CS6510
Applied Machine Learning

Classifier Evaluation

13 Aug 2016

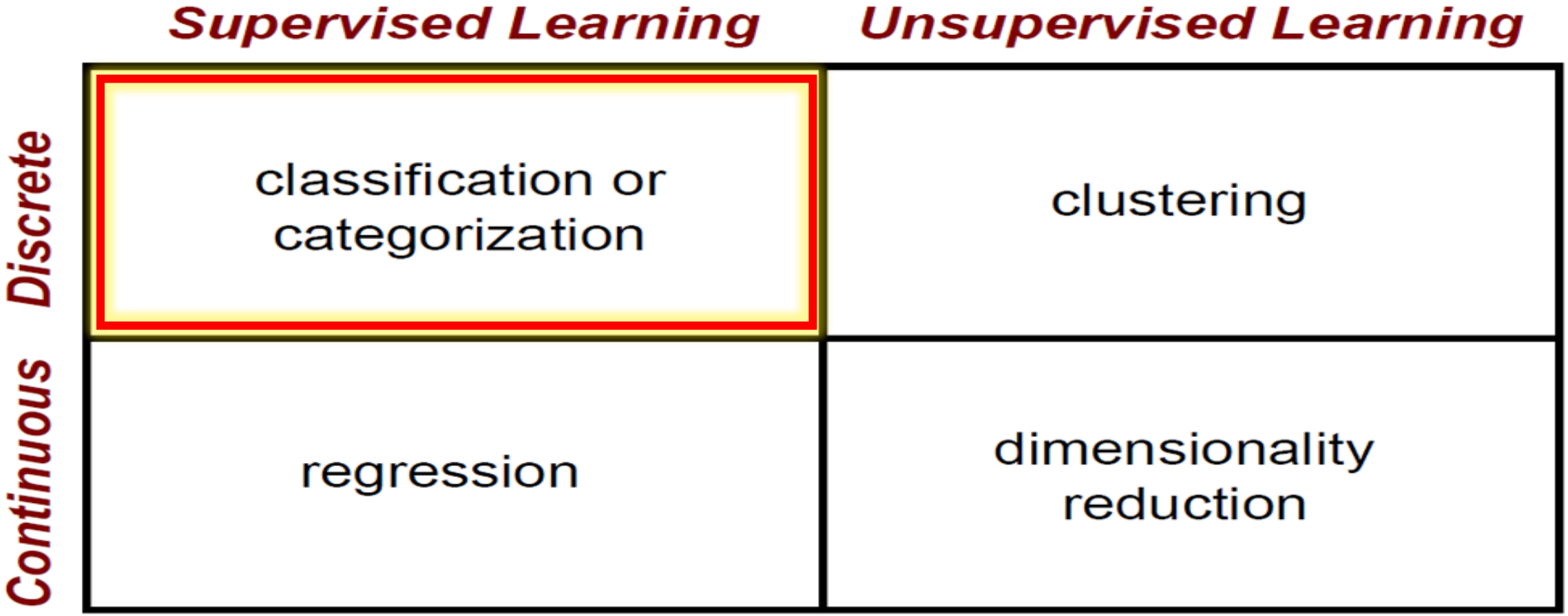
Vineeth N Balasubramanian

आई आई टी हैदराबाद
IIT Hyderabad

Administrivia

- Google Classroom – please join
- Class Lectures - please take notes

ML Problems: Recall



Classification Methods

- k-Nearest Neighbors
- Decision Trees
- Naïve Bayes
- Support Vector Machines
- Logistic Regression
- Neural Networks
- Ensemble Methods (Boosting, Random Forests)

How to evaluate? Standard measure: Accuracy

Training vs Generalization Error

- Training Error
  - Not very useful
  - Relatively easy to obtain low error
- Generalization Error
  - How well we do on future data

$$E_{train} = \frac{1}{n}\sum_{i=1}^{n} error(f_D(\mathbf{x}_i), y_i)$$

same? different by how much?

training examples

value we predicted

true value

$$E_{gen} = \int error(f_D(\mathbf{x}), y)\, p(y, \mathbf{x})\, d\mathbf{x}$$

over all possible x,y

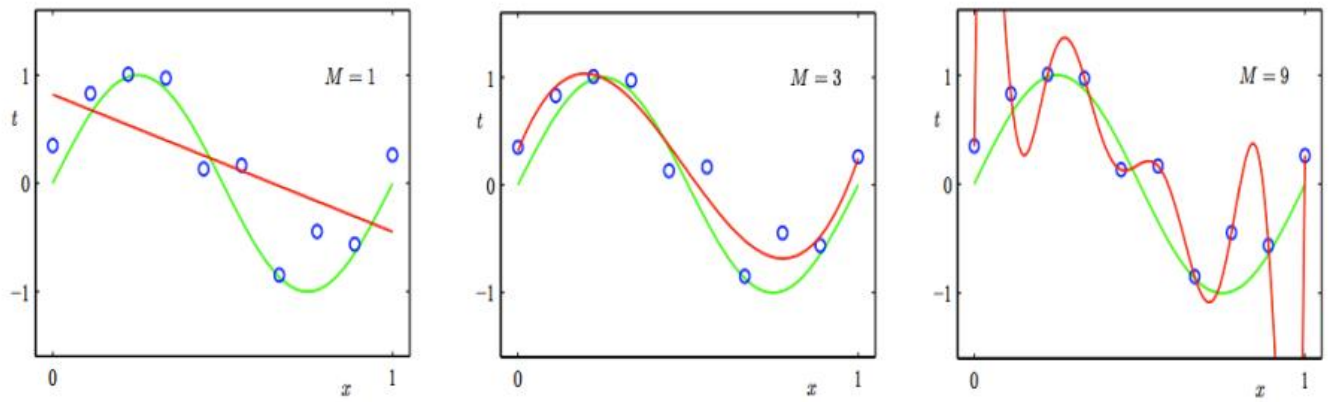
error as before

how often we expect to see such x and y

How to compute generalization error?

Underfitting and Overfitting

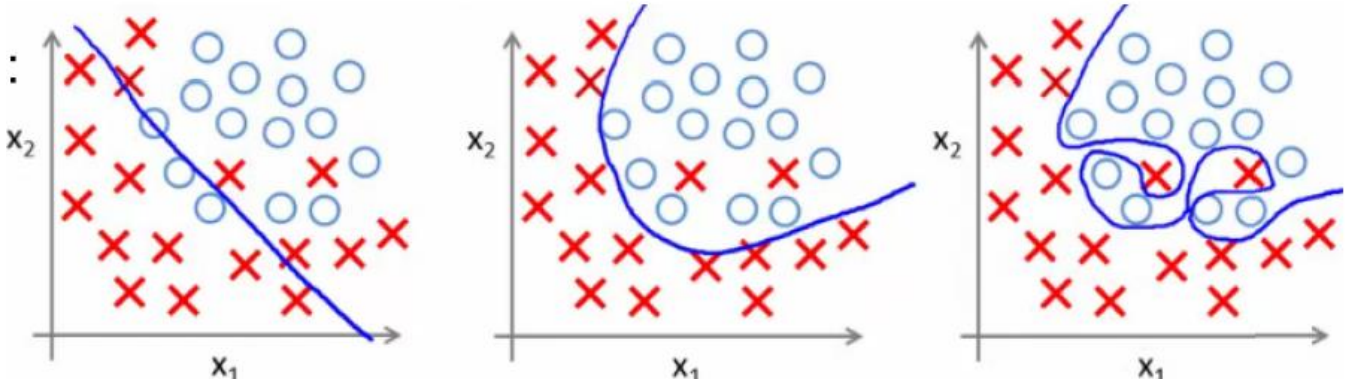
Regression



predictor too inflexible: cannot capture pattern

predictor too flexible: fits noise in the data

Classification



Estimating Generalization Error

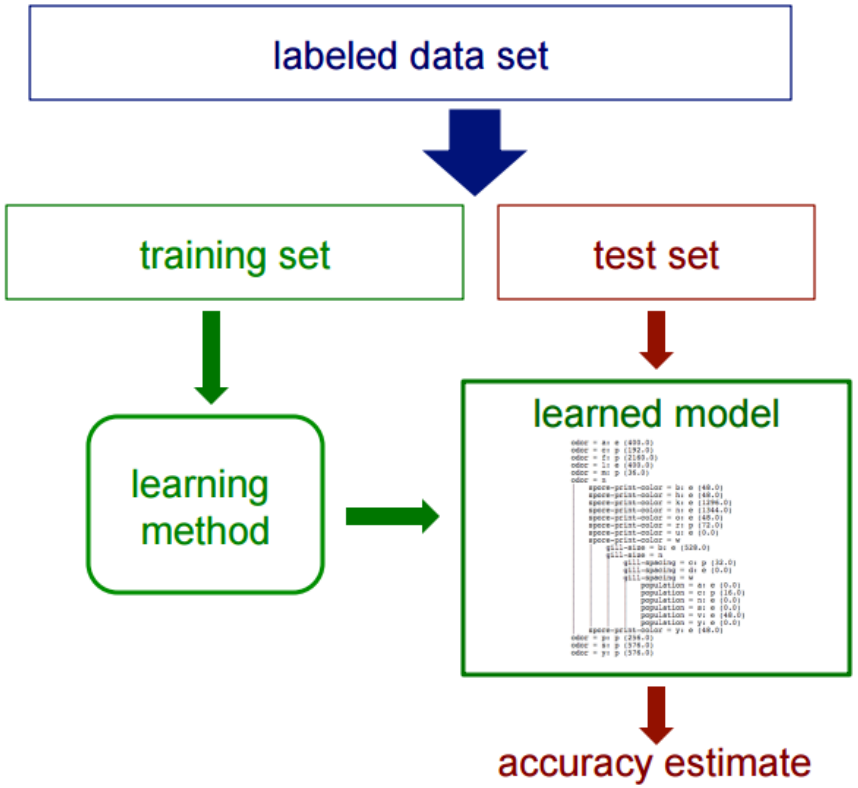
- Testing Error
  - Set aside part of training data (testing set)
  - Learn a predictor without using any of this test data
  - Predict values for testing set, compute error
  - This is an estimate of generalization error

