Model Evaluation

Lecture 3

Outline

1. Estimating error
2. Types of mistakes, ROC Curves

Model Evaluation

- When evaluating a model, one metric we can use is error on the training set (***resubstitution error***)
  - # misclassified / # training instances
  - Is this useful? (e.g. consider 1-NN)

- This motivates a test set, with which to characterize generalization of the model
  - Important that the testing data is never used to build the model!
  - More testing data = tighter confidence on generalization estimate

Validation Set

- One approach in an ML-application pipeline is to use a ***validation*** dataset (could be a ***holdout*** from the training set)

- Each model is built using just training; the validation dataset is then used to compare performance and/or select model parameters

- But still, the final performance is only measured via an independent test set

More Training Data = Better

- In general, the greater the amount of training data, the better we expect the learning algorithm to perform
  - But we also want reasonable amounts of validation/testing data!
- So how do we not delude ourselves, achieve high performance, *and* a reasonable expectation of future performance?

k-Fold Cross-Validation

- Basic approach
  - Divide the data into k randomly selected partitions (typically 10)
  - For each, use the fold as test data, the remainder as training data (i.e. repeat the train/test process k times)
  - Average results
- To control for unfortunate outcomes in random selection, consider repeating (e.g. 10 x 10-fold cross validation = 100 train/test)
  - Expensive!

Other Estimates

- Leave-One-Out
  - n-fold cross-validation, keeping only a single test instance per evaluation
- The 0.632 Bootstrap $$\lim_{n \to \infty} (1 - \frac{1}{n})^n = \frac{1}{e} \approx 0.368$$
  - Sample the training set with substitution n times: becomes the training set
  - Any instance not selected becomes a test instance
  - Estimate = 0.632(test error) + 0.368(train error)
  - Average over several samples

Accuracy Issues

- Accuracy is often too simple a metric when characterizing algorithm performance

- Typical complications:
  - Skewed class distribution (change over time!)
  - Unequal classification error costs

- Examples
  - Airline screening
  - Fraud detection
  - Medical diagnosis

Classifier Performance

- Let us consider a binary classifier with only two classes: **P**ositive, **N**egative
- Now consider the four possible outcomes (***confusion matrix***)

True Class

Classifier Output	Positive	Negative
Positive	**T**rue **P**ositive	**F**alse **P**ositive
Negative	**F**alse **N**egative	**T**rue **N**egative

Common Performance Metrics

		True Class	
		Positive	Negative
Classifier Output	Positive	True Positive	False Positive
	Negative	False Negative	True Negative

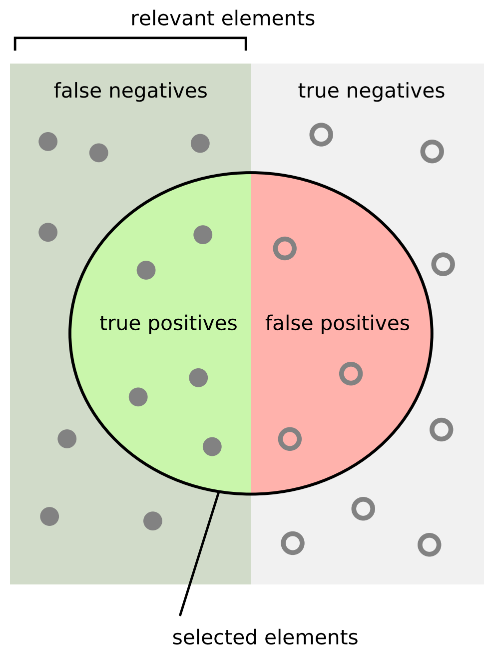
$$\text{accuracy} = \frac{\text{TP} + \text{TN}}{\text{P} + \text{N}}$$

$$\text{fp rate} = \frac{\text{FP}}{\text{N}}$$

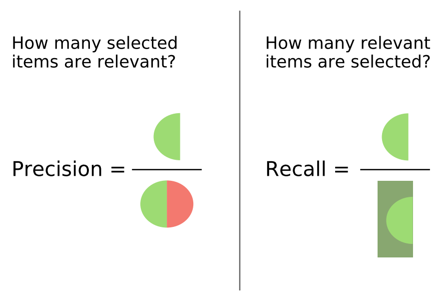
$$\text{tp rate} = \text{recall} = \frac{\text{TP}}{\text{P}} \qquad \text{precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}$$

