40.016 The Analytics Edge

Forecasting the Supreme Court's decisions with CARTs (Part 1)

Stefano Galelli

SUTD

Term 6, 2020

Course overview

Domains: Wine analytics, Challenger, Framingham Heart Study, Oscars, Sports, Economics, Lex Analytics, Ethics in Analytics, Text Analytics, Netflix, Aviation.

Tools: Linear Regression, Principal Component Analysis, Logistic Regression, Multinomial Logit, Model Selection, Classification and Regression Trees, Random Forests, Naïve Bayes Classifier, Clustering, Optimization.

Outline

- Brief Introduction to the US Supreme Court
- The Supreme Court Forecasting Project
- Decision Trees
- Regression Trees
- Classification Trees
- Advantages and Disadvantages of CARTs

Brief Introduction to the US Supreme Court

What is the Supreme Court? See:
https://www.youtube.com/watch?v=QVIVEKY5YWI

Brief Introduction to the US Supreme Court

What is the Supreme Court? See:
https://www.youtube.com/watch?v=QVIVEKY5YWI

Key points:

- Nine justices, or judges, appointed by the US President
- Lifetime tenure
- The court handles $\sim$ 80 cases per year
- A decision happens when the majority agrees on an outcome

Brief Introduction to the US Supreme Court

How does a case get to the Supreme Court? See: https://www.youtube.com/watch?v=KEjgAXxrkXY

Categories for case selection:

- Cases of national importance
- Lower court invalidates federal law
- Resolve split decision

A **key point**: The decision is to affirm or reverse, so we can model it as a *binary variable*.

The Supreme Court Forecasting Project

This is a study published by Martin et al. (2004), who:

- Used data spanning the period 1994-2001 (longest period with the same justices)
- Compared predictions (for the year 2002) made by legal experts and statistical models
- Found very interesting results:
  - Accuracy on the entire court decision –> models, 75%; experts, 59.1%
  - Accuracy at the individual justice level –> models, 66.7%; experts, 67.9%

The Supreme Court Forecasting Project

Our data:

- 623 observations (about 80 cases per year), 20 variables
- Output variable, or predictand: `result`, which takes value 0 (liberal) or 1 (conservative)
- Input variables, or predictors:
  - `petit`: petitioner type (e.g., US, employer, injured person)
  - `respon`: type of respondent
  - `circuit`: circuit of origin of the case
  - `unconst`: whether the petitioner argued the constitutionality of a law of practice
  - `lctdir`: ideaological direction of the lower court (liberal or conservative)
  - `issue`: issue area of the case

Decision Trees

- 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
- **Note:** We begin by considering Regression Trees

Regression Trees

Intuition: Suppose we are working on a regression problem with response variable Y and predictors X_1 and X_2. The underlying idea of Regression Trees is to divide, or partition, the predictor space into a number of regions, where we then apply a simple model.

Example:

Regression Trees

How do we learn a Regression Tree?

There are two main steps:

1. Divide the predictor space into J distinct and non-overlapping regions $(R_1, R_2, \ldots, R_J)$
2. For every observation that falls into the j-th region R_j, we make the same prediction c_j. Specifically, we take the mean of the response values for the training observations in R_j, that is

$$c_j = \text{ave}(y_i | x_i \in R_j).$$

(The choice of using the mean of y_i in region R_j is based on the criterion minimization of the sum of squares.)

Regression Trees

How do we learn a Regression Tree (cont'd)?

The problem of partitioning the predictor space into J regions (i.e., Step 1 in the previous) can be formulated as follows:

$$\min_{R_1,\ldots,R_J} \sum_{j=1}^{J} \sum_{i \in R_j} (y_i - c_j)^2.$$

In general, the problem is computationally unfeasible. To solve it, we use a top-down, greedy approach known as **recursive binary splitting**.

Regression Trees

How do we learn a Regression Tree (cont'd)?

Recursive binary splitting (starting with all variables in one region):

- Consider all predictors $X_1, \ldots, X_p$ and all possible values of the cut (or split) point s, and choose the predictor and cut point s.t. the resulting partition has the lowest RSS. More precisely, we define the following half-planes

$$R_1(j,s) = \{X | X_j < s\} \text{ and } R_2(j,s) = \{X | X_j \geq s\},$$

And we seek the value of j and s that minimizes

$$\sum_{i:x_i \in R_1(j,s)} (y_i - c_1)^2 + \sum_{i:x_i \in R_2(j,s)} (y_i - c_2)^2.$$

Regression Trees

How do we learn a Regression Tree (cont'd)?

Recursive binary splitting (starting with all variables in one region):

- Consider all predictors $X_1, \ldots, X_p$ and all possible values of the cut (or split) point s, and choose the predictor and cut point s.t. the resulting partition has the lowest RSS.
- We repeat the process, this time splitting one of the two previously identified regions. The process continues until an exit condition is met (e.g., minimum number of points in each region).

Classification Trees

Two differences w.r.t. Regression Trees:

- For each region, the prediction c_j is the **most commonly occurring class**
- When learning a tree, we cannot use the RSS. Instead, we use a **measure of impurity** (a split is pure if, for all branches, all the instances choosing a branch fall within the same class)

Classification Trees

Measures of impurity

1. Classification error rate:

$$E = 1 - \max_k(p_{mk})$$

where p_{mk} is the proportion of training observations in the m-th region that are from the k-th class.

Classification Trees

Measures of impurity (cont'd)

2. Gini index:

$$G = \sum_{k=1}^{K} p_{mk}(1 - p_{mk})$$

where K is the total number of classes, and G varies between 0 and 0.5.

Classification Trees

Measures of impurity (cont'd)

3. Entropy:

$$D = -\sum_{k=1}^{K}(p_{mk}\log(p_{mk}))$$

.

Since $0 \leq p_{mk} \leq 1$, and D varies between 0 and 1.

Back to R!

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

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

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.