$$E_{test} = \frac{1}{n} \sum_{i=1}^{n} error(f_D(\mathbf{x}_i), y_i)$$

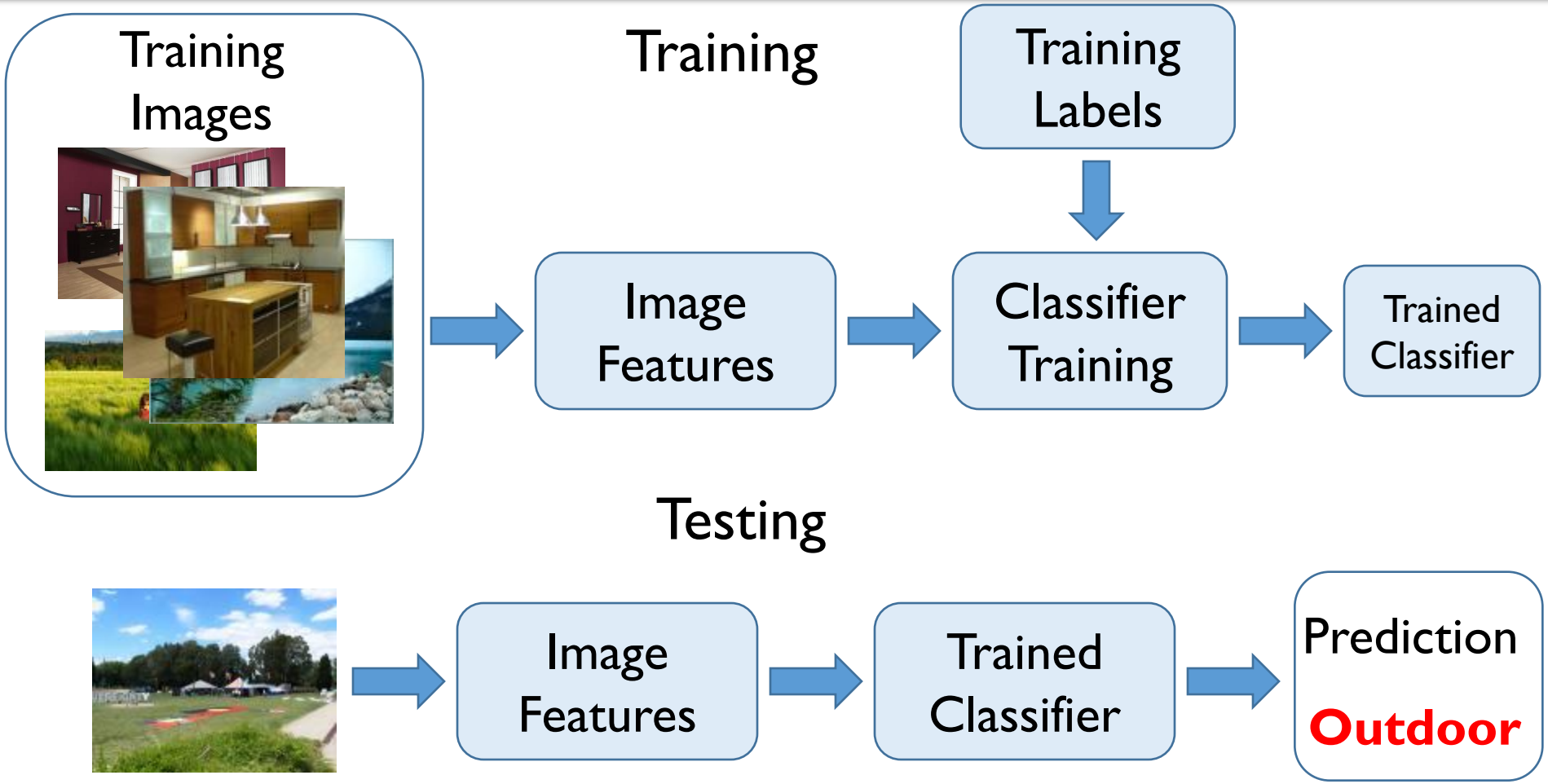
over testing set

Estimating Generalization Error

- Getting an unbiased estimate of the accuracy of a learned model



Example: Image Classification



Source: Derek Hoiem

CS6510
Applied Machine Learning

Classifier Evaluation

20 Aug 2016

Vineeth N Balasubramanian

आई आई टी हैदराबाद
IIT Hyderabad

Training, Validation, Test Sets

Training set

- NB: Count frequencies, DT: Pick attributes to split on

Validation set

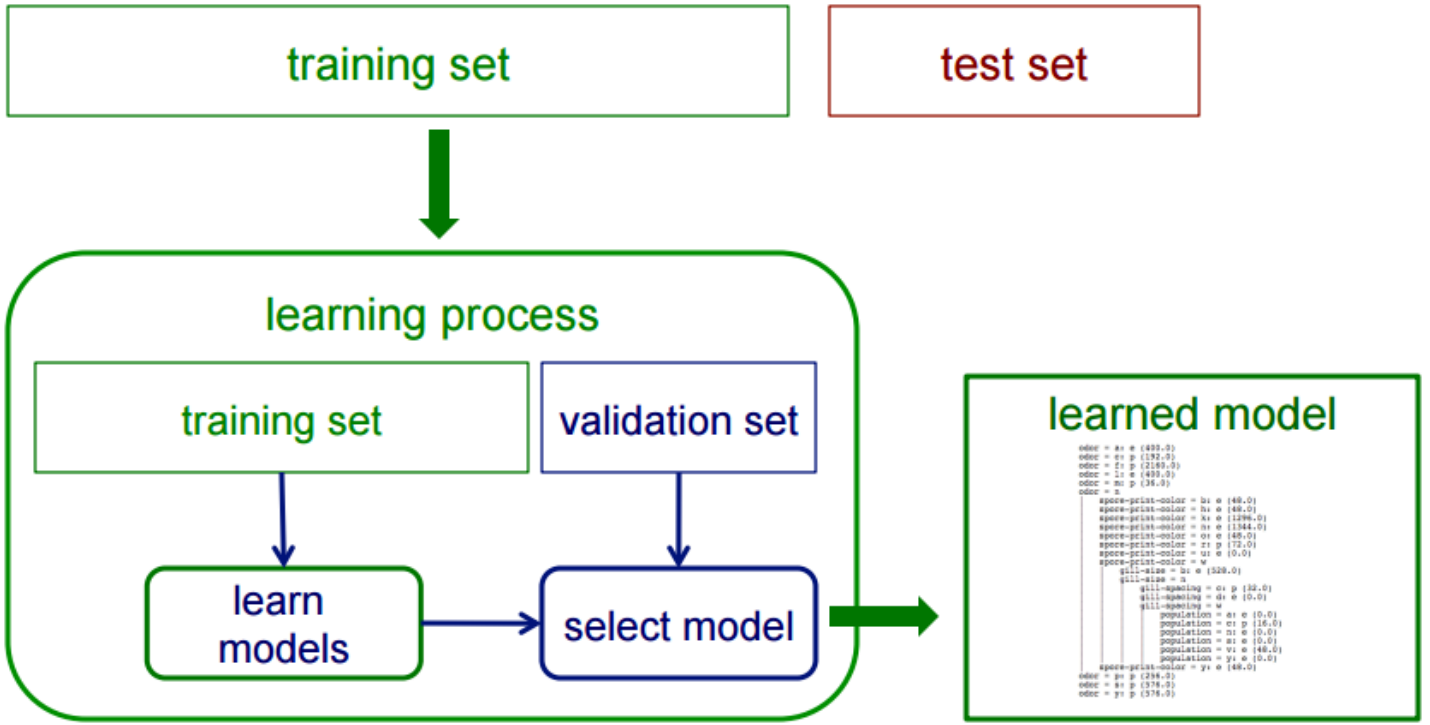
- Pick best-performing algorithm (NB vs DT vs..)
- Fine-tune parameters (Tree depth, k in kNN, c in SVM)

Testing set

- Run multiple trials and average

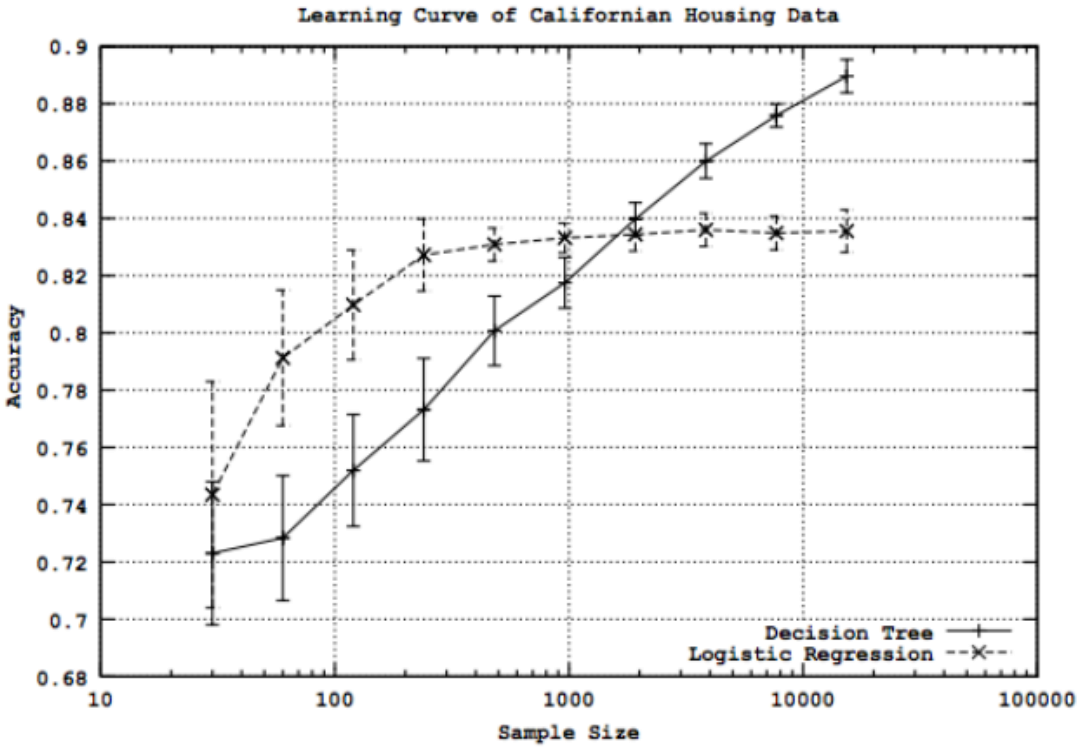
Use of Validation Sets

- If we want unbiased estimates of accuracy during the learning process:



Choosing Training, Validation, Test Sets

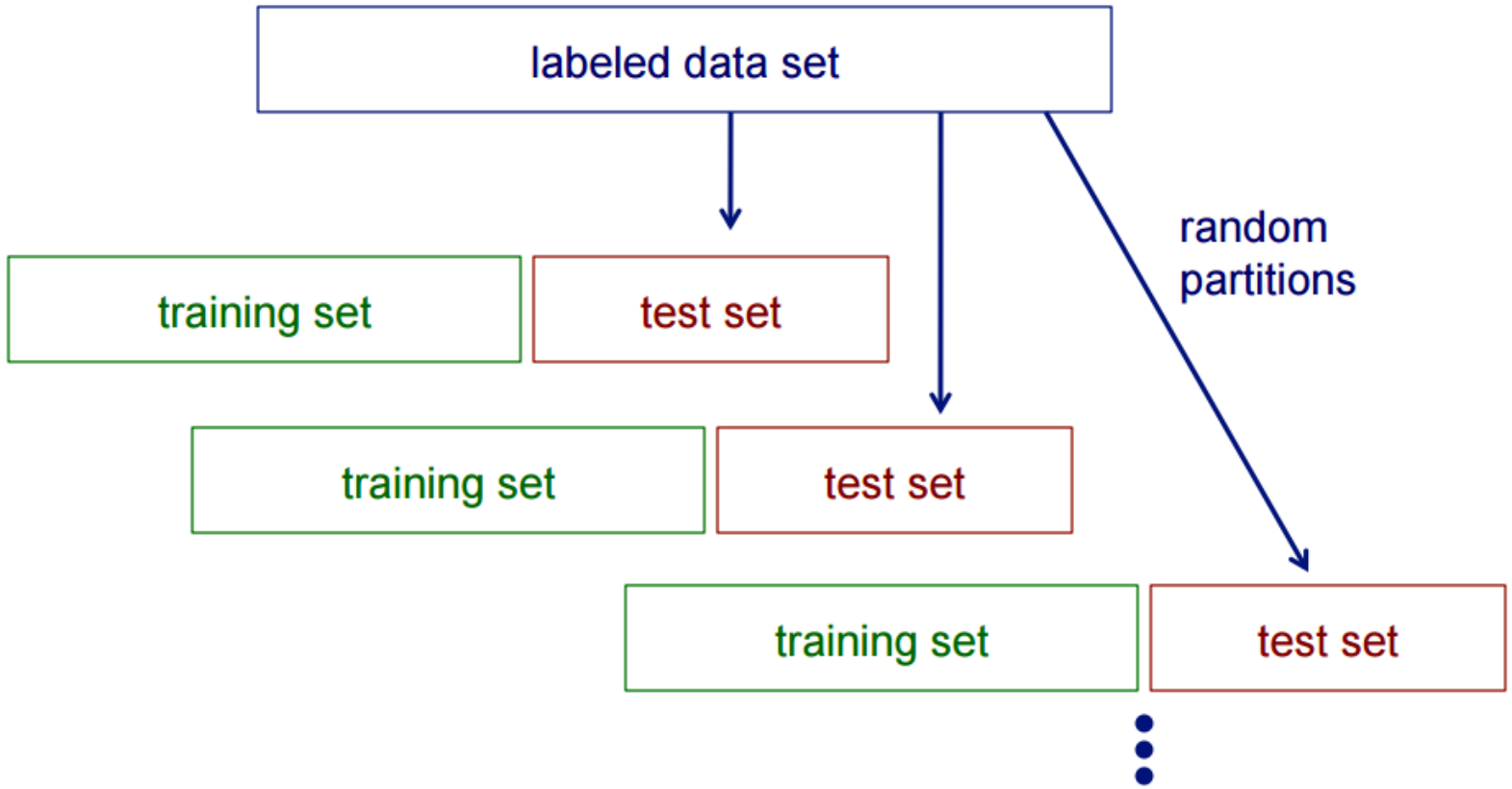
- Split randomly to avoid bias
- Large test set -> estimate future error as accurately as possible (vs) Large training set => better estimates
- How large should a training set be?
  - Study accuracy/error (vs) training set size