$$\text{F-measure} = \frac{2}{1/\text{precision} + 1/\text{recall}}$$

Precision/Recall



$$\text{precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}$$

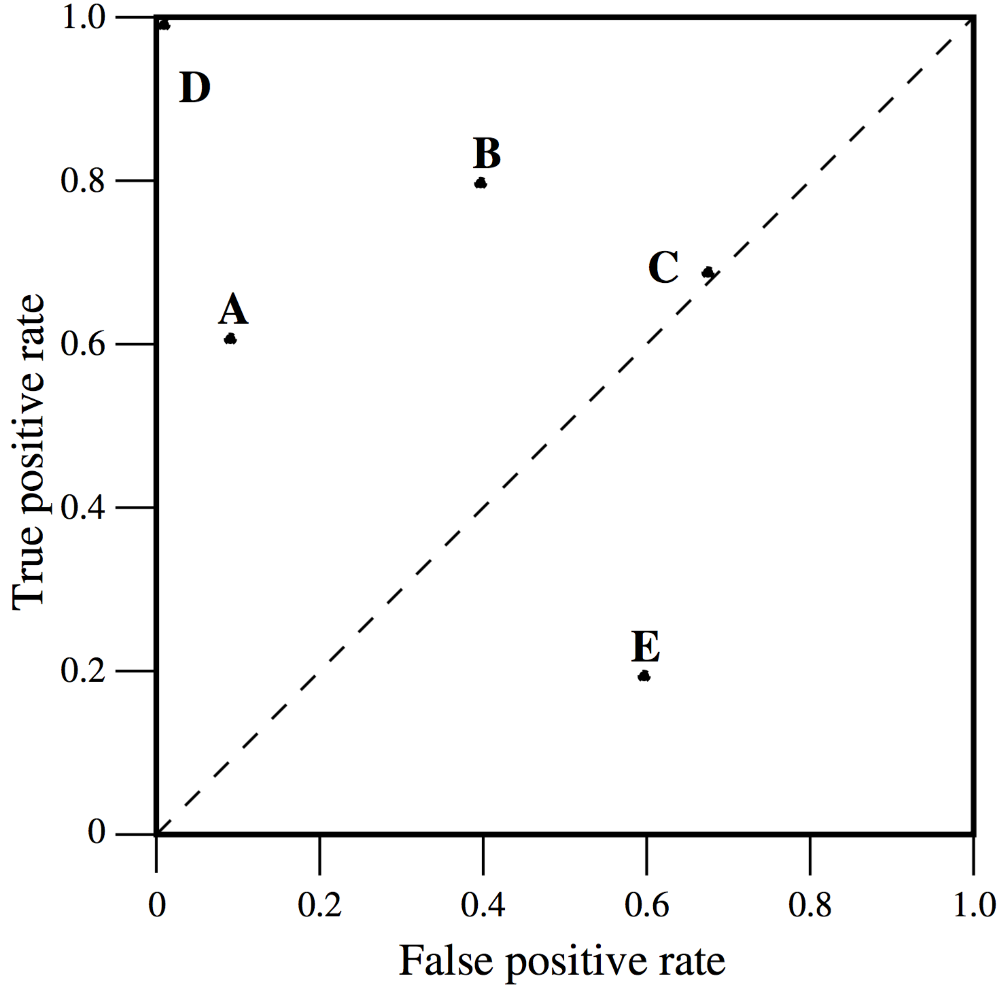


$$\text{tp rate} = \text{recall} = \frac{\text{TP}}{\text{P}}$$

ROC Graph

Receiver Operating Characteristics

$$\mathrm{TP}/\mathrm{P}$$

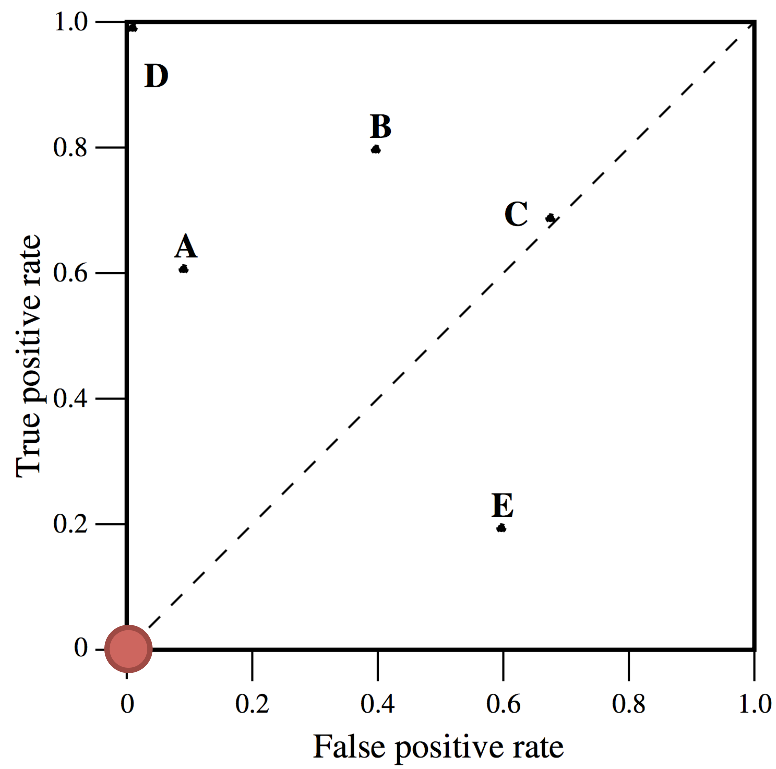


$$\mathrm{FP}/\mathrm{N}$$

Reading ROC: (0,0)

