The many faces of ROC analysis in machine learning

Peter A. Flach

University of Bristol, UK

www.cs.bris.ac.uk/~flach/

Objectives

- After this tutorial, you will be able to
  - **[model evaluation]** produce ROC plots for categorical and ranking classifiers and calculate their AUC; apply cross-validation in doing so;
  - **[model selection]** use the ROC convex hull method to select among categorical classifiers; determine the optimal decision threshold for a ranking classifier (calibration);
  - **[metrics]** analyse a variety of machine learning metrics by means of ROC isometrics; understand fundamental properties such as skew-sensitivity and equivalence between metrics;
  - **[model construction]** appreciate that one model can be many models from a ROC perspective; use ROC analysis to improve a model's AUC;
  - **[multi-class ROC]** understand multi-class approximations such as the MAUC metric and calibration of multi-class probability estimators; appreciate the main open problems in extending ROC analysis to multi-class classification.

Take-home messages

- It is almost always a good idea to distinguish performance between classes.
- ROC analysis is not just about ‘cost-sensitive learning’.
- Ranking is a more fundamental notion than classification.

Outline

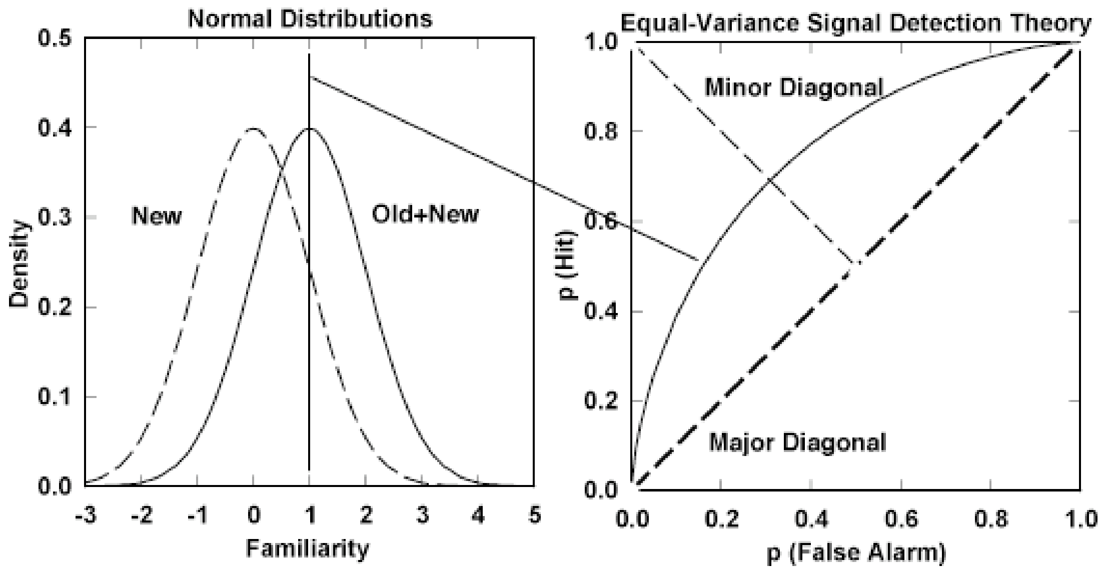
- Part I: Fundamentals (90 minutes)
    - categorical classification: ROC plots, random selection between models, the ROC convex hull, iso-accuracy lines
    - ranking: ROC curves, the AUC metric, turning rankers into classifiers, calibration, averaging
    - interpretation: concavities, majority class performance
    - alternatives: PN plots, precision-recall curves, DET curves, cost curves
- Part II: A broader view (60 minutes)
    - understanding ML metrics: isometrics, basic types of linear isometric plots, linear metrics and equivalences between them, non-linear metrics, skew-sensitivity
    - model manipulation: obtaining new models without re-training, ordering decision tree branches and rules,repairing concavities, locally adjusting rankings
- Part III: Multi-class ROC (30 minutes)
    - the general problem, multi-objective optimisation and the Pareto front, approximations to Area Under ROC Surface, calibrating multi-class probability estimators

Part I: Fundamentals

- Categorical classification:
  - ROC plots
  - random selection between models
  - the ROC convex hull
  - iso-accuracy lines
- Ranking:
  - ROC curves
  - the AUC metric
  - turning rankers into classifiers
  - calibration
- Alternatives:
  - PN plots
  - precision-recall curves
  - DET curves
  - cost curves

Receiver Operating Characteristic

- Originated from signal detection theory
  - binary signal corrupted by Gaussian noise
  - how to set the threshold (operating point) to distinguish between presence/absence of signal?
  - depends on (1) strength of signal, (2) noise variance, and (3) desired hit rate or false alarm rate



from http://wise.cgu.edu/sdt/

Signal detection theory

- slope of ROC curve is equal to likelihood ratio

$$L(x) = \frac{P(x \mid \text{signal})}{P(x \mid \text{noise})}$$

- if variances are equal, L(x) increases monotonically with x and ROC curve is convex
  - optimal threshold for x_0 such that $L(x_0) = \frac{P(\text{noise})}{P(\text{signal})}$
- concavities occur with unequal variances

ROC analysis for classification

- Based on contingency table or confusion matrix

	Predicted positive	Predicted negative
Positive examples	**True positives**	**False negatives**
Negative examples	**False positives**	**True negatives**

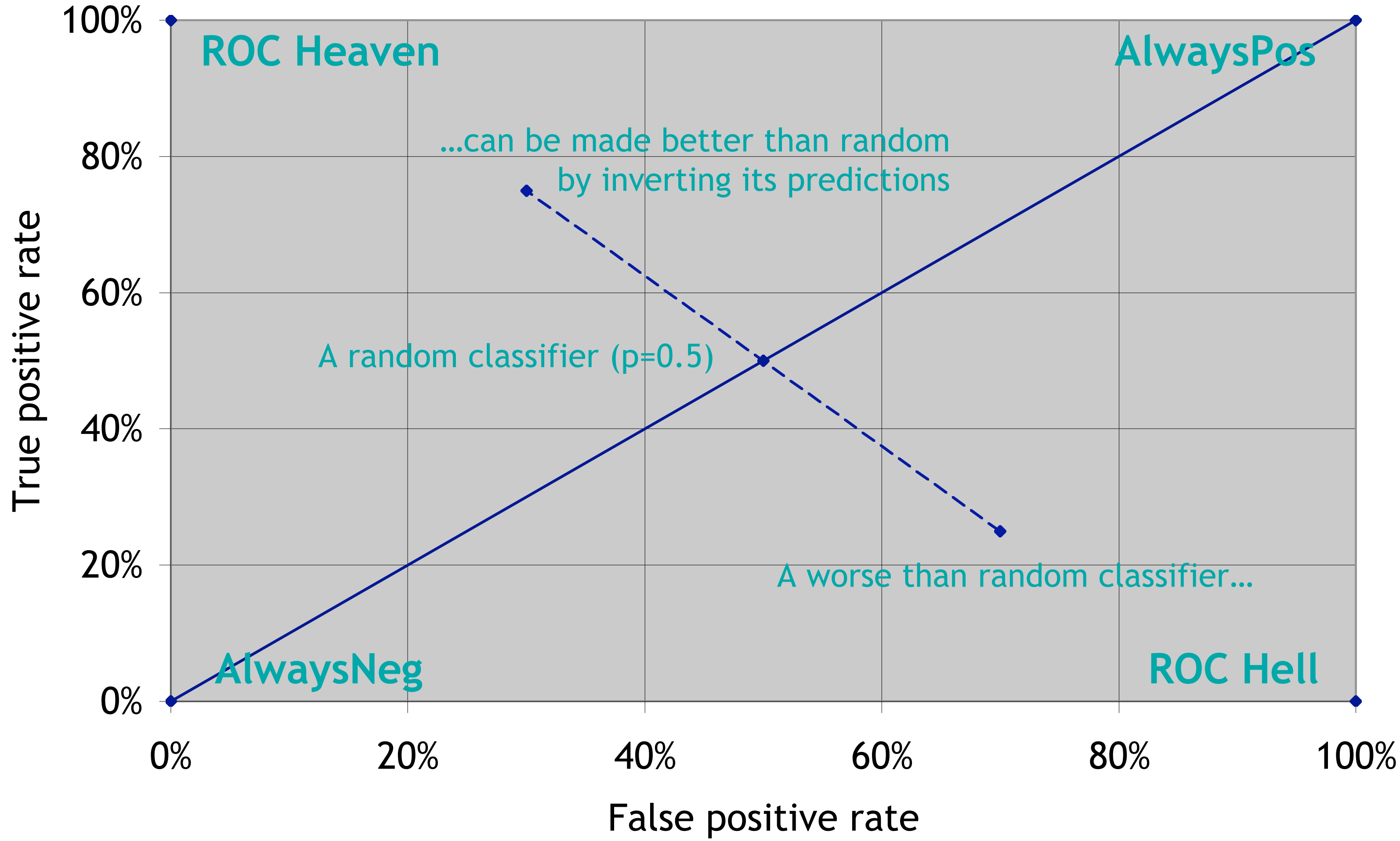
- Terminology:
  - true positive = hit
  - true negative = correct rejection
  - false positive = false alarm (aka Type I error)
  - false negative = miss (aka Type II error)
    - positive/negative refers to prediction
    - true/false refers to correctness

More terminology & notation

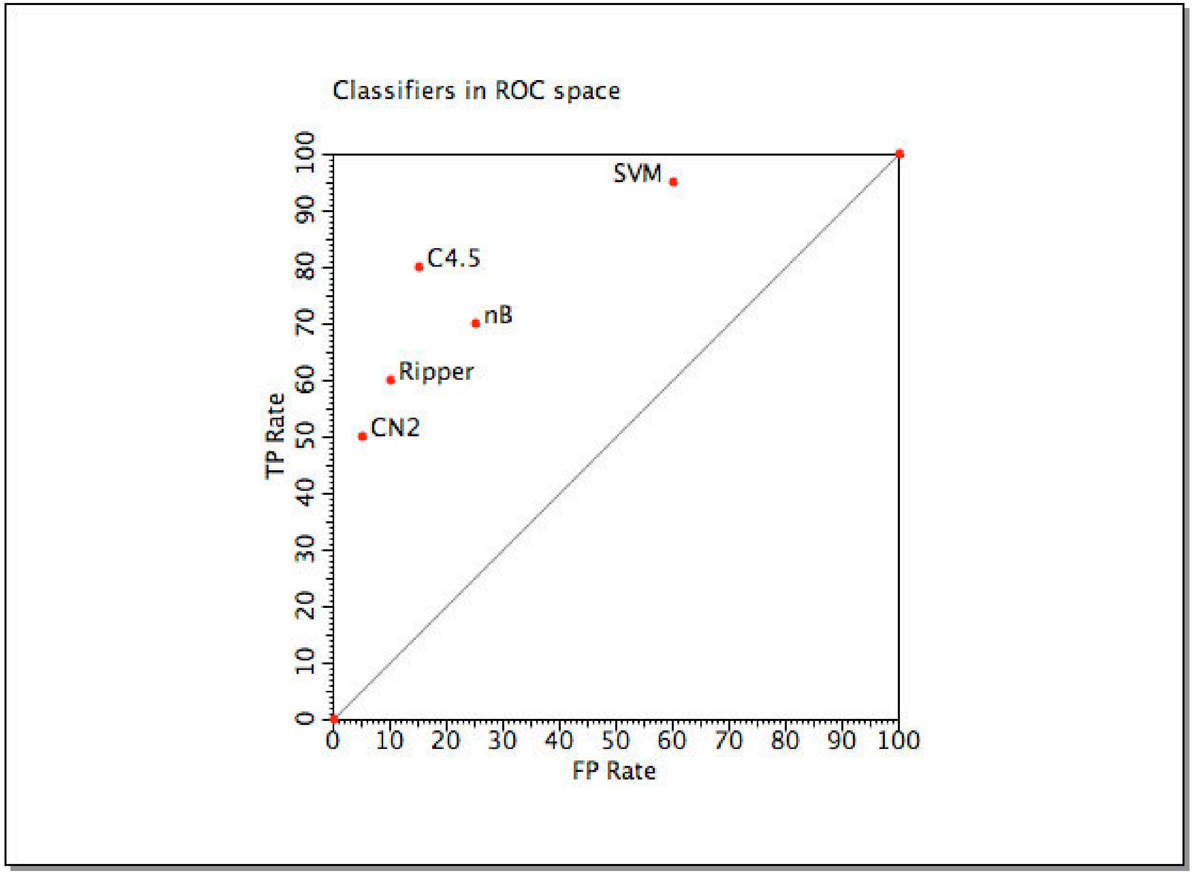
	Predicted positive	Predicted negative	
Positive examples	**TP**	**FN**	**Pos**
Negative examples	**FP**	**TN**	**Neg**
	PPos	**PNeg**	**N**

