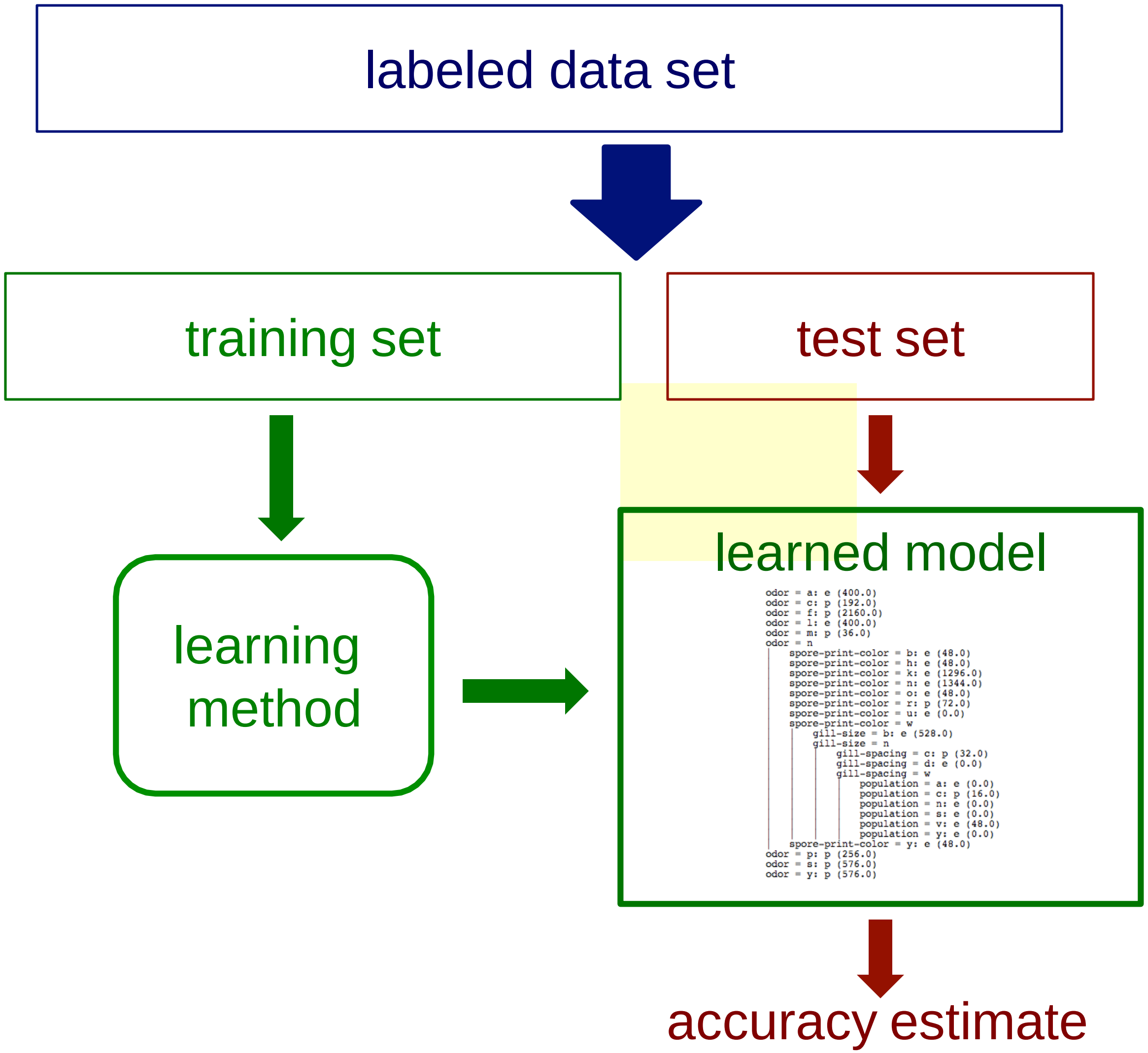
Test sets revisited

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



Which Classifier is better?

Almost as many answers as there are performance measures! (e.g., UCI Breast

Algo	Acc	RMSE	TPR	FPR	Prec	Rec	F	AUC	Info S
NB	71.7	.4534	.44	.16	.53	.44	.48	.7	48.11
C4.5	75.5	.4324	.27	.04	.74	.27	.4	.59	34.28
3NN	72.4	.5101	.32	.1	.56	.32	.41	.63	43.37
Ripp	71	.4494	.37	.14	.52	.37	.43	.6	22.34
SVM	69.6	.5515	.33	.15	.48	.33	.39	.59	54.89
Bagg	67.8	.4518	.17	.1	.4	.17	.23	.63	11.30
Boost	70.3	.4329	.42	.18	.5	.42	.46	.7	34.48
RanF	69.23	.47	.33	.15	.48	.33	.39	.63	20.78

https://www.icmla-conference.org › PE_Tutorial

Accuracy

		Expected	
Predicted		Pos	Neg
	Yes	TP	FP
	No	FN	TN
		P=TP+FN	N=FP+TN

- Accuracy = (TP+TN)/(P+N)
- Precision = TP/(TP+FP)
- Recall/TP rate = TP/P
- Specificity = TN/N
- FP Rate = FP/N = 1-Specificity
- F-measure = 2*Precision*Recall /(Precision+ Recall)

https://www.site.uottawa.ca › Courses › csi5387_2009 › ML_Evaluation1

What's wrong with Accuracy?

True class →	Pos	Neg
Yes	200	100
No	300	400
	P=500	N=500

True class →	Pos	Neg
Yes	400	300
No	100	200
	P=500	N=500

- Both classifiers obtain 60% accuracy
- They exhibit very different behaviours:
  - On the left: weak positive recognition rate/strong negative recognition rate
  - On the right: strong positive recognition rate/weak negative recognition rate

https://www.site.uottawa.ca › Courses › csi5387_2009 › ML_Evaluation1

What's wrong with Precision/Recall?

True class →	Pos	Neg
Yes	200	100
No	300	400
	P=500	N=500

True class →	Pos	Neg
Yes	200	100
No	300	0
	P=500	N=100

- Both classifiers obtain the same precision and recall values of 66.7% and 40%
- They exhibit very different behaviours:
  - Same positive recognition rate
  - Extremely different negative recognition rate: strong on the left / nil on the right
- Note: Accuracy has no problem catching this!

https://www.site.uottawa.ca › Courses › csi5387_2009 › ML_Evaluation1

So what can be done?

- Think about evaluation carefully prior to starting your experiments.
- Use performance measures other than accuracy and precision recall. E.g., ROC Analysis, combinations of measures. Also, think about the best measure for your problem.
- Use re-sampling methods other than cross-validation, when necessary: bootstrapping? Randomization?
- Use statistical tests other than the t-test: non-parametric tests; tests appropriate for many classifiers compared on many domains (the t-test is not appropriate for this case, which is the most common one).

➢ We will try to discuss some of these issues next time.

