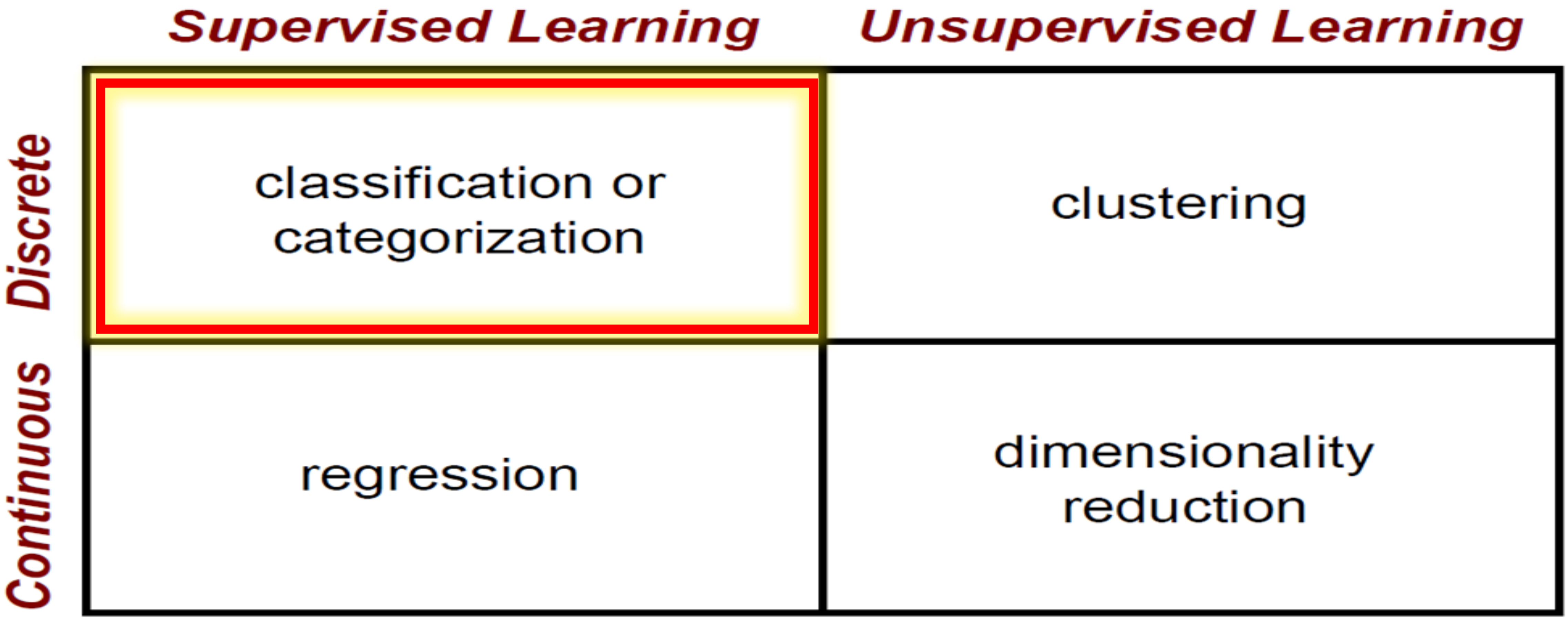
Classifier Evaluation

28 May 2019

Vineeth N Balasubramanian

आई आई टी हैदराबाद
IIT Hyderabad

ML Problems: Recall



	Supervised Learning	Unsupervised Learning
Discrete	classification or categorization	clustering
Continuous	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

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

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

How to compute generalization error?

Estimating Generalization Error

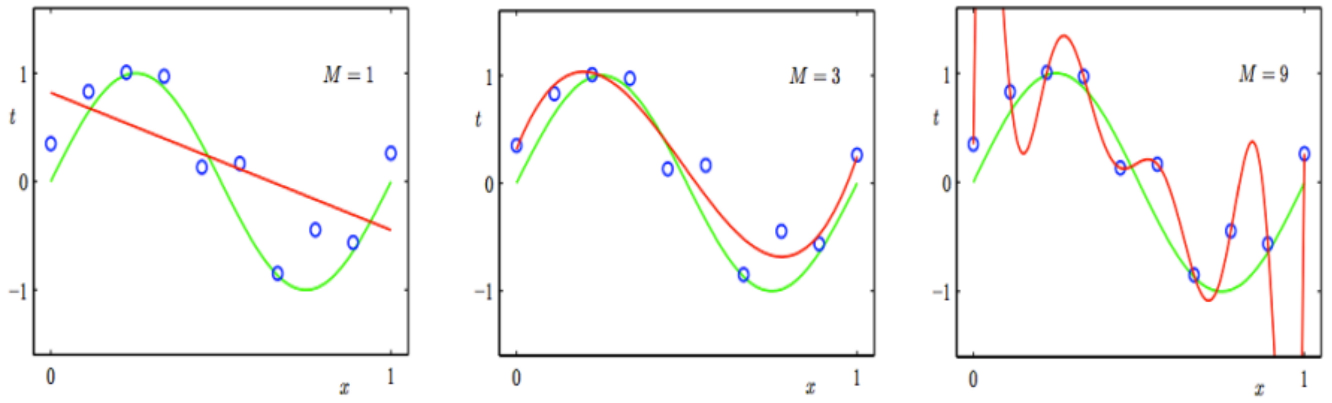
- Testing Error
  - Set aside part of training data (testing set)
  - Learn a predictor without using any of this test data
  - Predict values for testing set, compute error
  - This is an estimate of generalization error

over testing set

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

Underfitting and Overfitting

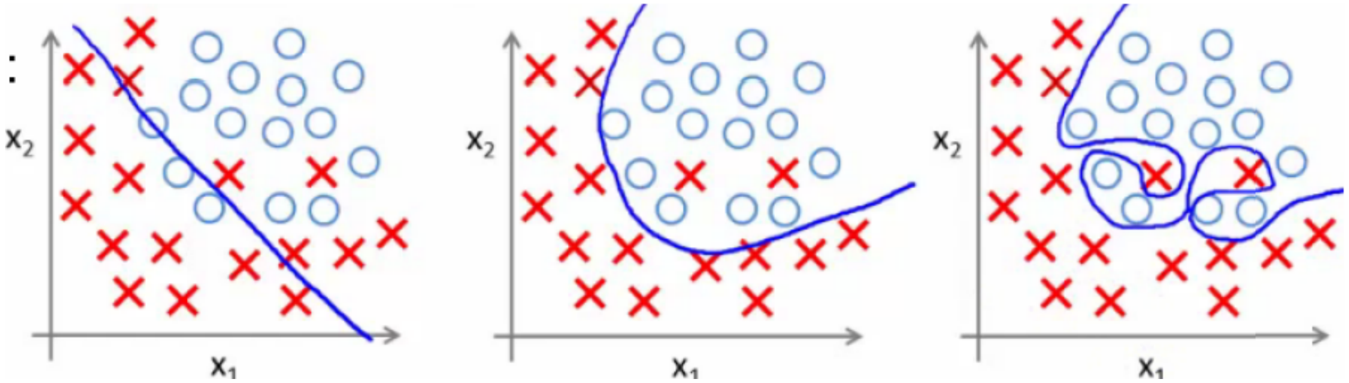
Regression



predictor too inflexible: cannot capture pattern

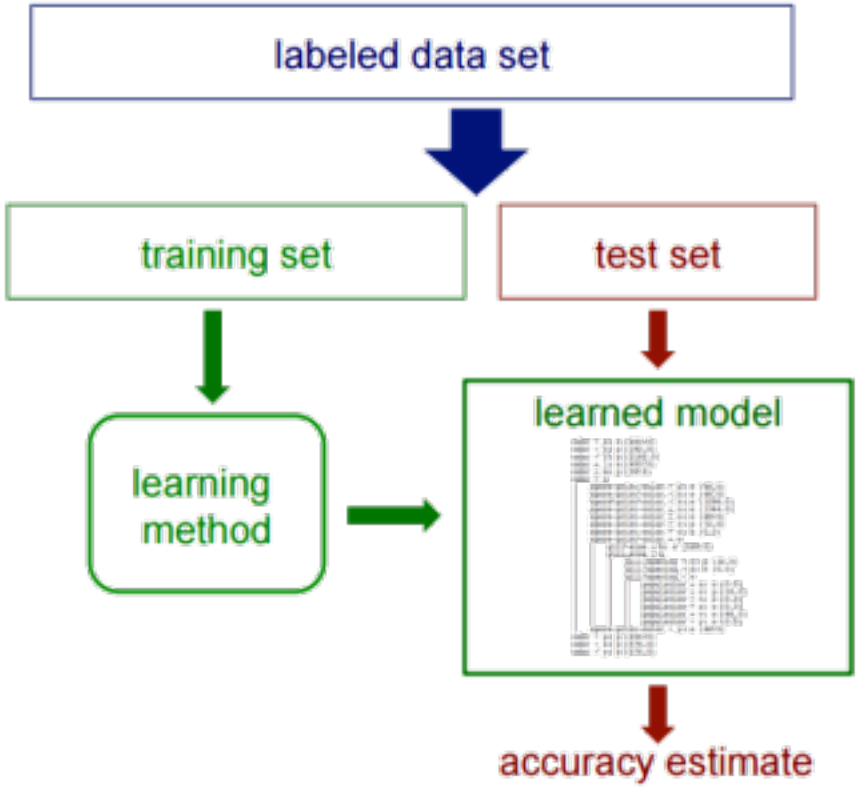
predictor too flexible: fits noise in the data