Never issue a positive classification

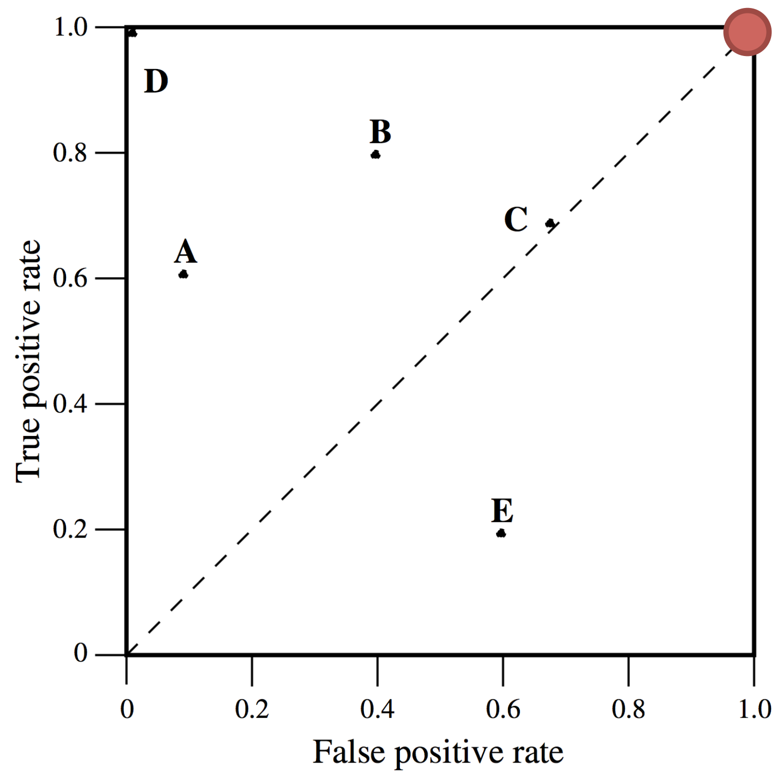
- No possibility of FP
- But also no TP…



Reading ROC: (1,1)

Always issue a positive classification

- Catches all the TP
- But also full FP…



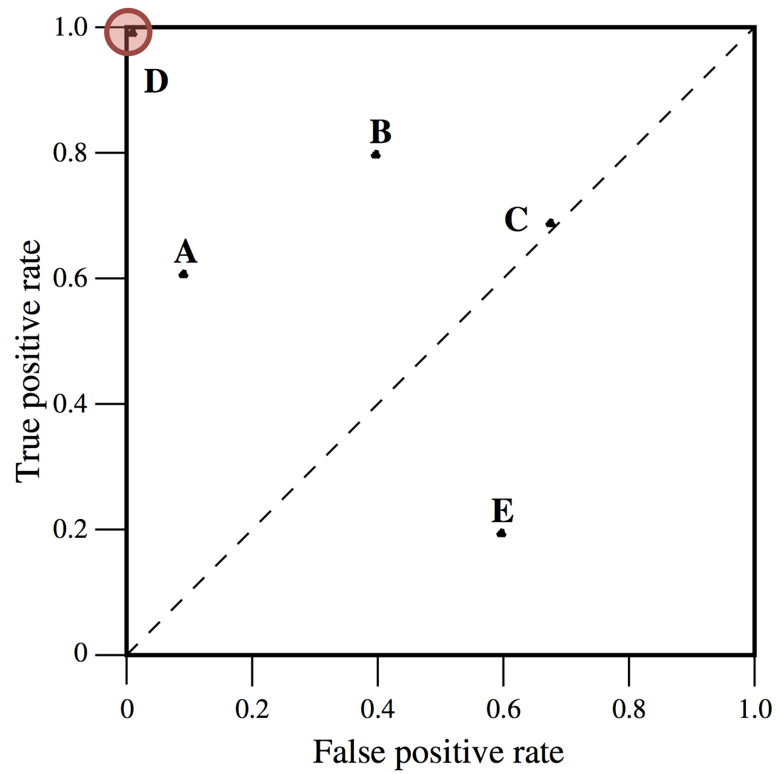
Reading ROC: (0,1)

Ideal classifier

- Catches all the TP
- But no FP's

Given two points on the graph, closer to (0,1) is considered "better"