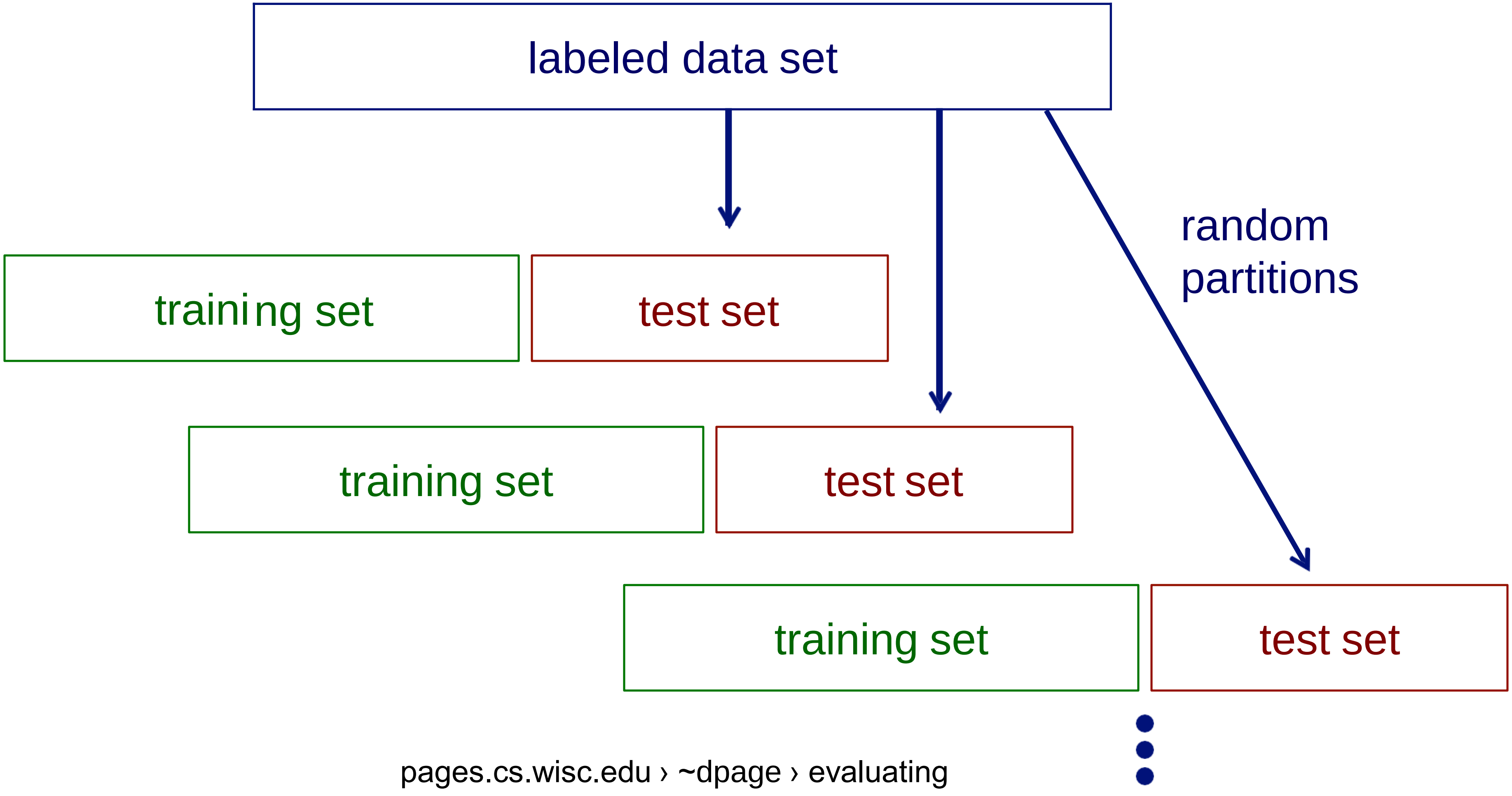
Limitations of using a single training/test partition

- we may not have enough data to make sufficiently large training and test sets
  - a larger test set gives us more reliable estimate of accuracy (i.e. a lower variance estimate)
  - but… a larger training set 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

Random resampling

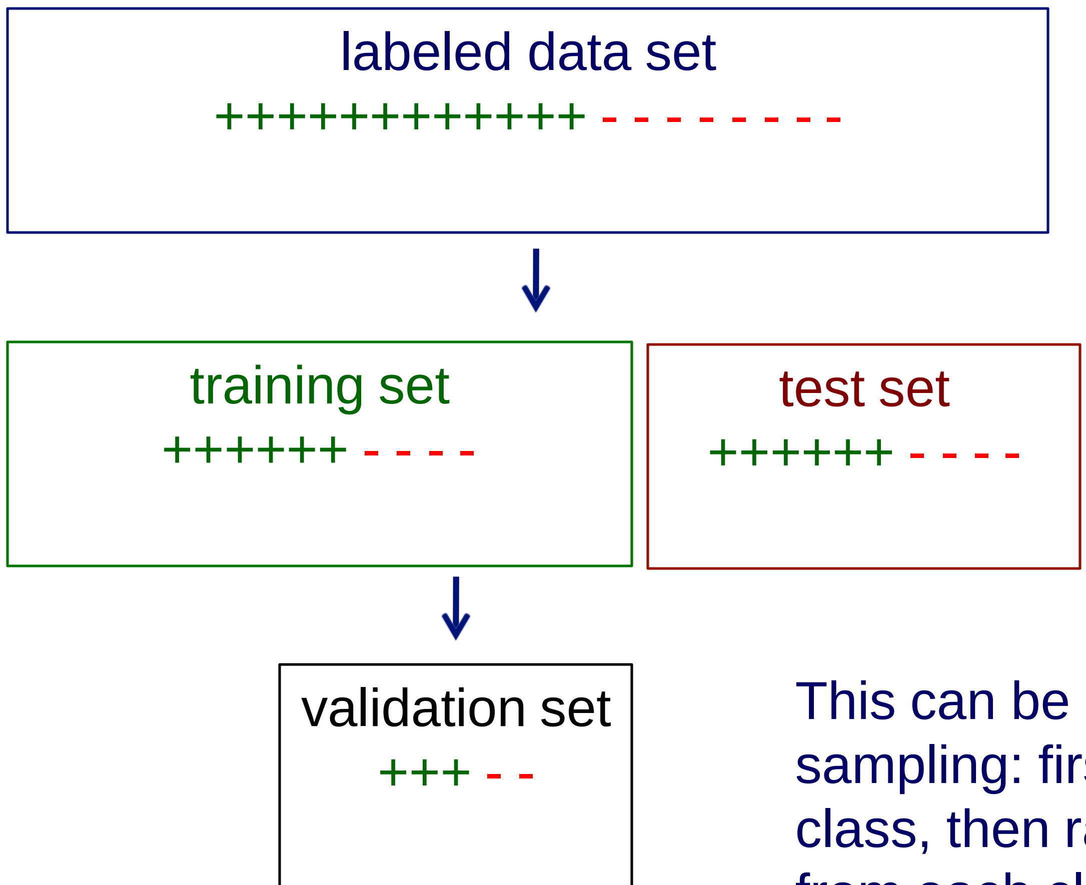
We can address the second issue by repeatedly randomly partitioning the available data into training and set sets.



