Big Data and Security

Jeffrey Borowitz, PhD
Lecturer
Sam Nunn School of International Affairs

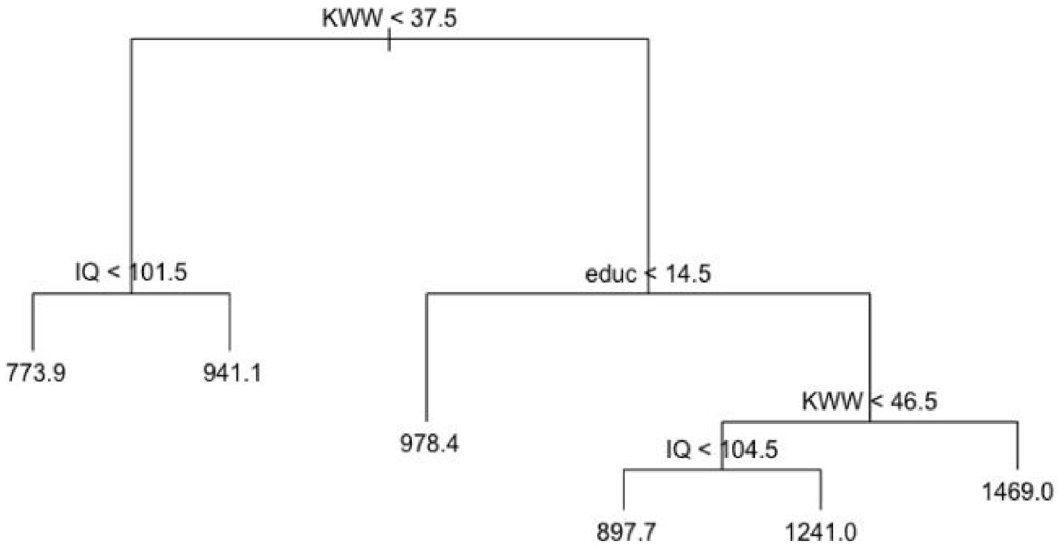
Trees

Decision Trees

- A very simple (mostly) nonparametric model
- You have X (maybe a bunch of them) and Y
    - Start with the average of Y and calculate the average squared difference between y_i and this average
    - Split the data in half in whatever way decreases this error the most
    - Look at each of your split areas and split again, potentially on a different variable
    - Keep going until splitting doesn't make as much difference
- At the end, you have a decision tree which allows a prediction of $\hat{y}$ by answering questions
- Note: this is not strictly non-parametric (in that it can't fit all possible functions of X)

Georgia Tech

A Decision Tree for Wages

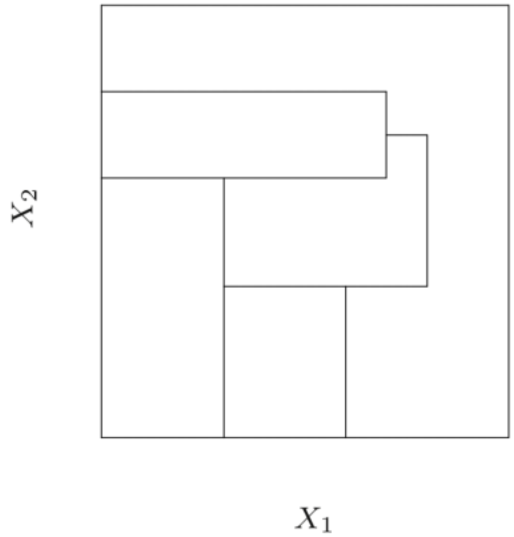


Trees Fit Data Better Than Regression

- Instead of explaining as much as possible with a line, trees are minimizing squared error whenever they split
- So they are often better predictors of $\hat{y}$ than regression
- How does this fit with the “correlation replaces causality” argument

Assumptions

- While trees don't assume specific functional forms, there are some shapes of data you can't reach by this process
- There are no smooth changes to the data
  - This is the **opposite** of regression



Trees and Overfitting

- Trees have the same problems with overfitting as lots of things
- So you want to use cross-validation, or optimize against AIC, or something

Random Forest

- One common “something” to avoid overfitting is a method called a random forest
- The procedure:
  - Take a random set of the data
  - Fit a tree
  - Repeat several times
  - Average the results
- Random forests work really well in many practical applications, and if tuned properly can avoid overfitting
- Random forests can be fit with a command from the randomForest package:
  - forest<- randomForest(inlf ~ educ + exper + expersq , data=wages)
  - data$predicted.inlf<- predict(forest, data, predict.all=TRUE)

Gradient Boosted Trees

- Once we fit a single tree, we can compute whether the tree was "right" or "wrong" for each data point.
- In gradient boosted trees, after fitting the first tree, we fit a second tree to a "modified" problem
  - In the modified problem, we increase the weight placed in our loss function on the points we missed last time.
  - We fit the problem again with these new weights, and combine the predictions from the latest fit with the last fit by averaging
- We then repeat: increase weight on the points that the initial and second model combined missed
- Performance:
  - Gradient boosted trees perform really well, by construction!
  - But they are again susceptible to overfitting (indeed, their performance comes from putting more weight on the "outliers")

Lesson Summary

- Decision trees are a type of non-parametric model perform prediction by asking binary questions about 1 variable at a time
- Random Forests are groups of trees fit together with different independent variables
- Gradient Boosted Trees are groups of trees fit together where more weight is given to fitting mistakes in new trees