- Useful for tuning meta-parameters



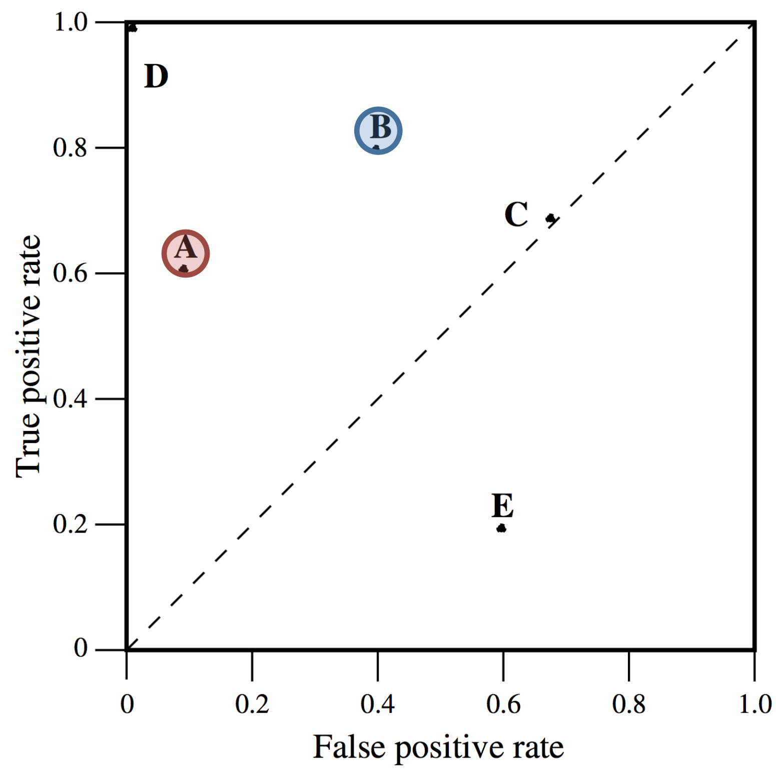
Conservative vs. Liberal

Conservative

- Positive classification only with strong evidence, lower FP
- More interesting in situations with many negative examples

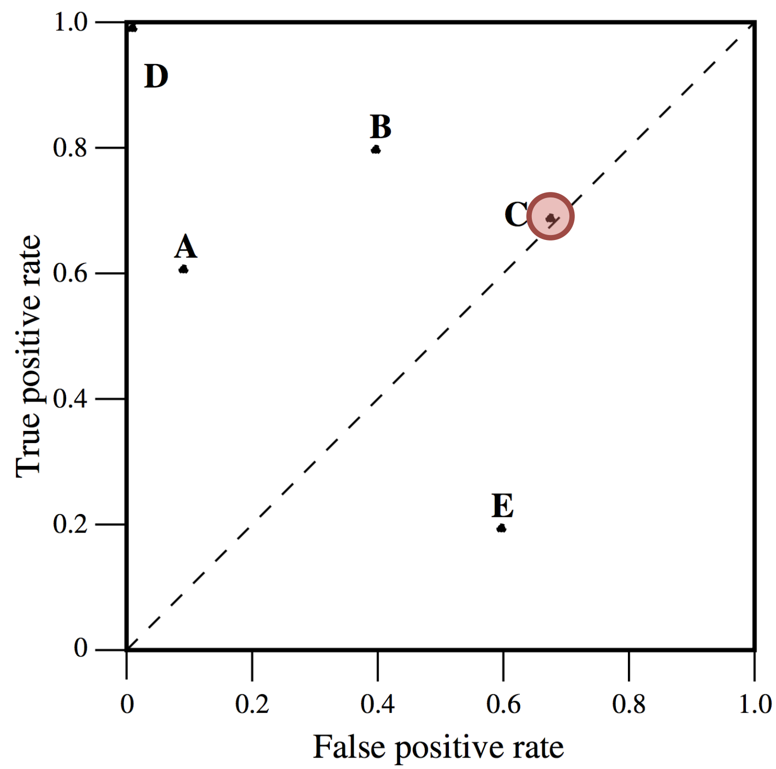
Liberal

- Positive classification with weaker evidence, higher FP



Reading ROC: Random

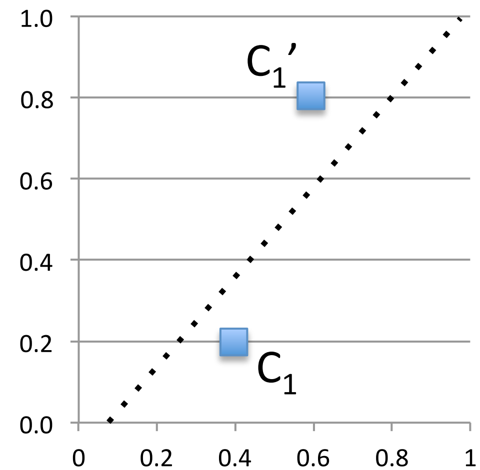
- The line y = x represents random selection
  - Classifier has NO information
- Anything below has information, but using it “poorly”
  - How to better use E?



