Regression Model Trees

Nate Wells

Math 243: Stat Learning

November 13th, 2020

Outline

In today's class, we will...

- Discuss model trees as a more flexible alternative to simple regression trees

Section 1

Model Trees

Motivation

One obvious limitation of regression trees is that each terminal node assigns each observation in a region *the same* prediction, based on the average response in that region.

Motivation

One obvious limitation of regression trees is that each terminal node assigns each observation in a region *the same* prediction, based on the average response in that region.

- This can be especially problematic if observations in the region have outliers.

Motivation

One obvious limitation of regression trees is that each terminal node assigns each observation in a region *the same* prediction, based on the average response in that region.

- This can be especially problematic if observations in the region have outliers.
- The fix? Rather than having each terminal node end in an constant prediction, we have nodes end with simple models that can be used to make predictions.

Motivation

One obvious limitation of regression trees is that each terminal node assigns each observation in a region *the same* prediction, based on the average response in that region.

- This can be especially problematic if observations in the region have outliers.
- The fix? Rather than having each terminal node end in an constant prediction, we have nodes end with simple models that can be used to make predictions.

This approach is called a **model tree**. One example algorithm is called **M5**, developed by Quinlan in 1995. It differs from CART in a few ways:

Motivation

One obvious limitation of regression trees is that each terminal node assigns each observation in a region *the same* prediction, based on the average response in that region.

- This can be especially problematic if observations in the region have outliers.
- The fix? Rather than having each terminal node end in an constant prediction, we have nodes end with simple models that can be used to make predictions.

This approach is called a **model tree**. One example algorithm is called **M5**, developed by Quinlan in 1995. It differs from CART in a few ways:

1. Different splitting criteria
2. Terminal nodes are models
3. Predictions are made using averages of models along the decision path

The M5 algorithm

- To begin, an initial split is found by searching all values of all predictors for pair that gives greatest reduction in node's error rate.
- For an initial data set S and partitions S_1 and S_2, with n_1 and n_2 observations each, and standrad deviations $SD(S_1)$ and $SD(S_2)$:

$$\text{reduction} = SD(S) - \frac{n_1}{n_1 + n_2} SD(S_1) - \frac{n_2}{n_1 + n_2} SD(S_2)$$

The M5 algorithm

- To begin, an initial split is found by searching all values of all predictors for pair that gives greatest reduction in node's error rate.
- For an initial data set S and partitions S_1 and S_2, with n_1 and n_2 observations each, and standrad deviations $SD(S_1)$ and $SD(S_2)$:

$$\text{reduction} = SD(S) - \frac{n_1}{n_1 + n_2} SD(S_1) - \frac{n_2}{n_1 + n_2} SD(S_2)$$

- An SLR model is created on each subset using the predictor chosen for split.

The M5 algorithm

- To begin, an initial split is found by searching all values of all predictors for pair that gives greatest reduction in node's error rate.
- For an initial data set S and partitions S_1 and S_2, with n_1 and n_2 observations each, and standrad deviations $SD(S_1)$ and $SD(S_2)$:

$$\text{reduction} = SD(S) - \frac{n_1}{n_1 + n_2} SD(S_1) - \frac{n_2}{n_1 + n_2} SD(S_2)$$

- An SLR model is created on each subset using the predictor chosen for split.
- The process repeats on each subset to create linear models that depend on the current split variable and all predecessor variables

The M5 algorithm

- To begin, an initial split is found by searching all values of all predictors for pair that gives greatest reduction in node's error rate.
- For an initial data set S and partitions S_1 and S_2, with n_1 and n_2 observations each, and standrad deviations $SD(S_1)$ and $SD(S_2)$:

$$\text{reduction} = SD(S) - \frac{n_1}{n_1 + n_2} SD(S_1) - \frac{n_2}{n_1 + n_2} SD(S_2)$$

- An SLR model is created on each subset using the predictor chosen for split.
- The process repeats on each subset to create linear models that depend on the current split variable and all predecessor variables
- The algorithm terminates once a predetermined stopping condition is met

The M5 algorithm

- To begin, an initial split is found by searching all values of all predictors for pair that gives greatest reduction in node's error rate.
- For an initial data set S and partitions S_1 and S_2, with n_1 and n_2 observations each, and standrad deviations $SD(S_1)$ and $SD(S_2)$:

$$\text{reduction} = SD(S) - \frac{n_1}{n_1 + n_2} SD(S_1) - \frac{n_2}{n_1 + n_2} SD(S_2)$$

- An SLR model is created on each subset using the predictor chosen for split.
- The process repeats on each subset to create linear models that depend on the current split variable and all predecessor variables
- The algorithm terminates once a predetermined stopping condition is met
- Each model (one for each node of the tree) is regularized in order to drop some extraneous predictors.

The M5 algorithm

- To begin, an initial split is found by searching all values of all predictors for pair that gives greatest reduction in node's error rate.
- For an initial data set S and partitions S_1 and S_2, with n_1 and n_2 observations each, and standrad deviations $SD(S_1)$ and $SD(S_2)$:

$$\text{reduction} = SD(S) - \frac{n_1}{n_1 + n_2} SD(S_1) - \frac{n_2}{n_1 + n_2} SD(S_2)$$

- An SLR model is created on each subset using the predictor chosen for split.
- The process repeats on each subset to create linear models that depend on the current split variable and all predecessor variables
- The algorithm terminates once a predetermined stopping condition is met
- Each model (one for each node of the tree) is regularized in order to drop some extraneous predictors.
- To make predictions for an observation in particular terminal node, a weighted average of models is used.

The algorithm in picture

The algorithm in R

We use the M5P function in the RWeka package to fit model trees

```
set.seed(1)
library(pdxTrees)
my_trees <- get_pdxTrees_parks() %>% sample_n(1000) %>%
  select(Pollution_Removal_oz, Tree_Height, Crown_Base_Height, Condition) %>%
  drop_na() %>% mutate_if( is.character, as.factor)
```

The algorithm in R

We use the M5P function in the RWeka package to fit model trees

```
set.seed(1)
library(pdxTrees)
my_trees <- get_pdxTrees_parks() %>% sample_n(1000) %>%
  select(Pollution_Removal_oz, Tree_Height, Crown_Base_Height, Condition) %>%
  drop_na() %>% mutate_if( is.character, as.factor)
```

```
library(RWeka)
m5tree <- M5P(Pollution_Removal_oz ~., data = my_trees)
summary(m5tree)
```

```
##
## === Summary ===
##
## Correlation coefficient                  0.6924
## Mean absolute error                      7.7385
## Root mean squared error                 12.1346
## Relative absolute error                 61.0104 %
## Root relative squared error             73.0128 %
## Total Number of Instances              984
```

M5 Model Tree

```
library(partykit)
plot(m5tree, gp = gpar(fontsize = 6))
```