- True positive rate tpr = TP/Pos = TP/TP+FN
  - fraction of positives correctly predicted
- False positive rate fpr = FP/Neg = FP/FP+TN
  - fraction of negatives incorrectly predicted
  - = 1 – true negative rate TN/FP+TN
- Accuracy acc = pos*tpr + neg*(1–fpr)
  - weighted average of true positive and true negative rates

A closer look at ROC space

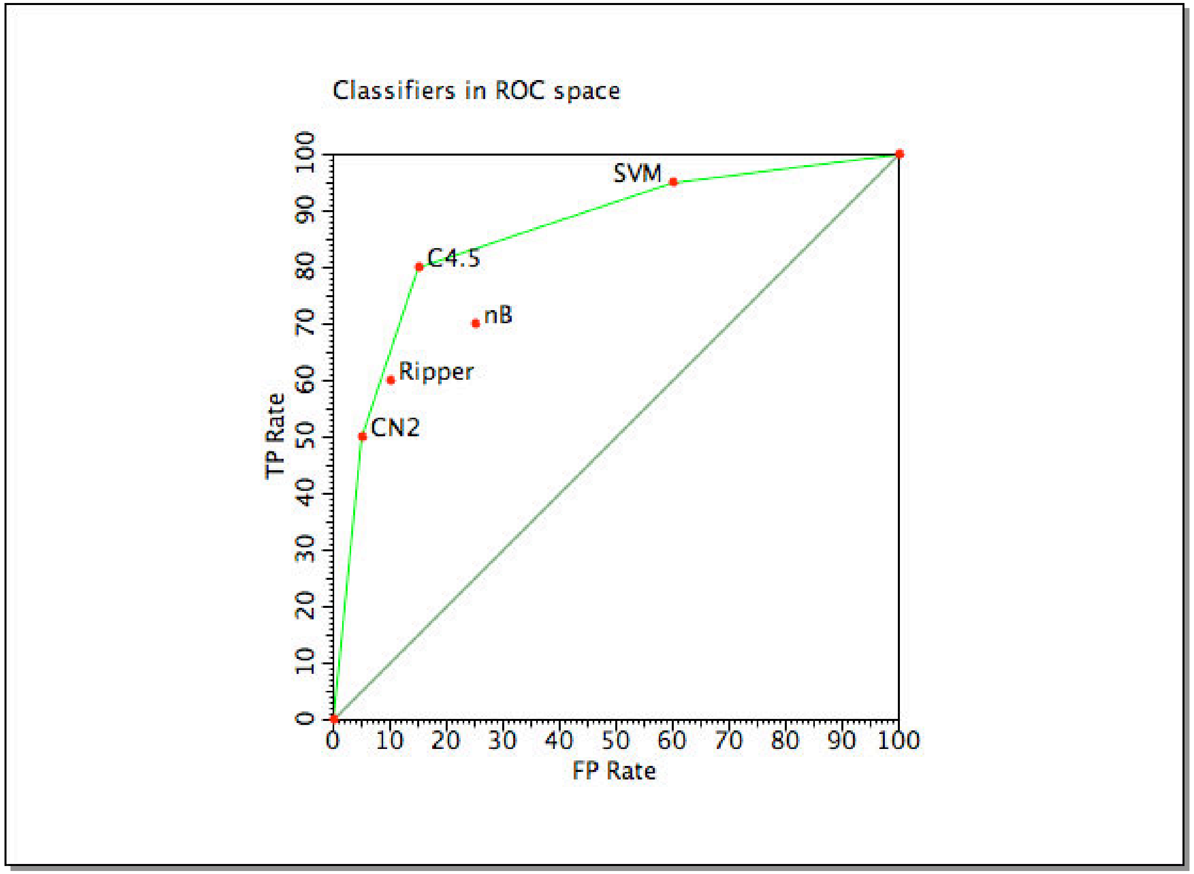


Example ROC plot



ROC plot produced by ROCon (http://www.cs.bris.ac.uk/Research/MachineLearning/rocon/)

The ROC convex hull



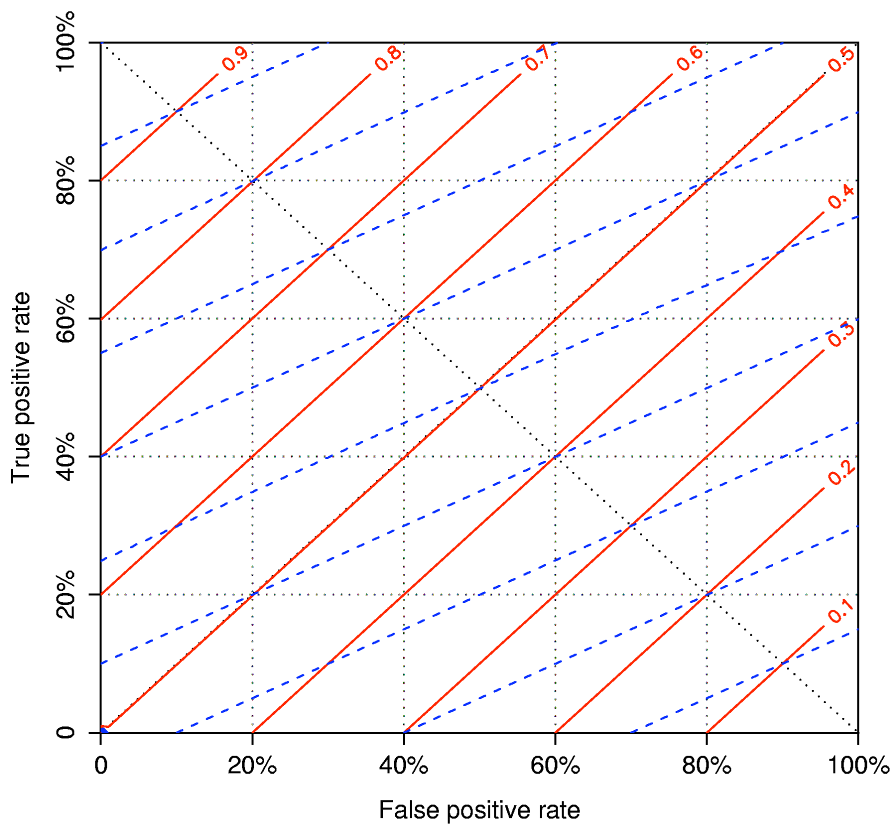
- Classifiers on the convex hull achieve the best accuracy for some class distributions
- Classifiers below the convex hull are always sub-optimal

Why is the convex hull a curve?

- Any performance on a line segment connecting two ROC points can be achieved by randomly choosing between them
  - the ascending default performance diagonal is just a special case
- The classifiers on the ROC convex hull can be combined to form the ROCCH-hybrid (Provost & Fawcett, 2001)
  - ordered sequence of classifiers
  - can be turned into a ranker
    - as with decision trees, see later

Iso-accuracy lines

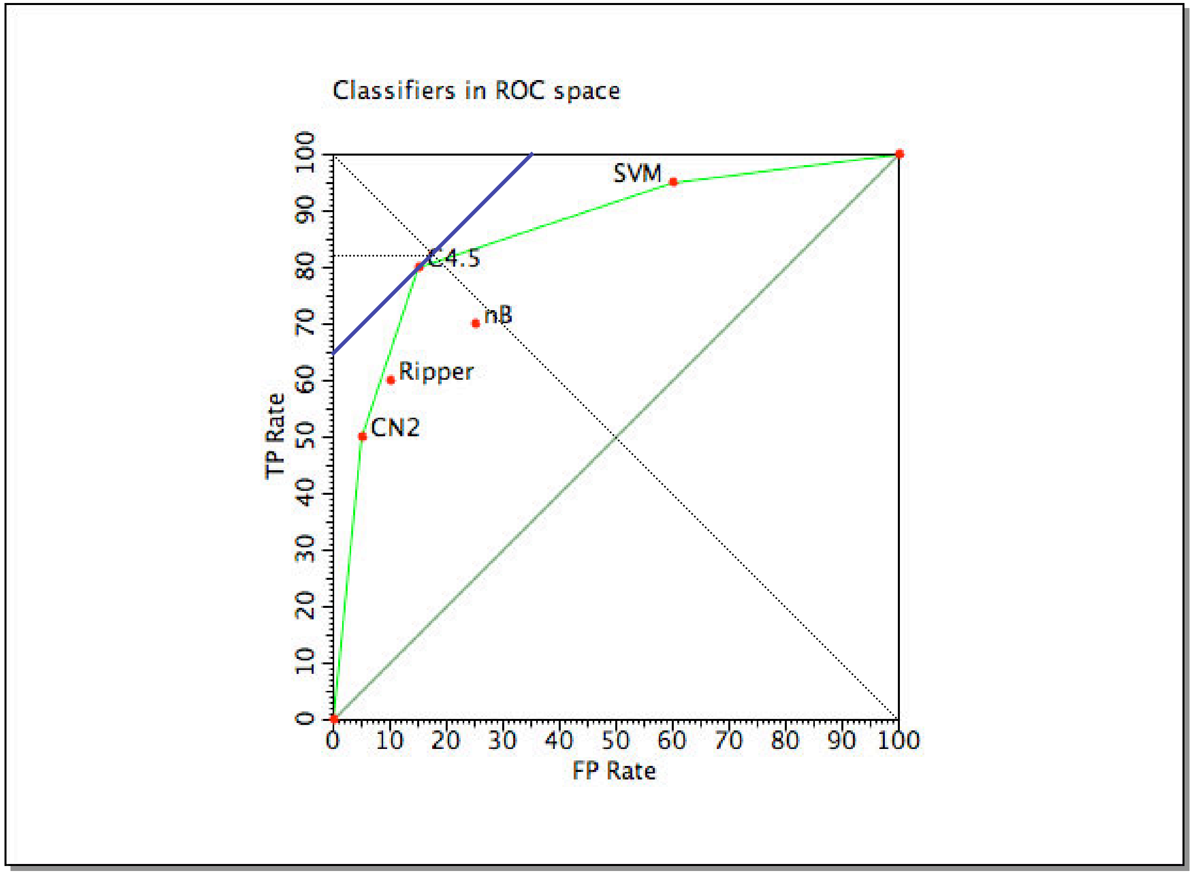
- Iso-accuracy line connects ROC points with the same accuracy
  - $pos*tpr + neg*(1-fpr) = a$
  - $tpr = \frac{a - neg}{pos} + \frac{neg}{pos} fpr$
- Parallel ascending lines with slope neg/pos
  - higher lines are better
  - on descending diagonal, $tpr = a$



Iso-accuracy & convex hull

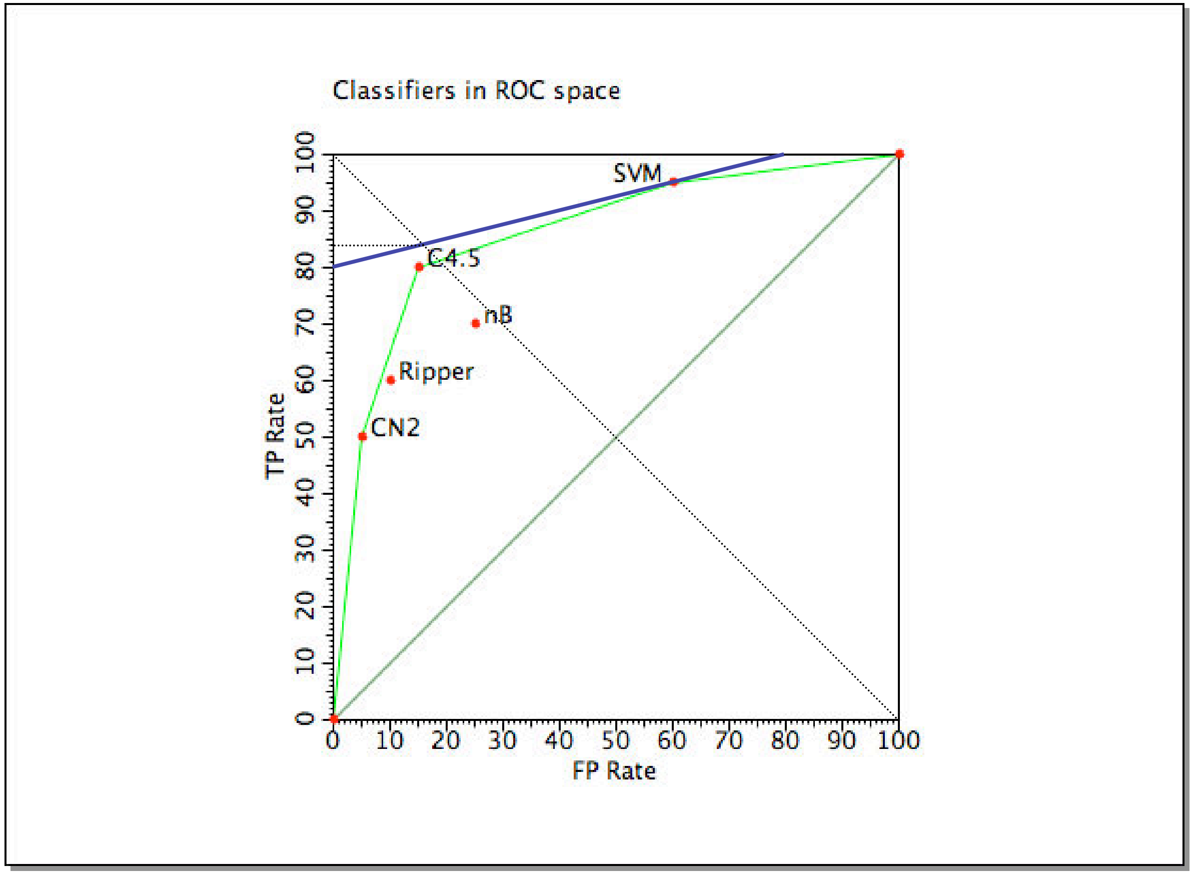
- Each line segment on the convex hull is an iso-accuracy line for a particular class distribution
  - under that distribution, the two classifiers on the end-points achieve the same accuracy
  - for distributions skewed towards negatives (steeper slope), the left one is better
  - for distributions skewed towards positives (flatter slope), the right one is better
- Each classifier on convex hull is optimal for a specific range of class distributions