pages.cs.wisc.edu › ~dpage › evaluating

Stratified sampling

When randomly selecting training or validation sets, we may want to ensure that class proportions are maintained in each selected set

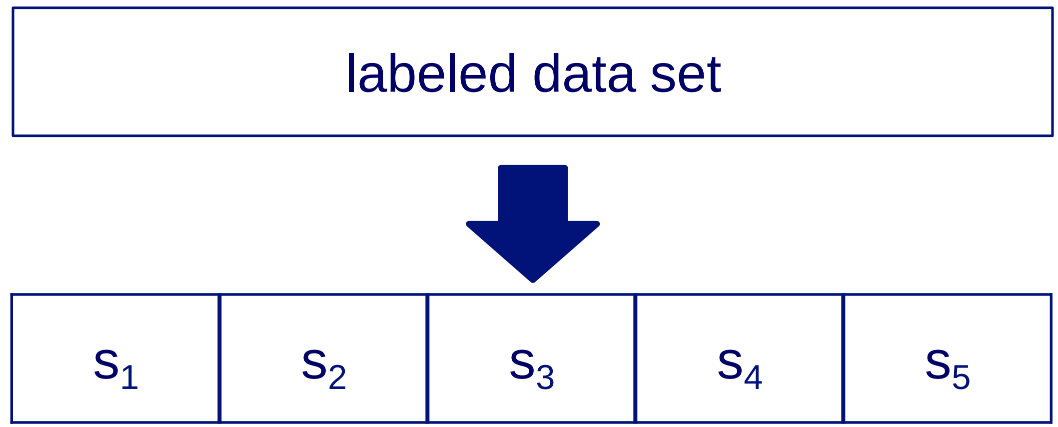


This can be done via stratified sampling: first stratify instances by class, then randomly select instances from each class proportionally.

pages.cs.wisc.edu › ~dpage › evaluating

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

pages.cs.wisc.edu › ~dpage › evaluating

k-Fold Cross-Validation

- 5-fold cross validation.

Iteration 1	Test	Train	Train	Train	Train
Iteration 2	Train	Test	Train	Train	Train
Iteration 3	Train	Train	Test	Train	Train
Iteration 4	Train	Train	Train	Test	Train
Iteration 5	Train	Train	Train	Train	Test

https://miro.medium.com/max/2736/1*rgba1BIOUys7wQcXcL4U5A.png

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%

pages.cs.wisc.edu › ~dpage › evaluating

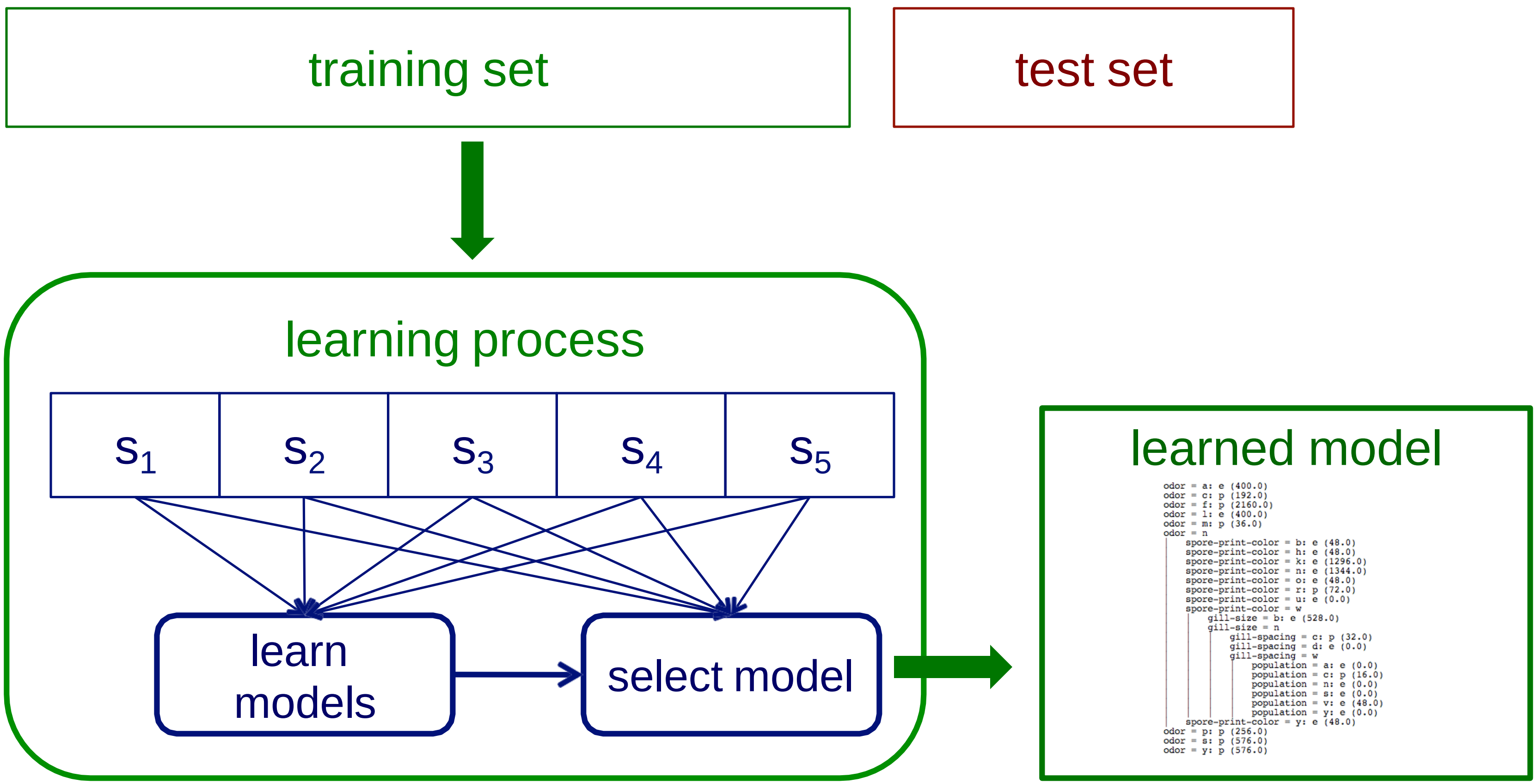
Cross validation

- 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 model</u>

pages.cs.wisc.edu › ~dpage › evaluating

Internal cross validation

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 level of decision-tree pruning)