Classification

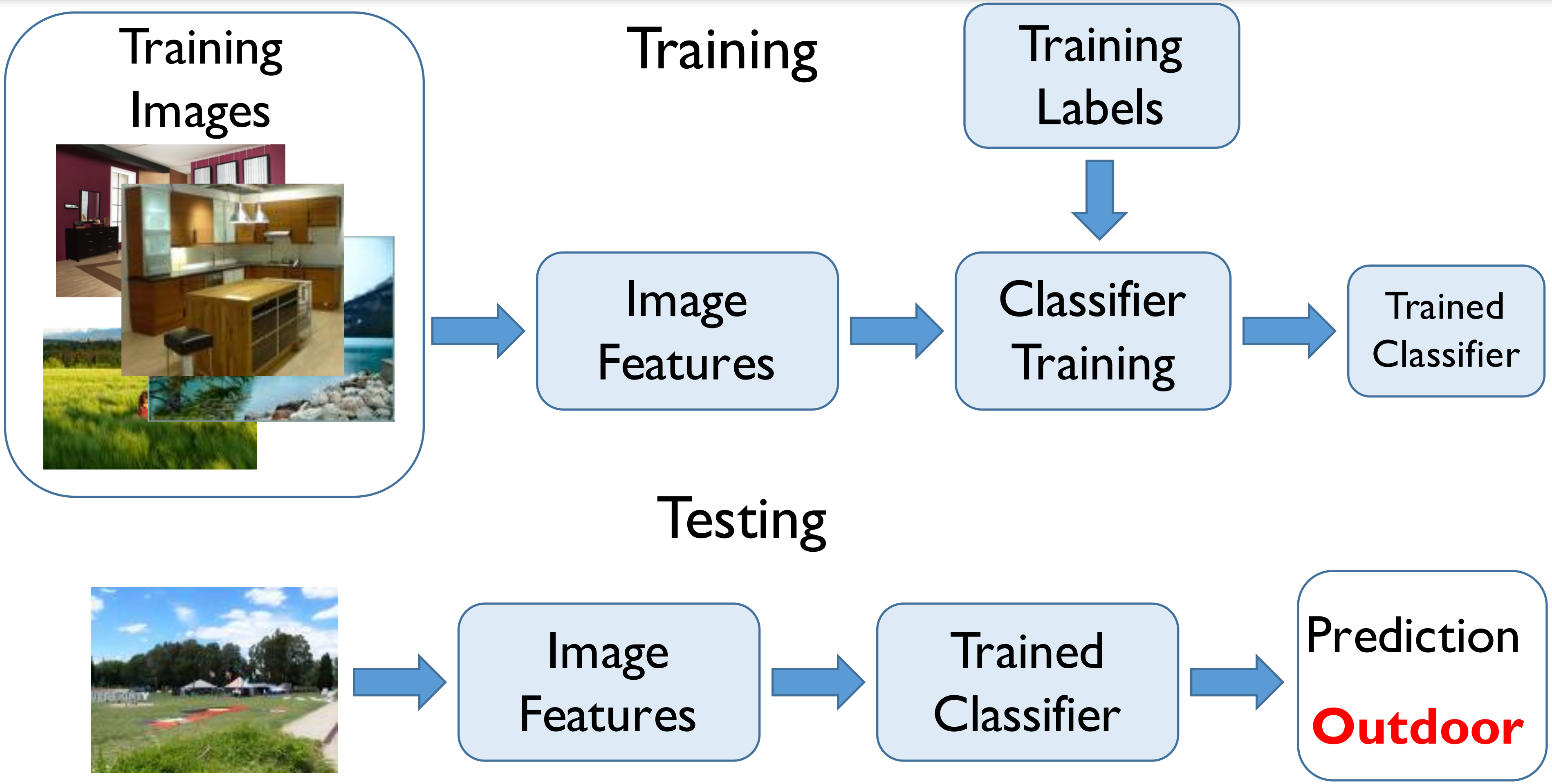


Estimating Generalization Error

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



Example: Image Classification



Training

Training Images

Training Labels

Image Features

Classifier Training

Trained Classifier

Testing

Image Features

Trained Classifier

Prediction

Outdoor

Source: Derek Hoiem

Training, Validation, Test Sets

Training set

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

Validation set

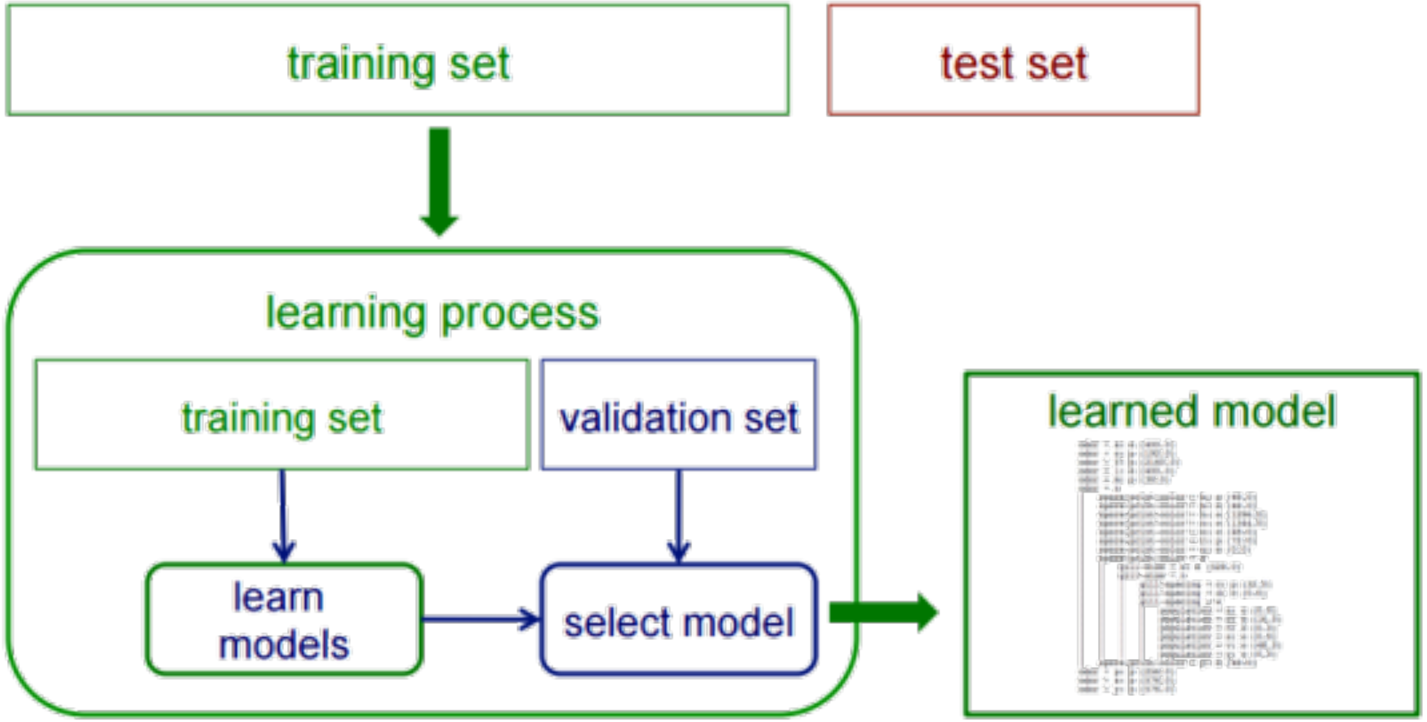
- Pick best-performing algorithm (NB vs DT vs..)
- Fine-tune parameters (Tree depth, k in kNN, c in SVM)

