

CS6510  
Applied Machine Learning

# Classifier Evaluation

5 Aug 2017

Vineeth N Balasubramanian



# ML Problems: Recall

	<i>Supervised Learning</i>	<i>Unsupervised Learning</i>
<i>Discrete</i>	classification or categorization	clustering
<i>Continuous</i>	regression	dimensionality reduction

# Classification Methods

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

How to evaluate?

# Training vs Generalization Error

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

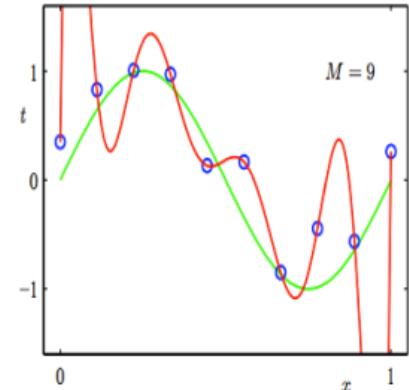
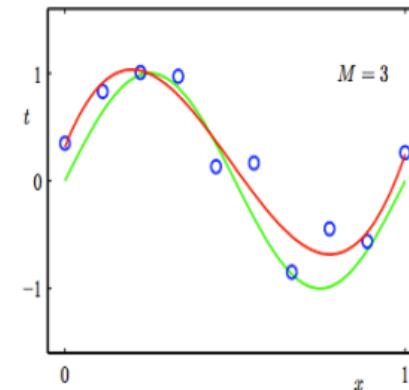
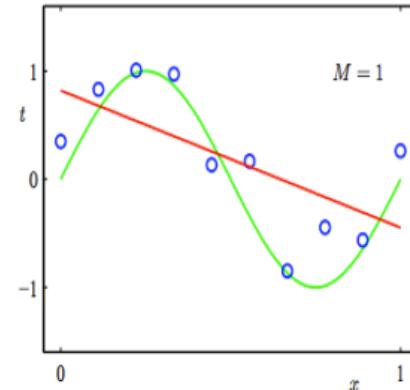
How to compute generalization error?

$$E_{train} = \frac{1}{n} \sum_{i=1}^n \underbrace{\text{error}(f_D(\mathbf{x}_i), y_i)}_{\substack{\text{value we} \\ \text{predicted}}} - \underbrace{y_i}_{\substack{\text{true} \\ \text{value}}} \quad \begin{array}{l} \text{same? different by how much?} \\ \text{training examples} \end{array}$$

$$E_{gen} = \int \underbrace{\text{error}(f_D(\mathbf{x}), y)}_{\substack{\text{over all} \\ \text{possible } x,y}} \underbrace{p(y, \mathbf{x})}_{\substack{\text{error as before} \\ \text{how often we expect} \\ \text{to see such } x \text{ and } y}} d\mathbf{x}$$

# Underfitting and Overfitting

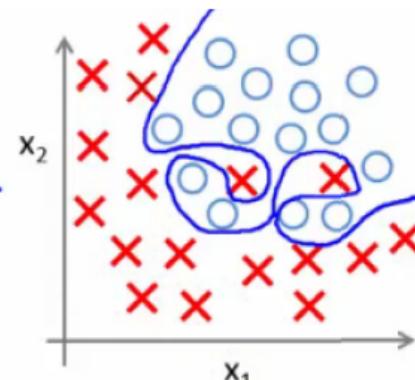
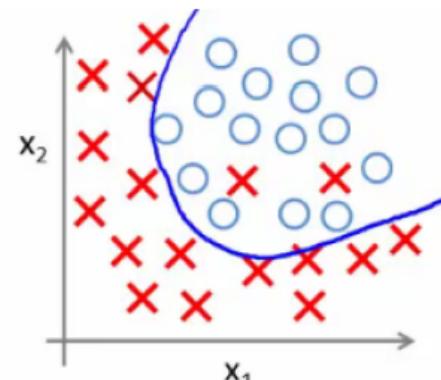
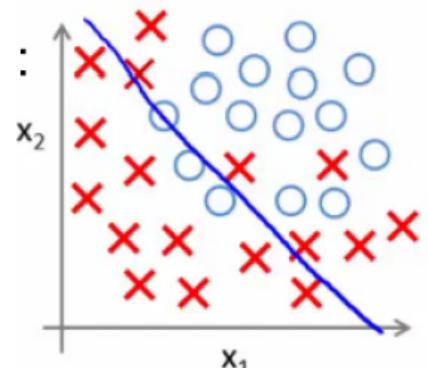
## Regression



predictor too inflexible:  
cannot capture pattern

predictor too flexible:  
fits noise in the data

## Classification



# Estimating Generalization Error

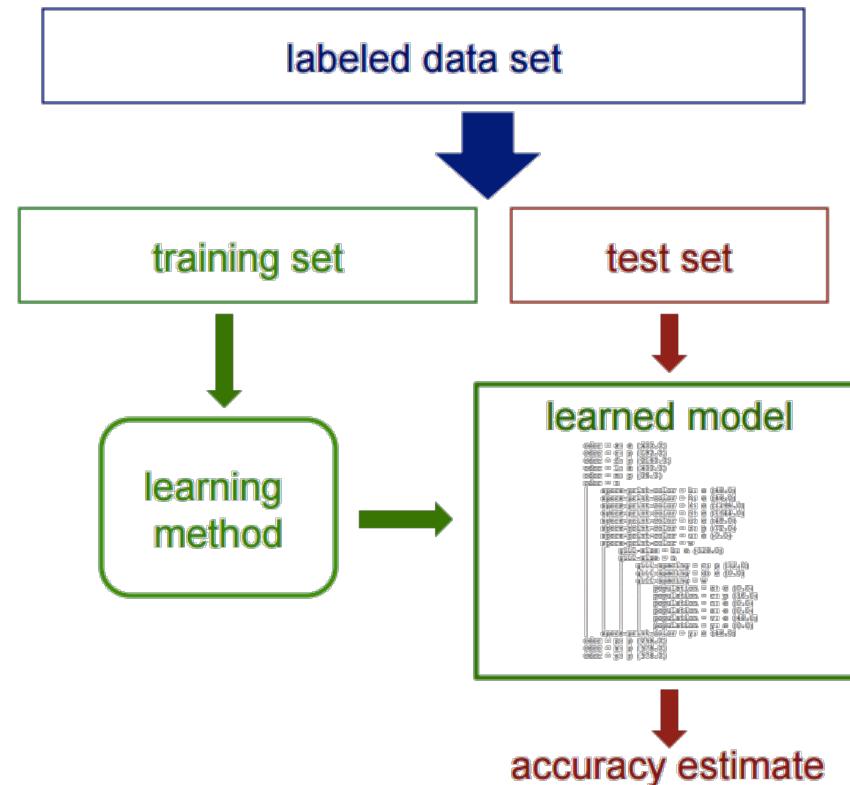
- Testing Error
  - Set aside part of training data (testing set)
  - Learn a predictor without using any of this test data
  - Predict values for testing set, compute error
  - This is an estimate of generalization error