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%

ROC (Receiver Operating Characteristic)

http://en.wikipedia.org/wiki/Receiver_operating_characteristic

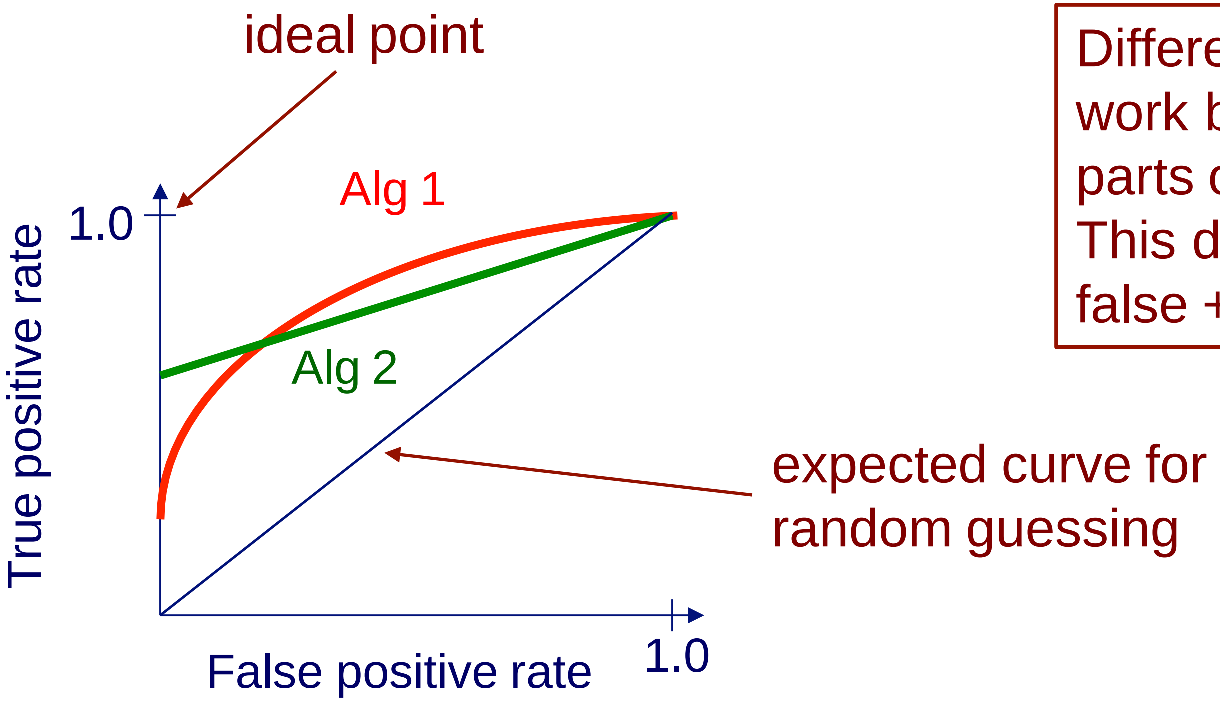
- Developed in 1950s for signal detection theory to analyze noisy signals
  - Characterize the trade-off between positive hits and false alarms
- ROC curve plots TP (on the y-axis) against FP (on the x-axis)
- Performance of each classifier represented as a point on the ROC curve
  - changing the threshold of algorithm, sample distribution or cost matrix changes the location of the point

