

# Advanced Microarray Data Analysis

**Predictive rules: modeling and assessment**

Vlad Popovici  
[vlad.popovici@isb-sib.ch](mailto:vlad.popovici@isb-sib.ch)

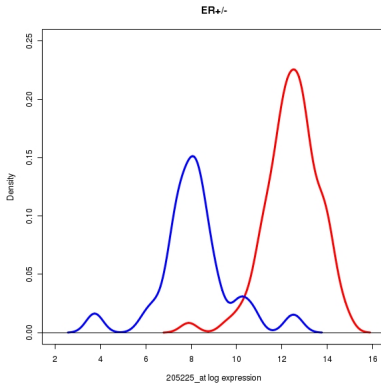
Swiss Institute of Bioinformatics

November 4th, 2009

# Outline

- 1 Introduction
- 2 Building classification rules
  - Statistical framework
  - Discriminant functions
- 3 Assessing the performance
  - Medical application context
  - Measures of accuracy for binary tests
  - Measures of accuracy for continuous tests: ROC and AUC
  - Performance estimation

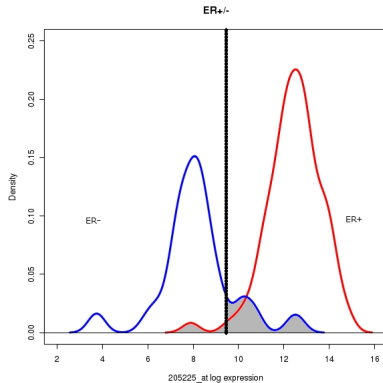
## Example: ER status prediction



Questions:

- How to decide which patient is ER+ and which is ER-?

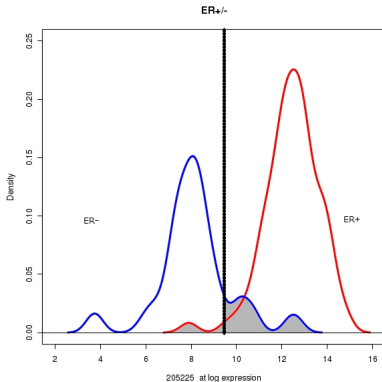
## Example: ER status prediction



Questions:

- How to decide which patient is ER+ and which is ER-?

## Example: ER status prediction

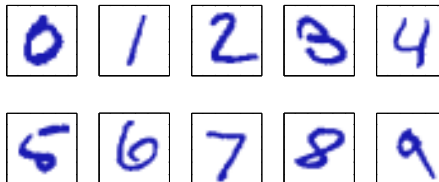


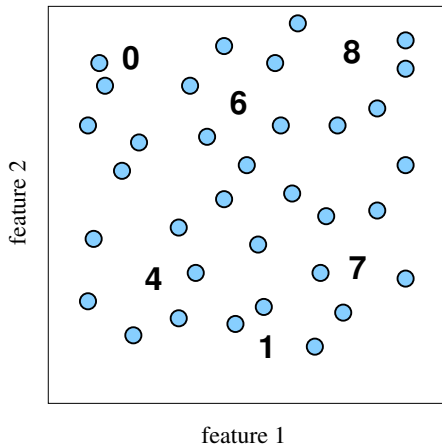
### Questions:

- How to decide which patient is ER+ and which is ER-?
- What is the expected error?
- What if I prefer to detect most of ER+?

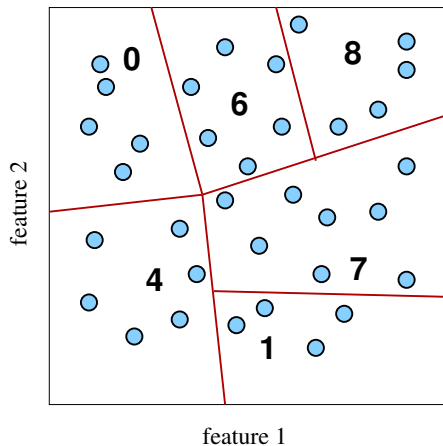
## Another example – multiclass problem

Digit recognition:





- disjoint support regions for each class/concept
- decision boundary



- disjoint support regions for each class/concept
- decision boundary



# Know your problem!

## Remember

Good study  $\longleftrightarrow$  clear objectives.

### Problems:

- *Class Comparison*: find genes differentially expressed between predefined classes;
- *Class Prediction*: predict one of the predefined classes using the gene expressions;
- *Class Discovery*: cluster analysis – define new classes using clusters of genes/specimens.

# Know your problem!

## Remember

Good study  $\longleftrightarrow$  clear objectives.

### Problems:

- *Class Comparison*: find genes differentially expressed between predefined classes;
- *Class Prediction*: predict one of the predefined classes using the gene expressions;
- *Class Discovery*: cluster analysis – define new classes using clusters of genes/specimens.

# Class prediction

Typical applications:

- predict treatment response
- predict patient relapse
- predict the phenotype
- toxico–genomics: predict which chemicals are toxic
- ...

# Class prediction

Typical applications:

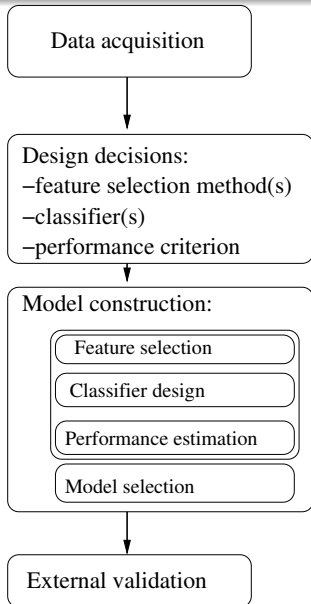
- predict treatment response
- predict patient relapse
- predict the phenotype
- toxico–genomics: predict which chemicals are toxic
- ...

Characteristics:

- *supervised learning*: requires labelled training data
- the goal is prediction accuracy
- uses some measure of similarity
- relies on feature selection
- quite often incorrectly used

## Usage problems:

- improper methodological approach:
  - well fitted model does not ensure good prediction (overfitted model)
  - too many features used in the model (curse of dimensionality)
  - feature selection on the full dataset(!)
- reproducibility:
  - improper/insufficient validation
  - batch effects unaccounted for
  - insufficiently documented
- therapeutic relevance



- Data acquisition: everything up to (and including) normalization
- Design decisions: should be taken before real modeling
- Model design: DO NOT USE ALL DATA AT ONCE!!
- External validation: other datasets; clinical trials: phase II and III

# Outline

- 1 Introduction
- 2 Building classification rules
  - Statistical framework
  - Discriminant functions
- 3 Assessing the performance
  - Medical application context
  - Measures of accuracy for binary tests
  - Measures of accuracy for continuous tests: ROC and AUC
  - Performance estimation

# Classification problem

- **Ingredients:**

- $d$  measurements  $\mathbf{x} = (x_1, \dots, x_d)^T \in \mathbb{R}^d$
- $K$  classes:  $C_1, C_2, \dots, C_K$
- $\pi_k = p(C_k)$  priors: proportion of cases from class  $C_k$  in the population
- $p(\mathbf{x}|C_k)$  (class-conditional) pdf for class  $k$
- **Goal:** given a new object  $\mathbf{x}$ , assign it to one of the  $K$  classes (or reject it)



Let  $\mathbf{x}$  be a (random) feature vector with corresponding class  $C$

- **classifier**: an allocation strategy

$$h : \mathbb{R}^d \rightarrow \{C_1, C_2, \dots, C_K\}$$

Note: in most cases,  $h(\mathbf{x}) = h(y(\mathbf{x}))$

- **probability of misclassification**

$$\text{pmc}(C_k) = \Pr\{h(\mathbf{x}) \neq C_k | C = C_k\}$$

## Measuring errors

- 0 – 1 loss function

$$L(C_k, C_l) = \mathbb{I}[C_k \neq C_l] = \begin{cases} 1 & \text{if } C_k \neq C_l, \\ 0 & \text{otherwise} \end{cases}$$

- risk function

$$\begin{aligned} R(h, C_k) &= E[L(C_k, h(\mathbf{x})) | C = C_k] \\ &= \sum_{l=1}^K L(C_k, C_l) \Pr\{h(\mathbf{x}) = C_l | C = C_k\} \stackrel{0-1 \text{ loss}}{=} \text{pmc}(C_k) \end{aligned}$$

- **total risk**: overall misclassification probability

$$R(h) = E[R(h, C_k)] = \sum_{k=1}^K \pi_k R(h, C_k) \stackrel{0-1 \text{ loss}}{=} \sum_{k=1}^K \pi_k \text{pmc}(C_k)$$

## Quantities of interest

- priors  $\pi_k = p(C_k)$
- class-conditional density  $p(\mathbf{x}|C_k)$
- posterior density  $p(C_k|\mathbf{x})$

### Bayes' theorem

$$p(C_k|\mathbf{x}) = \frac{\pi_k p(\mathbf{x}|C_k)}{p(\mathbf{x})}$$

## Alternative approaches

- model  $p(C_k|\mathbf{x})$  directly: parametric models that are optimized on some training set
- estimate  $\pi_k$  and  $p(\mathbf{x}|C_k)$  to obtain the posterior
- model the *separation boundary* directly

# Bayes optimal classifier

- posterior probability

$$p(C_k|\mathbf{x}) = \Pr\{C = C_k|\mathbf{x}\} = \frac{\pi_k p(\mathbf{x}|C_k)}{\sum_{l=1}^K \pi_l p(\mathbf{x}|C_l)}$$

- optimal classifier (Bayes rule; MAP rule)

$$h_0(\mathbf{x}) = \arg \max_k p(C_k|\mathbf{x})$$

- Bayes risk (under 0 – 1 loss):

$$R(h_0) = \sum_{k=1}^K \pi_k \text{pmc}(h_0)$$

## Plug-in principle

Assume some parametric form for class densities:

$p(\mathbf{x}|C_k) = p(\mathbf{x}|C_k; \theta)$ , with  $\theta \in \Theta$  being the parameters vector.

