The Supreme Court

An Introduction to Trees

Universal AI

The American Legal System

- The legal system of the United States operates at the state level and at the federal level
- Federal courts hear cases beyond the scope of state law
- Federal courts are divided into:
  - **District Courts**
    - Makes initial decision
  - **Circuit Courts**
    - Hears appeals from the district courts
  - **Supreme Court**
    - Highest level – makes final decision

The Supreme Court of the United States

- Consists of nine justices, appointed by the President
  - Justices are distinguished judges, professors of law, state and federal attorneys
- The Supreme Court of the United States (SCOTUS) decides on most difficult and controversial cases
  - Often involve interpretation of Constitution
  - Significant social, political and economic consequences

Notable SCOTUS Decisions

- Wickard v. Filburn (1942)
  - Congress allowed to intervene in industrial/economic activity
- Roe v. Wade (1973)
  - Legalized abortion
- Bush v. Gore (2000)
  - Decided outcome of presidential election!
- National Federation of Independent Business v. Sebelius (2012)
  - Patient Protection and Affordable Care Act ("ObamaCare") upheld the requirement that individuals must buy health insurance

Predicting Supreme Court Cases

- Legal academics and political scientists regularly make predictions of SCOTUS decisions from detailed studies of cases and individual justices
- In 2002, Andrew Martin, a professor of political science at Washington University in St. Louis, decided to instead predict decisions using a statistical model built from data
- Together with his colleagues, he decided to test this model against a panel of experts

Predicting Supreme Court Cases

- Martin used a method called Classification and Regression Trees (CART)
- Why CART?
  - Other models are generally not *interpretable*
  - Model coefficients indicate importance and relative effect of variables, but do not give a simple explanation of how decision is made

Data

- Cases from 1994 through 2001
- In this period, same nine justices presided SCOTUS
  - Breyer, Ginsburg, Kennedy, O'Connor, Rehnquist (Chief Justice), Scalia, Souter, Stevens, Thomas
  - Rare data set – longest period of time with the same set of justices in over 180 years
- We will focus on predicting Justice Stevens' decisions
  - Started out moderate, but became more liberal
  - Self-proclaimed conservative

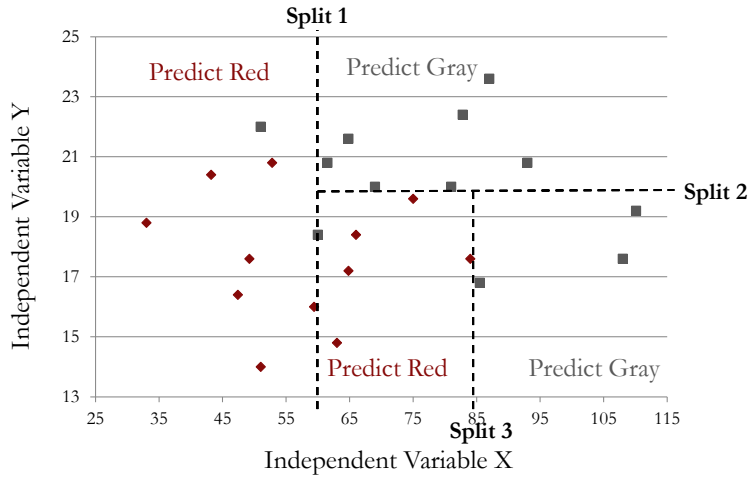
Variables

- **Dependent Variable**: Did Justice Stevens vote to reverse the lower court decision? 1 = reverse, 0 = affirm
- **Independent Variables**: Properties of the case
  - Circuit court of origin (1st – 11th, DC, FED)
  - Issue area of case (e.g., civil rights, federal taxation)
  - Type of petitioner, type of respondent (e.g., US, an employer)
  - Ideological direction of lower court decision (conservative or liberal)
  - Whether petitioner argued that a law/practice was unconstitutional

