Classifier performance evaluation

Václav Hlaváč

Czech Technical University in Prague
Czech Institute of Informatics, Robotics and Cybernetics
160 00 Prague 6, Jugoslávských partyzánů 1580/3, Czech Republic
http://people.ciirc.cvut.cz/hlavac, vaclav.hlavac@cvut.cz

Lecture plan

- Classifier performance as the statistical hypothesis testing. Training/test data.
- Criterial functions.
- Confusion matrix, characteristics.
- Receiver operation curve (ROC).

Application domains

- Classifiers (our main concern in this lecture), 1940s – radars, 1970s – medical diagnostics, 1990s – data mining.
- Regression.
- Estimation of probability densities.

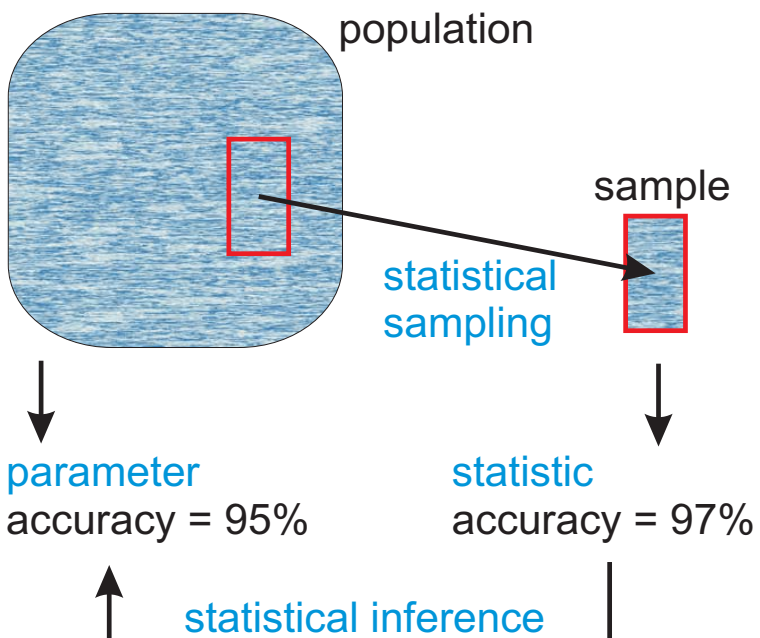
Classifier experimental evaluation

To what degree should we believe to the learned classifier?

- Classifiers (both supervised and unsupervised) are learned (trained) on a finite training multiset (also training data).
- A learned classifier has to be tested on a different test multiset (also test data) experimentally.
- The classifier performs on different data in the run mode that on which it has learned.
- The experimental performance on the test data is a proxy for the performance on unseen data. It checks the classifier's generalization ability.
- There is a need for a criterion function assessing the classifier performance experimentally, e.g., its error rate, accuracy, expected Bayesian risk (to be discussed later). A need for comparing classifiers experimentally.

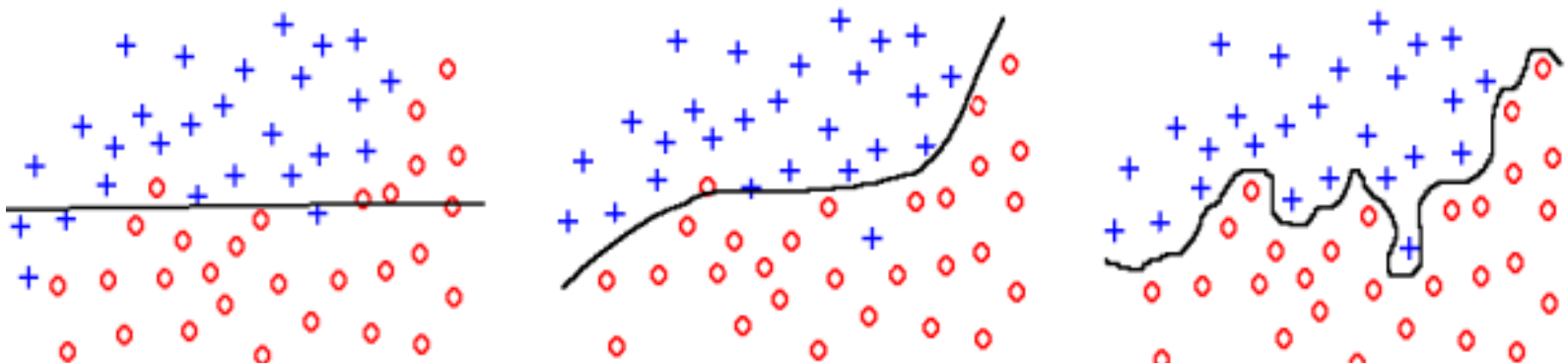
Evaluation is an instance of hypothesis testing

- Evaluation has to be treated as hypothesis testing in statistics.
- The value of the population parameter has to be statistically inferred based on the sample statistics (i.e., a training multiset in pattern recognition).



Danger of overfitting

- Learning the training data too precisely usually leads to poor classification results on new data.
- Classifier has to have the ability to generalize.



underfit fit overfit

Training vs. test data

Problem: Finite data are available only and have to be used both for training and testing.

- More training data gives better generalization.
- More test data gives better estimate for the classification error probability.
- Never evaluate performance on training data. The conclusion would be optimistically biased.

Partitioning of available finite multiset of data to training / test multisets.

- Hold out.
- Cross-validation.
- Bootstrap.

Once evaluation is finished, all the available data can be used to train the final classifier.

Hold out method

- Given data is randomly partitioned into two independent multisets.
  - Training multiset (e.g., 2/3 of data) for the statistical model construction, i.e., learning the classifier.
  - Test multiset (e.g., 1/3 of data) is hold out for the accuracy estimation of the classifier.
- Pros: Simple, easy to understand and implement.
- Cons: Not suitable for imbalanced training multisets.
- Repated hold out is a variation of the hold out method:

  Repeat the hold out k times, the accuracy is estimated as the average of the accuracies obtained.

K-fold cross-validation

- The training multiset is randomly divided into K disjoint multisets of equal size where each part has roughly the same class distribution.
- The classifier is trained K times, each time with a different multiset held out as a test multiset.
- The estimated error is the mean of these K errors.
- *Ron Kohavi, A Study of Cross-Validation and Bootstrap for Accuracy Estimation and Model Selection, IJCAI 1995.*
- Not suitable for imbalanced training multisets.

Courtesy: The presentation of Jin Tian, Iowa State Univ.; Figure Wikipedia.

Leave-one-out cross-validation

- A special case of K-fold cross-validation with $K = n$, where n is the total number of samples in the training multiset.
- n experiments are performed using $n - 1$ samples for training and the remaining sample for testing.
- Pros: Simple, easy to understand and implement.
- Cons: Computationally expensive.
- Leave-one-out cross-validation does not guarantee the same class distribution in training and test data!

  The extreme case:

  50% class A, 50% class B. Predict majority class label in the training data. True error 50%; Leave-one-out error estimate 100%!