Selecting the optimal classifier



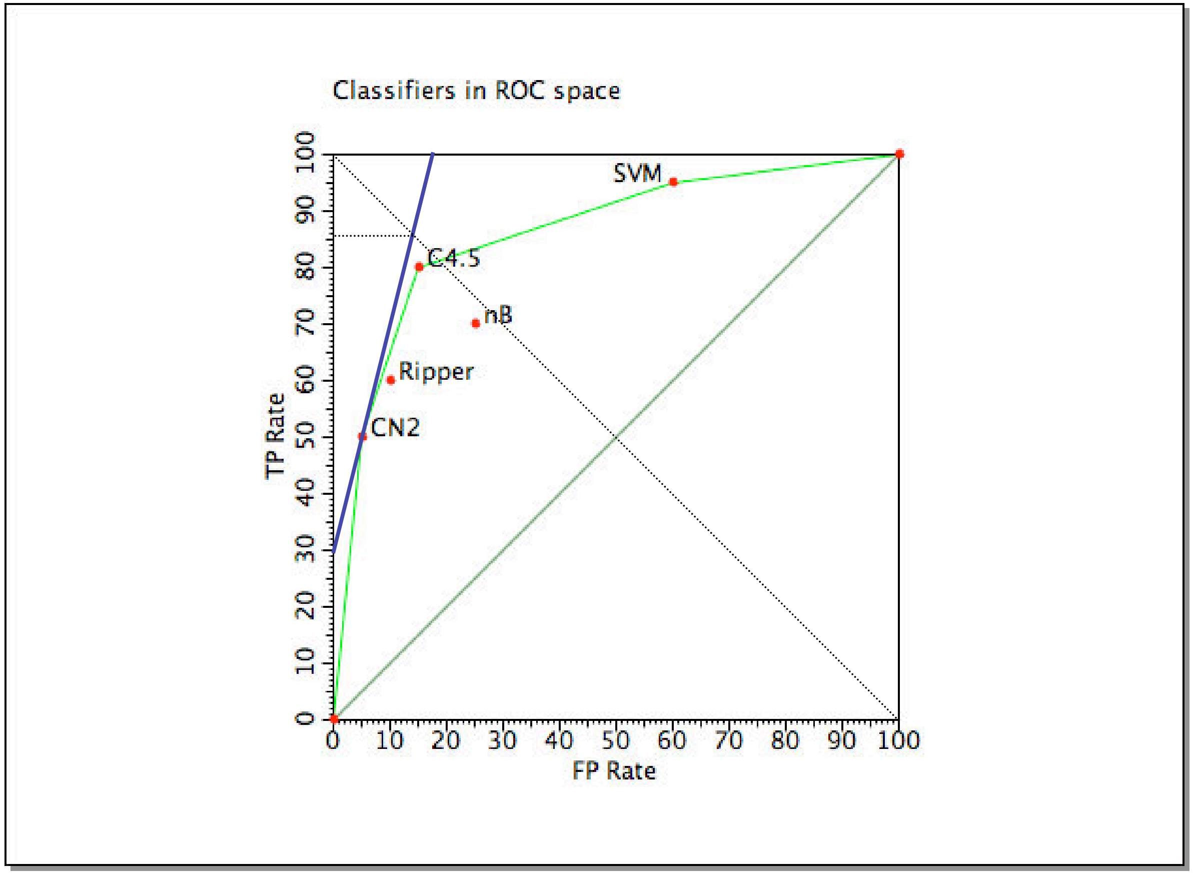
- For uniform class distribution, C4.5 is optimal
  - and achieves about 82% accuracy

Selecting the optimal classifier



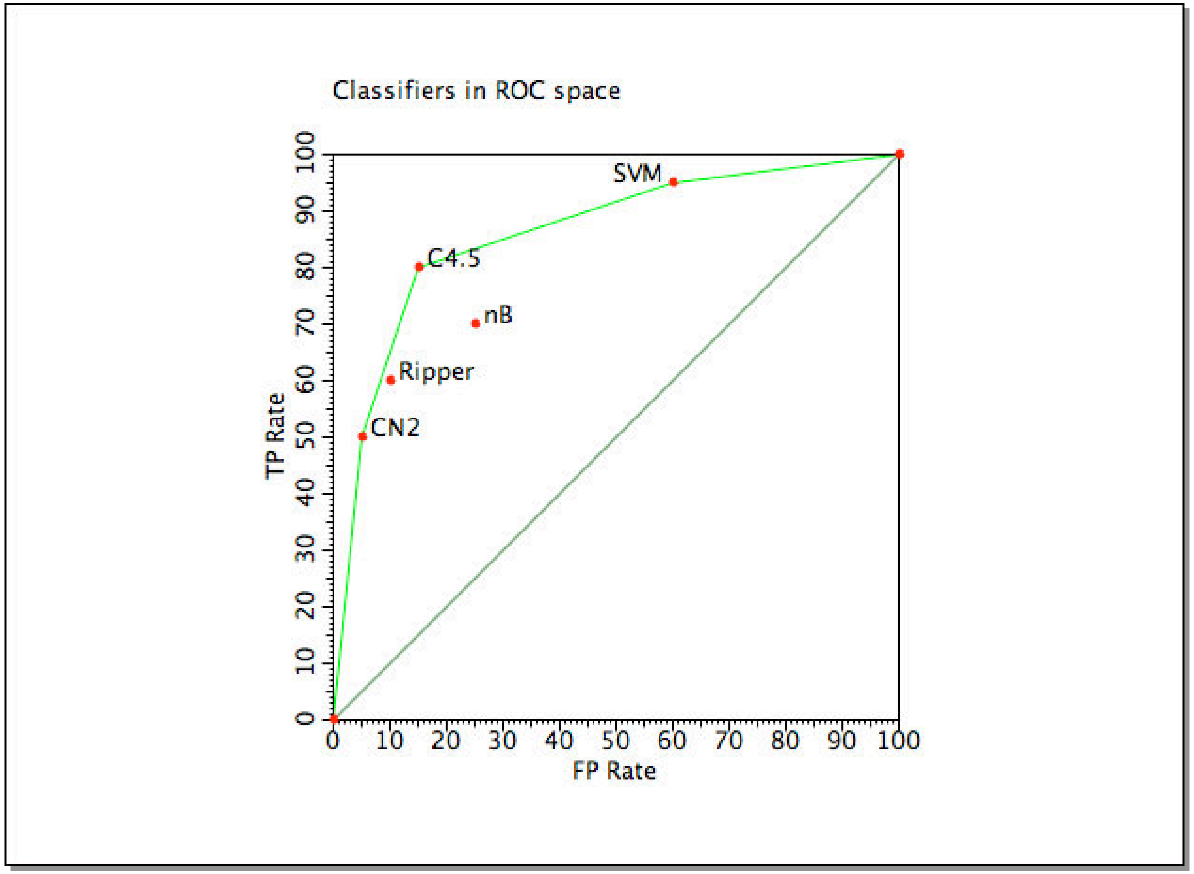
- With four times as many +ves as –ves, SVM is optimal
  - and achieves about 84% accuracy

Selecting the optimal classifier



- With four times as many –ves as +ves, CN2 is optimal
  - and achieves about 86% accuracy

Selecting the optimal classifier



- With less than 9% positives, AlwaysNeg is optimal
- With less than 11% negatives, AlwaysPos is optimal

Incorporating costs and profits

- Iso-accuracy and iso-error lines are the same
  - err = pos*(1–tpr) + neg*fpr
  - slope of iso-error line is neg/pos
- Incorporating misclassification costs:
  - cost = pos*(1–tpr)*C(-|+) + neg*fpr*C(+|-)
  - slope of iso-cost line is neg*C(+|–)/pos*C(–|+)
- Incorporating correct classification profits:
  - cost = pos*(1–tpr)*C(-|+) + neg*fpr*C(+|-) +
    pos*tpr*C(+|+) + neg*(1–fpr)*C(-|-)
  - slope of iso-yield line is
    neg*[C(+|-)-C(-|-)]/pos*[C(-|+)-C(+|+)]

Skew

- From a decision-making perspective, the cost matrix has one degree of freedom
  - need full cost matrix to determine absolute yield
- There is no reason to distinguish between cost skew and class skew
  - skew ratio expresses relative importance of negatives vs. positives
- ROC analysis deals with skew-sensitivity rather than cost-sensitivity

Rankers and classifiers

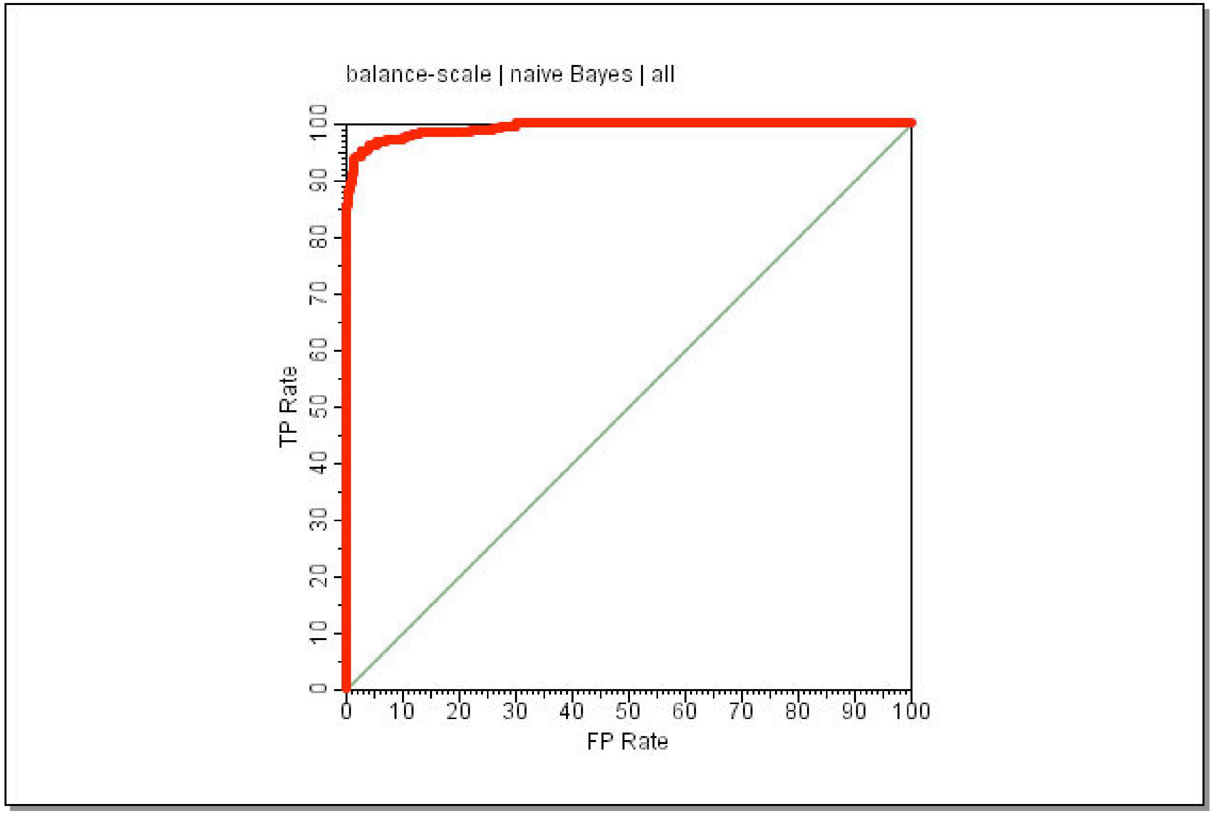
- A scoring classifier outputs scores f(x,+) and f(x,-) for each class
  - e.g. estimate class-conditional likelihoods P(x|+) and P(x|-)
  - scores don't need to be normalised
- f(x) = f(x,+)/f(x,-) can be used to rank instances from most to least likely positive
  - e.g. likelihood ratio P(x|+)/P(x|-)
- Rankers can be turned into classifiers by setting a threshold on f(x)

