Random Forests

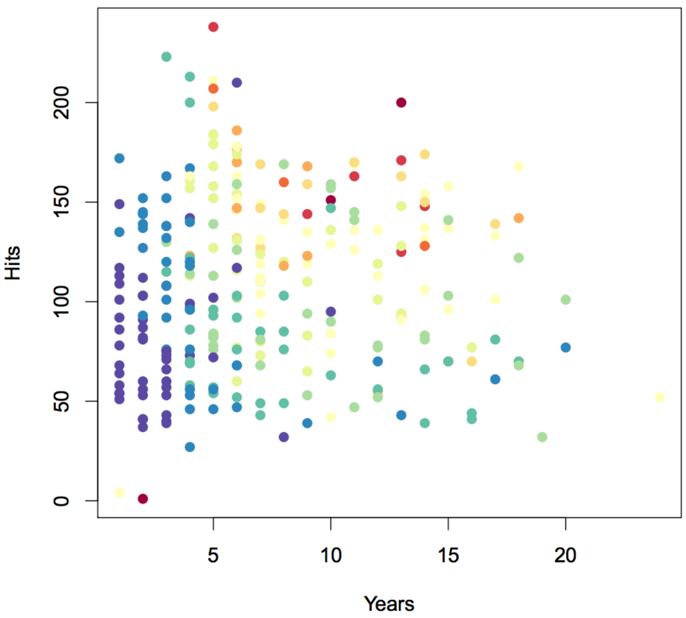
Overview

- Review of Decision Trees
  - Regression
  - Classification
- Bagging
- Random Forest
  - OOB Error
  - Feature importance

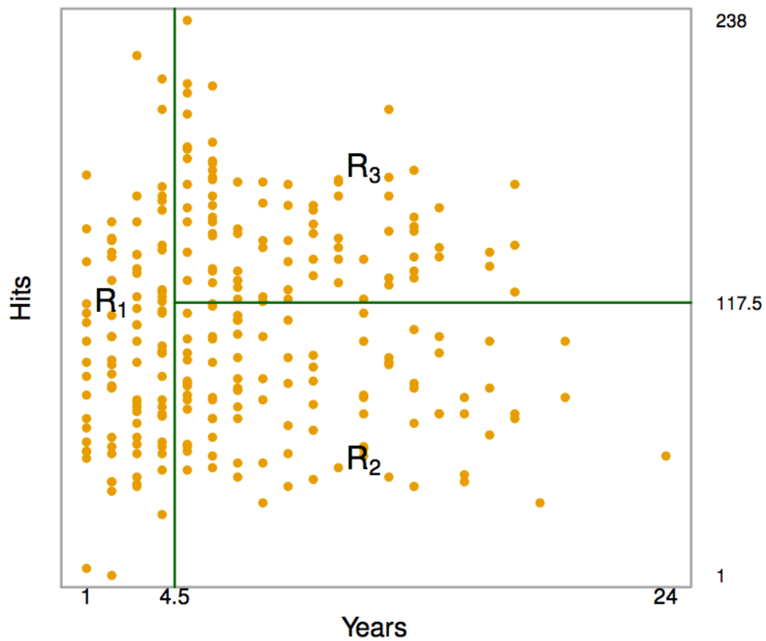
Decision Trees – Regression

Baseball salaries:

