

CS109 – Data Science

Verena Kaynig-Fittkau

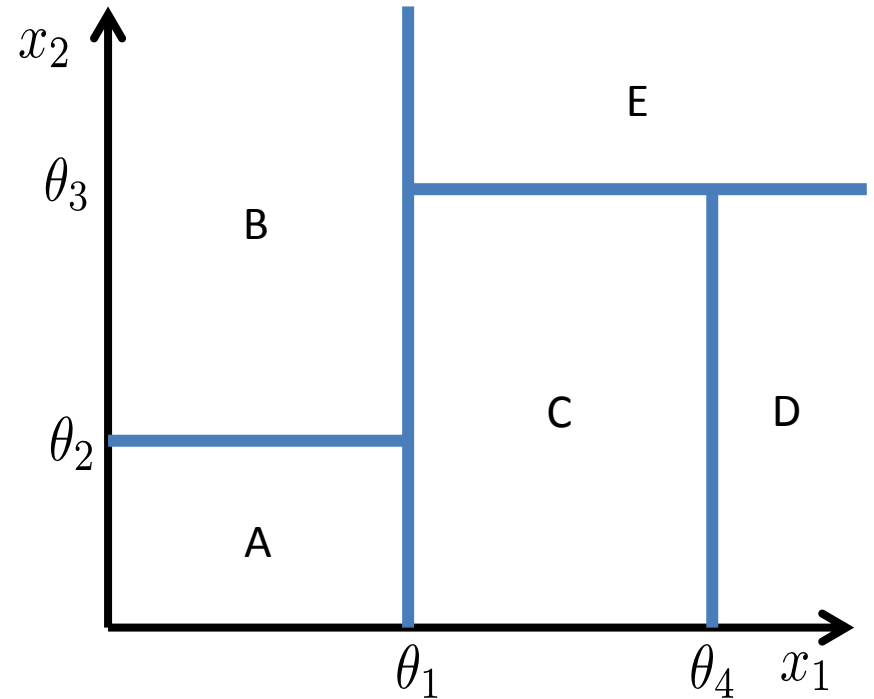
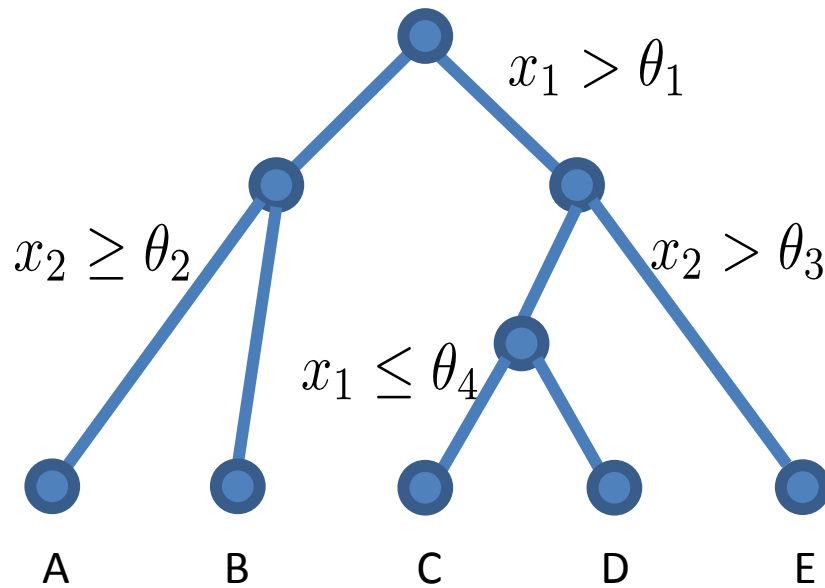
vkaynig@seas.harvard.edu

staff@cs109.org

Announcements

- Don't forget to register your final project group!
- HW4 is out
- Problem 2a post on piazza – have a look

Decision Tree - Idea



DecisionTree in sklearn

- <http://scikit-learn.org/stable/modules/generated/sklearn.tree.DecisionTreeClassifier.html>

Decision Trees vs SVM

Characteristic	SVM	Trees
Natural handling of data of “mixed” type	▼	▲
Handling of missing values	▼	▲
Robustness to outliers in input space	▼	▲
Insensitive to monotone transformations of inputs	▼	▲
Computational scalability (large N)	▼	▲
Ability to deal with irrelevant inputs	▼	▲
Ability to extract linear combinations of features	▲	▼
Interpretability	▼	◆
Predictive power	▲	▼

Abalone data

Wisdom of Crowds

The collective knowledge of a **diverse and independent** body of people typically exceeds the knowledge of any single individual, and can be harnessed by voting.

James Surowiecki

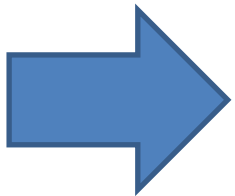


Netflix Prize

- Take home messages:
 - SVD rocks!
 - Initially great progress, then significantly slowed down
 - Ensembles were the method of choice
 - This was a surprise!
 - And a bit of a disappointment

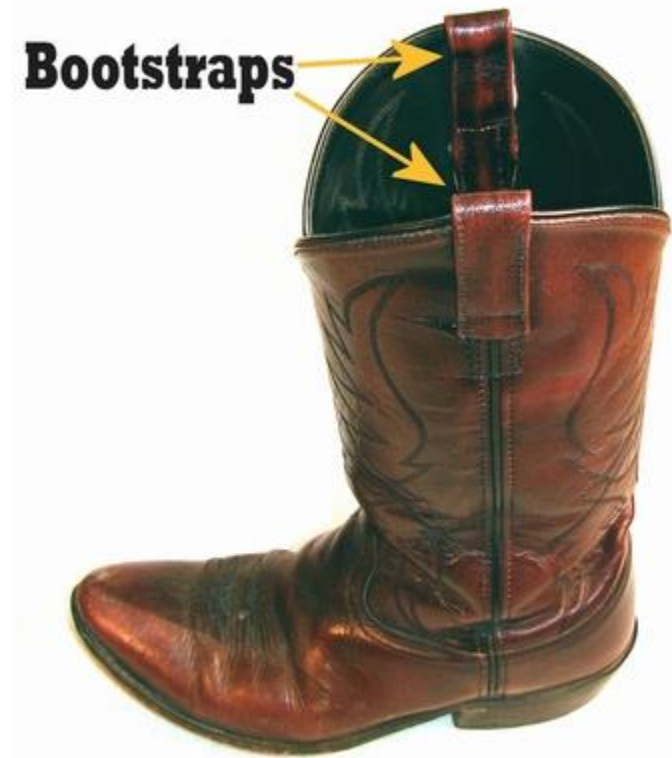
Ensemble Methods

- A single decision tree does not perform well
- But, it is super fast
- What if we learn multiple trees?



For multiple trees we need even more data!

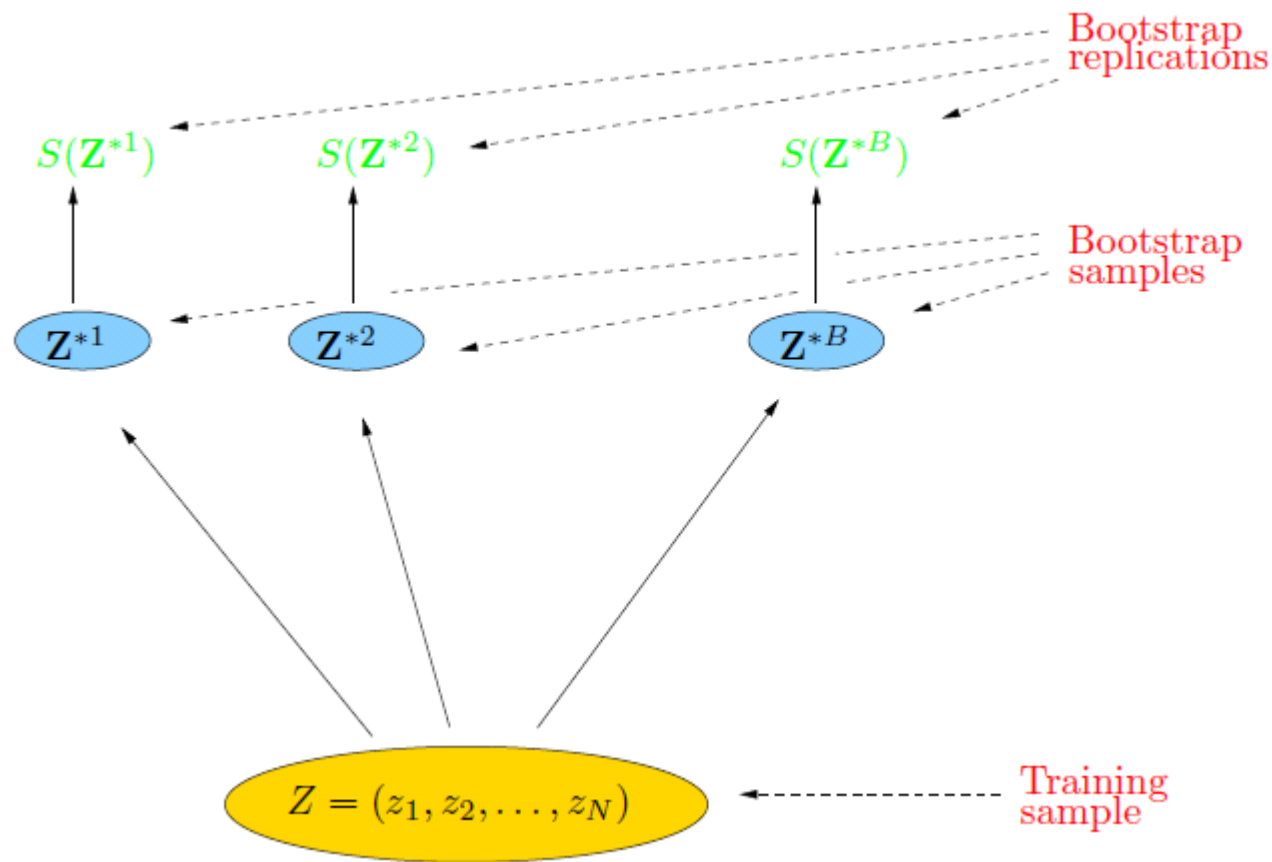
Bootstrap



Bootstrap

- Resampling method from statistics
- Useful to get error bars on estimates
- Take N data points
- Draw N times with replacement
- Get estimate from each bootstrapped sample

Bootstrap

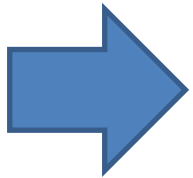


This is neat!

- I can generate more data!
- Can I do cross validation on this?

