# Evaluating Machine Learning Methods: Part 1

Yingyu Liang
Computer Sciences 760
Fall 2017

http://pages.cs.wisc.edu/~yliang/cs760/

Some of the slides in these lectures have been adapted/borrowed from materials developed by Mark Craven, David Page, Jude Shavlik, Tom Mitchell, Nina Balcan, Elad Hazan, Tom Dietterich, and Pedro Domingos.

# Goals for the lecture

you should understand the following concepts

- bias of an estimator
- learning curves
- stratified sampling
- cross validation
- confusion matrices
- TP, FP, TN, FN
- ROC curves

# Goals for the next lecture

you should understand the following concepts

- PR curves
- confidence intervals for error
- pairwise $t$-tests for comparing learning systems
- scatter plots for comparing learning systems
- lesion studies

# Bias of an estimator

$\theta$ true value of parameter of interest (e.g. model accuracy)
$\hat{\theta}$ estimator of parameter of interest (e.g. test set accuracy)

$$\mathrm{Bias}[\hat{\theta}] = \mathrm{E}[\hat{\theta}] - \theta$$

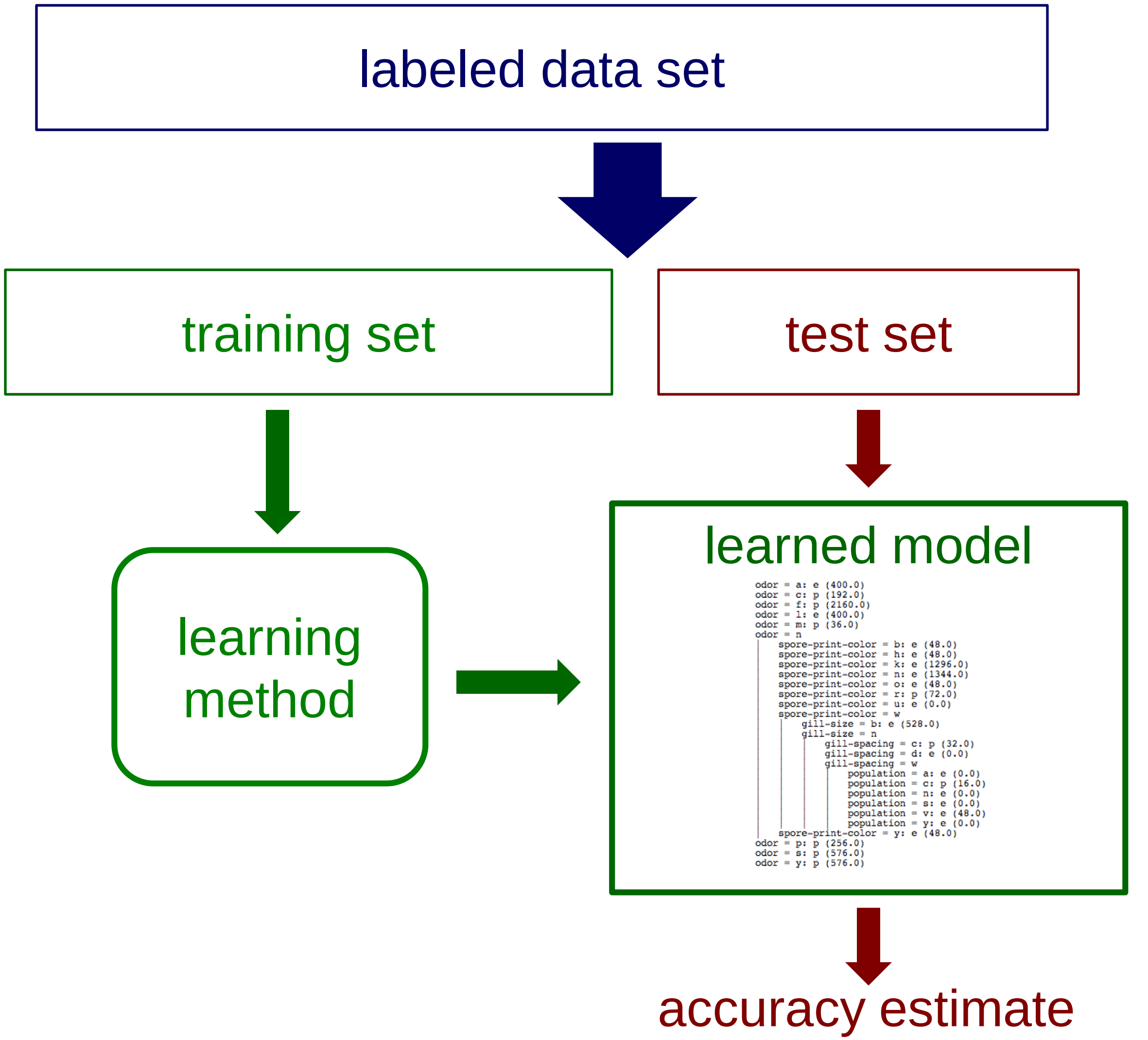
e.g. polling methodologies often have an inherent bias

