Supervised Learning 1

Blaise Hanczar (812-E)

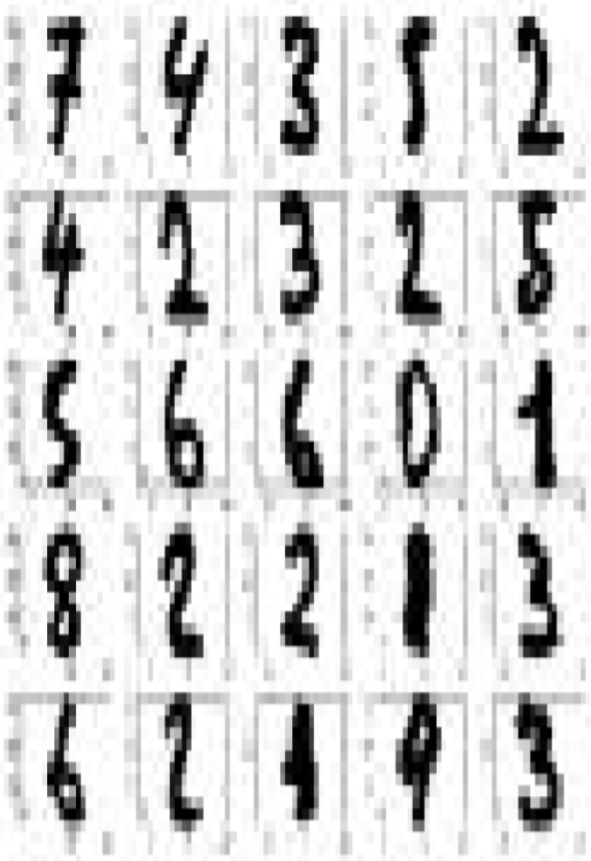
Outline

- Introduction to classification
- Bayes classifier
- Performance of a classifier
- Linear Discriminant Analysis

Principle

Objective: Learning a prediction model (classifier) from training examples.

Application : Handwritten recognition

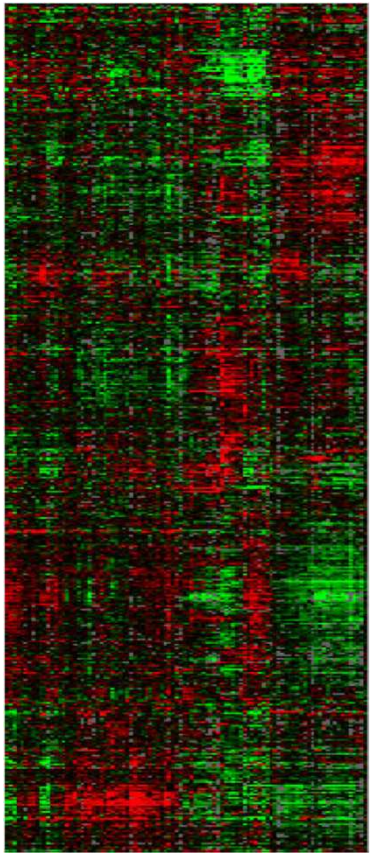


Handwritten digits

Prediction of the zip code from handwritten digits

Class: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9

Application : medical



Gene expression

Prediction of a diagnostic or response to a treatment from clinical or genomic data

Class:

- Sick / Safe
- Good responder / Bad responder
- Type of cancer

Application : Image analysis

Face images

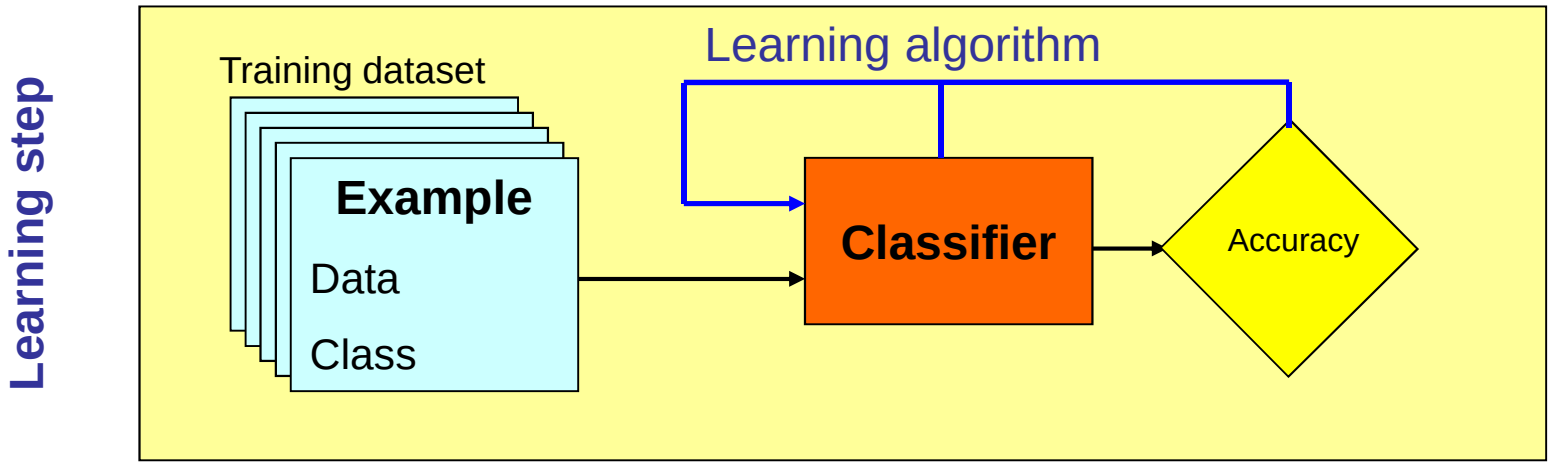
Prediction of the face expression from images

Class: Smile/ No smile

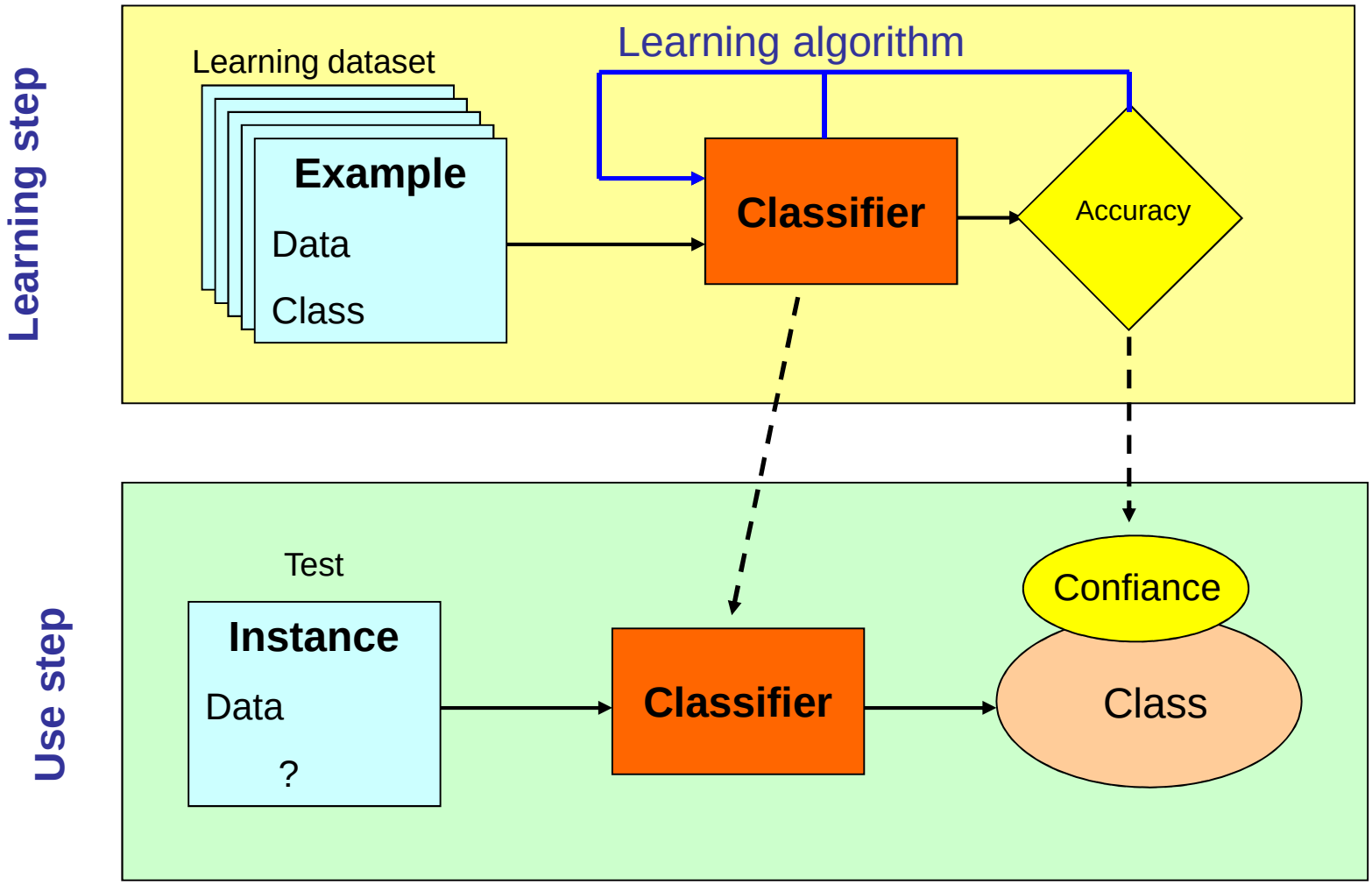
Applications

- Medical diagnostic
- Spam detection
- Credit decision
- Car design optimization
- Prediction of the winning hand in poker game
- Customer fidelity
- Etc…

Model construction



Use the model

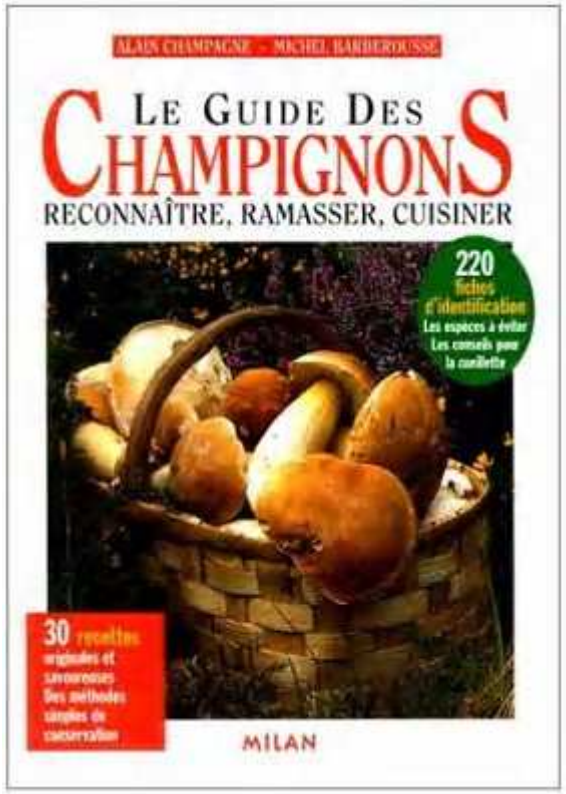


Spécifier un objectif

Manger des champignons en évitant de mourir

Discerner les champignons:

- comestibles 
- empoisonnés 



Collection of objects

- Creation of a set of objects
- Identification of the class by an expert

Preprocessing of the data

Measure a set of feature

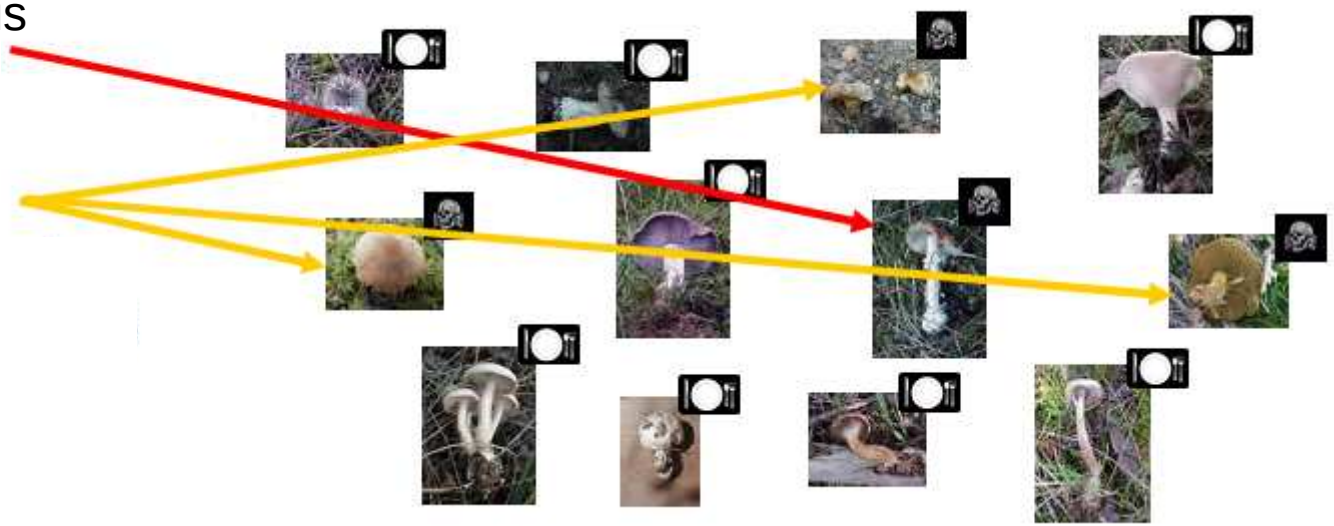
- Heigth/width of foot
- Heigth/width of hat
- smell
- weight

Modelisation

Identification of the following classification rule:

If a mushroom is
red or yellow

Then
It is not eatable



Apply the classification rule on new mushroom

Bayes theorem

- Bayes theorem in classification context:

$$P(C \mid X) = \frac{P(X \mid C)P(C)}{P(X)}$$

- *P(C)*: Prior probability of the classes
- *P(X)*: Probability of the data X
- *P(X|C)*: Likelihood (probability of see the data X given the class C)
- *P(C|X)*: Posterior probability to belong to the class X given the data X

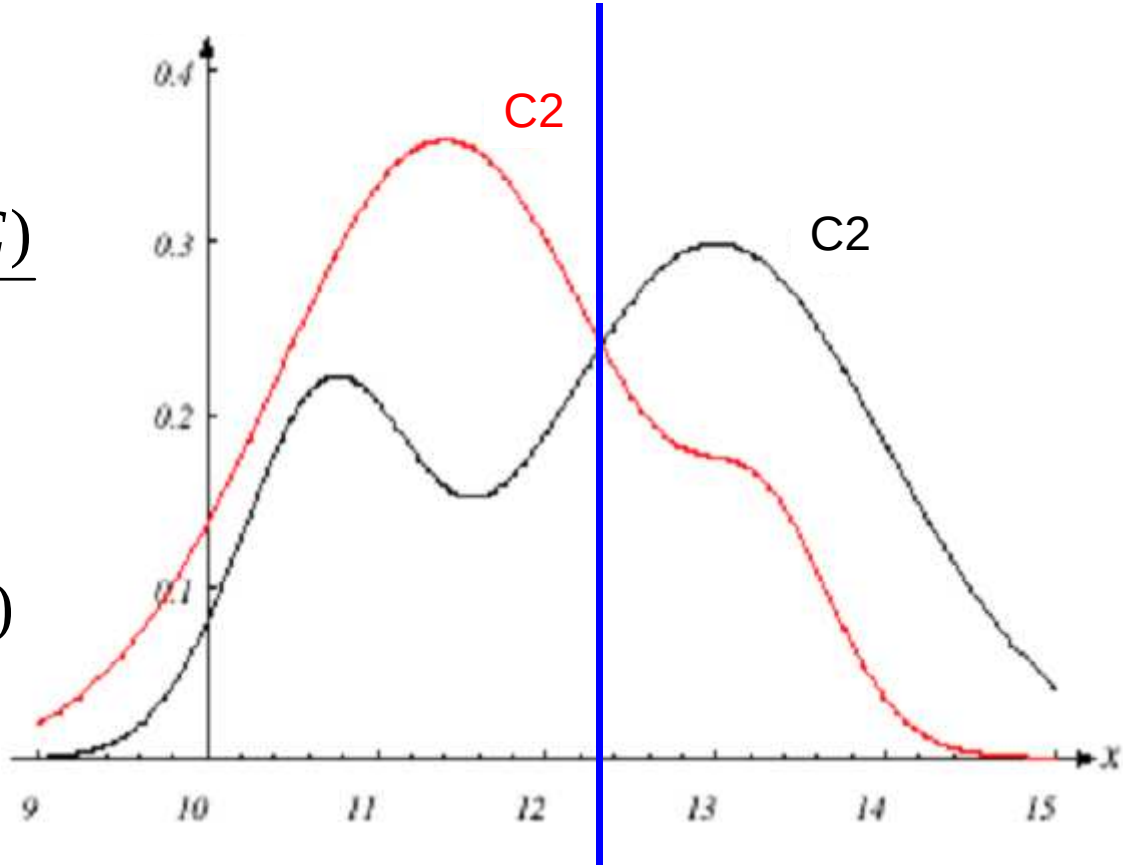
Bayes classifier

Posterior probability:

$$P(C \mid X) = \frac{P(X \mid C)P(C)}{P(X)}$$

Bayes classifier:

$$C^* = \arg\max_C (P(C \mid X))$$



Decision function

Decision function of the classes

$$g(x) = P(\omega_1 \mid x) - P(\omega_2 \mid x)$$

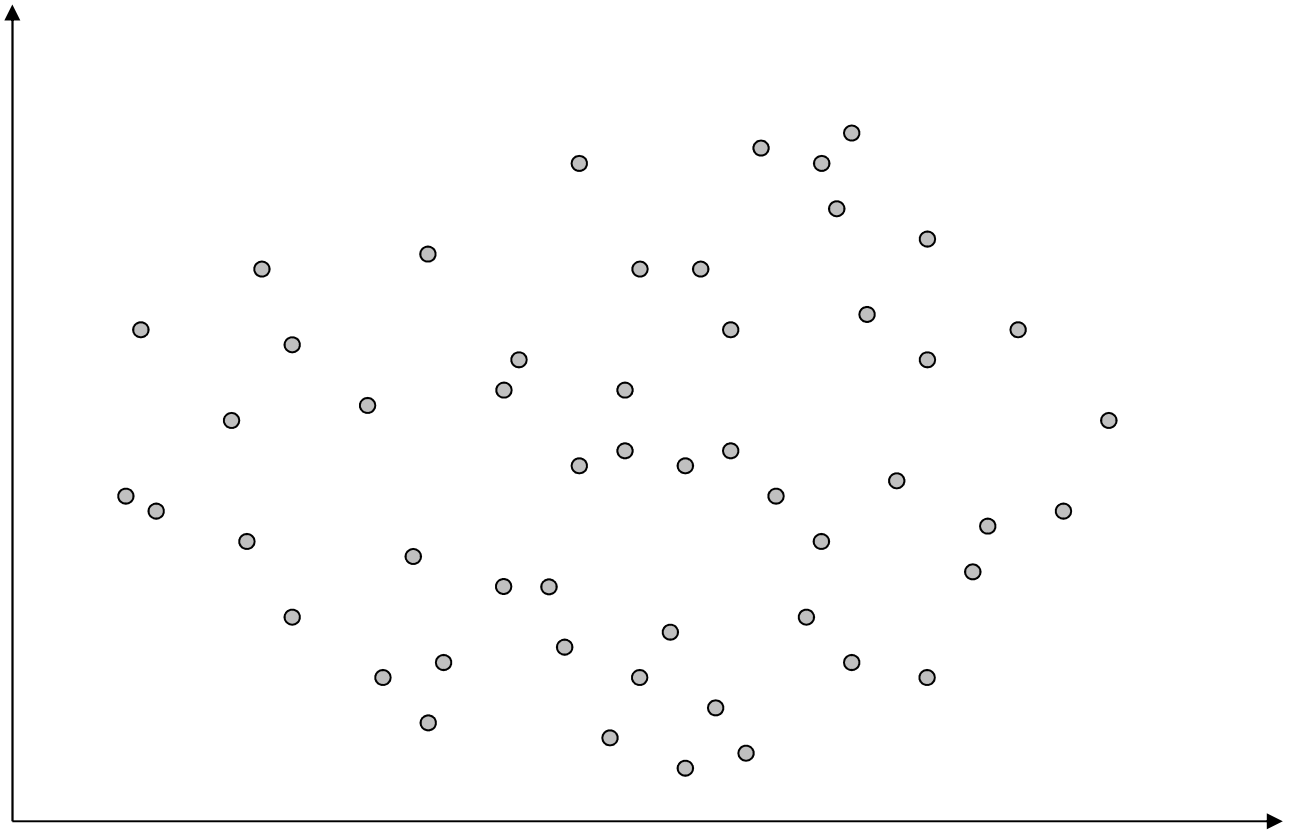
$$= \ln \frac{P(x \mid \omega_1)}{P(x \mid \omega_2)} + \ln \frac{P(\omega_1)}{P(\omega_2)}$$

