

Lecture 17 Tree-based Methods I: Regression Tree

ECE 625: Data Analysis and Knowledge Discovery

Di Niu

Department of Electrical and Computer Engineering
University of Alberta

March 18, 2021

Outline

Introduction

Baseball salary data

Regression Tree

Summary and Remark

Tree-based Methods

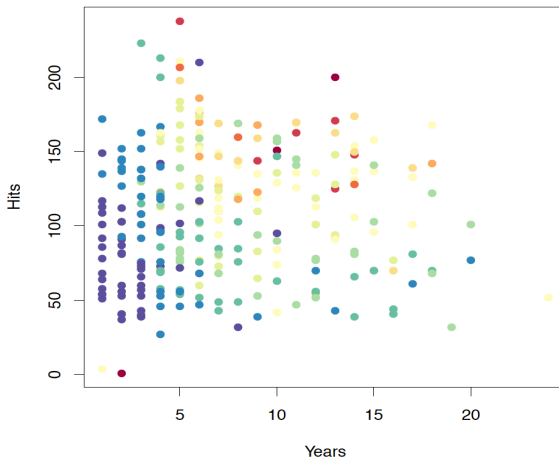
- ▶ We describe **tree-based methods** for regression and classification.
- ▶ These involve **stratifying** or **segmenting** the predictor space into a number of simple regions. **feature space**
- ▶ Since the set of splitting rules used to segment the predictor space can be summarized in a tree, these types of approaches are known as **decision-tree methods**.
- ▶ **Decision trees** can be applied to both **regression and classification problems**.
- ▶ We first consider regression problems, and then move on to classification.

Tree-based Methods

- ▶ Tree-based methods are **simple and useful** for interpretation.
- ▶ However they typically are **not competitive** with the best supervised learning approaches in terms of prediction accuracy.
- ▶ Hence we also discuss **SVM, Neural Network** **bagging, random forests, and boosting**. These methods grow multiple trees which are then combined to yield a single consensus prediction. **ensemble methods**
- ▶ Combining a large number of trees can often result in dramatic improvements in **prediction accuracy**, at the expense of some **loss of interpretation**.

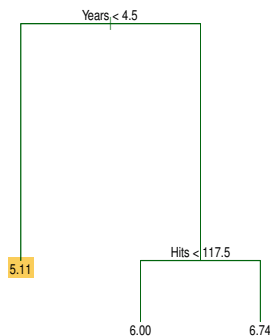
Baseball salary data

Salary is color-coded from low (blue, green) to high (yellow, red)



Decision tree

- ▶ For the Hitters data, a regression tree for predicting the log salary of a baseball player, based on the number of years that he has played in the major leagues and the number of hits that he made in the previous year.
- ▶ At a given internal node, the label (of the form $X_j < t_k$) indicates the left-hand branch emanating from that split, and the right-hand branch corresponds to $X_j \geq t_k$.
 - ▶ For instance, the split at the top of the tree results in two large branches. The left-hand branch corresponds to **Years** < 4.5, and the right-hand branch corresponds to **Years** ≥ 4.5.
 - ▶ The tree has **two internal nodes** and **three terminal nodes, or leaves**. The number in each leaf is the mean of the response for the observations that fall there.



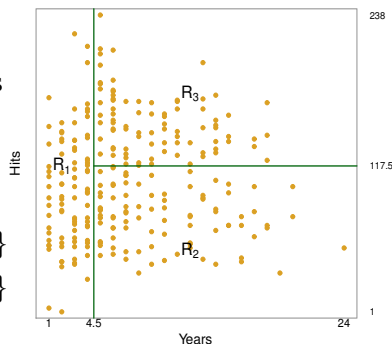
Baseball salary data

- ▶ Overall, the tree stratifies or segments the players into three regions of predictor space:

- ▶ $R_1 = \{X | \text{Years} < 4.5\}$

- ▶ $R_2 = \{X | \text{Years} \geq 4.5, \text{Hits} < 117.5, \}$

- ▶ $R_3 = \{X | \text{Years} \geq 4.5, \text{Hits} \geq 117.5, \}$



Terminology of Trees

- ▶ In keeping with the **tree** analogy, the regions R_1 , R_2 , and R_3 are known as **terminal nodes**.
- ▶ Decision trees are typically drawn **upside down**, in the sense that the leaves are at the bottom of the tree.
- ▶ The points along the tree where the predictor space is split are referred to as **internal nodes**.
- ▶ In the hitters tree, the two internal nodes are indicated by the text **Years** < 4.5 and **Hits** < 117.5.