(Blue, Green) for low salaries
(Yellow, Red) for high salaries



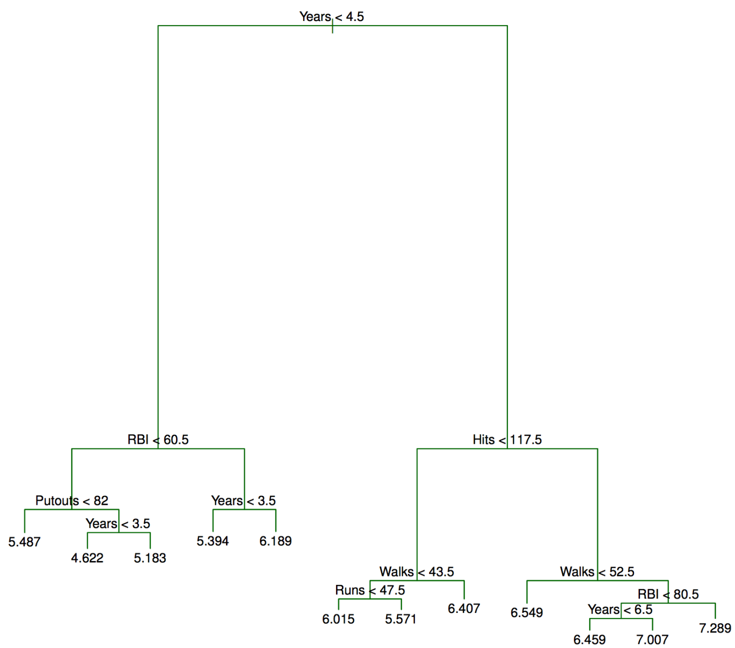
Decision Trees – Regression



At each split, we aim to minimize:

$$\sum_{m=1}^{|T|} \sum_{i:\, x_i \in R_m} (y_i - \hat{y}_{R_m})^2$$

Decision Trees – Regression



This should feel familiar!

In Lasso/Ridge, attack high variance of *linear regression* with cost penalty λ

Here attack high variance of *decision tree* with cost penalty α

When to stop?

- You don't! Best practice to grow a very large "bushy" tree and prune backwards.

How?

- Let |T| = # of terminal nodes
- Then for any penalty term α, we have a subtree T which minimizes

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

- Cross-validate as usual to choose α and it's corresponding tree.

Decision Trees – Classification

Making Predictions

At each terminal node (or rectangular region), predict

- Regression: Average
- Classification: Most commonly occurring class

How to split?

At each potential splitting node, minimize (in terms of information gain)

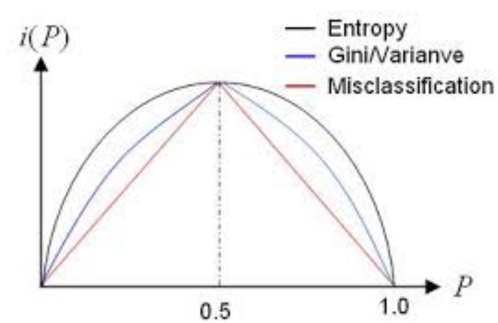
- Regression: RSS
- Classification:

Classification Error Rate $E = 1 - \max_k(\hat{p}_{mk})$

Gini index $G = \sum_{k=1}^{K} \hat{p}_{mk}(1 - \hat{p}_{mk})$

Cross-entropy $D = -\sum_{k=1}^{K} \hat{p}_{mk} \log \hat{p}_{mk}$

$\hat{p}_{mk}$ is proportion in m-th region in k-th class



Bagging

- Previously we looked at post-pruning our single decision tree to attack the variance

- Instead, we can just grow many large "bushy" trees and average away the variance (*central limit theorem*) by growing lots of trees (*bootstrapping*)!

Bagging

Making Predictions

Training set → ***B*** bootstrapped training datasets

- Regression: Average prediction of B trees

$$\hat{f}_{\text{bag}}(x) = \frac{1}{B}\sum_{b=1}^{B}\hat{f}^{*b}(x)$$

- Classification: Majority vote among B trees

Error Estimation

- Since bootstrapped, each tree only uses about 2/3 of observations → remaining 1/3 can be used to estimate OOB (out-of-bag) error. Like Test-error!

Random Forests

Same idea except at each split considered choose a random selection of m predictors

Typically $m \approx \sqrt{p}$ so that if you have 100 predictors, you randomly 10 candidate features at **each** split point.

This "decorrelation" of the trees leads to improved performance over bagging.

