40.016 The Analytics Edge

Forecasting the Supreme Court's decisions with CARTs (Part 2)

Stefano Galelli

SUTD

Term 6, 2020

Outline

- Brief recap on CARTs
- Pruning
- Trees versus linear models
- Advantages and Disadvantages of CARTs

Brief recap on CARTs

- Decision Trees can be applied to both regression and classification problems
- The term **Classification And Regression Tree (CART)** is used to refer to procedures that learn a Classification or Regression Tree

Brief recap on CARTs

Structure of a Decision Tree

Brief recap on CARTs

Learning algorithm

Two main steps (regression):

1. Divide the predictor space into J distinct and non-overlapping regions $(R_1, R_2, \ldots, R_J)$. This process uses **recursive binary splitting** and minimizes the RSS.

Brief recap on CARTs

Learning algorithm

Two main steps (regression):

1. Divide the predictor space into J distinct and non-overlapping regions $(R_1, R_2, \ldots, R_J)$. This process uses **recursive binary splitting** and minimizes the RSS.

Brief recap on CARTs

Learning algorithm

Two main steps (regression):

1. Divide the predictor space into J distinct and non-overlapping regions $(R_1, R_2, \ldots, R_J)$. This process uses **recursive binary splitting** and minimizes the RSS.
2. For every observation that falls into the j-th region R_j, we make the same prediction c_j (mean of the response values for the training observations in R_j).

Brief recap on CARTs

Learning algorithm

For Classification Trees we use the same procedure, but:

1. We use a measure of impurity (instead of RSS) in the partitioning process.
2. The prediction c_j is the most commonly occurring class.

Back to R!

To learn CARTs, we will use the function `rpart`, implemented in the package ... `rpart`:

```
rpart(formula, data, method, control, ...)
```

Pruning

- The CART's learning algorithm is likely to build complex trees that overfit the training data.
- A smaller tree (with fewer region $R_1, R_2, \ldots, R_J$) may lead to lower *variance* at the cost of a little *bias*.

Pruning

- The CART's learning algorithm is likely to build complex trees that overfit the training data.
- A smaller tree (with fewer region $R_1, R_2, \ldots, R_J$) may lead to lower *variance* at the cost of a little *bias*.

Pruning

Idea:

1. Grow a large tree T_0, and then
2. *Prune* it back to obtain a *subtree*

Some considerations:

- To determine how to prune the tree, we can use the *cross-validation error*
- We cannot calculate the cross-validation error for all trees, because there are too many subtrees –> it would take too long!

Pruning

Cost complexity pruning (or **weakest link pruning**)

For each value of a nonnegative tuning parameter α, there corresponds a subtree $T \subset T_0$ s.t. the value

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

is as small as possible.

Pruning

Cost complexity pruning (or **weakest link pruning**)

For each value of a nonnegative tuning parameter α, there corresponds a subtree $T \subset T_0$ s.t. the value

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

is as small as possible.

Note:

- When α increases, we pay a price for building a tree with many leaves
- The above expression is a reminiscent of the LASSO, which controls the complexity of a linear model (Week 5, Lecture 2)

Pruning

(Full) algorithm for building a Regression Tree:

1. Use *recursive binary splitting* to grow a large tree on the training data
2. Use *cost complexity pruning* to obtain a sequence of subtrees as a function of α
3. Use *k-fold cross-validation* to choose the best value of α
4. Return the subtree from Step 2 that corresponds to the chosen value of α

Pruning

How do we prune a Classification Tree?

We follow the same procedure, keeping in mind that the model error is calculated with a **measure of impurity**. This leads to the following expression

$$\sum_{m=1}^{T} E_m + \alpha|T|$$

which we still want to minimize.

Trees versus linear models

Let's compare a linear regression model

$$f(X) = \beta_0 + \sum_{j=1}^{p} X_j \beta_j$$

to a regression tree

$$f(X) = \sum_{m=1}^{M} c_m \cdot 1_{X \in R_m}$$

Trees versus linear models

Which model is better? The answer depends on the problem at hand.

Example:

Advantages and Disadvantages of CARTs

Pros:

- Interpretability
- Can be displayed graphically
- Can handle qualitative predictors (that take no continuous values)
- No assumptions on the relationship between input and output variables

Cons:

- They are not very accurate
- Not robust

References

- Martin et al. (2004) Competing approaches to predicting supreme court decision making. *Perspectives on Politics*, 2 (4), 761–767.
- James et al. (2014) *An Introduction to Statistical Learning with Applications in R*, Springer, 2014. Chapter 8.1.