Example

C_1 Output	Truth
F	T
F	T
F	T
F	T
T	T
T	F
T	F
F	F
F	F
F	F

- Plot C_1 on an ROC curve



Ranking Classifier

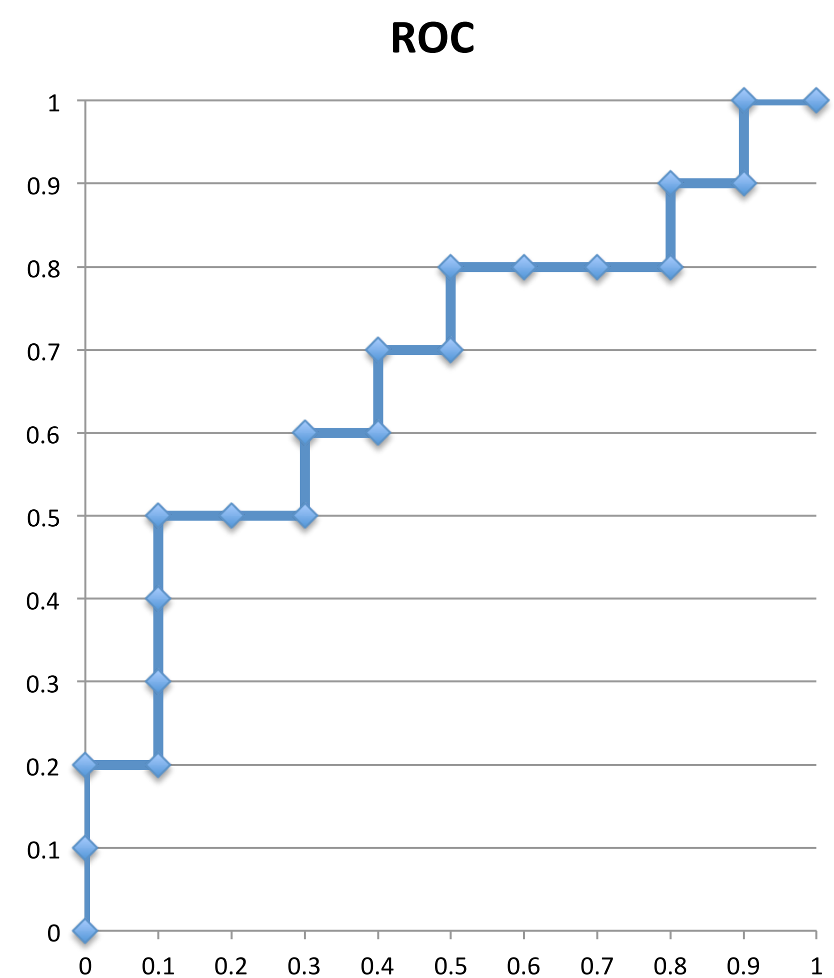
- Many algorithms can output not only a class, but also a "score"
  - Sometimes this is probability/confidence, otherwise an arbitrary value sufficient to rank
    - Such as kNN voting!
- Committing to a classification threshold yields a point in ROC space
  - Incrementally shifting the threshold yields an ROC curve, characterizing algorithm performance

Example

Inst#	Class	Score	Inst#	Class	Score
1	p	.9	11	p	.4
2	p	.8	12	n	.39
3	n	.7	13	p	.38
4	p	.6	14	n	.37
5	p	.55	15	n	.36
6	p	.54	16	n	.35
7	n	.53	17	p	.34
8	n	.52	18	n	.33
9	p	.51	19	p	.30
10	n	.505	20	n	.1

- Produce the ROC curve
- What is the optimal threshold in terms of accuracy?