Testing set

- Run multiple trials and average

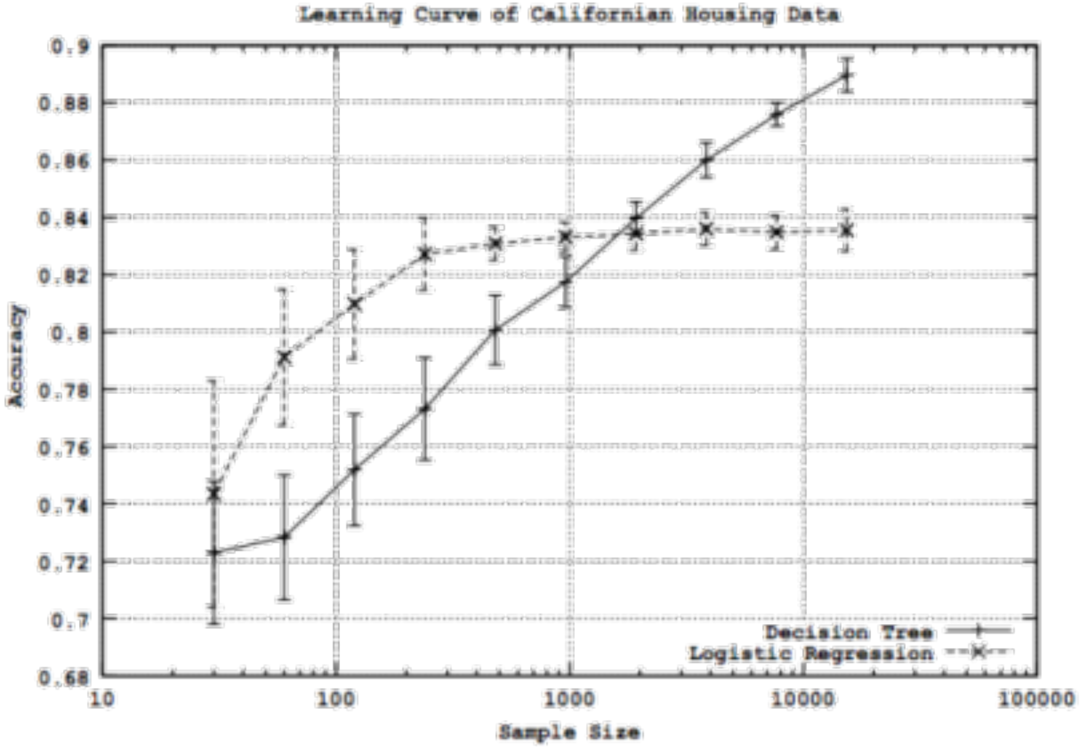
Use of Validation Sets

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



Choosing Training, Validation, Test Sets

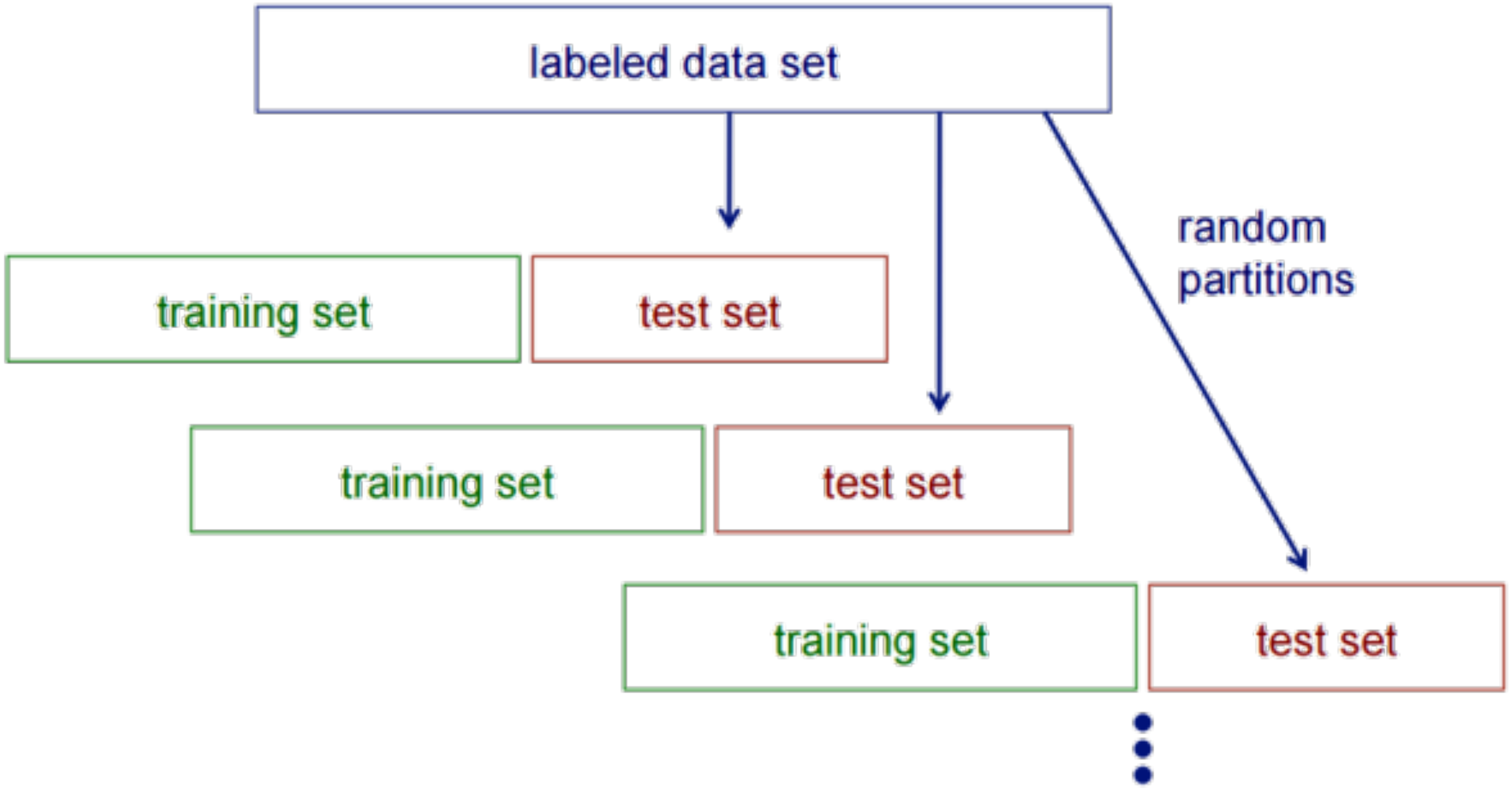
- Split randomly to avoid bias
- Large test set -> estimate future error as accurately as possible (vs) Large training set => better estimates
- How large should a training set be?
    - Study accuracy/error (vs) training set size



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

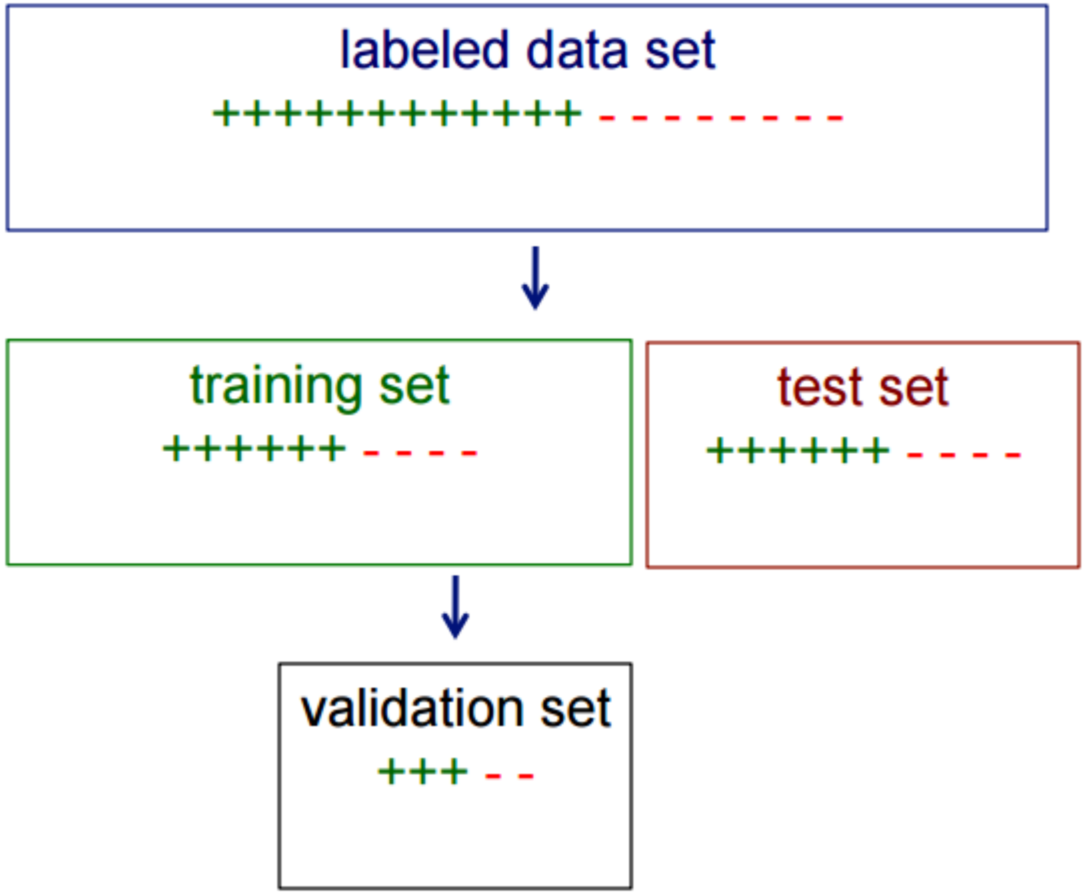
Random Resampling

- 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.