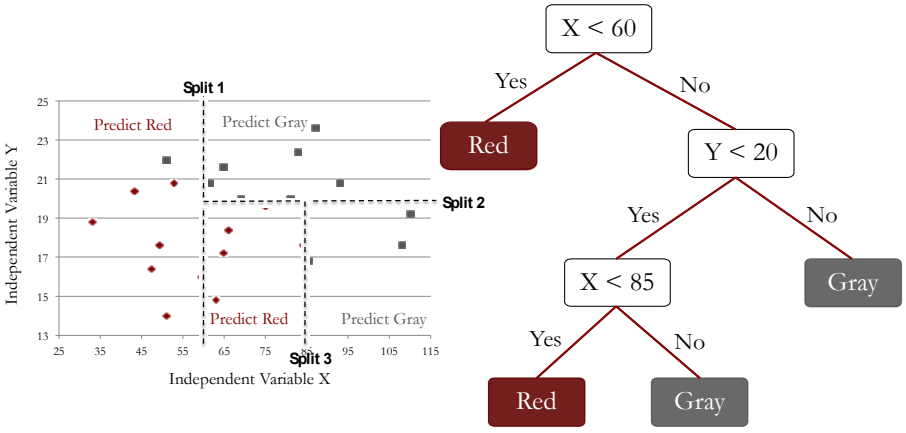
Classification and Regression Trees (CART)

- Build a tree by **splitting on independent variables**
- To predict the outcome for an observation, **follow the splits** and at the end, predict the most frequent outcome in the training set
- Does not assume a linear model
- Interpretable

Splits in CART



Final Tree



When Does CART Stop Splitting?

- There are different ways to control how many splits are generated
  - One way is by setting a lower bound for the number of points in each subset
- This parameter is called **minbucket**
  - The smaller it is, the more splits will be generated
  - If it is too small, overfitting will occur
  - If it is too large, model will be too simple and accuracy will be poor

Parameter Selection

- How should we set the minbucket parameter?
- One way is to select the value that gives the best testing set accuracy
  - This is not right!
- There are two standard ways of setting parameter values:
  - Validation set
  - Cross-validation

Complexity Parameter

- One way to limit the size of our tree is with minbucket
- There is another option called the complexity parameter (**cp**)
- Like Adjusted R^2
  - Measures trade-off between model complexity and accuracy on the training set
- Smaller cp leads to a bigger tree (might overfit)

Predictions from CART

- At each leaf of the tree, we have a bucket of observations, which may contain both outcomes (i.e., affirm and reverse)
- For each leaf, we can compute percentage of points in one group
  - Example: 10 affirm, 2 reverse → 10/(10+2) = 0.867
- We use a threshold to obtain a prediction
  - Default threshold of 0.5 corresponds to picking most frequent outcome

Random Forests

- Designed to improve prediction accuracy of CART
- Works by building a large number of CART trees
  - Makes model less interpretable
- To make a prediction for a new observation, each tree "votes" on the outcome, and we pick the outcome that receives the majority of the votes

Building Many Trees

1. Each tree can split on only a random subset of the variables
2. Each tree is built from a "bagged"/"bootstrapped" sample of the data
   - Select observations randomly with replacement
   - Example – original data: 1 2 3 4 5
   - New "data":
     1. 2 3 1 2 5
     2. 3 1 4 5 1
     3. 4 4 2 1 5

Random Forest Parameters

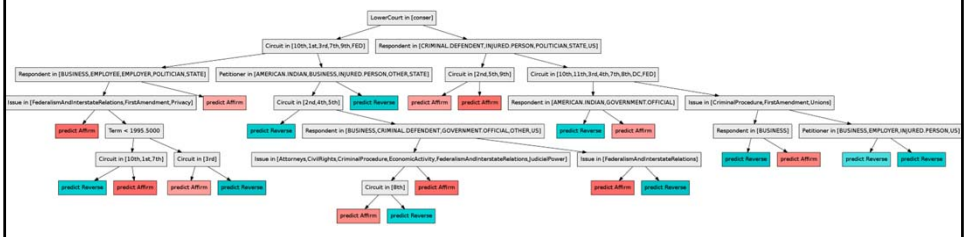
- Minimum number of observations in a subset
  - In R, this is controlled by the nodesize parameter
  - Smaller nodesize may take longer in R
- Number of trees
  - In R, this is the ntree parameter
  - Should not be too small, because bagging procedure may miss observations
  - More trees take longer to build

Gradient Boosted Decision Trees

- Similar to Random Forest, addresses limitation of CART by mixing many trees
- Random Forest trains trees **independently** and averages the result
- Boosting trains trees **sequentially** where each tree is trained to predict the mistakes of previous trees
- Many good libraries: XGBoost, LightGBM