Bootstrap vs Cross-validation

- Bootstrap has overlap in data sets



Do not use simple bootstrap to generate train and test data from same data set.

$$p(n \in Z^{*i}) = 1 - \left(1 - \frac{1}{N}\right)^N$$

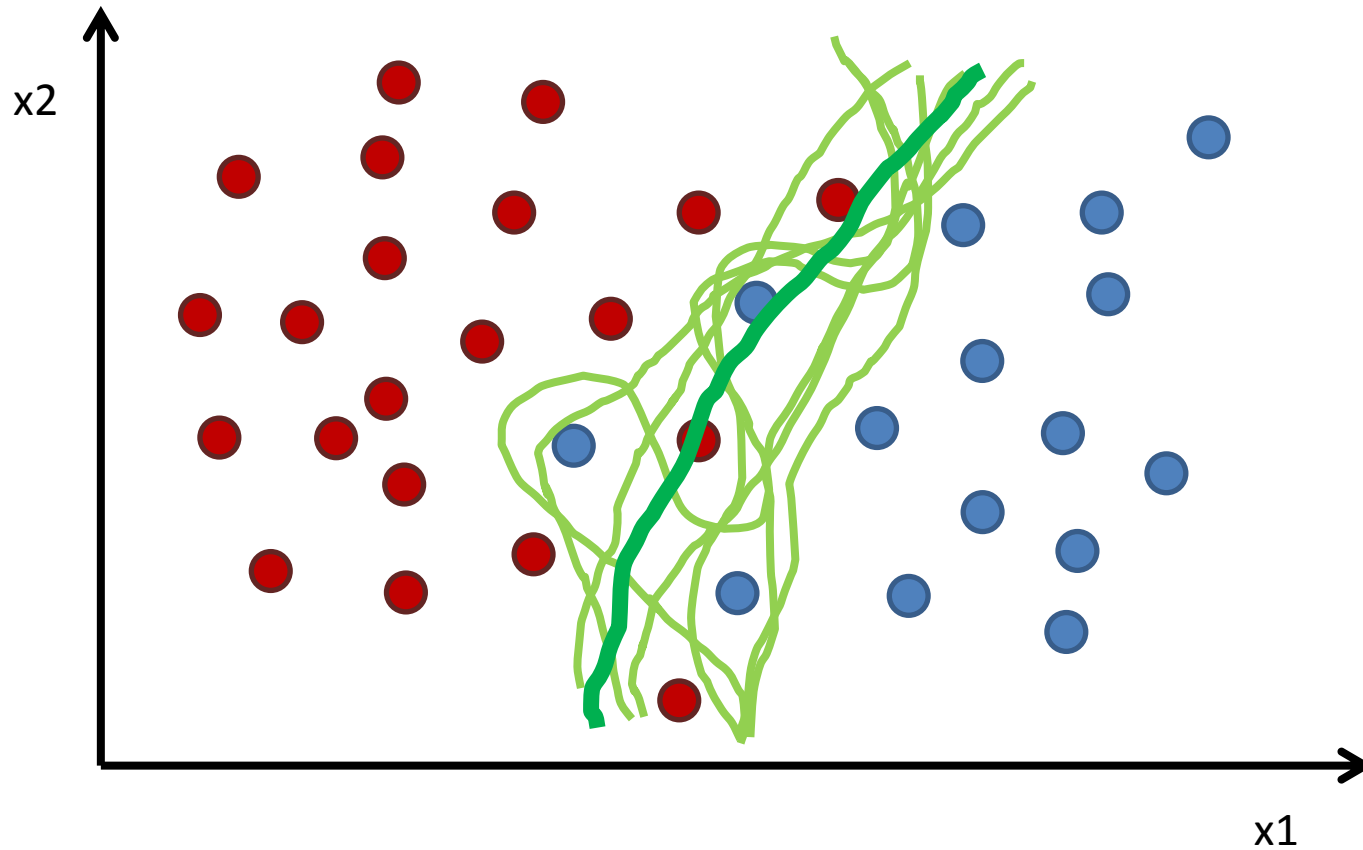
$$\approx 0.632$$

This number is important later

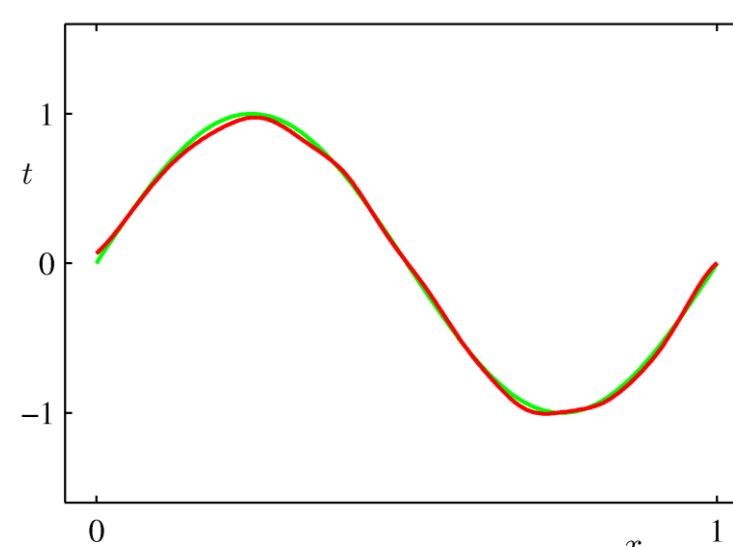
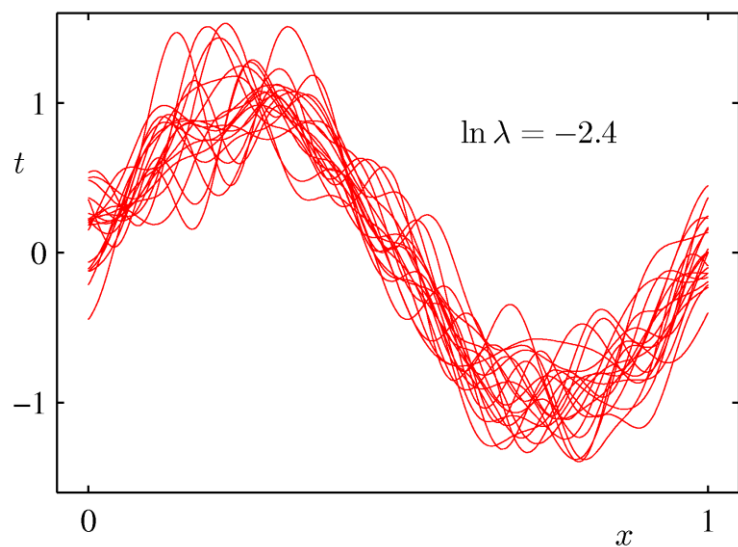
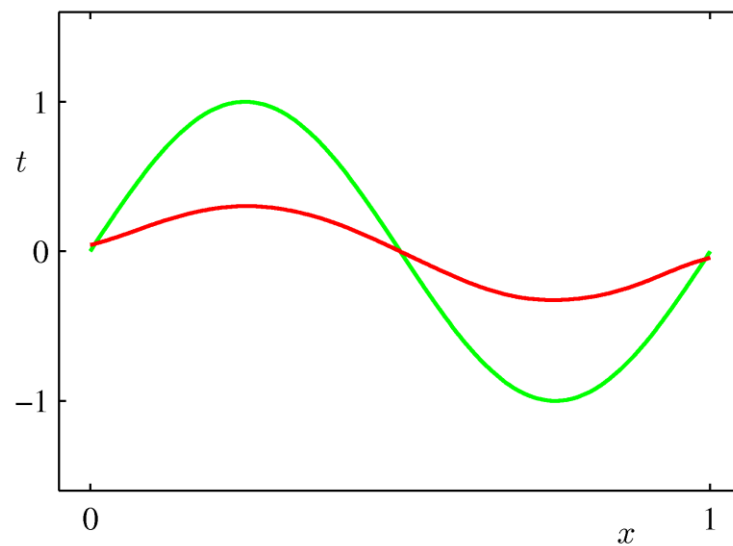
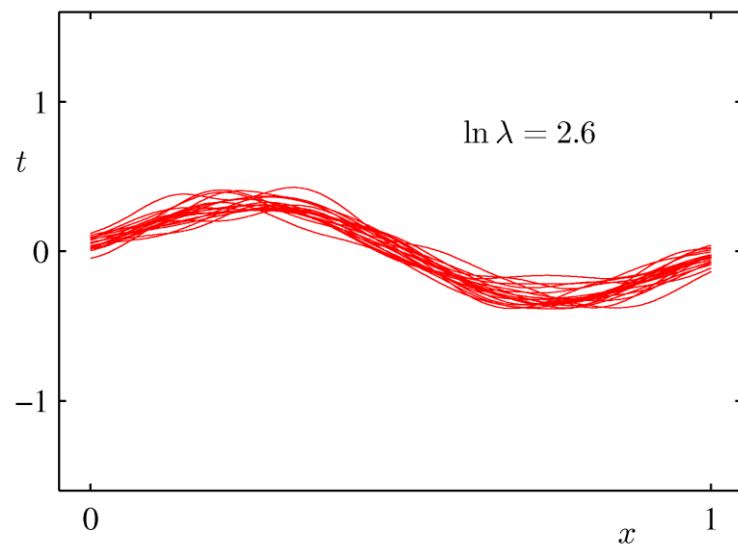
Bagging

- Bootstrap aggregating
- Sample with replacement from your data set
- Learn a classifier for each bootstrap sample
- Average the results