http://www.childrensmercy.org/stats/ask/roc.asp

www2.cs.uh.edu › ~ceick › Topic11

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

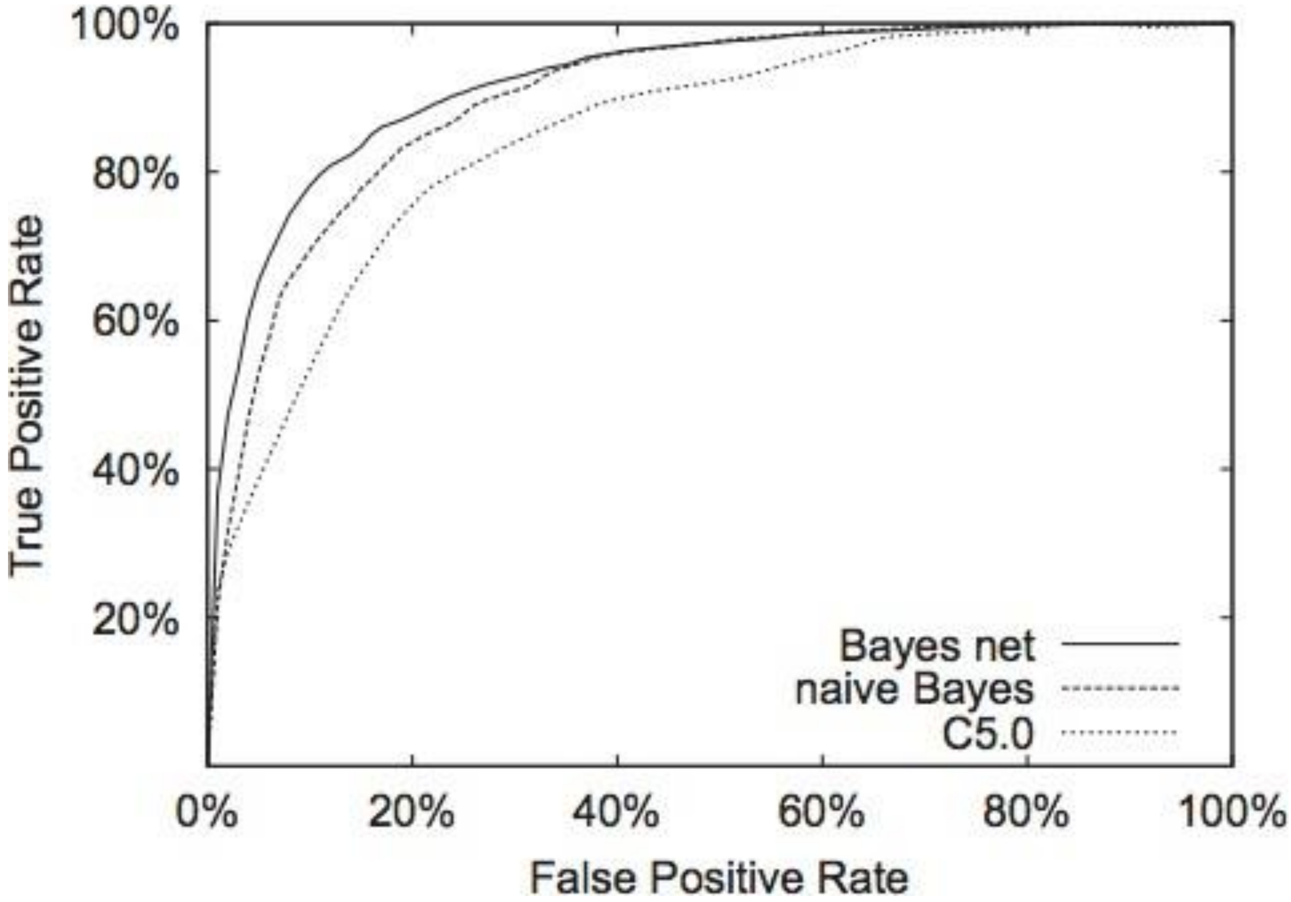
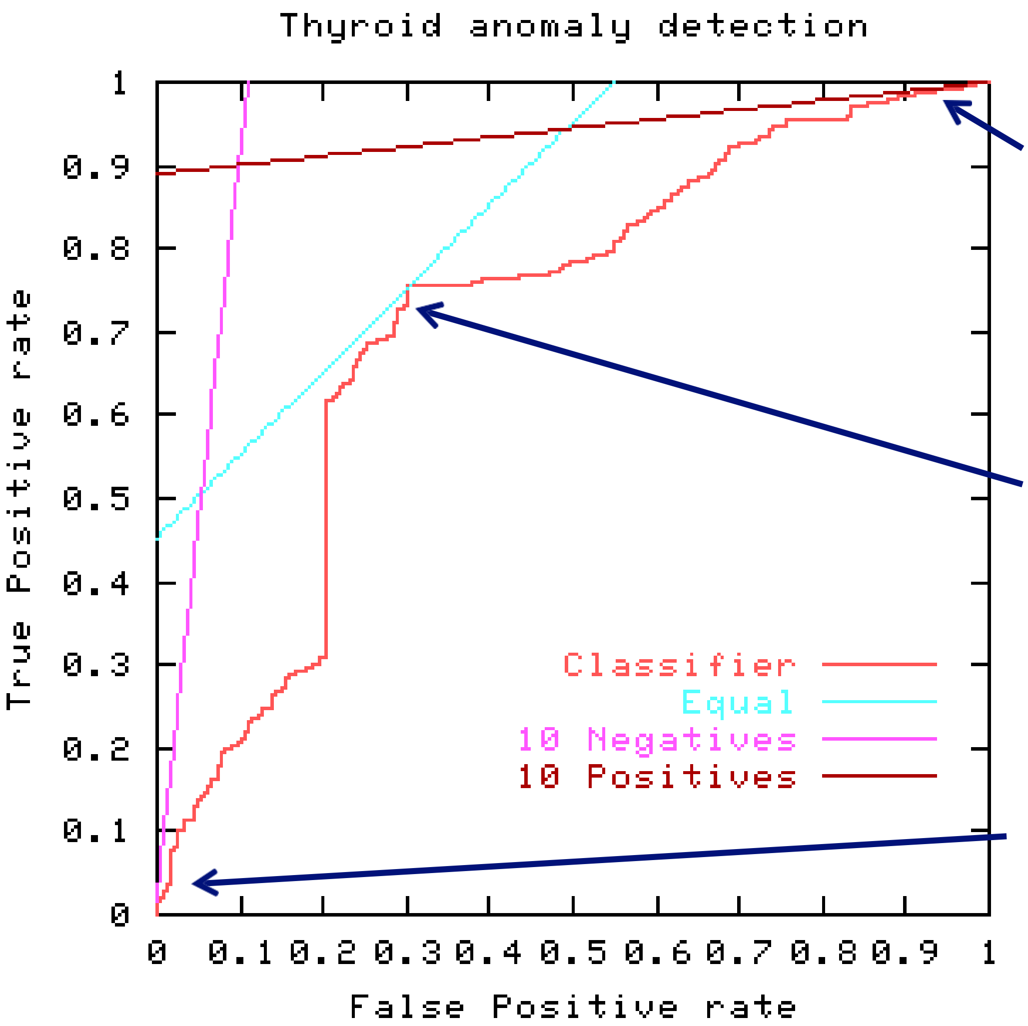


figure from Bockhorst et al., *Bioinformatics* 2003

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

Algorithm for creating an ROC curve

1. sort test-set predictions according to confidence that each instance is positive

2. step through sorted list from high to low confidence

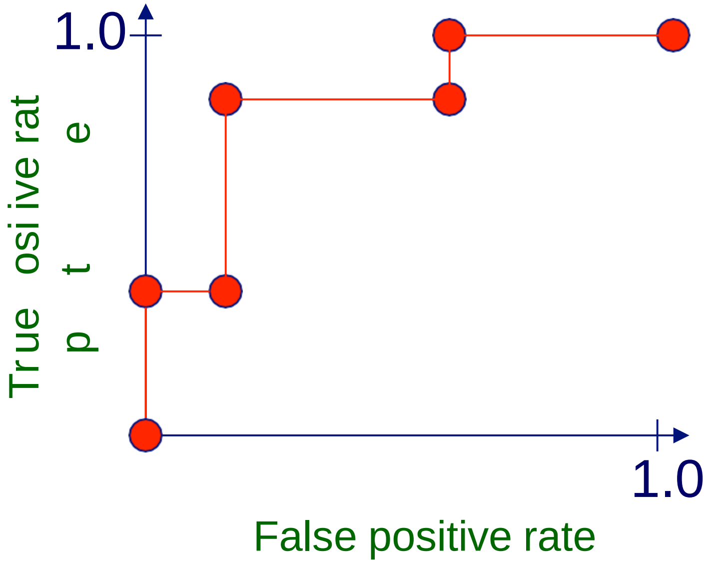
    i. locate a *threshold* between instances with opposite classes (keeping instances with the same confidence value on the same side of threshold)

    ii. compute TPR, FPR for instances above threshold

    iii. output (FPR, TPR) coordinate

Plotting an ROC curve

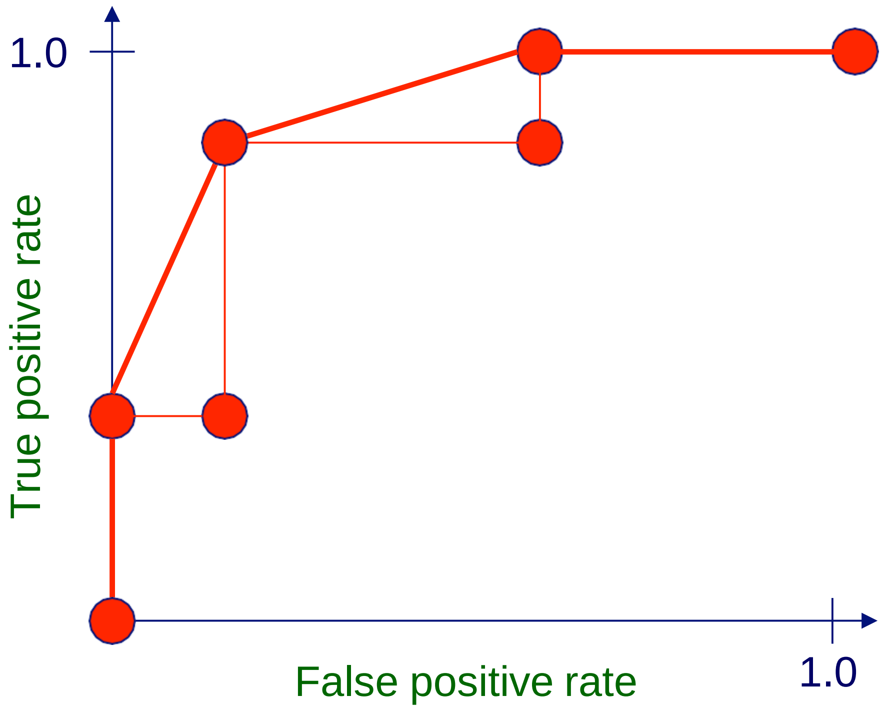
instance	confidence positive		correct class
Ex 9	.99		+
Ex 7	.98	TPR= 2/5, FPR= 0/5	+
Ex 1	.72	TPR= 2/5, FPR= 1/5	-
Ex 2	.70		+
Ex 6	.65	TPR= 4/5, FPR= 1/5	+
Ex 10	.51		-
Ex 3	.39	TPR= 4/5, FPR= 3/5	-
Ex 5	.24	TPR= 5/5, FPR= 3/5	+
Ex 4	.11		-
Ex 8	.01	TPR= 5/5, FPR= 5/5	-