Then

$$\hat{p}(C_k|\mathbf{x}) = \frac{\pi_k p(\mathbf{x}|C_k; \hat{\theta})}{\sum_{l=1}^K \pi_l p(\mathbf{x}|C_l; \hat{\theta})}.$$

and

$$\hat{h} = \arg \max_k \hat{p}(k|\mathbf{x}).$$

## Bayes classifier for known distributions

Let  $\mathbf{x} \in C_k \sim \mathcal{N}_d(\mu_k, \Sigma)$  :

$$p(\mathbf{x}) = ((2\pi)^d |\Sigma|)^{-\frac{1}{2}} \exp\left(-\frac{1}{2}(\mathbf{x} - \mu_k)^T \Sigma^{-1}(\mathbf{x} - \mu_k)\right) \quad \mathbf{x} \in \mathbb{R}^d$$

## Bayes classifier for known distributions

Let  $\mathbf{x} \in C_k \sim \mathcal{N}_d(\mu_k, \Sigma)$  :

$$p(\mathbf{x}) = ((2\pi)^d |\Sigma|)^{-\frac{1}{2}} \exp\left(-\frac{1}{2}(\mathbf{x} - \mu_k)^T \Sigma^{-1}(\mathbf{x} - \mu_k)\right) \quad \mathbf{x} \in \mathbb{R}^d$$

Bayes rule becomes: assign  $\mathbf{x}$  to the "closest" class:

$$h_0(\mathbf{x}) = \arg \min_k \left\{ (\mathbf{x} - \mu_k)^T \Sigma^{-1}(\mathbf{x} - \mu_k) - 2 \ln \pi_k \right\}$$



## Bayes classifier for known distributions

Let  $\mathbf{x} \in C_k \sim \mathcal{N}_d(\mu_k, \Sigma)$  :

$$p(\mathbf{x}) = ((2\pi)^d |\Sigma|)^{-\frac{1}{2}} \exp\left(-\frac{1}{2}(\mathbf{x} - \mu_k)^T \Sigma^{-1}(\mathbf{x} - \mu_k)\right) \quad \mathbf{x} \in \mathbb{R}^d$$

Bayes rule becomes: assign  $\mathbf{x}$  to the "closest" class:

$$h_0(\mathbf{x}) = \arg \min_k \left\{ (\mathbf{x} - \mu_k)^T \Sigma^{-1}(\mathbf{x} - \mu_k) - 2 \ln \pi_k \right\}$$

**Mahalanobis distance:**

$$\delta(\mathbf{x}, \mu) = \sqrt{(\mathbf{x} - \mu)^T \Sigma^{-1}(\mathbf{x} - \mu)}$$

## Linear discriminant analysis (population version)

$$h_0(\mathbf{x}) = \arg \min_k \left\{ -2\mu_k^T \Sigma^{-1} \mathbf{x} + \mu_k^T \Sigma^{-1} \mu_k - 2 \ln \pi_k \right\}$$

## Linear discriminant analysis (population version)

$$h_0(\mathbf{x}) = \arg \min_k \left\{ -2\mu_k^T \Sigma^{-1} \mathbf{x} + \mu_k^T \Sigma^{-1} \mu_k - 2 \ln \pi_k \right\}$$

For 2 classes (0 – 1 loss),  $\pi_1 + \pi_2 = 1$ :

$$\text{pmc} = \pi_1 \Phi \left( -0.5\delta + \frac{1}{\delta} \ln \frac{\pi_1}{1 - \pi_1} \right) + (1 - \pi_1) \Phi \left( -0.5\delta - \frac{1}{\delta} \ln \frac{\pi_1}{1 - \pi_1} \right)$$

where  $\Phi(\cdot)$  is the CDF for the standard normal and  $\delta = \delta(\mu_1, \mu_2)$ .

# LDA

- optimal solution for  $\mathcal{N}(\mu_k, \Sigma)$  class-conditional densities
- similar solution can be obtained in a distribution-free derivation
- if  $\Sigma = \lambda I$  one obtains the Diagonal LDA
- if classes have different  $\Sigma$ , the boundary is quadratic

# Learning from data

- training set

$$\{(\mathbf{x}_i, \mathbf{t}_i) | i = 1, \dots, N\} \subset \mathbb{R}^d$$

- $\mathbf{t}$  *convenient encoding* of the corresponding class  $C$ .

Examples:

- 3 classes,  $C_1, C_2, C_3$ ,  $\mathbf{x} \in C_2$  a possible encoding:  $\mathbf{t} = (0, 1, 0)^T$
- 2 classes:  $C_1, C_2$ , a possible encoding:  $t = \mathbb{I}[\mathbf{x} \in C_1]$
- **learn from data**: estimate the parameters of the model such that the expected risk of misclassification is minimized

# Outline

- 1 Introduction
- 2 Building classification rules
  - Statistical framework
  - Discriminant functions
- 3 Assessing the performance
  - Medical application context
  - Measures of accuracy for binary tests
  - Measures of accuracy for continuous tests: ROC and AUC
  - Performance estimation

## Two classes, linear functions

Model:

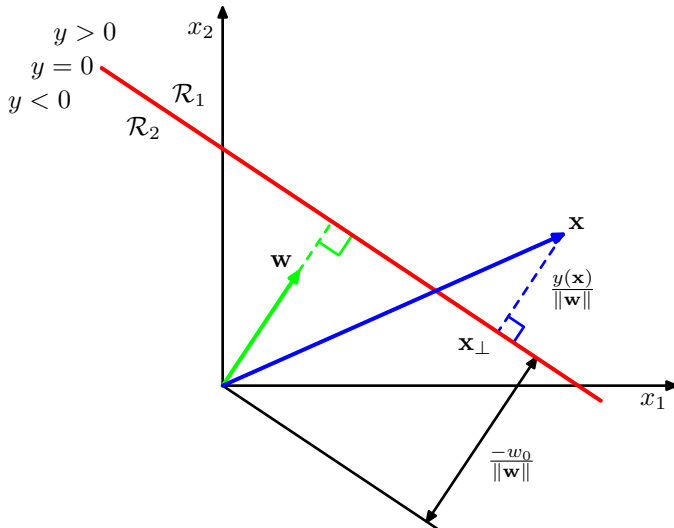
$$y(\mathbf{x}) = \mathbf{w}^T \mathbf{x} + w_0$$

Classification rule:

$$h(\mathbf{x}) = \begin{cases} C_1 & \text{if } y(\mathbf{x}) \geq 0 \\ C_2 & \text{otherwise} \end{cases}$$

Decision boundary:  $y(\mathbf{x}) = 0$  – a  $(d - 1)$ -dimensional hyperplane

# Geometry of the decision surface



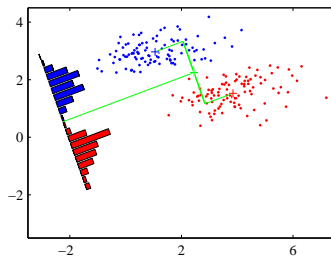
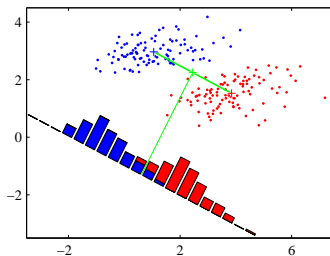