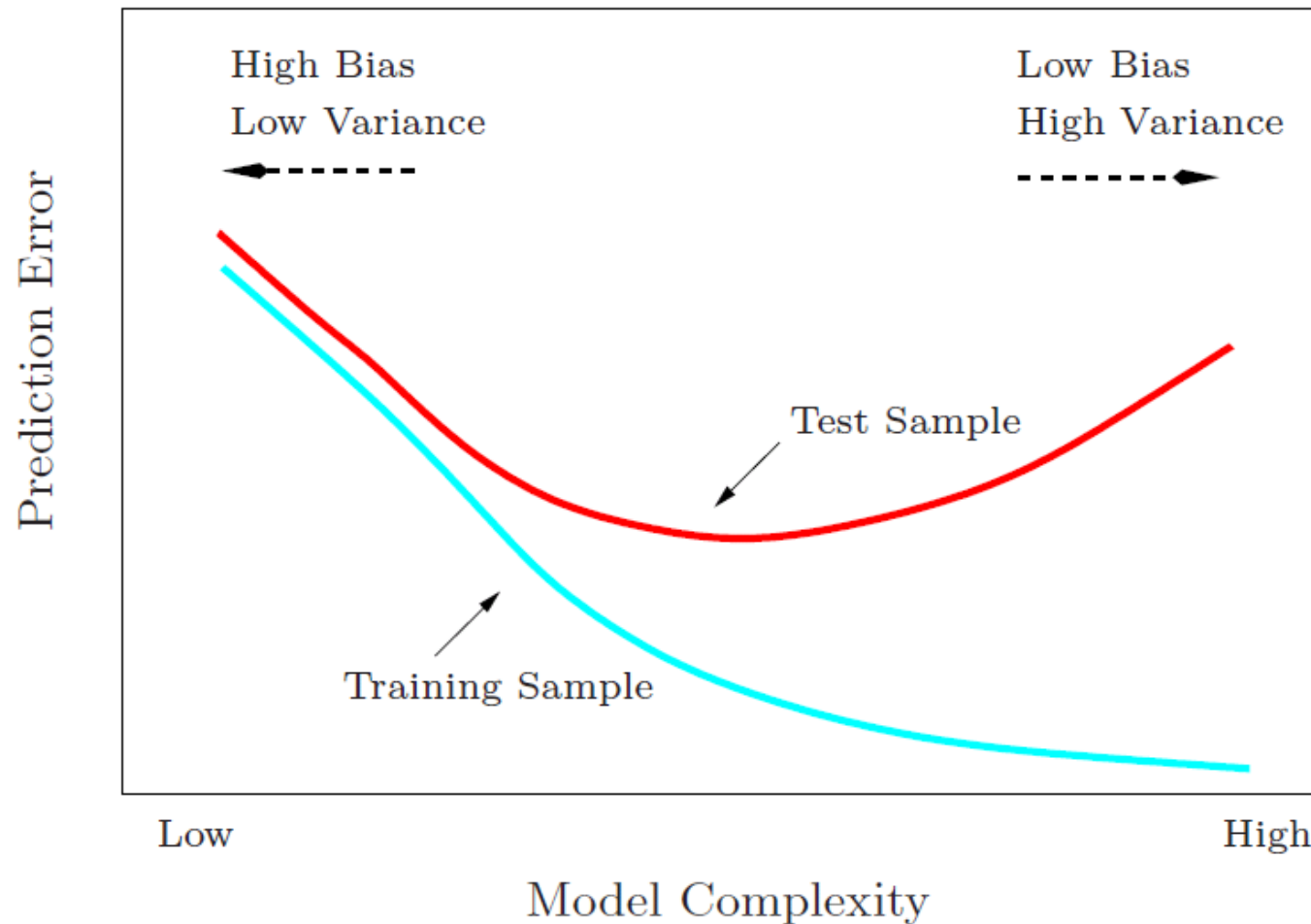
Bagging Example



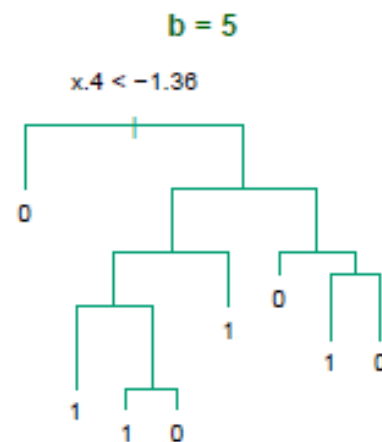
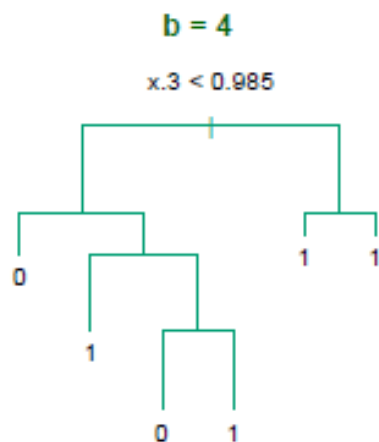
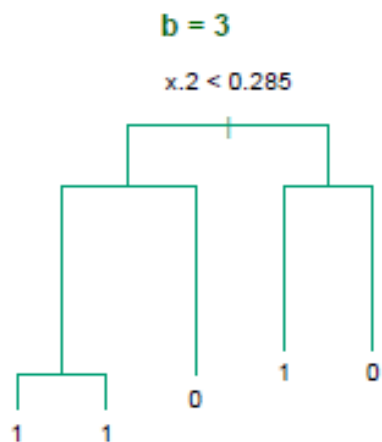
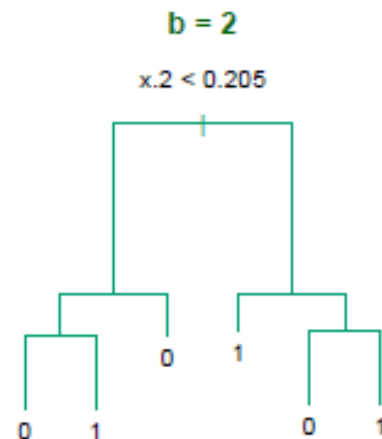
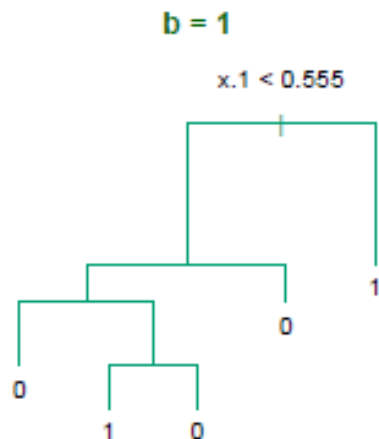
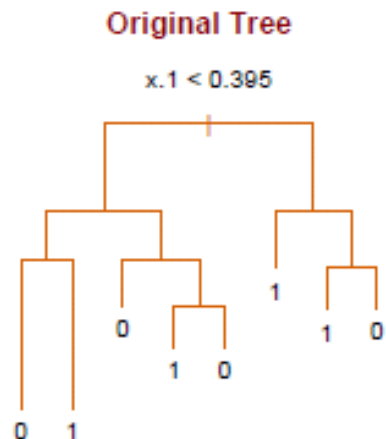
Bias-Variance Trade-off



