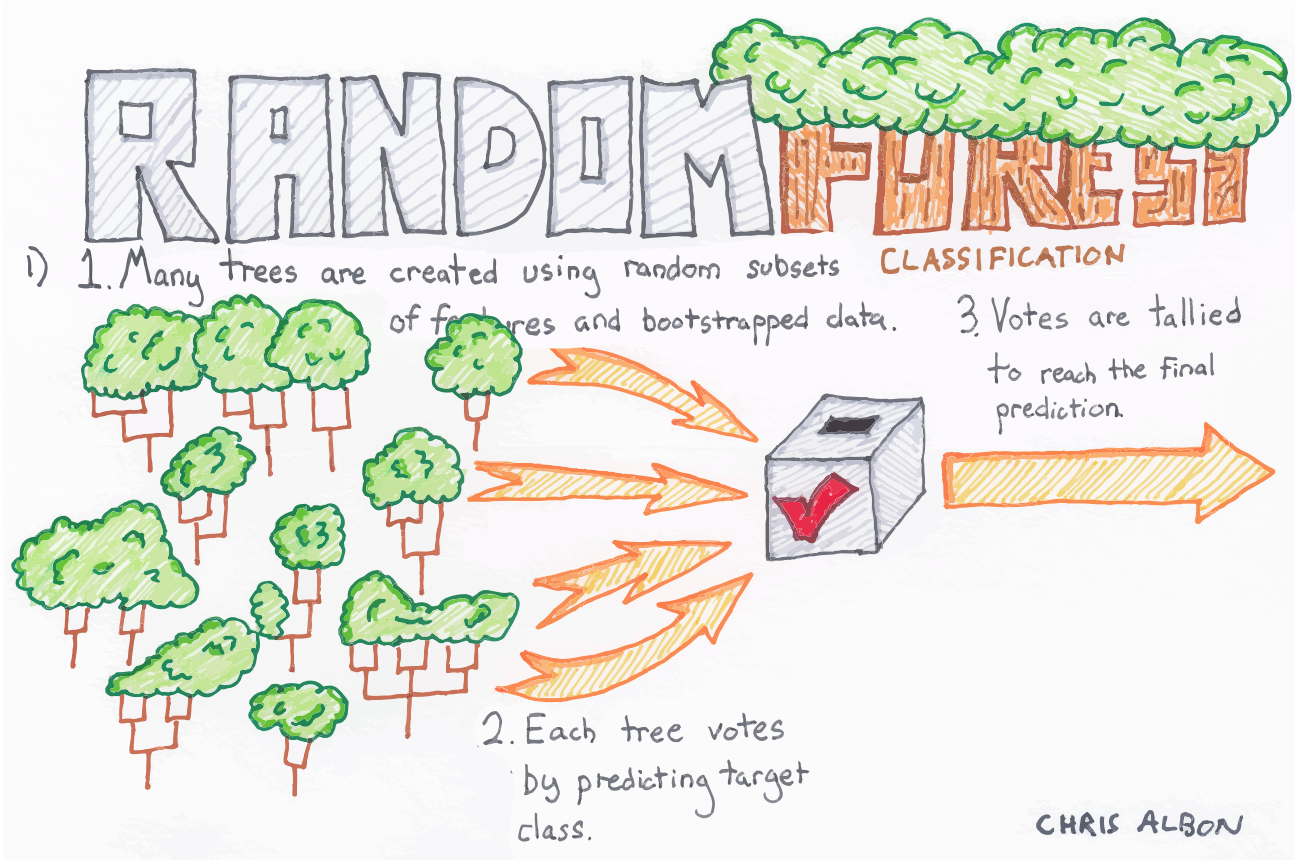
CMSC 471

ML: Bagging

What's better than a tree?

Random Forest

- Can often improve performance of decision tree classifiers using a set of decision trees (a forest)
- Each tree trained on a random subset of training data
- Classify a data instance using all trees
- Combine answers to make classification
  - E.g., vote for most common class


RANDOM FORESTS
CLASSIFICATION
1) 1. Many trees are created using random subsets of features and bootstrapped data.
2. Each tree votes by predicting target class.
3. Votes are tallied to reach the final prediction.
CHRIS ALBON

cf. Wisdom of the Crowd

- Statistician Francis Galton observed a 1906 contest to guess an ox's weight at a country fair. 800 people entered. He noted that their average guess (1,197lb) was very close to the actual weight (1,198lb)

- When getting human annotations training data for machine learning, standard practice is get ≥ 3 annotations and take majority vote

cf. *abbreviation (short for Latin: confer/conferatur) refer reader to other material to make a comparison*