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

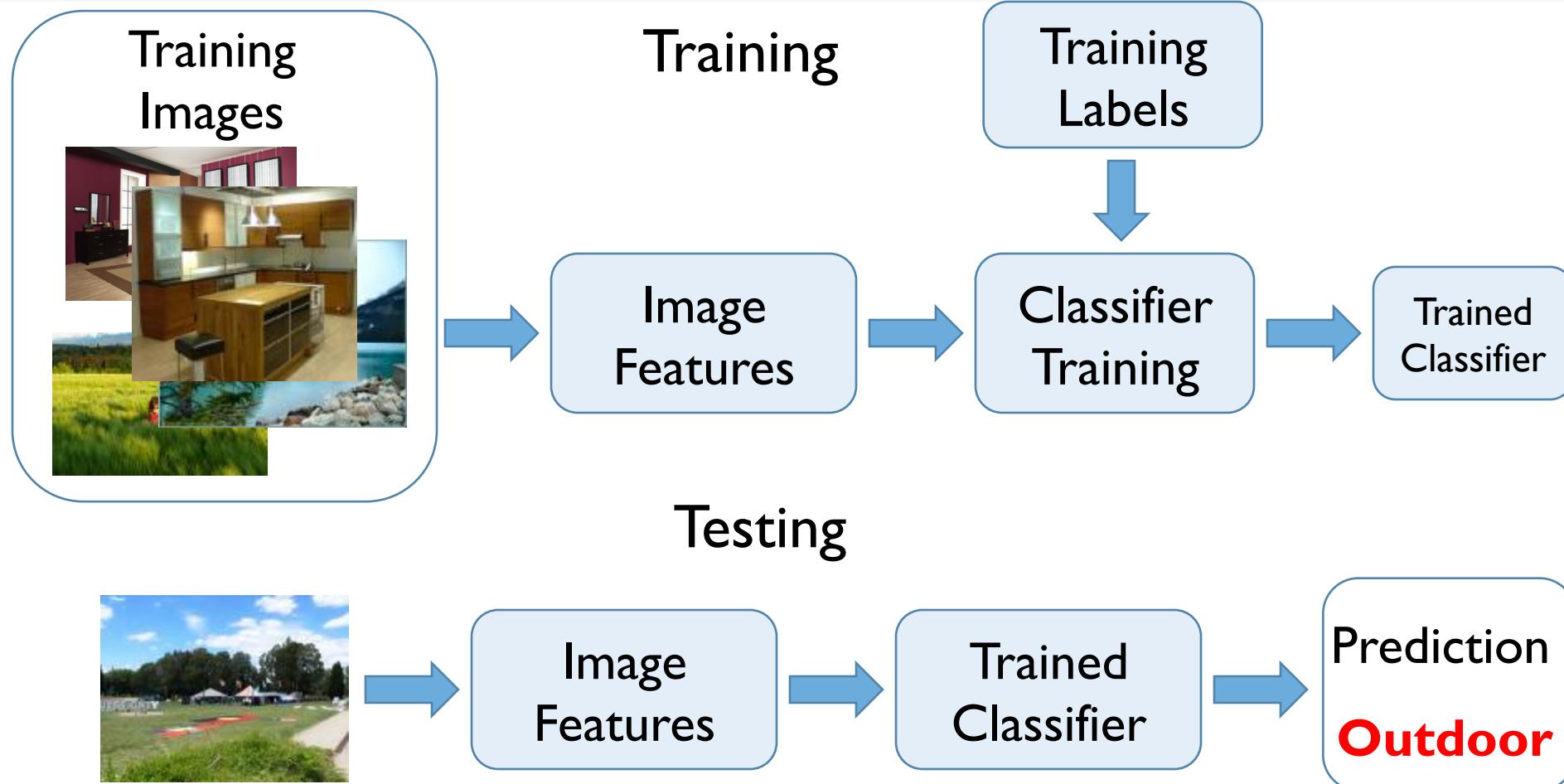
over testing set

# Estimating Generalization Error

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



# Example: Image Classification



Source: Derek Hoiem

# Training, Validation, Test Sets

## Training set

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

## Validation set

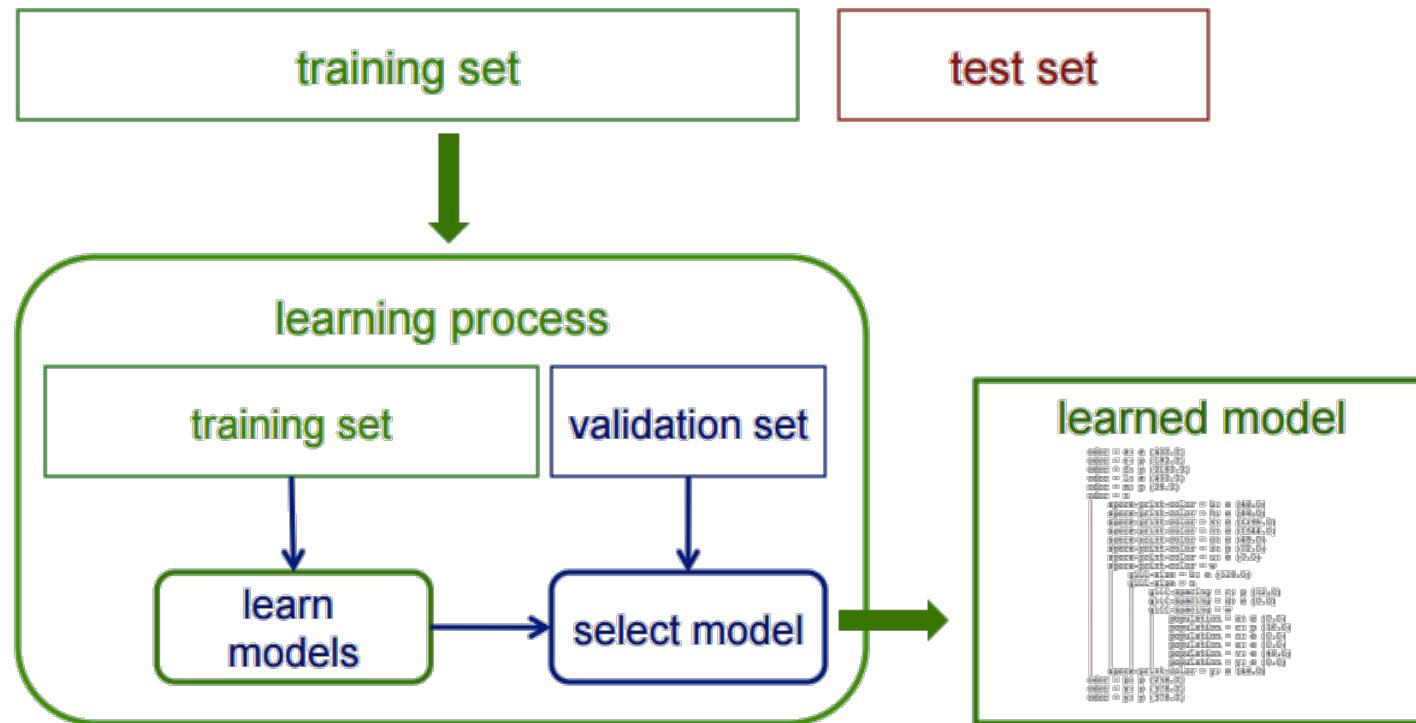
- Pick best-performing algorithm (NB vs DT vs..)
- Fine-tune parameters (Tree depth, k in kNN, c in SVM)