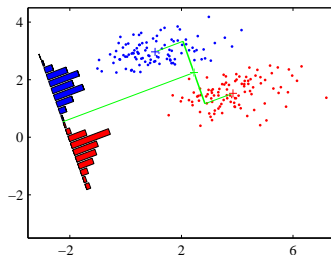
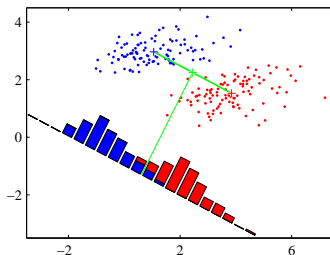
# Fisher's linear discriminant

## Idea

Find a direction  $\mathbf{w}$  such that the projected data  $y_k = \mathbf{w}^T \mathbf{x}_k$  can be classified by applying the rule

$$f(y_k) = \begin{cases} C_1 & \text{if } y_k \geq -w_0 \\ C_2 & \text{otherwise} \end{cases}$$





Fisher's criterion:  $\mathbf{w}^* = \arg \max_{\mathbf{w}} J(\mathbf{w}) = \arg \max_{\mathbf{w}} \frac{\mathbf{w}^T S_B \mathbf{w}}{\mathbf{w}^T S_W \mathbf{w}}$

$$S_B = (\mathbf{m}_2 - \mathbf{m}_1)(\mathbf{m}_2 - \mathbf{m}_1)^T$$

$$S_W = \sum_{i \in C_1} (\mathbf{x}_i - \mathbf{m}_1)(\mathbf{x}_i - \mathbf{m}_1)^T + \sum_{i \in C_2} (\mathbf{x}_i - \mathbf{m}_2)(\mathbf{x}_i - \mathbf{m}_2)^T$$

$$\Rightarrow \mathbf{w}^* \propto S_W^{-1}(\mathbf{m}_2 - \mathbf{m}_1)$$

Note:  $w_0 = ?$

- under normality assumption, leads to a formula as before...
- can be estimated from data

# Logistic regression

$$\begin{aligned} p(C_1|\mathbf{x}) &= \frac{p(\mathbf{x}|C_1)p(C_1)}{p(\mathbf{x}|C_1)p(C_1) + p(\mathbf{x}|C_2)p(C_2)} \\ &= \frac{1}{1 + \exp(-a)} = \sigma(a) \end{aligned}$$

where

$$a = \ln \frac{p(\mathbf{x}|C_1)p(C_1)}{p(\mathbf{x}|C_2)p(C_2)}$$

Assuming  $p(\mathbf{x}|C_k) = \mathcal{N}(\mu_k, \Sigma)$ ,  $k = 1, 2$  then

$$p(C_1|\mathbf{x}) = \sigma(\mathbf{w}^T \mathbf{x} + w_0)$$

where

$$\begin{aligned}\mathbf{w} &= \Sigma^{-1}(\mu_1 - \mu_2) \\ w_0 &= -\frac{1}{2}\mu_1^T \Sigma^{-1} \mu_1 + \frac{1}{2}\mu_2^T \Sigma^{-1} \mu_2 + \ln \frac{p(C_1)}{p(C_2)}\end{aligned}$$

Maximum likelihood estimates for  $\mu, \Sigma$ :

- $\mu_1, \mu_2$  are the usual mean estimates
- $\Sigma = \frac{N_1}{N} S_1 + \frac{N_2}{N} S_2$   
where

$$S_1 = \frac{1}{N_1} \sum_{i \in C_1} (\mathbf{x}_i - \mu_1)(\mathbf{x}_i - \mu_1)^T$$

$$S_2 = \frac{1}{N_2} \sum_{i \in C_2} (\mathbf{x}_i - \mu_2)(\mathbf{x}_i - \mu_2)^T$$

(Compare  $\Sigma$  with  $S_W$  from Fisher's discriminant.)

# Maximum margin classifiers

Let the training set be  $\{(\mathbf{x}_k, t_k) | k = 1, \dots, N\}$ , with  $t_k \in \{\pm 1\}$ .

Consider the model

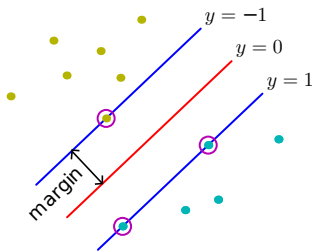
$$y(\mathbf{x}) = \mathbf{w}\phi(\mathbf{x}) + b$$

where  $\phi : \mathbb{R}^d \rightarrow \mathbb{R}^p$  is some fixed feature-space transformation.



Separable case:

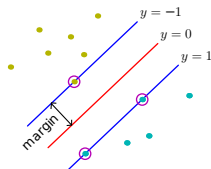
$$\begin{cases} y(\mathbf{x}_k) < 0, & \text{for } t_k = -1 \\ y(\mathbf{x}_k) \geq 0, & \text{for } t_k = +1 \end{cases} \iff t_k \cdot y(\mathbf{x}_k) \geq 0$$



## Idea

(Computational learning theory:) Largest margin leads to smallest generalization error.

$$\arg \max_{\mathbf{w}, b} \left\{ \frac{1}{\|\mathbf{w}\|} \min_n \left[ t_n (\mathbf{w}^T \phi(\mathbf{x}_n) + b) \right] \right\}$$



## Idea

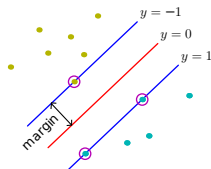
(Computational learning theory:) Largest margin leads to smallest generalization error.

$$\arg \max_{\mathbf{w}, b} \left\{ \frac{1}{\|\mathbf{w}\|} \min_n \left[ t_n(\mathbf{w}^T \phi(\mathbf{x}_n) + b) \right] \right\}$$

$$\Leftrightarrow \arg \min_{\mathbf{w}, b} \frac{1}{2} \|\mathbf{w}\|^2$$

subject to

$$t_n(\mathbf{w}^T \phi(\mathbf{x}_n) + b) \geq 1, \forall n = 1, \dots, N$$



## Important

The quantity

$$t_k \cdot y(\mathbf{x}_k)$$

is called **the (functional) margin** of the point  $\mathbf{x}_k$ .

By introducing the **kernel function**

$$k(\mathbf{x}, \mathbf{x}') = \phi(\mathbf{x}_n)^T \phi(\mathbf{x}')$$

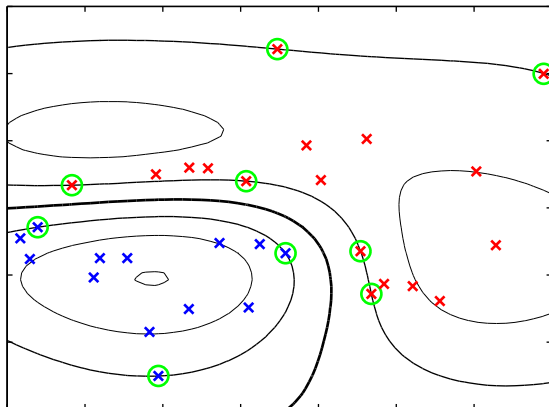
the model can be written as

$$y(\mathbf{x}) = \sum_{n=1}^N a_n t_n k(\mathbf{x}, \mathbf{x}_n) + b,$$

where  $a_n$  are the solution of the dual optimization problem (not discussed here).

Note:  $a_n \neq 0$  correspond to **support vectors**.

## Example (with Gaussian kernel):



# AdaBoost

Consider the *exponential loss function* with the corresponding minimizing criterion:

$$J(y) = E [\exp(-ty(\mathbf{x}))].$$

**Lemma (Friedman, Hastie, Tibshirani)**

$J(y)$  is minimized at

$$y(\mathbf{x}) = \frac{1}{2} \ln \frac{\Pr(t = 1|\mathbf{x})}{\Pr(t = -1|\mathbf{x})}.$$

Consider the space of classifiers (functions) that can be built from a training set:  $\mathcal{H} = \{h_m : \mathbb{R}^d \rightarrow \{-1, 1\}\}$  (*ensemble of classifiers*). We search for an approximation of the solution in the form of

$$y(\mathbf{x}) = \sum_{m=1}^M c_m h_m(\mathbf{x}).$$

The classification is given by  $\text{sign}(y(\mathbf{x}))$ .



