Yes
No
No
Yes
No
Chol < 244
No
No
Sex < 0.5
No
Yes
Age < 52
Yes
No
Thal:b
No
ChestPain:a
No
Yes
RestECG < 1
Yes
Yes
Yes


Training
Cross−Validation
Test
Error
0.0
0.1
0.2
0.3
0.4
0.5
0.6
5
10
15
Tree Size


Thal:a
Ca < 0.5
Ca < 0.5
MaxHR < 161.5
ChestPain:bc
Yes
Yes
No
No
No
Yes

Trees Versus Linear Models



- Top Row: True linear boundary; Bottom row: true non-linear boundary.
- Left column: linear model; Right column: tree-based model.

Advantages and Disadvantages of Trees

- Trees are very easy to explain to people. In fact, they are even easier to explain than linear regression!
- Some people believe that decision trees more closely mirror human decision-making than do the regression and classification approaches.
- Trees can be displayed graphically, and are easily interpreted even by a non-expert (especially if they are small).
- Trees can easily handle qualitative predictors without the need to create dummy variables
- Unfortunately, trees generally do not have the same level of predictive accuracy as some of the other regression and classification approaches seen in this book.
- However, by aggregating many decision trees, the predictive performance of trees can be substantially improved. We introduce these concepts next.

Summary and Remark

- Pruning a tree
- Classification Tree
- Read textbook Chapter 9 and `R` code
- Do `R` lab