Afternoon

Random Forest, deeper dive

- Variable Importance
- Bias-Variance Tradeoff
  - Why is it so good to bag trees?
- Tuning
  - m; minimum node size
- Trees revisited
  - Categorical predictors
  - Loss Matrix
  - Surrogate Predictors

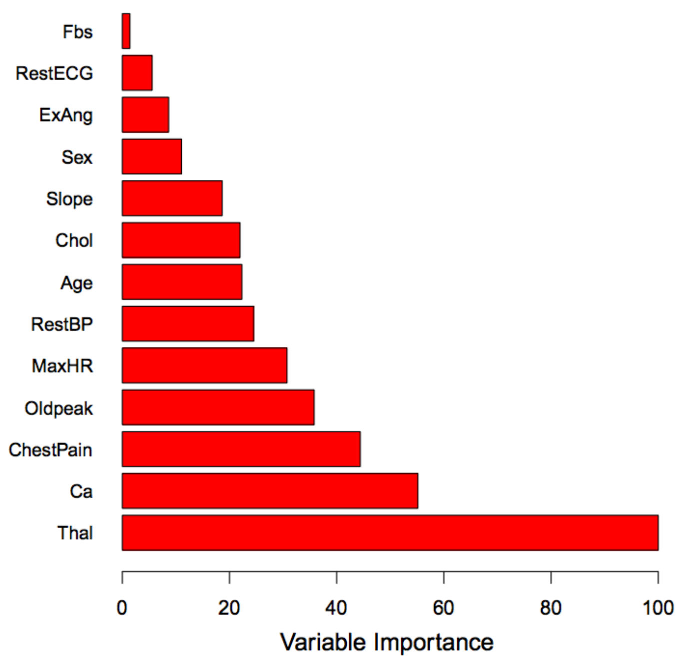
Bagging/RF – Feature Importance

Bagged/Random Forest Regression trees:

- Record total amount RSS decreases due to splits over the predict, averaged over all *B trees* ➔ *Larger value indicates "importance"*

Bagged/Random Forest Classification trees:

- Record total amount Gini index decreases due to splits over given predictor predict, averaged over all *B trees* ➔ *Larger value indicates "importance"*



Variable importance of `Heart` data

Feature Importance

Alternative way to calculate variable importance

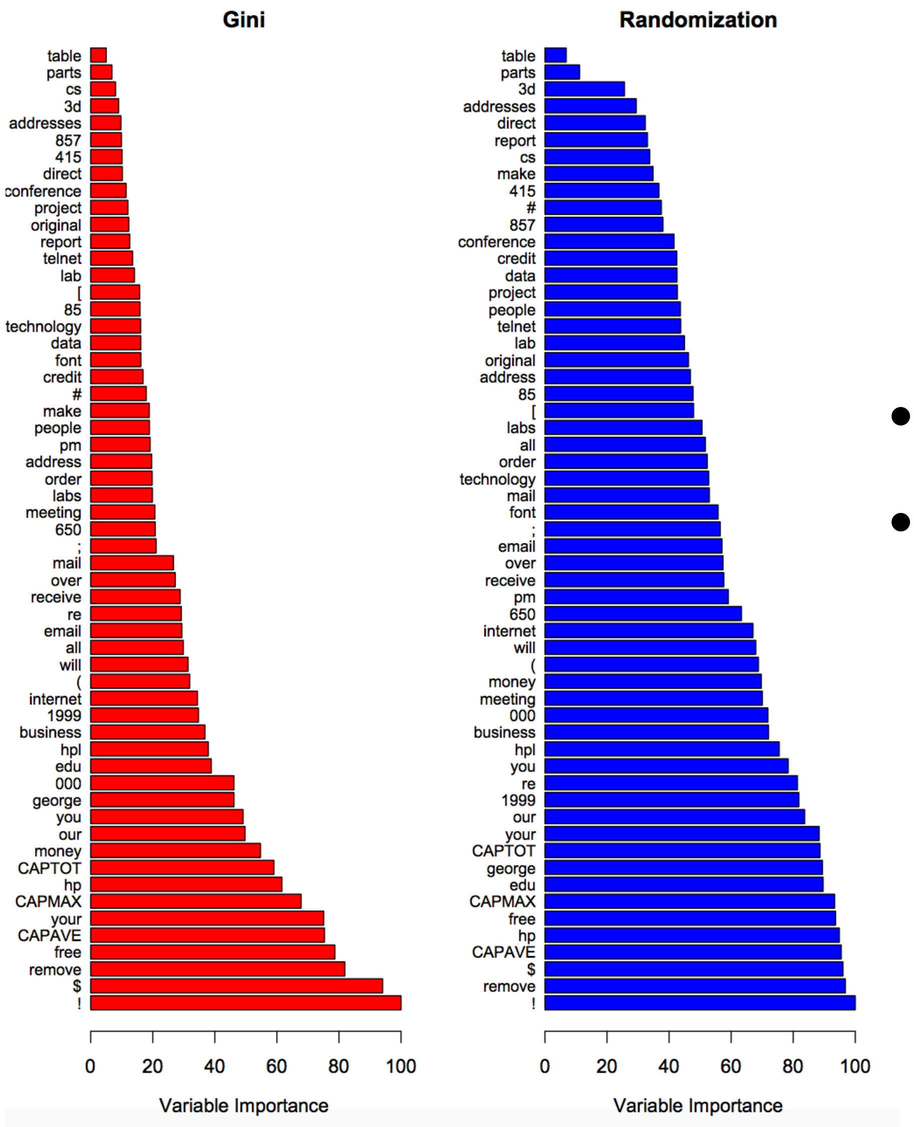
To evaluate importance of *j*th variable…