**FiveThirtyEight**

| POLLSTER | LIVE CALLER WITH CELLPHONES | INTERNET | NCPP/ AAPOR/ ROPER | POLLS ANALYZED | SIMPLE AVERAGE ERROR | RACES CALLED CORRECTLY | ADVANCED +/- | PREDICTIVE +/- | 538 GRADE | BANNED BY 538 | MEAN-REVERTED BIAS |
|---|---|---|---|---|---|---|---|---|---|---|---|
| **SurveyUSA** | | | ● | 763 | 4.6 | 90% | -1.0 | **-0.8** | A | | D+0.1 |
| **YouGov** | | ● | | 707 | 6.7 | 93% | -0.3 | **+0.1** | B | | D+1.6 |
| **Rasmussen Reports/ Pulse Opinion Research** | | | | 657 | 5.3 | 79% | +0.4 | **+0.7** | C+ | | R+2.0 |
| **Zogby Interactive/JZ Analytics** | | ● | | 465 | 5.6 | 78% | +0.8 | **+1.2** | C- | | R+0.8 |
| **Mason-Dixon Polling & Research, Inc.** | ● | | | 415 | 5.2 | 86% | -0.4 | **-0.2** | B+ | | R+1.0 |
| **Public Policy Polling** | | | | 383 | 4.9 | 82% | -0.5 | **-0.1** | B+ | | R+0.2 |
| **Research 2000** | | | | 279 | 5.5 | 88% | +0.2 | **+0.6** | F | ✖ | D+1.4 |

# Test sets revisited

How can we get an unbiased estimate of the accuracy of a learned model?



# Test sets revisited

How can we get an unbiased estimate of the accuracy of a learned model?

- when learning a model, you should pretend that you don't have the test data yet (it is "in the mail")*

- if the test-set labels influence the learned model in any way, accuracy estimates will be biased

* In some applications it is reasonable to assume that you have access to the feature vector (i.e. $\boldsymbol{x}$) but not the $y$ part of each test instance.

# Learning curves

How does the accuracy of a learning method change as a function of the training-set size?