## Testing set

- Run multiple trials and average

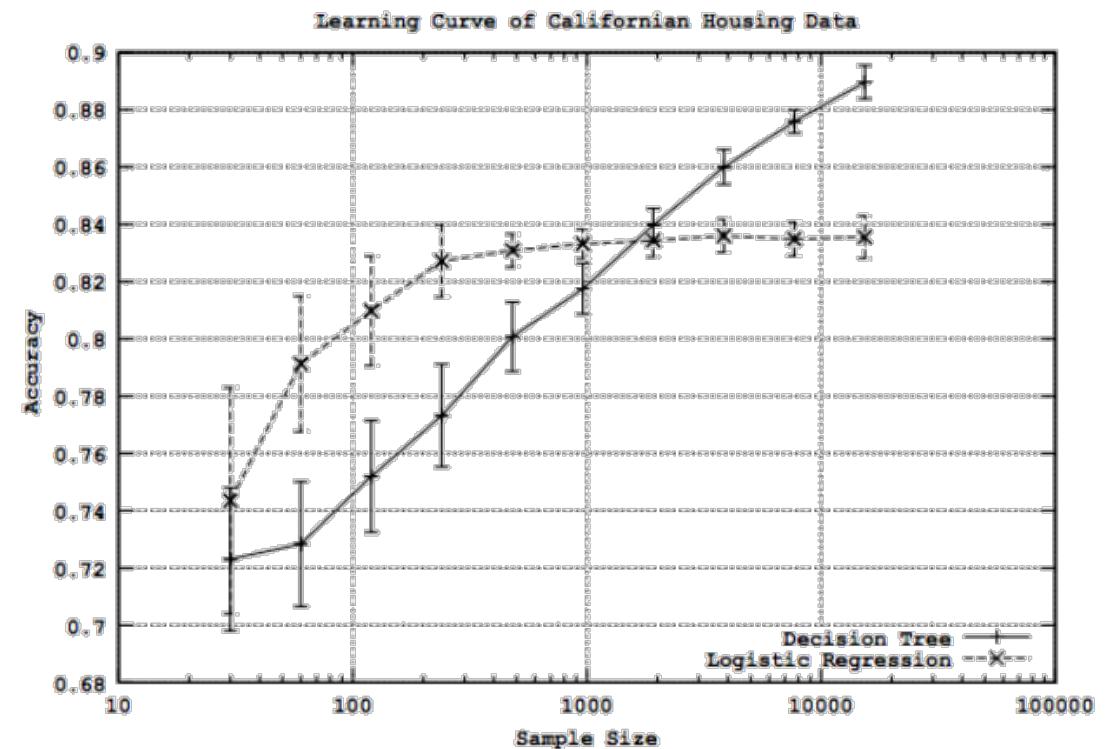
# Use of Validation Sets

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



# Choosing Training, Validation, Test Sets

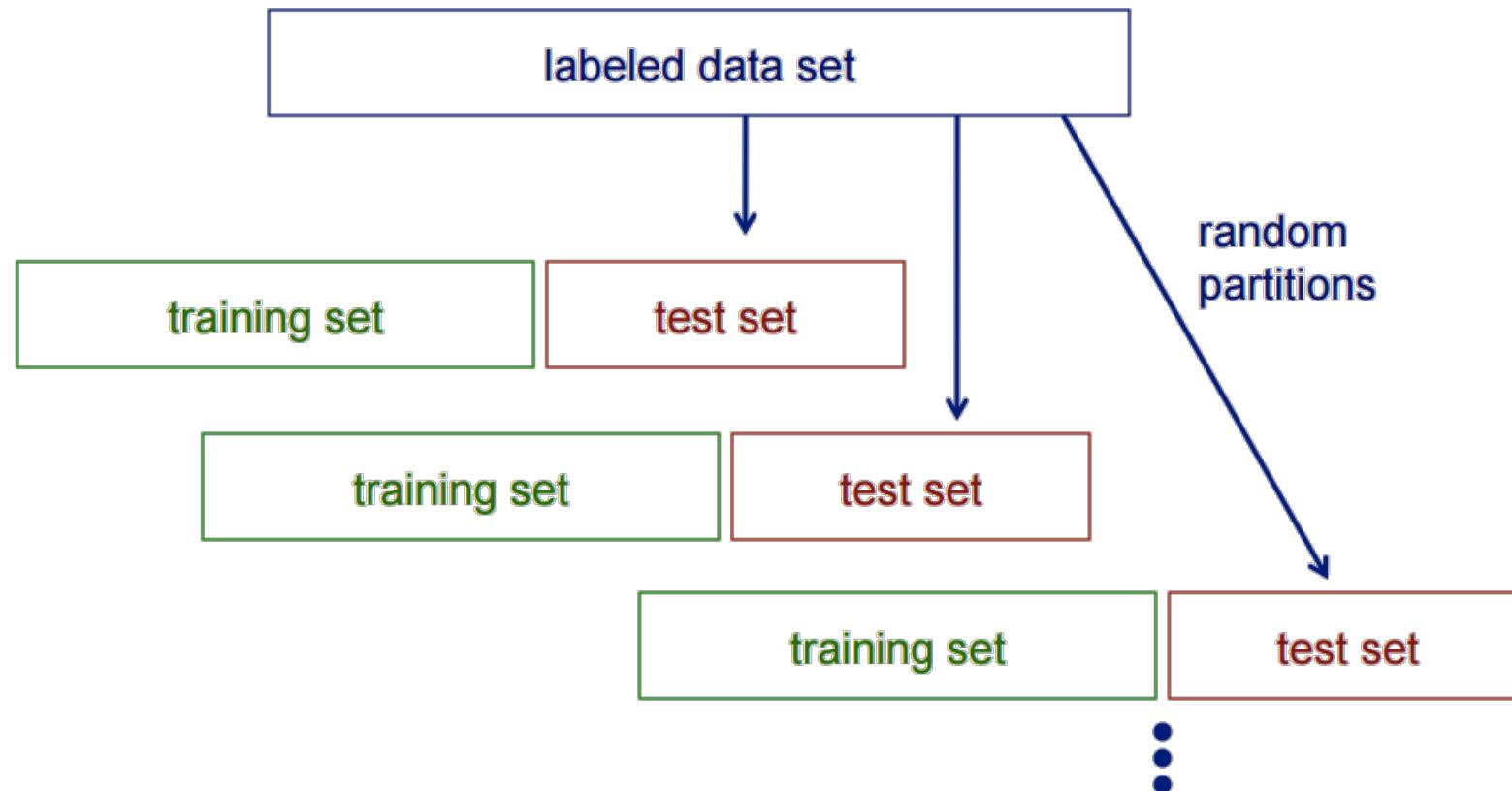
- Split **randomly** to avoid bias
- Large test set -> estimate future error as accurately as possible (vs)  
Large training set => better estimates
- How large should a training set be?
  - Study accuracy/error (vs) training set size