Random Forests/Boosting vs CART

- CART after validation:
  - Test set accuracy: 54.6%, test set AUC: 0.619



- Black box after validation:
  - Test set accuracy: 61.0%, test set AUC: 0.648

Random Forests/Boosting vs CART

Black-box models

Patient info

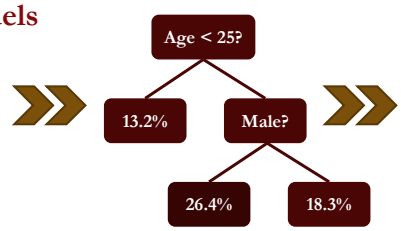
Age:	30
Gender:	male
Albumin:	2.8g/dL
Sepsis:	none
INR:	1.1
Diabetic:	yes
…	

Mortality risk: 26.4%

Interpretable models

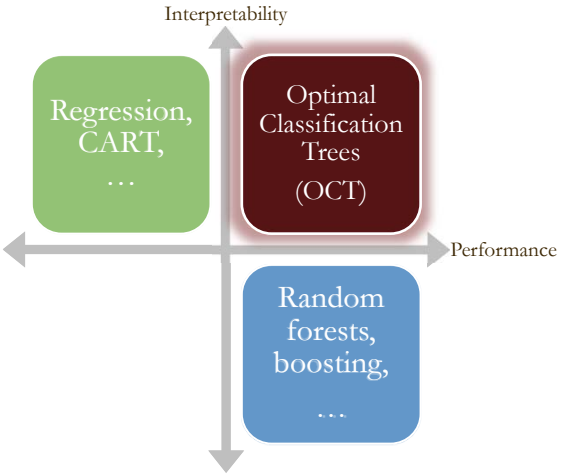
Patient info

Age:	30
Gender:	male
Albumin:	2.8g/dL
Sepsis:	none
INR:	1.1
Diabetic:	yes
…	



Mortality risk: 26.4%

Random Forests/Boosting vs CART



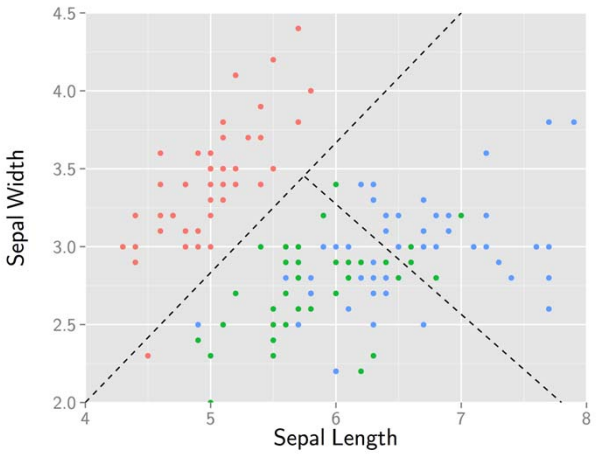
- Typically black-boxes outperform interpretable models
- OCT is competitive with black-boxes and is interpretable

Optimal Trees

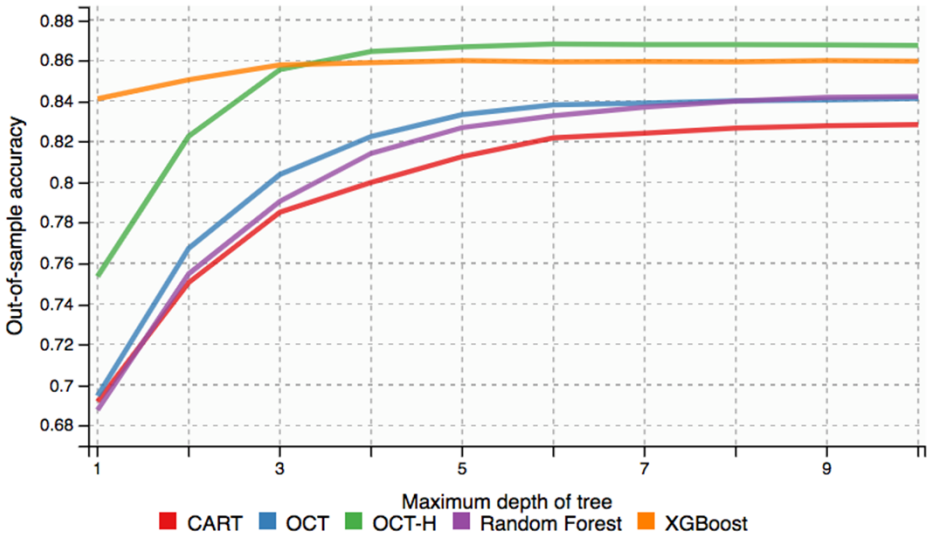
- CART's greedy training means splits are only are only locally-optimal, overall tree could be far from optimal
- We have developed a new method for finding optimal decision trees:
  - Bertsimas and Dunn. "Optimal Classification Trees". Machine Learning, 2017
- Our method uses modern optimization techniques to train the entire tree in one step, rather than split-by-split like existing methods