$$g_i(x) = \ln P(x \mid \omega_i) + \ln P(\omega_i)$$

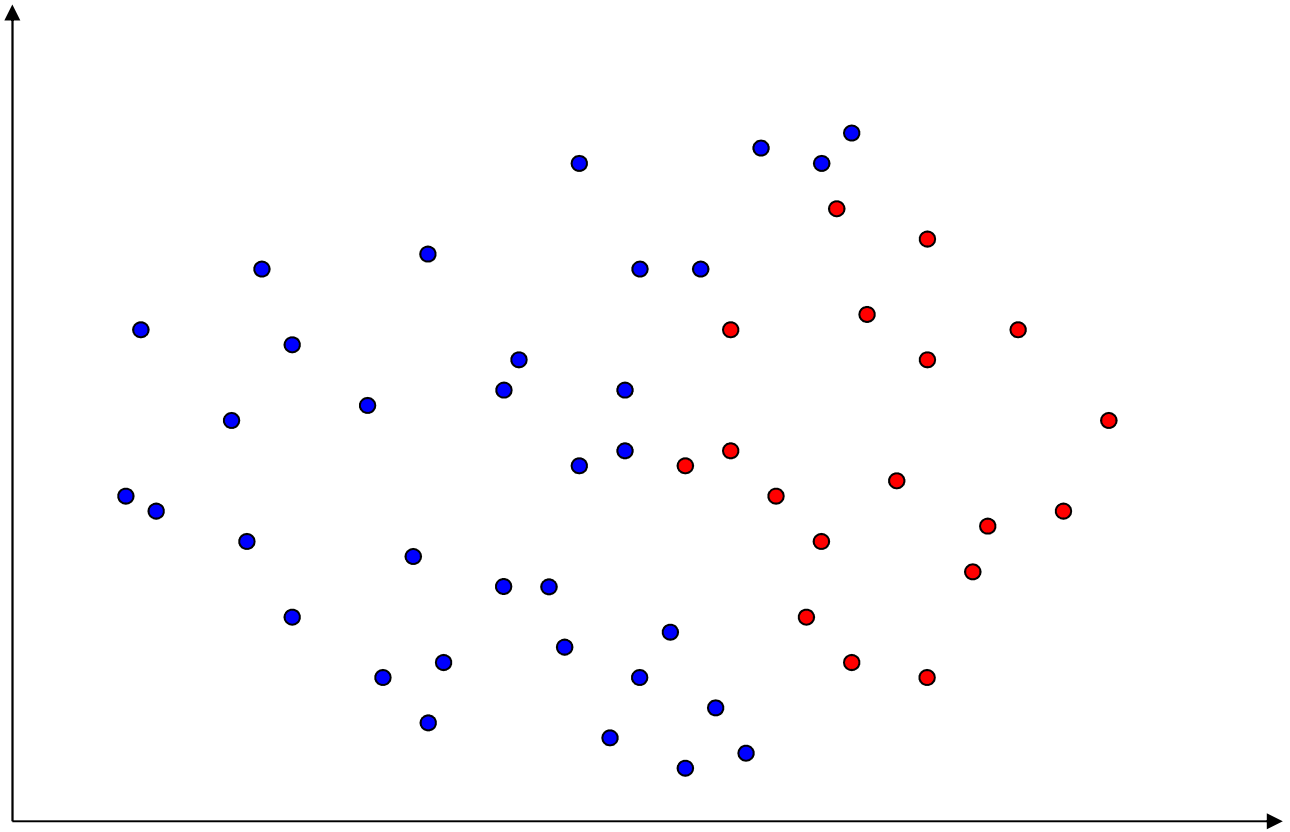
The distribution of the classes is generally unknown

We could put some assumptions on this distribution.

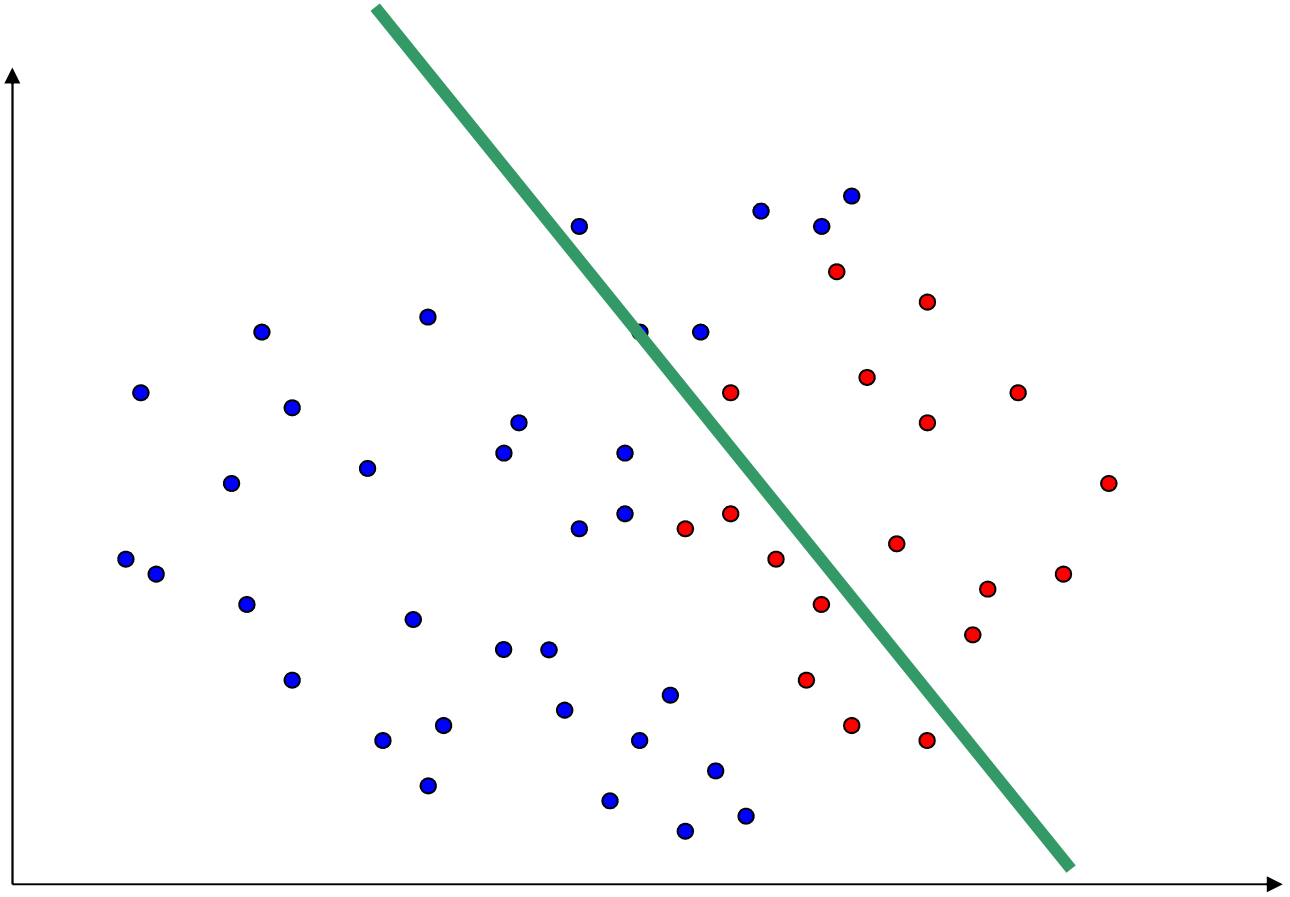
Representation of the examples in the features space