Courtesy: Perlich et al. Journal of Machine Learning Research, 2003

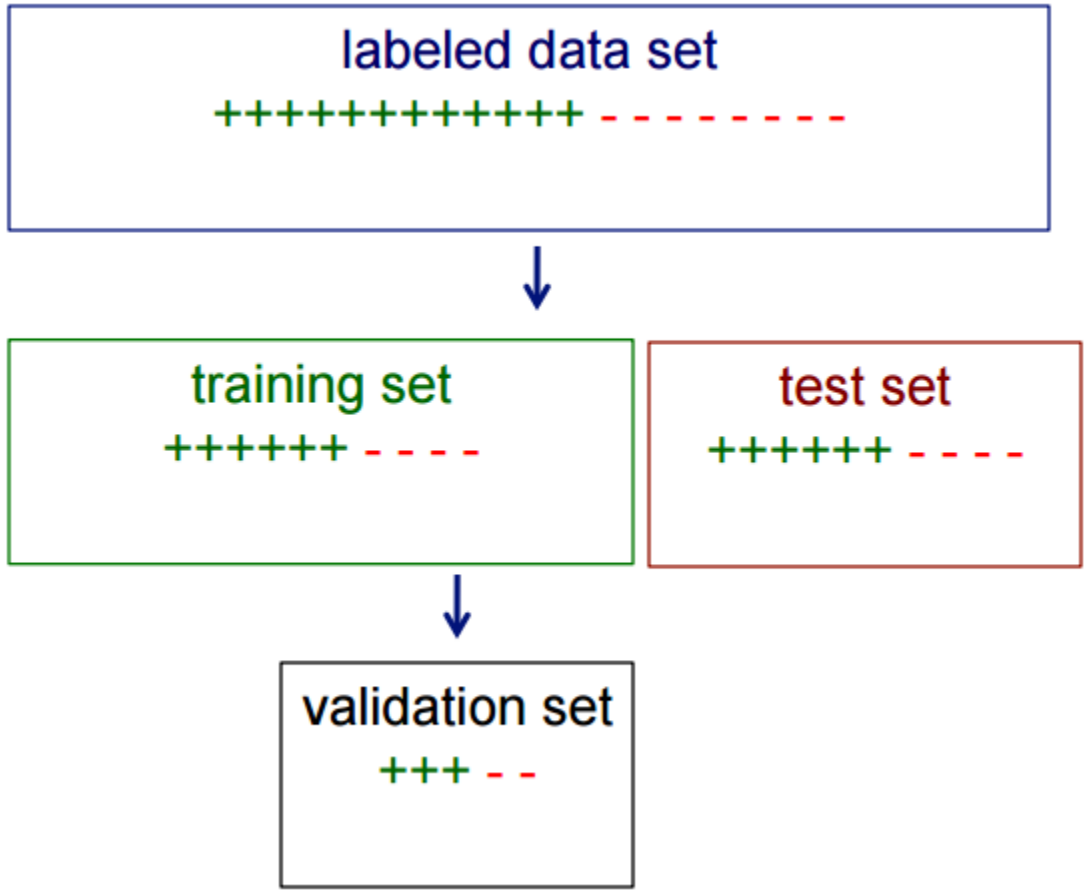
Random Resampling

- We can artificially increase training set size using random resampling:



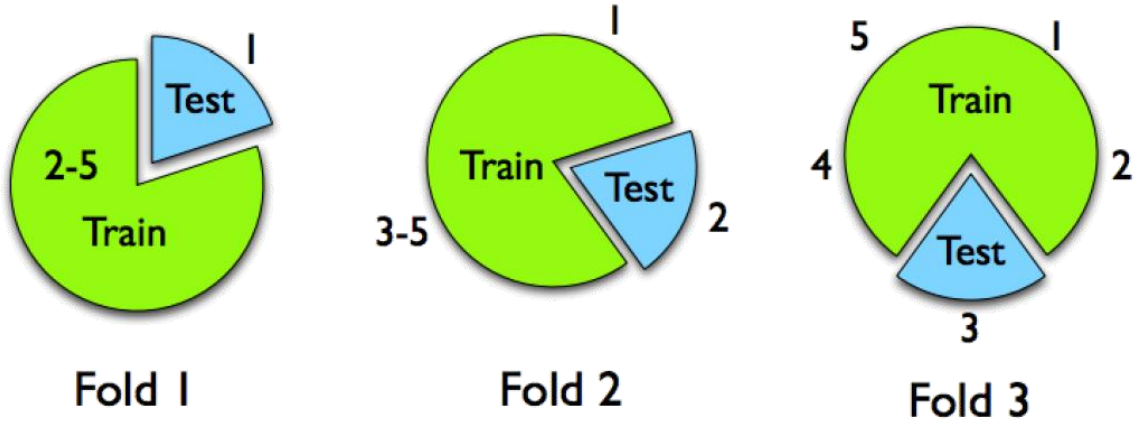
Stratified Sampling

- When randomly selecting training or validation sets, we may want to ensure that class proportions are maintained in each selected set
- This can be done via stratified sampling: first stratify instances by class, then randomly select instances from each class proportionally.



Model Selection

- Resubstitution
- K-fold cross-validation



- Leave-one-out
  - N-fold cross-validation

Cross-Validation: Example

- Suppose we have 100 instances, and we want to estimate accuracy with cross validation

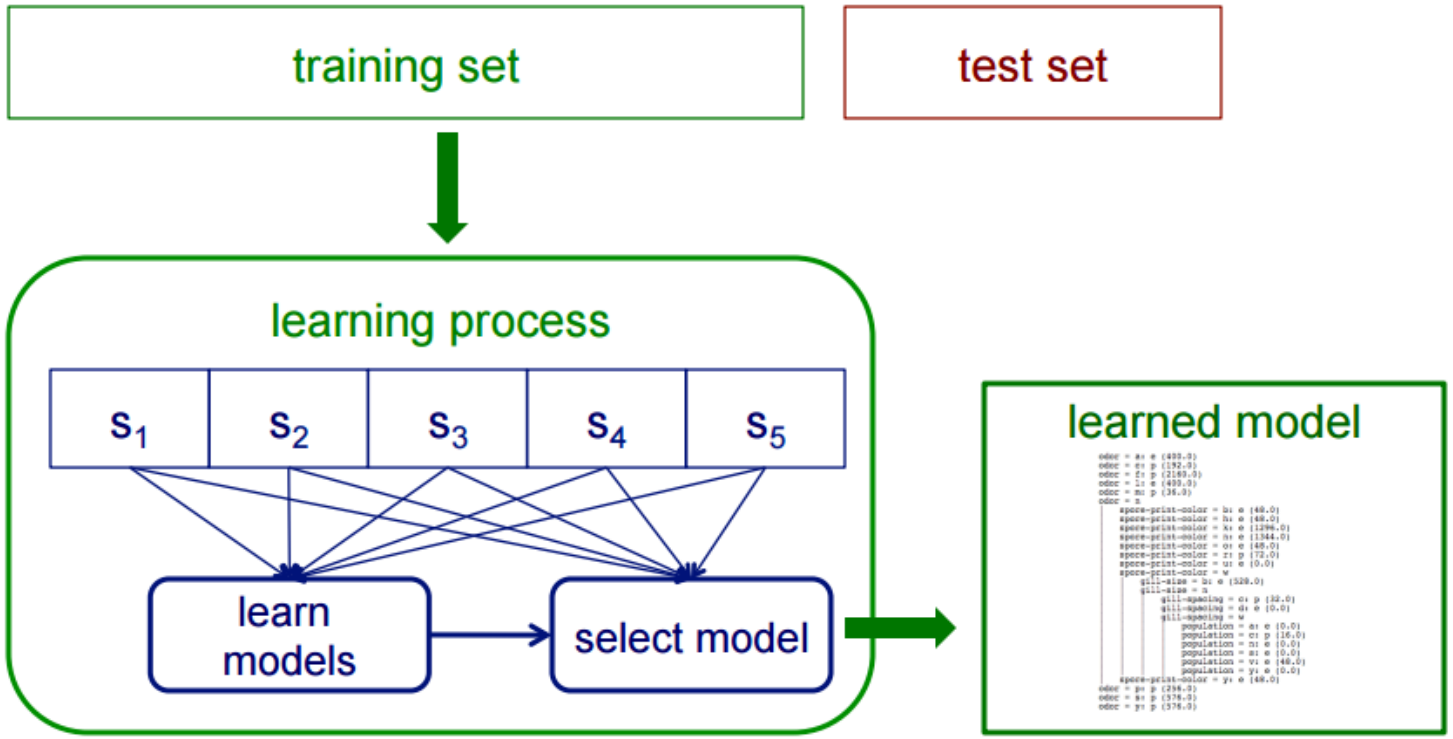
iteration	train on	test on	correct
1	S_2 S_3 S_4 S_5	S_1	11 / 20
2	S_1 S_3 S_4 S_5	S_2	17 / 20
3	S_1 S_2 S_4 S_5	S_3	16 / 20
4	S_1 S_2 S_3 S_5	S_4	13 / 20
5	S_1 S_2 S_3 S_4	S_5	16 / 20

Accuracy = 73/100 = 73%

Note: Whenever we use multiple training sets, as in CV and random resampling, we are evaluating a learning method as opposed to an individual learned model

Cross-Validation: Example

- Instead of a single validation set, we can use cross-validation within a training set to select a model (e.g. to choose the best k in k-NN)



Evaluation Measures

- Classification
  - How often we classify something right/wrong
- Regression
  - How close are we to what we're trying to predict
- Ranking/Search
  - How correct are the top-k results?
- Clustering
  - How well we describe our data (Not straightforward)

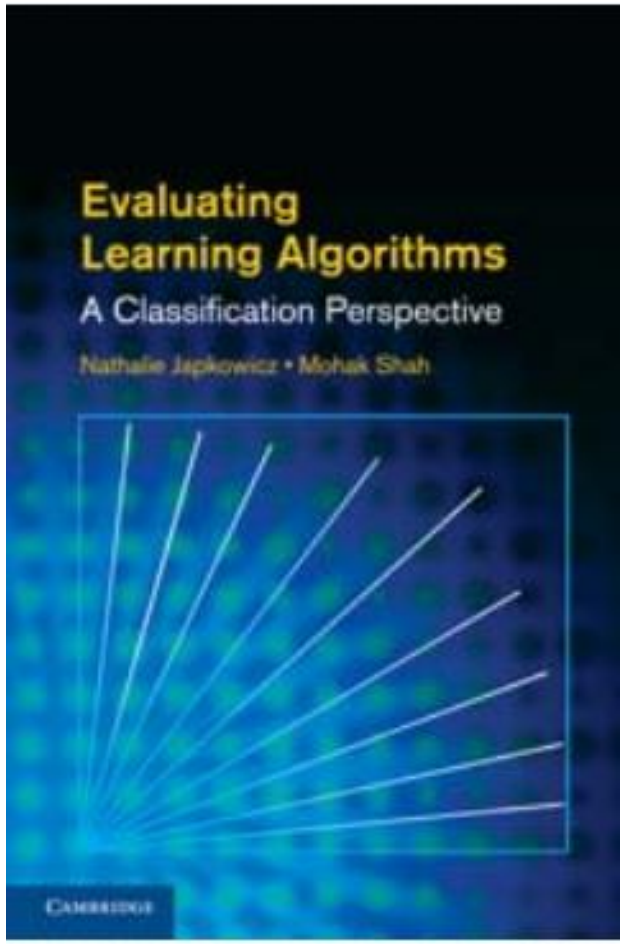
Is accuracy adequate?