Courtesy: The presentation of Jin Tian, Iowa State Univ..

Bootstrap aggregating, called also bagging

- The bootstrap uses sampling with replacement to form the training multiset.
- Given: the training multiset T consisting of n entries.
- Bootstrap generates m new datasets T_i each of size $n' < n$ by sampling T uniformly with replacement. The consequence is that some entries can be repeated in T_i.
- In a special case (called 632 boosting) when $n' = n$, for large n, T_i is expected to have $1 - \frac{1}{e} \approx 63.2\%$ of unique samples. The rest are duplicates.
- The m statistical models (e.g., classifiers, regressors) are learned using the above m bootstrap samples.
- The statistical models are combined, e.g. by averaging the output (for regression) or by voting (for classification).
- Proposed in: *Breiman, Leo (1996). Bagging predictors. Machine Learning 24 (2): 123–140.*

Recomended experimental validation procedure

- Use K-fold cross-validation ($K = 5$ or $K = 10$) for estimating performance estimates (accuracy, etc.).
- Compute the mean value of performance estimate, and standard deviation and confidence intervals.
- Report mean values of performance estimates and their standard deviations or 95% confidence intervals around the mean.

Courtesy: the presentation of Jin Tian, Iowa State Univ..

Criterion function to assess classifier performance

- Accuracy, error rate.
  - *Accuracy* is the percent of correct classifications.
  - *Error rate* = is the percent of incorrect classifications.
  - *Accuracy* = *1* - *Error rate*.
  - Problems with the accuracy:

    Assumes equal costs for misclassification.

    Assumes relatively uniform class distribution (cf., 0.5% patients of certain disease in the population).
- Other characteristics derived from the confusion matrix (to be explained later).
- Expected Bayesian risk (to be explained later).

Confusion matrix, two classes only

The confusion matrix is also called the contingency table.

		predicted	
		negative	positive
actual examples	negative	a TN - True Negative correct rejections	b FP - False Positive false alarms type I error
	positive	c FN - False Negative misses, type II error overlooked danger	d TP - True Positive hits

R. Kohavi, F. Provost: Glossary of terms, Machine Learning, Vol. 30, No. 2/3, 1998, pp. 271-274.

Performance measures calculated from the confusion matrix entries:

- Accuracy $= (a + d)/(a + b + c + d) =$ $(TN + TP)/total$
- **True positive rate**, recall, sensitivity $= d/(c + d) =$ $TP/actual\ positive$
- Specificity, true negative rate $= a/(a + b) =$ $TN/actual\ negative$
- Precision, predicted positive value $= d/(b + d) =$ $TP/predicted\ positive$
- **False positive rate**, false alarm $= b/(a + b) =$ $FP/actual\ negative = 1$ - $specificity$
- False negative rate $= c/(c + d) = FN/actual\ positive$

Confusion matrix, # of classes > 2

- The toy example (courtesy Stockman) shows predicted and true class labels of optical character recognition for numerals 0-9. There were 100 examples of each number class available for the evaluation. Empirical performance is given in percents.
- The classifier allows the reject option, class label R.
- Notice, e.g., unavoidable confusion between 4 and 9.

	class j predicted by a classifier										
true class i	'0'	'1'	'2'	'3'	'4'	'5'	'6'	'7'	'8'	'9'	'R'
'0'	97	0	0	0	0	0	1	0	0	1	1
'1'	0	98	0	0	1	0	0	1	0	0	0
'2'	0	0	96	1	0	1	0	1	0	0	1
'3'	0	0	2	95	0	1	0	0	1	0	1
'4'	0	0	0	0	98	0	0	0	0	2	0
'5'	0	0	0	1	0	97	0	0	0	0	2
'6'	1	0	0	0	0	1	98	0	0	0	0
'7'	0	0	1	0	0	0	0	98	0	0	1
'8'	0	0	0	1	0	0	1	0	96	1	1
'9'	1	0	0	0	3	1	0	0	0	95	0

Unequal costs of decisions

- Examples:

  Medical diagnosis: The cost of falsely indicated breast cancer in population screening is smaller than the cost of missing a true disease.

  Defense against ballistic missiles: The cost of missing a real attack is much higher than the cost of false alarm.

- Bayesian risk is able to represent unequal costs of decisions.
- We will show that there is a tradeoff between apriori probability of the class and the induced cost.

Criterion, Bayesian risk

For a multiset of observations X, set of hidden states Y and decisions D, statistical model given by the joint probability $p_{XY}: X \times Y \to \mathbb{R}$ and the penalty function $W: Y \times D \to \mathbb{R}$ and a decision strategy $Q: X \to D$ the Bayesian risk is given as

$$R(Q) = \sum_{x \in X} \sum_{y \in Y} p_{XY}(x, y)\, W(y, Q(x))\,.$$

- It is difficult to fulfil the assumption that a statistical model p_{XY} is known in many practical tasks.
- If the Bayesian risk would be calculated on the finite (training) multiset then it would be too optimistic.
- Two substitutions for Baysesian risk are used in practical tasks:
  - Expected risk.
  - Structural risk.

Two paradigms for learning classifiers

- Choose a class Q of decision functions (classifiers) $q\colon X \to Y$.
- Find $q^* \in Q$ by minimizing some criterion function on the training multiset that approximates the risk $R(q)$ (which cannot be computed).
- Learning paradigm is defined by the criterion function:
  1. Expected risk minimization, in which the true risk is approximated by the error rate on the training multiset,

$$R_{\mathrm{emp}}(q(x,\Theta)) = \frac{1}{L}\sum_{i=1}^{L} W(y_i, q(x_i,\Theta)) \ ,$$

$$\Theta^* = \operatorname*{argmin}_{\Theta} R_{\mathrm{emp}}(q(x,\Theta)) \ .$$

  Examples: Perceptron, Neural nets (Back-propagation), etc.
  2. Structural risk minimization which introduces the guaranteed risk $J(Q)$, $R(Q) < J(Q)$. *Example: SVM (Support Vector Machines).*

Problem of unknown class distribution and costs

We already know: The class distribution and the costs of each error determine the goodness of classifiers.

Additional problem:

- In many circumstances, until the application time, we do not know the class distribution and/or it is difficult to estimate the cost matrix. E.g., an email spam filter.
- Statistical models have to be learned in advance.

Possible solution: Incremental learning.

Unbalanced problems and data

- Classes have often unequal frequency.
  - Medical diagnosis: 95 % healthy, 5% disease.
  - e-Commerce: 99 % do not buy, 1 % buy.
  - Security: 99.999 % of citizens are not terrorists.
- Similar situation for multiclass classifiers.
- Majority class classifier can be 99 % correct but useless.
  Example: OCR, 99 % correct, error at the every second line. This is why OCR is not widely used.
- Learning methods for supervised classification will often overfit the majority class

---

- How should we train classifiers and evaluated them for unbalanced problems?

Balancing unbalanced data (1)

Two class problems:

- Build a balanced training multiset, use it for classifier training.
  - E.g., by oversampling. Select randomly a desired number of minority class instances.
  - Add equal number of randomly selected majority class instances.
- Build a balanced test multiset (different from training multiset, of course) and test the classifier using it.