this can be assessed by plotting *learning curves*

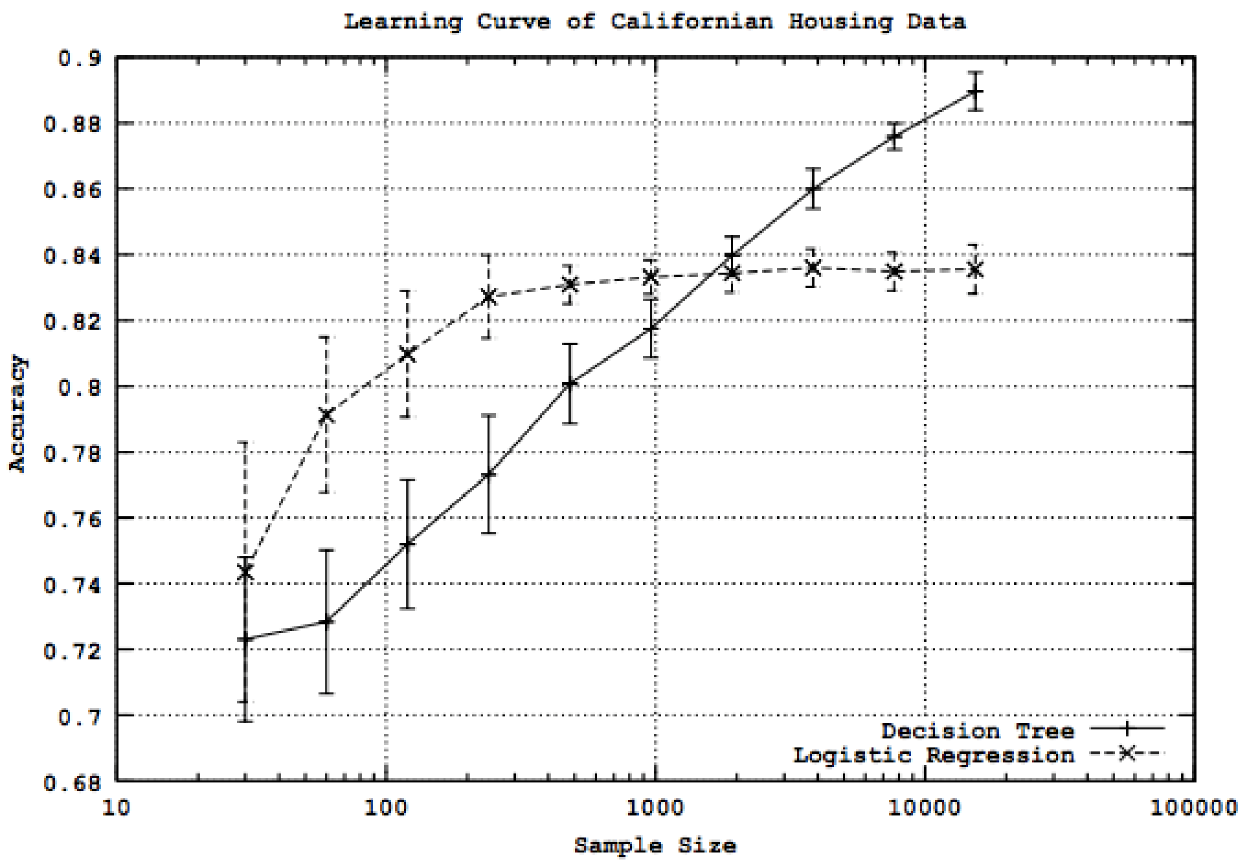
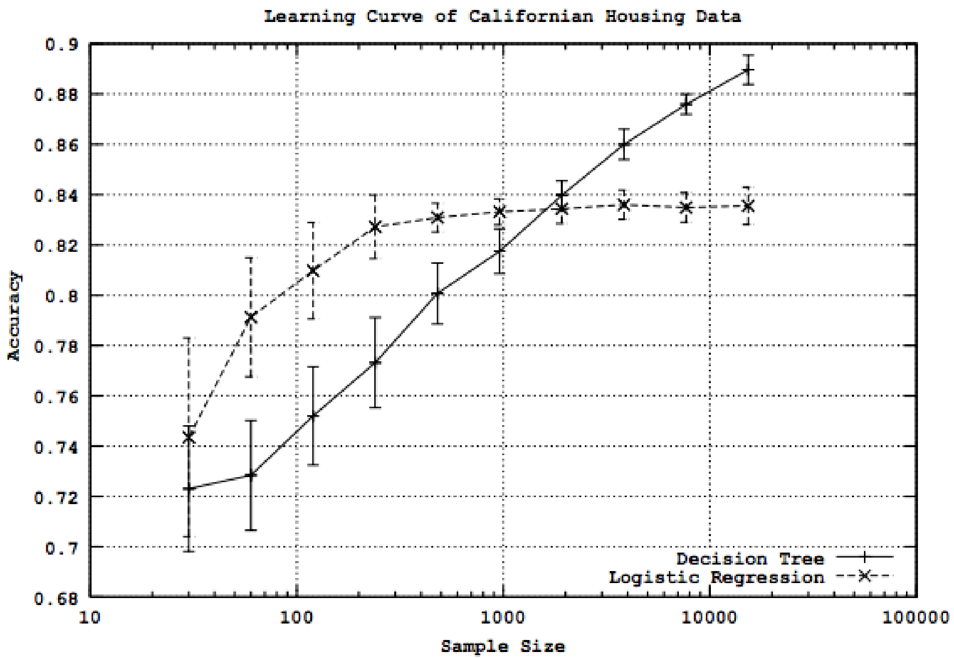



Figure from Perlich et al. *Journal of Machine Learning Research*, 2003

# Learning curves

given training/test set partition

- for each sample size $s$ on learning curve
  - (optionally) repeat $n$ times
    - randomly select $s$ instances from training set
    - learn model
    - evaluate model on test set to determine accuracy $a$
    - plot $(s, a)$ or ($s$, avg. accuracy and error bars)



# Limitations of using a single training/test partition

- we may not have enough data to make sufficiently large training and test sets
  - a <u>larger test set</u> gives us more reliable estimate of accuracy (i.e. a lower variance estimate)
  - but… a <u>larger training set</u> will be more representative of how much data we actually have for learning process