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

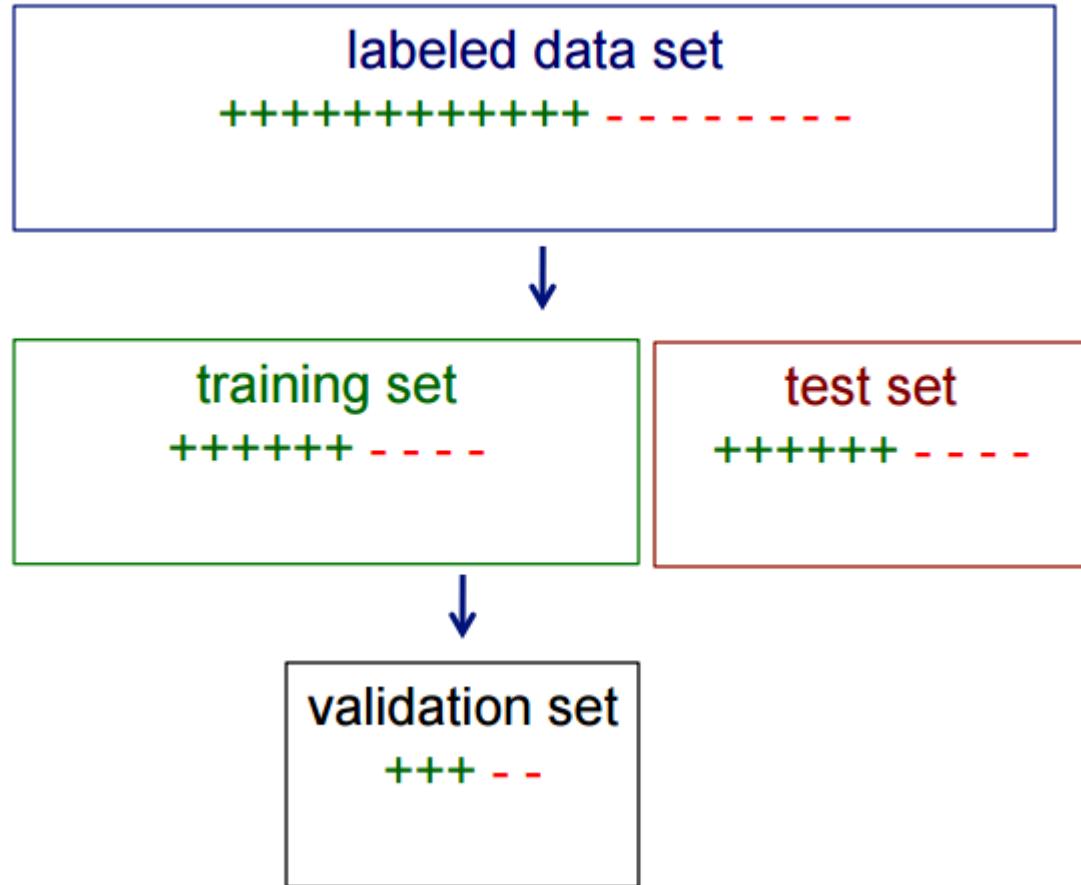
# Random Resampling

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



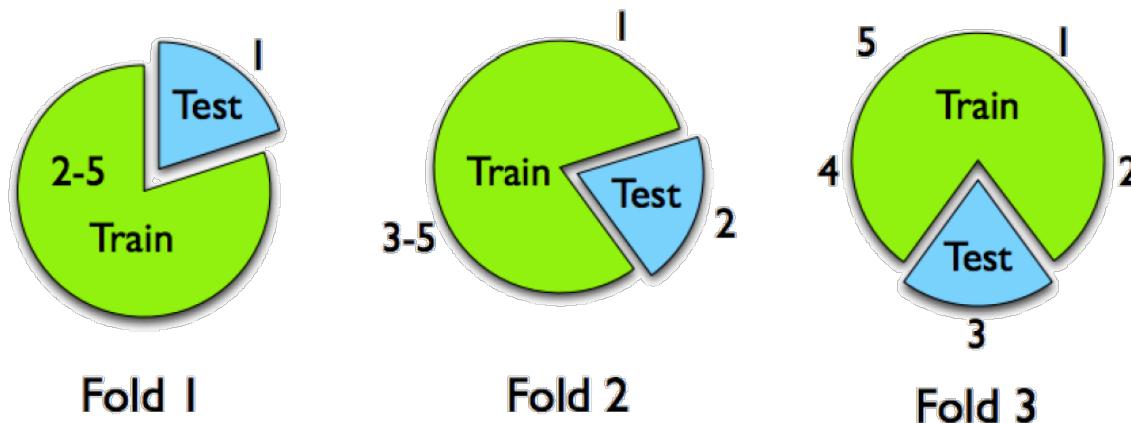
# Stratified Sampling

- 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

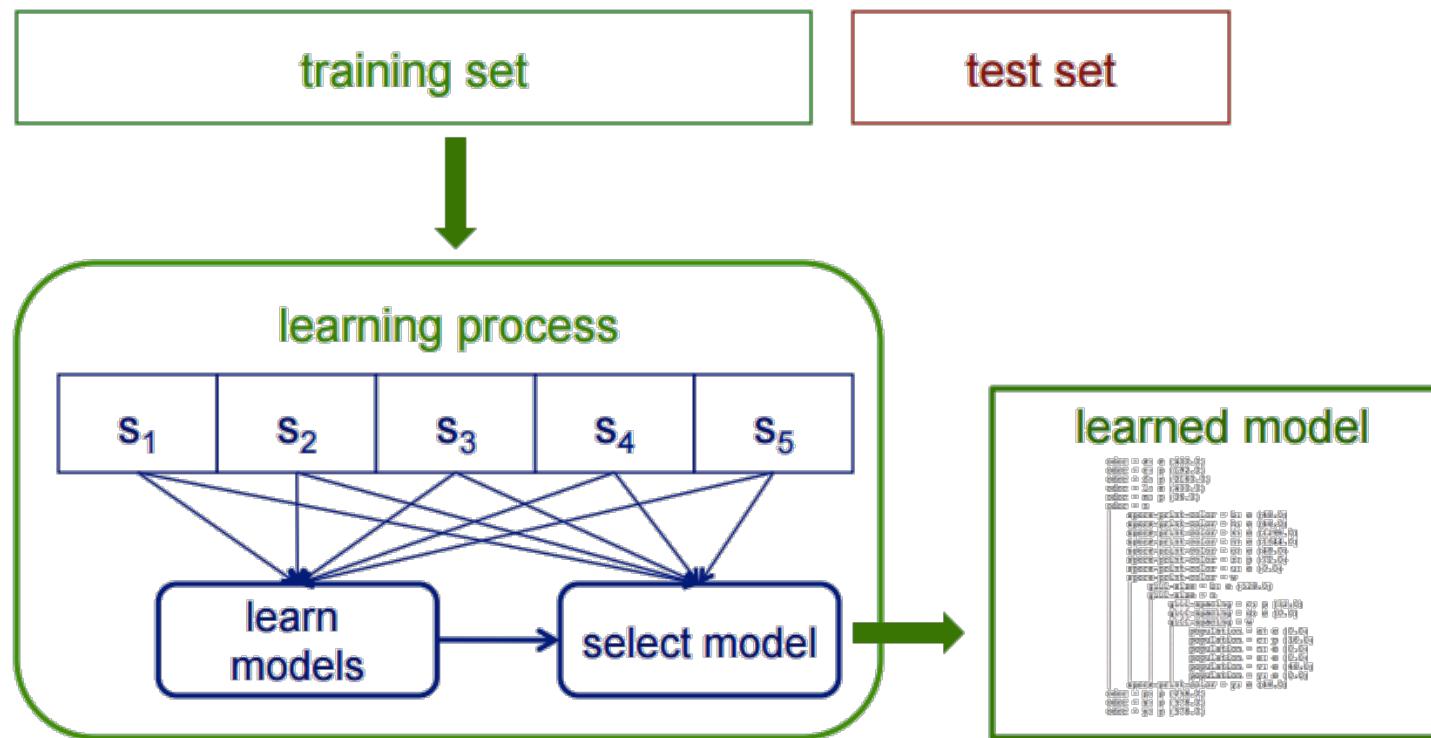
iteration	train on	test on	correct
1	s <sub>2</sub> s <sub>3</sub> s <sub>4</sub> s <sub>5</sub>	s <sub>1</sub>	11 / 20
2	s <sub>1</sub> s <sub>3</sub> s <sub>4</sub> s <sub>5</sub>	s <sub>2</sub>	17 / 20
3	s <sub>1</sub> s <sub>2</sub> s <sub>4</sub> s <sub>5</sub>	s <sub>3</sub>	16 / 20
4	s <sub>1</sub> s <sub>2</sub> s <sub>3</sub> s <sub>5</sub>	s <sub>4</sub>	13 / 20
5	s <sub>1</sub> s <sub>2</sub> s <sub>3</sub> s <sub>4</sub>	s <sub>5</sub>	16 / 20

$$\text{Classification Accuracy} = 73/100 = 73\%$$

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?

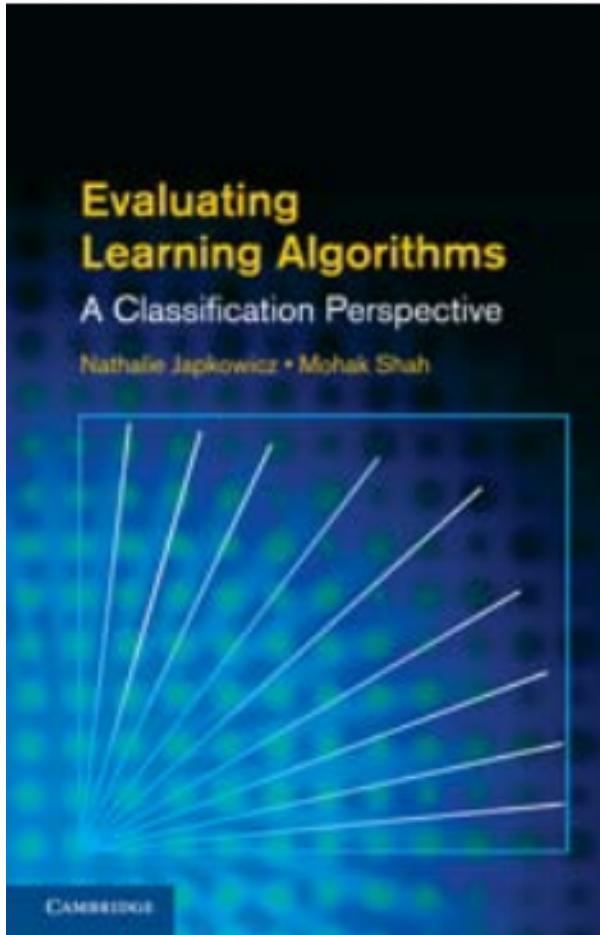
- Accuracy may not be useful in cases where
  - There is a large class skew
    - Is 98% accuracy good if 97% of the instances are negative?
  - There are differential misclassification costs – say, getting a positive wrong costs more than getting a negative wrong
    - Consider a medical domain in which 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](http://www.icmla-conference.org/icmla11/PE_Tutorial.pdf)



# Classification Error: Beyond Accuracy

In 2-class problems:

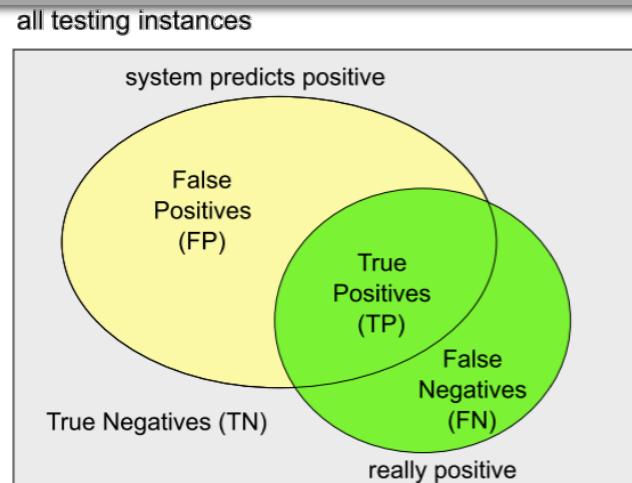
		actual class	
		positive	negative
predicted class	positive	true positives (TP)	false positives (FP)
	negative	false negatives (FN)	true negatives (TN)

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

# Classification Performance Measures

Predict positive?

	Yes	No
Yes	TP	FN
No	FP	TN



- Classification Error:  $\frac{errors}{total} = \frac{FP+FN}{TP+TN+FP+FN}$
- Accuracy = 1-Error:  $\frac{correct}{total} = \frac{TP+TN}{TP+TN+FP+FN}$
- False Alarm = False Positive rate =  $FP / (FP+TN)$
- Miss = False Negative rate =  $FN / (TP+FN)$
- Recall = True Positive rate =  $TP / (TP+FN)$
- Precision =  $TP / (TP+FP)$

meaningless  
if classes  
imbalanced

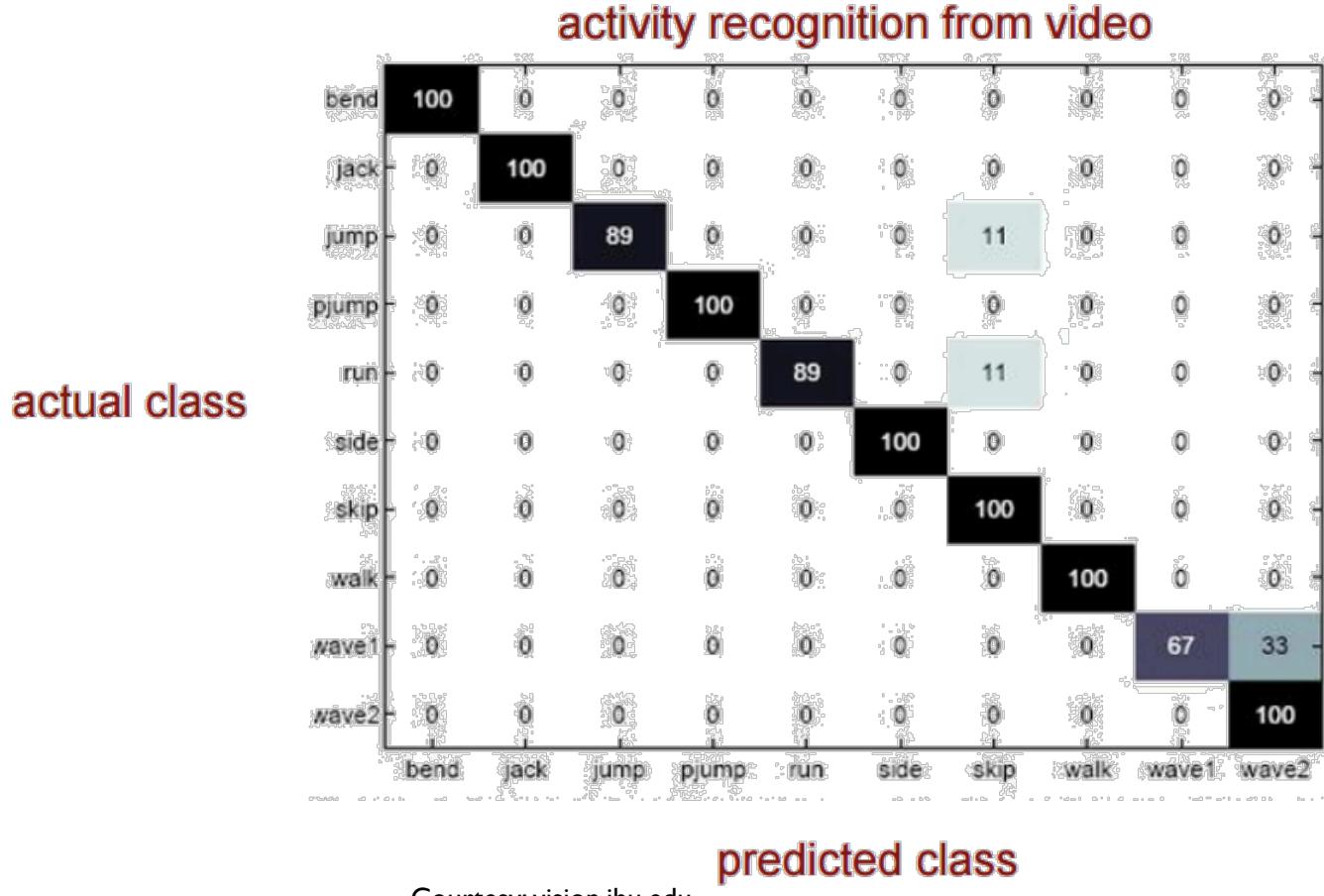
always report  
in pairs, e.g.:  
Miss / FA or  
Recall / Prec.

- True Positive Rate also called “**Sensitivity**”
- “**Specificity**” =  $1 - \text{False Alarm}$
- “**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



# 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)
  - $\text{Cost} = C_{FP} * FP + C_{FN} * FN$
- F-measure (Information Retrieval)
  - $F1 = 2/(1/\text{Recall} + 1/\text{Precision})$