Multiclass problems:

- Generalize 'balancing' to multiple classes.
- Ensure that each class is represented with approximately equal proportions in training and test datasets.

Balancing unbalanced data (2)

Two balancing methods are used:

1. During pre-processing: one can add minority-class data points or remove majority-class ones. These methods are called oversampling or undersampling respectively.

   Basic method is SMOOTE (= Synthetic minority over-sampling technique). Three steps:

   (a) Construct the list of the k-nearest neighbors of each minority data point (usually $k = 5$);

   (b) n added minority points are randomly chosen and for each minority data point x_1 picked, a random nearest neighbor x_2 of x_1 is chosen;

   (c) Create a new data point between x_1 and x_2 using linear interpolation.

   Chawla N. V. et al. 2002. Many improvements in the literature.

2. In the penalty function: one can increase the minority-class weight and decrease the majority class one.

Thoughts about balancing by modifying penalty function

Natural hesitation: Balancing the training multiset changes the underlying statistical problem. Can we change it?

Hesitation is justified. The statistical problem is indeed changed.

Good news:

- Balancing can be seen as the change of the penalty function. In the balanced training multiset, the majority class patterns occur less often. It means that the penalty assigned to them is lowered proportionally to their relative frequency.

$$R(q^*) = \min_{q \in D} \sum_{x \in X} \sum_{y \in Y} p_{XY}(x, y)\, W(y, q(x))$$

$$R(q^*) = \min_{q(x) \in D} \sum_{x \in X} \sum_{y \in Y} p(x)\, p_{Y|X}(y|x)\, W(y, q(x))$$

- This modified problem is what the end user usually wants as she/he is interested in a good performance for the minority class.

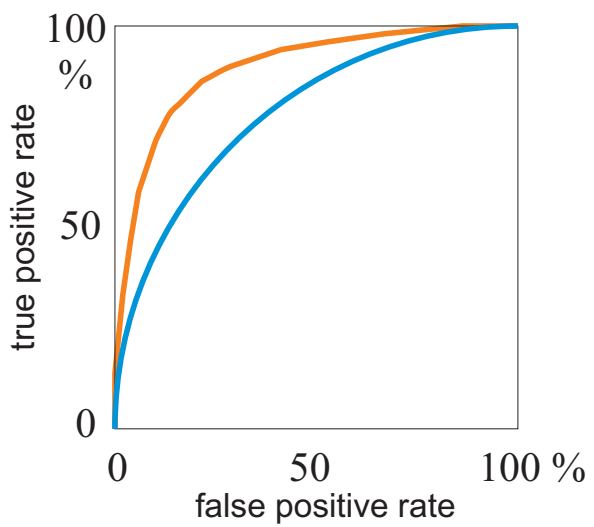
Scalar characteristics are not good for evaluating performance

- Scalar characteristics as the accuracy, expected cost, area under ROC curve (AUC, to be explained soon) do not provide enough information.
- We are interested in:
  - How are errors distributed across the classes?
  - How will each classifier perform in different testing conditions (costs or class ratios other than those measured in the experiment)?
- Two numbers – true positive rate (hits) and false positive rate (false alarms) – are much more informative than the single number.
- These two numbers are better visualized by a curve, e.g., by a Receiver Operating Characteristic (ROC), which informs about:
  - Performance for all possible misclassification costs.
  - Performance for all possible class ratios.
  - Under what conditions the classifier c_1 outperforms the classifier c_2?

ROC – Receiver Operating Characteristic

- A graphical plot showing the true positive rate (hits) against the false positive rate (false alarms).
- Called also often ROC curve.
- Originates in WWII processing of radar signals.
- Useful for the evaluation of dichotomic classifiers performance.
- Characterizes a degree of overlap of classes for a single feature.
- Decision is based on a single threshold Θ (called also operating point).

- Generally, the false positive rate (false alarms) go up with attempts to detect higher percentages of true objects (true positive rate, hits).
- Different ROC curves correspond to different classifiers. The single curve is the result of changing threshold Θ.



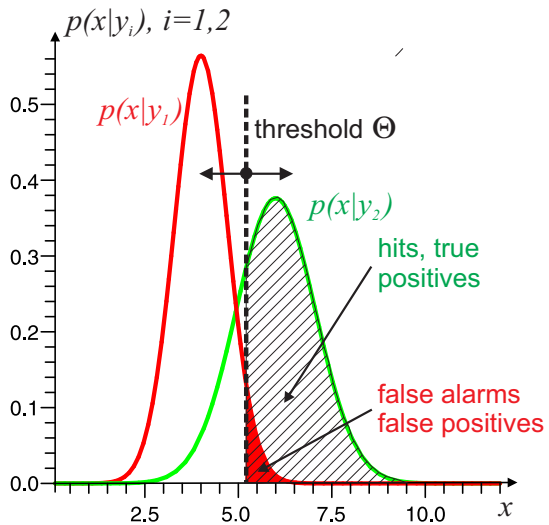
A model problem
Receiver of weak radar signals

- Suppose a receiver detecting a single weak pulse (e.g., a radar reflection from a plane, a dim flash of light).
- A dichotomic decision, two hidden states y_1, y_2:
  - y_1 – a plane is not present (true negative) or
  - y_2 – a plane is present (true positive).
- Assume a simple statistical model – two Gaussians.
- Internal signal of the receiver, voltage x with the mean
  μ_1 when the plane (external signal) is not present and
  μ_2 when the plane is present.
- Random variables due to random noise in the receiver and outside of it. $p(x|y_i) = N(\mu_i, \sigma_i^2)$, $i = 1, 2$.

Four probabilities involved

- Let suppose that the involved probability distributions are Gaussians and the correct decision of the receiver are known.
- The mean values μ_1, μ_2, standard deviations σ_1, σ_2, and the threshold Θ are not known too.
- There are four conditional probabilities involved:
  - Hit (true positive) $p(x > \Theta \,|\, x \in y_2)$.
  - False alarm, type I error (false positive) $p(x > \Theta \,|\, x \in y_1)$.
  - Miss, overlooked danger, type II error (false negative) $p(x < \Theta \,|\, x \in y_2)$.
  - Correct rejection (true negative) $p(x < \Theta \,|\, x \in y_1)$.