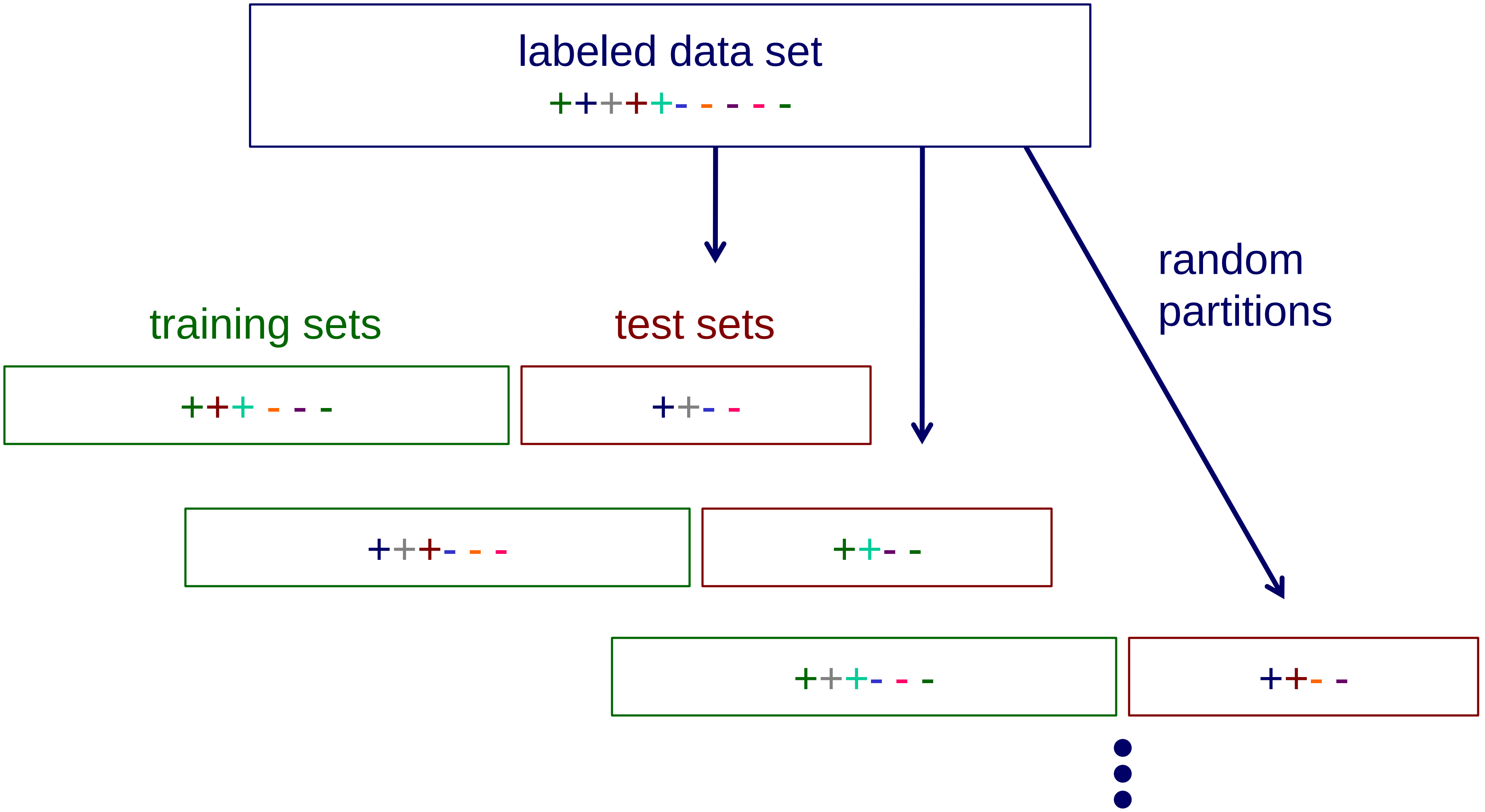
- a single training set doesn't tell us how sensitive accuracy is to a particular training sample

# Using multiple training/test partitions

- two general approaches for doing this
  - random resampling
  - cross validation