- Accuracy may not be useful in cases where
  - There is a large class skew
    - Is 98% accuracy good if 97% of the instances are negative?
  - There are differential misclassification costs – say, getting a positive wrong costs more than getting a negative wrong
    - Consider a medical domain in which a false positive results in an extraneous test but a false negative results in a failure to treat a disease
  - We are most interested in a subset of high-confidence predictions

Classification Error: Beyond Accuracy

Evaluating Learning Algorithms: A Classification Perspective

Nathalie Japkowicz & Mohak Shah
Cambridge University Press, 2011



Good tutorial on the topic: http://www.icmla-conference.org/icmla11/PE_Tutorial.pdf

Classification Error: Beyond Accuracy

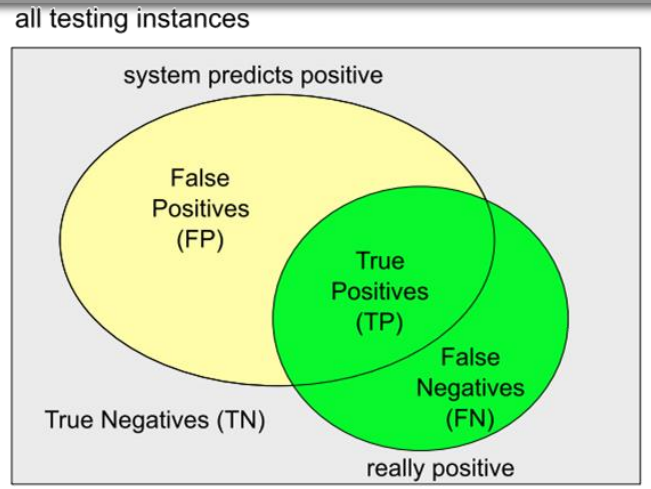
In 2-class problems:

		actual class: positive	actual class: negative
predicted class	positive	true positives (TP)	false positives (FP)
predicted class	negative	false negatives (FN)	true negatives (TN)

$$\text{accuracy} = \frac{TP + TN}{TP + FP + FN + TN}$$

Classification Performance Measures

Really positive?	Predict positive? Yes	No
Yes	TP	FN
No	FP	TN



- True Positive Rate also called "Sensitivity"
- "Specificity" = 1 – False Alarm

- Classification Error: $\frac{errors}{total} = \frac{FP+FN}{TP+TN+FP+FN}$
- Accuracy = 1-Error: $\frac{correct}{total} = \frac{TP+TN}{TP+TN+FP+FN}$

} meaningless if classes imbalanced

- False Alarm = False Positive rate = FP / (FP+TN)
- Miss = False Negative rate = FN / (TP+FN)
- Recall = True Positive rate = TP / (TP+FN)
- Precision = TP / (TP+FP)

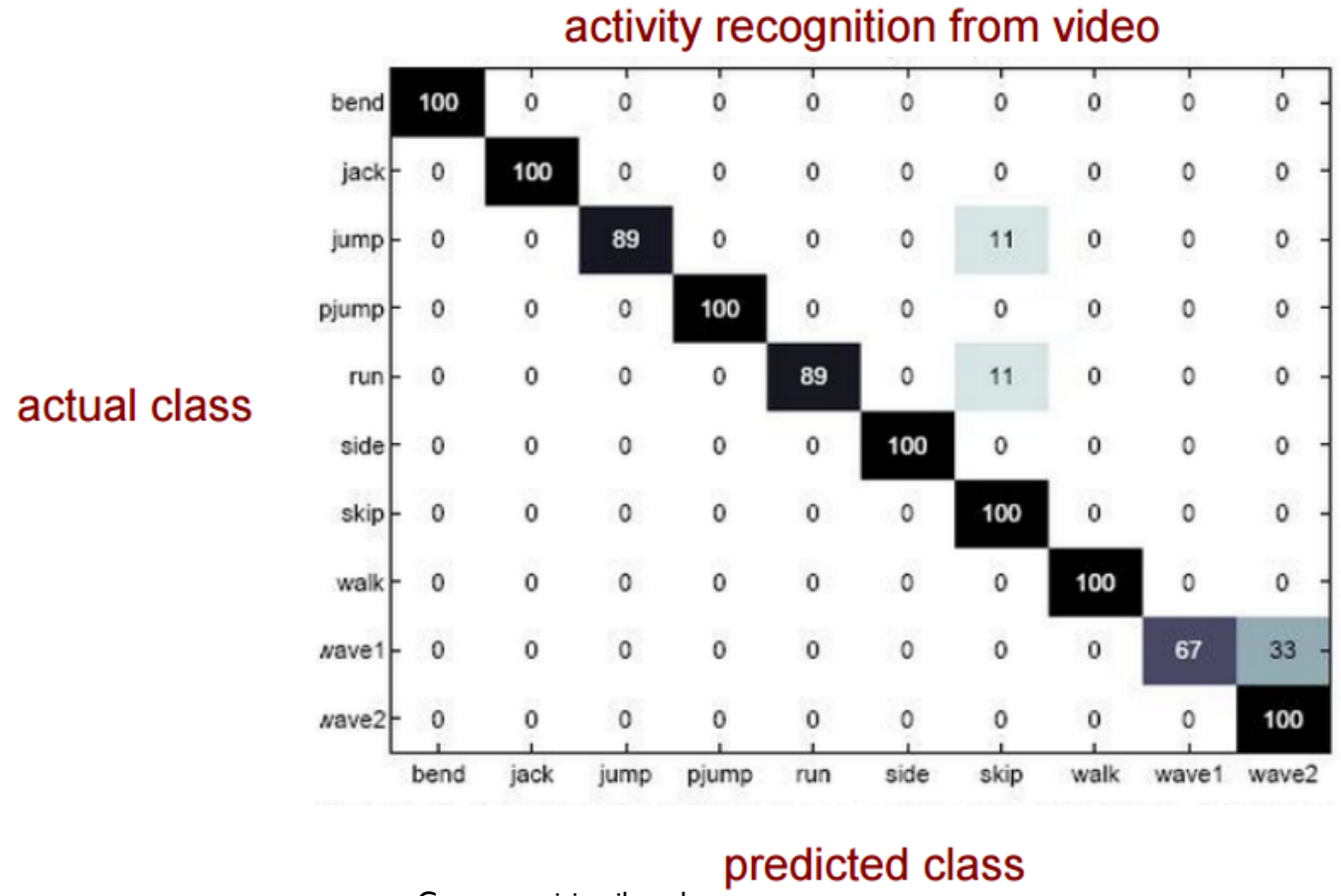
} always report in pairs, e.g.: Miss / FA or Recall / Prec.

- "Sensitivity" = Probability of a positive test given a patient has the disease
- "Specificity" = Probability of a negative test given a patient is well

Classification Error: Beyond Accuracy

For multi-class problems?

Confusion Matrix



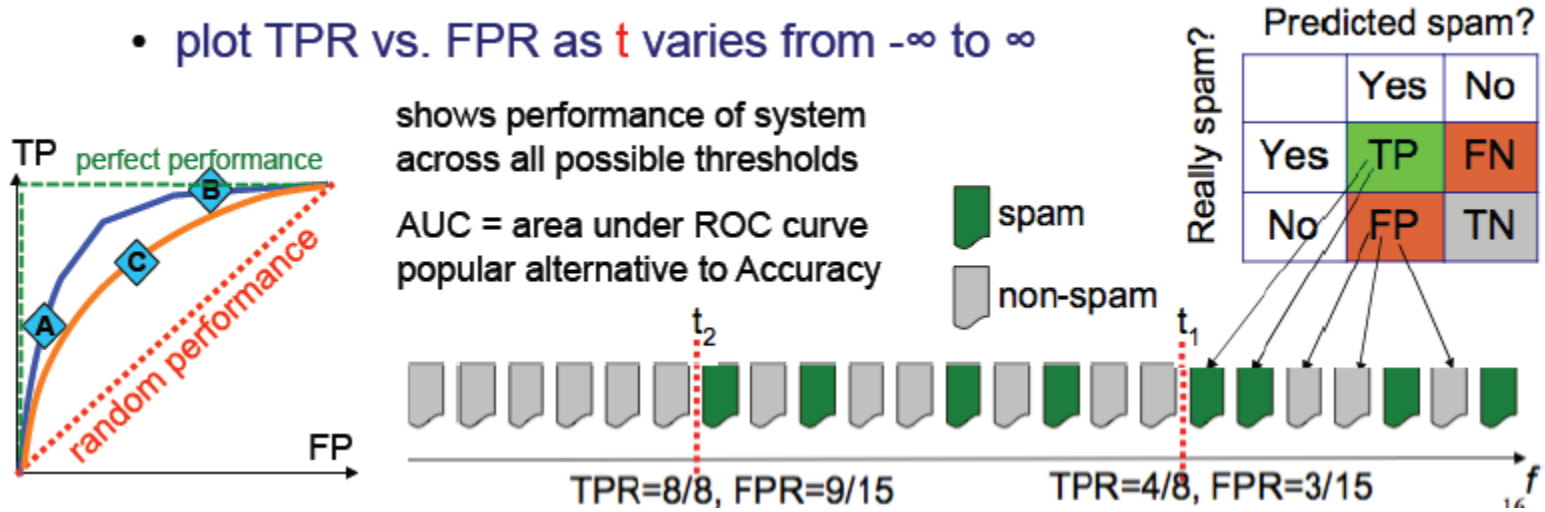
Courtesy: vision.jhu.edu

Utility and Cost

- Sometimes, there is a cost for each error
  - E.g. Earthquake prediction
    - False positive: Cost of preventive measures
    - False negative: Cost of recovery
- Detection Cost (Event detection)
  - Cost = $C_{FP} * FP + C_{FN} * FN$
- F-measure (Information Retrieval)
  - F1 = 2/(1/Recall + 1/Precision)

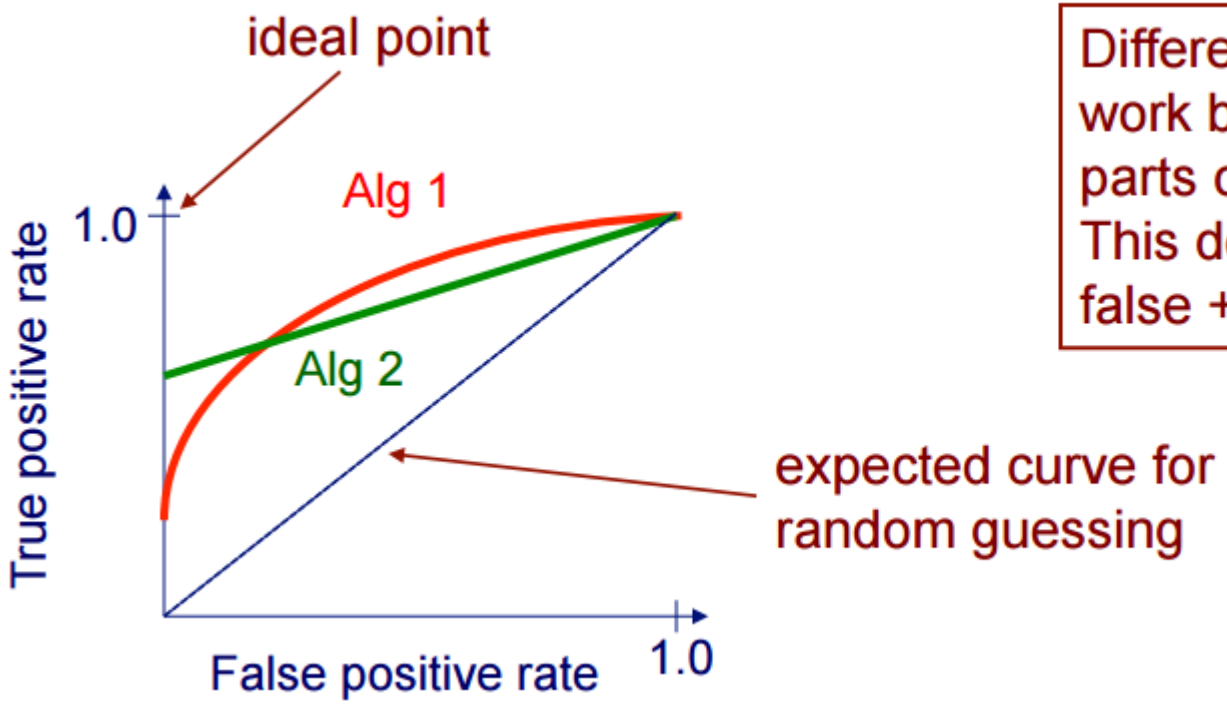
ROC Curves

- Many algorithms compute "confidence" f(x)
  - Threshold to get decision: spam if f(x) > t, non-spam if f(x) <= t
  - Threshold to determines error rates
- Receiver Operating Characteristic (ROC)