Radar receiver example, graphically



Any decision threshold Θ on the voltage x, $x > \Theta$, determines:

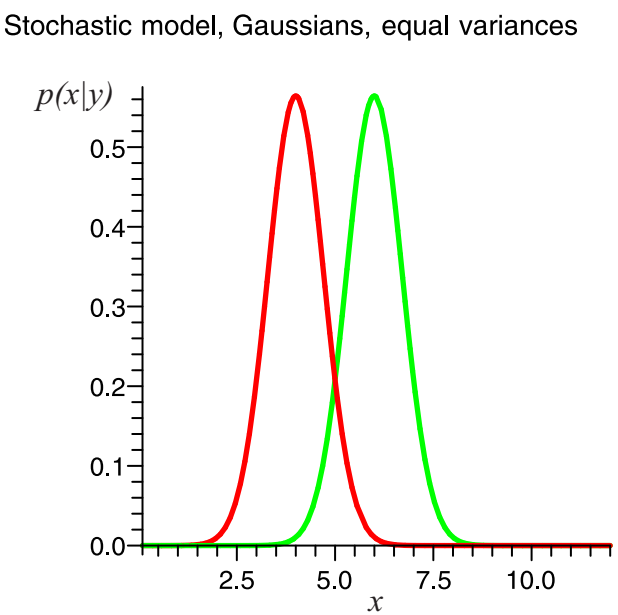
Hits – their probability is the hatched area under the curve $p(x|y_1)$.

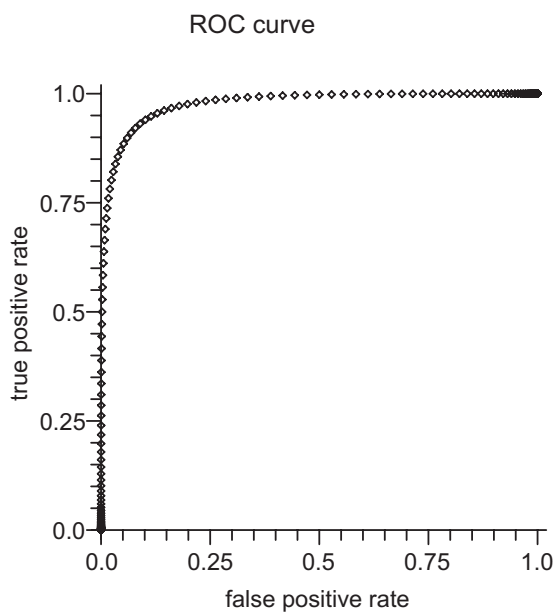
False alarms – their probability is the red filled area under the curve $p(x|y_2)$.

Courtesy to Silvio Maffei for pointing to the mistake in the picture.

ROC – Example A
Two Gaussians with equal variances

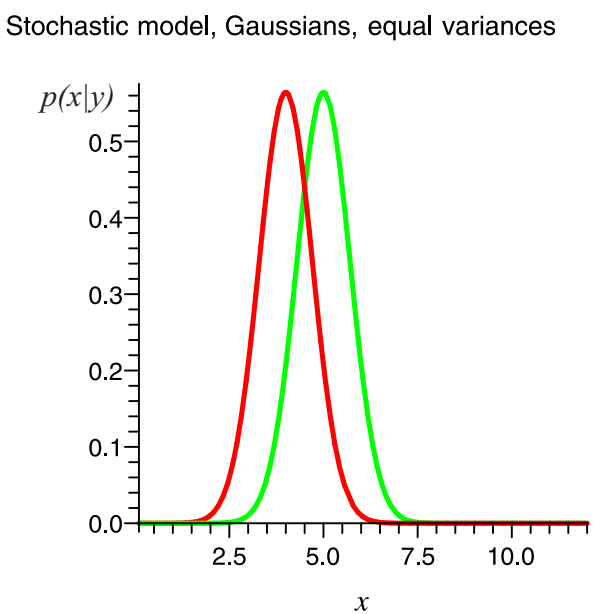
- Two Gaussians, $\mu_1 = 4.0$, $\mu_2 = 6.0$, $\sigma_1 = \sigma_2 = 1.0$
- Less overlap, better discriminability.
- In this special case, ROC is convex.

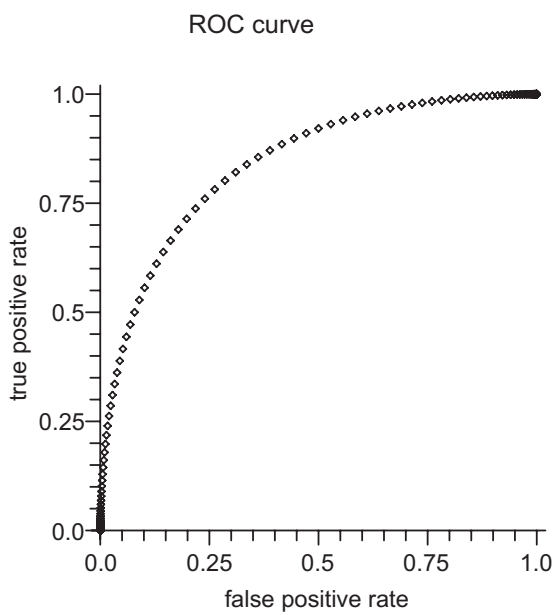




ROC – Example B
Two gaussians with equal variances

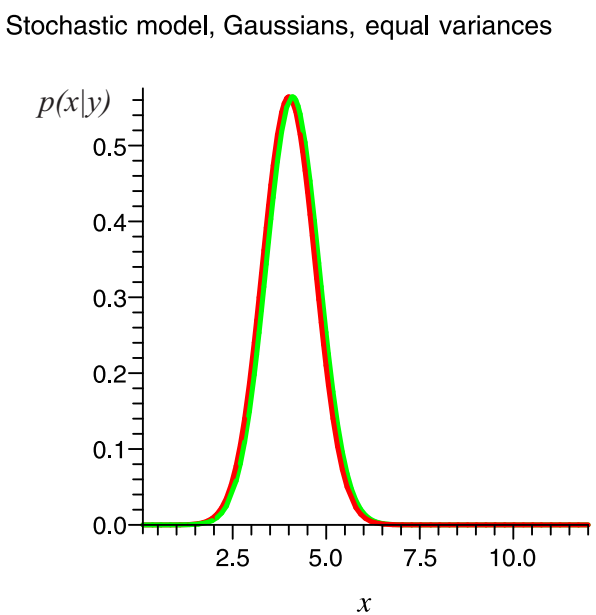
- Two Gaussians, $\mu_1 = 4.0$, $\mu_2 = 5.0$, $\sigma_1 = \sigma_2 = 1.0$
- More overlap, worse discriminability.
- In this special case, ROC is convex.

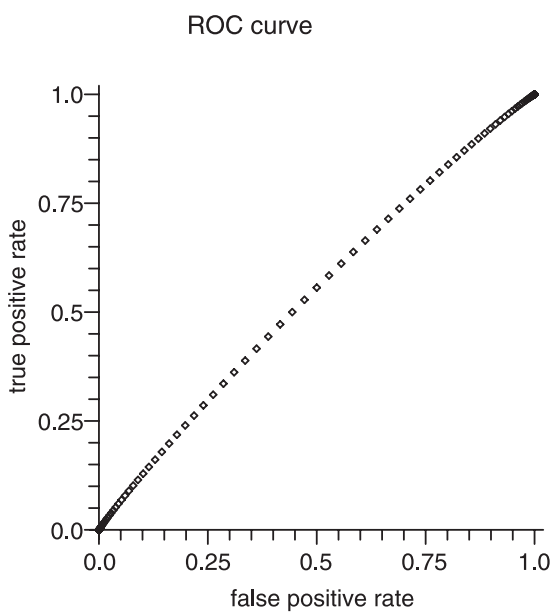




ROC – Example C
Two gaussians with equal variances

- Two Gaussians, $\mu_1 = 4.0$, $\mu_2 = 4.1$, $\sigma_1 = \sigma_2 = 1.0$
- Almost total overlap, almost no discriminability.
- In this special case, ROC is convex.