Variants of Optimal Trees

- **OCT**: trees with parallel splits (one variable per split)
- **OCT-H**: trees with hyperplane splits (can use multiple variables per split if beneficial)



Performance of OCT



Summary of Optimal Trees

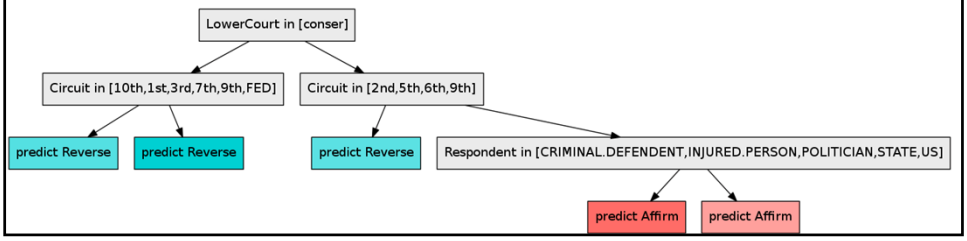
- Practitioners often have to choose between interpretability (CART) or performance (random forest)
- Optimal Trees is a new method that maintains interpretability but delivers state-of-the-art performance

Random Forests vs CART vs OCT

- CART after validation:
  - Test set accuracy: 54.6%, test set AUC: 0.619

- Black box after validation:
  - Test set accuracy: 61.0%, test set AUC: 0.648

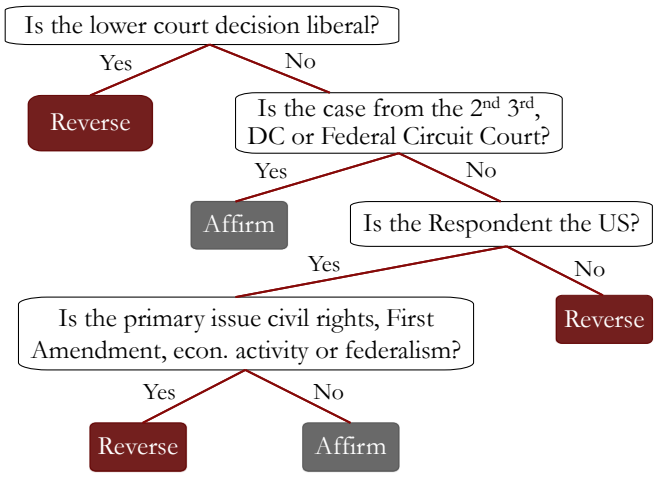
- OCT after validation: test set accuracy: 63.8%, test set AUC: 0.670



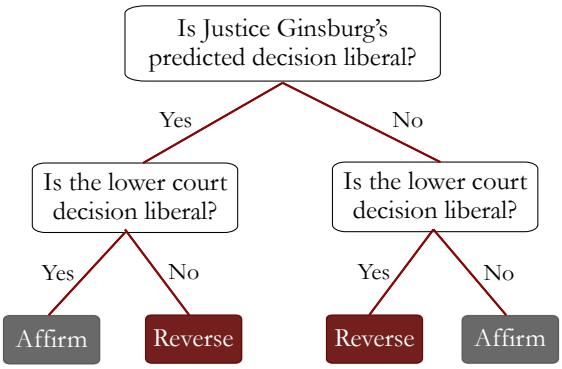
Martin's Model

- Used 628 previous SCOTUS cases between 1994 and 2001
- Made predictions for the 68 cases that would be decided in October 2002, before the term started
- Two stage approach based on CART:
  - First stage: one tree to predict a unanimous liberal decision, other tree to predict unanimous conservative decision
    - If conflicting predictions or predict no, move to next stage
  - Second stage consists of predicting decision of each individual justice, and using majority decision as prediction

Tree for Justice O'Connor



Tree for Justice Souter



"Make a liberal decision" "Make a conservative decision"