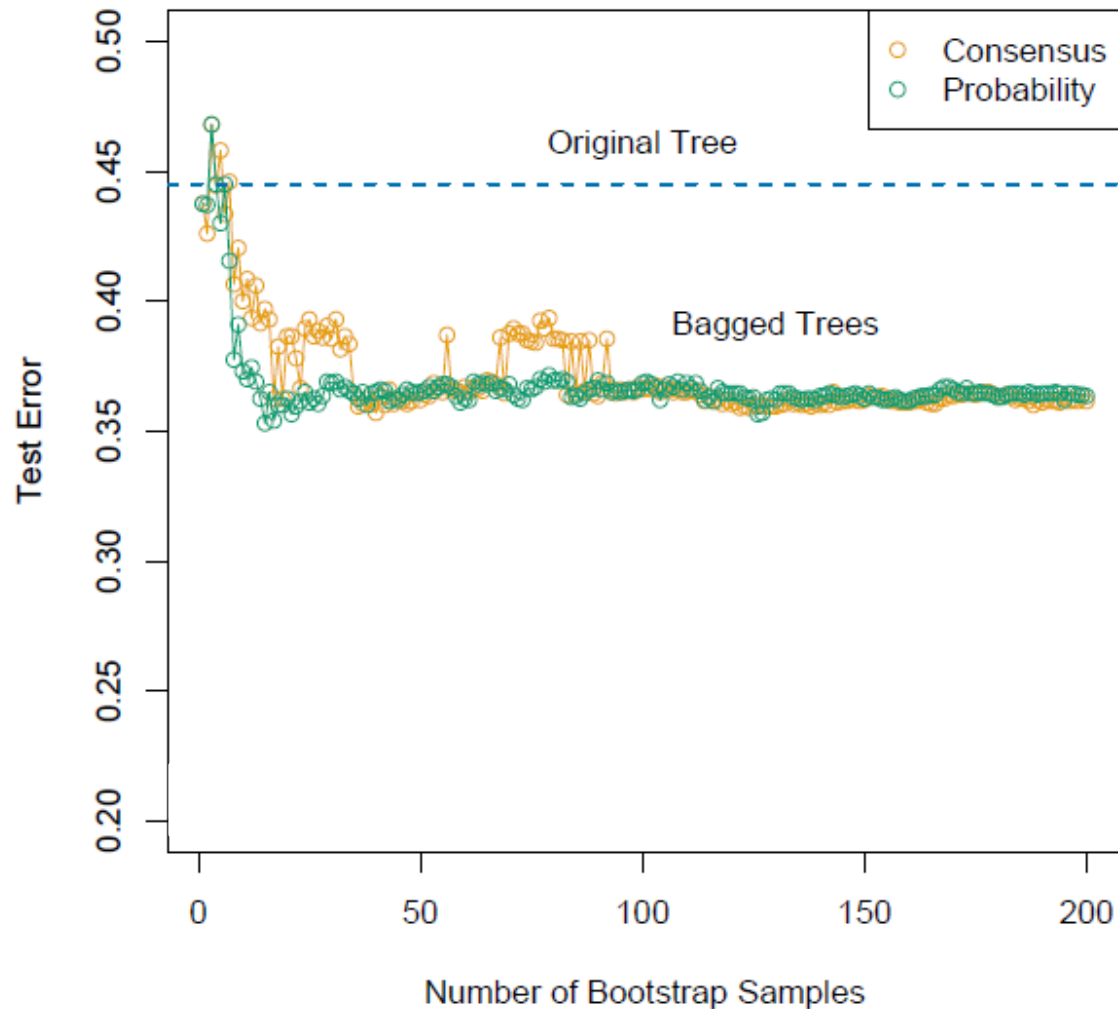
We Have Seen This Before



Bagging Decision Trees



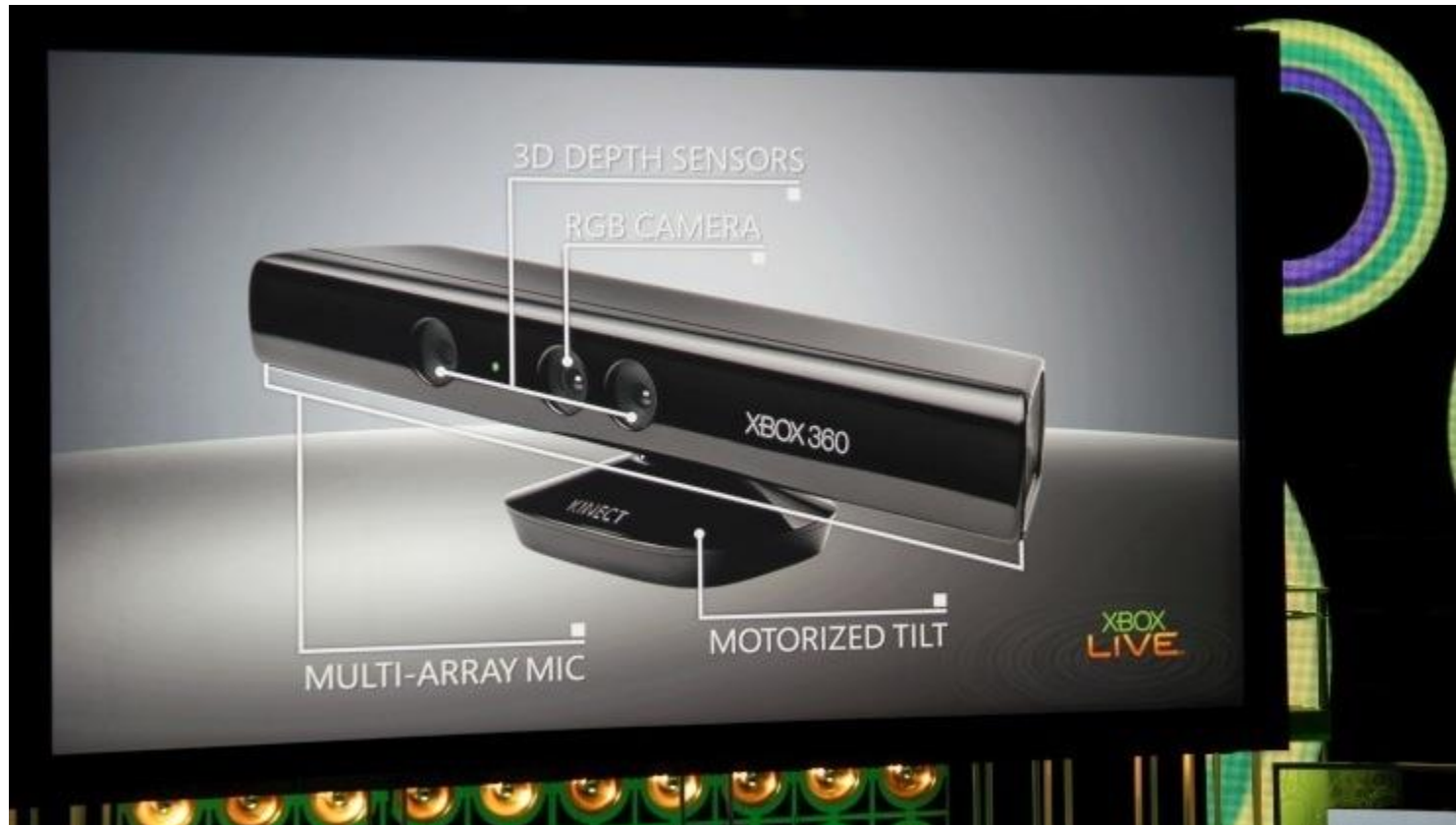
Bagging Decision Trees



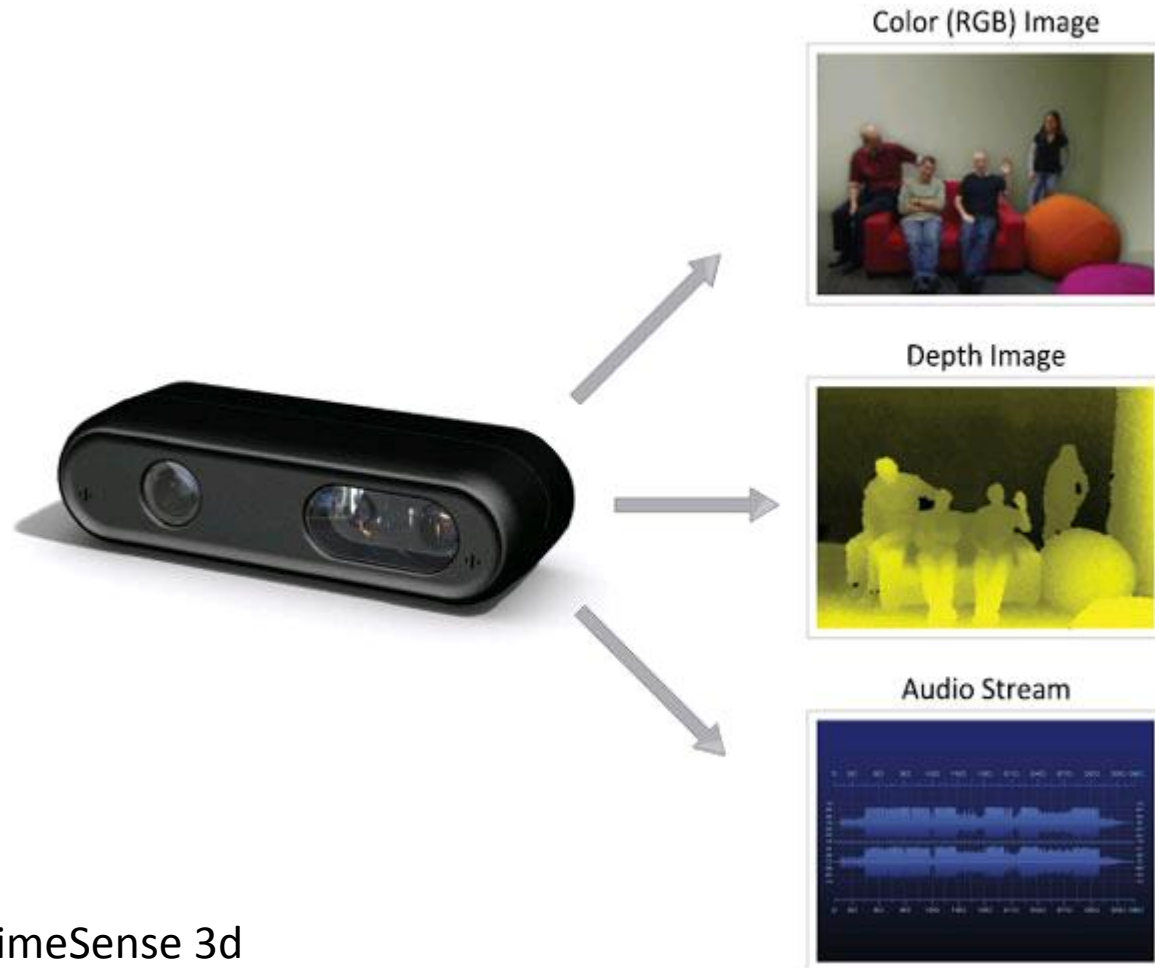
Bagging

- Reduces overfitting (variance)
- Normally uses one type of classifier
- Decision trees are popular
- Not helping with linear models
- Easy to parallelize

Kinect

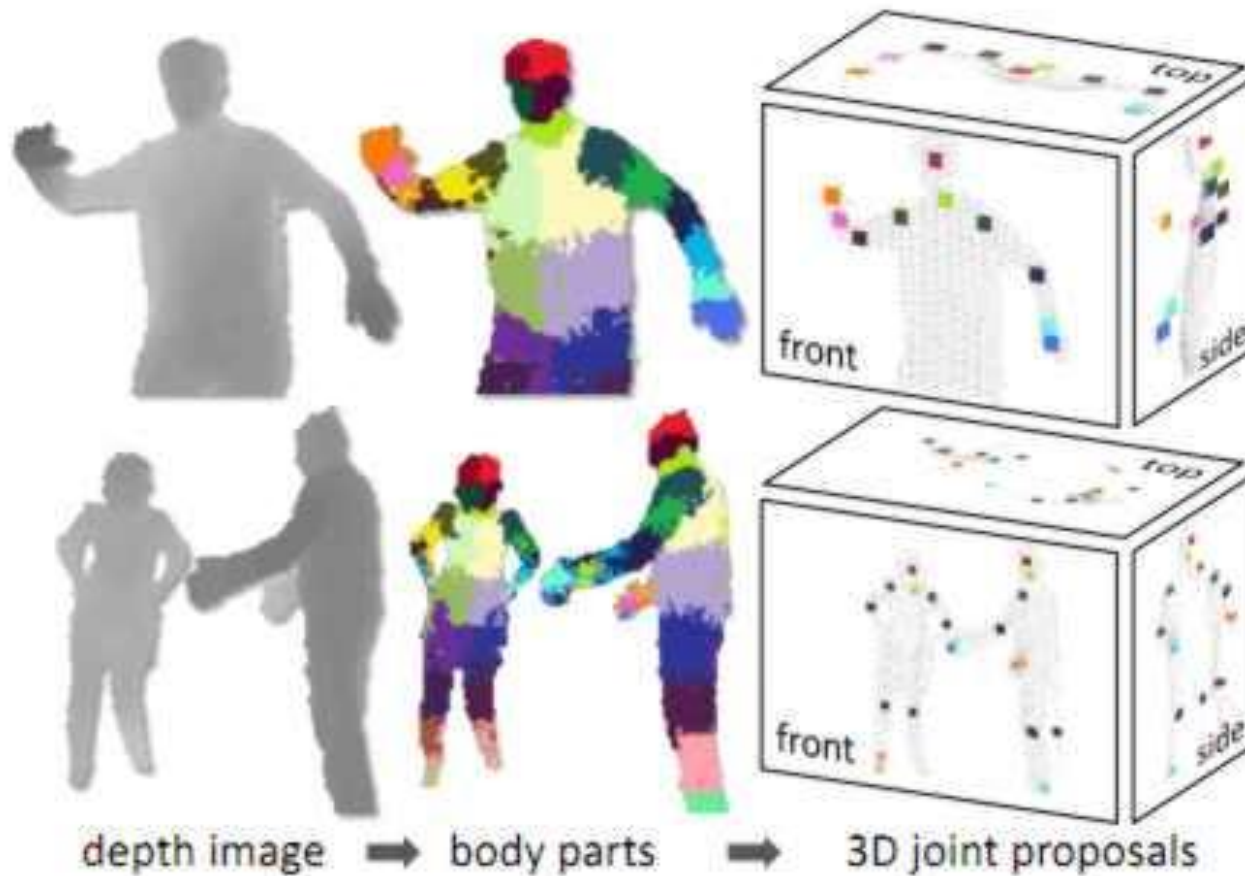


Kinect Sensor



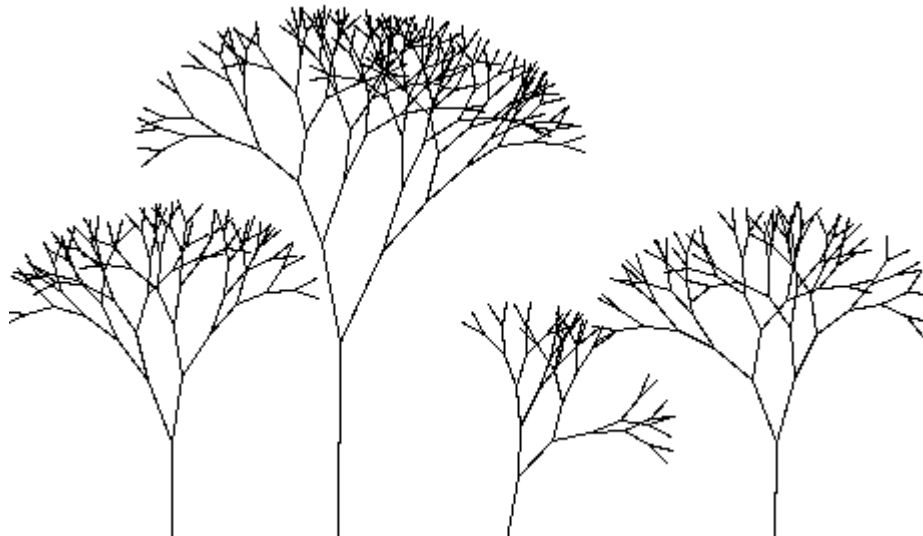
PrimeSense 3d

Random Forest – Body Part Recognition



Random Forest

- Builds upon the idea of bagging
- Each tree build from bootstrap sample
- Node splits calculated from **random feature subsets**