ROC and the likelihood ratio

- Under the assumption of the Gaussian signal corrupted by the Gaussian noise, the slope of the ROC curve equals to the likelihood ratio

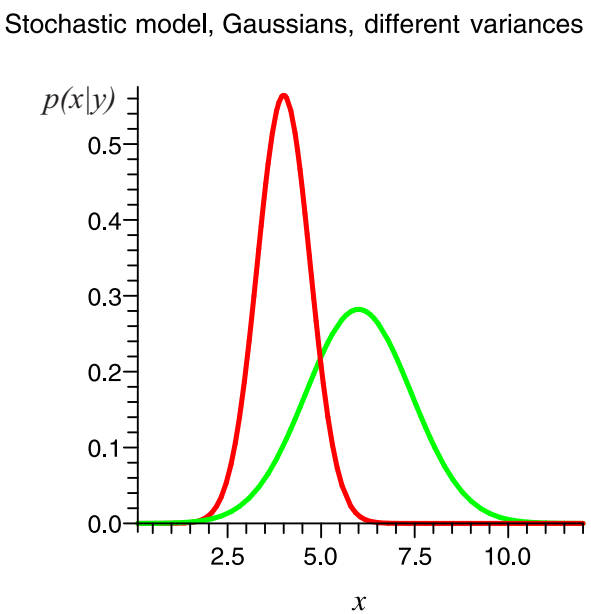
$$L(x) = \frac{p(x|\text{noise})}{p(x|\text{signal})} .$$

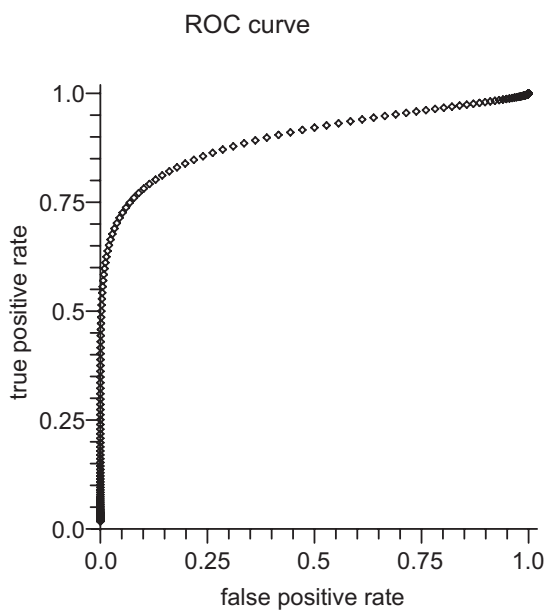
- In the even more special case, when standard deviations $\sigma_1 = \sigma_2$ then
  - $L(x)$ increases monotonically with x. Consequently, ROC becomes a convex curve.
  - The optimal threshold Θ becomes

$$\Theta = \frac{p(\text{noise})}{p(\text{signal})} .$$

ROC – Example D
Two gaussians with different variances

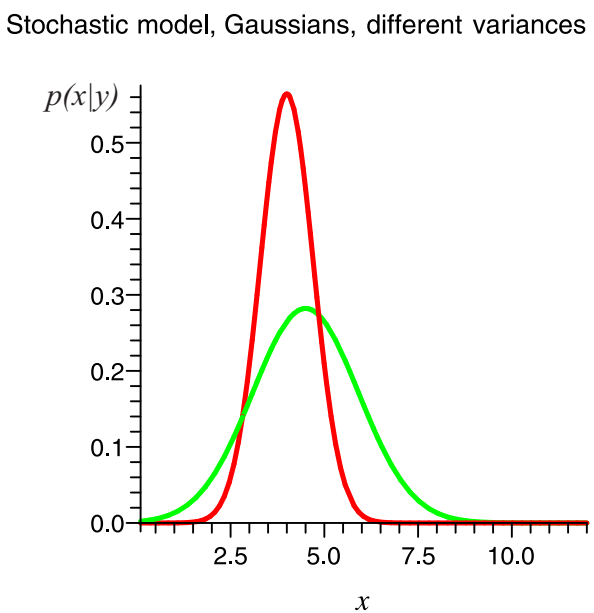
- Two Gaussians, $\mu_1 = 4.0$, $\mu_2 = 6.0$, $\sigma_1 = 1.0$, $\sigma_2 = 2.0$
- Less overlap, better discriminability.
- In general, ROC is not convex.

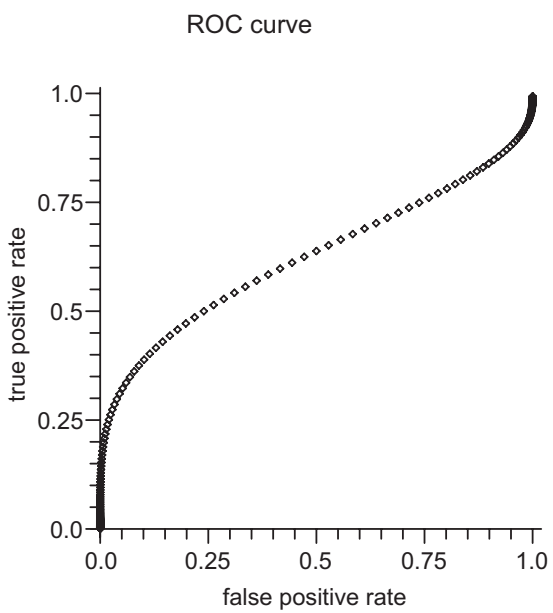




ROC – Example E
Two gaussians with different variances

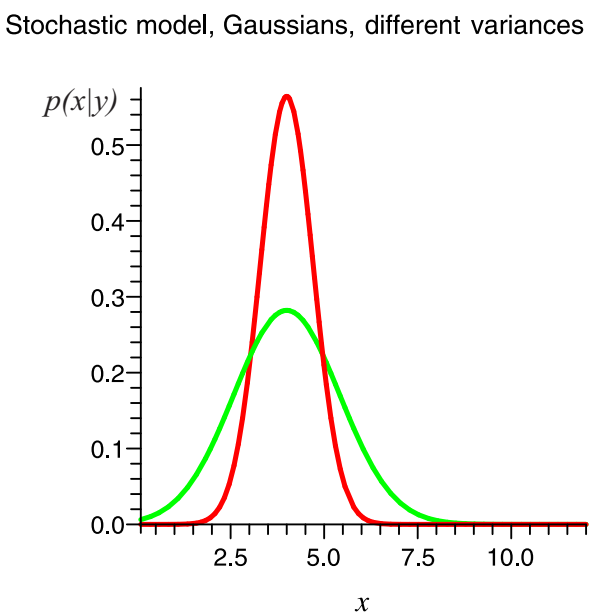
- Two Gaussians, $\mu_1 = 4.0$, $\mu_2 = 4.5$, $\sigma_1 = 1.0$, $\sigma_2 = 2.0$
- More overlap, worse discriminability.
- In general, ROC is not convex.