Drawing ROC curves for rankers

- Naïve method:
  - consider all possible thresholds
    - in fact, only k+1 for k instances
  - construct contingency table for each threshold
  - plot in ROC space

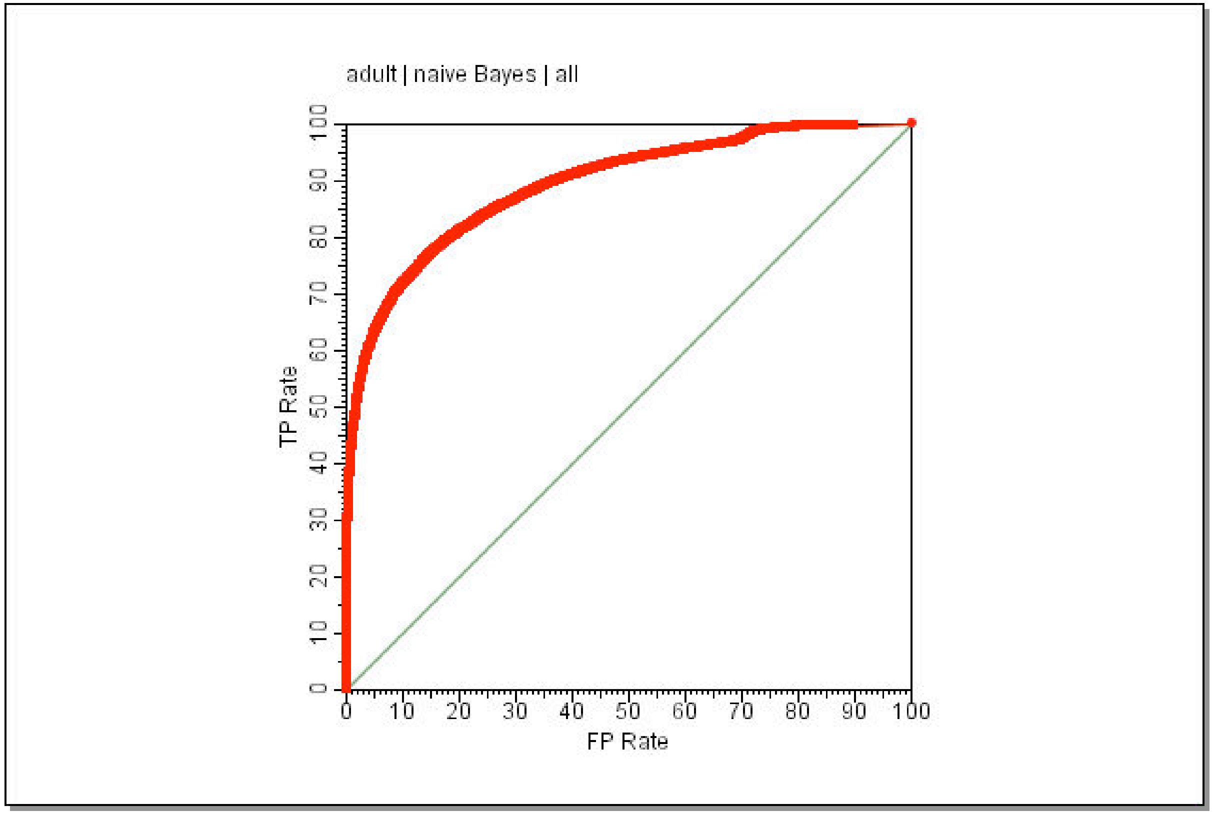
- Practical method:
  - rank test instances on decreasing score f(x)
  - starting in (0,0), if the next instance in the ranking is +ve move 1/Pos up, if it is –ve move 1/Neg to the right
    - make diagonal move in case of ties

Some example ROC curves



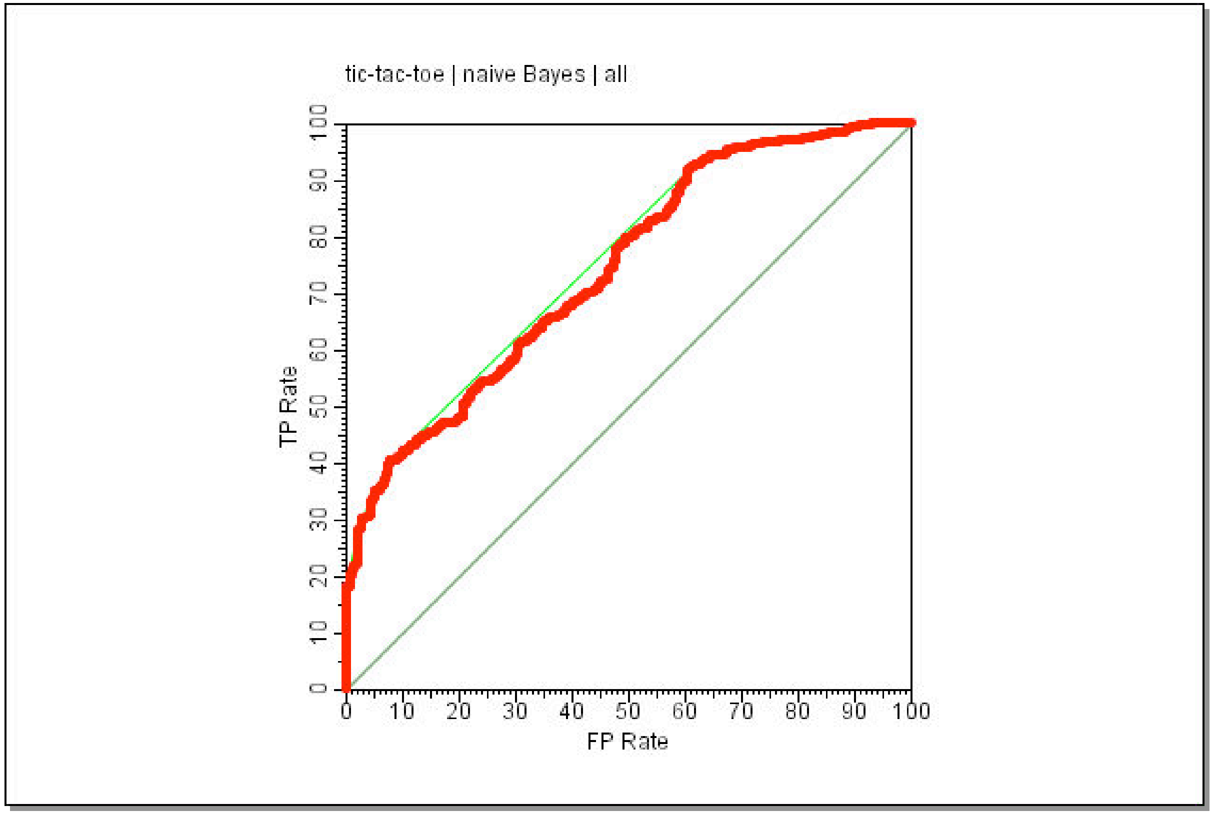
- Good separation between classes, convex curve