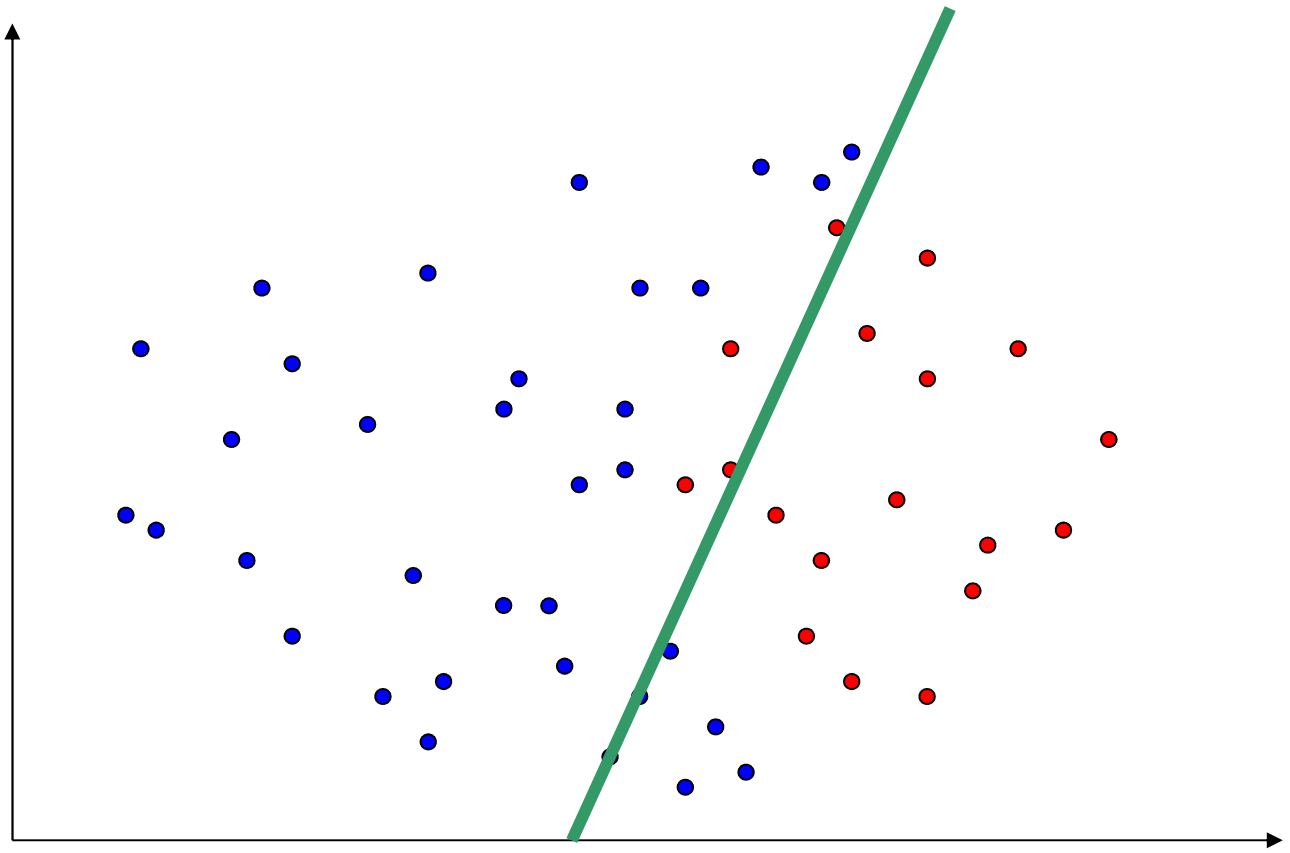
Identification of the classes



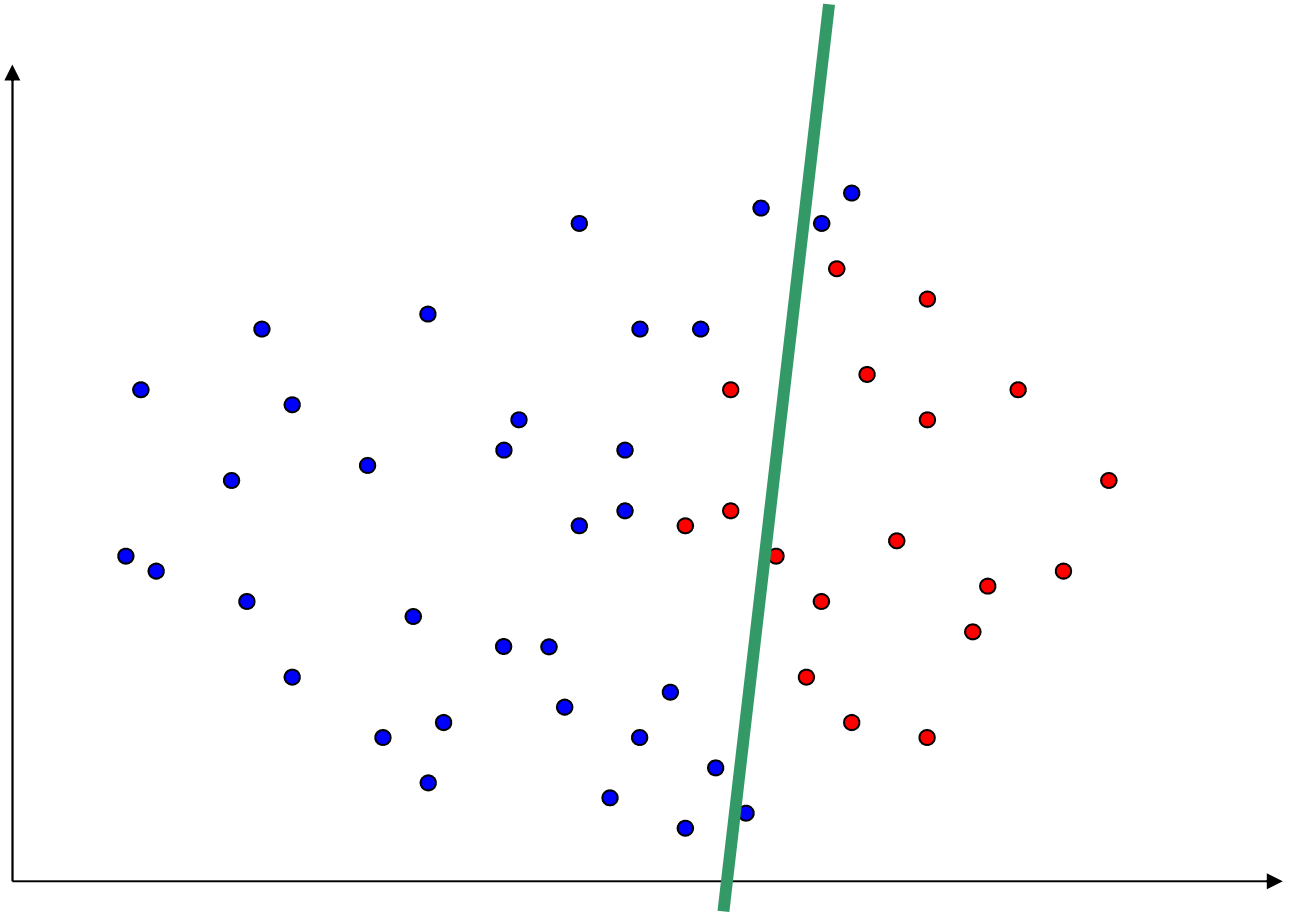
Learning the model



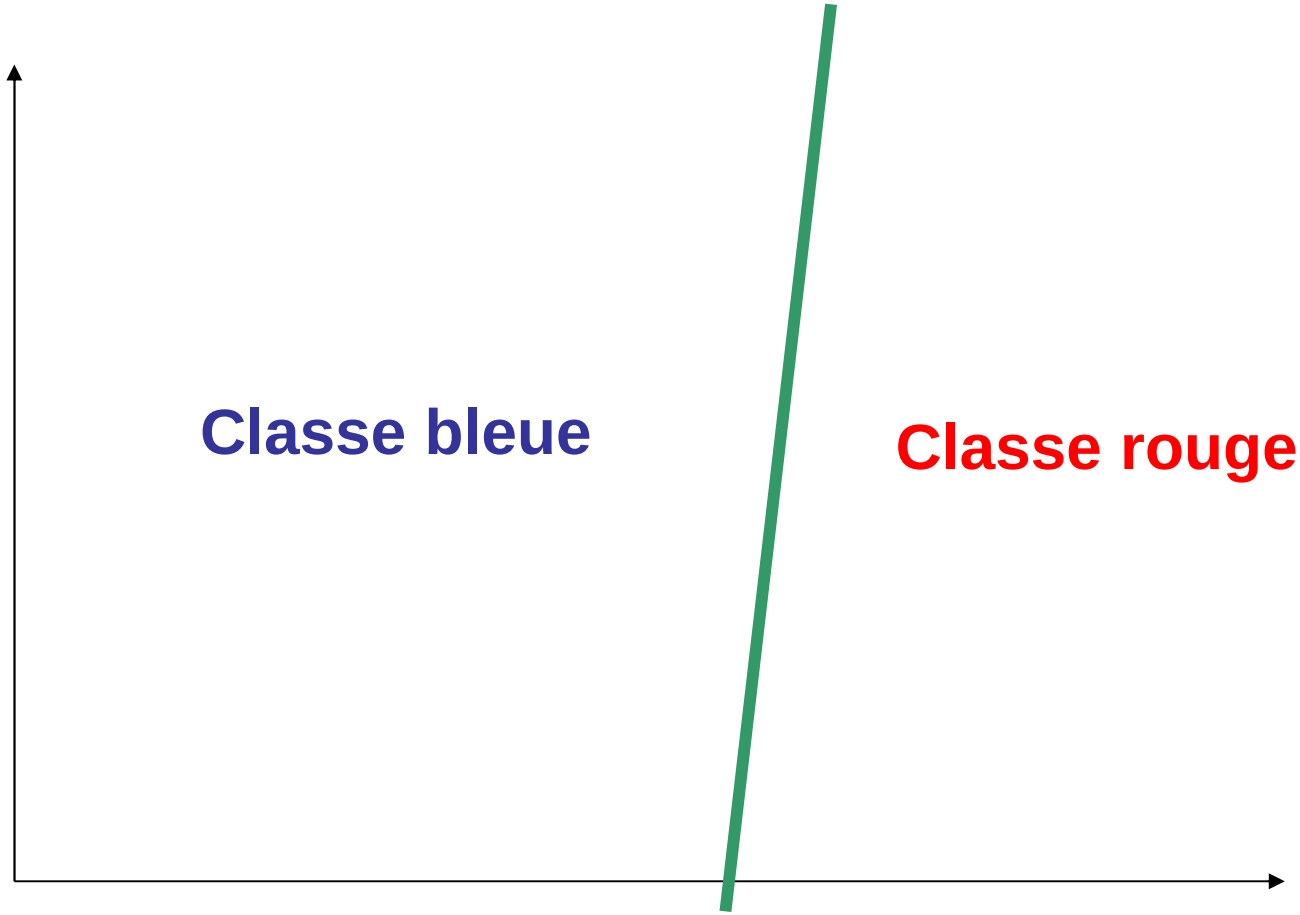
Learning the model



Learning the model



Classifier



Types of errors

2 classes YES / NO

Confusion matrix

	Predict Positve	Predict negative	Total
True class positive	True Positive TP	False Negative FN	P
True class negative	False Positive FP	True Negative TN	N

True positive rate: TPR = TP / P

False positive rate: FPR = FP / P

True negative rate: TNR = TN / N

False negative rate: FNR = FN / N

Classification cost

$$cost = \pi_P C_{TP} TPR + \pi_N C_{FP} FPR + \pi_N C_{TN} TNR + \pi_P C_{FN} FNR$$

	Predict YES	Predict NO
True class YES	C_{TP}	C_{FN}
True class NO	C_{FP}	C_{TN}

Classification cost

$$cost = \pi_P C_{TP} TPR + \pi_N C_{FP} FPR + \pi_N C_{TN} TNR + \pi_P C_{FN} FNR$$

The costs of TP and TN are generally consider null

$C_{TP} = C_{TN} = 0$

$$cost = \pi_N C_{FP} FPR + \pi_P C_{FN} FNR$$

	Predict YES	Predict NO
True class YES	C_{TP}	C_{FN}
True class NO	C_{FP}	C_{TN}

Classification cost

$$cost = \pi_P C_{TP} TPR + \pi_N C_{FP} FPR + \pi_N C_{TN} TNR + \pi_P C_{FN} FNR$$

The costs of TP and TN are generally consider null

$C_{TP} = C_{TN} = 0$

$$cost = \pi_N C_{FP} FPR + \pi_P C_{FN} FNR$$

We can fix the cost to FN to 1

$CFN = 1, \quad R = C_{FP}/C_{FN}$

$$\text{cost} = \pi_N \, R \, FPR + \pi_P \, FNR$$

If π_P and π_N are unknown, they can be estimated from the training examples

$\pi_p = P/(N+P)$ and $\pi_N = N/(N+P)$

$$\text{cost} = R \, \frac{FP}{N+P} + \frac{FN}{N+P}$$

	Predict YES	Predict NO
True class YES	C_{TP}	C_{FN}
True class NO	C_{FP}	C_{TN}

Other performance measures

- Error rate: $\mathrm{err} = 1/2(FPR + FNR)$
  It is a special case of the classification cost where:
  $C_{FP} = C_{FN} = 1$ and $\pi_p = \pi_N = 1/2$
- Accuracy: $acc = 1 - err$
- Precision: $pres = TP/(TP + FP)$
- Recall, Sensitivity: $rec = TP/P$
- Specificity: $spe = TN/(TN + FP)$
- F-measure: $Fmes = 2/(1/pres + 1/rec)$