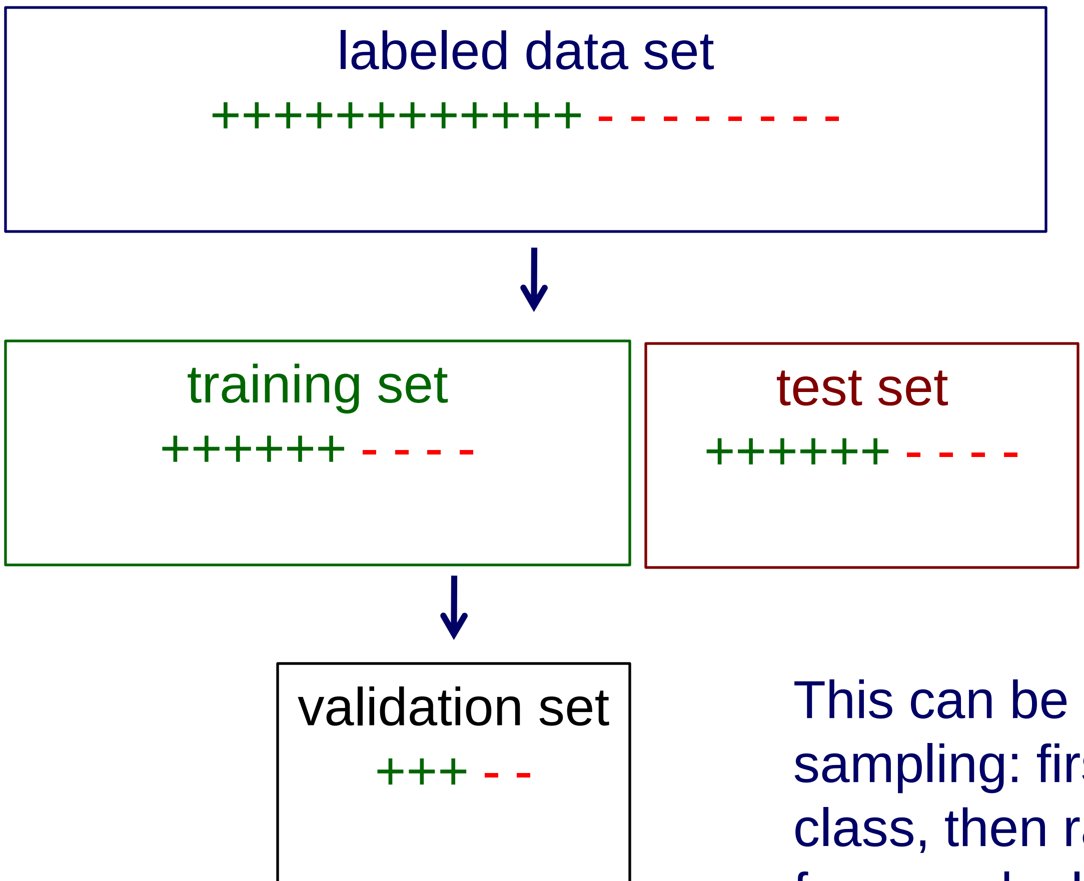
# Random resampling

We can address the second issue by repeatedly randomly partitioning the available data into training and test sets.



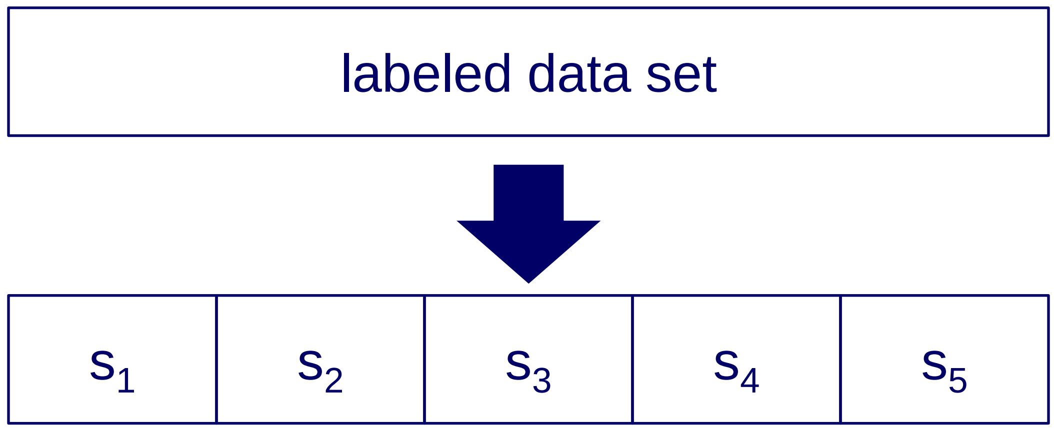
# Stratified sampling

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.

# Cross validation




partition data
into *n* subsamples

iteratively leave one subsample out for the test set, train on the rest

| iteration | train on | test on |
|---|---|---|
| 1 | $s_2$ $s_3$ $s_4$ $s_5$ | $s_1$ |
| 2 | $s_1$ $s_3$ $s_4$ $s_5$ | $s_2$ |
| 3 | $s_1$ $s_2$ $s_4$ $s_5$ | $s_3$ |
| 4 | $s_1$ $s_2$ $s_3$ $s_5$ | $s_4$ |
| 5 | $s_1$ $s_2$ $s_3$ $s_4$ | $s_5$ |

# 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%

# Cross validation

- 10-fold cross validation is common, but smaller values of *n* are often used when learning takes a lot of time
- in *leave-one-out* cross validation, *n* = # instances
- in *stratified* cross validation, stratified sampling is used when partitioning the data
- CV makes efficient use of the available data for testing
- note that whenever we use multiple training sets, as in CV and random resampling, we are evaluating a <u>learning method</u> as opposed to an <u>individual learned hypothesis</u>

# Confusion matrices

How can we understand what types of mistakes a learned model makes?