labeled data set

++++++++++++ - - - - - - - -

training set

++++++ - - - -

test set

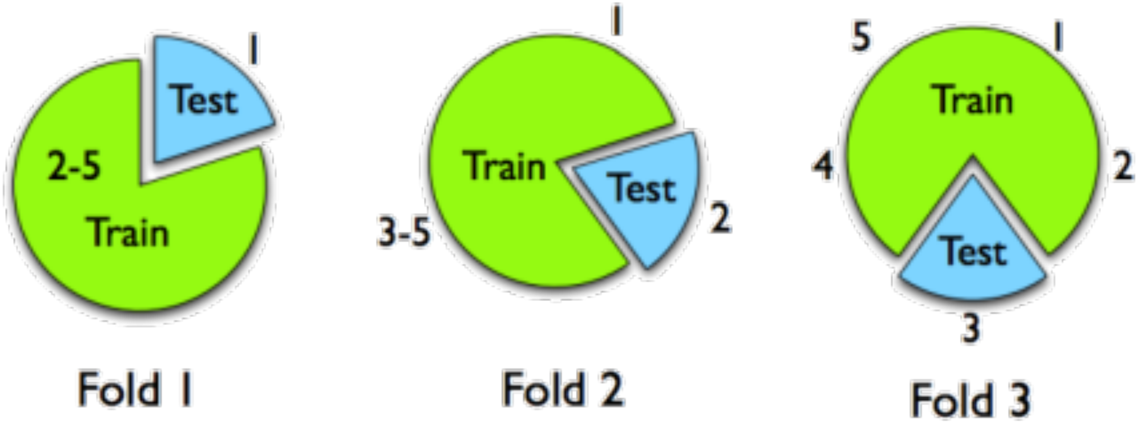
++++++ - - - -

validation set

+++ - -

Model Selection

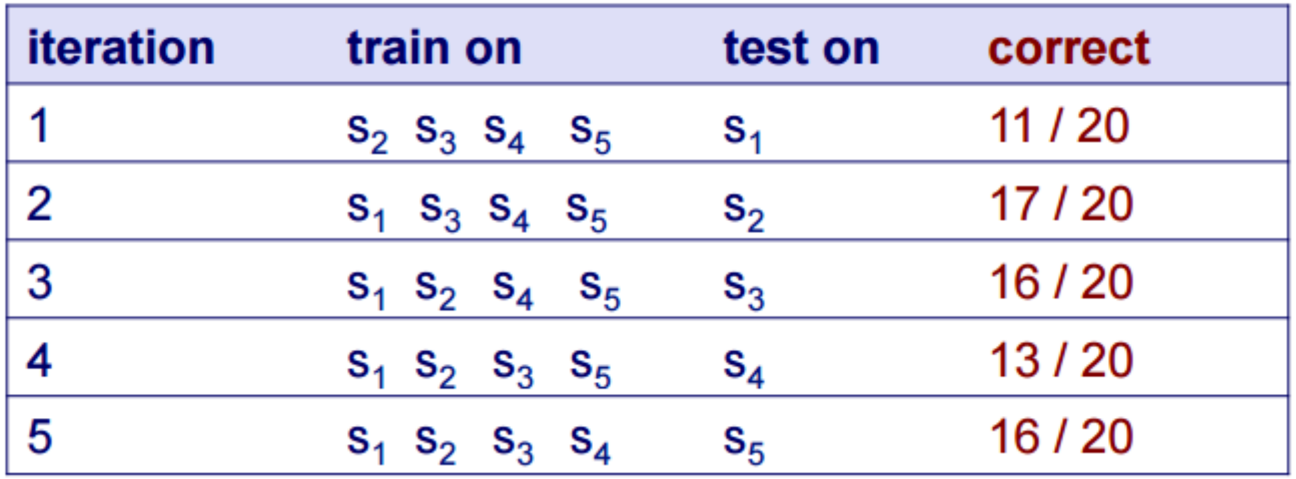
- Resubstitution
- K-fold cross-validation



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

Cross-Validation: Example

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



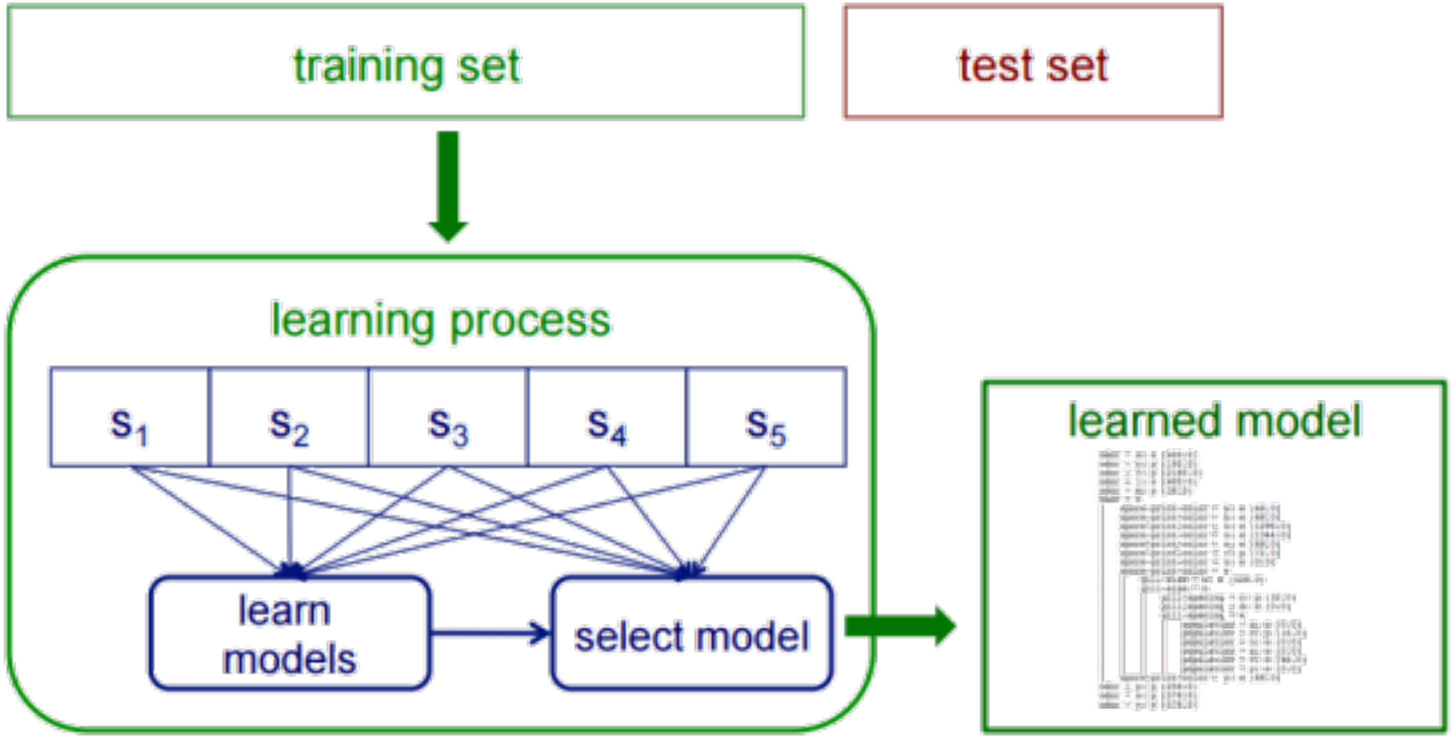
iteration	train on	test on	correct
1	s_2 s_3 s_4 s_5	s_1	11 / 20
2	s_1 s_3 s_4 s_5	s_2	17 / 20
3	s_1 s_2 s_4 s_5	s_3	16 / 20
4	s_1 s_2 s_3 s_5	s_4	13 / 20
5	s_1 s_2 s_3 s_4	s_5	16 / 20

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?

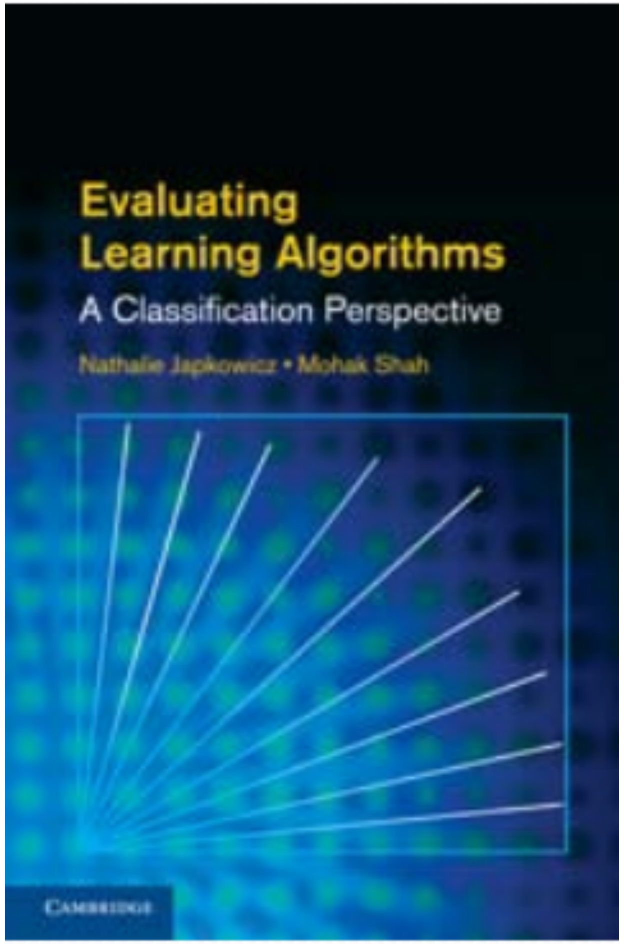
- Accuracy may not be useful in cases where
  - There is a large class skew
    - Is 98% accuracy good if 97% of the instances are negative?
  - There are differential misclassification costs – say, getting a positive wrong costs more than getting a negative wrong
    - Consider a medical domain in which 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