CS6510  
Applied Machine Learning

# Classifier Evaluation

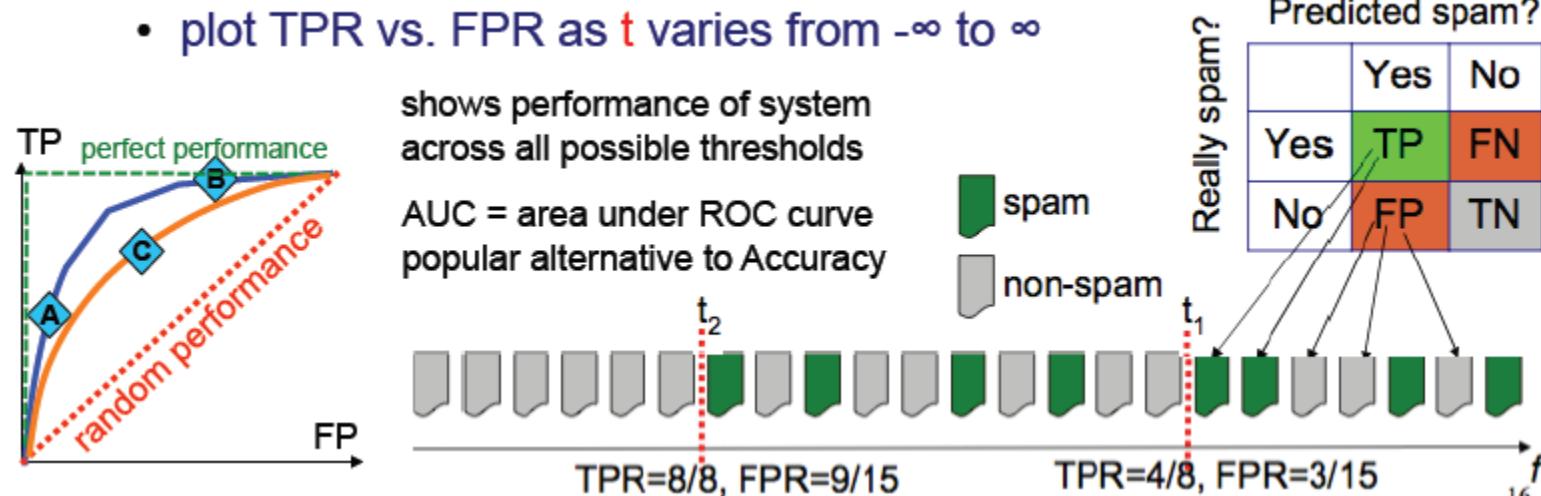
12 Aug 2017

Vineeth N Balasubramanian



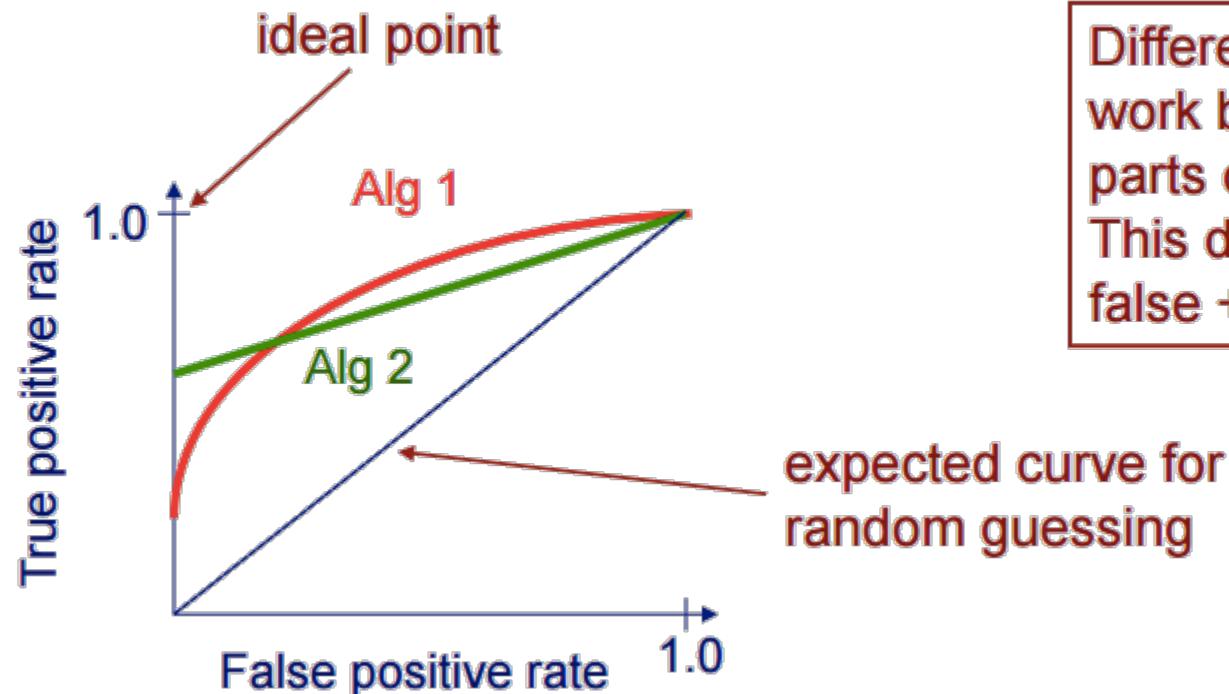
# ROC Curves

- Many algorithms compute “confidence”  $f(x)$ 
  - Threshold to get decision: spam if  $f(x) > t$ , non-spam if  $f(x) \leq t$
  - Threshold to determines error rates
- Receiver Operating Characteristic (ROC)



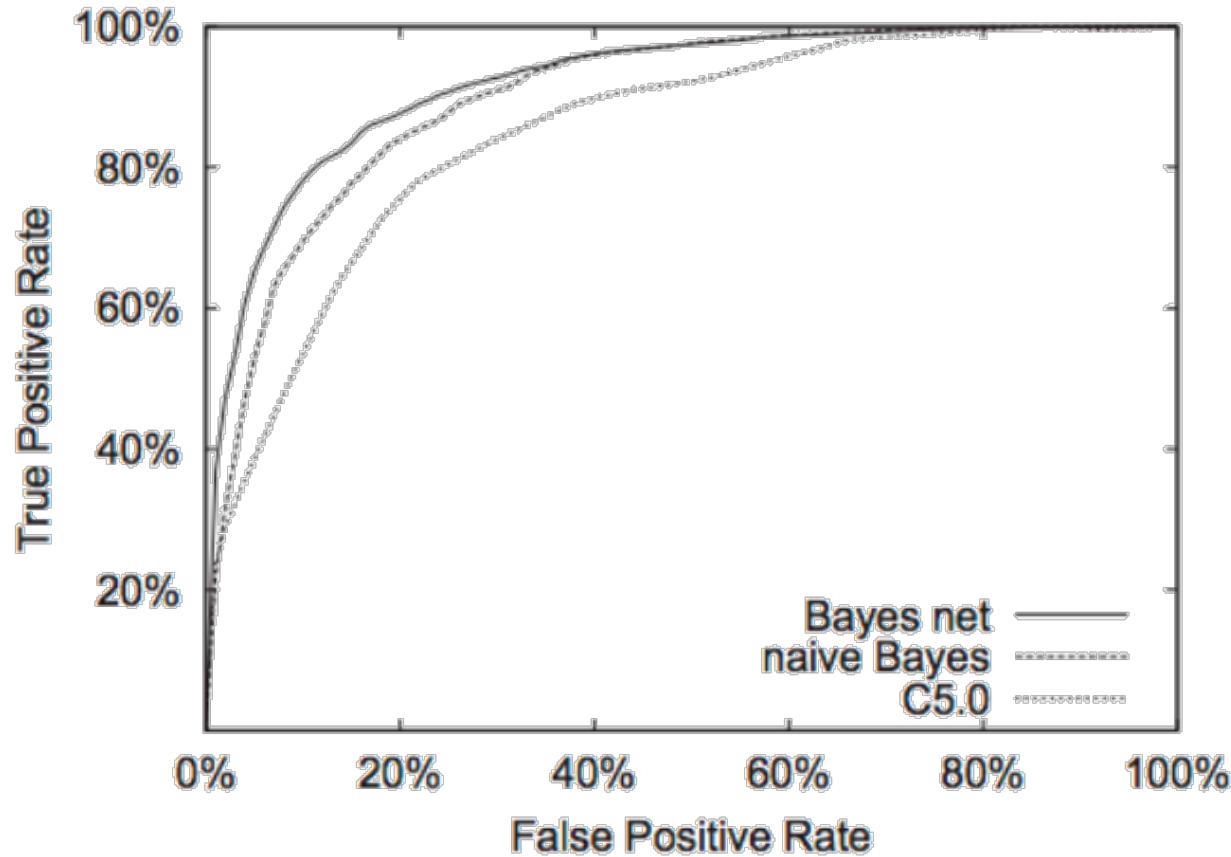
# ROC Curves

- 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



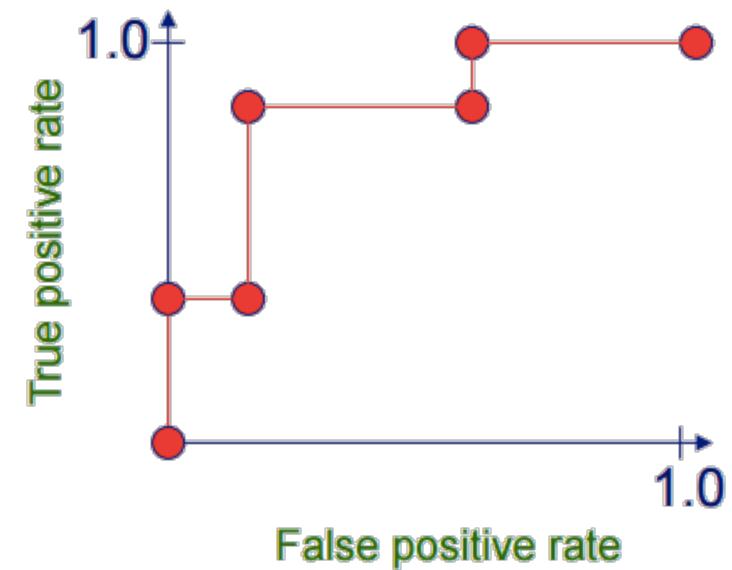
Courtesy: Bockhorst et al., Bioinformatics 2003