Plotting an ROC curve

can interpolate between points to get *convex hull*

- convex hull: repeatedly, while possible, perform interpolations that skip one data point and discard any point that lies below a line
- interpolated points are achievable in theory: can flip weighted coin to choose between classifiers represented by plotted points



ROC curves

Does a low false-positive rate indicate that most positive predictions (i.e. predictions with confidence > some threshold) are correct?

suppose our TPR is 0.9, and FPR is 0.01

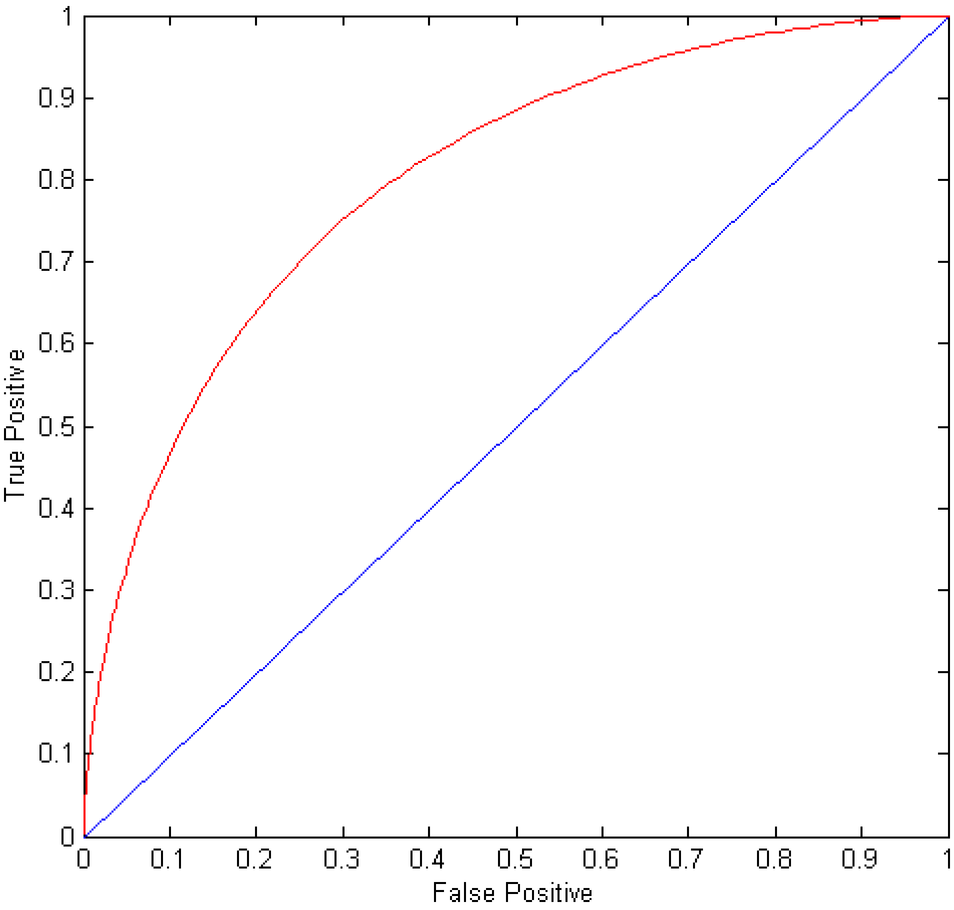
fraction of instances that are positive	fraction of positive predictions that are correct
0.5	0.989
0.1	0.909
0.01	0.476
0.001	0.083

ROC Curve

(TP,FP):

- (0,0): declare everything
  to be negative class
- (1,1): declare everything
  to be positive class
- (1,0): ideal

- Diagonal line:
  - Random guessing
  - Below diagonal line:
  prediction is opposite of the true class



www2.cs.uh.edu › ~ceick › Topic11

How to Construct an ROC curve

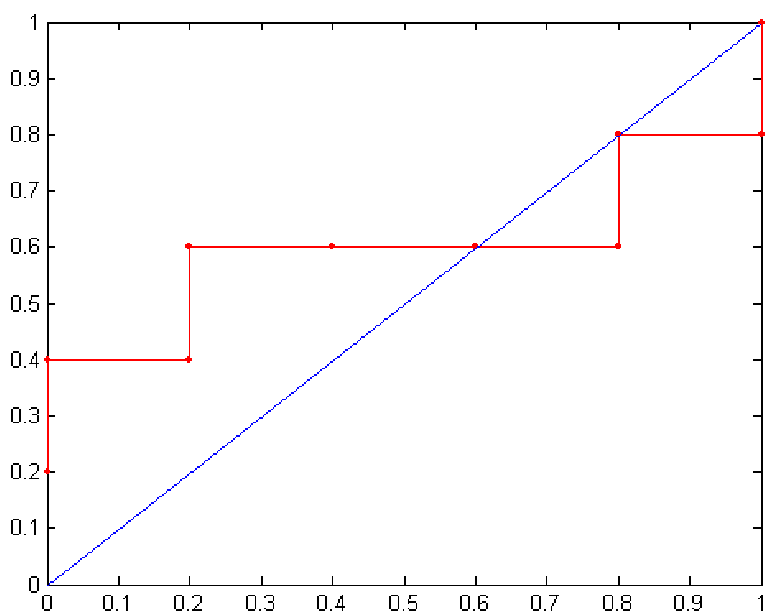
Instance	P(+|A)	True Class
1	0.95	+
2	0.93	+
3	0.87	-
4	0.85	-
5	0.85	-
6	0.85	+
7	0.76	-
8	0.53	+
9	0.43	-
10	0.25	+