Some example ROC curves



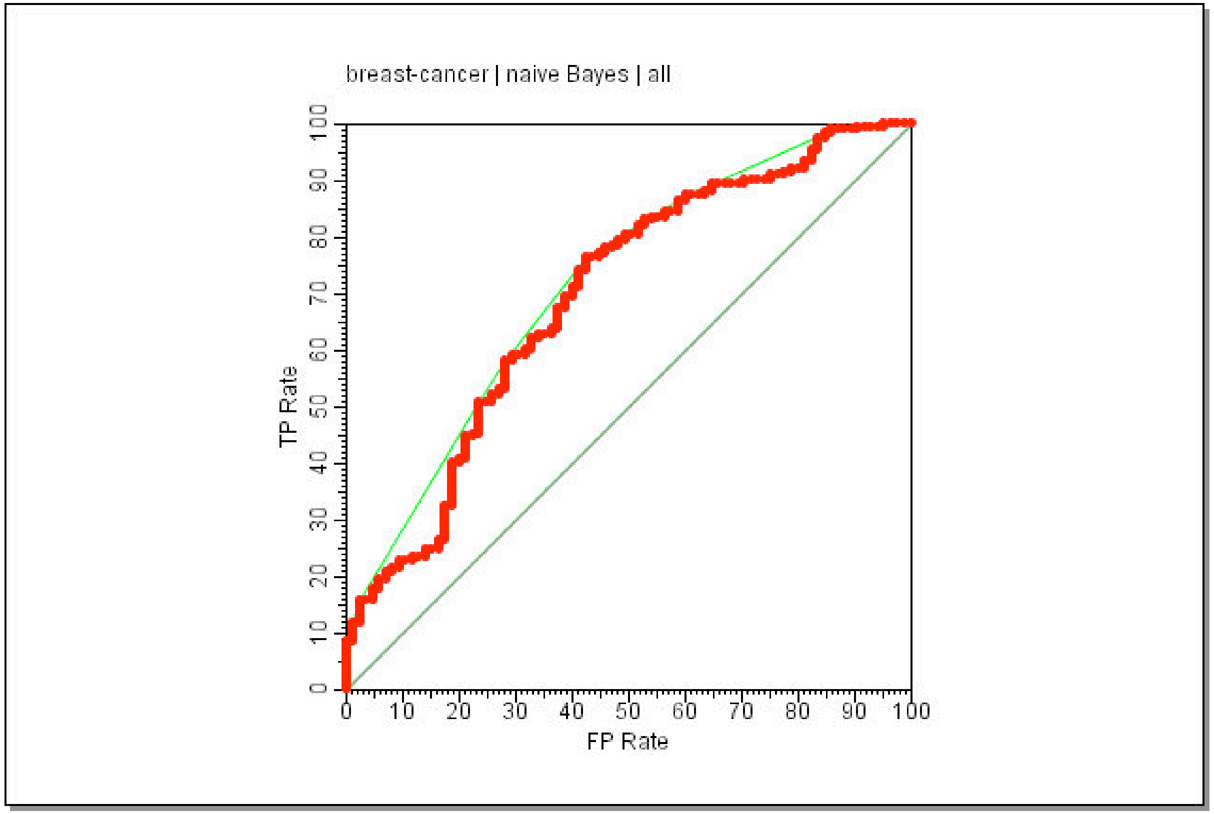
- Reasonable separation, mostly convex

Some example ROC curves



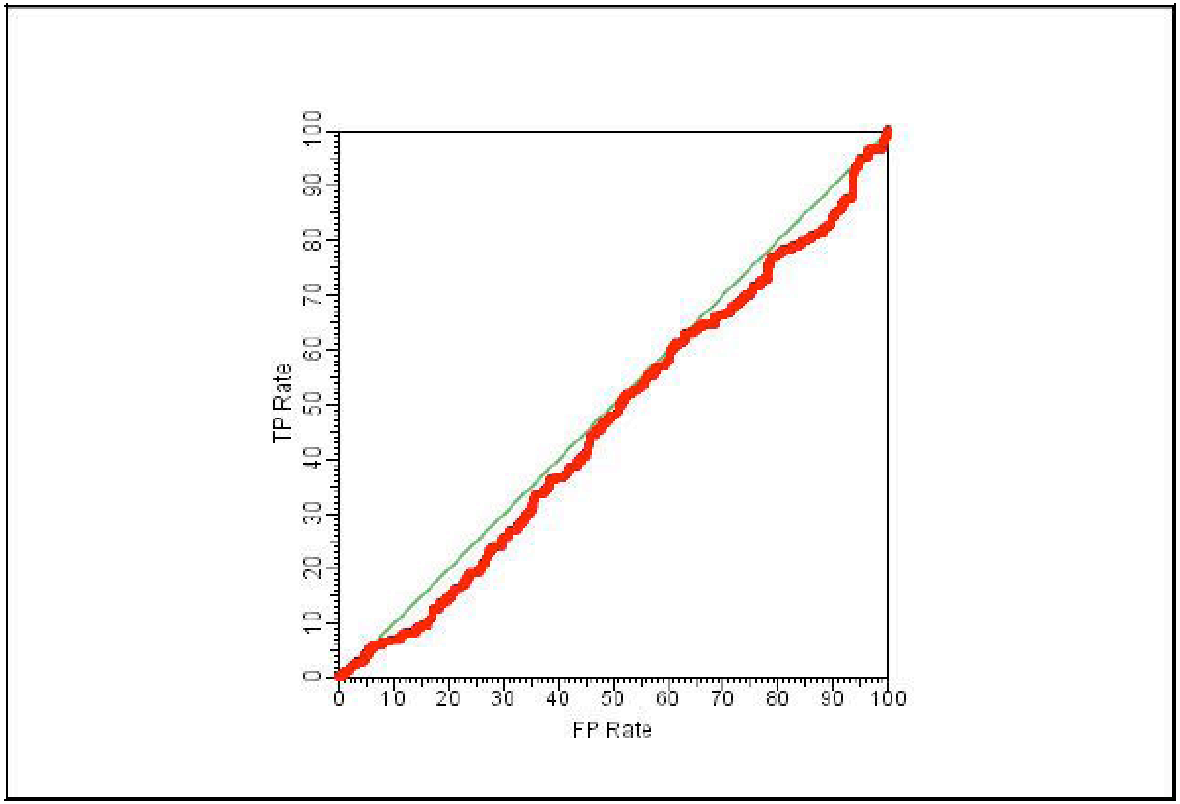
- Fairly poor separation, mostly convex

Some example ROC curves



- Poor separation, large and small concavities

Some example ROC curves



- Random performance

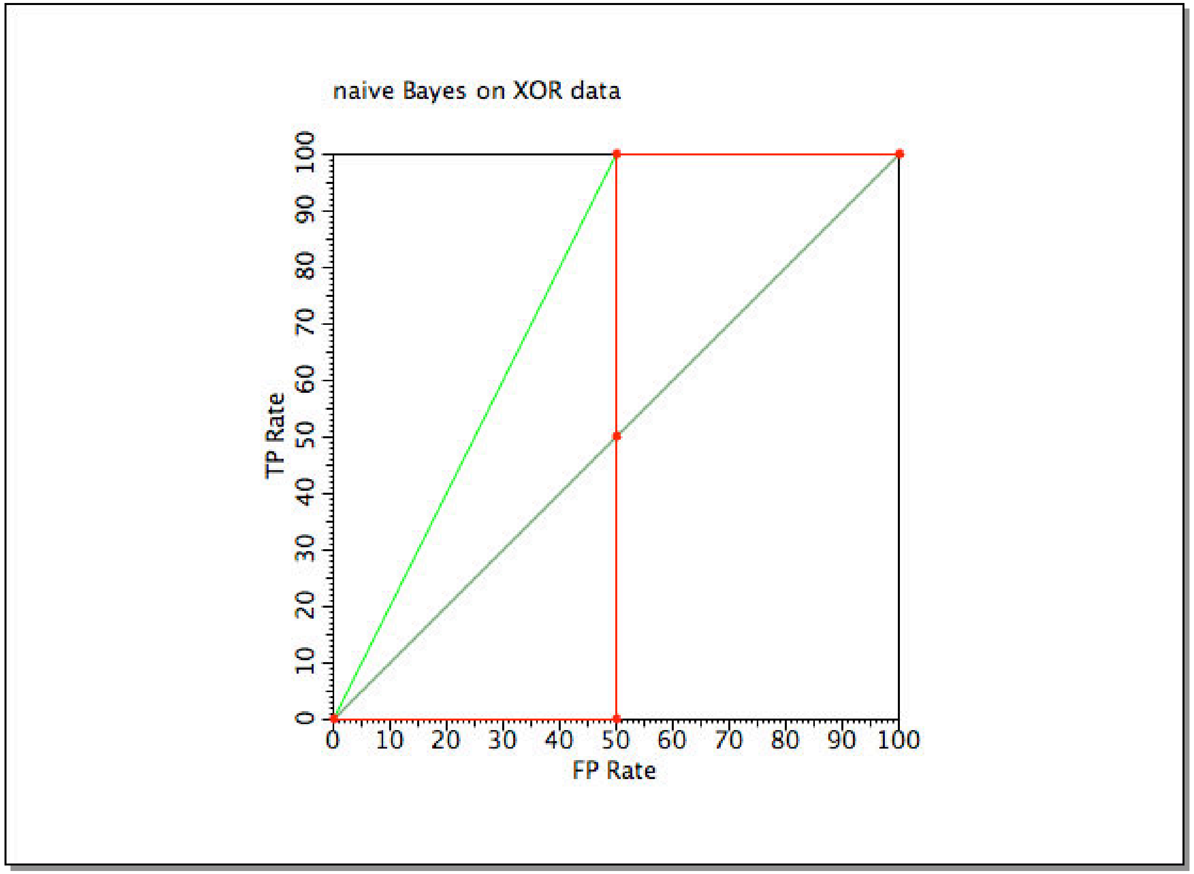
ROC curves for rankers

- The curve visualises the quality of the ranker or probabilistic model on a test set, without committing to a classification threshold
  - aggregates over all possible thresholds
- The slope of the curve indicates class distribution in that segment of the ranking
  - diagonal segment -> locally random behaviour
- Concavities indicate locally worse than random behaviour
  - convex hull corresponds to discretising scores
  - can potentially do better: repairing concavities

The AUC metric

- The Area Under ROC Curve (AUC) assesses the ranking in terms of separation of the classes
  - all the +ves before the –ves: AUC=1
  - random ordering: AUC=0.5
  - all the –ves before the +ves: AUC=0
- Equivalent to the Mann-Whitney-Wilcoxon sum of ranks test
  - estimates probability that randomly chosen +ve is ranked before randomly chosen –ve
  - $\frac{S_+ - Pos(Pos+1)/2}{Pos \cdot Neg}$ where S_+ is the sum of ranks of +ves
- Gini coefficient = 2*AUC–1 (area above diag.)
  - NB. not the same as Gini index!

AUC=0.5 not always random

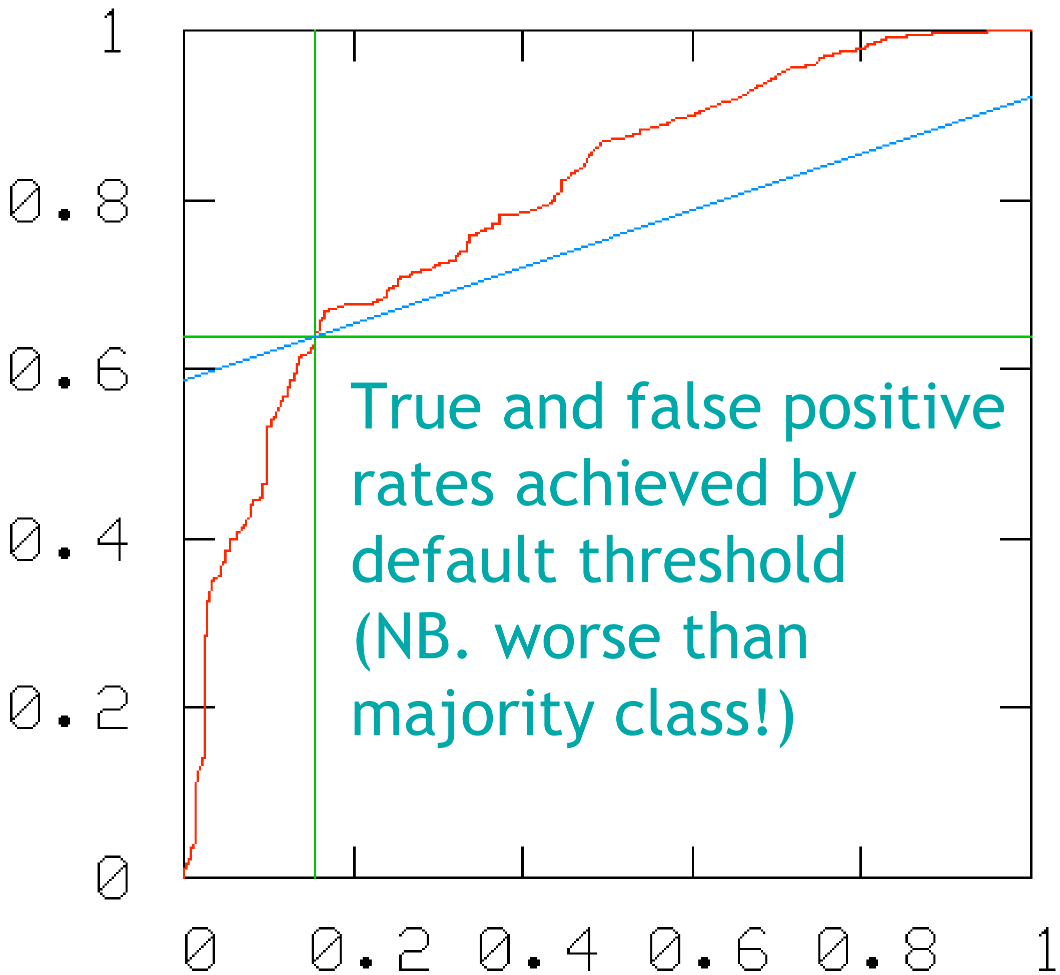


- Poor performance because data requires two classification boundaries

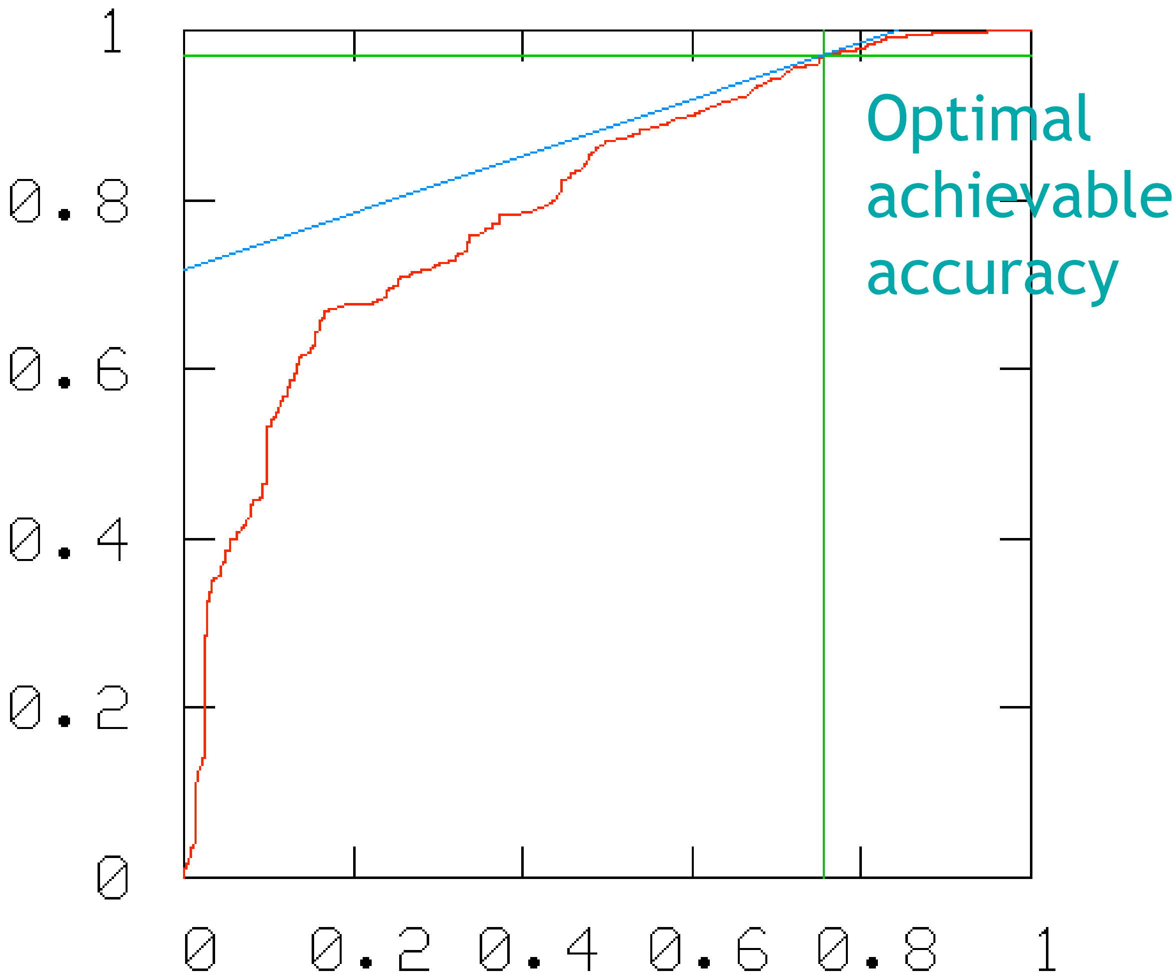
Turning rankers into classifiers

- Requires decision rule, i.e. setting a threshold on the scores f(x)
  - e.g. Bayesian: predict positive if $\frac{P(x \mid +)}{P(x \mid -)} > \frac{Neg}{Pos}$
  - equivalently: $\frac{P(x \mid +) \cdot Pos}{P(x \mid -) \cdot Neg} > 1$
- If scores are calibrated we can use a default threshold of 1
  - with uncalibrated scores we need to learn the threshold from the data
  - NB. naïve Bayes is uncalibrated
    - i.e. don't use Pos/Neg as prior!