Interpretation of Results

- ▶ **Years** is the most important factor in determining **Salary**, and players with less experience earn lower salaries than more experienced players.
- ▶ Given that a player is less experienced, the number of **Hits** that he made in the previous year seems to play little role in his **Salary**.
- ▶ But among players who have been in the major leagues for five or more years, the number of **Hits** made in the previous year does affect **Salary**, and players who made more **Hits** last year tend to have higher salaries.
- ▶ Surely an over-simplification, but compared to a regression model, it is easy to display, interpret and explain.

Building a Regression Tree

- ▶ We divide the predictor space — that is, the set of possible values for X_1, X_2, \dots, X_p — into J distinct and non-overlapping regions, R_1, R_2, \dots, R_J .
- ▶ For every observation that falls into the region R_j , we make the same prediction, which is simply the mean of the response values for the training observations in R_j .
- ▶ In theory, the regions could have any shape. However, we choose to divide the predictor space into high-dimensional rectangles, or **boxes**, for simplicity and for ease of interpretation of the resulting predictive model.
- ▶ The goal is to find boxes R_1, R_2, \dots, R_J that minimize the RSS, given by $\sum_{j=1}^J \sum_{i \in R_j} (y_i - \hat{y}_{R_j})^2$, where \hat{y}_{R_j} is the mean response for the training observations within the j th box.

Building a Regression Tree

- ▶ Unfortunately, it is computationally infeasible to consider every possible partition of the feature space into J boxes.
- ▶ For this reason, we take a **top-down**, **greedy** approach that is known as **Recursive Binary Splitting**.
- ▶ The approach is **top-down** because it begins at the top of the tree and then successively splits the predictor space; each split is indicated via two new branches further down on the tree.
- ▶ It is **greedy** because at each step of the tree-building process, the **best** split is made at that particular step, rather than looking ahead and picking a split that will lead to a better tree in some future step.

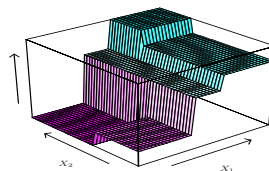
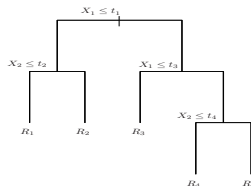
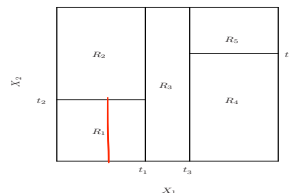
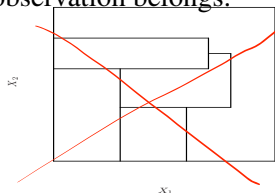
1. region selection
2. predictor selection
3. cutpoint selection

Building a Regression Tree

- ▶ We first select the predictor X_j and the cutpoint s such that splitting the predictor space into the regions $\{X_j|X_j < s\}$ and $\{X_j|X_j \geq s\}$ leads to the greatest possible reduction in RSS.
- ▶ Next, we repeat the process, looking for the best predictor and best cutpoint in order to split the data further so as to minimize the RSS within each of the resulting regions.
- ▶ However, this time, instead of splitting the entire predictor space, we split one of the two previously identified regions. We now have three regions.
- ▶ Again, we look to split one of these three regions further, so as to minimize the RSS. The process continues until a stopping criterion is reached; for instance, we may continue until no region contains more than five observations.

Prediction and Examples

- We predict the response for a given test observation using the mean of the training observations in the region to which that test observation belongs.



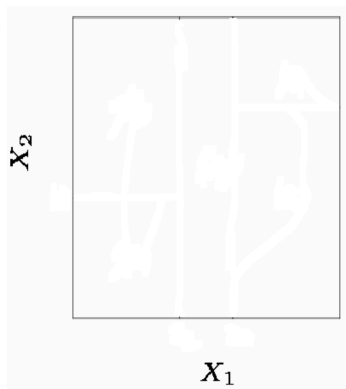
Details of previous figure

- ▶ **Top Left:** A partition of two-dimensional feature space that could not result from recursive binary splitting.
- ▶ **Top Right:** The output of recursive binary splitting on a two-dimensional example.
- ▶ **Bottom Left:** A tree corresponding to the partition in the top right panel.
- ▶ **Bottom Right:** A perspective plot of the prediction surface corresponding to that tree.

Building a Regression Tree: Example

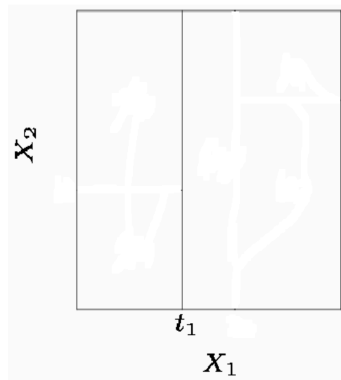
Recursive binary splitting

- ▶ Generally we create the partitions by iteratively splitting one of the X variables into two regions.
- ▶ First split on $X_1 = t_1$.
- ▶ If $X_1 < t_1$, split on $X_2 = t_2$.
- ▶ If $X_1 > t_1$, split on $X_1 = t_3$.
- ▶ If $X_1 > t_3$, split on $X_2 = t_4$.
- ▶ When we create partitions this way we can always represent them using a tree structure.