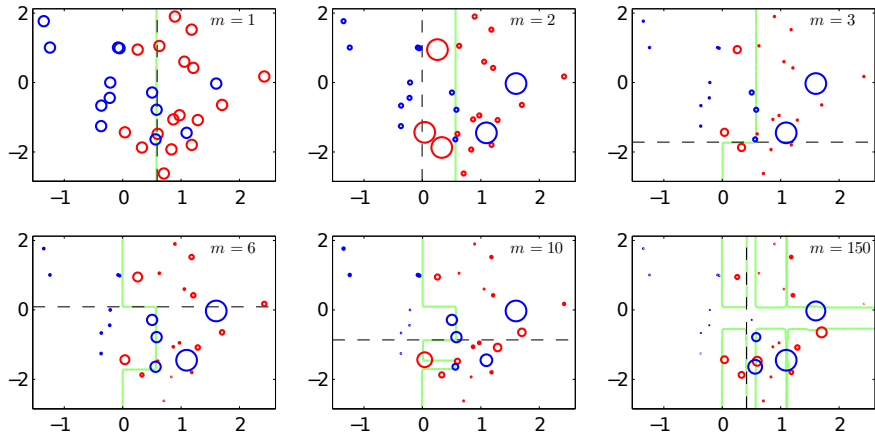
## AdaBoost algorithm (Freund and Schapire)

**Require:** training set  $\{(t_k, \mathbf{x}_k) | k = 1, \dots, N\}$  and number of iterations  $M$

**Ensure:**  $y(\mathbf{x}) = \sum_{m=1}^M c_m h_m(\mathbf{x})$

- 1: start with  $w_i = 1/N, i = 1, \dots, N$
- 2: **for**  $m = 1, \dots, M$  **do**
- 3:   fit the classifier  $h_m(\mathbf{x}) \in \pm 1$ , using the weights  $w_i$  on the training data
- 4:   compute  $err_m = E_w [\mathbb{I}[t \neq h_m(\mathbf{x})]]$ , and let  $c_m = \ln((1 - err_m)/err_m)$
- 5:   set  $w_i \leftarrow w_i \exp(c_m \mathbb{I}[t_i \neq h_m(\mathbf{x}_i)])$ ,  $i = 1, \dots, N$
- 6:   renormalize:  $w_i \leftarrow w_i / \sum_k w_k$
- 7: **end for**

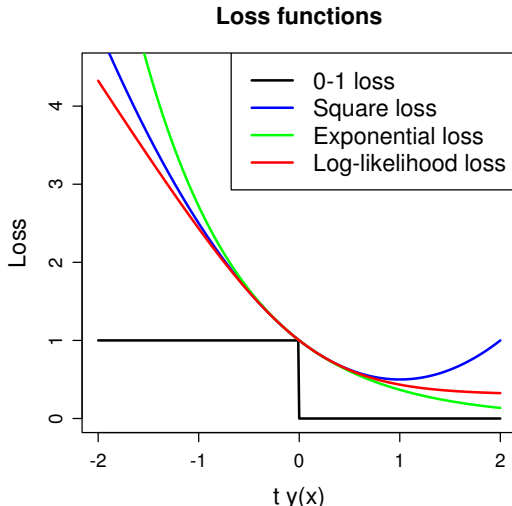
## Example: AdaBoost with decision stumps.



Some properties of AdaBoost-like classifiers:

- converges towards a large margin classifier
- stagewise forward fitting procedure
- very resistant to overfitting
- can be seen as a voting scheme for combining classifiers

# A word about loss functions



## Some references

- ① Bishop, *Pattern recognition and machine learning*, Springer 2006
- ② Kuncheva, *Combining pattern classifiers*, Wiley 2005
- ③ Ripley, *Pattern recognition and neural networks*, CUP 2006
- ④ Hastie, Tibshirani, Friedman, *The elements of statistical learning*, Springer 2001

# Outline

- 1 Introduction
- 2 Building classification rules
  - Statistical framework
  - Discriminant functions
- 3 Assessing the performance**
  - Medical application context**
  - Measures of accuracy for binary tests
  - Measures of accuracy for continuous tests: ROC and AUC
  - Performance estimation

## Context

- classifier  $\leftrightarrow$  test
- a score  $y(\mathbf{X})$  is assigned to a *vector of measurements*  $\mathbf{X}$
- the score can be *categorical* (e.g. *binary*), *ordinal*, or *continuous*
- a *prediction* is made based on the score
- convention (for binary tests): positive test  $\equiv$  diseased

## Medical tests

- **diagnostic**: detect the 'diseased' condition in a patient
- **prognostic**: predict a clinical outcome of interest (e.g. 'recurrence' vs. 'no-recurrence')
- **screening**: a diagnostic test applied to a large population of healthy individuals (low prevalence); it is followed by other tests



# Elements to consider when designing a study

- **test result scale:** binary, ordinal, continuous
- **sampling strategy:**
  - *case–control:* a fixed number of cases (diseased) and control (healthy) patients are selected;
  - *cohort:* a set of subjects is selected from the target population; true disease status must be known by other means;
  - *cohort with selection:* ascertainment of true disease status is conditioned by the test result.

# Elements to consider when designing a study

- **comparing tests:**
  - *paired design*: the tests are applied to the same subjects (correlations b/w test results; check that tests do not interfere);
  - *unpaired design*: each test unit is subject to a single test
- **integrity of the tests**: knowledge of the true disease status does not influence the assessment of the test (e.g. use blinded design)

## Elements to consider when designing a study

Be aware of potential sources of bias:

- verification bias: non-random selection of the test set;
- errors in gold standard
- spectrum bias: test set not representative for the population;
- for screening tests:
  - lead-time bias: earlier detection might indicate erroneously beneficial effects on the outcome;
  - length-bias: slowly progressing diseases are over-represented in the population relatively to all cases;
  - overdiagnosis bias: sub-clinical cases may regress and never become a clinical problem, but they are nevertheless detected

# Outline

- 1 Introduction
- 2 Building classification rules
  - Statistical framework
  - Discriminant functions
- 3 **Assessing the performance**
  - Medical application context
  - **Measures of accuracy for binary tests**
  - Measures of accuracy for continuous tests: ROC and AUC
  - Performance estimation

# Confusion matrix

Let

- the true label (disease status):

$$D = \begin{cases} 0 & \text{if non-diseased} \\ 1 & \text{if diseased} \end{cases}$$

- the predicted label:

$$Y = \begin{cases} 0 & \text{if negative for non-diseased} \\ 1 & \text{if positive for disease} \end{cases}$$

For continuous tests,  $Y = \mathbb{I}[y(\mathbf{X}) \geq \theta]$ .

## Confusion matrix

	Gold standard	
	$D = 0$	$D = 1$
$Y = 0$	true negative	false negative
$Y = 1$	false positive	true positive

# Confusion matrix

	Gold standard		
	$D = 0$	$D = 1$	
$Y = 0$	true negative $P[Y = 0 D = 0]$	false negative $P[Y = 0 D = 1]$	$P[Y = 0]$
$Y = 1$	false positive $P[Y = 1 D = 0]$	true positive $P[Y = 1 D = 1]$	$P[Y = 1]$
	$P[D = 0]$	$P[D = 1]$	

- false positive fraction: FPF (aka 1-Specificity)
- true positive fraction: TPF (aka Sensitivity)
- PPV = Positive Predicted Value
- NPV = Negative Predicted Value

# Confusion matrix

	Gold standard		
	$D = 0$	$D = 1$	
$Y = 0$	true negative $P[Y = 0 D = 0]$	false negative $P[Y = 0 D = 1]$	$P[Y = 0]$
$Y = 1$	false positive $P[Y = 1 D = 0]$	true positive $P[Y = 1 D = 1]$	$P[Y = 1]$
	$P[D = 0]$	$P[D = 1]$	

## Goal

Estimate conditional and marginal probabilities.



## Disease-centric measures

- true/false positive fractions:

$$\text{TPF} = P[Y = 1|D = 1], \quad \text{FPF} = P[Y = 1|D = 0]$$

- alternatively, from a hypothesis testing perspective ( $H_0 : D = 0$ ): FPF: significance and TPF: statistical power of the test
- let the *prevalence* be  $\rho = P[D = 1]$ , then the probability of error is

$$P[Y \neq D] = \rho(1 - \text{TPF}) + (1 - \rho) \text{FPF}$$

- are both needed to characterize the performance
- are independent of prevalence (but complete description of the performance needs the prevalence – the marginal  $P[D = 1]$ )

## Predicted values

$$\text{NPV} = P[D = 0|Y = 0], \text{PPV} = P[D = 1|Y = 1]$$

- quantify the clinical value of the test: likelihood of disease when tested positive
- perfect test:  $\text{PPV} = \text{NPV} = 1$ ; totally uninformative test:  $\text{PPV} = \rho, \text{NPV} = 1 - \rho$

Also,

$$\text{PPV} = \frac{\rho \text{ TPF}}{\rho \text{ TPF} + (1 - \rho) \text{ FPF}} \quad \text{NPV} = \frac{(1 - \rho)(1 - \text{FPF})}{(1 - \rho)(1 - \text{FPF}) + \rho(1 - \text{TPF})}$$

## Likelihood ratios

- positive diagnostic likelihood ratio:  $\text{DLR}^+ = \frac{P[Y=1|D=1]}{P[Y=1|D=0]}$
- negative diagnostic likelihood ratio:  $\text{DLR}^- = \frac{P[Y=0|D=1]}{P[Y=0|D=0]}$