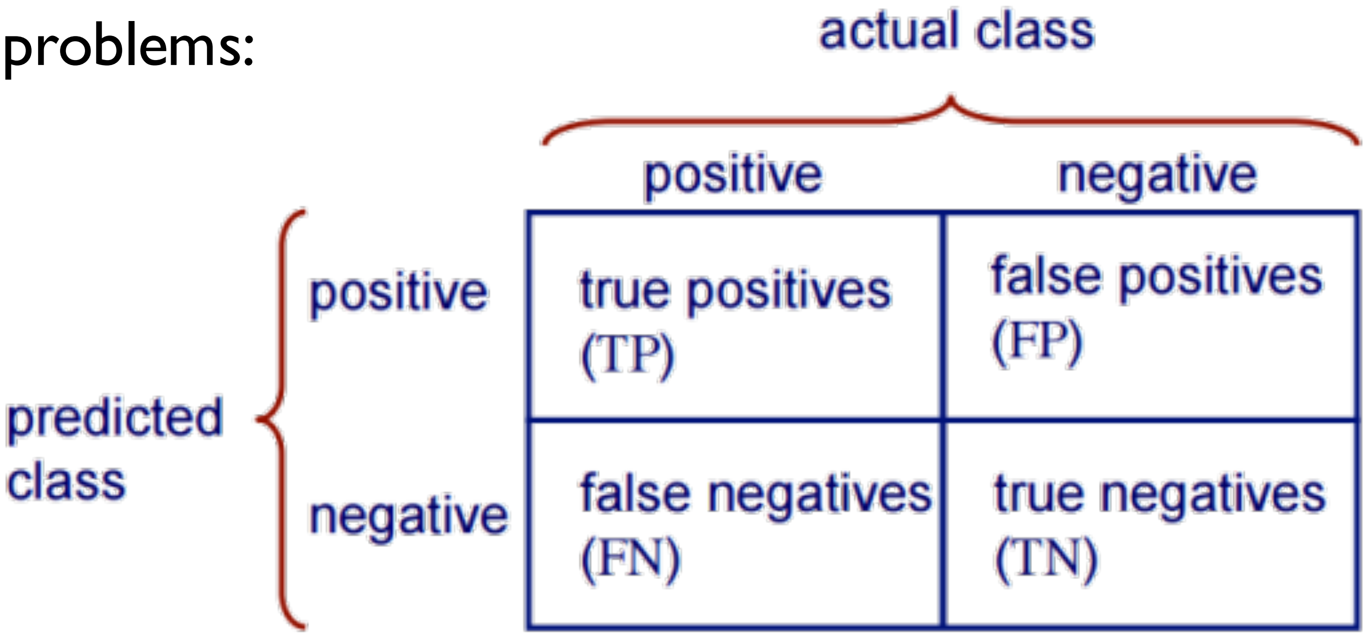
Nathalie Japkowicz & Mohak Shah
Cambridge University Press, 2011

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



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)

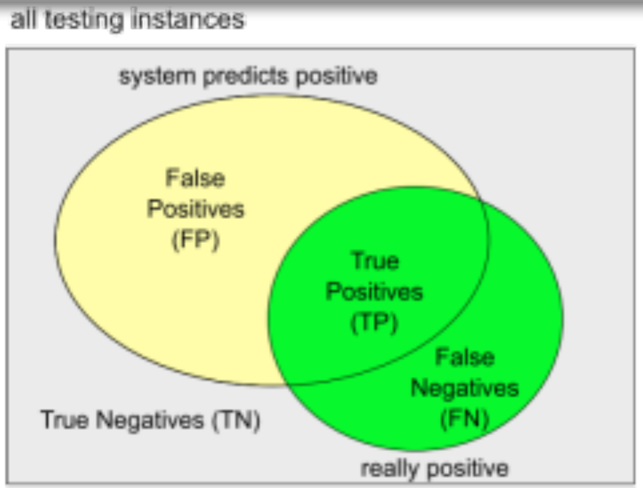
$$\text{accuracy} = \frac{TP + TN}{TP + FP + FN + TN}$$

Classification Performance Measures

Predict positive?

Really positive?

	Yes	No
Yes	TP	FN
No	FP	TN



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

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.

- True Positive Rate also called “Sensitivity”
- “Specificity” = 1 – 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

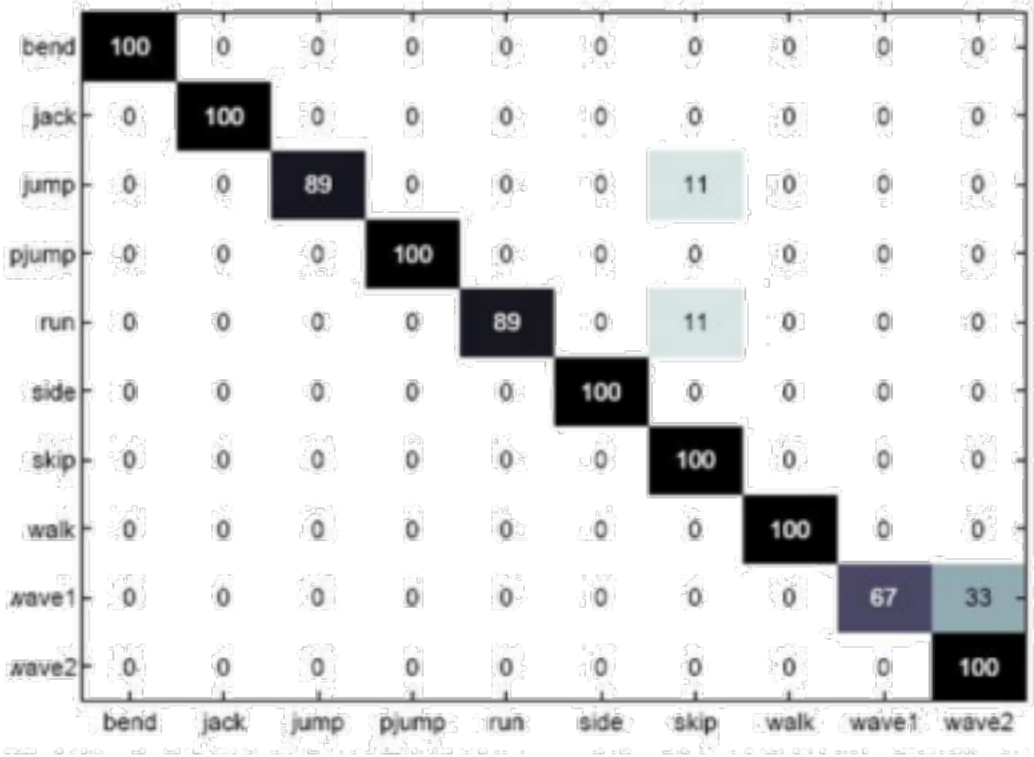
Classification Error: Beyond Accuracy

For multi-class problems?

Confusion Matrix

activity recognition from video

actual class



	bend	jack	jump	pjump	run	side	skip	walk	wave1	wave2
bend	100	0	0	0	0	0	0	0	0	0
jack	0	100	0	0	0	0	0	0	0	0
jump	0	0	89	0	0	0	11	0	0	0
pjump	0	0	0	100	0	0	0	0	0	0
run	0	0	0	0	89	0	11	0	0	0
side	0	0	0	0	0	100	0	0	0	0
skip	0	0	0	0	0	0	100	0	0	0
walk	0	0	0	0	0	0	0	100	0	0
wave1	0	0	0	0	0	0	0	0	67	33
wave2	0	0	0	0	0	0	0	0	0	100

predicted class

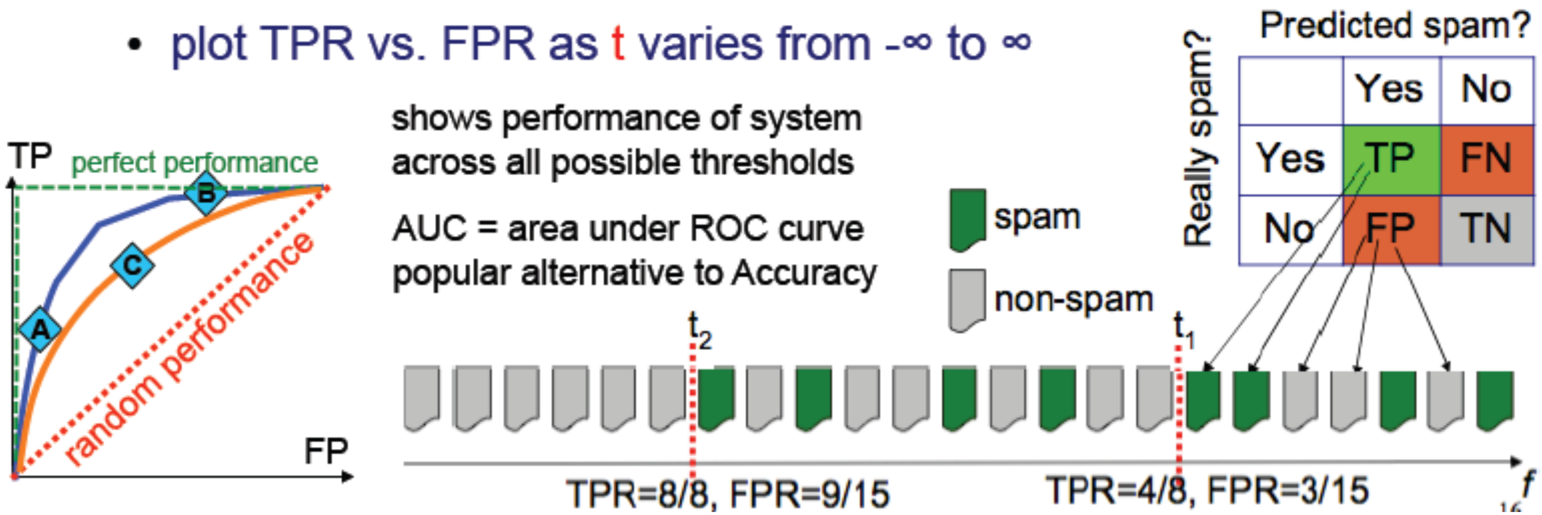
Courtesy: vision.jhu.edu

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 determine error rates
- Receiver Operating Characteristic (ROC)



- plot TPR vs. FPR as t varies from -∞ to ∞

shows performance of system across all possible thresholds

AUC = area under ROC curve
popular alternative to Accuracy

Predicted spam?