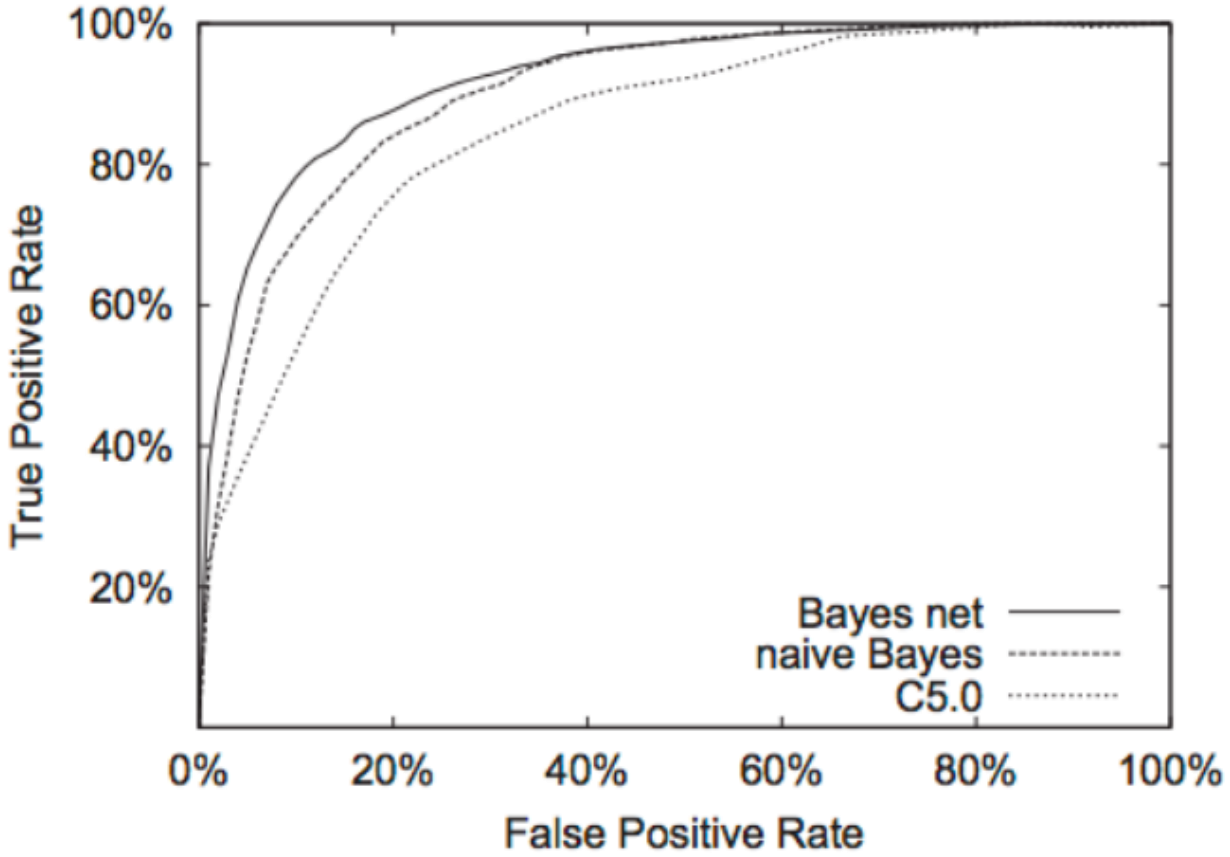
ROC Curves

- A Receiver Operating Characteristic (ROC) curve plots the TP-rate vs. the FP-rate as a threshold on the confidence of an instance being positive is varied



Different methods can work better in different parts of ROC space. This depends on cost of false + vs. false -

ROC Curve: Example



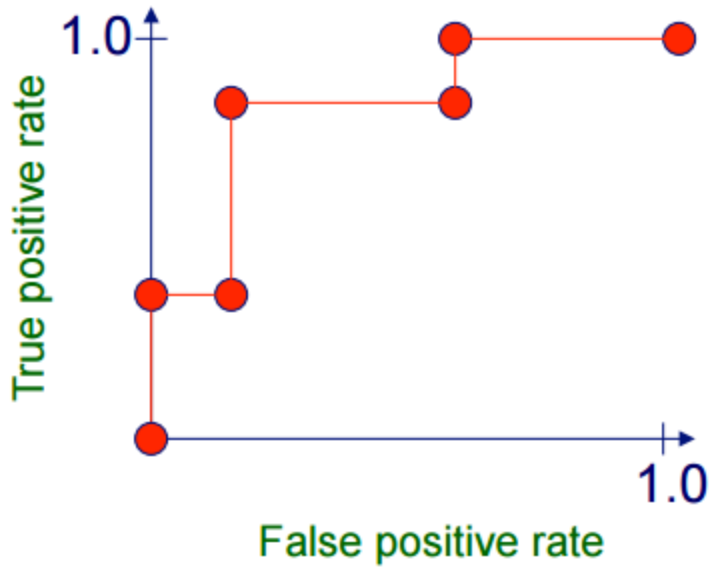
Courtesy: Bockhorst et al., Bioinformatics 2003

ROC Curve: Algorithm

- Sort test-set predictions according to confidence that each instance is positive
- Step through sorted list from high to low confidence
  - Locate a threshold between instances with opposite classes (keeping instances with the same confidence value on the same side of threshold)
  - Compute TPR, FPR for instances above threshold
  - Output (FPR, TPR) coordinate

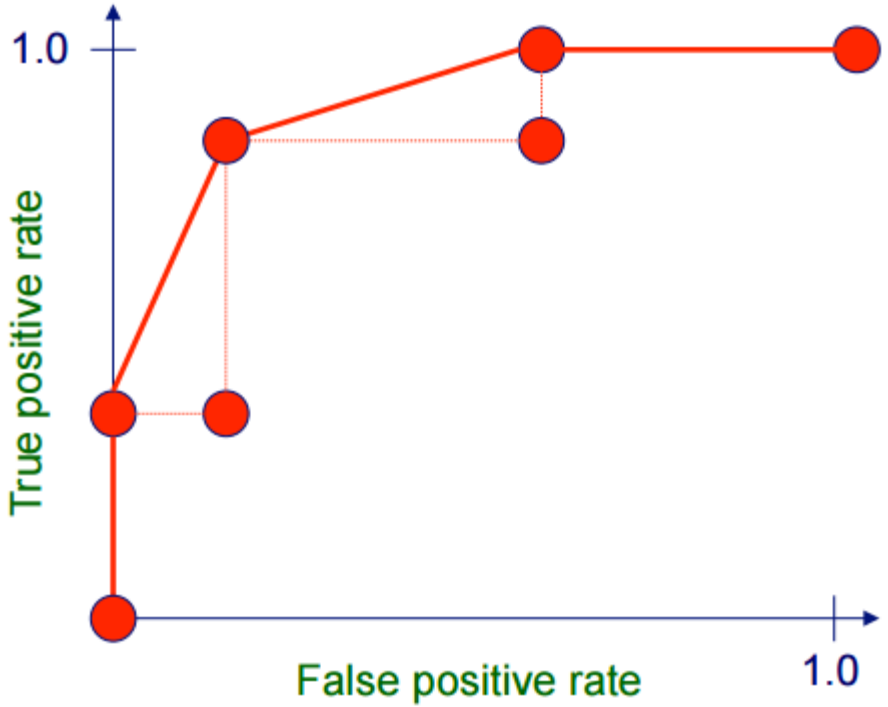
Plotting an ROC Curve

instance	confidence positive		correct class
Ex 9	.99		+
Ex 7	.98	TPR= 2/5, FPR= 0/5	+
Ex 1	.72	TPR= 2/5, FPR= 1/5	-
Ex 2	.70		+
Ex 6	.65	TPR= 4/5, FPR= 1/5	+
Ex 10	.51		-
Ex 3	.39	TPR= 4/5, FPR= 3/5	-
Ex 5	.24	TPR= 5/5, FPR= 3/5	+
Ex 4	.11		-
Ex 8	.01	TPR= 5/5, FPR= 5/5	-