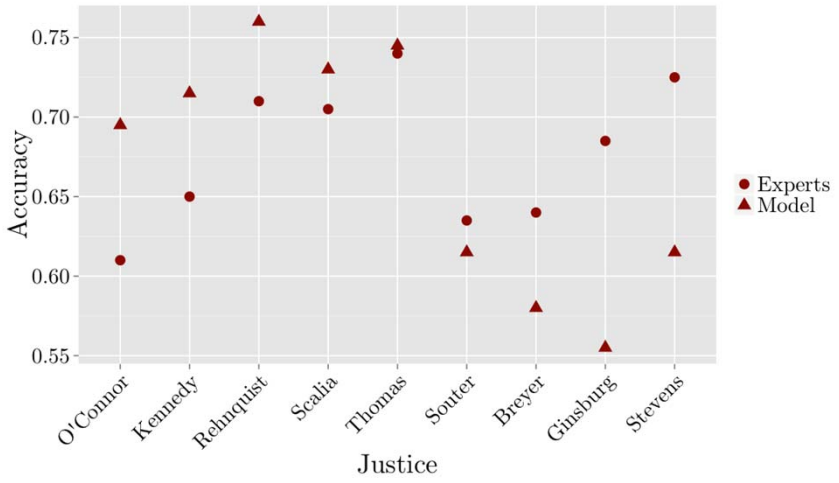
The Experts

- Martin and his colleagues recruited 83 legal experts
  - 71 academics and 12 attorneys
  - 38 previously clerked for a Supreme Court justice, 33 were chaired professors and 5 were current or former law school deans
- Experts only asked to predict within their area of expertise; more than one expert to each case
- Allowed to consider any source of information, but not allowed to communicate with each other regarding predictions

The Results

- For the 68 cases in October 2002:
- Overall case predictions:
  - Model accuracy: 75%
  - Experts accuracy: 59%
- Individual justice predictions:
  - Model accuracy: 67%
  - Experts accuracy: 68%

Individual Justice Predictions



The Analytics Edge

- Predicting Supreme Court decisions is very valuable to firms, politicians and non-governmental organizations
- A model that predicts these decisions can be more accurate and faster than experts
  - CART model based on very high-level details of case beats experts who can process much more detailed and complex information

Cross Validation

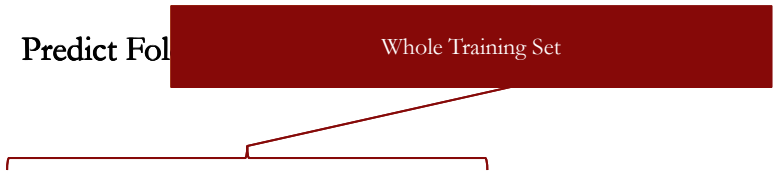
This is another approach to selecting good parameter values.

K-fold Cross-Validation

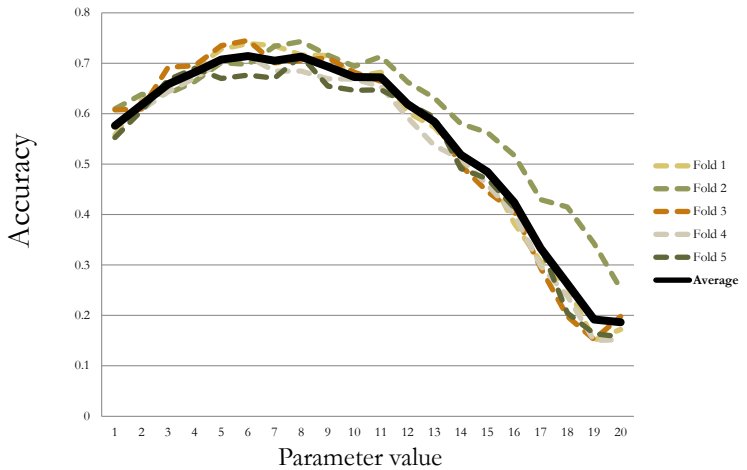
- One way to properly select a parameter's value is to use k-fold cross-validation
  - Given training set, split into k pieces ("folds")
  - Use k-1 folds to estimate a model, and test model on remaining one fold ("validation set") for each candidate parameter value
  - Repeat for each of the k folds
  - For each candidate parameter value, average accuracy over the k folds, or validation sets

K-fold Cross-Validation Graphically

- Assume five folds (k = 5)



Output of k-fold Cross-Validation



Output of k-fold Cross-Validation