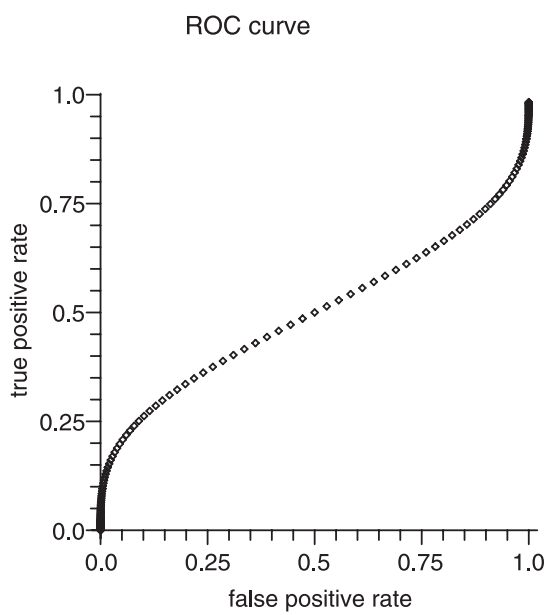




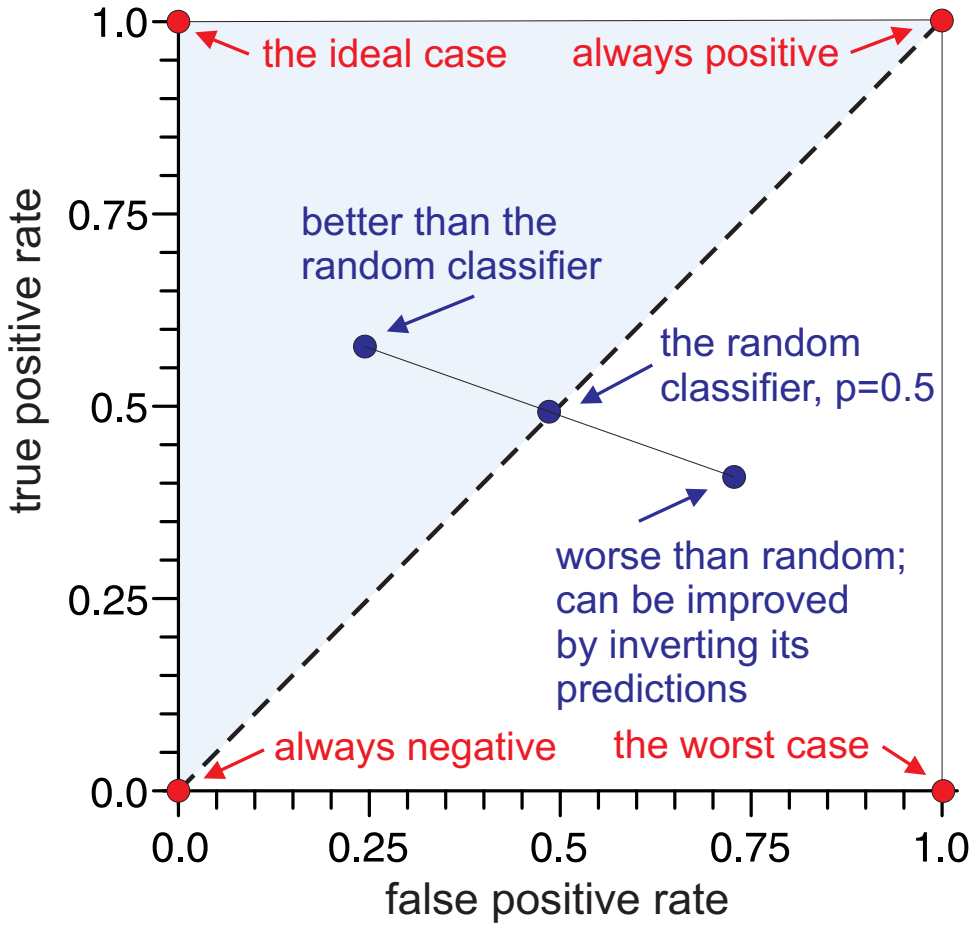
ROC – Example F
Two gaussians with different variances

- Two Gaussians, $\mu_1 = 4.0$, $\mu_2 = 4.0$, $\sigma_1 = 1.0$, $\sigma_2 = 2.0$
- Maximal overlap, the worst discriminability.
- In general, ROC is not convex.