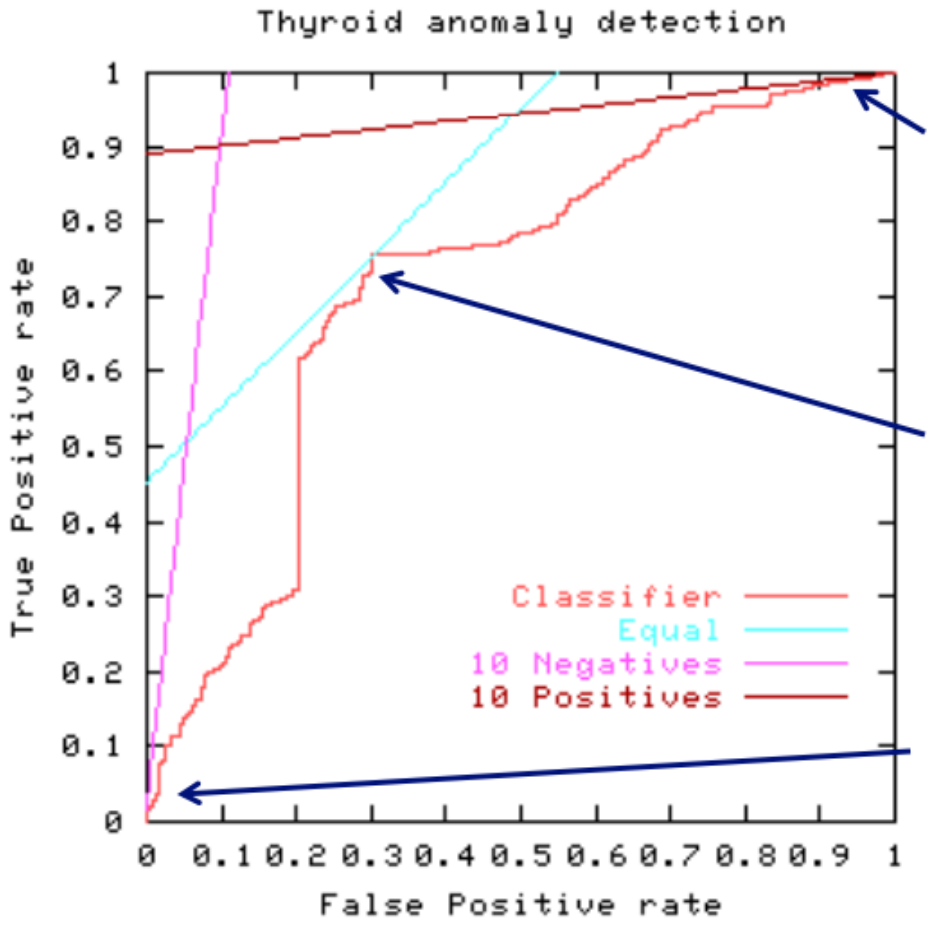


Plotting an ROC Curve

- Can interpolate between points to get convex hull



ROC Curves and Misclassification Costs

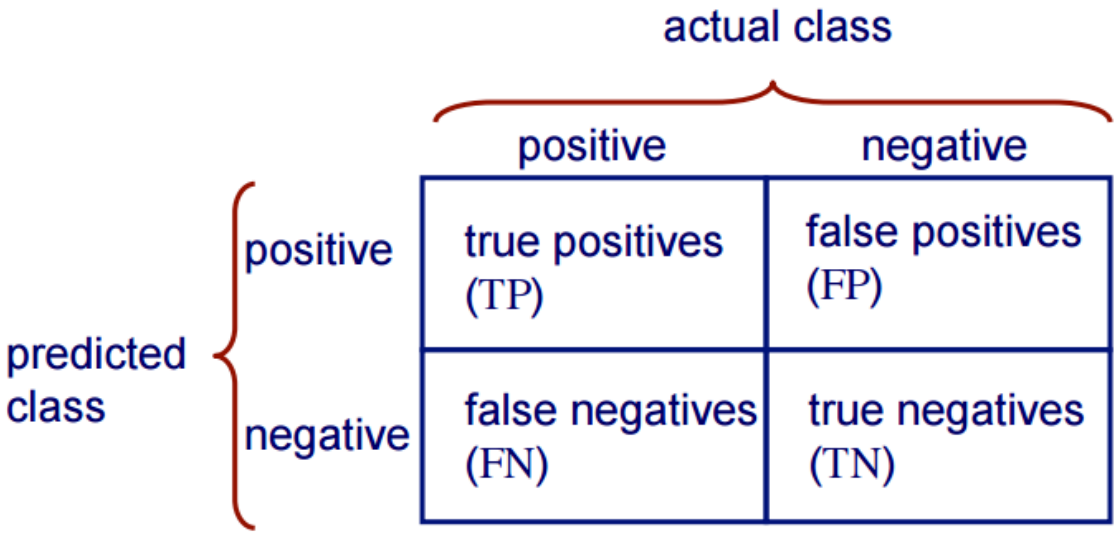


best operating point when
FN costs 10× FP

best operating point when
cost of misclassifying positives
and negatives is equal

best operating point when
FP costs 10× FN

Recall: Precision-Recall

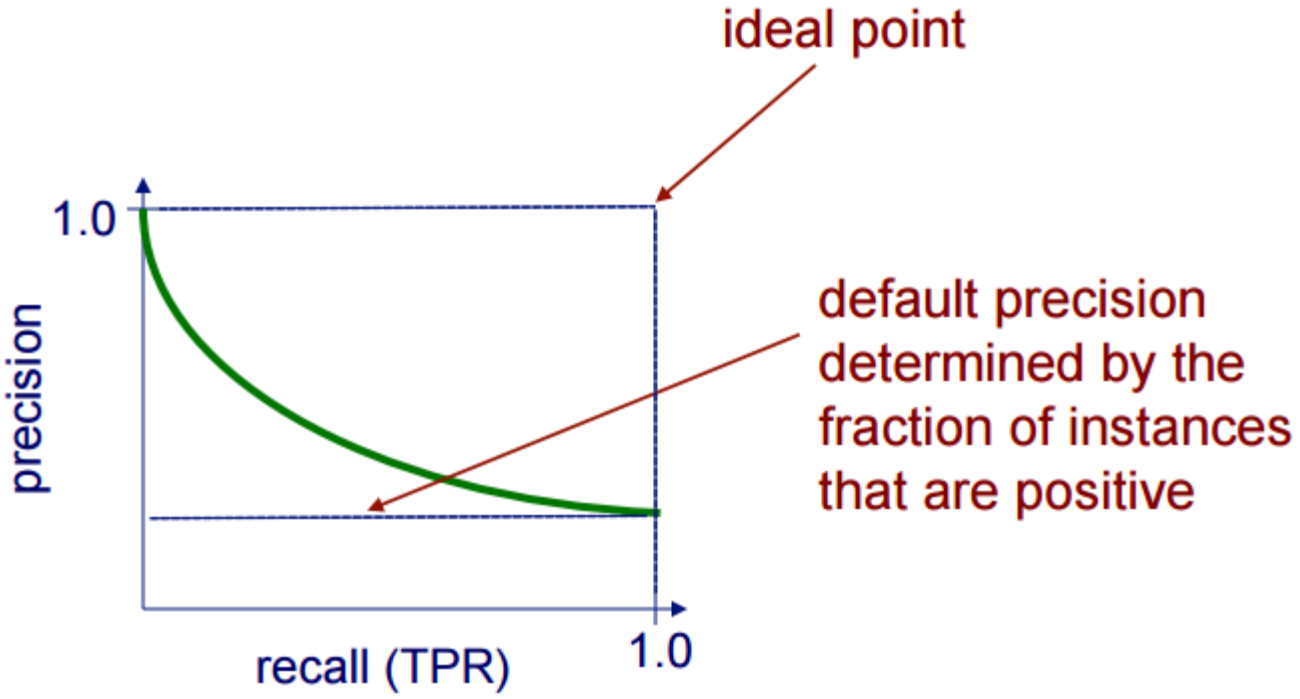


$$\text{recall (TP rate)} = \frac{\text{TP}}{\text{actual pos}} = \frac{\text{TP}}{\text{TP} + \text{FN}}$$

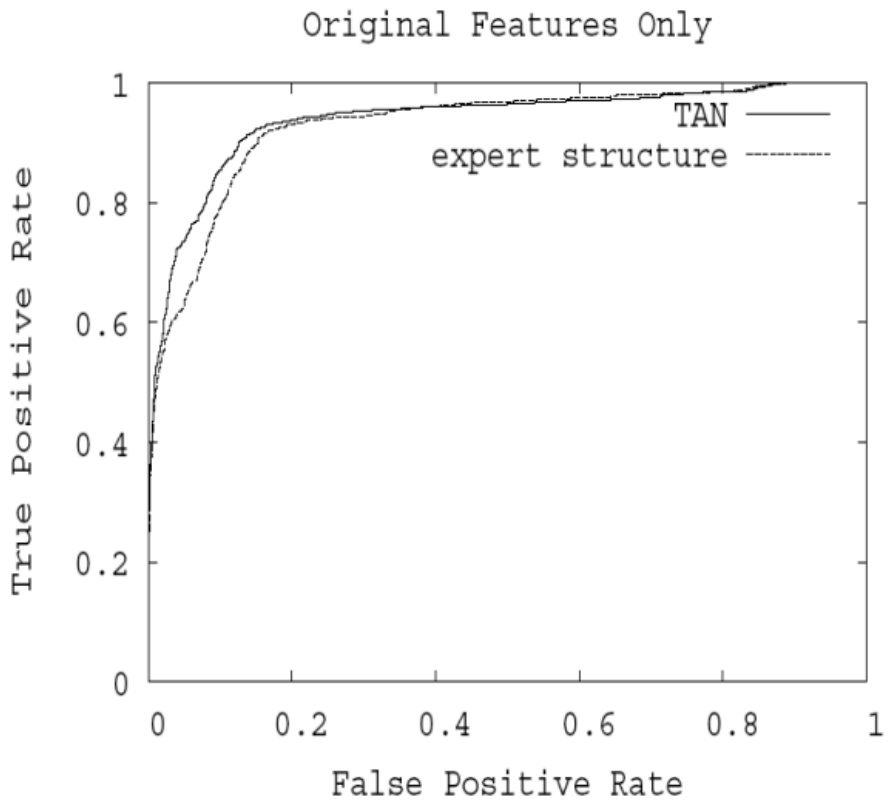
$$\text{precision} = \frac{\text{TP}}{\text{predicted pos}} = \frac{\text{TP}}{\text{TP} + \text{FP}}$$

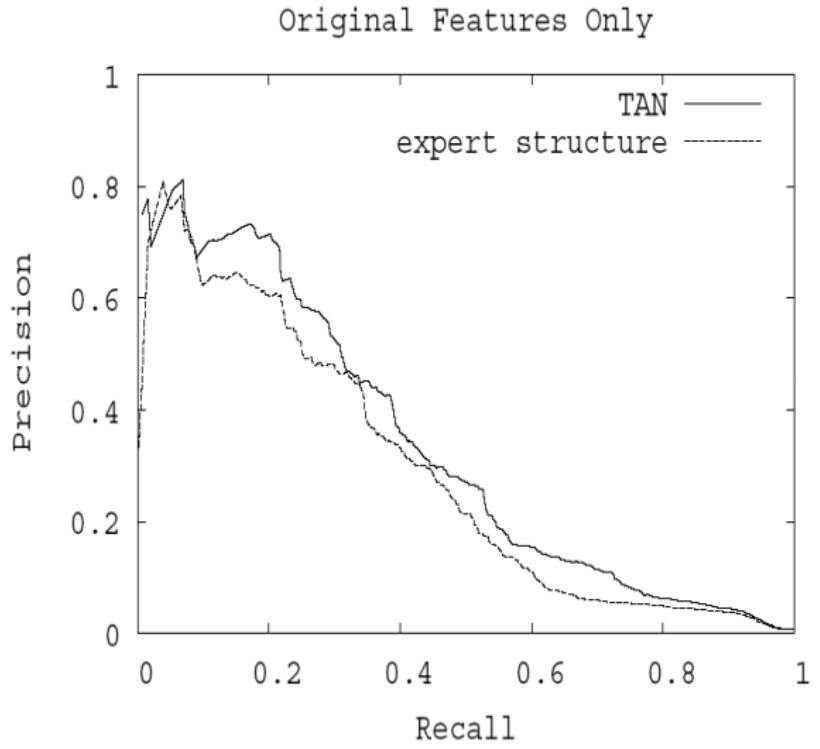
Precision/Recall Curves

- A precision/recall curve plots the precision vs. recall (TP-rate) as a threshold on the confidence of an instance being positive is varied



ROC + PR Curves: Example





Courtesy: Page, Univ of Wisconsion-Madison

ROC + PR Curves: Summary

- Both
  - Allow predictive performance to be assessed at various levels of confidence
  - Assume binary classification tasks
  - Sometimes summarized by calculating area under the curve
- ROC curves
  - Insensitive to changes in class distribution (ROC curve does not change if the proportion of positive and negative instances in the test set are varied)
  - Can identify optimal classification thresholds for tasks with differential misclassification costs
- Precision/Recall curves
  - Show the fraction of predictions that are false positives
  - Well-suited for tasks with lots of negative instances. Why?