# 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

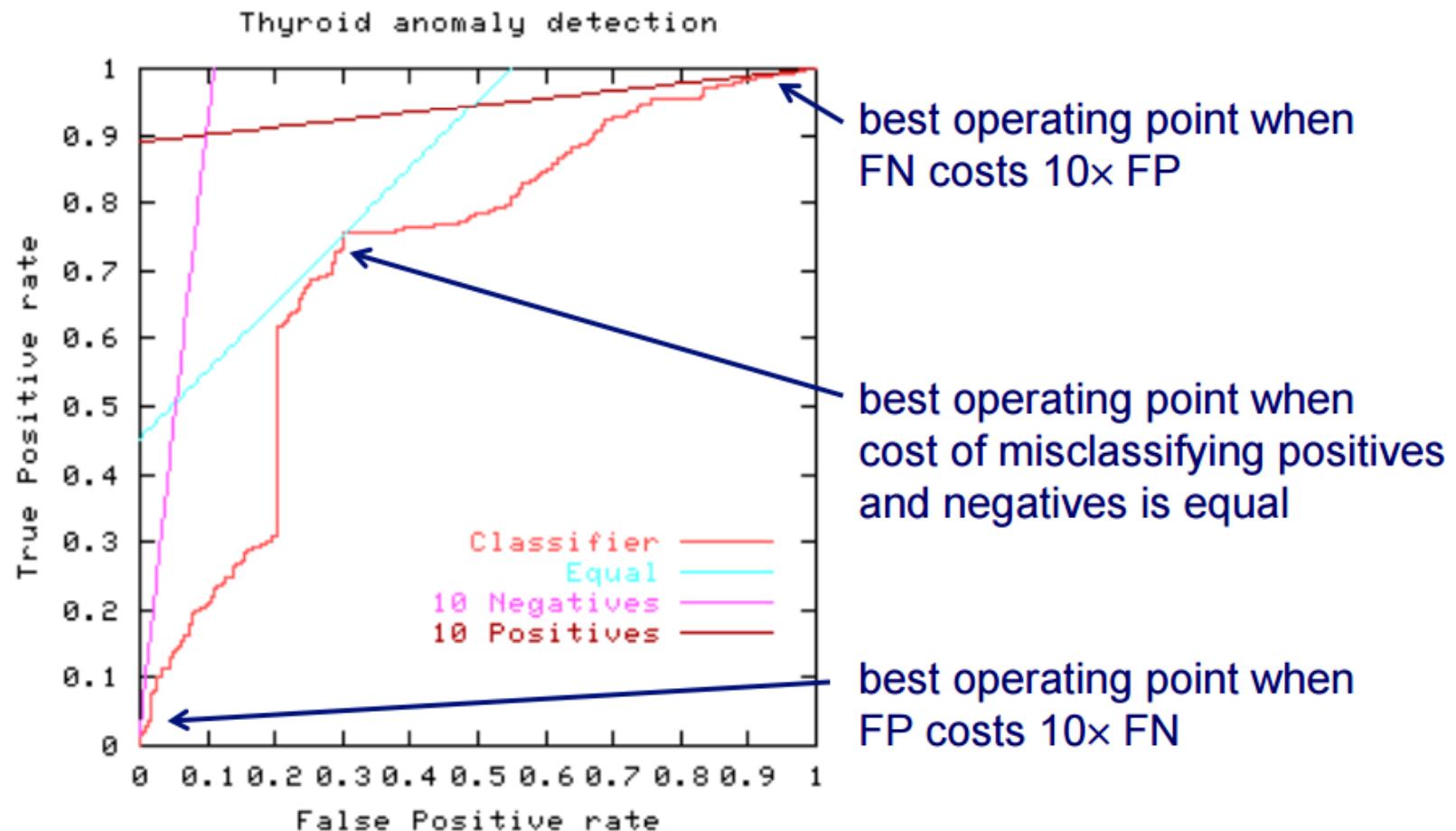


# Plotting an ROC Curve

- Can interpolate between points to get convex hull



# ROC Curves and Misclassification Costs



# Recall: Precision-Recall

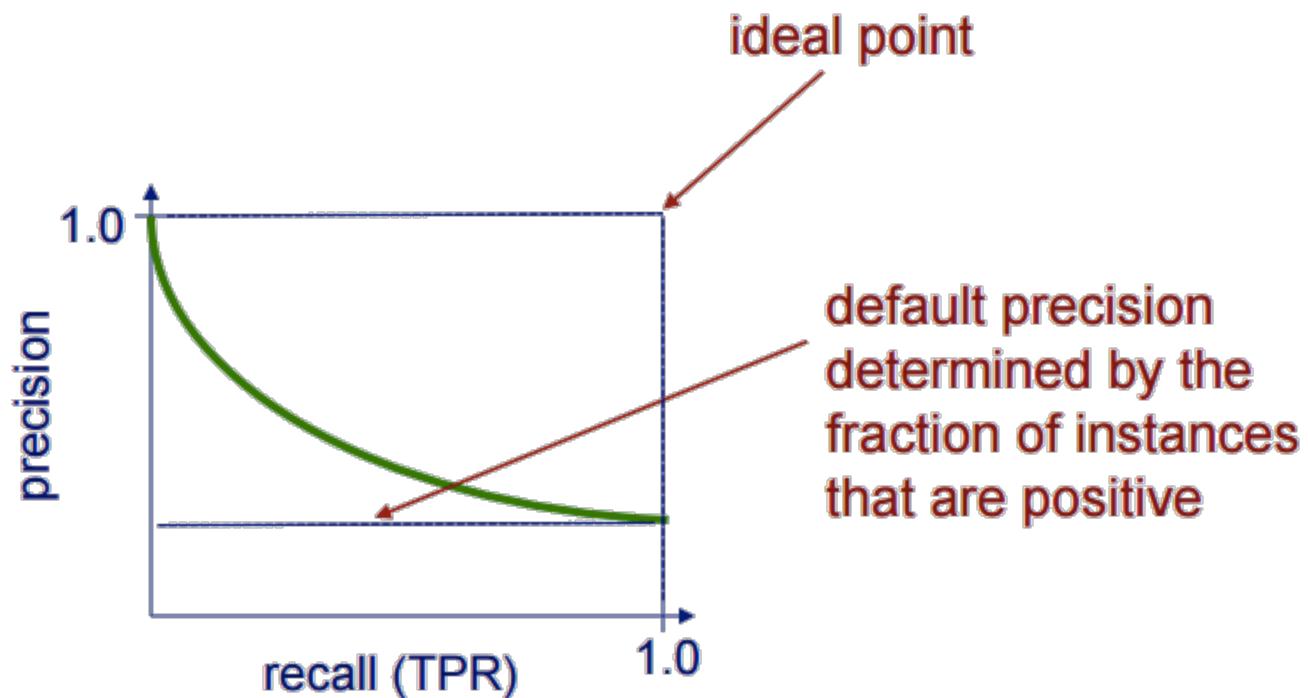
		actual class	
		positive	negative
predicted class	positive	true positives (TP)	false positives (FP)
	negative	false negatives (FN)	true negatives (TN)