task: 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

figure from vision.jhu.edu

# Confusion matrix for 2-class problems

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

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

$$\text{error} = 1 - \text{accuracy} = \frac{FP + FN}{TP + FP + FN + TN}$$

# Is accuracy an adequate measure of predictive performance?

accuracy may not be useful measure in cases where

- there is a large class skew
  - Is 98% accuracy good when 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

# Other accuracy metrics

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

# Other accuracy metrics

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

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

# Other accuracy metrics

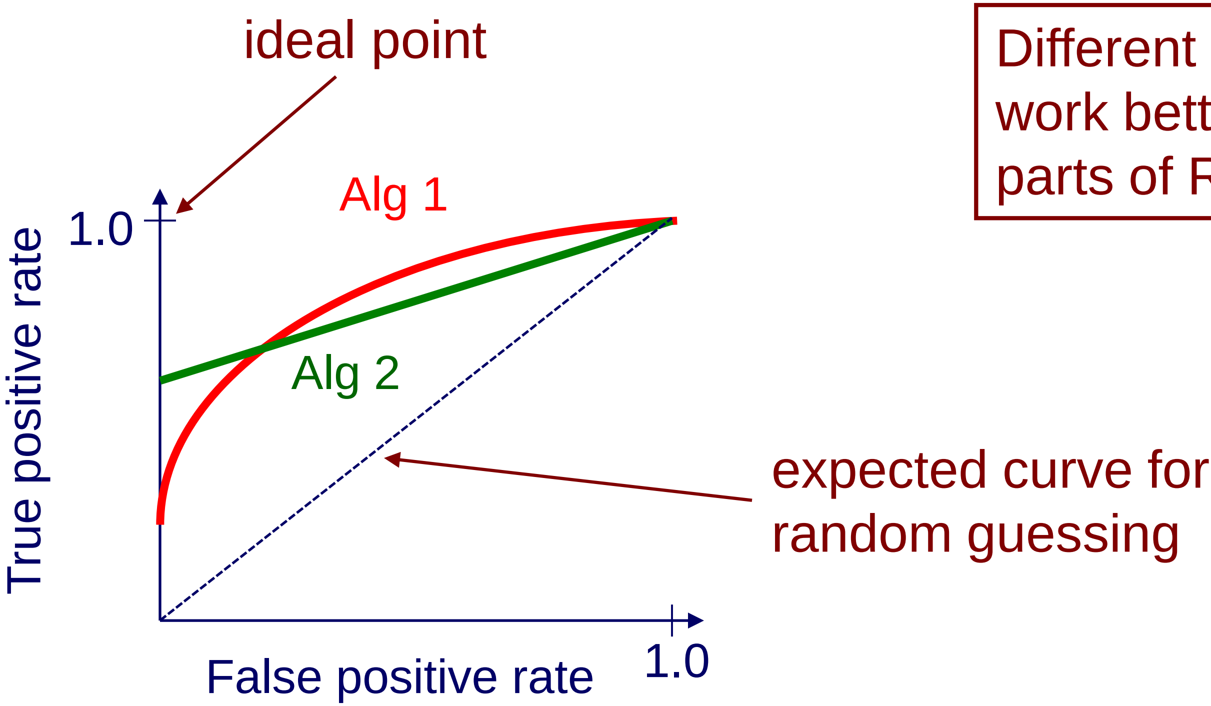
| | | actual class | |
|---|---|---|---|
| | | positive | negative |
| predicted class | positive | true positives (TP) | false positives (FP) |
| | negative | false negatives (FN) | true negatives (TN) |

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

$$\text{false positive rate} = \frac{\mathrm{FP}}{\text{actual neg}} = \frac{\mathrm{FP}}{\mathrm{TN} + \mathrm{FP}}$$

# 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.

# Algorithm for creating an ROC curve

let $\left(\left(y^{(1)},\ c^{(1)}\right)\ldots\ \left(y^{(m)},\ c^{(m)}\right)\right)$ be the test-set instances sorted according to predicted confidence $c^{(i)}$ that each instance is positive

let $num\_neg,\ num\_pos$ be the number of negative/positive instances in the test set

$TP = 0,\ \ FP = 0$

$last\_TP = 0$

for $i = 1$ to $m$

    // find thresholds where there is a pos instance on high side, neg instance on low side

    if ($i$ > 1) and ( $c^{(i)} \neq c^{(i-1)}$ ) and ( $y^{(i)}$ == neg ) and ( $TP > last\_TP$ )