Random Forests Benefits

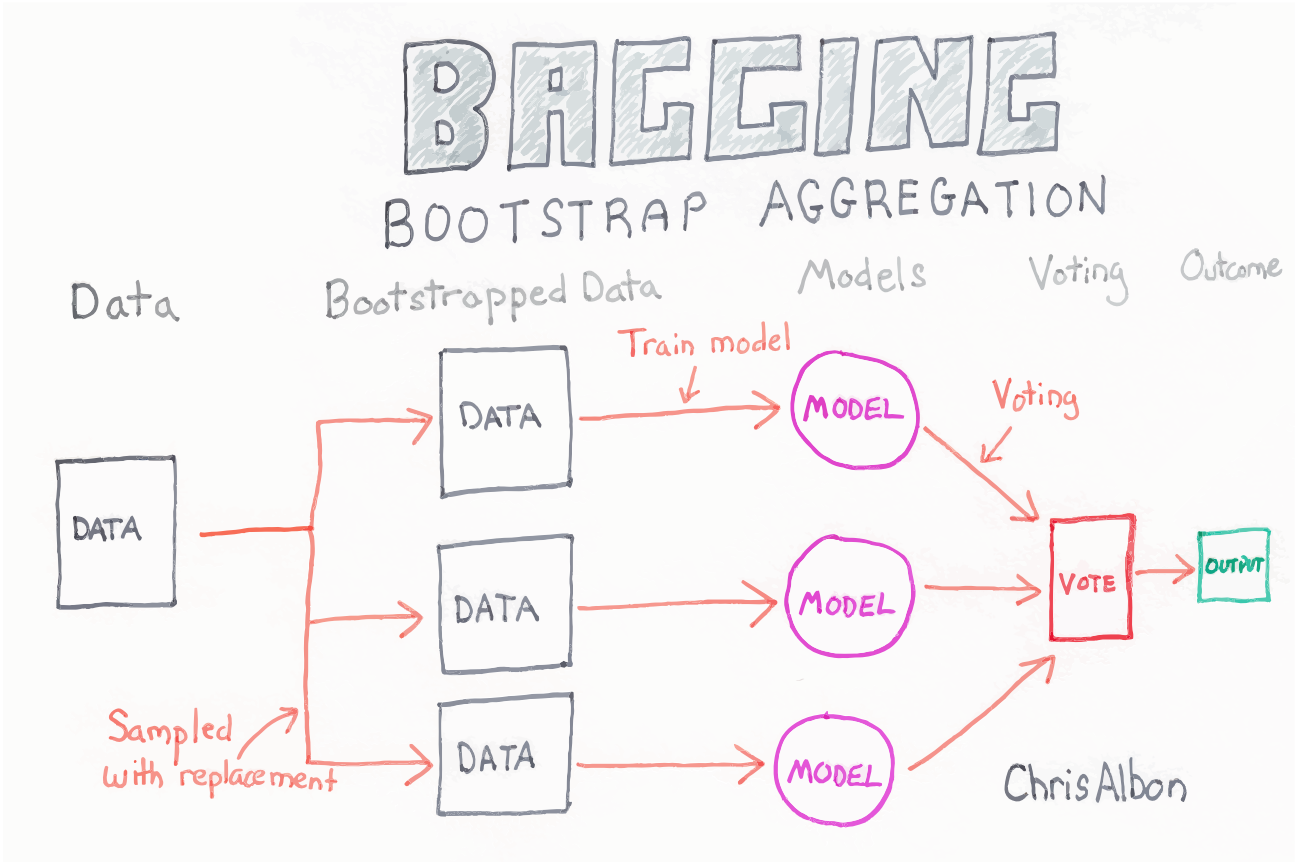
- Decision trees not the strongest modeling approach
- Random forests make them much stronger
- => more **robust** than a single decision tree
  - Limit overfitting to given dataset
  - Reduce errors due to training data bias
  - Stable performance if some noise added to training data

Bagging

- Idea can be used on any classifier!
- Improve classification by combining classifications of randomly selected training subsets
- Bagging = ***B**ootstrap **agg**regat**ing***

  *An **ensemble** meta-algorithm that can improve stability & accuracy of algorithms for statistical classification and regression*

- Helps avoid overfitting
- AKA ensembling


BAGGING
BOOTSTRAP AGGREGATION
Data
Bootstrapped Data
Models
Voting
Outcome
DATA
DATA
DATA
DATA
Train model
MODEL
MODEL
MODEL
Voting
VOTE
OUTPUT
Sampled with replacement
Chris Albon

Choosing subsets of training data

- Classic bagging: select random subset of training instances **with replacement**
- Pasting: select random subset of training instances
- Random Subspaces: use all training instances, but with a random subset of features
- Random Patches: random subset of instances and random subset of features
- What's best? YMMV: depends on problem, training data, algorithm