## Likelihood ratios

- positive diagnostic likelihood ratio:  $DLR^+ = \frac{P[Y=1|D=1]}{P[Y=1|D=0]}$
- negative diagnostic likelihood ratio:  $DLR^- = \frac{P[Y=0|D=1]}{P[Y=0|D=0]}$
- quantify the increase in knowledge about the presence of the disease that is gained through the diagnostic test:

## Likelihood ratios

- positive diagnostic likelihood ratio:  $DLR^+ = \frac{P[Y=1|D=1]}{P[Y=1|D=0]}$
- negative diagnostic likelihood ratio:  $DLR^- = \frac{P[Y=0|D=1]}{P[Y=0|D=0]}$
- quantify the increase in knowledge about the presence of the disease that is gained through the diagnostic test:
  - pre-test odds (of having the disease):  $\frac{P[D=1]}{P[D=0]} = \frac{\rho}{1-\rho}$

## Likelihood ratios

- positive diagnostic likelihood ratio:  $DLR^+ = \frac{P[Y=1|D=1]}{P[Y=1|D=0]}$
- negative diagnostic likelihood ratio:  $DLR^- = \frac{P[Y=0|D=1]}{P[Y=0|D=0]}$
- quantify the increase in knowledge about the presence of the disease that is gained through the diagnostic test:
  - pre-test odds (of having the disease):  $\frac{P[D=1]}{P[D=0]} = \frac{\rho}{1-\rho}$
  - post-test odds: for  $Y = 1$ : pre-test odds  $\times DLR^+$ ; for  $Y = 0$ : pre-test odds  $\times DLR^-$

## Likelihood ratios

- positive diagnostic likelihood ratio:  $\text{DLR}^+ = \frac{P[Y=1|D=1]}{P[Y=1|D=0]}$
- negative diagnostic likelihood ratio:  $\text{DLR}^- = \frac{P[Y=0|D=1]}{P[Y=0|D=0]}$
- quantify the increase in knowledge about the presence of the disease that is gained through the diagnostic test:
  - pre-test odds (of having the disease):  $\frac{P[D=1]}{P[D=0]} = \frac{\rho}{1-\rho}$
  - post-test odds: for  $Y = 1$ : pre-test odds  $\times \text{DLR}^+$ ; for  $Y = 0$ : pre-test odds  $\times \text{DLR}^-$
- $\text{DLR}^+ = \frac{\text{TPF}}{\text{FPF}}$
- $\text{DLR}^- = \frac{1-\text{TPF}}{1-\text{FPF}}$

# Notations

Sample:  $\{(D_i, Y_i) \mid i = 1, \dots, n\}$

		Gold standard		
		$D = 0$	$D = 1$	
Confusion matrix:	$Y = 0$	$n_{\bar{D}}^-$	$n_D^-$	$n^-$
	$Y = 1$	$n_{\bar{D}}^+$	$n_D^+$	$n^+$
		$n_{\bar{D}}$	$n_D$	



## Cohort studies

Estimation and inference for (FPF, TPF) and (PPV, NPV):

- the sample is randomly selected from the population, iid
- → the sample approximates the true proportions

# Cohort studies

Estimation and inference for (FPF, TPF) and (PPV, NPV):

- the sample is randomly selected from the population, iid
- $\rightarrow$  the sample approximates the true proportions
- $\hat{TPF} = \frac{n_D^+}{n_D^+ + n_D^-}$ ,  $\hat{FPF} = \frac{n_{\bar{D}}^+}{n_{\bar{D}}^+ + n_{\bar{D}}^-}$

## Cohort studies

Estimation and inference for (FPF, TPF) and (PPV, NPV):

- the sample is randomly selected from the population, iid
- $\rightarrow$  the sample approximates the true proportions
- $\hat{TPF} = \frac{n_D^+}{n_D^+ + n_D^-}$ ,  $\hat{FPF} = \frac{n_{\bar{D}}^+}{n_{\bar{D}}^+ + n_{\bar{D}}^-}$
- $\hat{PPV} = \frac{n_D^+}{n_D^+ + n_{\bar{D}}^+}$ ,  $\hat{NPV} = \frac{n_{\bar{D}}^-}{n_{\bar{D}}^- + n_D^-}$

## Cohort studies

Estimation and inference for (FPF, TPF) and (PPV, NPV):

- the sample is randomly selected from the population, iid
- $\rightarrow$  the sample approximates the true proportions
- $\hat{TPF} = \frac{n_D^+}{n_D^+ + n_D^-}$ ,  $\hat{FPF} = \frac{n_{\bar{D}}^+}{n_{\bar{D}}^+ + n_{\bar{D}}^-}$
- $\hat{PPV} = \frac{n_D^+}{n_D^+ + n_{\bar{D}}^+}$ ,  $\hat{NPV} = \frac{n_{\bar{D}}^-}{n_{\bar{D}}^- + n_D^-}$
- these estimators are random variables from a Bernoulli trial
- $\rightarrow$  CIs can be obtained from binomial distribution

- *Bernoulli trial*: experiment with a random binary outcome
- *binomial distribution*: discrete pdf of the number of successes in  $n$  independent Bernoulli trials with success probability  $p$
- $X \sim \mathcal{B}(n, p)$  :

$$P[X = k] = \binom{n}{k} p^k (1 - p)^{n-k}$$

$$E[X] = np$$

$$\text{Var}[X] = np(1 - p)$$

- as  $n \rightarrow \infty$ ,

$$\frac{X - np}{\sqrt{np(1 - p)}} \sim \mathcal{N}(0, 1)$$

- simplest CI: normal approximation
- other formulas for CI: Wilson score intervals; Clopper–Pearson interval; Agresti-Coull
- Bayesian CIs

- simplest CI: normal approximation
- other formulas for CI: Wilson score intervals; Clopper–Pearson interval; Agresti-Coull
- Bayesian CIs

### Warning

The normal approximation is poor for FPF or TPF close to 0 or 1.

Example: a test for predicting pCR in breast cancer yields

	pCR=0	pCR=1
predicted 0	61	5
predicted 1	24	10

$$\hat{T}PF = 0.67, \quad \hat{F}PF = 0.28$$

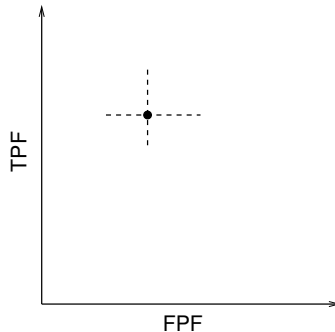
$$\hat{P}PV = 0.29, \quad \hat{N}PV = 0.92$$

95% confidence intervals: (Note: in R, use the package *binom*.)

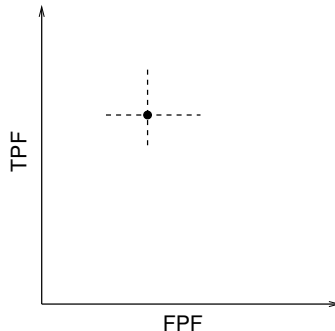
- normal approx.:  $\hat{F}PF \in (0.197, 0.391)$ ,  $\hat{T}PF \in (0.428, 0.905)$
- Wilson:  $\hat{F}PF \in (0.208, 0.398)$ ,  $\hat{T}PF \in (0.417, 0.848)$
- Bayesian:  $\hat{F}PF \in (0.205, 0.397)$ ,  $\hat{T}PF \in (0.416, 0.860)$



# Joint confidence intervals

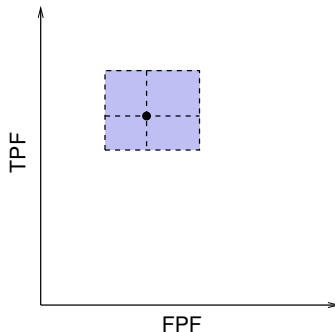


## Joint confidence intervals



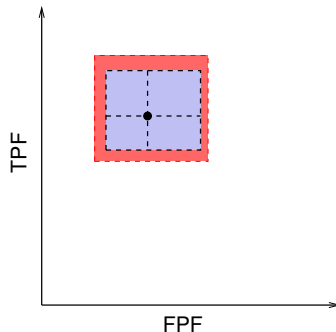
- what is the joint  $100(1 - \alpha)\%$  confidence region for (FPF, TPF)?

## Joint confidence intervals



- what is the joint  $100(1 - \alpha)\%$  confidence region for (FPF, TPF)?

## Joint confidence intervals



- what is the joint  $100(1 - \alpha)\%$  confidence region for (FPF, TPF)?

## Rectangular confidence regions

If  $(P_{low}, P_{up})$  and  $(Q_{low}, Q_{up})$  are the  $1 - \alpha^*$  univariate confidence intervals for two binomial random variables  $P$  and  $Q$ , then the rectangle

$$R \equiv (P_{low}, P_{up}) \times (Q_{low}, Q_{up})$$

is a  $(1 - \alpha)$  confidence region for  $(P, Q)$ , where  $\alpha = 1 - (1 - \alpha^*)^2$ .