(1) When *b*th tree is grown, OOB samples passed down through tree → record accuracy

(2) Values of *j*th variable randomly permuted in OOB samples → compute new (lower) accuracy

→ Average decrease in accuracy over all trees

Comparison of Feature Importances



- Note similarity in rankings
- More even distribution for Randomization (2nd way)

Feature importance in sklearn

http://scikit-learn.org/stable/modules/ensemble.html#feature-importance-evaluation

- Basically, the higher in the tree the feature is, the more important it is in determining the result of a data point.
- The expected fraction of data points that reach a node is used as an estimate of that feature's importance for that tree.
- Finally, average those values across all trees to get the feature's importance.

Bias-Variance "Tradeoff"

Bias

- Deep tree → Relatively low bias
- Expectation of average of B trees same as expectation of any one of the trees

Variance

- Where we really win!
- Average of *B* i.d. (identically distributed) random variables, with pairwise correlation ρ, has variance…

$$\boxed{\rho\sigma^2 + \frac{1-\rho}{B}\sigma^2}$$

What happens as *B* increases?
What does *ρ* depend on?

Tuning

Classification

$$m = \sqrt{p}$$

minimum node size = 1

Regression

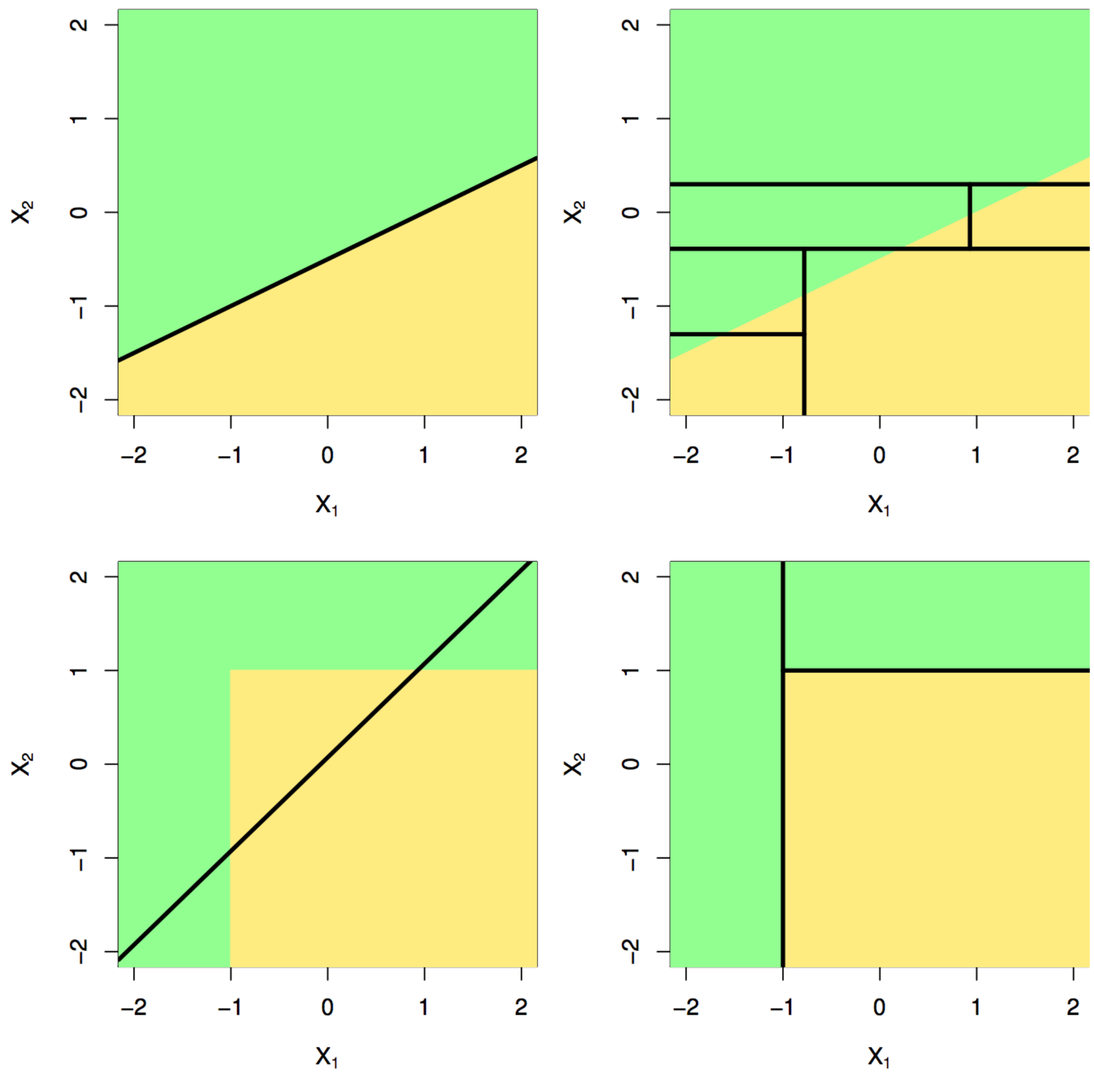
$$m = p/3$$

minimum node size = 5

“max_features”
“min_samples_leaf"

Suggested defaults by the inventors.
But in practice you can tune these just like you did for λ in Lasso/Ridge!

Trees Revisited



Trees Revisited

Extra credit reading: Elements of Statistical Learning, 9.2.4

Categorical Predictors