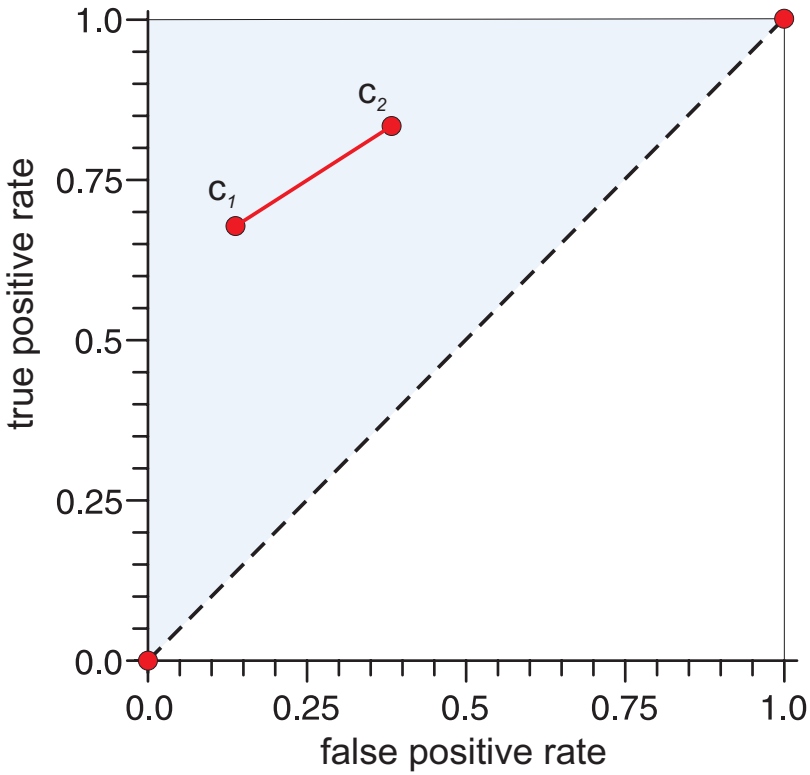
Properties of the ROC space



‘Continuity’ of the ROC

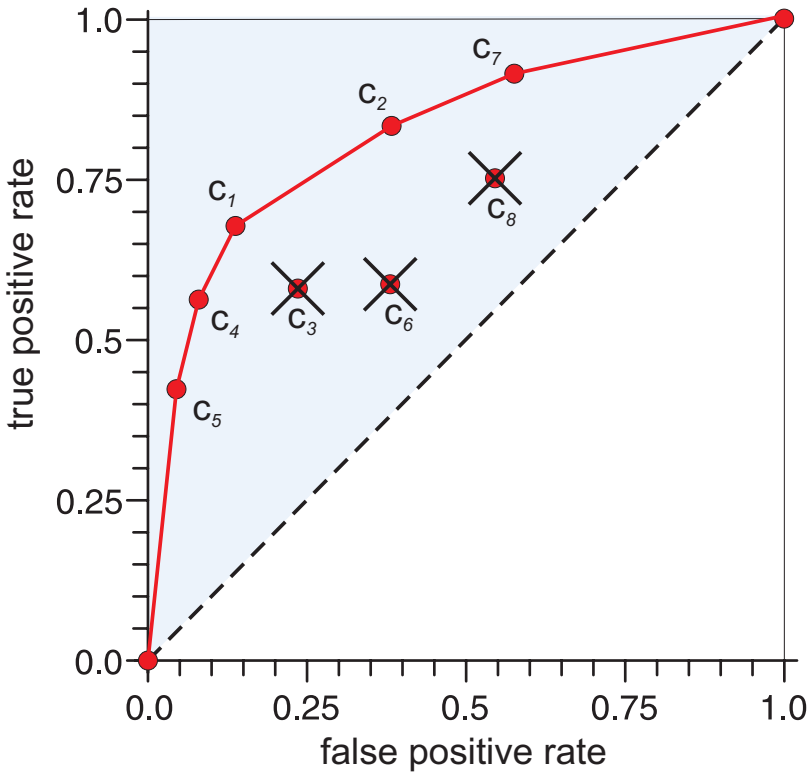
Given two classifiers c_1 and c_2.

- Classifiers c_1 and c_2 can be linearly weighted to create ‘intermediate’ classifiers and
- In such a way, a continuum of classifiers can be imagined which is denoted by a red line in the figure.



More classifiers, ROC construction

- The convex hull covering classifiers in the ROC space is constructed.
- Classifiers on the convex hull achieve always the best performance for some class probability distributions (i.e., the ratio of positive and negative examples).
- Classifiers inside the convex hull perform worse and can be discarded.
- No penalties have been involved so far. To come ...



ROC convex hull and iso-accuracy

- 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 than 45^o), the left classifier is better.
  - For distributions skewed towards positives (flatter slope than 45^o), the right classifier is better.
- Each classifier on convex hull is optimal for a specific range of class distributions.

Accuracy expressed in the ROC space

Accuracy a

$a = pos \cdot TPR + neg \cdot (1 - FPR)$.

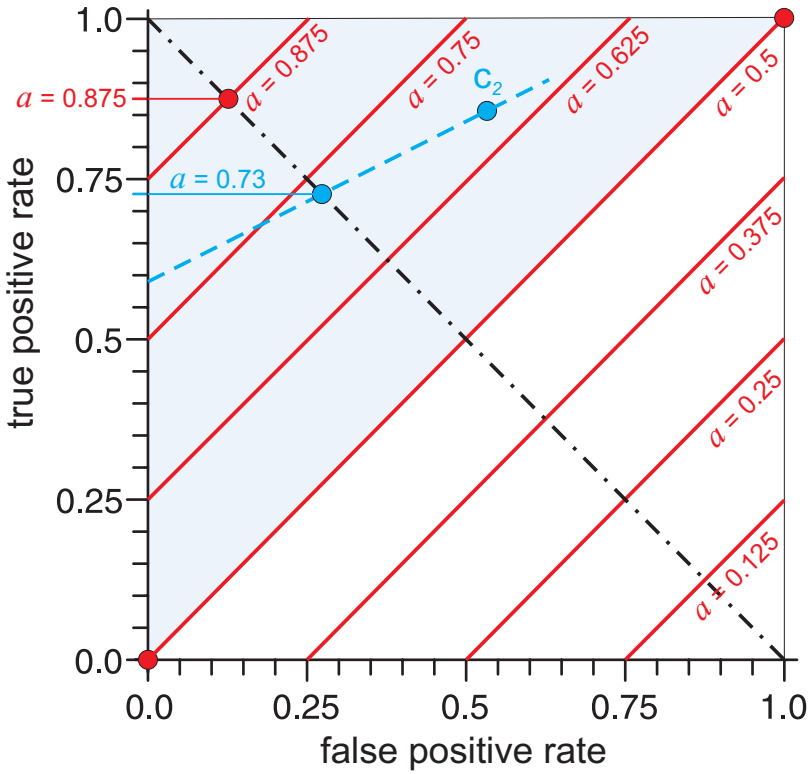
Express the accuracy in the ROC space as $TPR = f(FPR)$.

$$TPR = \frac{a - neg}{pos} + \frac{neg}{pos} \cdot FPR.$$

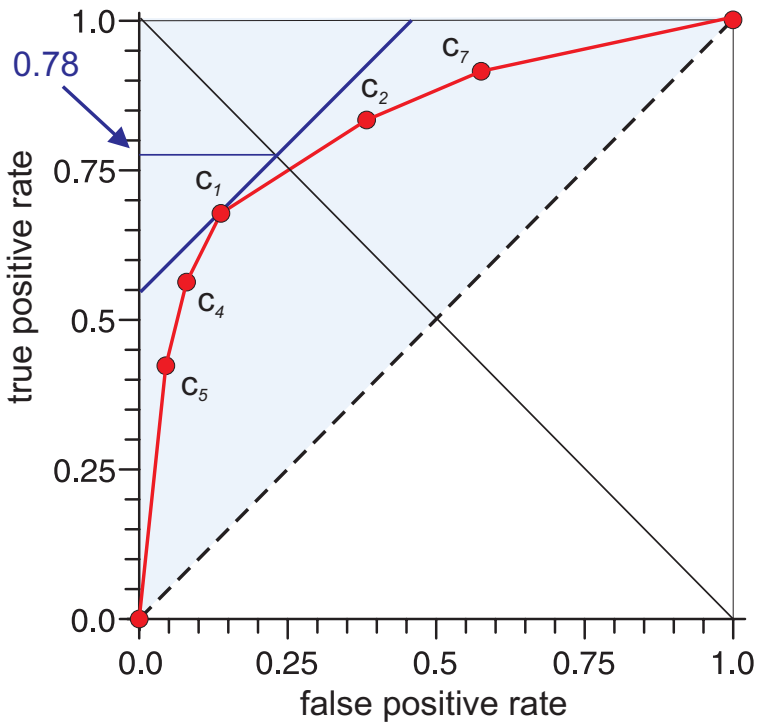
Iso-accuracy are given by straight lines with the slope neg/pos, i.e., by a degree of balance in the test multiset. The balanced multiset $\Leftrightarrow 45^o$.

The diagonal $\sim TPR = FPR = a$.

A blue line is a iso-accuracy line for a general classifier c_2.