        $FPR = FP\ /\ num\_neg,\ \ \ TPR = TP\ /\ num\_pos$

        output ($FPR,\ TPR$) coordinate

        $last\_TP = TP$

    if $y^{(i)}$ == pos

        $++TP$

    else

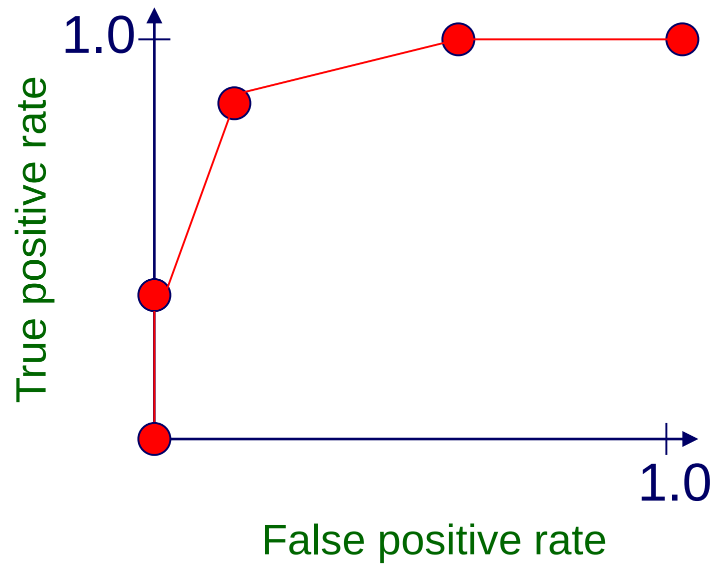
        $++FP$

$FPR = FP\ /\ num\_neg,\ \ TPR = TP\ /\ num\_pos$

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 | | - |
| Ex 2 | .70 | | + |
| Ex 6 | .65 | TPR= 4/5, FPR= 1/5 | + |
| Ex 10 | .51 | | - |
| Ex 3 | .39 | | - |
| Ex 5 | .24 | TPR= 5/5, FPR= 3/5 | + |
| Ex 4 | .11 | | - |
| Ex 8 | .01 | TPR= 5/5, FPR= 5/5 | - |