- Use classifier that produces posterior probability for each test instance P(+|A)
- Sort the instances according to P(+|A) in decreasing order
- Apply threshold at each unique value of P(+|A)
- Count the number of TP, FP, TN, FN at each threshold
- TP rate, TPR = TP/(TP+FN)
- FP rate, FPR = FP/(FP + TN)

How to construct an ROC curve

Class	+	-	+	-	-	-	+	-	+	+	
Threshold >=	0.25	0.43	0.53	0.76	0.85	0.85	0.85	0.87	0.93	0.95	1.00
TP	5	4	4	3	3	3	3	2	2	1	0
FP	5	5	4	4	3	2	1	1	0	0	0
TN	0	0	1	1	2	3	4	4	5	5	5
FN	0	1	1	2	2	2	2	3	3	4	5
→ TPR	1	0.8	0.8	0.6	0.6	0.6	0.6	0.4	0.4	0.2	0
→ FPR	1	1	0.8	0.8	0.6	0.4	0.2	0.2	0	0	0

ROC Curve:



Reverse of above order

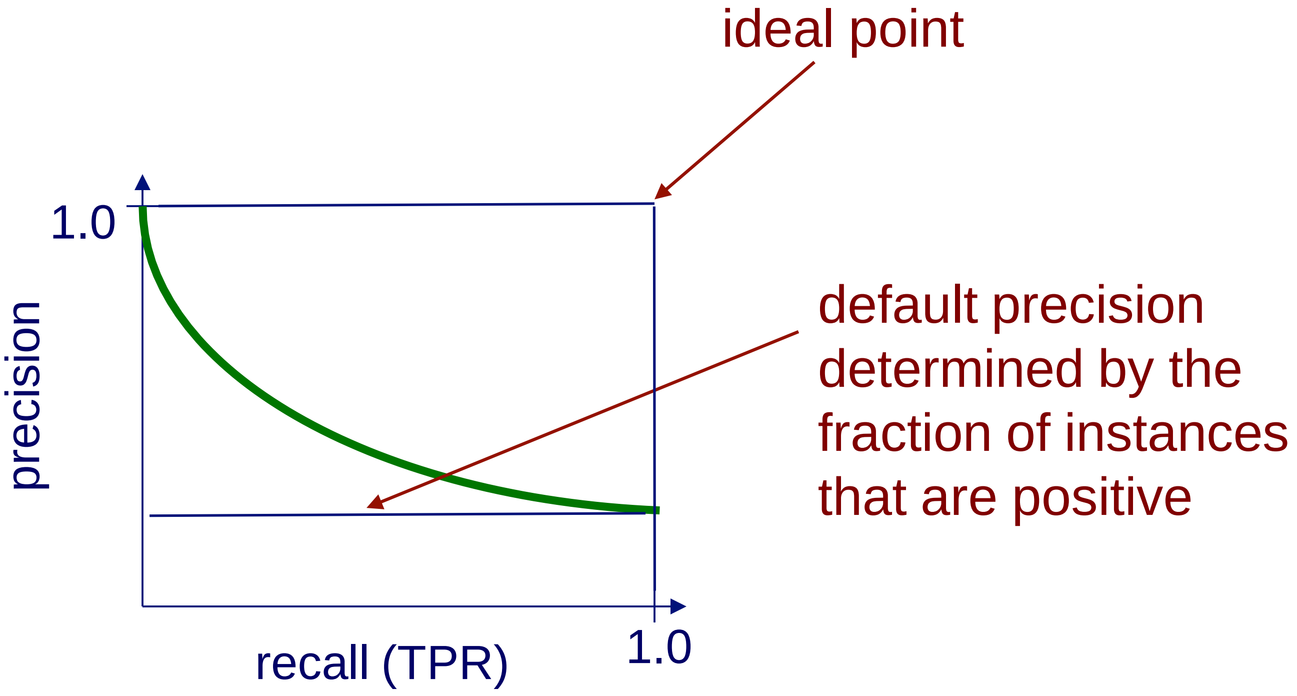
+ + - + - - - + - +

↑ +

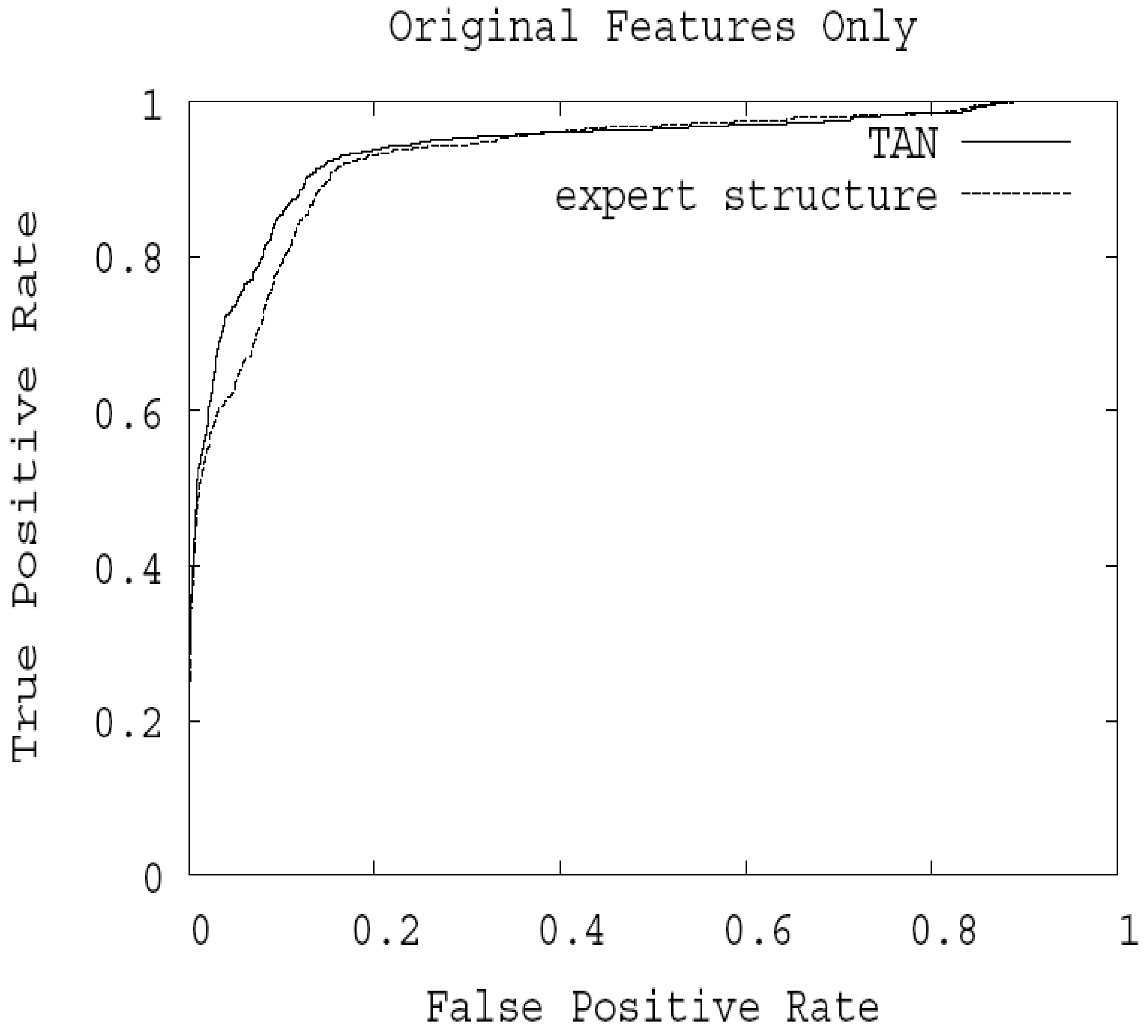
→

Precision/recall curves

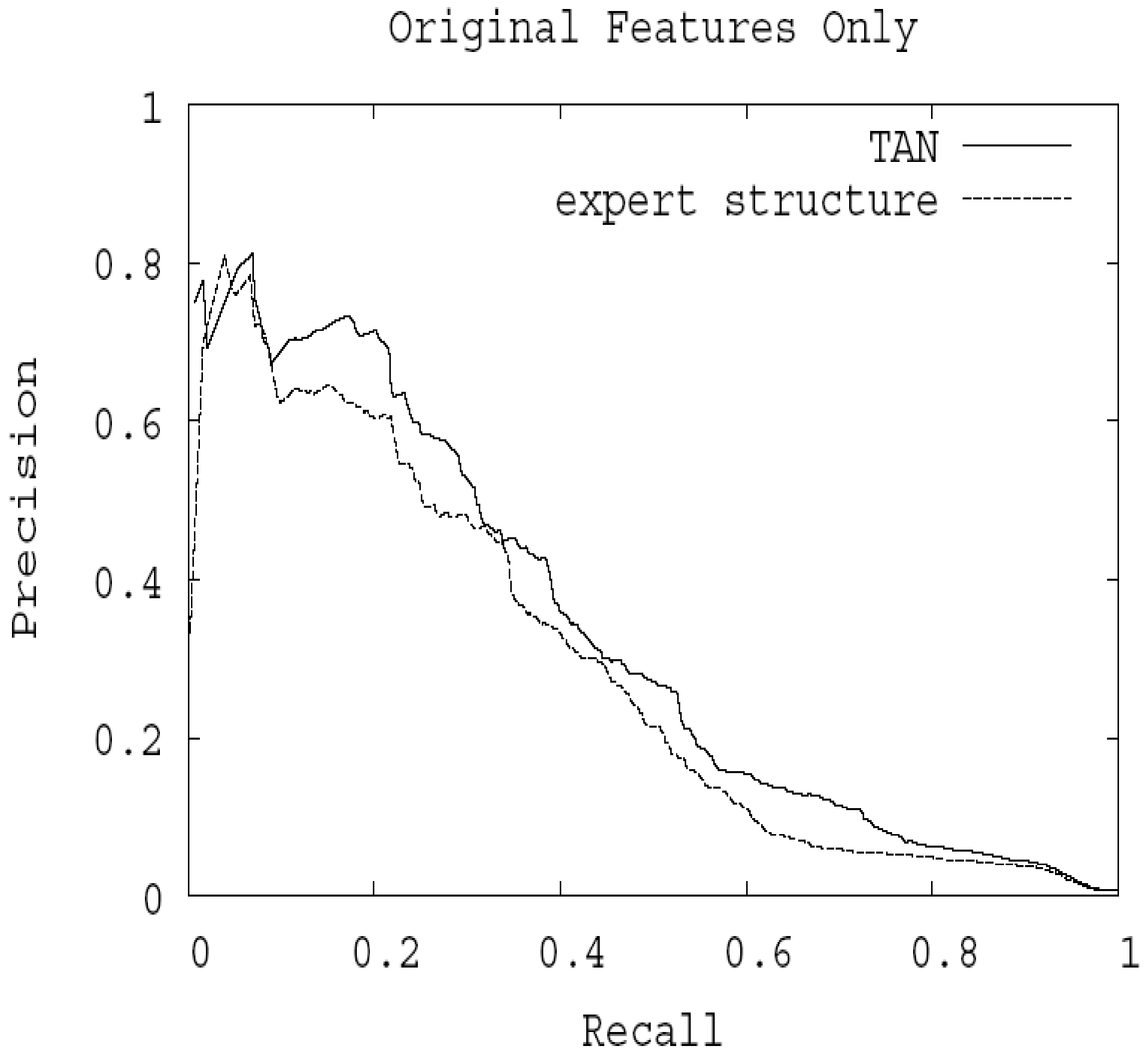
A *precision/recall curve* plots the precision vs. recall (TP-rate) as a threshold on the confidence of an instance being positive is varied



Mammography Example: ROC



Mammography Example: PR



How do we get one ROC/PR curve when we do 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

Comments on ROC and PR curves

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 <u>the fraction of predictions</u> that are false positives