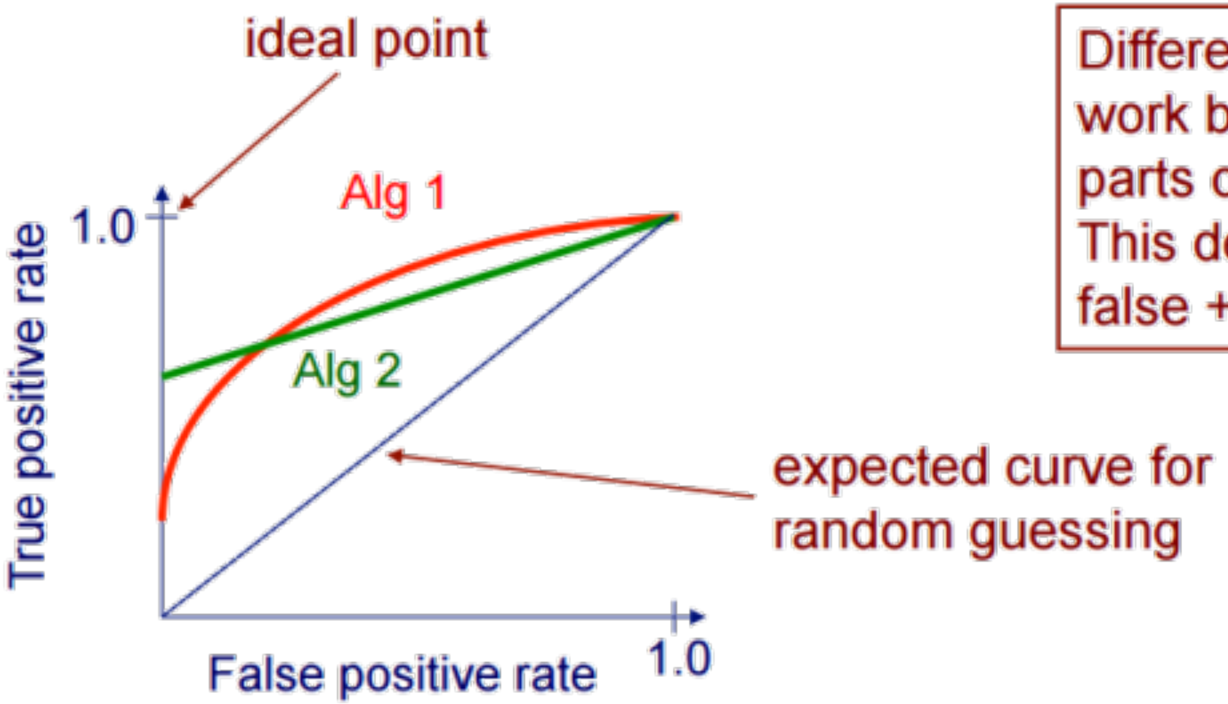
Really spam?	Yes	No
Yes	TP	FN
No	FP	TN

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

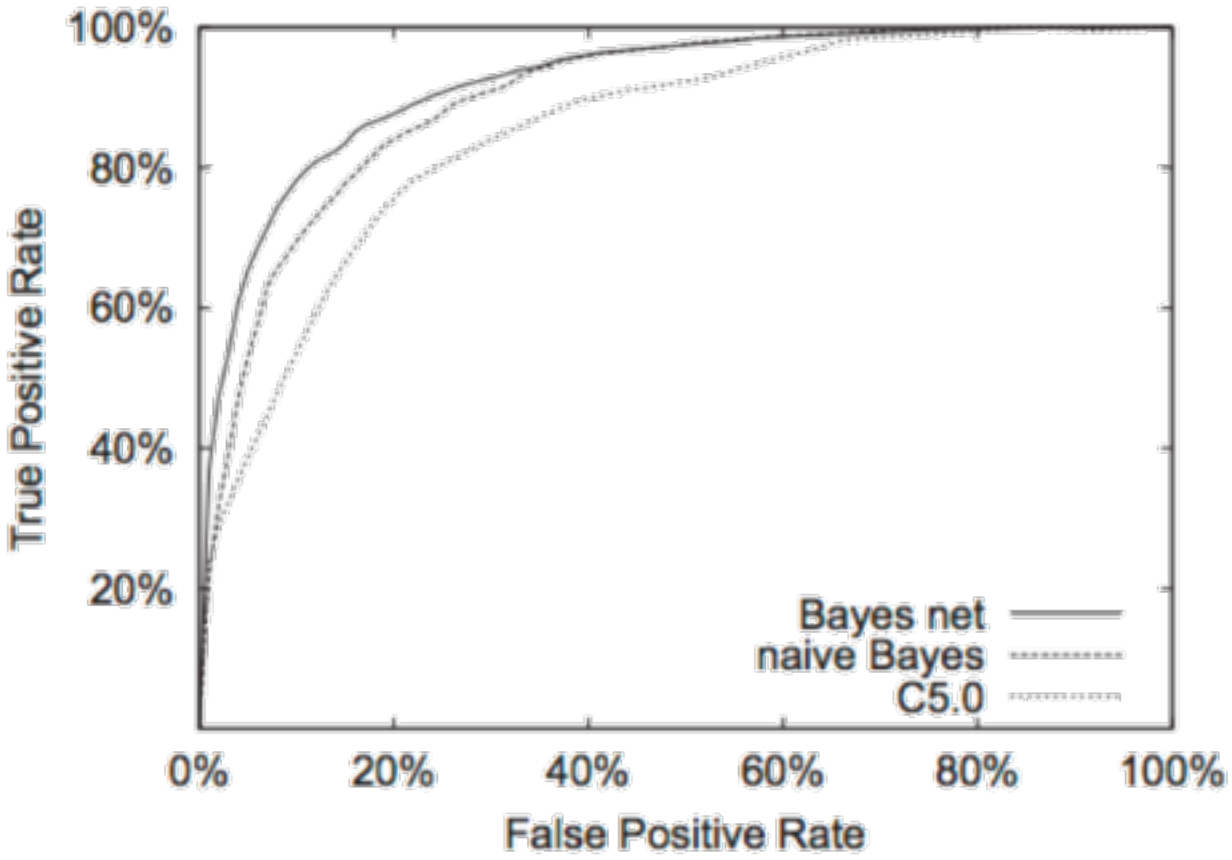
ROC Curves

- 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 -

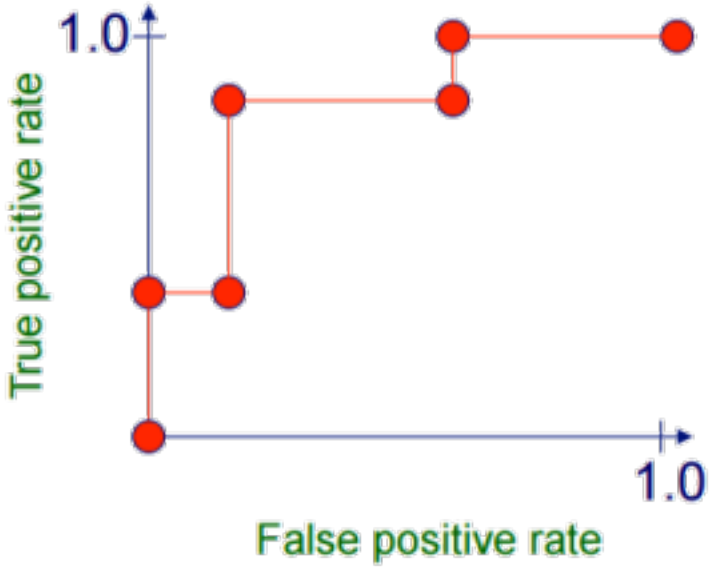
ROC Curve: Example



Courtesy: Bockhorst et al., Bioinformatics 2003

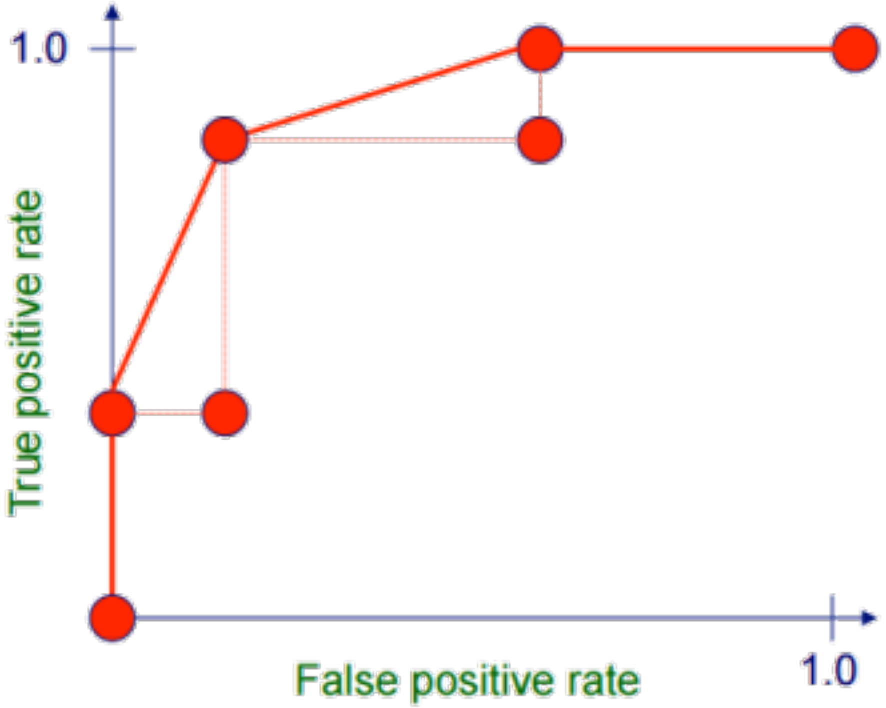
Plotting an ROC Curve

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	-

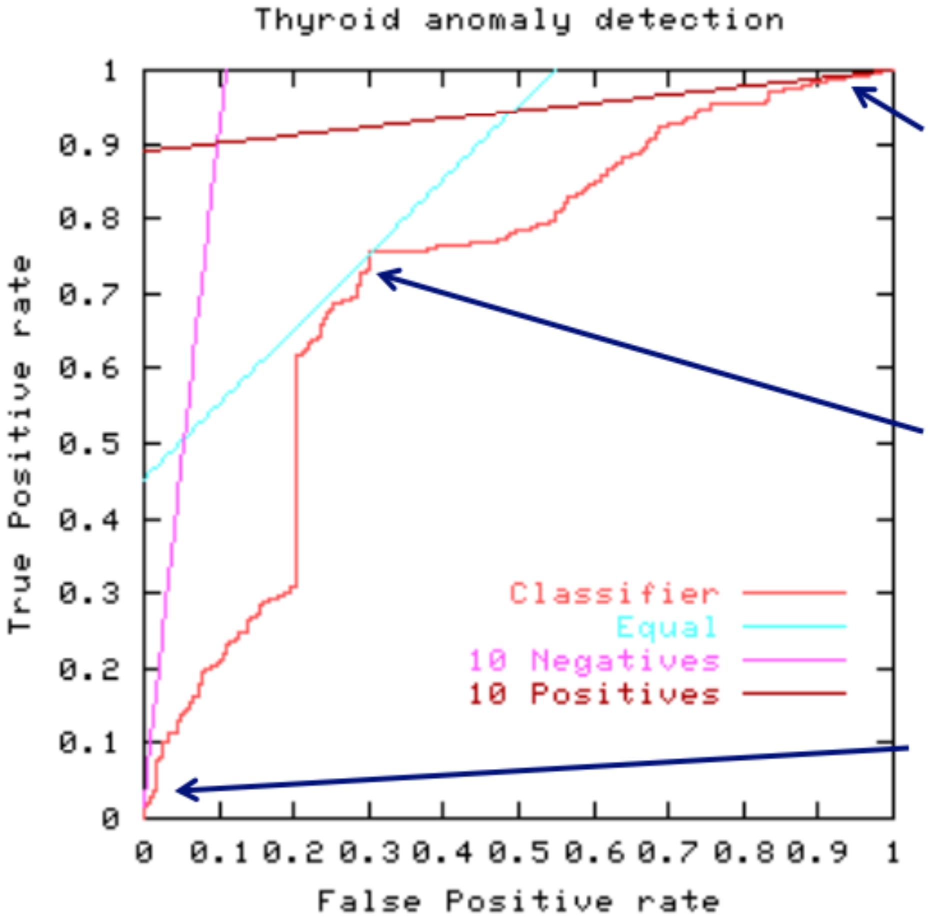