Random Forest

- All trees are fully grown
- No pruning
- Two parameters
 - number of features
 - number of trees

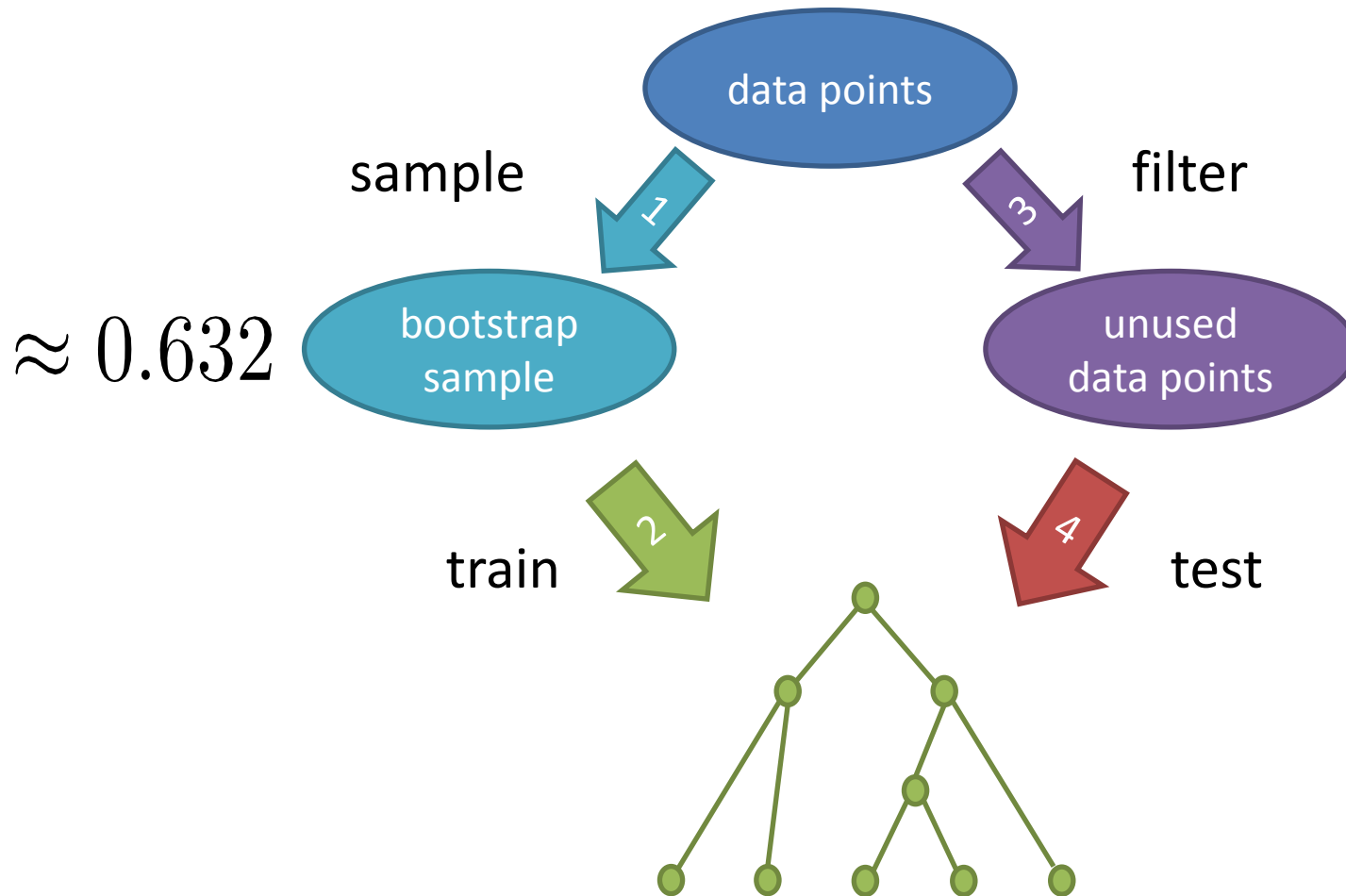
Random Forest Error Rate

- Error depends on:
 - Correlation between trees (higher is worse)
 - Strength of single trees (higher is better)
 - Why?
- Increasing number of features for each split:
 - Increases correlation
 - Increases strength of single trees

Out of Bag Error

- Each tree is trained on a bootstrapped sample
- About $1/3$ of data points not used for training
- Predict unseen points with each tree
- Measure error

Out of Bag Error



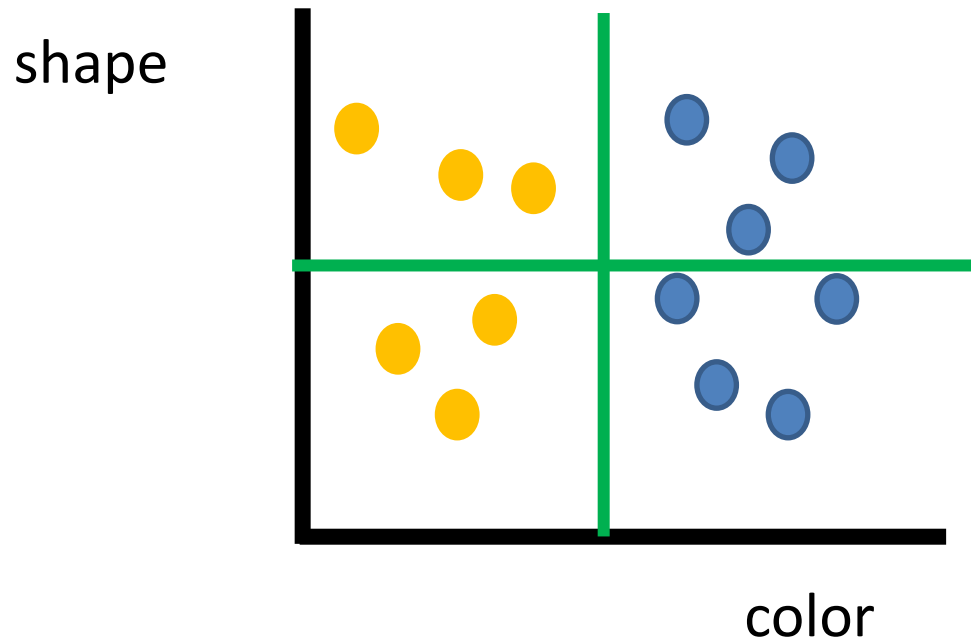
Out of Bag Error

- Very similar to cross-validation
- Measured during training
- Can be too optimistic

Variable Importance - 1

- Again use out of bag samples
- Predict class for these samples
- Randomly permute values of one feature
- Predict classes again
- Measure decrease in accuracy

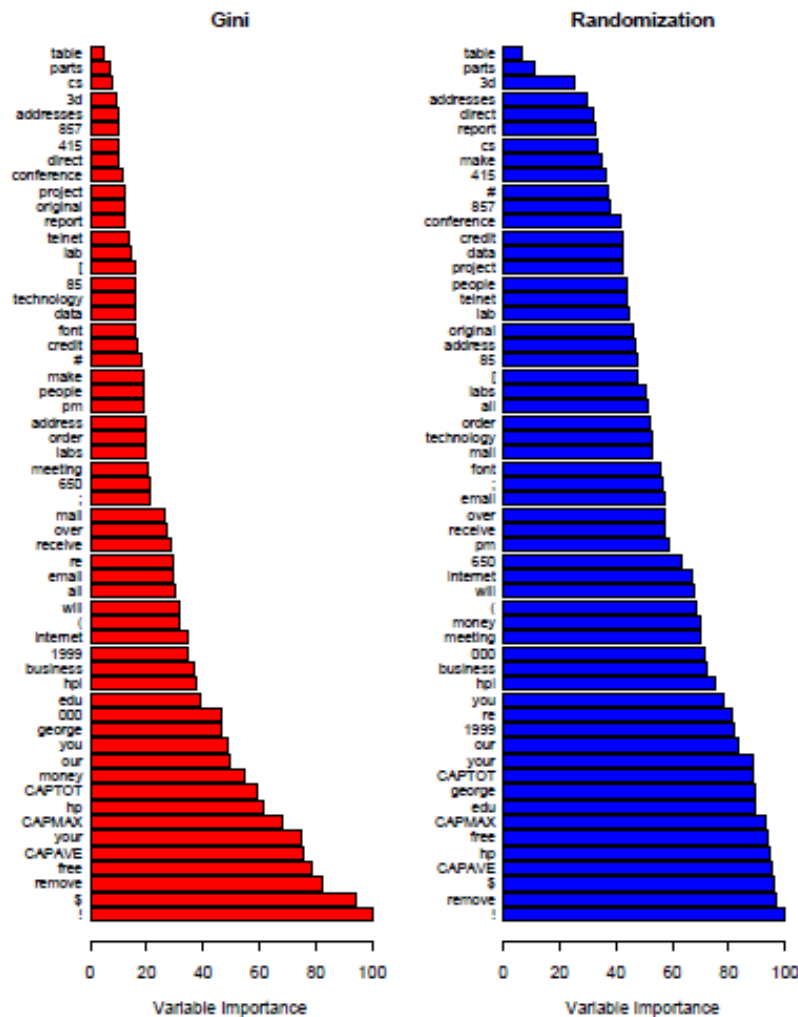
Variable Importance - 1



Variable Importance - 2

- Measure split criterion improvement
- Record improvements for each feature
- Accumulate over whole ensemble

Example: Spam classification



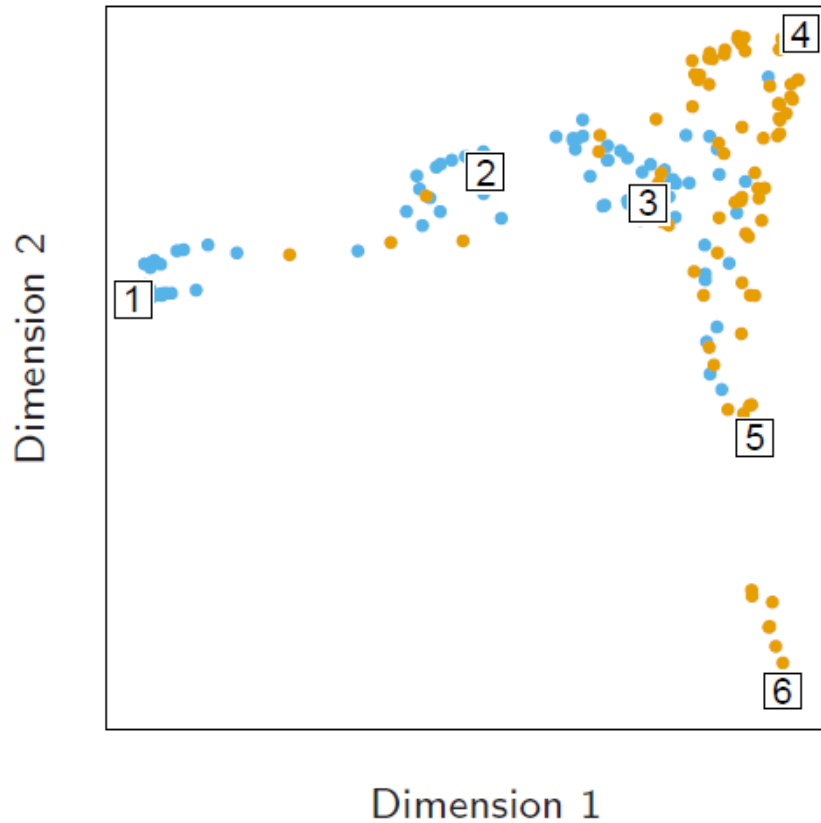
Randomization tends to spread out the variable importance more uniformly.