0.54 = 70%



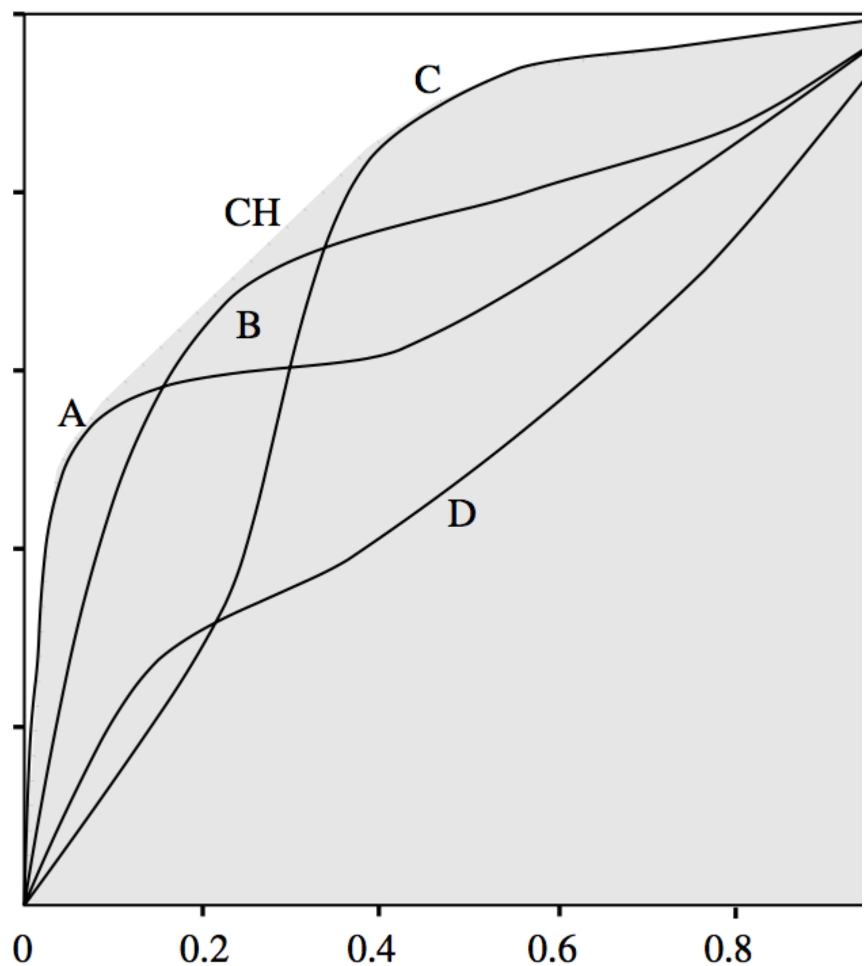
ROC Invariance

- Because ROC curve is based upon TP/FP *rates*, the representation is invariant to class distribution and error costs

- This can be ideal for choosing algorithms for applications in dynamic environments

Convex Hull

- If we are comparing multiple algorithms in ROC space, the ***convex hull*** identifies the "best" classifier under "any" conditions
- Can disregard classifiers not on the CH (e.g. B, D)
- Can produce classifiers on the CH via proportional sampling

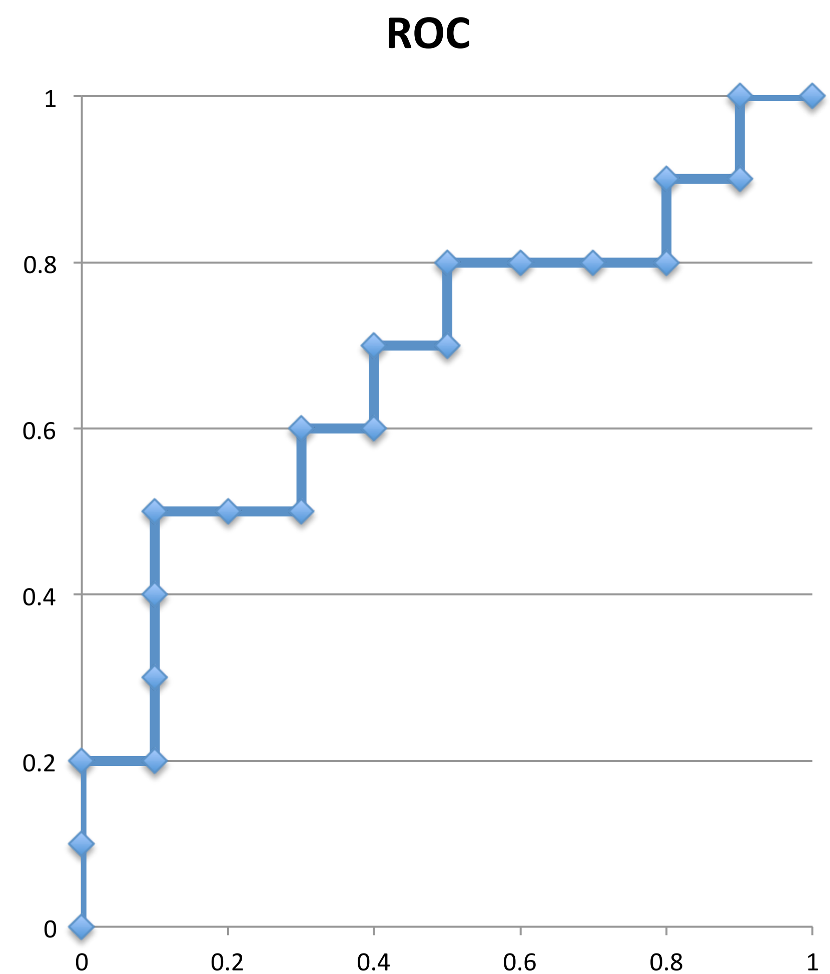


Area Under an ROC Curve (AUC)

- To compare classifiers, we can reduce the 2D ROC curve to a scalar ***AUC***
  - Value between [0, 1]
    - Review: what range matters?
- FYI: related to other statistical tests
  - Wilcoxen test of ranks
  - Gini + 1 = 2 x AUC