ROC space

ROC: Receiver Operating Characteristic

Graphic where x-axis is FPR and y-axis is TPR

o Classifer
FPR = 0.08
TPR = 0.50

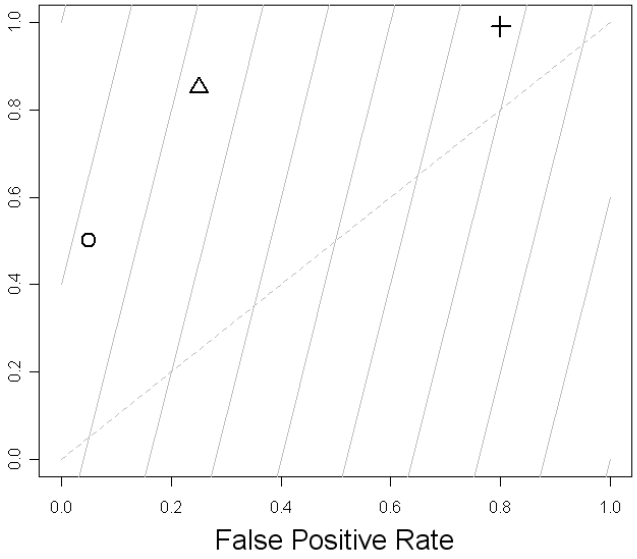
Δ Classifer
FPR = 0.28
TPR = 0.84

\+ Classifer
FPR = 0.80
TPR = 1

ROC space

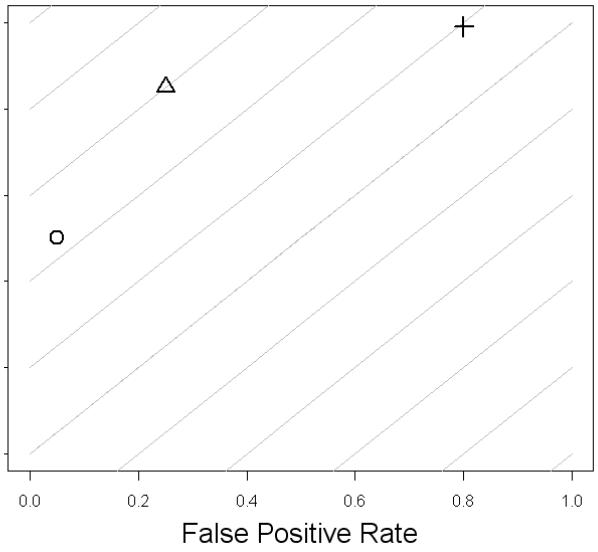
Visualisation of classifier performance by iso-cost lines in ROC space

Iso-cost line: $TPR = A\,FPR + B$ with $A = \frac{\pi_N C_{FP}}{\pi_P C_{FN}}$



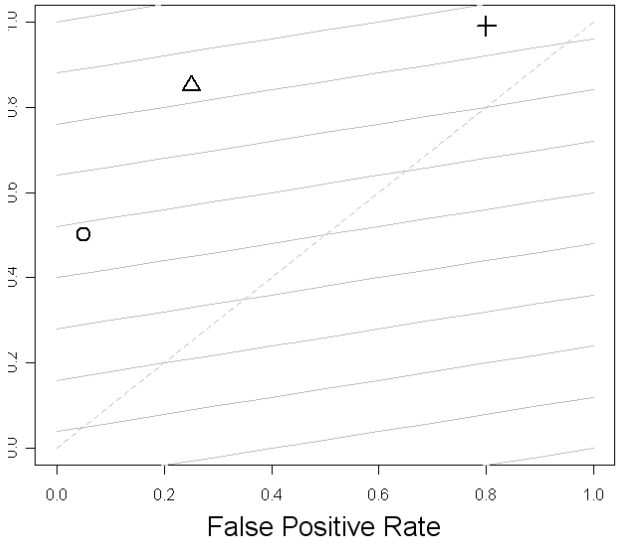
$C_{FN}=1$ $C_{FP}=5$

$\pi_N=\pi_P=1$



$C_{FN}=1$ $C_{FP}=1$

$\pi_N=\pi_P=1$

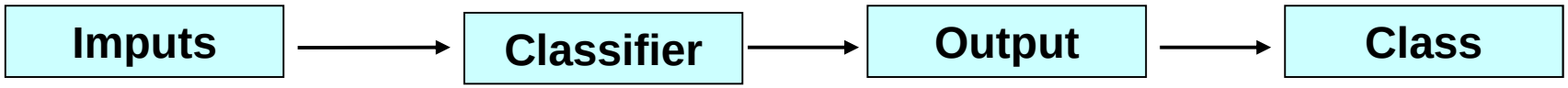


$C_{FN}=5$ $C_{FP}=1$

$\pi_N=\pi_P=1$

ROC curves

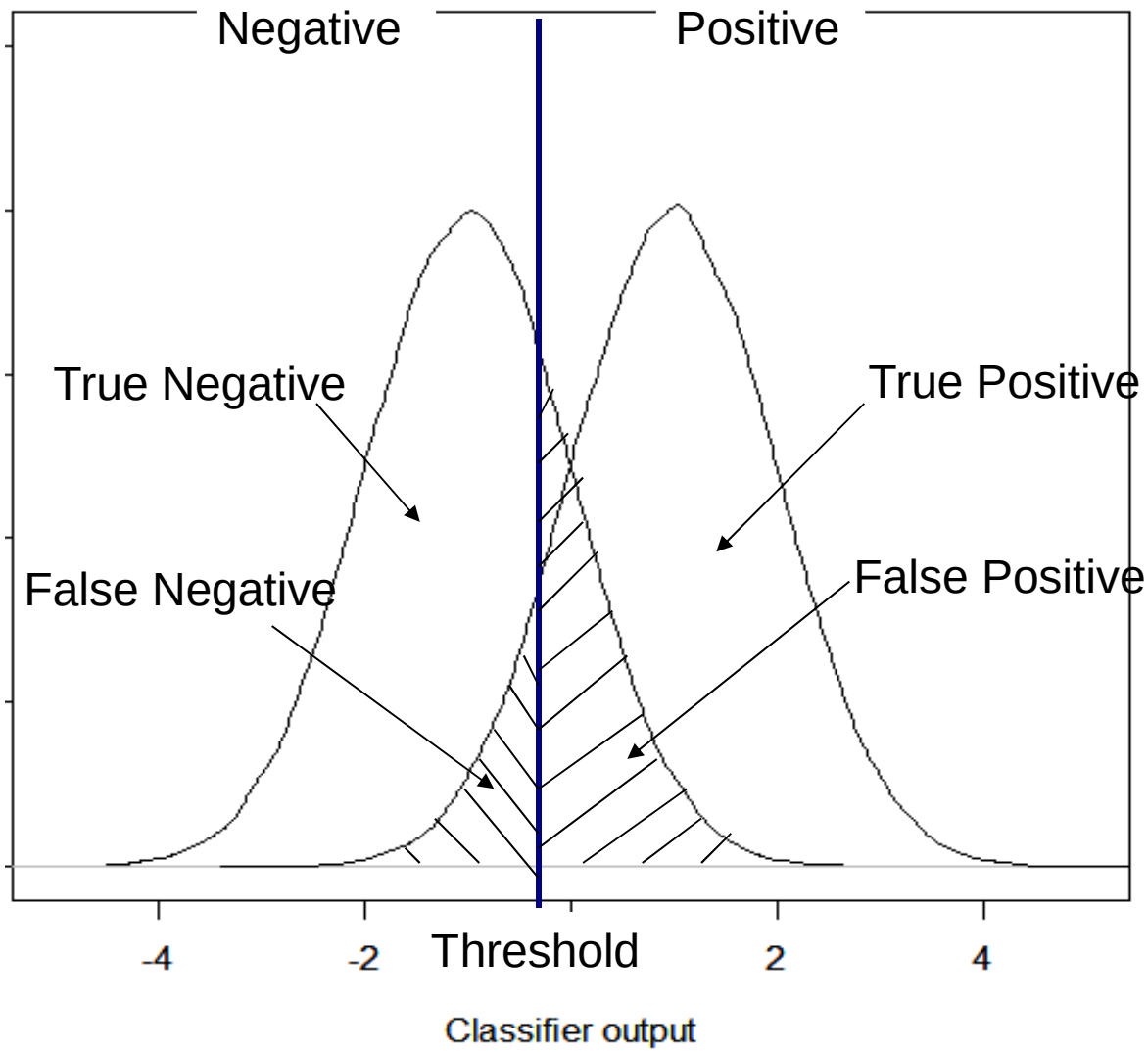
- The output of classifiers is not a classes but a continuous values
- A decision threshold is then applied to this output to determine the class



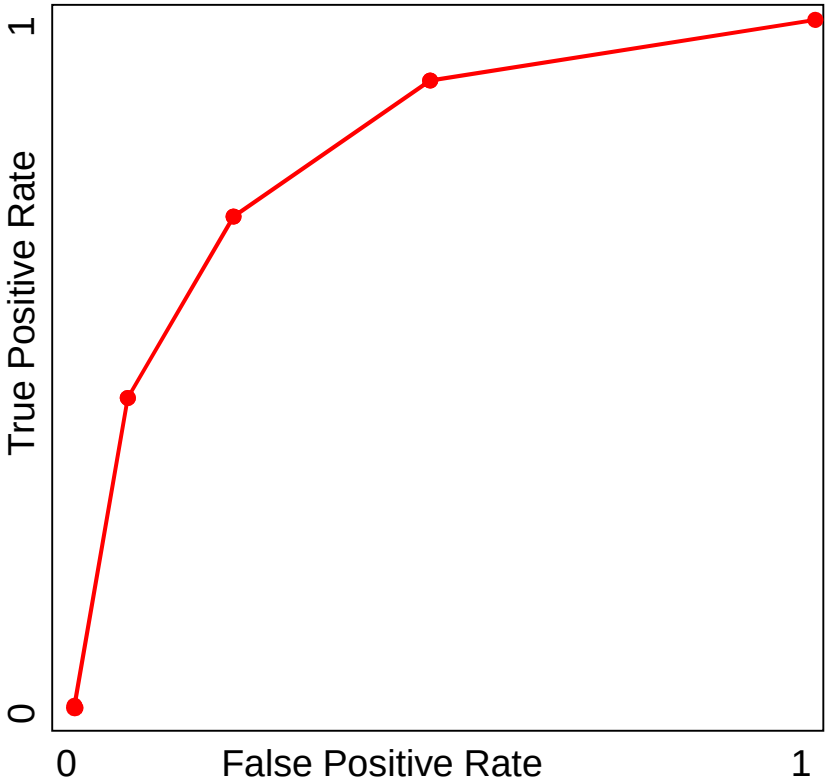
$$\text{if } f(x) > T \text{ then assign class } Positive$$