## Rectangular confidence regions

If  $(P_{low}, P_{up})$  and  $(Q_{low}, Q_{up})$  are the  $1 - \alpha^*$  univariate confidence intervals for two binomial random variables  $P$  and  $Q$ , then the rectangle

$$R \equiv (P_{low}, P_{up}) \times (Q_{low}, Q_{up})$$

is a  $(1 - \alpha)$  confidence region for  $(P, Q)$ , where  $\alpha = 1 - (1 - \alpha^*)^2$ .

Examples:

- 95% univariate CI lead to a 90.25% confidence region
- for a 95% confidence region, 97.5% univariate CIs are needed

## Case-control studies

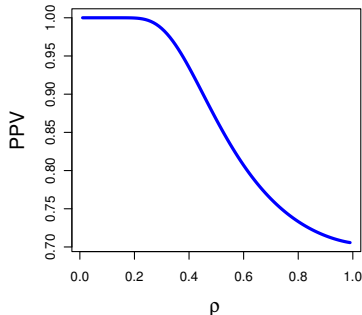
- sampling is done based on the actual disease status
- the sample **does not** necessarily reflect the true disease prevalence
- (FPF, TPF) and ( $\text{DLR}^+$ ,  $\text{DLR}^-$ ) are estimated as before

Assuming that there is an estimate for the prevalence  $\rho$ ,

$$\text{logit PPV} = \text{logit } \rho + \ln \text{DLR}^+$$

$$\text{logit NPV} = \text{logit } \frac{1}{\rho} + \ln \frac{1}{\text{DLR}^-},$$

where  $\text{logit}(x) = \ln \frac{x}{1-x}$ .





## Other performance measures

- Accuracy =  $1 - P[Y \neq D] = \frac{n_D^- + n_D^+}{n} = \rho \text{ Se} + (1 - \rho) \text{ Sp}$
- Matthew's correlation coefficient

$$\text{MCC} = \frac{n_D^+ n_D^- - n_D^+ n_D^-}{[(n_D^+ + n_D^+)(n_D^+ + n_D^-)(n_D^- + n_D^+)(n_D^- + n_D^-)]^{0.5}}$$

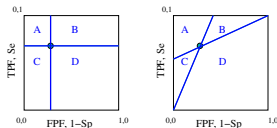
- MSE,  $\kappa$ , etc., etc.

# Outline

- 1 Introduction
- 2 Building classification rules
  - Statistical framework
  - Discriminant functions
- 3 **Assessing the performance**
  - Medical application context
  - Measures of accuracy for binary tests
  - **Measures of accuracy for continuous tests: ROC and AUC**
  - Performance estimation

## A motivating example

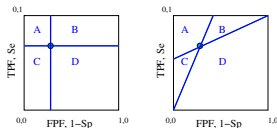
Using (FPF, TPF) or ( $\text{DLR}^-$ ,  $\text{DLR}^+$ ) for comparing tests:



- single point performance measure: partition the space in 4 regions

## A motivating example

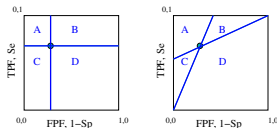
Using  $(\text{FPF}, \text{TPF})$  or  $(\text{DLR}^-, \text{DLR}^+)$  for comparing tests:



- single point performance measure: partition the space in 4 regions
- region A: better than current test

## A motivating example

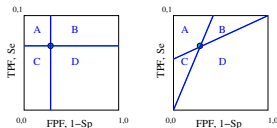
Using  $(\text{FPF}, \text{TPF})$  or  $(\text{DLR}^-, \text{DLR}^+)$  for comparing tests:



- single point performance measure: partition the space in 4 regions
- region A: better than current test
- region D: worse than current test

## A motivating example

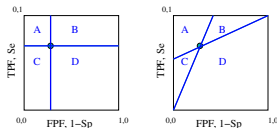
Using  $(\text{FPF}, \text{TPF})$  or  $(\text{DLR}^-, \text{DLR}^+)$  for comparing tests:



- single point performance measure: partition the space in 4 regions
- region A: better than current test
- region D: worse than current test
- regions B,C: less clear

## A motivating example

Using  $(\text{FPF}, \text{TPF})$  or  $(\text{DLR}^-, \text{DLR}^+)$  for comparing tests:



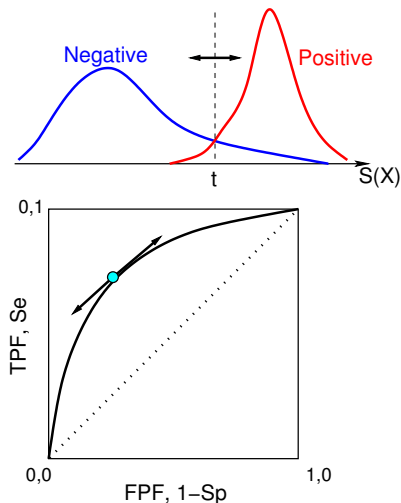
- single point performance measure: partition the space in 4 regions
- region A: better than current test
- region D: worse than current test
- regions B,C: less clear
- note that regions B,C for DLRs are smaller

Other issues with single point performance metrics:

- difficulty in selecting the optimal threshold: different context may need different *operating regimes*
- additive batch effects may spoil the single-point performance

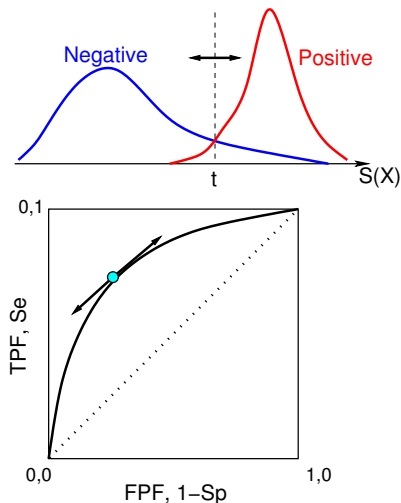


# ROC curves: Theory



- continuous test score  
 $Y = y(\mathbf{X})$
- $FPF(t) = P[Y \geq t | D = 0]$
- $TPF(t) = P[Y \geq t | D = 1]$
- $ROC = \{(FPF(t), TPF(t)) | \forall t \in \mathbb{R}\}$

# ROC curves: Theory



- continuous test score  
 $Y = y(\mathbf{X})$
- $FPF(t) = P[Y \geq t | D = 0]$
- $TPF(t) = P[Y \geq t | D = 1]$
- ROC =  
 $\{(FPF(t), TPF(t)) | \forall t \in \mathbb{R}\}$
- $\lim_{t \rightarrow \infty} FPF(t) =$   
 $\lim_{t \rightarrow \infty} TPF(t) = 0$
- $\lim_{t \rightarrow -\infty} FPF(t) =$   
 $\lim_{t \rightarrow -\infty} TPF(t) = 1$

## Properties of the ROC curves:

- monotone increasing function

## Properties of the ROC curves:

- monotone increasing function
- ROC curve is invariant to strictly increasing transformations of the scores  $Y = y(\mathbf{X})$

## Properties of the ROC curves:

- monotone increasing function
- ROC curve is invariant to strictly increasing transformations of the scores  $Y = y(\mathbf{X})$
- parametric model:

$$\text{ROC} = \{(\alpha, \text{TPF}(\text{FPF}^{-1}(\alpha))) | \forall \alpha \in (0, 1)\}$$

## Properties of the ROC curves:

- monotone increasing function
- ROC curve is invariant to strictly increasing transformations of the scores  $Y = y(\mathbf{X})$
- parametric model:

$$\text{ROC} = \{(\alpha, \text{TPF}(\text{FPF}^{-1}(\alpha))) | \forall \alpha \in (0, 1)\}$$

- $\text{ROC}(0) = 0$ ,  $\text{ROC}(1) = 1$ , and

$$\frac{\partial \text{ROC}(t)}{\partial t} = \frac{f_D(\text{FPF}^{-1}(t))}{f_{\bar{D}}(\text{FPF}^{-1}(t))},$$

where  $f_D$  and  $f_{\bar{D}}$  are the probability densities of the scores within diseased and healthy populations, respectively.

## Properties of the ROC curves:

- monotone increasing function
- ROC curve is invariant to strictly increasing transformations of the scores  $Y = y(\mathbf{X})$
- parametric model:

$$\text{ROC} = \{(\alpha, \text{TPF}(\text{FPF}^{-1}(\alpha))) | \forall \alpha \in (0, 1)\}$$

- $\text{ROC}(0) = 0$ ,  $\text{ROC}(1) = 1$ , and

$$\frac{\partial \text{ROC}(t)}{\partial t} = \frac{f_D(\text{FPF}^{-1}(t))}{f_{\bar{D}}(\text{FPF}^{-1}(t))},$$

where  $f_D$  and  $f_{\bar{D}}$  are the probability densities of the scores within diseased and healthy populations, respectively.