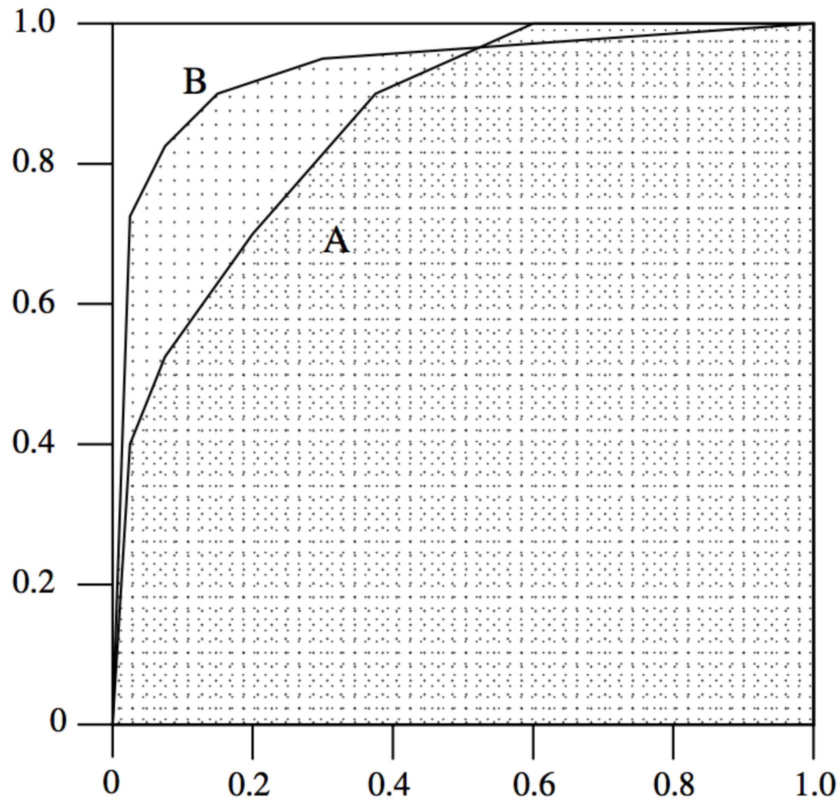
Example

Inst#	Class	Score	Inst#	Class	Score
1	p	.9	11	p	.4
2	p	.8	12	n	.39
3	n	.7	13	p	.38
4	p	.6	14	n	.37
5	p	.55	15	n	.36
6	p	.54	16	n	.35
7	n	.53	17	p	.34
8	n	.52	18	n	.33
9	p	.51	19	p	.30
10	n	.505	20	n	.1