Plotting ROC/PR Curves during Cross-Validation

- Approach 1
  - Make assumption that confidence values are comparable across folds
  - Pool predictions from all test sets
  - Plot the curve from the pooled predictions
- Approach 2 (for ROC curves)
  - Plot individual curves for all test sets
  - View each curve as a function
  - Plot the average curve for this set of functions

Other Performance Measures

- Kullback-Leibler Divergence: $D_{\mathrm{KL}}(P\|Q) = \sum_i P(i)\ \log \frac{P(i)}{Q(i)}$
- Gini Statistic:
  - 2 * AUC – 1
- F-score: Harmonic mean of precision and recall
  - (2 * precision * recall)/(precision+recall)
- Akaike Information Criterion:
  - AIC = 2k – 2 ln (L), where L is the max value of the likelihood function for the model, and k is the number of model parameters

Pitfalls

- Is my held-aside test data really representative of going out to collect new data?
  - Even if your methodology is fine, someone may have collected features for positive examples differently than for negatives – should be randomized
  - Example: samples from cancer processed by different people or on different days than samples for normal controls

Pitfalls

- Did I repeat my entire data processing procedure on every fold of cross-validation, using only the training data for that fold?
  - On each fold of cross-validation, did I ever access in any way the label of a test case?
  - Any preprocessing done over entire data set (feature selection, parameter tuning, threshold selection) must not use labels from test set

Pitfalls

- Have I modified my algorithm so many times, or tried so many approaches, on this same data set that I (the human) am overfitting it?
  - Have I continually modified my preprocessing or learning algorithm until I got some improvement on this data set?
  - If so, I really need to get some additional data now to at least test on

Confidence Intervals on Error

- Given the observed error (accuracy) of a model over a limited sample of data, how well does this error characterize its accuracy over additional instances?
- Suppose we have
  - a learned model h
  - a test set S containing n instances drawn independently of one another and independent of h
  - h makes r errors over the n instances
- Our best estimate of the error of h is: $error_S(h) = \frac{r}{n}$

Confidence Intervals on Error

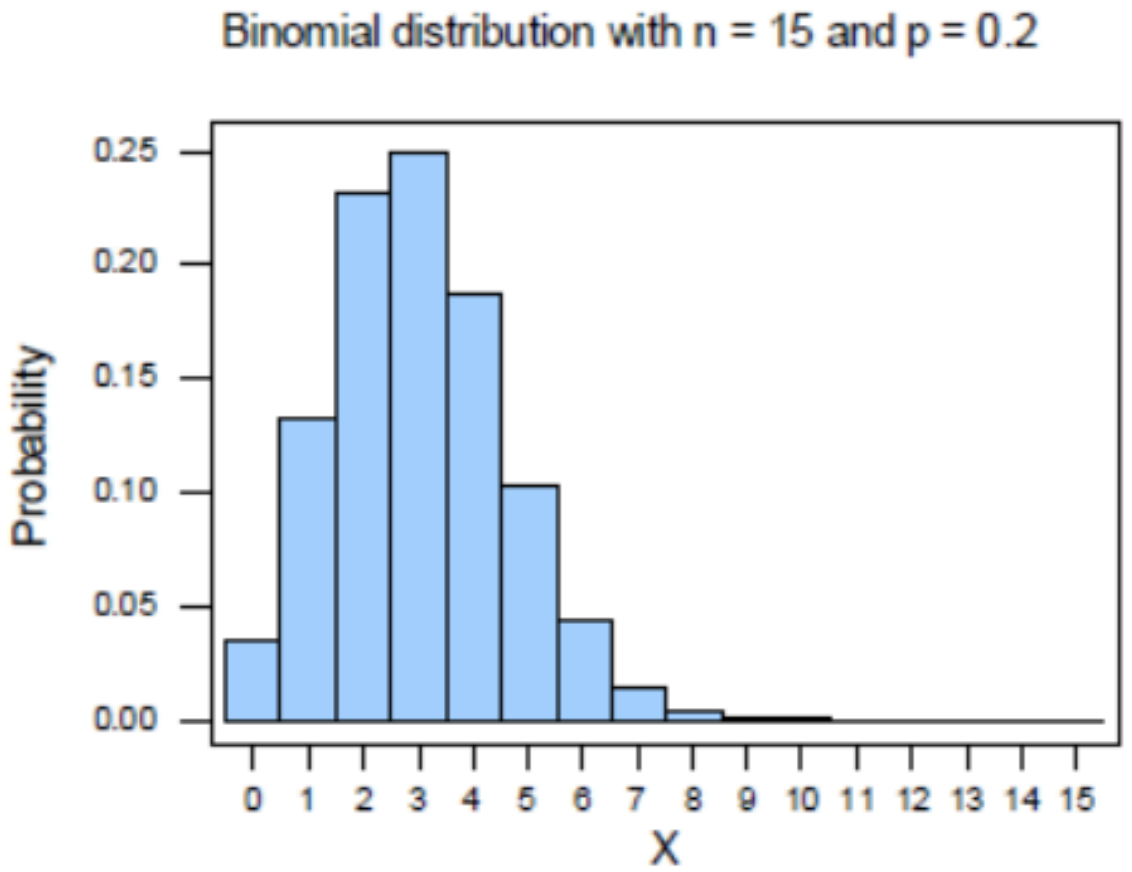
- With approximately N% probability, the true error lies in the interval

$$error_S(h) \pm z_N \sqrt{\frac{error_S(h)(1-error_S(h))}{n}}$$

- where z_N is a constant that depends on N (e.g. for 95% confidence, $z_N = 1.96$)
- How did we get this?

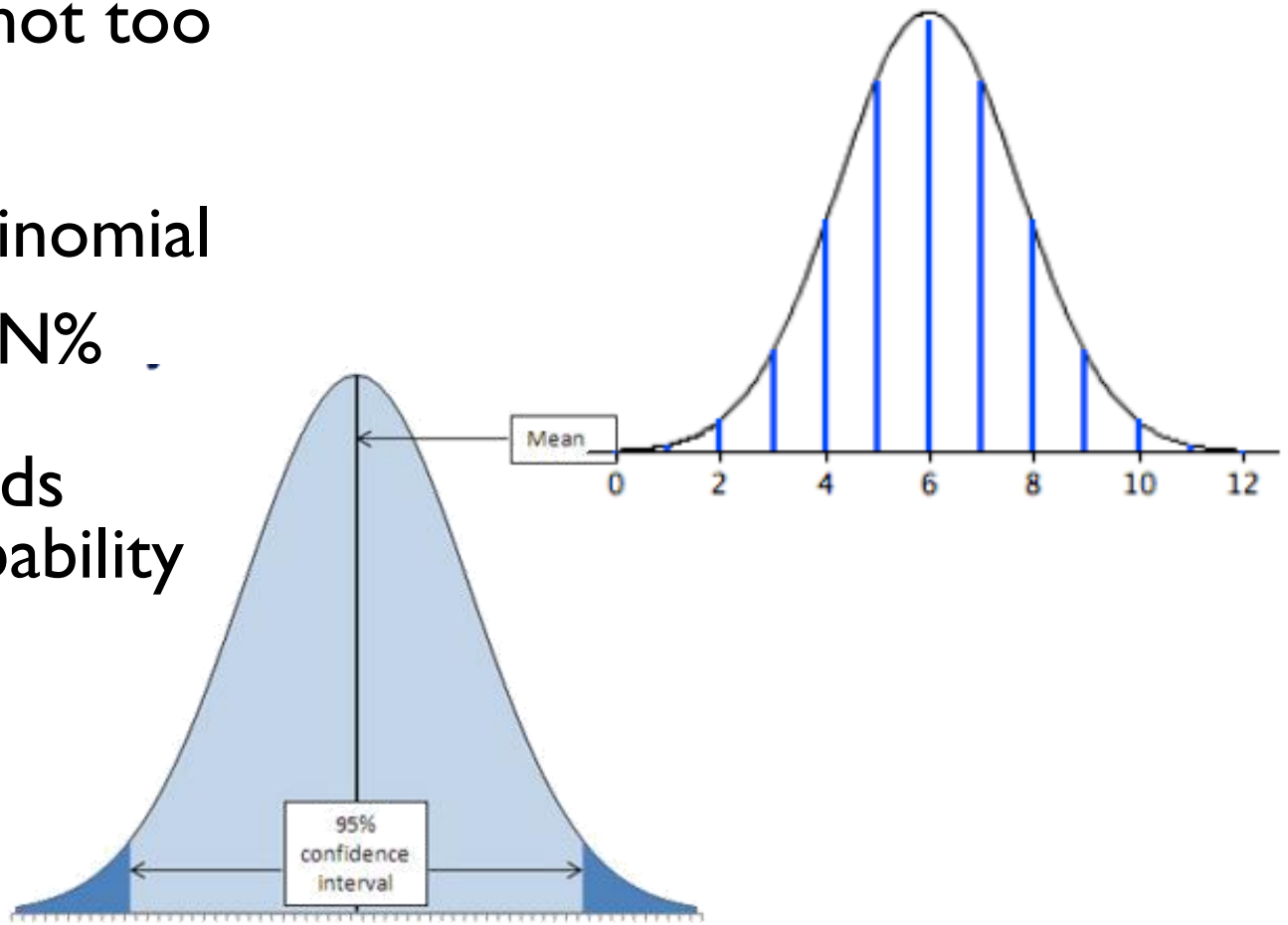
Confidence Intervals on Error

- Our estimate of the error follows a binomial distribution given by n and p (the true error rate over the data distribution)
- Simplest (and most common) way to determine a binomial confidence interval is to use the normal approximation



Confidence Intervals on Error

- When n ≥ 30, and p is not too extreme, the normal distribution is a good approximation to the binomial
- We can determine the N% confidence interval by determining what bounds contain N% of the probability mass under the normal



Empirical Confidence Bounds

- Bootstrapping: Given n examples in data set, randomly, uniformly, independently (with replacement) draw n examples – bootstrap sample
- Repeat 1000 (or 10,000) times:
  - Draw bootstrap sample
  - Repeat entire cross-validation process
- Assuming 95% confidence interval
  - Lower (upper) bound is result such that 2.5% of runs yield lower (higher)

Comparing Learning Systems

- How can we determine if one learning system provides better performance than another for a particular task? across a set of tasks / data sets?

Motivating Example

	Accuracies on test sets				
System 1:	80%	50	75	…	99
System 2:	79	49	74	…	98
δ :	+1	+1	+1	…	+1

- Mean accuracy for System 1 is better, but the standard deviations for the two clearly overlap
- Notice that System 1 is always better than System 2

Comparing Learning Systems

- consider δ's as observed values of a set of i.i.d. random variables
- Null hypothesis: the 2 learning systems have the same accuracy
- Alternative hypothesis: one of the systems is more accurate than the other
- Hypothesis test:
  - Use paired t-test to determine probability p that mean of δ's would arise from null hypothesis
  - If p is sufficiently small (typically < 0.05) then reject the null hypothesis

Comparing Systems using a Paired t-Test

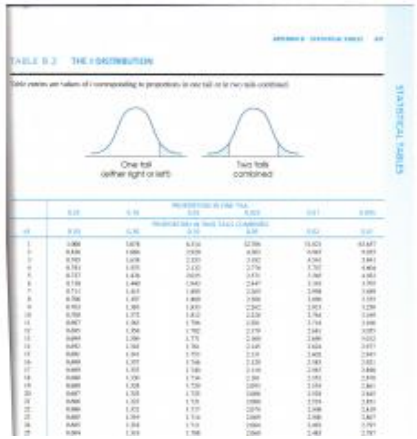
- Calculate the sample mean

$$\bar{\delta} = \frac{1}{n}\sum_{i=1}^{n}\delta_i$$

- Calculate the t-statistic

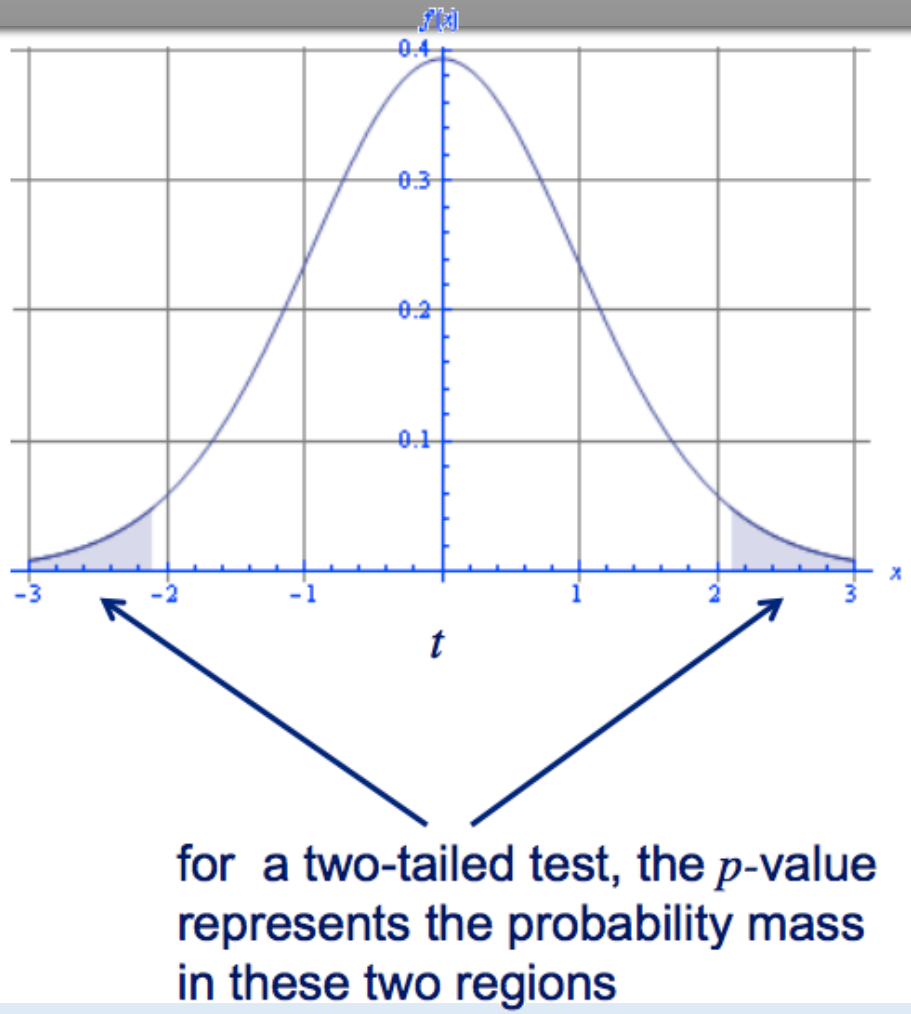
$$t = \frac{\bar{\delta}}{\sqrt{\frac{1}{n(n-1)}\sum_{i=1}^{n}(\delta_i - \bar{\delta})^2}}$$