Proximity

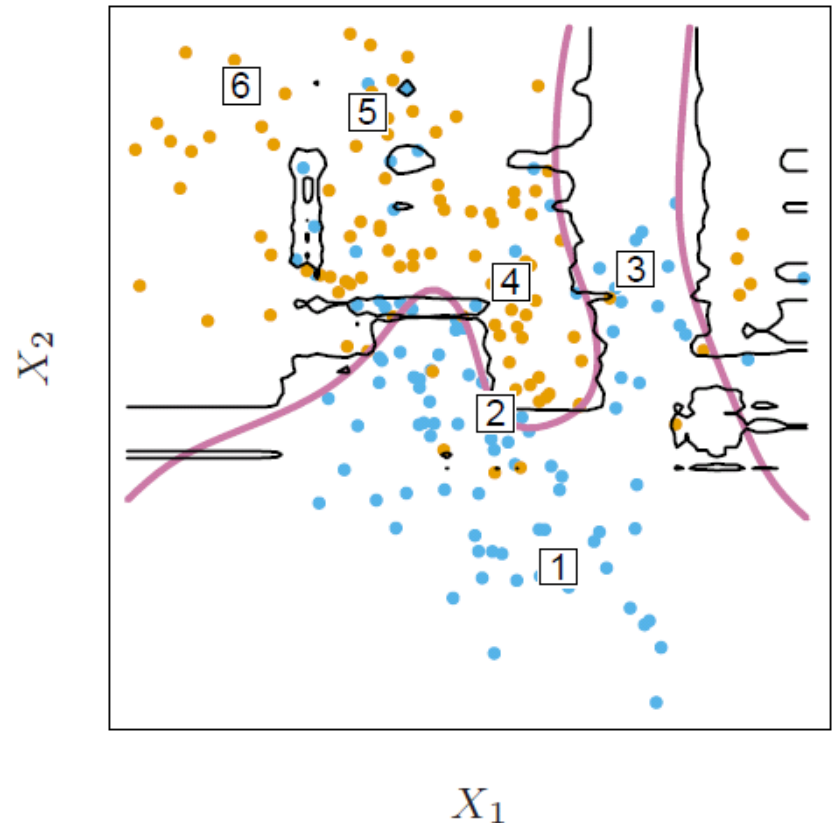
- Pairwise distance: $N \times N$ matrix
- Classify complete data set for each tree
- Same leaf \Rightarrow increase proximity
- Normalize by the number of trees

Proximity Visualization

Proximity Plot

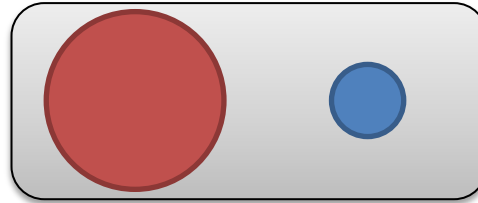


Random Forest Classifier

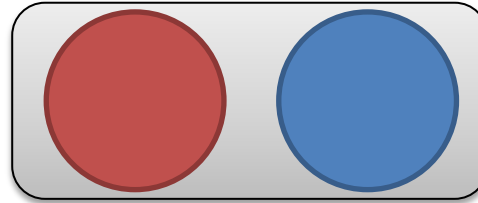


Unbalanced Classes

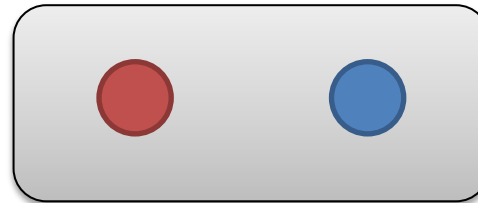
- The Problem:



- Oversample:

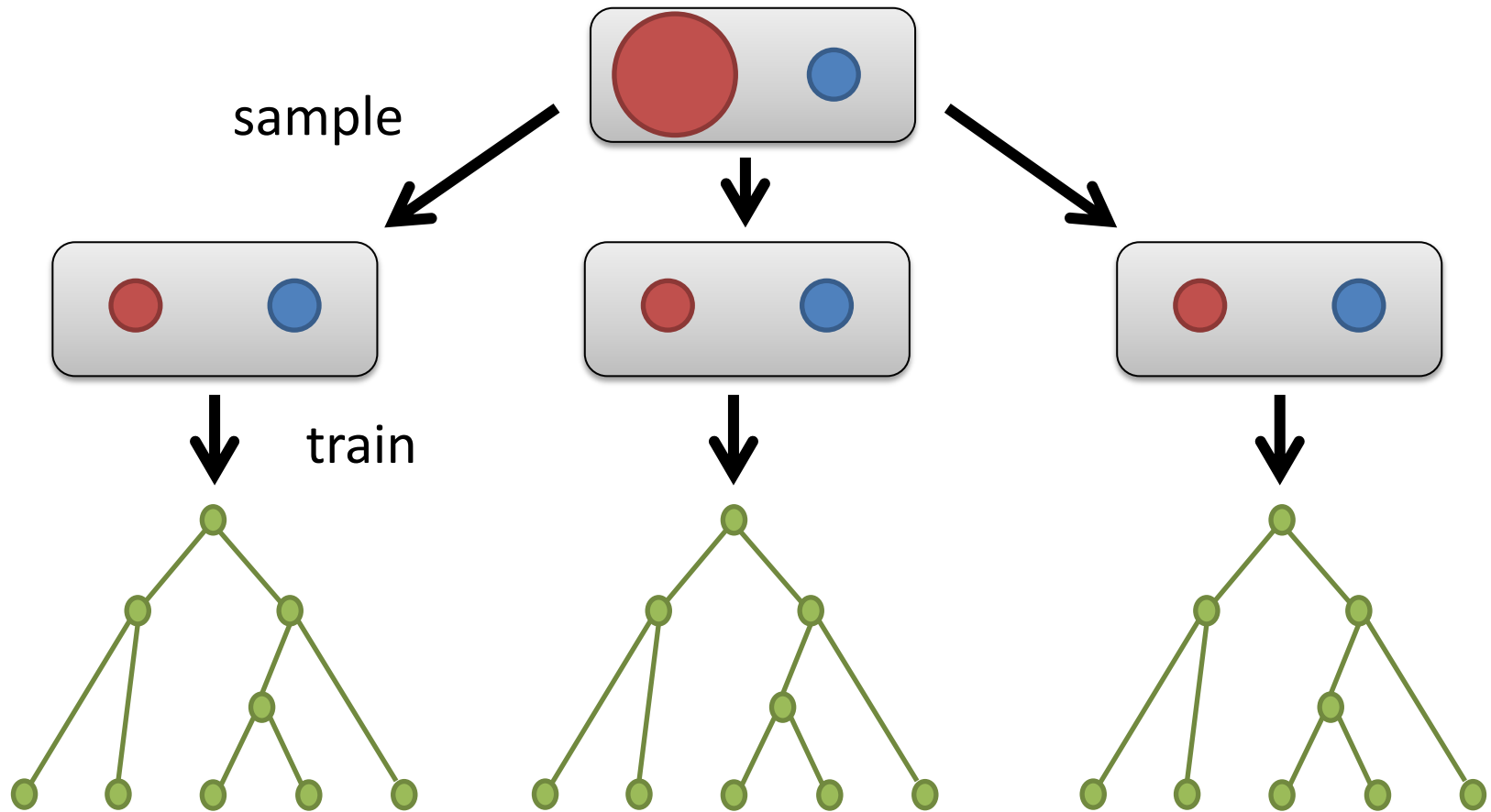


- Subsample:



- Subsample for each tree!

Random Forest Subsampling



Random Forest

- Similar to Bagging
- Easy to parallelize
- Packaged with some neat functions:
 - Out of bag error
 - Feature importance measure
 - Proximity estimation

Error Measures

- True positive (tp)
- True negative (tn)
- False positive (fp)
- False negative (fn)

		predicted	
		1	-1
true	1	tp	fn
	-1	fp	tn

TPR and FPR

- True Positive Rate:

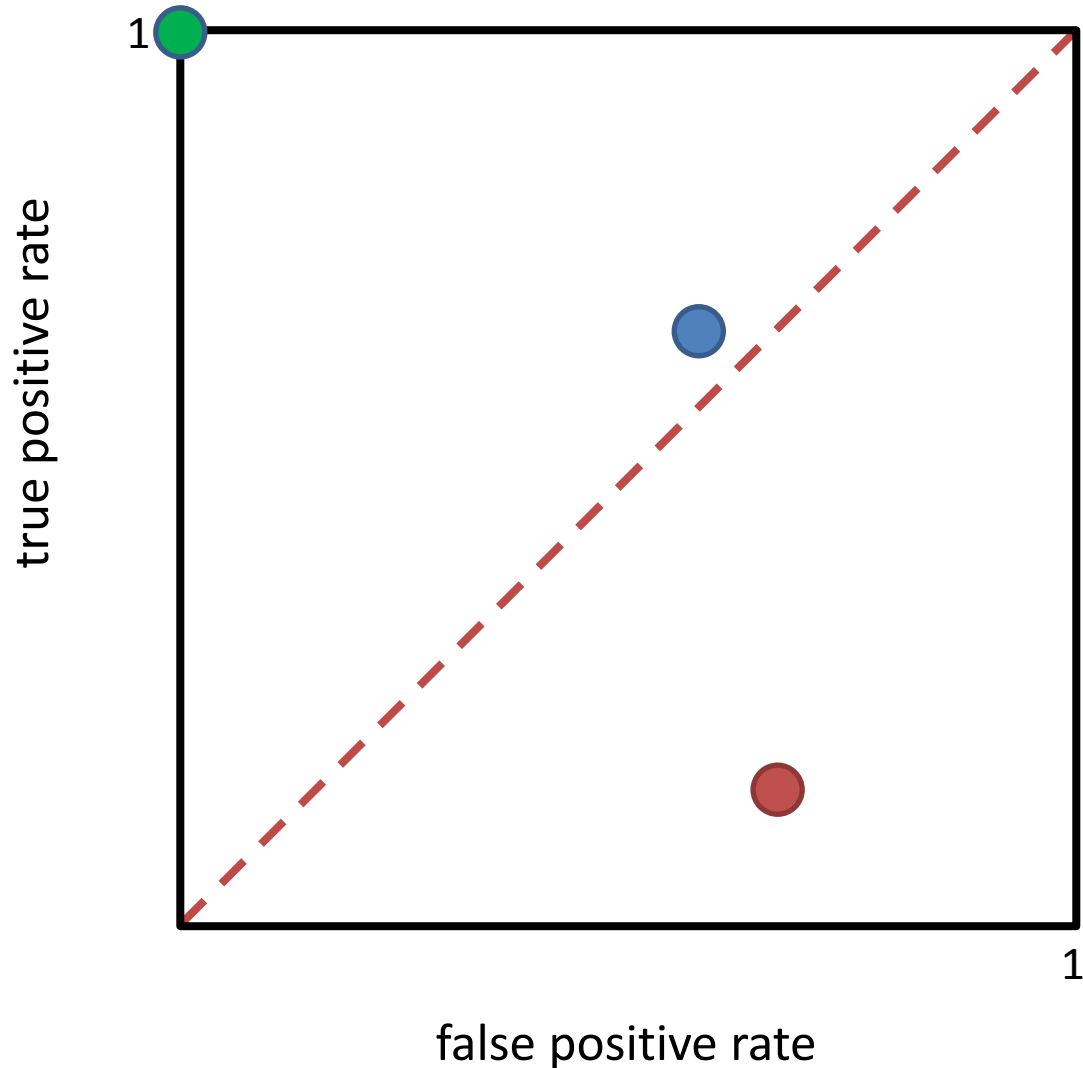
$$\frac{tp}{tp + fn}$$

- False Positive Rate:

$$\frac{fp}{fp + tn}$$

		predicted	
		1	-1
true	1	tp	fn
	-1	fp	tn

Receiver Operating Characteristic

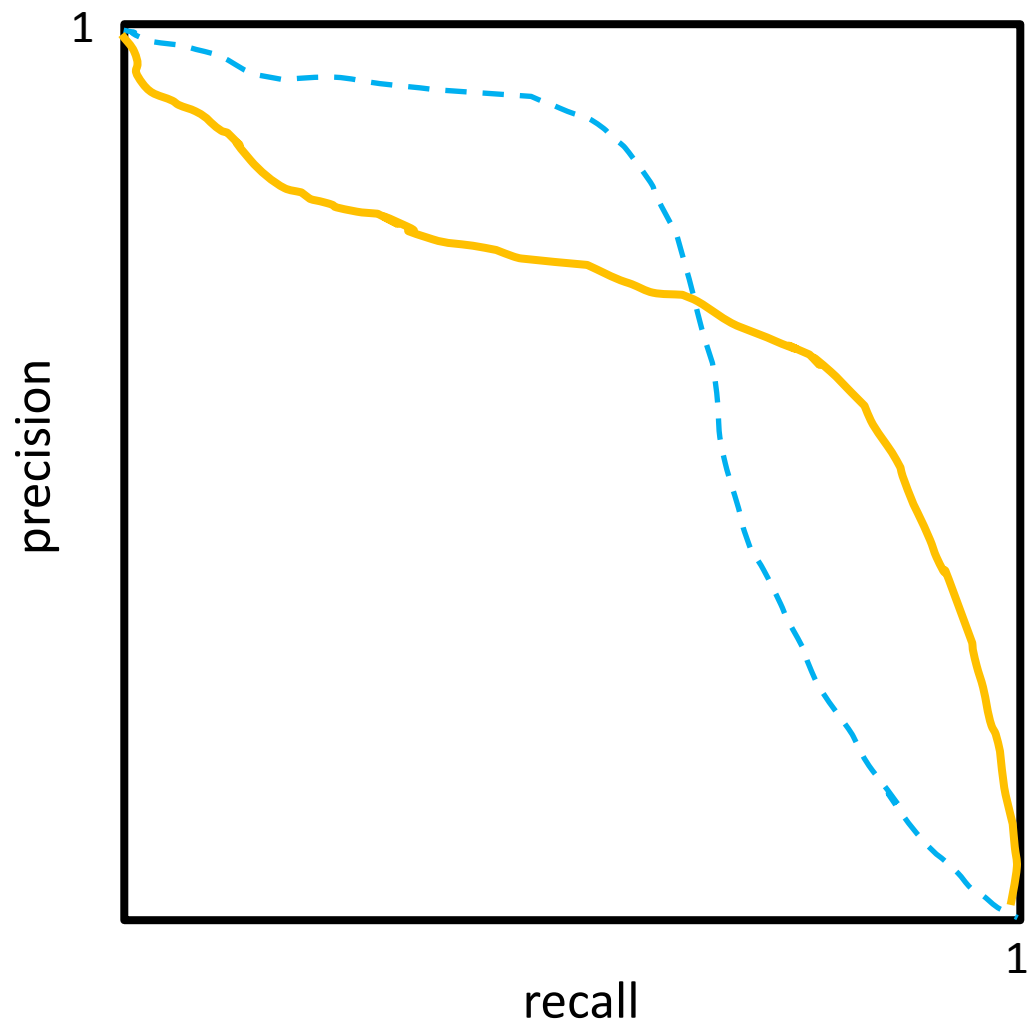


Precision Recall

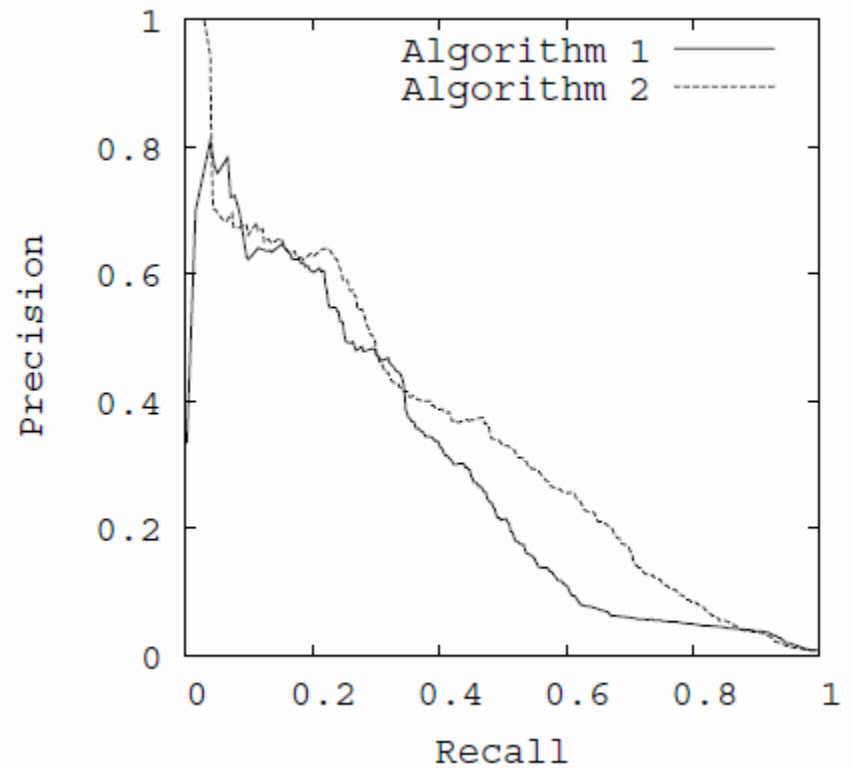
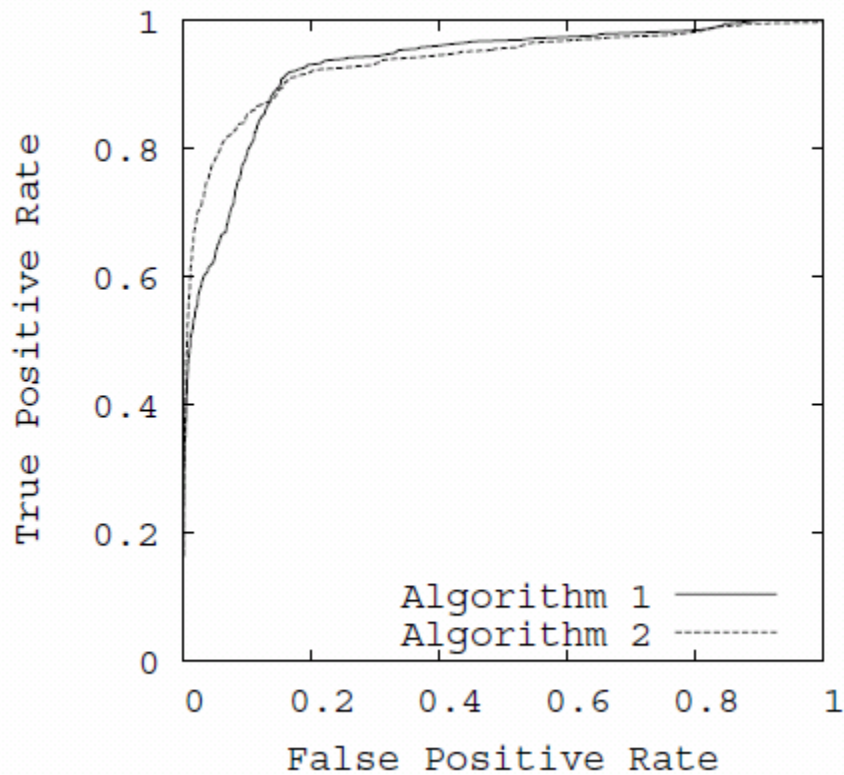
- Recall: $\frac{tp}{tp + fn}$
- Precision: $\frac{tp}{tp + fp}$

		predicted	
		1	-1
true	1	tp	fn
	-1	fp	tn

Precision Recall Curve



Comparison



J. Davis & M. Goadrich,
“The Relationship Between Precision-Recall and ROC Curves.”,
ICML (2006)

F-measure

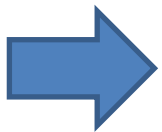
- Weighted average of precision and recall

$$F_{\beta} = \frac{(\beta^2 + 1) \cdot P \cdot R}{\beta^2 \cdot P + R}$$

- Usual case: $\beta = 1$
- Increasing β allocates weight to recall

Boosting

- Also ensemble method like Bagging
- But:
 - weak learners evolve over time
 - votes are weighted
- Better than Bagging for many applications