# ROC curve example

task: recognizing genomic units called operons

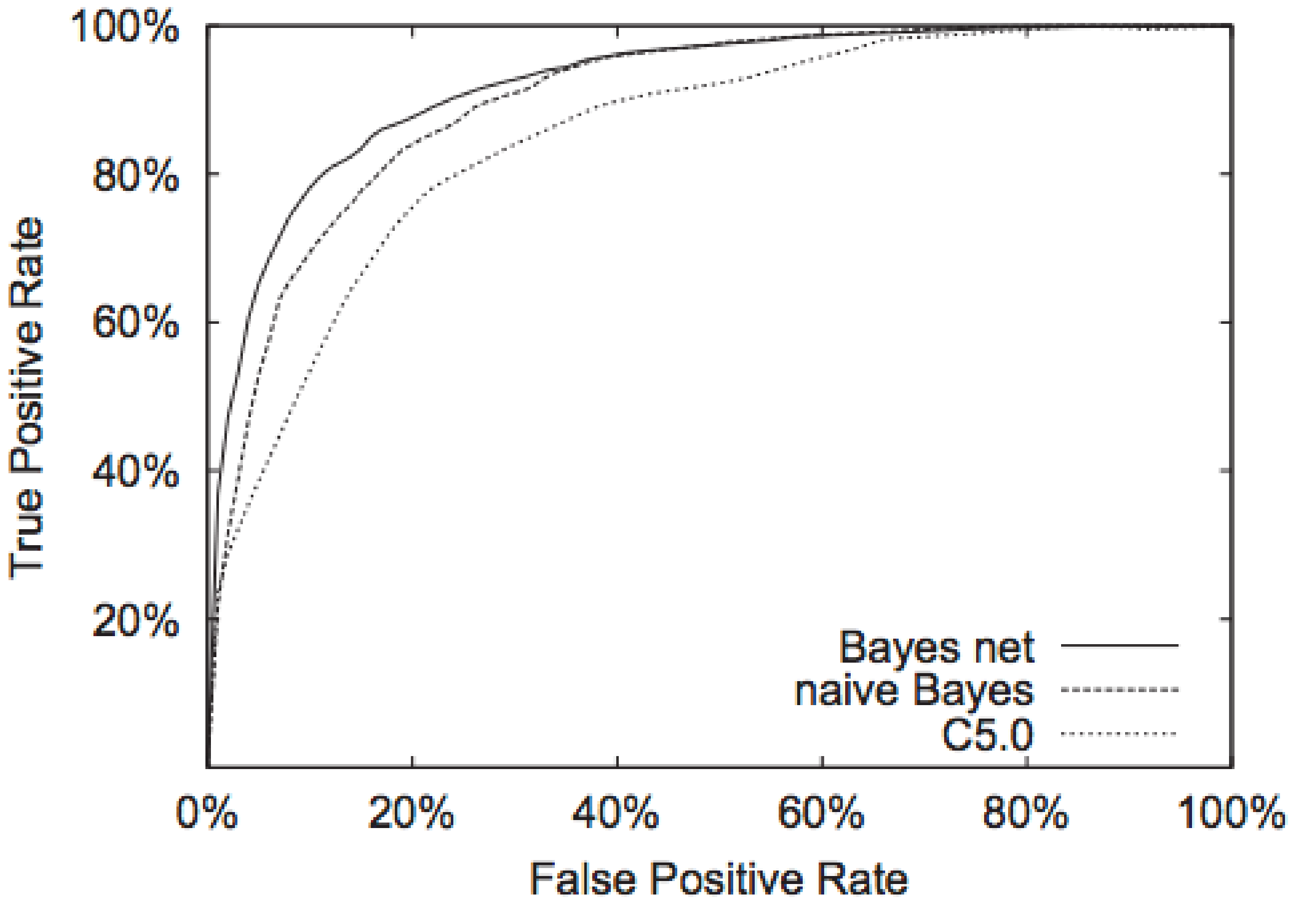
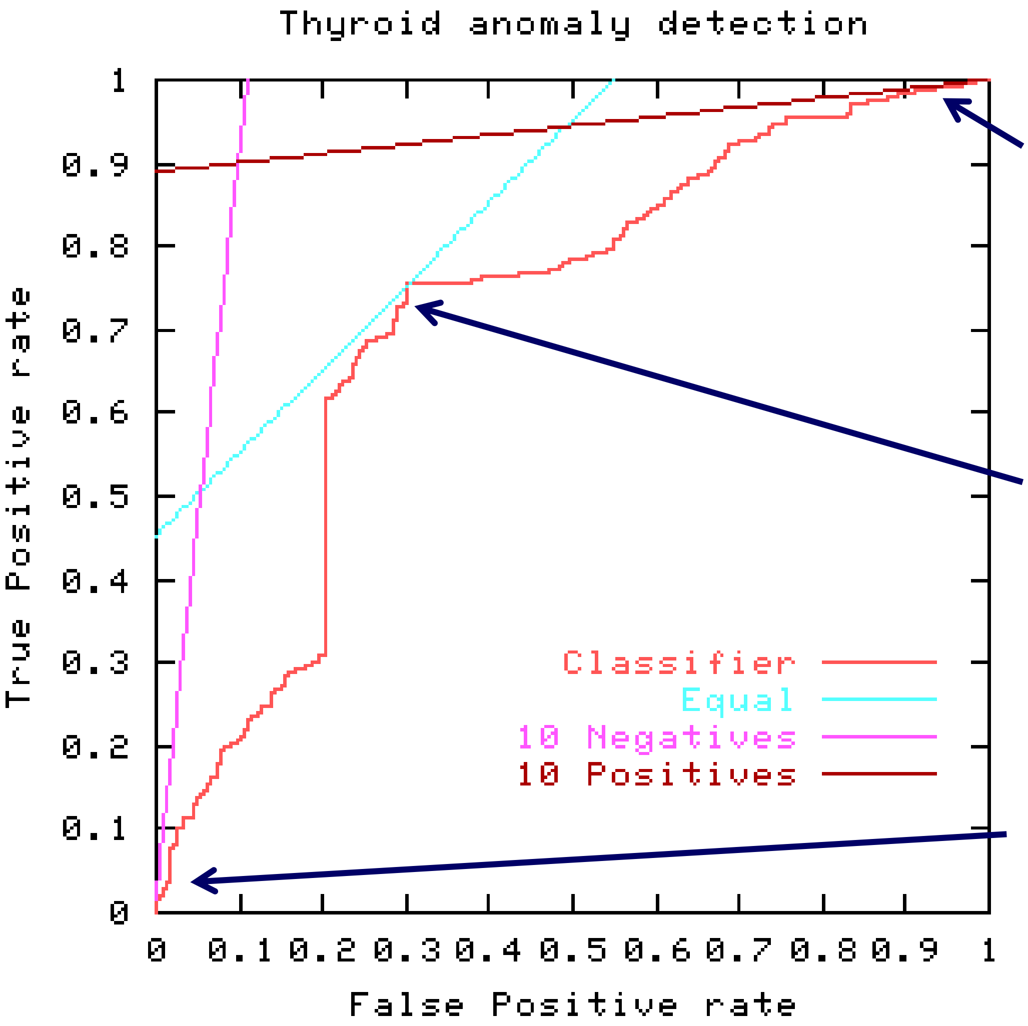



figure from Bockhorst et al., *Bioinformatics* 2003

# ROC curves and misclassification costs

The best operating point depends on the relative costs of FN and FP misclassifications




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