- Determine the corresponding p-value, by looking up t in a table of values for the Student's t-distribution with n-1 degrees of freedom



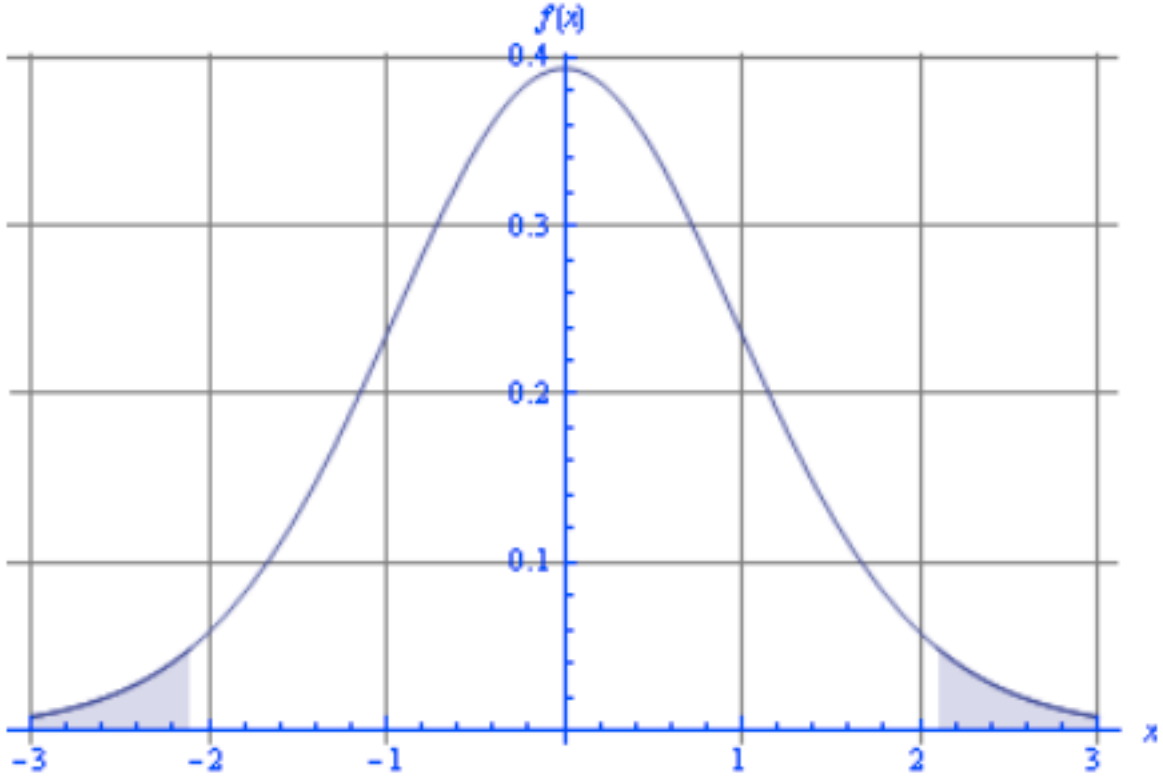
Comparing Systems using a Paired t-Test

- The null distribution of our t statistic looks like this
- The p-value indicates how far out in a tail our t statistic is
- If the p-value is sufficiently small, we reject the null hypothesis, since it is unlikely we'd get such a t by chance



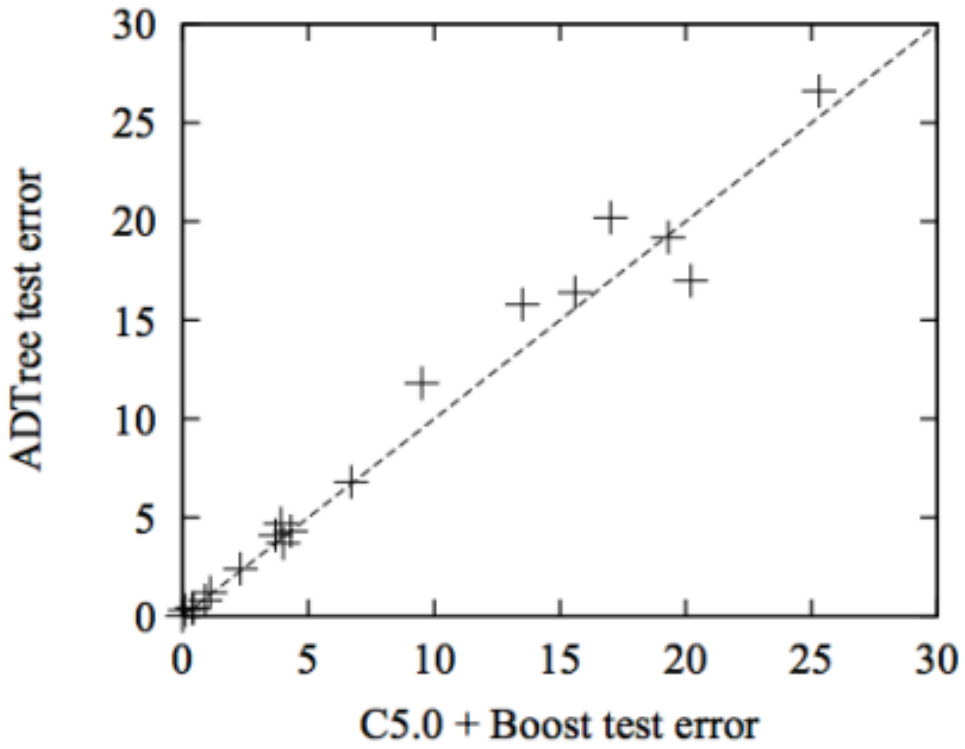
Why do we use a two-tailed test?

- A two-tailed test asks the question: is the accuracy of the two systems different
- A one-tailed test asks the question: is system A better than system B
- A priori, we don't know which learning system will be more accurate (if there is a difference) – we want to allow that either one might be



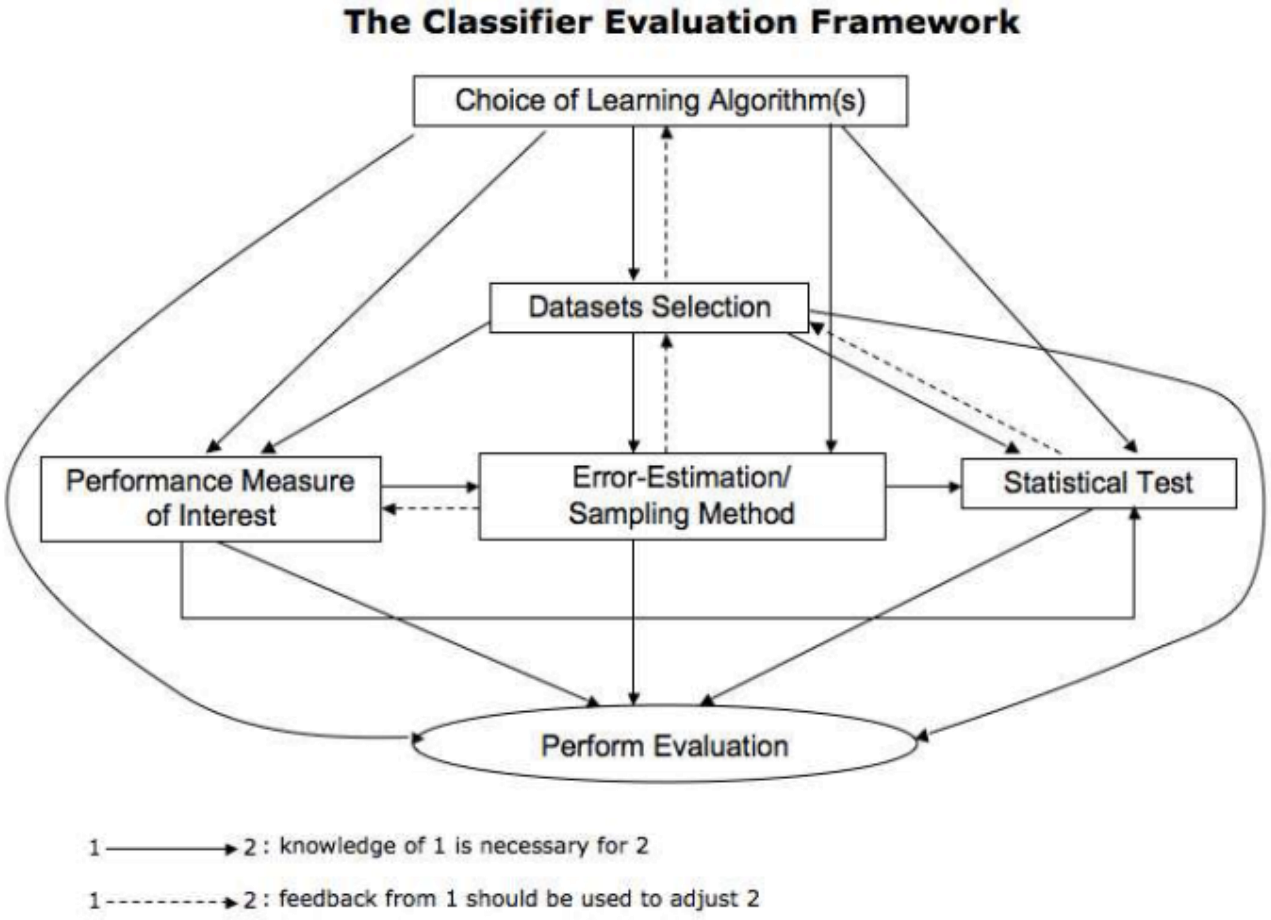
Pairwise Method Comparison

- We can compare the performance of two methods A and B by plotting (A performance, B performance) across numerous data sets



Courtesy: Freund & Mason, ICML 1999

Classifier Evaluation



Summary

- Rigorous statistical evaluation is extremely important in experimental computer science in general and machine learning in particular
- How good is a learned hypothesis?
- How close is the estimated performance to the true performance?
- Is one hypothesis better than another?
- Is one learning algorithm better than another on a particular learning task?

References

- Key References
  - Chapter 4, Introduction to Data Mining by Tan, Steinbach and Kumar
    - http://www-users.cs.umn.edu/~kumar/dmbook/index.php
  - Chapter 5 ('Evaluating Hypotheses'), Machine Learning by Tom Mitchell
    - http://www.cs.cmu.edu/~tom/mlbook.html
- Other Recommended References
  - http://www.icmla-conference.org/icmla11/PE_Tutorial.pdf (Tutorial on Performance Evaluation of Classifiers)

Homework

- Go through recommended reading materials
- Work on HW1!