- *the ROC curve describes the relationship between the two distributions, and is independent of them*

Note that

$$\frac{\partial \text{ROC}(t)}{\partial t} = \frac{P[Y = t|D = 1]}{P[Y = t|D = 0]} = \mathcal{LR}(t)$$

→ the **likelihood ratio** at threshold  $t$ .

- if  $\mathcal{LR}$  is monotonically increasing, then the classification rule of the form  $\mathcal{LR} > t$  is optimal
- the ROC curve based on  $\mathcal{LR}$  is uniformly above all other curves
- the optimal ROC curve is *concave*;  $\Rightarrow$  its slope is a monotone decreasing function



## Summary indices

Area under the ROC curve (AUC):

$$\text{AUC} = \int_0^1 \text{ROC}(t) dt$$

Properties:

- $0.5 \leq \text{AUC} \leq 1$

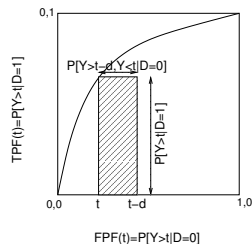
## Summary indices

Area under the ROC curve (AUC):

$$\text{AUC} = \int_0^1 \text{ROC}(t) dt$$

Properties:

- $0.5 \leq \text{AUC} \leq 1$
- $\text{AUC} = P[Y_D > Y_{\bar{D}}]$



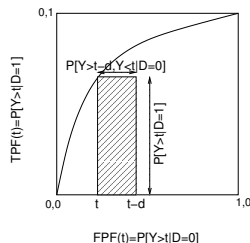
## Summary indices

Area under the ROC curve (AUC):

$$\text{AUC} = \int_0^1 \text{ROC}(t) dt$$

Properties:

- $0.5 \leq \text{AUC} \leq 1$
- $\text{AUC} = P[Y_D > Y_{\bar{D}}] \rightarrow$  the probability of correctly ordering a random pair of cases (Mann–Whitney–Wilcoxon U-statistic)



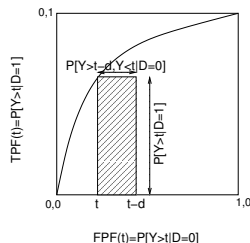
## Summary indices

Area under the ROC curve (AUC):

$$\text{AUC} = \int_0^1 \text{ROC}(t) dt$$

Properties:

- $0.5 \leq \text{AUC} \leq 1$
- $\text{AUC} = P[Y_D > Y_{\bar{D}}] \rightarrow$  the probability of correctly ordering a random pair of cases (Mann–Whitney–Wilcoxon U–statistic)
- $\text{AUC} = \int_0^1 \text{TPF}(\text{FPF}^{-1}(t)) dt = - \int_{-\infty}^{\infty} \text{TPF}(t) d\text{FPF}(t)$



# The binormal ROC curve

Assuming normal distributions for the scores:

$$Y_D \sim \mathcal{N}(\mu_D, \sigma_D^2); \quad Y_{\bar{D}} \sim \mathcal{N}(\mu_{\bar{D}}, \sigma_{\bar{D}}^2),$$

ROC becomes:

$$\text{ROC}(t) = \Phi\left(\frac{\mu_D - \mu_{\bar{D}}}{\sigma_D} + \frac{\sigma_{\bar{D}}}{\sigma_D} \Phi^{-1}(t)\right)$$

## The binormal ROC curve

Assuming normal distributions for the scores:

$$Y_D \sim \mathcal{N}(\mu_D, \sigma_D^2); \quad Y_{\bar{D}} \sim \mathcal{N}(\mu_{\bar{D}}, \sigma_{\bar{D}}^2),$$

ROC becomes:

$$\text{ROC}(t) = \Phi\left(\frac{\mu_D - \mu_{\bar{D}}}{\sigma_D} + \frac{\sigma_{\bar{D}}}{\sigma_D} \Phi^{-1}(t)\right)$$

General form

$$\text{ROC}(t) = \Phi(\alpha + \beta \Phi^{-1}(t))$$

where  $\alpha, \beta > 0$  and  $\Phi$  is the standard normal CDF.

Properties:

- $AUC = \Phi\left(\frac{\alpha}{\sqrt{1+\beta^2}}\right)$

Properties:

- $AUC = \Phi\left(\frac{\alpha}{\sqrt{1+\beta^2}}\right)$
- binormal assumption: there exists some monotone strictly increasing function  $h(\cdot)$  which makes  $Y_D$  and  $Y_{\bar{D}}$  normally distributed



## Properties:

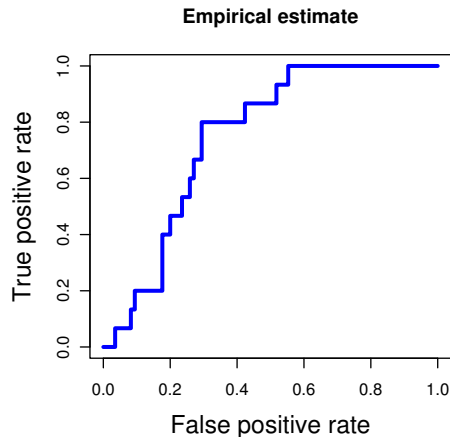
- $AUC = \Phi\left(\frac{\alpha}{\sqrt{1+\beta^2}}\right)$
- binormal assumption: there exists some monotone strictly increasing function  $h(\cdot)$  which makes  $Y_D$  and  $Y_{\bar{D}}$  normally distributed
- if the ROC is binormal,  $ROC(t) = \Phi(\alpha + \beta\Phi^{-1}(t))$ , then  $h(s) = -\Phi^{-1}(FPF(s))$  transforms the scores  $Y_D$  and  $Y_{\bar{D}}$  into normally distributed random variables.

## Empirical estimates of ROC

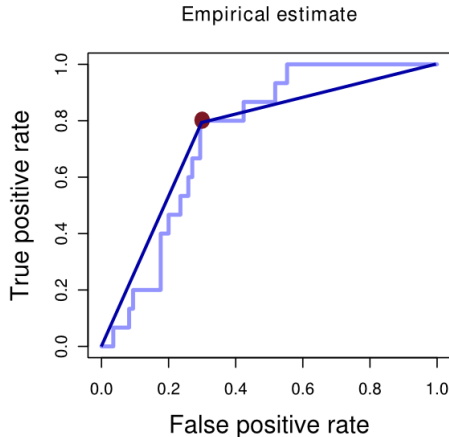
$$\hat{ROC}_e(t) = \hat{TPF}(\hat{FPF}^{-1}(t)) :$$

$$\hat{TPF}(t) = \sum_{i=1}^{n_D} \mathbb{I}[Y_{Di} \geq t]$$

$$\hat{FPF}(t) = \sum_{i=1}^{n_{\bar{D}}} \mathbb{I}[Y_{\bar{D}i} \geq t]$$



## “ROC” for single threshold



# Empirical estimates of AUC

Mann–Whitney–Wilcoxon U–statistic:

$$\hat{AUC}_e = \frac{1}{n_D n_{\bar{D}}} \sum_{i=1}^{n_D} \sum_{j=1}^{n_{\bar{D}}} (\mathbb{I}[Y_{Di} > Y_{\bar{D}j}] + 0.5\mathbb{I}[Y_{Di} = Y_{\bar{D}j}])$$

Note: if only one point in the (FPF, TPF) space is given,  
 $\hat{AUC} = 0.5(1 + \text{TPF} - \text{FPF})$ .

## AUC: sampling variability

$$\text{Var}(\hat{\text{AUC}}_e) = \frac{1}{n_D n_{\bar{D}}} [\text{AUC}(1 - \text{AUC}) + (n_D - 1)(Q_1 - \text{AUC}^2) \\ + (n_{\bar{D}} - 1)(Q_2 - \text{AUC}^2)]$$

where

$$Q_1 = P[Y_{Di} \geq Y_{\bar{D}j}, Y_{Dk} \geq Y_{\bar{D}j}]$$

$$Q_2 = P[Y_{Di} \geq Y_{\bar{D}j}, Y_{Di} \geq Y_{\bar{D}k}].$$

## Semi-parametric models

Start from

$$\hat{ROC}(t) = \hat{TPF}(\hat{FPF}^{-1}(t|\hat{\alpha})|\hat{\beta})$$

and assume some parametric form for TPF and FPF for which estimate the parameters from data.

Ex. of semi-parametric model:

$$Y_{Di} = \mu_D + \sigma_D \varepsilon_i$$

$$Y_{\bar{D}i} = \mu_{\bar{D}} + \sigma_{\bar{D}} \varepsilon_i$$

where  $\varepsilon$  have mean 0 and variance 1 and follow some distribution function  $S$ .