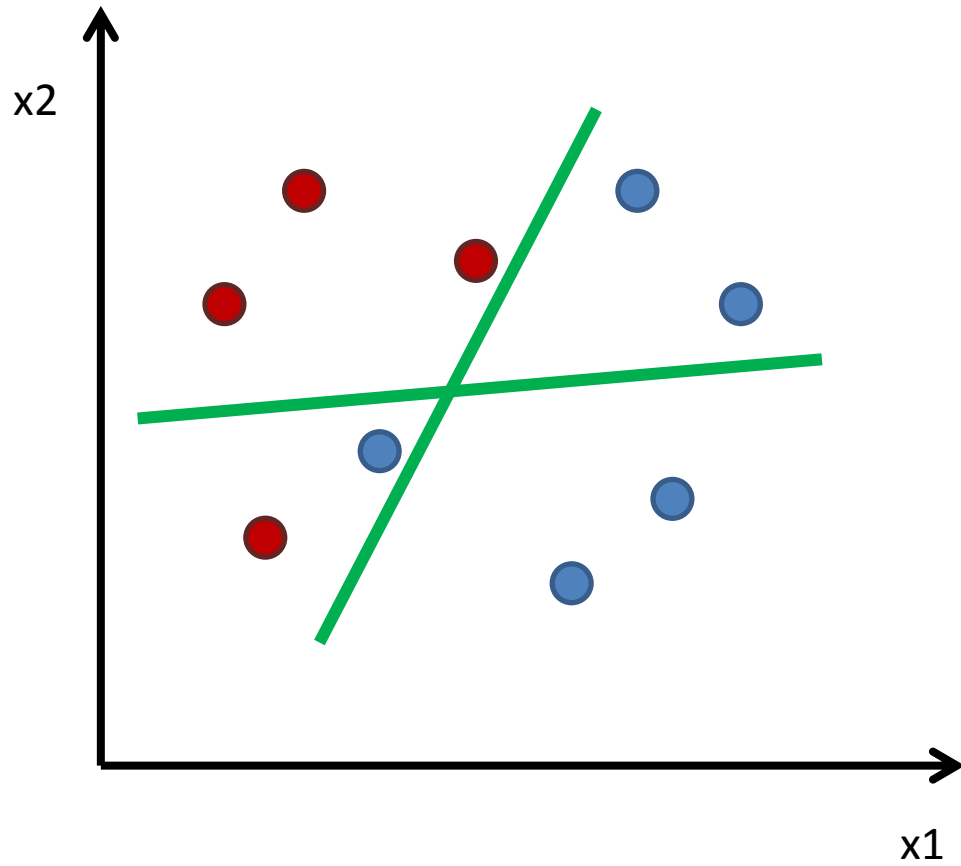
Very popular method

Boosting

“Boosting is one of the most powerful learning ideas introduced in the last twenty years.”

Hastie et al., “The Elements of Statistical Learning: Data Mining, Inference, and Prediction”, Springer (2009)

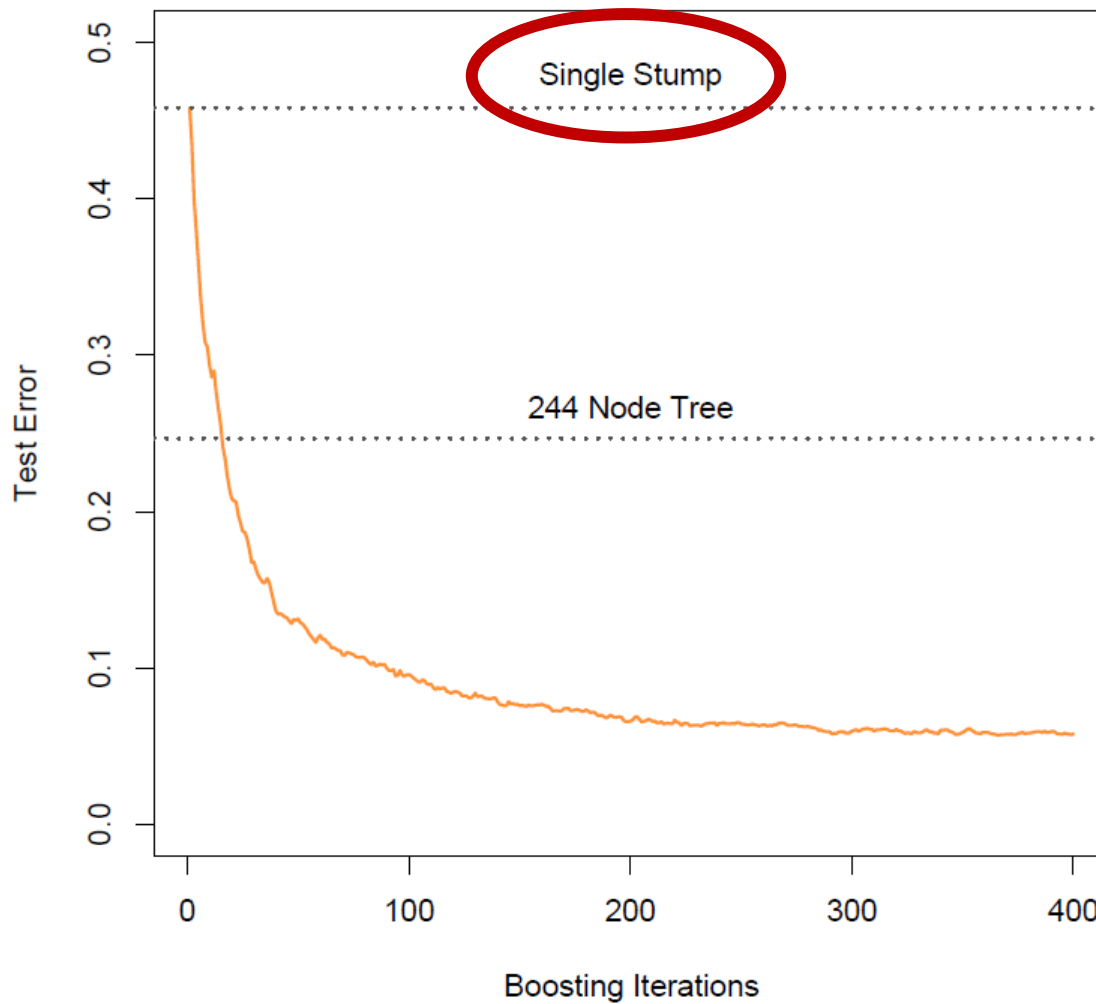
Adaboost



AdaBoost

- Initialize weights for data points
- For each iteration:
 - Fit classifier to training data
 - Compute weighted classification error
 - Compute weight for classifier from the error
 - Update weights for data points
- Final classifier is weighted sum of all single classifiers

AdaBoost



AdaBoost

AdaBoost in Action

Kai O. Arras

Social Robotics Lab, University of Freiburg

Nov 2009  Social Robotics Laboratory

AdaBoost

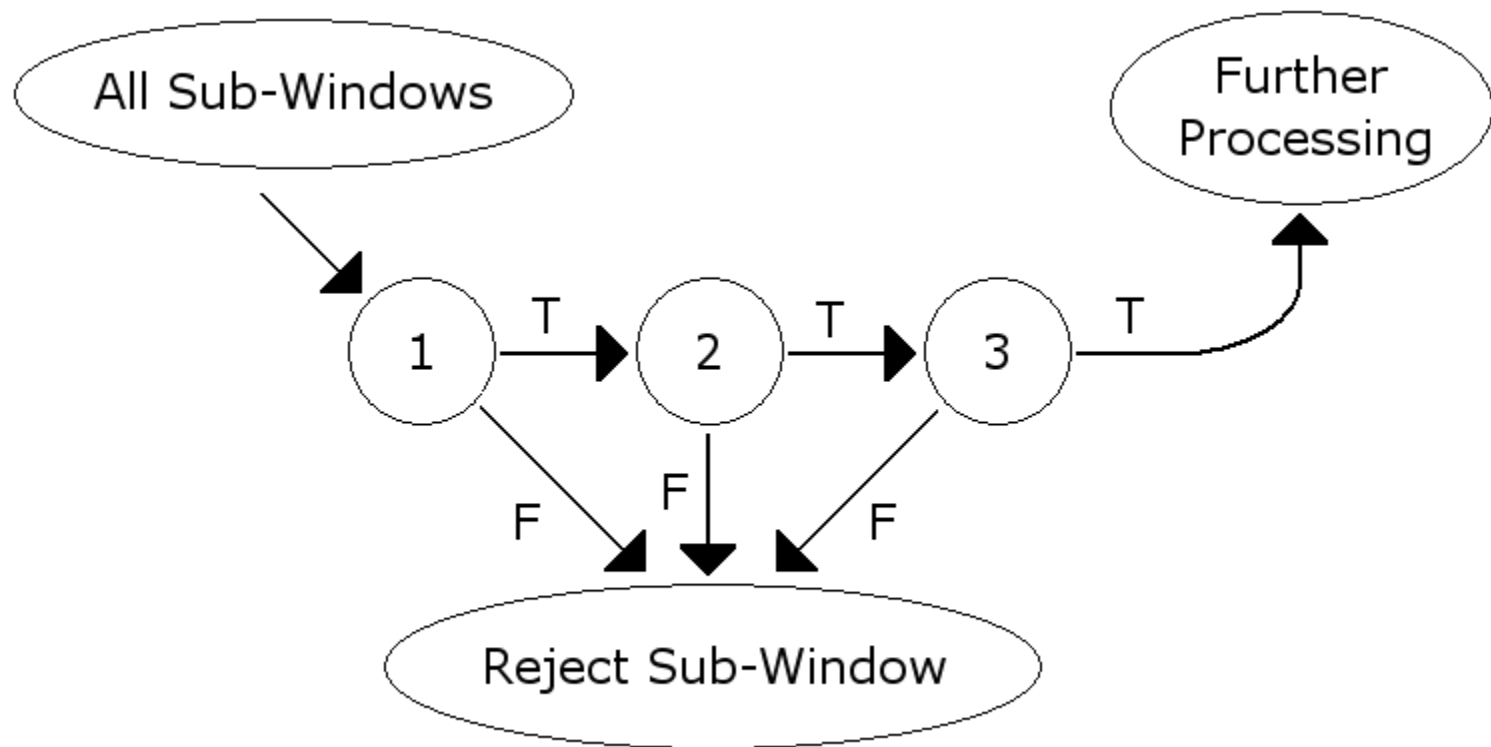
- Introduced by Freund and Schapire in 1995
- Worked great, nobody understood why!
- Then five years later (Friedman et al. 2000):
 - Adaboost minimizes exponential loss function.
- There still are open questions.

Cascade Classifier

- Ensemble methods are **strong**
- But prediction is **slow**
- **Solution**: Make prediction faster

Idea: Build a cascade

Cascade Classifier



Cascade Classifier

- Developed for fast object recognition
- Each classifier depends on its predecessor
- False positive rate:

$$F = \prod_{i=1}^K f_i$$

- Detection rate:

$$D = \prod_{i=1}^K d_i$$

Performance Example

- 10 stage classifier
- Want: detection rate = 0.9
- Need: detection rate 0.99 **per stage**
- But: high **false positive** rate is ok
 $(0.3^{10} \approx 6 \times 10^{-6})$

Viola Jones Face Detection



Viola Jones Face Detection

- Takes long to train
- Prediction in real time!
- Widely used today