- If q possible unordered values, $2^{q-1} - 1$ possible partitions of q values into 2 groups!
- Can order predictor classes according to proportion in class 1
- Split predictor as if splitting ordered predictor

Loss Matrix

- Can weight the classes using a user-specified loss matrix

Missing Predictor Variables

- Can use surrogate predictors and split points
- Exploit correlations between predictors to alleviate effects of missing data

Questions

- Describe the random forest algorithm, step by step
  - How to build single tree?
  - How many to build?
  - How does final classification/regression estimate happen?
- What happens as number of trees, *B* increases, for bagging or random forest?
- Why does Random Forest outperform bagging?
- How does the Random Forest "win" at the Bias-Variance

Questions

- Describe the random forest algorithm, step by step
  - How to build single tree? Deep tree generally, since will average away variance
  - How many to build? Many as computationally reasonable
  - How does final classification/regression estimate happen? Majority/Average
- What happens as number of trees, *B* increases, for bagging or random forest? Variance decreases. Greater the B, the better, although there are diminishing returns after a certain point. Also incurring some perhaps undesirable computational cost
- Why does Random Forest outperform bagging?

  "Decorrelate" the trees through random selection of *m* at each node.

  $$\rho\sigma^2 + \frac{1-\rho}{B}\sigma^2$$
- How does the Random Forest "win" at the Bias-Variance

  Mostly wins through "averaging away" the variance. Again $\rho\sigma^2 + \frac{1-\rho}{B}\sigma^2$