Uncalibrated threshold



Calibrated threshold



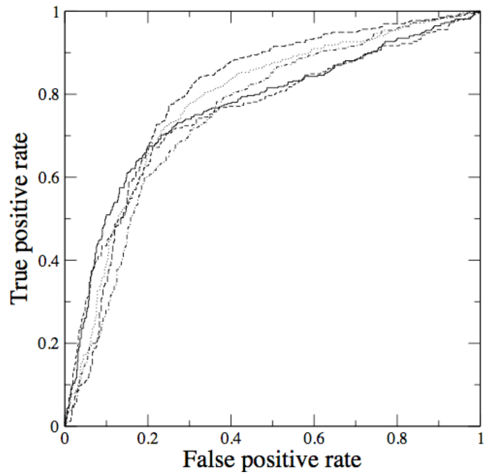
Calibration

- Easy in the two-class case: calculate accuracy in each point/threshold while tracing the curve, and return the threshold with maximum accuracy
  - NB. only calibrates the threshold, not the probabilities -> (Zadrozny & Elkan, 2002)
- Non-trivial in the multi-class case
  - discussed later

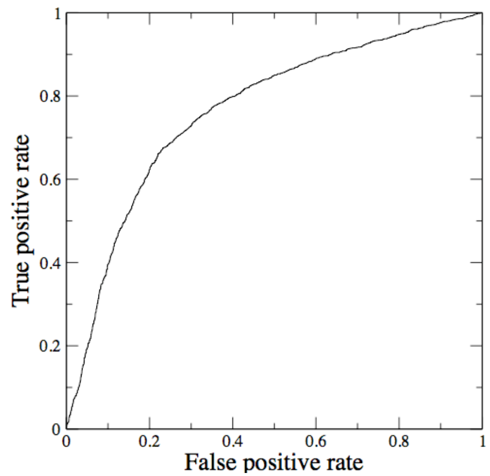
Averaging ROC curves

- To obtain a cross-validated ROC curve
  - just combine all test folds with scores for each instance, and draw a single ROC curve
- To obtain cross-validated AUC estimate with error bounds
  - calculate AUC in each test fold and average
  - or calculate AUC from single cv-ed curve and use bootstrap resampling for error bounds
- To obtain ROC curve with error bars
  - vertical averaging (sample at fixed fpr points)
  - threshold averaging (sample at fixed thresholds)
  - see (Fawcett, 2004)

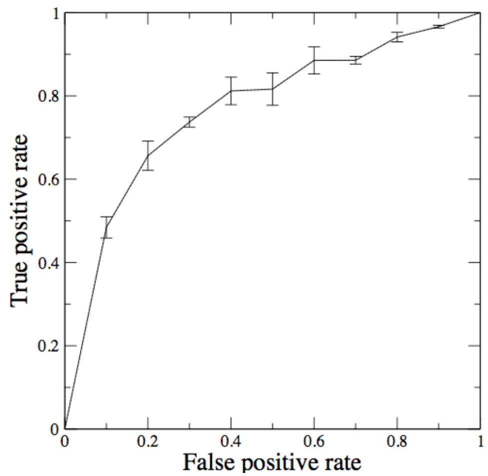
Averaging ROC curves



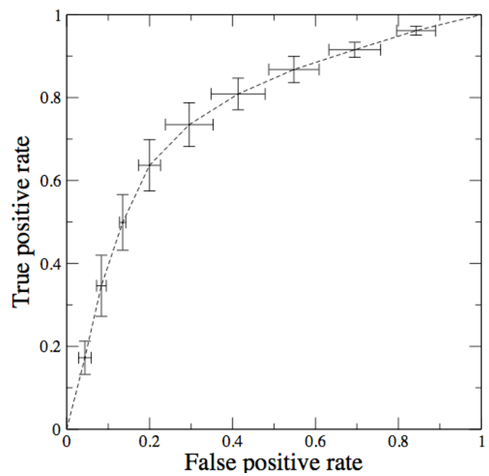
(a) ROC curves from five test samples



(b) ROC curve from combining the samples



(c) Vertical averaging, fixing fpr

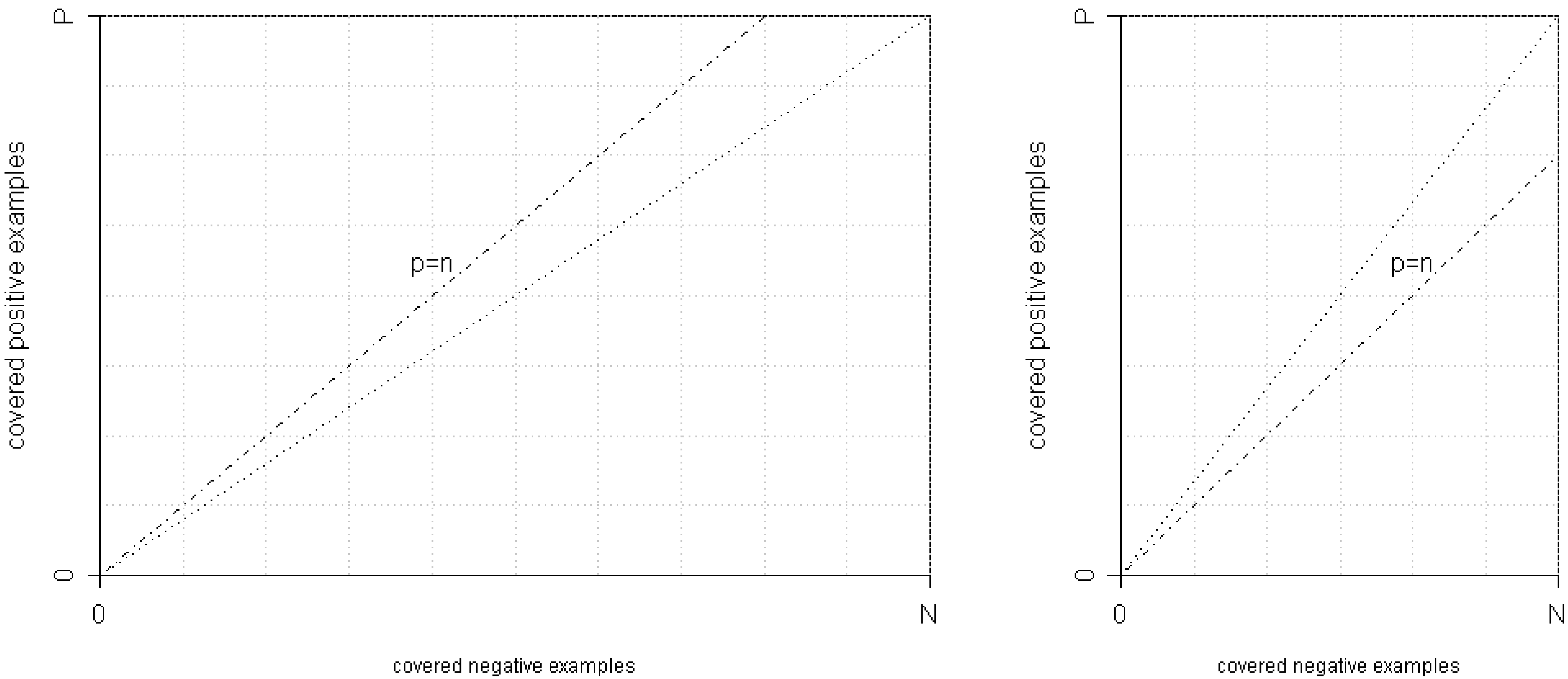


(d) Threshold averaging

From (Fawcett, 2004)

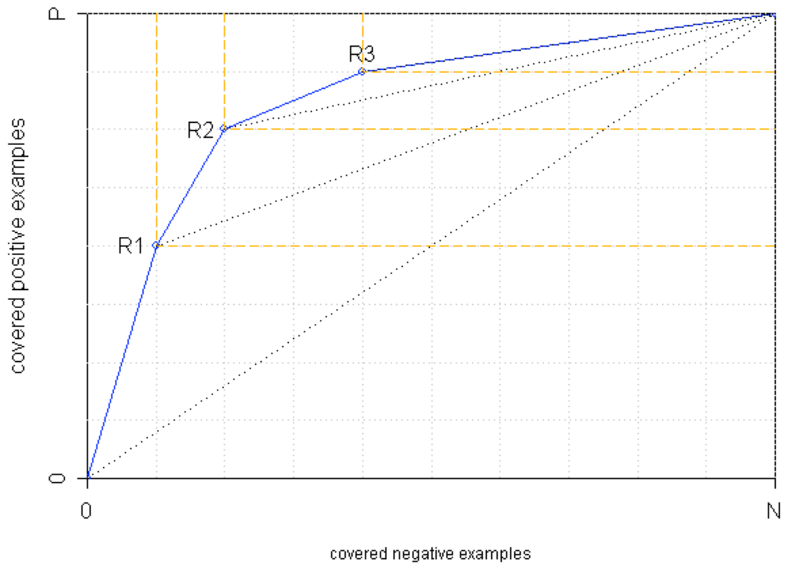
PN spaces

- PN spaces are ROC spaces with non-normalised axes
  - x-axis: covered –ves n (instead of fpr = n/Neg)
  - y-axis: covered +ves p (instead of tpr = p/Pos)



PN spaces vs. ROC spaces

- PN spaces can be used if class distribution (reflected by shape) is fixed
  - good for analysing behaviour of learning algorithm on single dataset (Gamberger & Lavrac, 2002; Fürnkranz & Flach, 2003)
- In PN spaces, iso-accuracy lines always have slope 1
  - PN spaces can be nested to reflect covering strategy



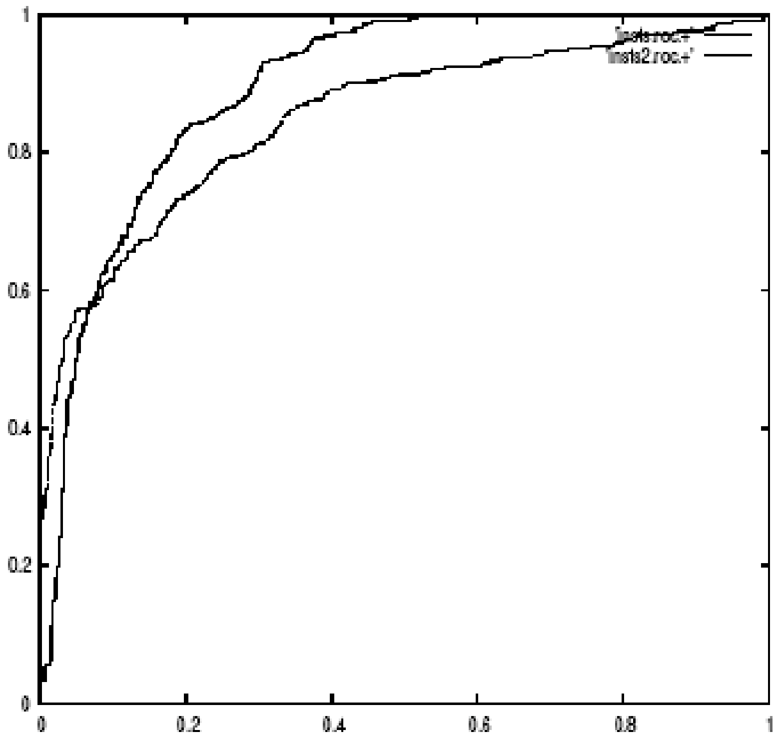
Precision-recall curves

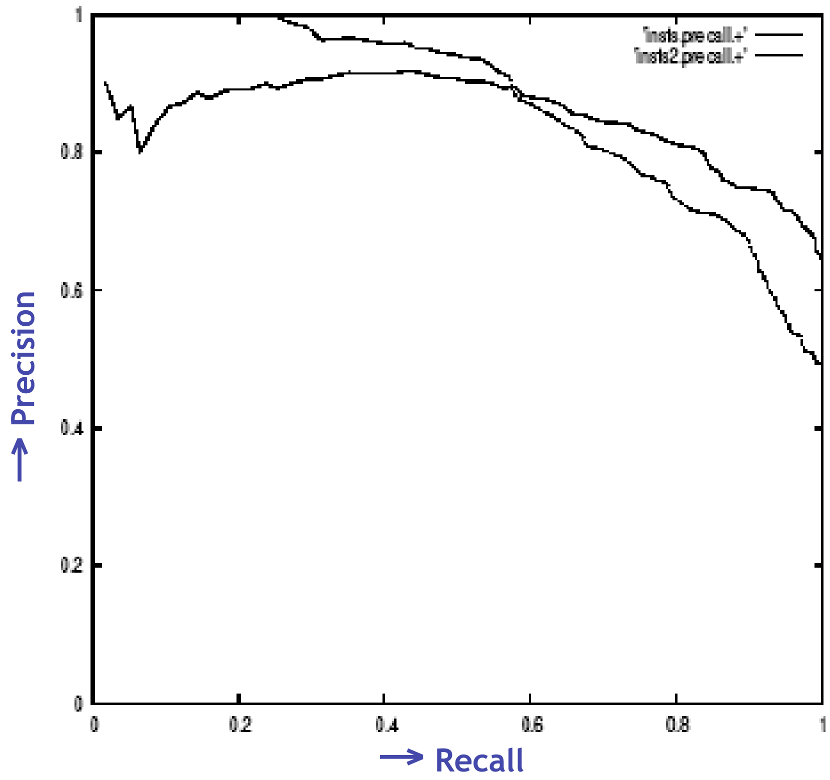
	Predicted positive	Predicted negative	
Positive examples	**TP**	**FN**	**Pos**
Negative examples	**FP**	**TN**	**Neg**
	PPos	**PNeg**	**N**