Plotting an ROC Curve

- Can interpolate between points to get convex hull



ROC Curves and Misclassification Costs

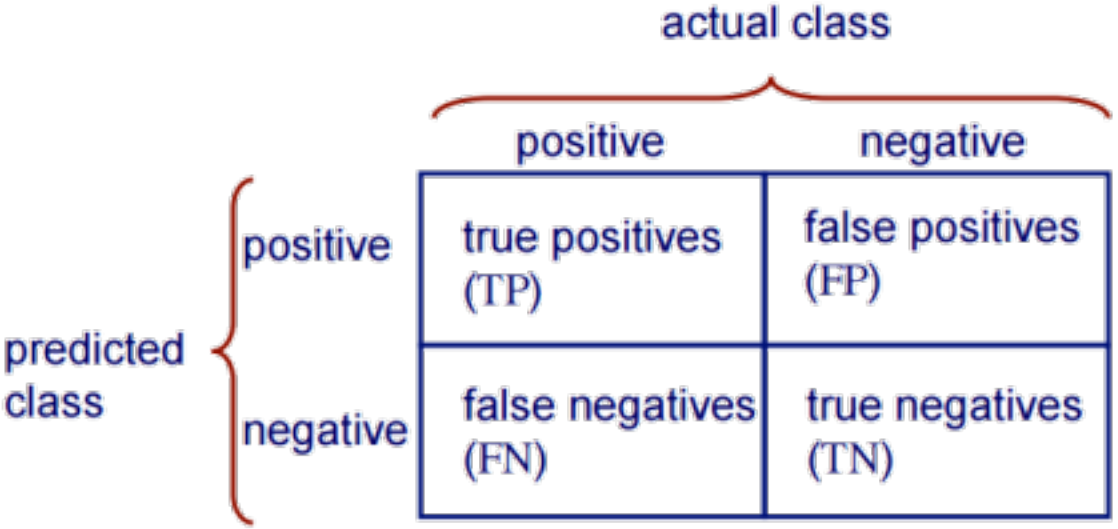


best operating point when FN costs 10× FP

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

best operating point when FP costs 10× FN

Recall: Precision-Recall



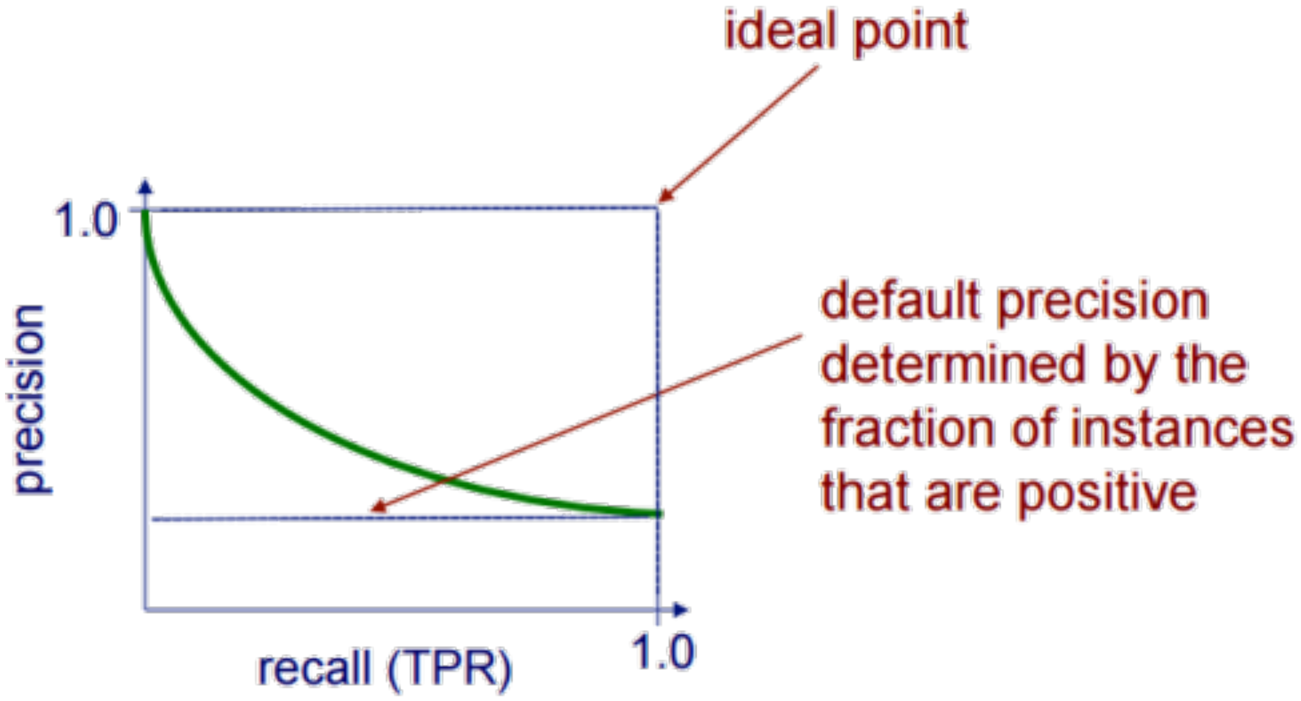
		actual class	
		positive	negative
predicted class	positive	true positives (TP)	false positives (FP)
	negative	false negatives (FN)	true negatives (TN)