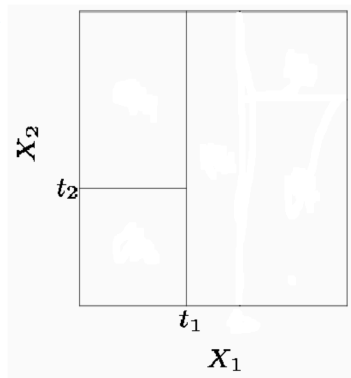
Building a Regression Tree: Example

- ▶ Generally we create the partitions by iteratively splitting one of the X variables into two regions.
- ▶ First split on $X_1 = t_1$.
- ▶ If $X_1 < t_1$, split on $X_2 = t_2$.
- ▶ If $X_1 > t_1$, split on $X_1 = t_3$.
- ▶ If $X_1 > t_3$, split on $X_2 = t_4$.
- ▶ When we create partitions this way we can always represent them using a tree structure.



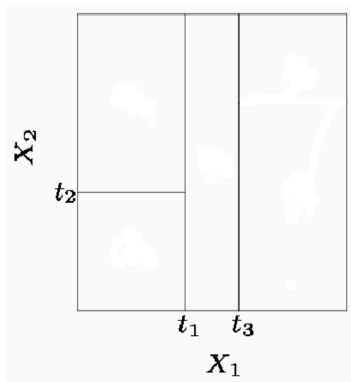
Building a Regression Tree: Example

- ▶ Generally we create the partitions by iteratively splitting one of the X variables into two regions.
- ▶ First split on $X_1 = t_1$.
- ▶ If $X_1 < t_1$, split on $X_2 = t_2$.
- ▶ If $X_1 > t_1$, split on $X_1 = t_3$.
- ▶ If $X_1 > t_3$, split on $X_2 = t_4$.
- ▶ When we create partitions this way we can always represent them using a tree structure.



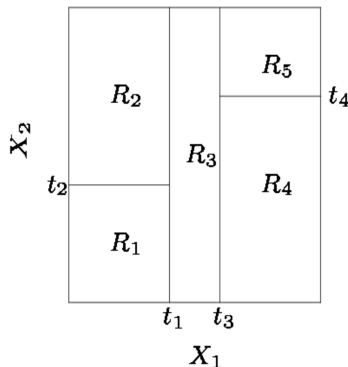
Building a Regression Tree: Example

- ▶ Generally we create the partitions by iteratively splitting one of the X variables into two regions.
- ▶ First split on $X_1 = t_1$.
- ▶ If $X_1 < t_1$, split on $X_2 = t_2$.
- ▶ If $X_1 > t_1$, split on $X_1 = t_3$.
- ▶ If $X_1 > t_3$, split on $X_2 = t_4$.
- ▶ When we create partitions this way we can always represent them using a tree structure.



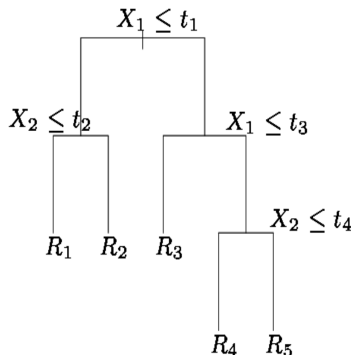
Building a Regression Tree: Example

- ▶ Generally we create the partitions by iteratively splitting one of the X variables into two regions.
- ▶ First split on $X_1 = t_1$.
- ▶ If $X_1 < t_1$, split on $X_2 = t_2$.
- ▶ If $X_1 > t_1$, split on $X_1 = t_3$.
- ▶ If $X_1 > t_3$, split on $X_2 = t_4$.
- ▶ When we create partitions this way we can always represent them using a tree structure.



Building a Regression Tree: Example

- ▶ Generally we create the partitions by iteratively splitting one of the X variables into two regions.
- ▶ First split on $X_1 = t_1$.
- ▶ If $X_1 < t_1$, split on $X_2 = t_2$.
- ▶ If $X_1 > t_1$, split on $X_1 = t_3$.
- ▶ If $X_1 > t_3$, split on $X_2 = t_4$.
- ▶ When we create partitions this way we can always represent them using a tree structure.



Summary and Remark

- ▶ Introduction
- ▶ Regression Tree
- ▶ Read textbook Chapter 9 and R code
- ▶ Do R lab