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

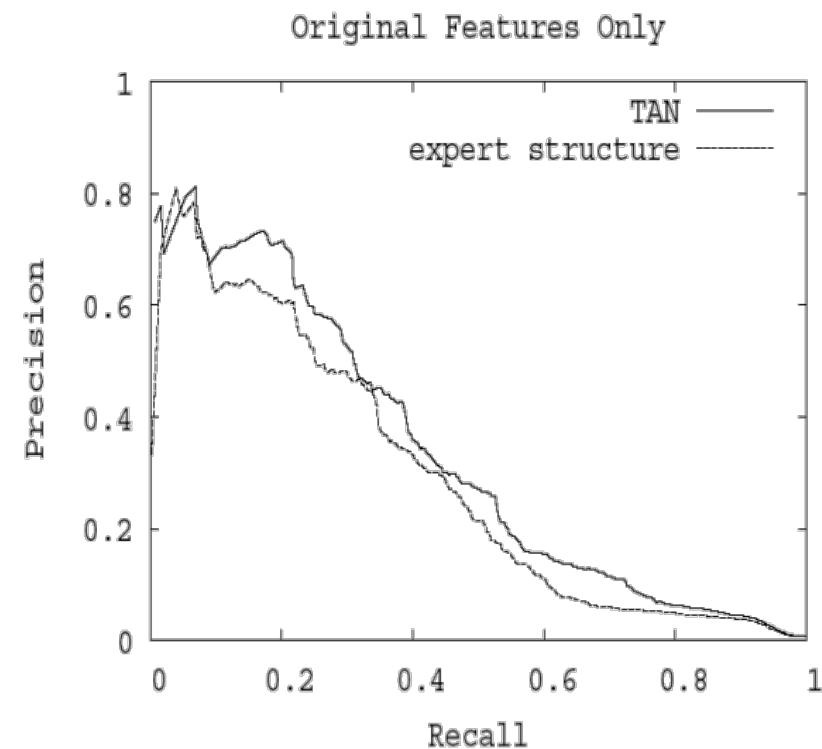
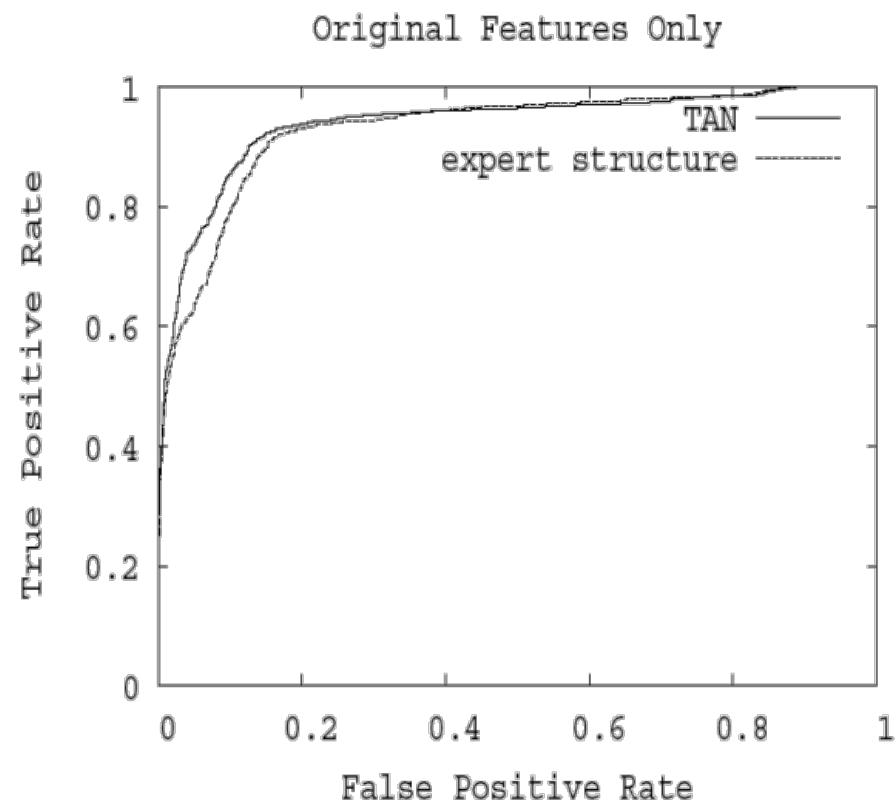
$$\text{precision} = \frac{\text{TP}}{\text{predicted pos}} = \frac{\text{TP}}{\text{TP} + \text{FP}}$$

# 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 Wisconsin-Madison

# Other Performance Measures

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

# Pitfalls

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

# 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:  $error_S(h) = \frac{r}{n}$

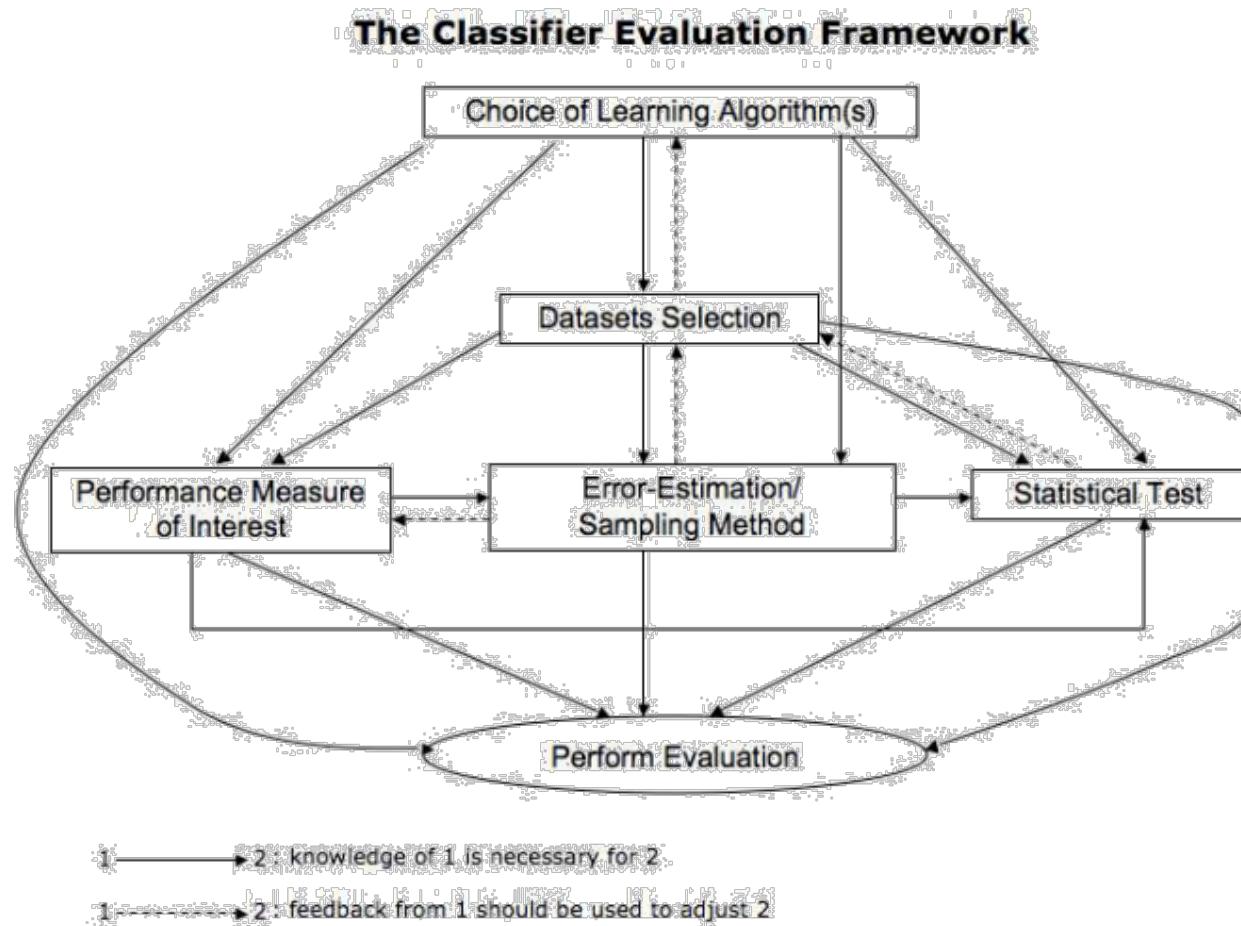
# Confidence Intervals on Error

- 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? (You will know in HWI!)

# 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 19, EA Introduction to ML, 2<sup>nd</sup> Edn
- Other Recommended References
  - [http://www.icmla-conference.org/icmla11/PE\\_Tutorial.pdf](http://www.icmla-conference.org/icmla11/PE_Tutorial.pdf) (Tutorial on Performance Evaluation of Classifiers)
  - Chapter 5 ('Evaluating Hypotheses'), Machine Learning by Tom Mitchell
    - <http://www.cs.cmu.edu/~tom/mlbook.html>

# Homework

- Go through recommended reading materials
- HW1 to be shared soon