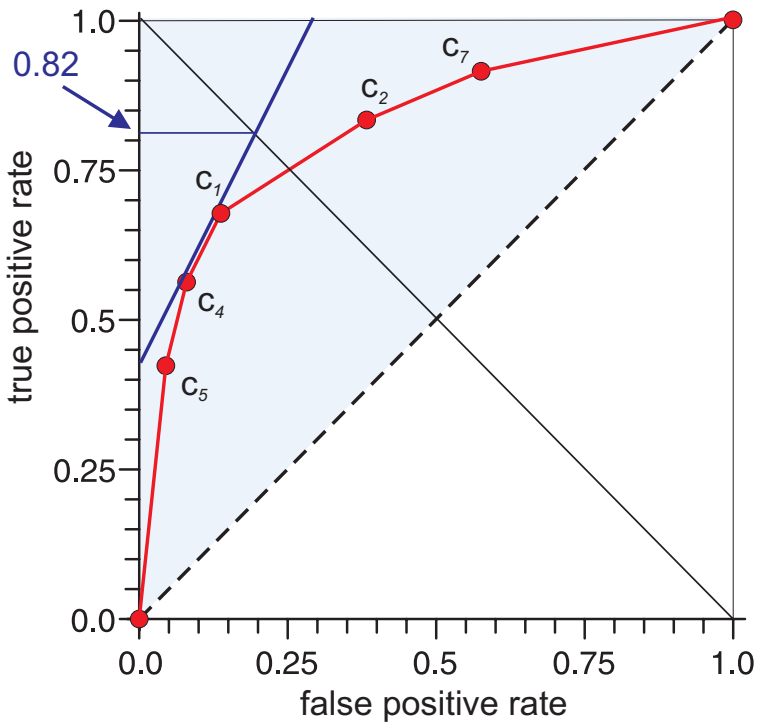
Selecting the optimal classifier (1)



For a balanced training multiset (i.e., as many +ves as -ves), see solid blue line.

Classifier c_1 achieves $\approx$ 78 % true positive rate.

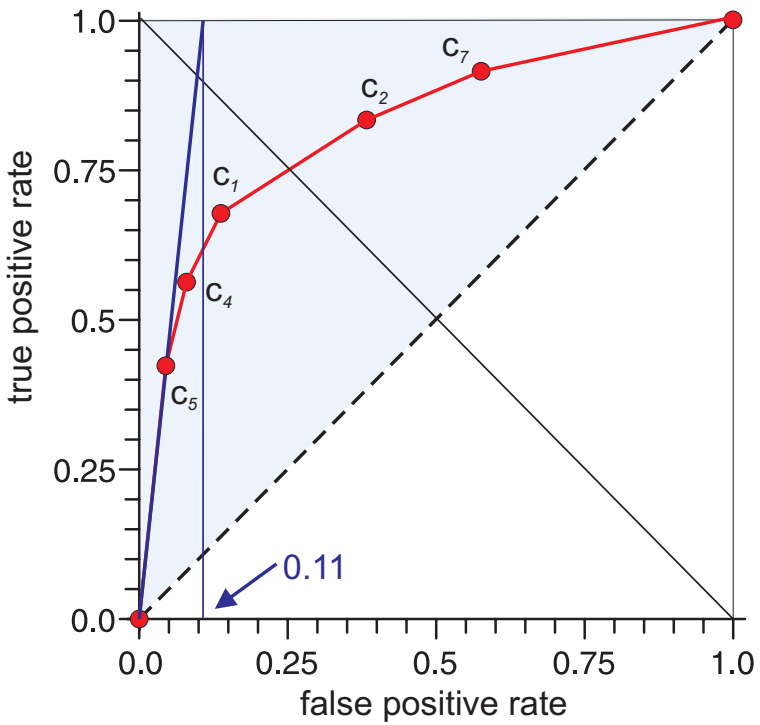
Selecting the optimal classifier (2)



For a training multiset with half +ves than -ves), see solid blue line.

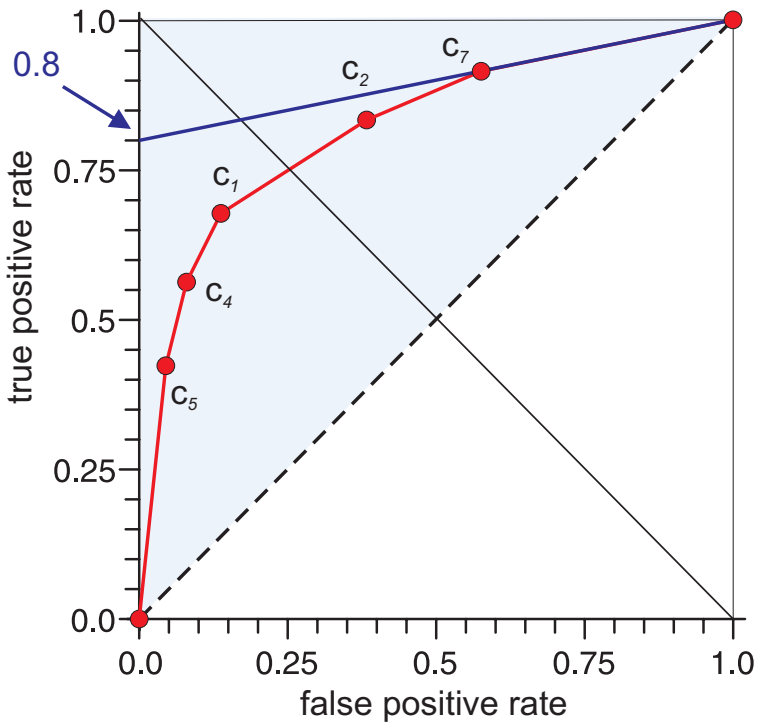
Classifier c_1 achieves $\approx$ 82 % true positive rate.

Selecting the optimal classifier (3)



For less than 11 % +ves, the always negative is the best.

Selecting the optimal classifier (4)



For more than 80 % +ves, the always positive is the best.

Benefit of the ROC approach

- It is possible to distinguish operationally between

  Discriminability – inherent property of the detection system.
  Decision bias – implied by the loss function changable by the user.

- The discriminability can be determined from ROC curve.
- If the Gaussian assumption holds then the Bayesian error rate can be calculated.

ROC – generalization to arbitrary multidimensional distribution

- The approach can be used for any two multidimensional distributions $p(\mathbf{x}|y_1)$, $p(\mathbf{x}|y_2)$ provided they overlap and thus have nonzero Bayes classification error.
- Unlike in one-dimensional case, there may be many decision boundaries corresponding to particular true positive rate, each with different false positive rate.
- Consequently, we cannot determine discriminability from true positive and false positive rate without knowing more about underlying decision rule.

Optimal true/false positive rates? Unfortunately not

- This is a rarely attainable ideal for a multidimensional case.
- We should have found the decision rule of all the decision rules giving the measured true positive rate, the rule which has the minimum false negative rate.
- This would need huge computational resources (like Monte Carlo).
- In practice, we forgo optimality.