- Precision prec = TP/PPos = TP/TP+FP
  - fraction of positive predictions correct
- Recall rec = tpr = TP/Pos = TP/TP+FN
  - fraction of positives correctly predicted
- Note: neither depends on true negatives
  - makes sense in information retrieval, where true negatives tend to dominate —> low fpr easy

PR curves vs. ROC curves

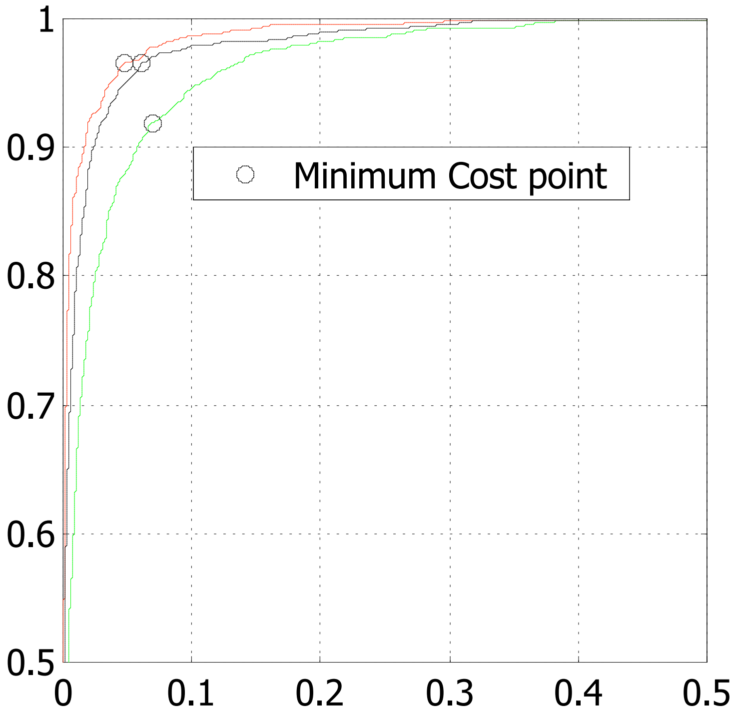
- Two ROC curves
- Corresponding PR curves

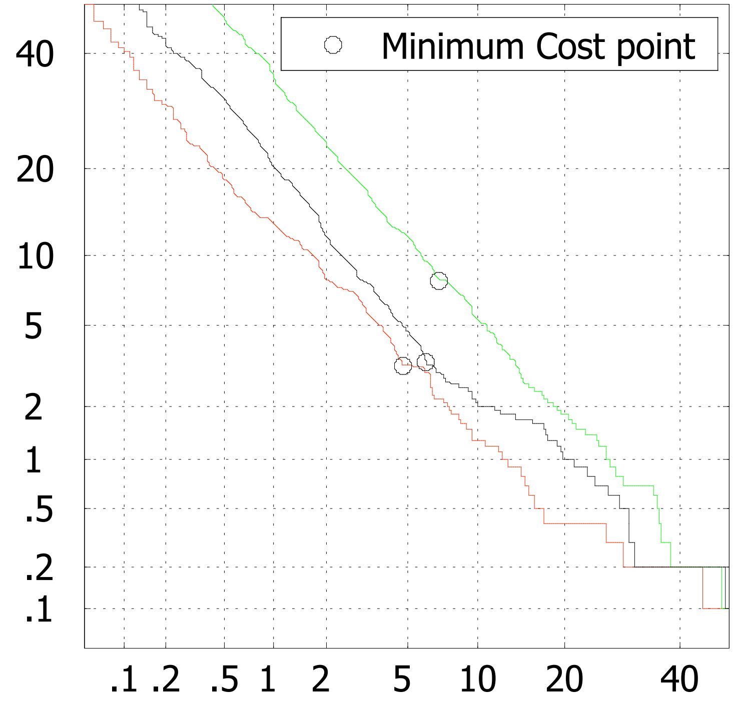




From (Fawcett, 2004)

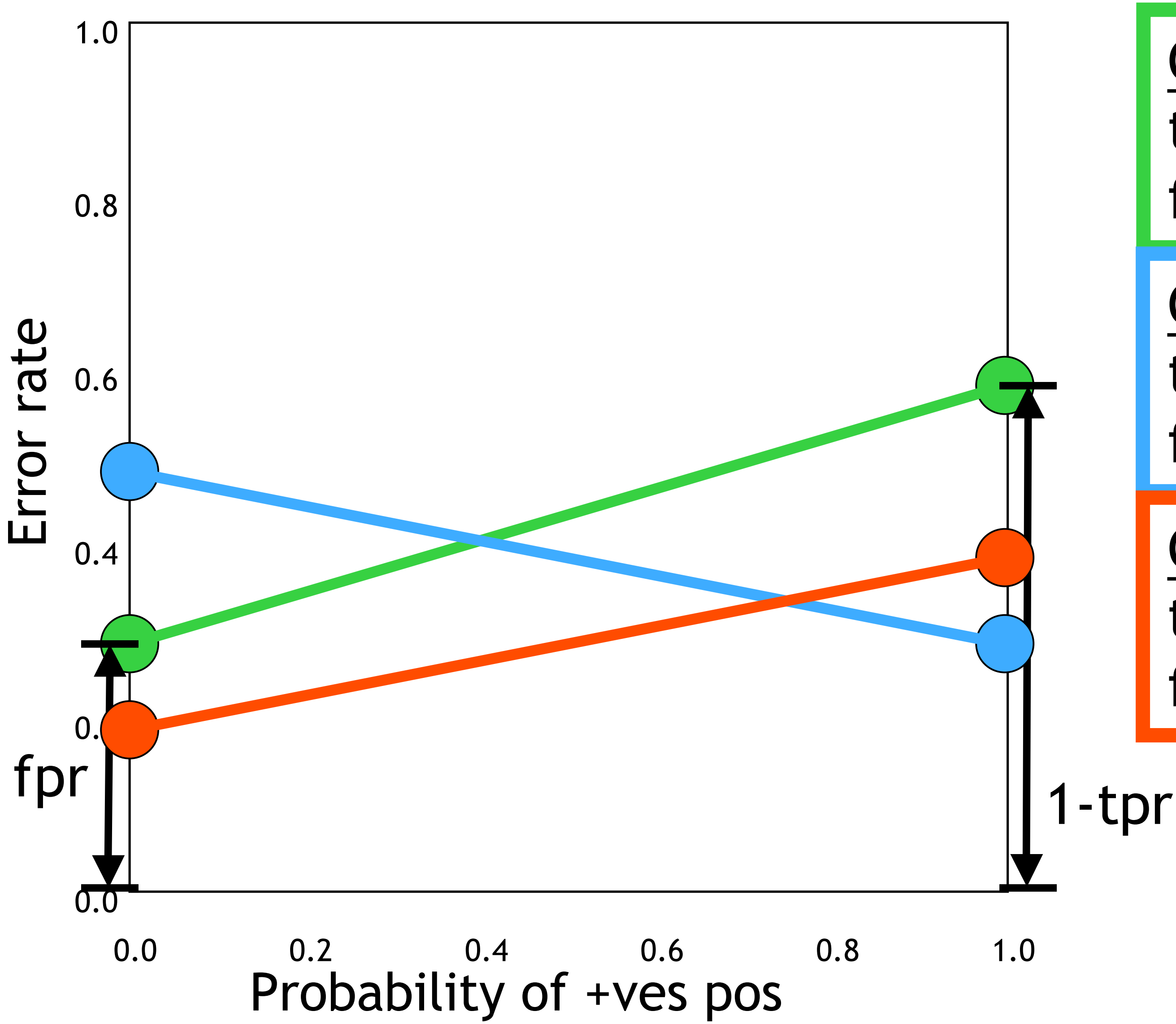
DET curves (Martin et al., 1997)





- Detection Error Trade-off
  - false negative rate instead of true positive rate
  - re-scaling using normal deviate scale

Cost curves (Drummond & Holte, 2001)