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

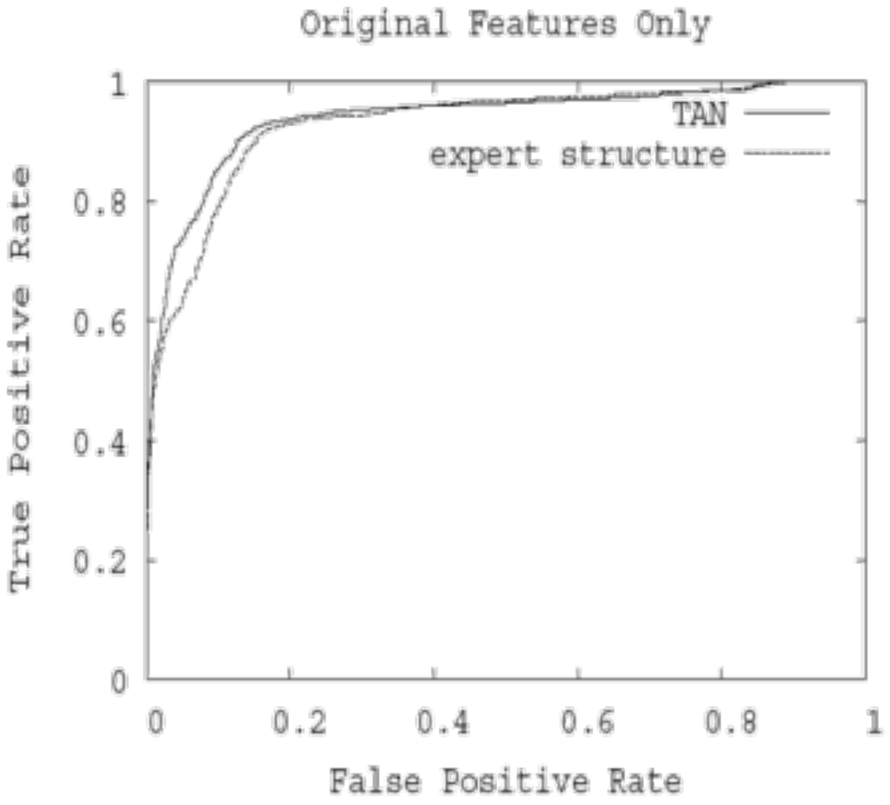
$$\text{precision} = \frac{\text{TP}}{\text{predicted pos}} = \frac{\text{TP}}{\text{TP} + \text{FP}}$$

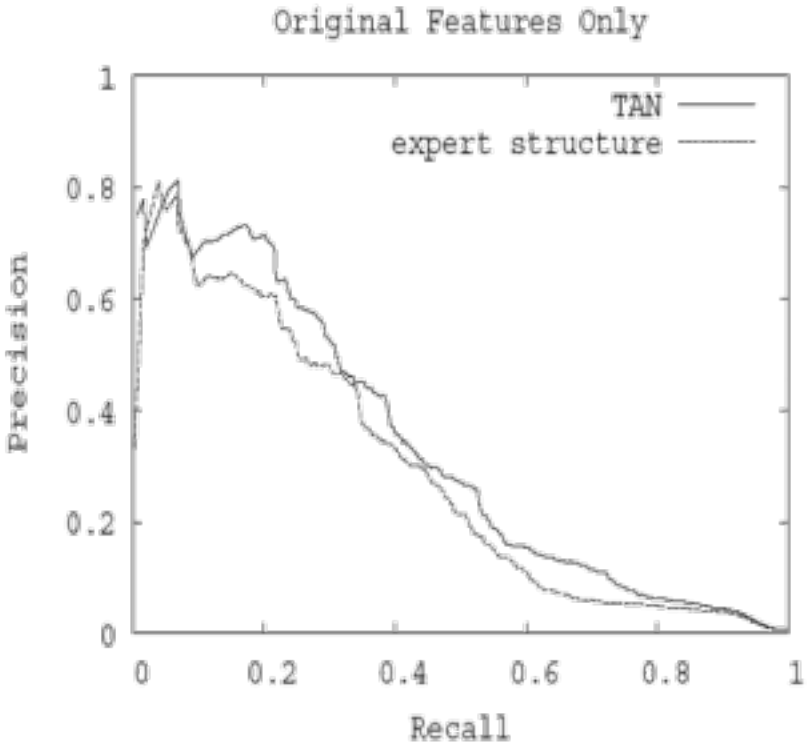
Precision/Recall Curves

- 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 Wisconsion-Madison

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
  - Used for relative comparison between models

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

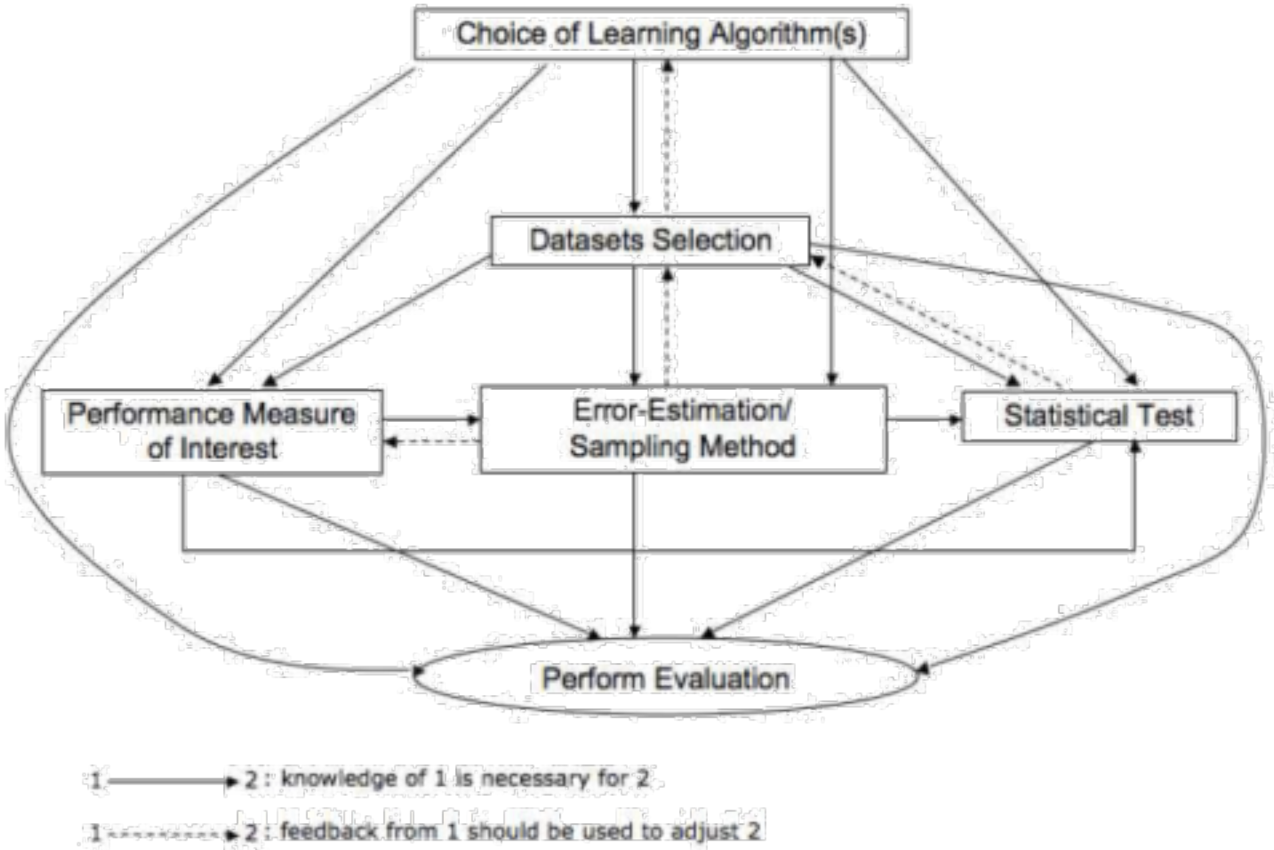
- With approximately N% probability, the true error lies in the interval

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

- where z_N is a constant that depends on N (e.g. for 95% confidence, z_N =1.96)
- For more information, please see
  - https://machinelearningmastery.com/confidence-intervals-for-machine-learning/

Classifier Evaluation

The Classifier Evaluation Framework


Choice of Learning Algorithm(s)
Datasets Selection
Performance Measure of Interest
Error-Estimation/ Sampling Method
Statistical Test
Perform Evaluation

1 ⟶ 2 : knowledge of 1 is necessary for 2

1 - - - - ▸ 2 : feedback from 1 should be used to adjust 2

Summarizing: 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

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
  - Introduction to Machine Learning, Ethem Alpaydin, 2nd Edn, Chapter 19
  - Pattern Recognition and Machine Learning, Christopher Bishop, Chapter 1 (Sec 1.1-1.5)
- Other Recommended References
  - 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