- well suited for tasks with lots of negative instances

RMSE

- The Root-Mean Squared Error (RMSE) is usually used for regression, but can also be used with probabilistic classifiers. The formula for the RMSE is:

$$RMSE(f) = sqrt(1/m \Sigma_{i=1}^{m}(f(x_i) - y_i)^2))$$

where m is the number of test examples, $f(x_i)$, the classifier's probabilistic output on x_i and y_i theactual label.

ID	$f(x_i)$	y_i	$(f(x_i) - y_i)^2$
1	.95	1	.0025
2	.6	0	.36
3	.8	1	.04
4	.75	0	.5625
5	.9	1	.01

RMSE(f) = sqrt(1/5 * (.0025+.36+.04+.5625+.01))
= sqrt(0.975/5) = 0.4416

Information Score

- Kononenko and Bratko's Information Score assumes a prior P(y) on the labels. This could be estimated by the class distribution of the training data. The output (posterior probability) of a probabilistic classifier is P(y|f), where f is the classifier. I(a) is the indicator fonction.
- $IS(x) = I(P(y|f) \geq P(y)) * (-\log(P(y))+\log(P(y|f)) +$
  $+ I(P(y|f) < P(y)) * (-\log(1-P(y))+\log(1-P(y|f)))$
- $IS_{avg} = 1/m \, \Sigma_{i=1}^{m}(IS(x_i))$

x	$P(y_i\|f)$	y_i	IS(x)
1	.95	1	0.66
2	.6	0	0
3	.8	1	.42
4	.75	0	.32
5	.9	1	.59

$P(y=1) = 3/5 = 0.6$
$P(y=0) = 2/5 = 0.4$
$IS(x_1) = 1 * (-\log(.6) + \log(.95)) +$
$0 * (-\log(.4) + \log).05))$
$= 0.66$

$IS_{avg} = 1/5\ (0.66+0+.42+.32+.59)$
$= 0.40$