Ex. of semi-parametric model:

$$Y_{Di} = \mu_D + \sigma_D \varepsilon_i$$

$$Y_{\bar{D}i} = \mu_{\bar{D}} + \sigma_{\bar{D}} \varepsilon_i$$

where  $\varepsilon$  have mean 0 and variance 1 and follow some distribution function  $S$ .

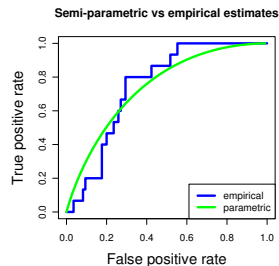
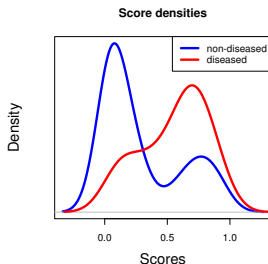
$$\hat{S}(t) = \frac{1}{n_D + n_{\bar{D}}} \left\{ \sum_i \mathbb{I} \left[ \frac{Y_{Di} - \hat{\mu}_D}{\hat{\sigma}_D} \geq t \right] + \sum_i \mathbb{I} \left[ \frac{Y_{\bar{D}i} - \hat{\mu}_{\bar{D}}}{\hat{\sigma}_{\bar{D}}} \geq t \right] \right\}$$

which leads to

$$\text{ROC}(t) = \hat{S} \left( (\hat{\mu}_{\bar{D}} - \hat{\mu}_D) / \hat{\sigma}_D + (\hat{\sigma}_{\bar{D}} / \hat{\sigma}_D) S^{-1}(t) \right)$$



## Ex: empirical vs. semi-parametric estimation



$$\hat{AUC}_e \approx 0.7475; \quad \hat{AUC}_{sp} \approx 0.7418$$

## Bibliography

- Pepe, *The statistical evaluation of medical tests for classification and prediction*, OUP, 2004
- Krzanowski, Hand, *ROC curves for continuous data*, CRC Press, 2009

# Outline

- 1 Introduction
- 2 Building classification rules
  - Statistical framework
  - Discriminant functions
- 3 Assessing the performance**
  - Medical application context
  - Measures of accuracy for binary tests
  - Measures of accuracy for continuous tests: ROC and AUC
  - Performance estimation**

## Why estimation?

- finite training data
- no formula for CI without distribution assumptions
- often, a single data set is available for both model building and performance measuring
- performance estimated on the modeling data is optimistically biased

### Idea

Split (maybe repeatedly) the available data into a training and a validation set, and assess the performance only on the data that has not been used in building the model.

# Resampling methods

- simple split-sample approach

# Resampling methods

- simple split-sample approach
- $k$ -fold cross-validation

## Resampling methods

- simple split-sample approach
- $k$ -fold cross-validation
- Monte-Carlo cross-validation

## Resampling methods

- simple split–sample approach
- $k$ –fold cross–validation
- Monte–Carlo cross–validation
- repeated  $k$ –fold cross–validation



## Resampling methods

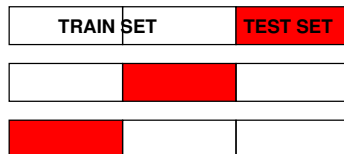
- simple split-sample approach
- $k$ -fold cross-validation
- Monte-Carlo cross-validation
- repeated  $k$ -fold cross-validation
- leave-one-out

# Resampling methods

- simple split-sample approach
- $k$ -fold cross-validation
- Monte-Carlo cross-validation
- repeated  $k$ -fold cross-validation
- leave-one-out
- bootstrapping
- ...

## $k$ -fold cross-validation

- separated train and test sets
- randomly divided data into  $k$  subsets (folds) – you may also choose to enforce the proportion of the classes (stratified CV)
- train on  $k - 1$  folds and test on the holdout fold
- estimate the error as the average error measured on holdout folds



- usually  $k = 5$  or  $k = 10$
- if  $k = n \Rightarrow$  leave-one-out estimator
- improved estimation: repeated  $k$ -CV (e.g.  $100 \times (5 - CV)$ )

## $k$ -fold cross-validation

From  $k$  folds:

- $\epsilon_1, \dots, \epsilon_k$  errors on the test folds

## $k$ -fold cross-validation

From  $k$  folds:

- $\epsilon_1, \dots, \epsilon_k$  errors on the test folds
- $\hat{E}_{k-CV} = \frac{1}{k} \sum_{j=1}^k \epsilon_j$

## $k$ -fold cross-validation

From  $k$  folds:

- $\epsilon_1, \dots, \epsilon_k$  errors on the test folds
- $\hat{E}_{k-CV} = \frac{1}{k} \sum_{j=1}^k \epsilon_j$
- estimated standard deviation

## $k$ -fold cross-validation

From  $k$  folds:

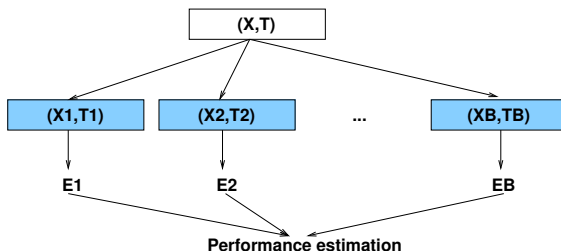
- $\epsilon_1, \dots, \epsilon_k$  errors on the test folds
- $\hat{E}_{k-CV} = \frac{1}{k} \sum_{j=1}^k \epsilon_j$
- estimated standard deviation

Confidence intervals (simple version – normal approximation):

$$E \approx \hat{E} \pm \left( \frac{0.5}{n} + z \sqrt{\frac{\hat{E}(1 - \hat{E})}{n}} \right)$$

where  $n$  is the dataset size and  $z = \Phi^{-1}(1 - \alpha/2)$ , for a  $1 - \alpha$  confidence interval (e.g.  $z = 1.96$  for 95% conf. interval)

# Bootstrap error estimation



- 1 generate a new dataset  $(X_b, T_b)$  by *resampling with replacement* from the original dataset  $(X, T)$
- 2 train the classifier on  $(X_b, T_b)$  and test on the left out data, to obtain an error  $\hat{E}_b$ .
- 3 repeat 1–2 for  $b = 1, \dots, B$  and collect  $\hat{E}_b$ .



## Bootstrap error estimation

- estimate the error: for example, use the *.632 estimator*

$$\hat{E} = 0.368E_0 + 0.632 \frac{1}{B} \sum_{b=1}^B \hat{E}_b$$

where  $E_0$  is the error rate on the full training set  $(X, T)$ .

- use the empirical distribution of  $\hat{E}_b$  to obtain confidence intervals

## LPO bootstrap

Classification rule:

$$\hat{h}(x) \underset{C_2}{\overset{C_1}{\geq}} \theta$$

where  $\hat{h}$  is the estimated log-likelihood ratio and  $C_i$  are the class labels.

*Empirical* AUC (conditioned on the training set) can be approximated by:

$$\widehat{\text{AUC}} = \frac{1}{n_1 n_2} \sum_{j=1}^{n_2} \sum_{i=1}^{n_1} \psi(\hat{h}(x_i|C_1), \hat{h}(x_j|C_2))$$

where  $\psi$  is the Mann-Whitney kernel,

$$\psi(a, b) = \begin{cases} 1 & a > b \\ \frac{1}{2} & a = b \\ 0 & a < b \end{cases}$$

Estimation of the *expected* AUC by LPO bootstrap:

$$\widehat{AUC}^{LPO} = \frac{1}{n_1 n_2} \sum_{j=1}^{n_2} \sum_{i=1}^{n_1} \widehat{AUC}_{i,j}$$
$$\widehat{AUC}_{i,j} = \frac{\sum_{b=1}^B l_j^b l_i^b \psi(\hat{h}_b(x_i), \hat{h}_b(x_j))}{\sum_{b=1}^B l_j^b l_i^b}$$

When 2 independent data sets are available, one can estimate:

- the expected value of the conditional AUC: expectation over the population of training sets *of the same size*;
- variability of the performance estimate due to finite train set;
- variability of the performance estimate due to finite validation sets;

Yousef et al., *Assessing classifiers from two independent data sets using ROC analysis: a nonparametric approach*, PAMI

2006

## What we do learn from CV:

- the expected performance of the modeling recipe;
- the imprecision in estimating the performance;
- we can have a look at:
  - what are the most stable features
  - what are the points always missclassified

## What we do learn from CV:

- the expected performance of the modeling recipe;
- the imprecision in estimating the performance;
- we can have a look at:
  - what are the most stable features
  - what are the points always missclassified

## What we do not learn from CV:

- the best features
- the best classifier
- the best meta-parameters

We obtain these by training on the full dataset (no CV).

THE END