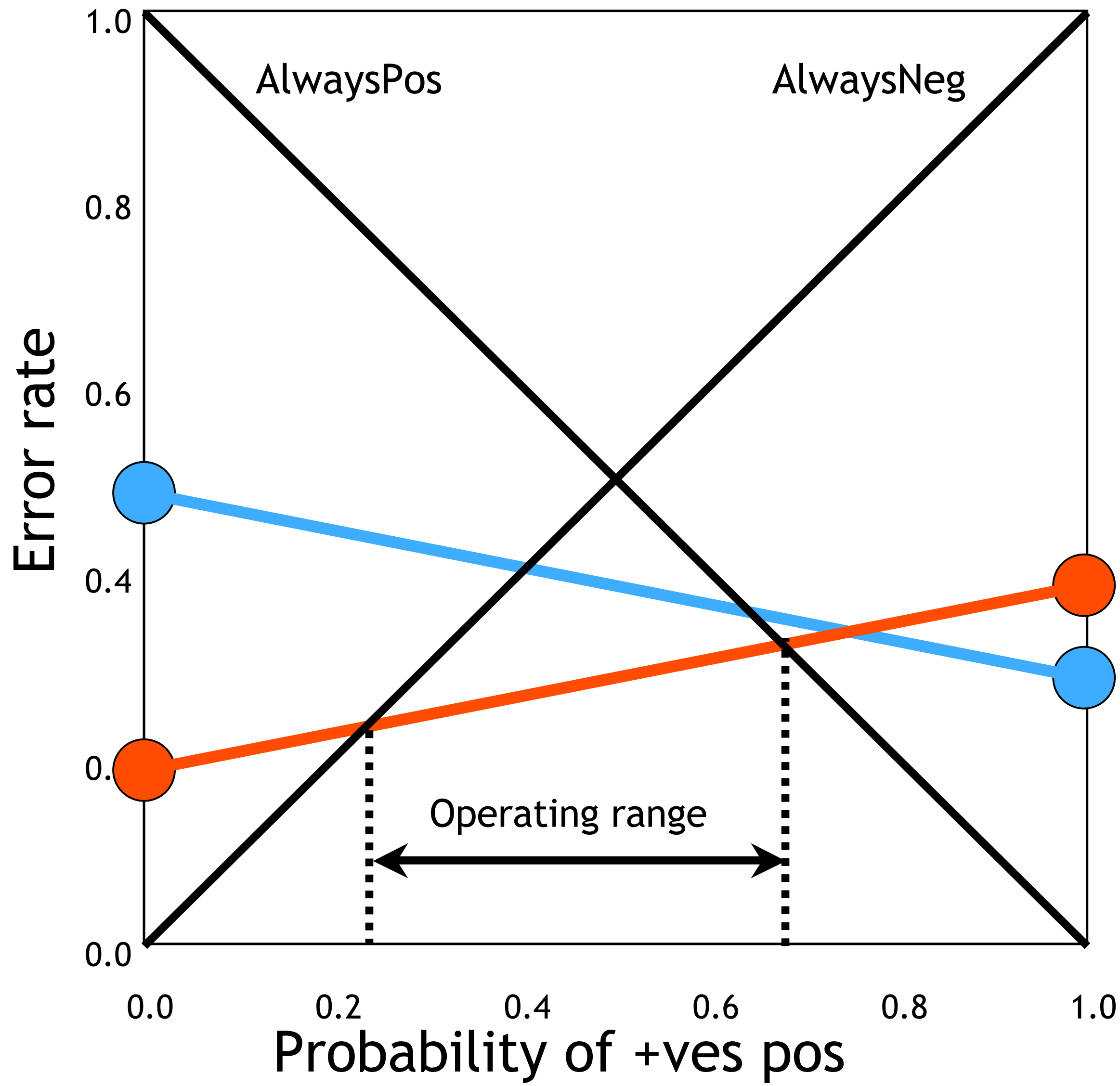


Classifier 1
tpr = 0.4
fpr = 0.3

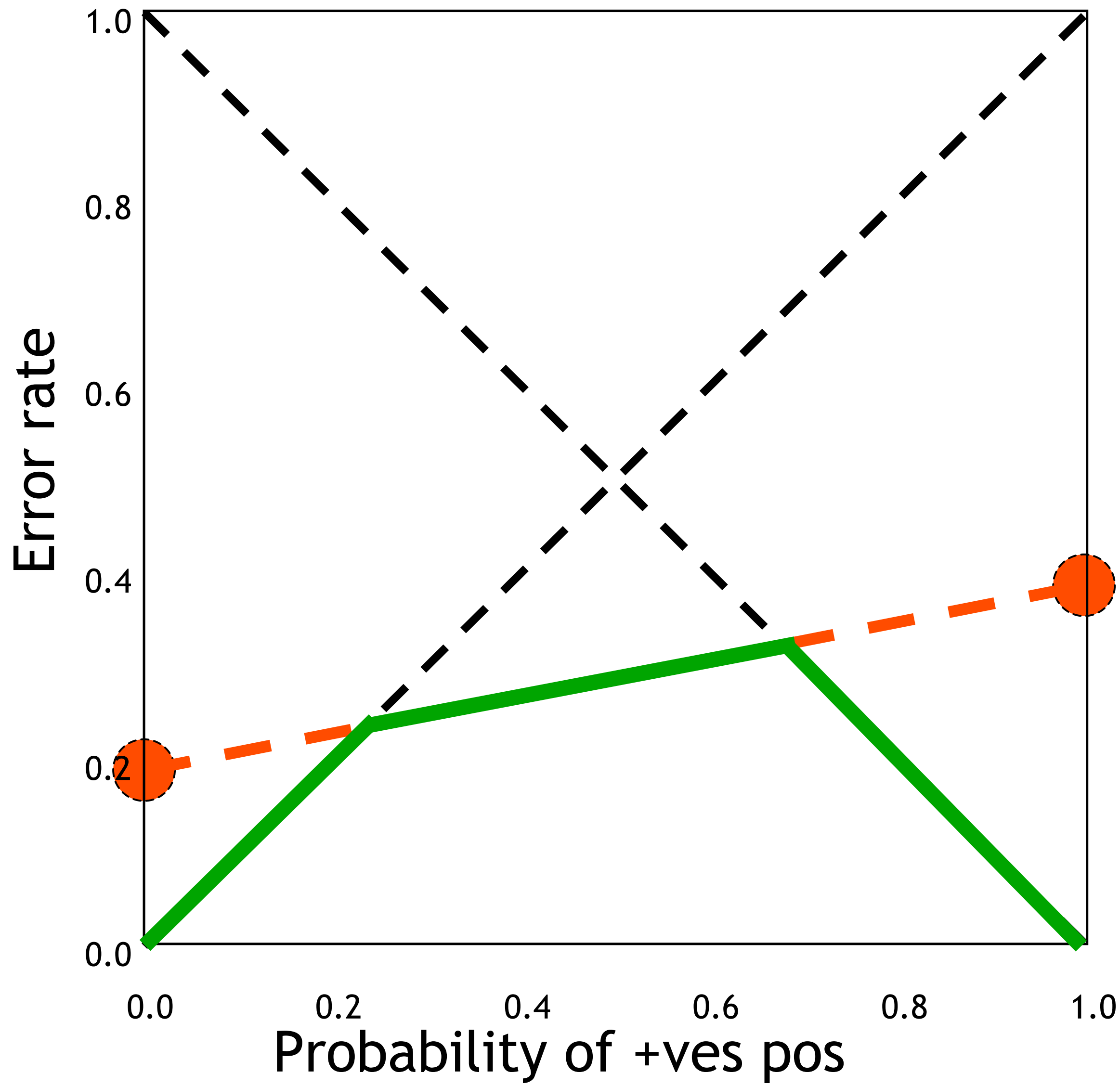
Classifier 2
tpr = 0.7
fpr = 0.5

Classifier 3
tpr = 0.6
fpr = 0.2

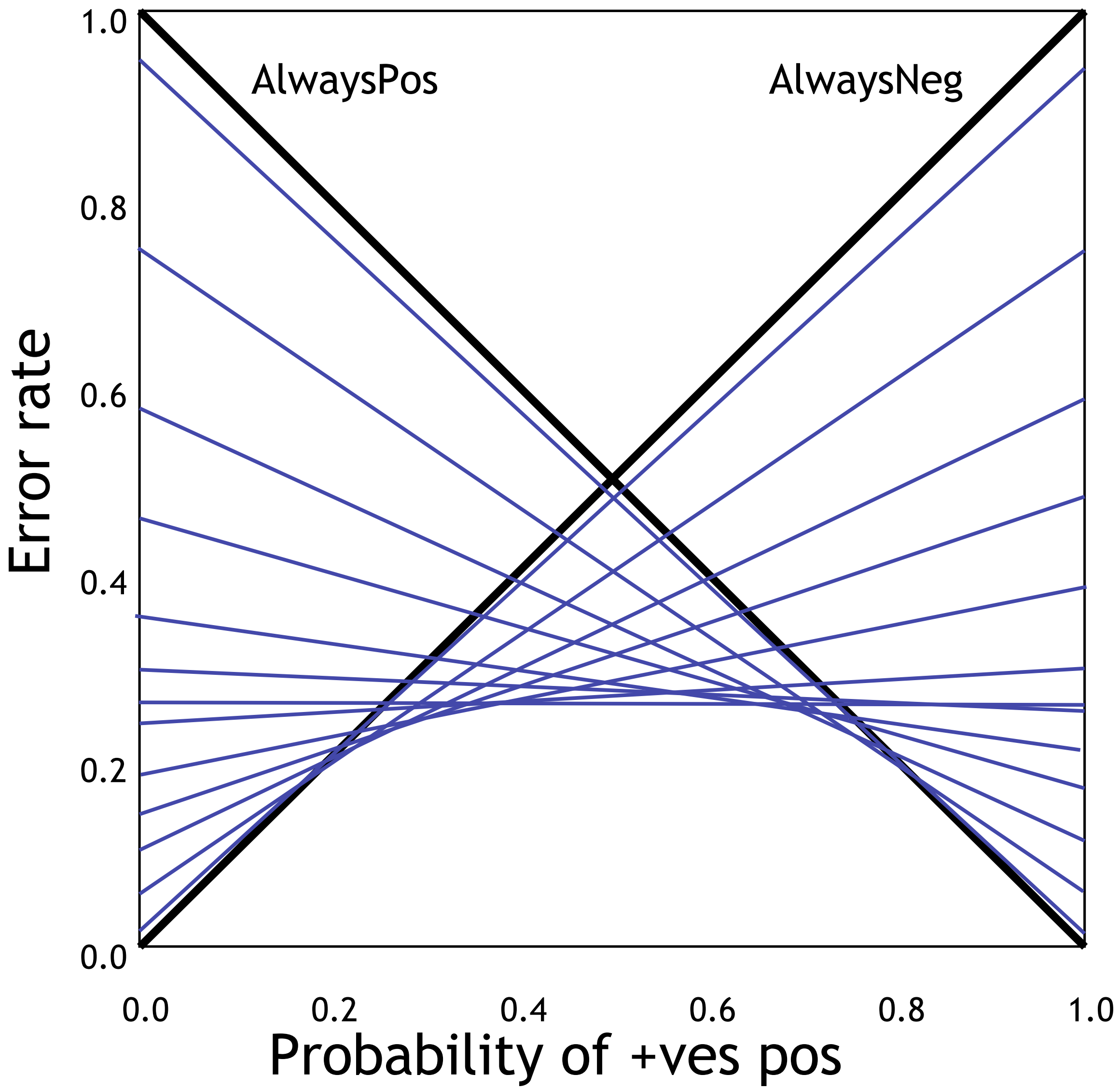
Operating range



Lower envelope



Varying thresholds



Taking costs into account

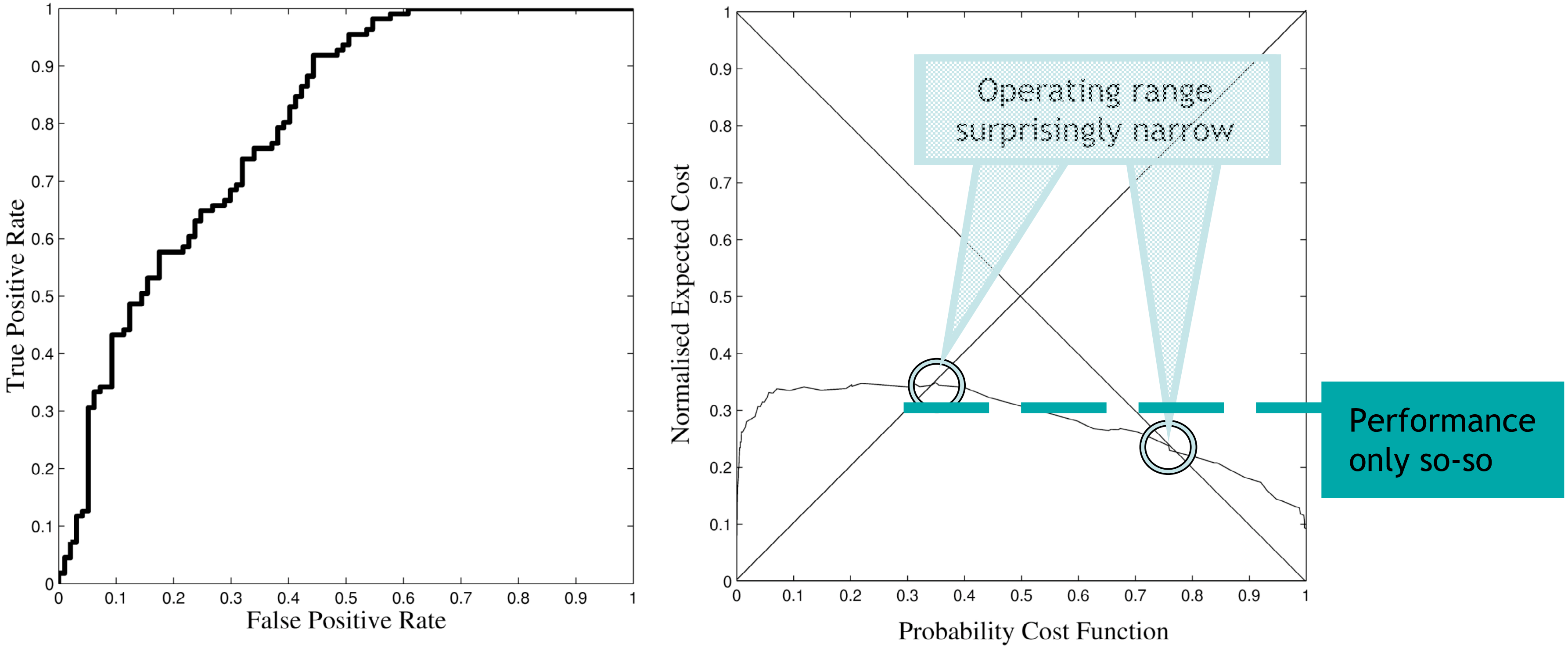
- Error rate is err = (1–tpr)*pos + fpr*(1–pos)

- Define probability cost function as

$$pcf = \frac{pos \cdot C(-|+)}{pos \cdot C(-|+) + neg \cdot C(+|-)}$$

- Normalised expected cost is
  nec = (1–tpr)*pcf + fpr*(1–pcf)

ROC curve vs. cost curve



Summary of Part I

- ROC analysis is useful for evaluating performance of classifiers and rankers
  - key idea: separate performance on classes
- ROC curves contain a wealth of information for understanding and improving performance of classifiers
  - requires visual inspection