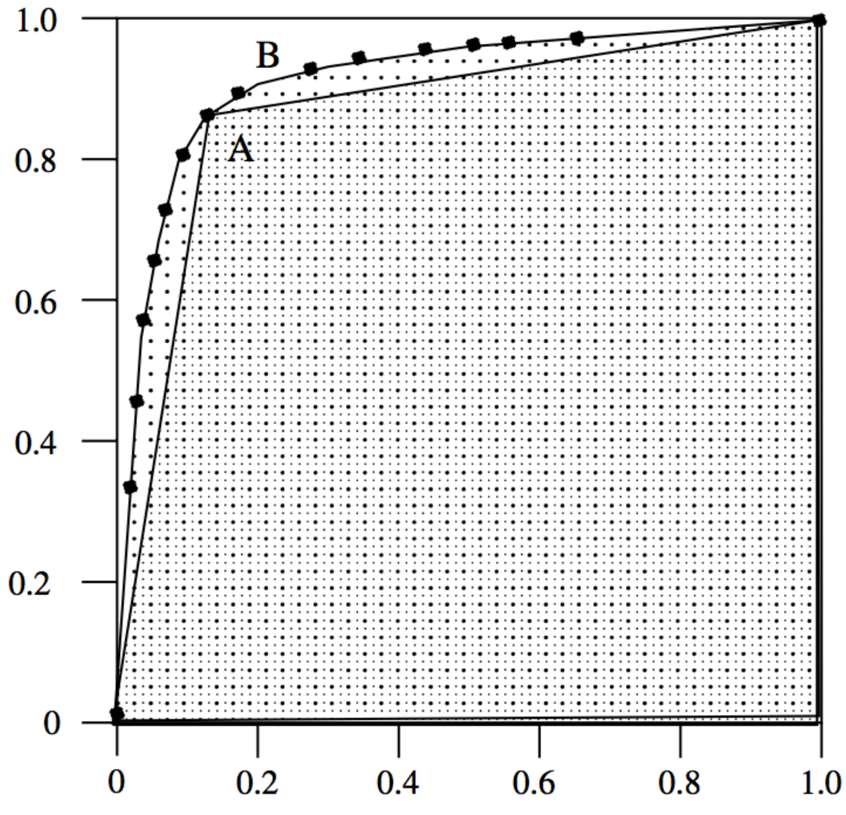
- Compute the AUC
  - 0.68

AUC Limitations



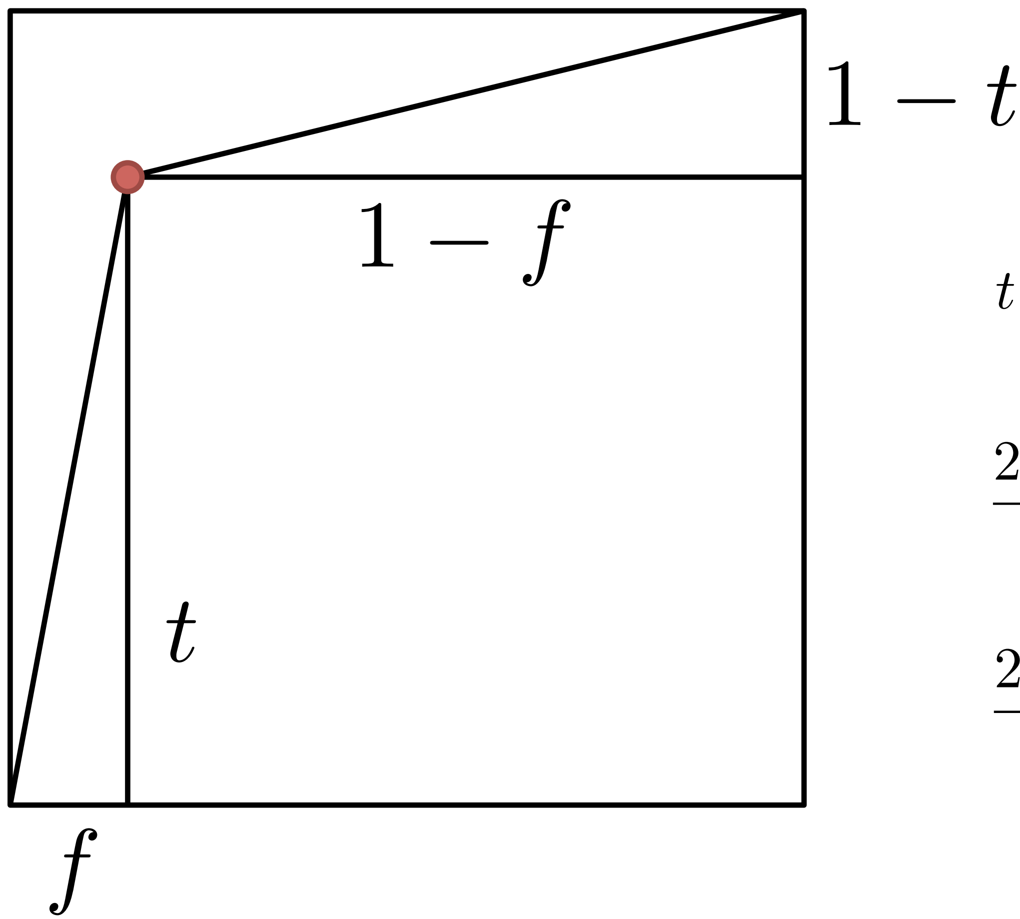
- B generally better
  - $AUC_B > AUC_A$
- But A is better for particular FP range (>0.6)

AUC for Discrete vs. Scoring



$$\text{AUC} = \frac{t - f + 1}{2}$$

Proof



$$t(1-f)+\frac{tf}{2}+\frac{(1-f)(1-t)}{2}$$

$$\frac{2t(1-f)+tf+(1-f)(1-t)}{2}$$

$$\frac{2t-2tf+tf+1-t-f+ft}{2}$$

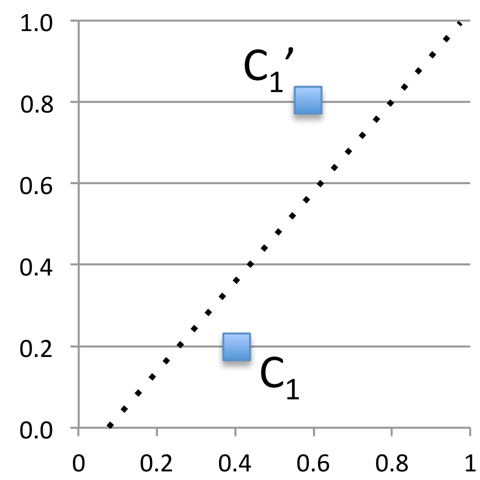
$$\frac{t-f+1}{2}$$

Example

C_1 Output	Truth
F	T
F	T
F	T
F	T
T	T
T	F
T	F
F	F
F	F
F	F

Compute the AUC of the classifier that makes best use of the information in C_1

0.6



ROC Issues Not Covered

- Efficient generation
- Ideal classifiers under particular conditions
- Confidence over ROC curves
- Multi-class classifiers

Summary

- When dealing with a fixed training set, make use of evaluation techniques to estimate error (k-fold cross validation)

- To characterize/compare classifier performance independent of class skew/ error costs, make use of ROC curves