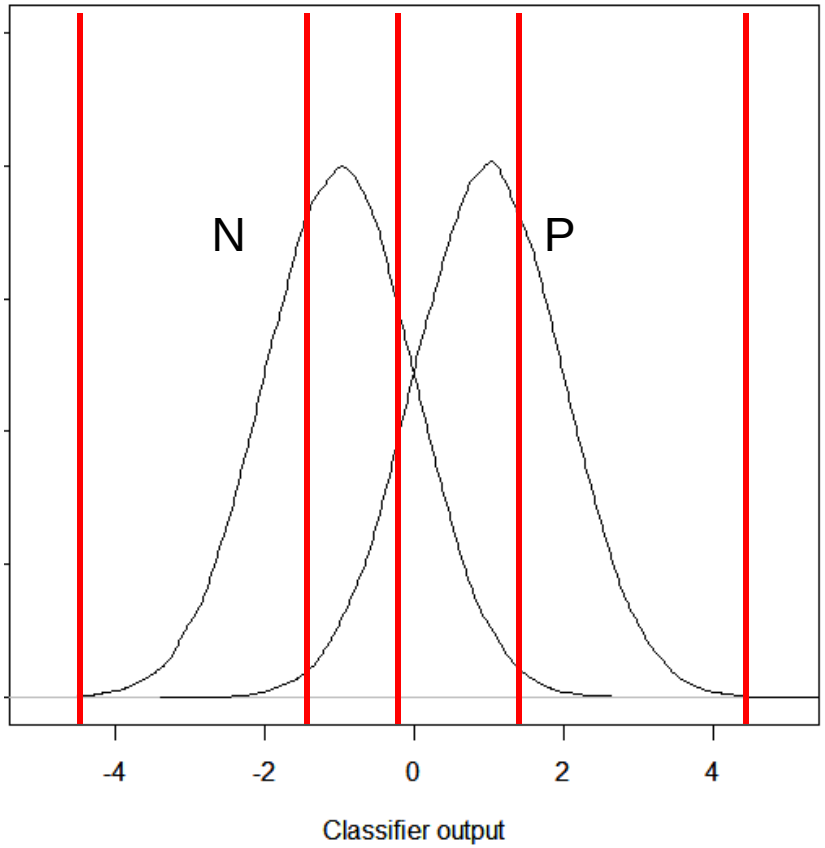
$$\text{if } f(x) \leq T \text{ then assign class } Negative$$

ROC curves


Negative
Positive
True Negative
True Positive
False Negative
False Positive
-4
-2
Threshold
2
4
Classifier output

ROC curves


True Positive Rate
0
1
0
False Positive Rate
1


N
P
-4
-2
0
2
4
Classifier output

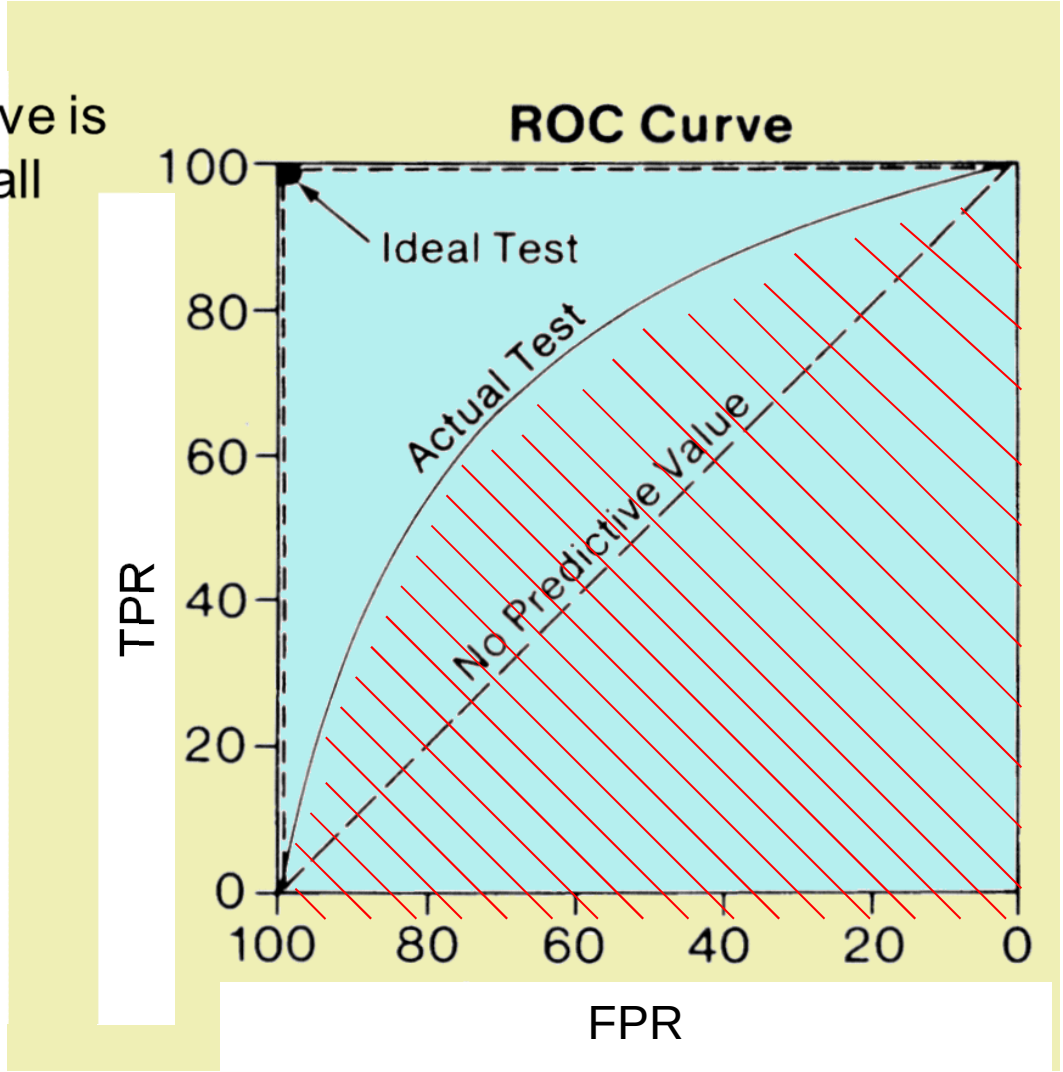
Courbe ROC

The area under (AUC) the ROC curve is a measure of performance over all cost values

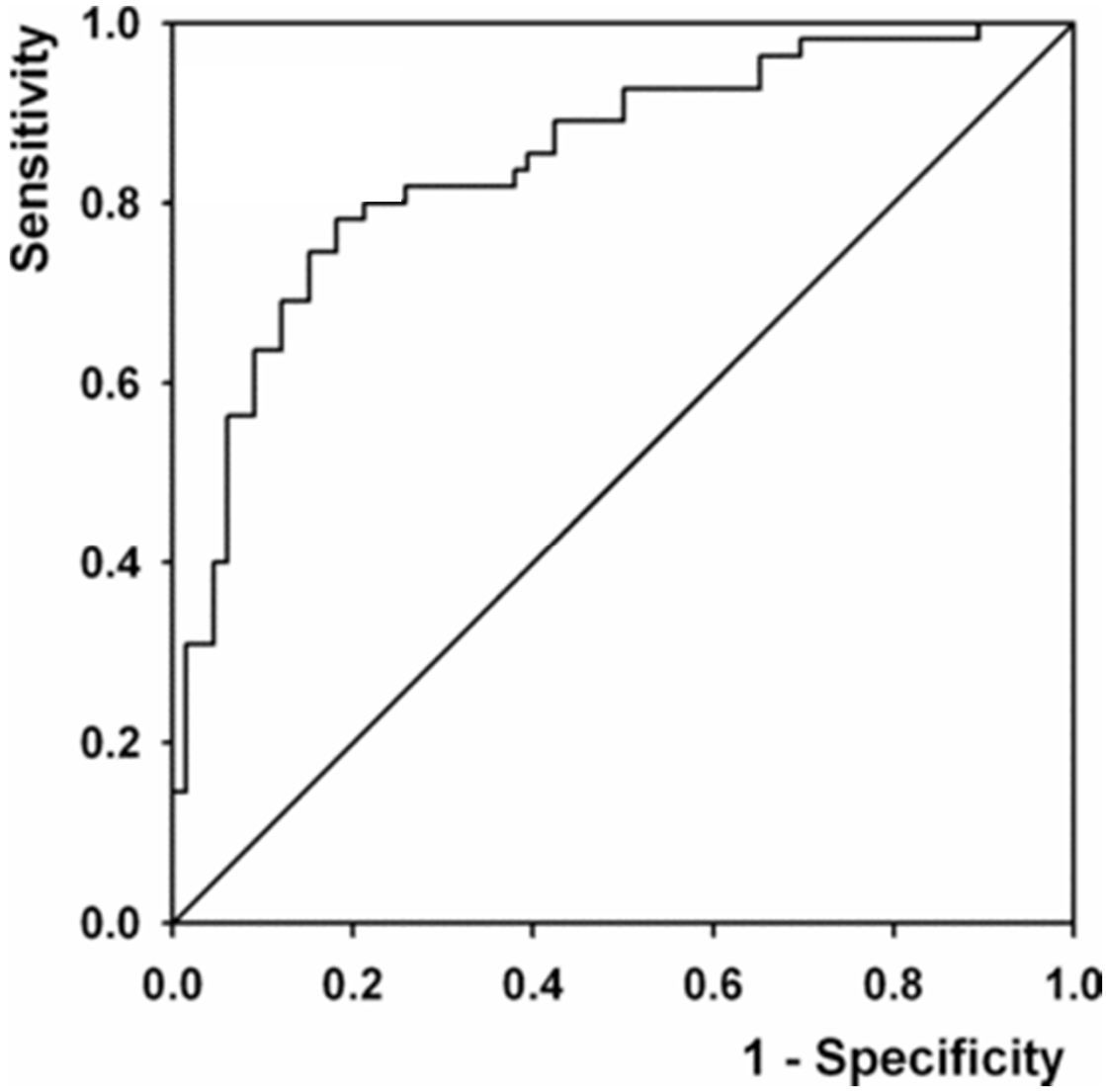
Computation of the AUC:

$$AUC = \frac{S_0 - P(P+1)/2}{P.N}$$

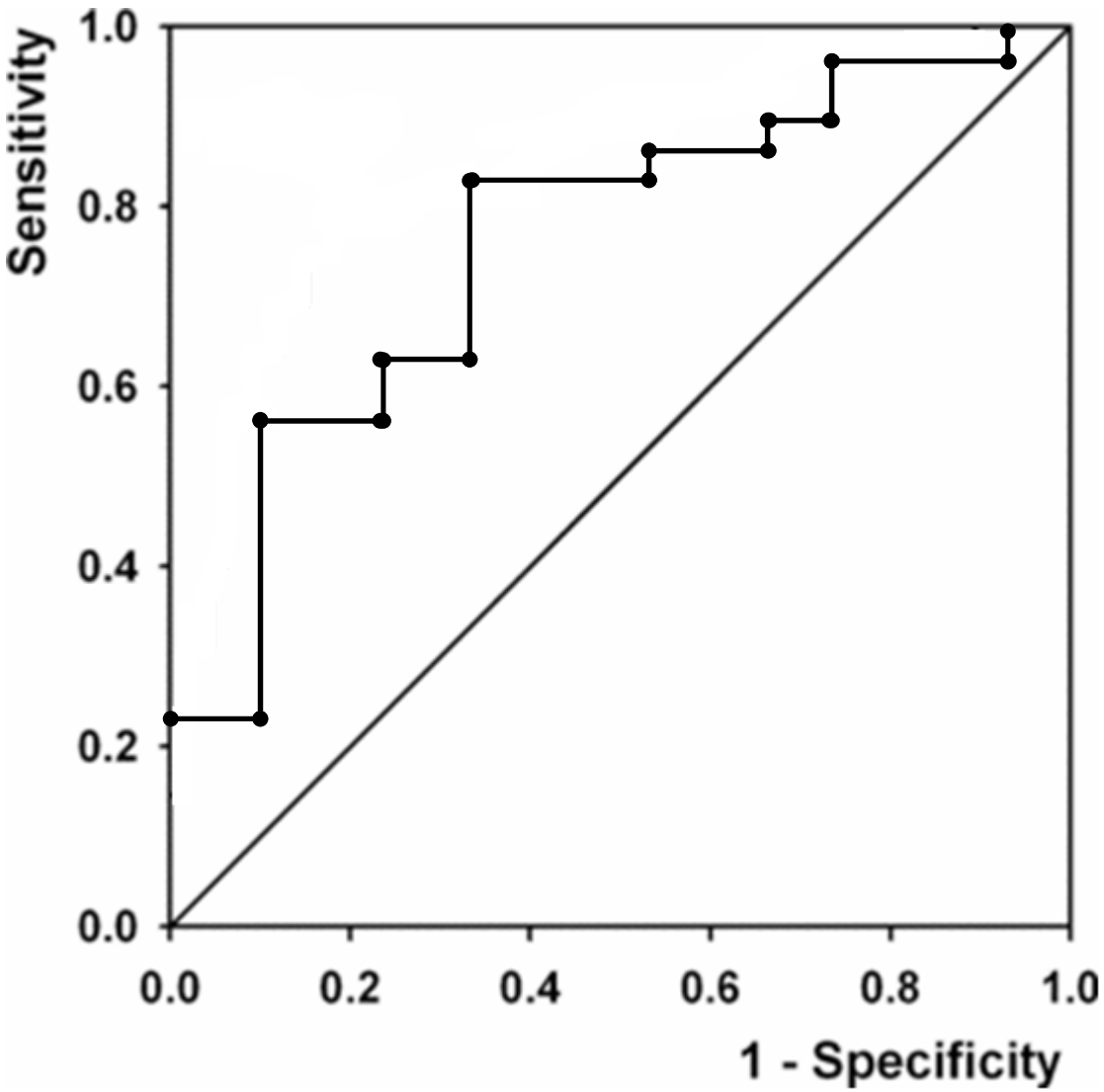
S_0 is the sum of ranks of examples in the positive class



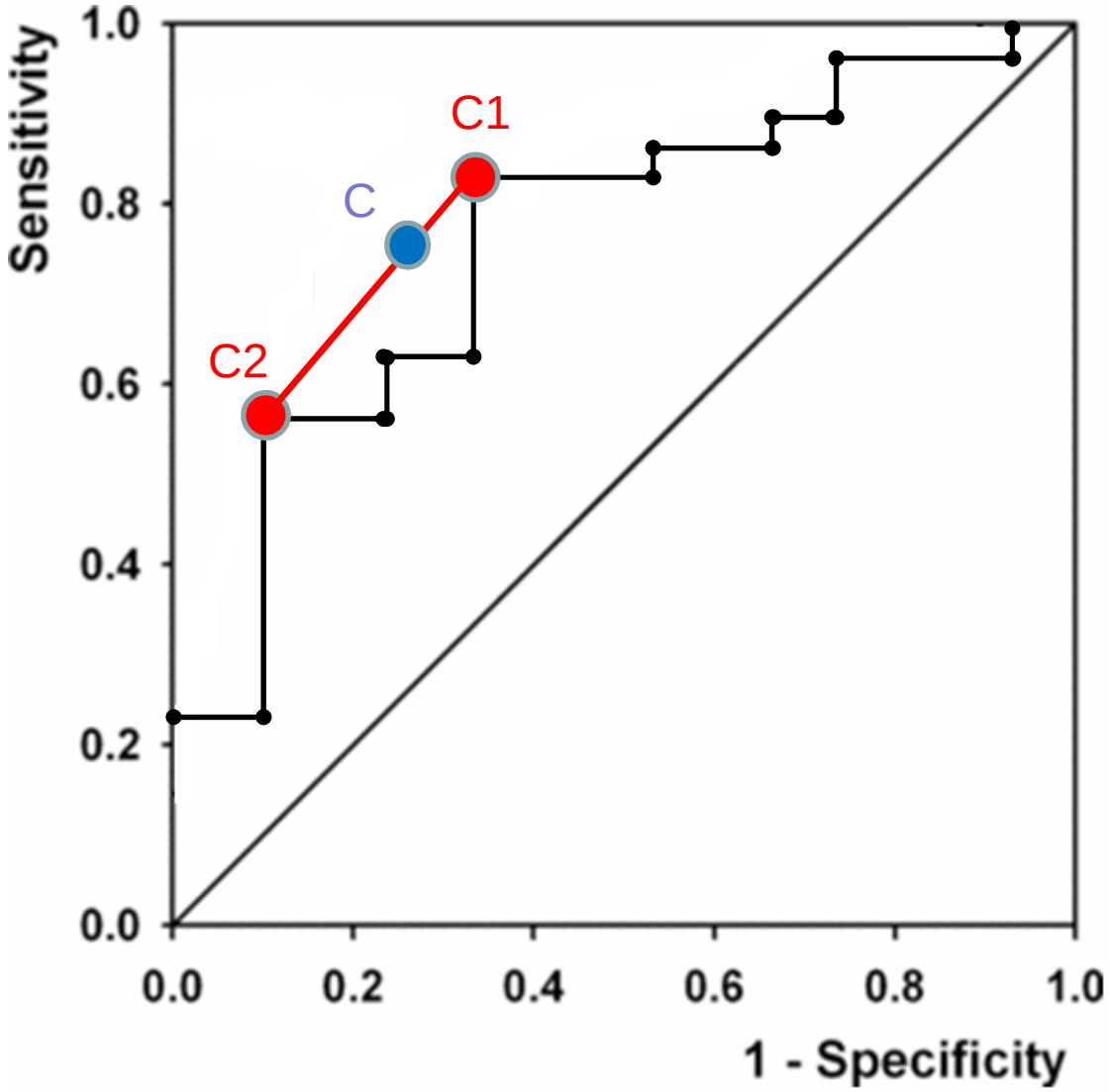
ROC curve in practice



ROC Convex Hull



ROC Convex Hull

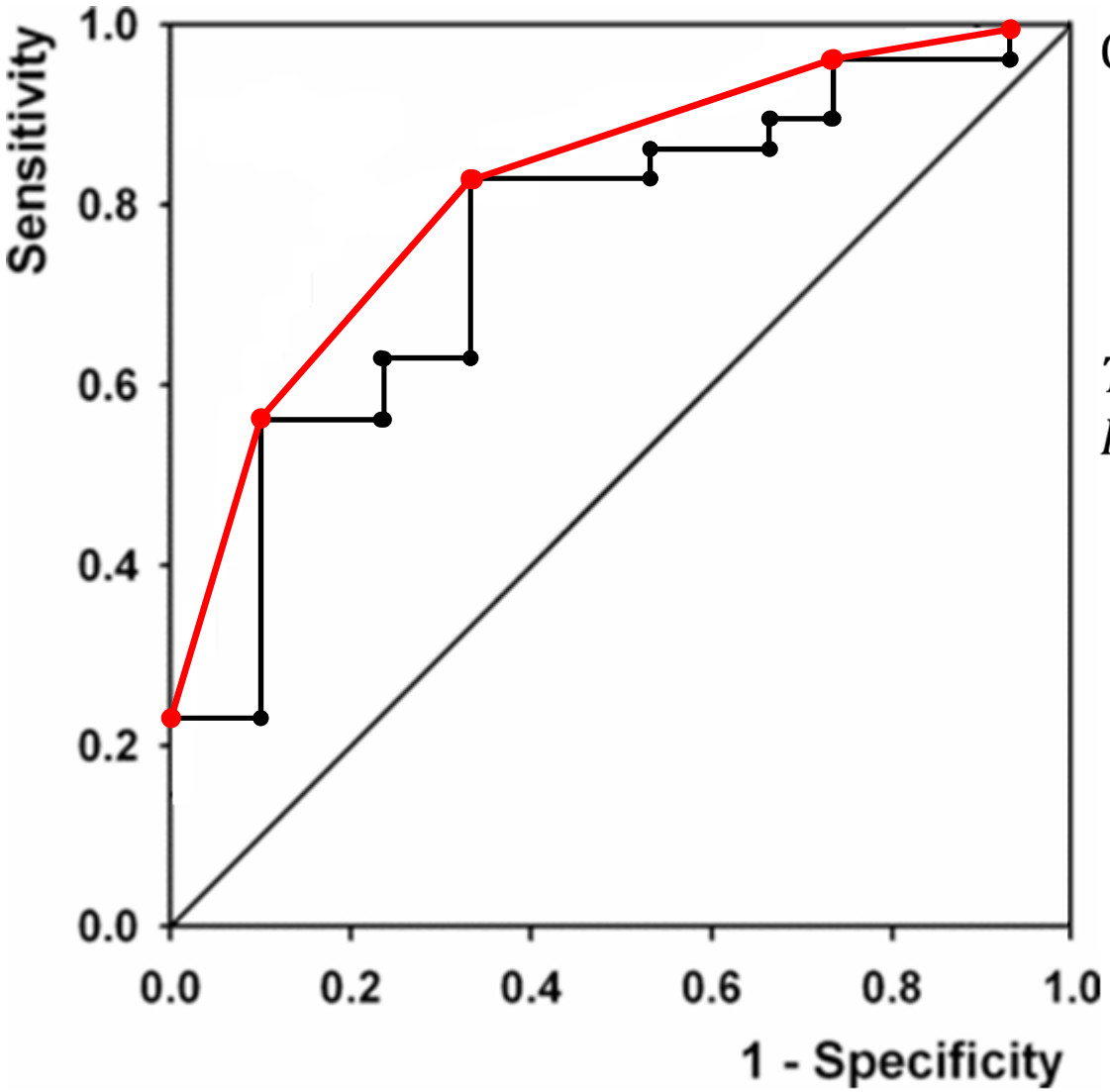


Combined classifier:

$$C = \begin{cases} C1 & 75\% \\ C2 & 25\% \end{cases}$$

$$TPR_C = 0.75\, TPR_{C1} + 0.25\, TPR_{C2}$$
$$FPR_C = 0.75\, FPR_{C1} + 0.25\, FPR_{C2}$$

ROC Convex Hull



Combined classifier:

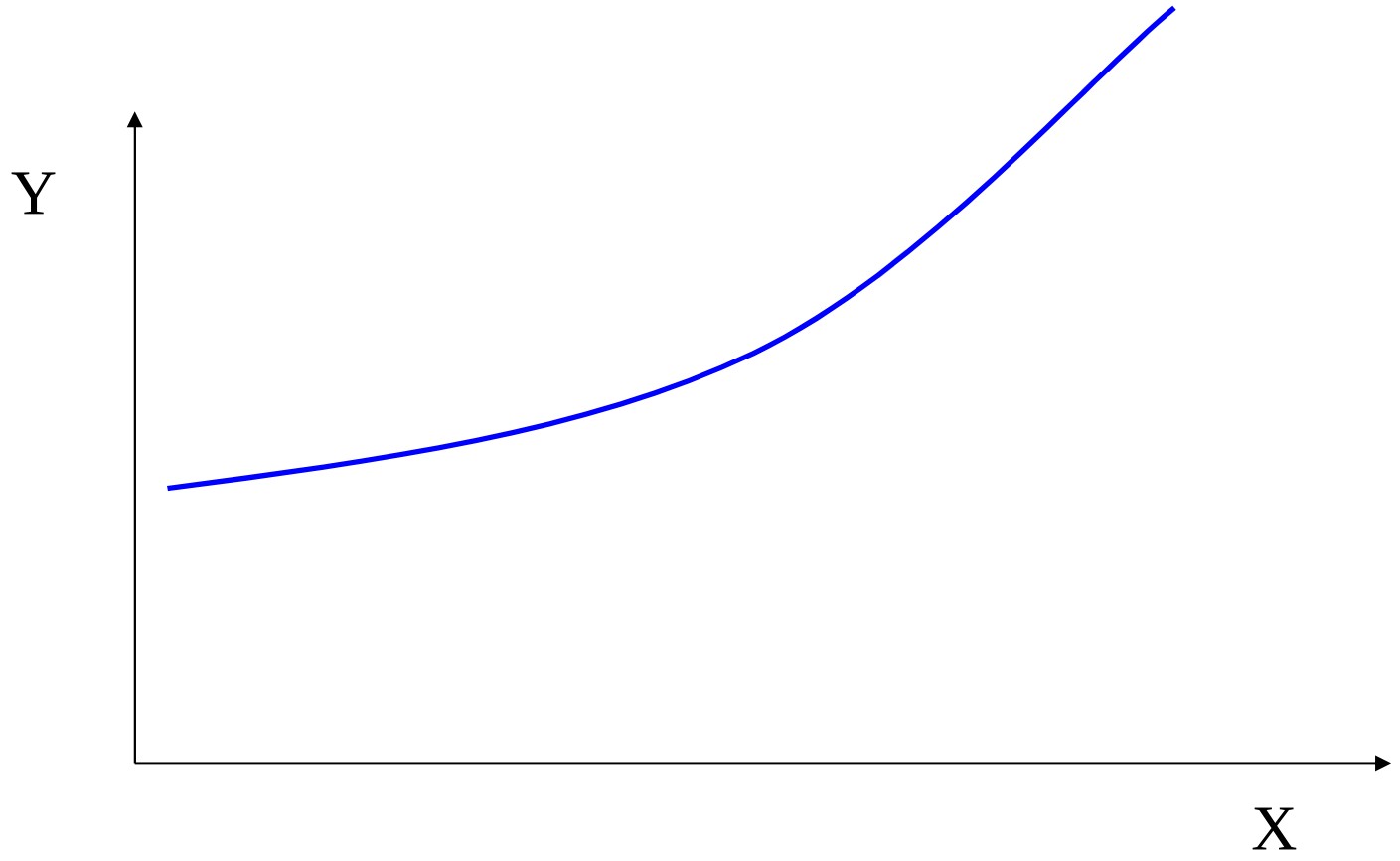
$$C = \begin{cases} C1 & 75\% \\ C2 & 25\% \end{cases}$$

$$TPR_C = 0.75\, TPR_{C1} + 0.25\, TPR_{C2}$$
$$FPR_C = 0.75\, FPR_{C1} + 0.25\, FPR_{C2}$$

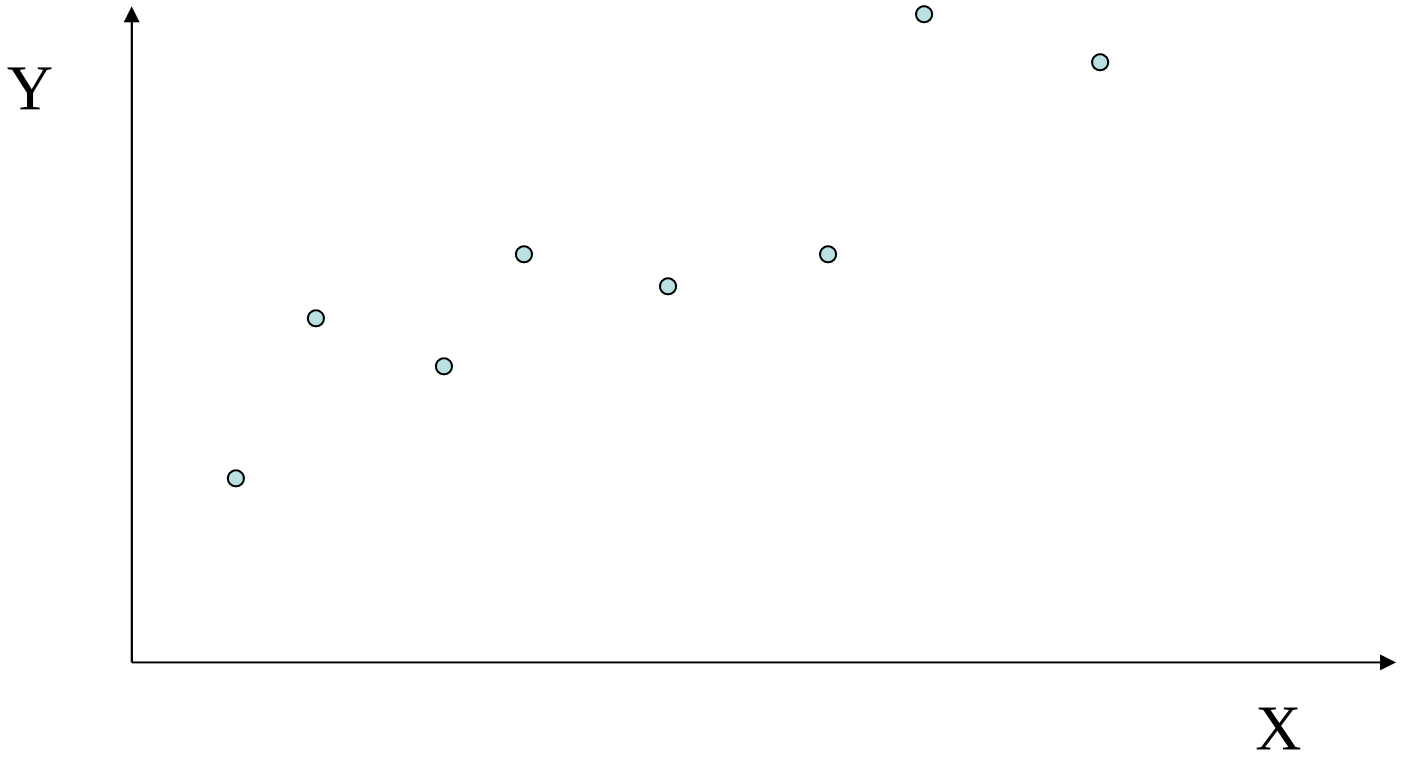
Generalization VS Overfitting

- Training performance
  - Overfitting of the training data
- Test performance
  - Capacity to generalize the model
- We want classifier with high generalization performance
- It is always easy to obtain a good training performance

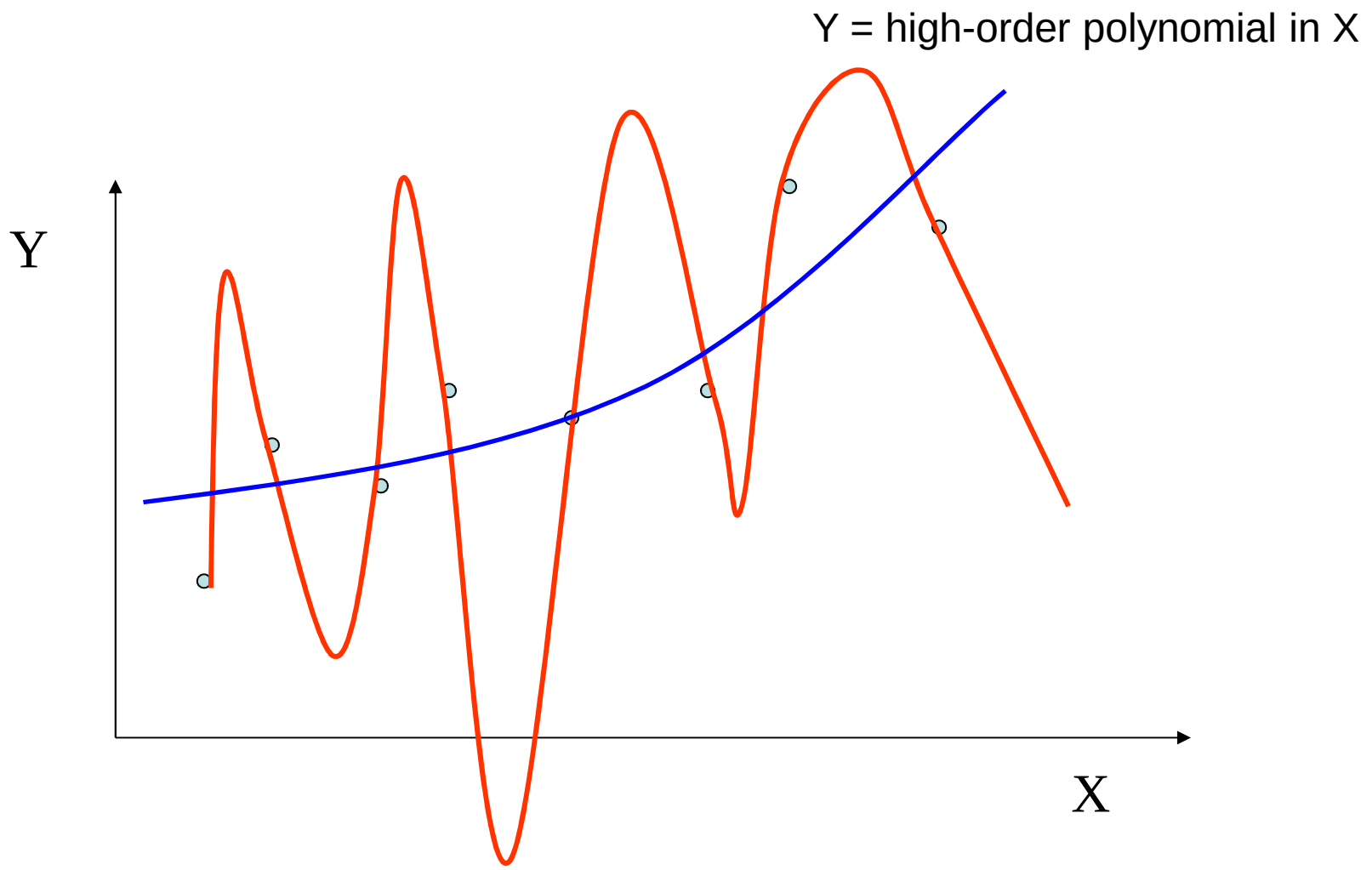
Example



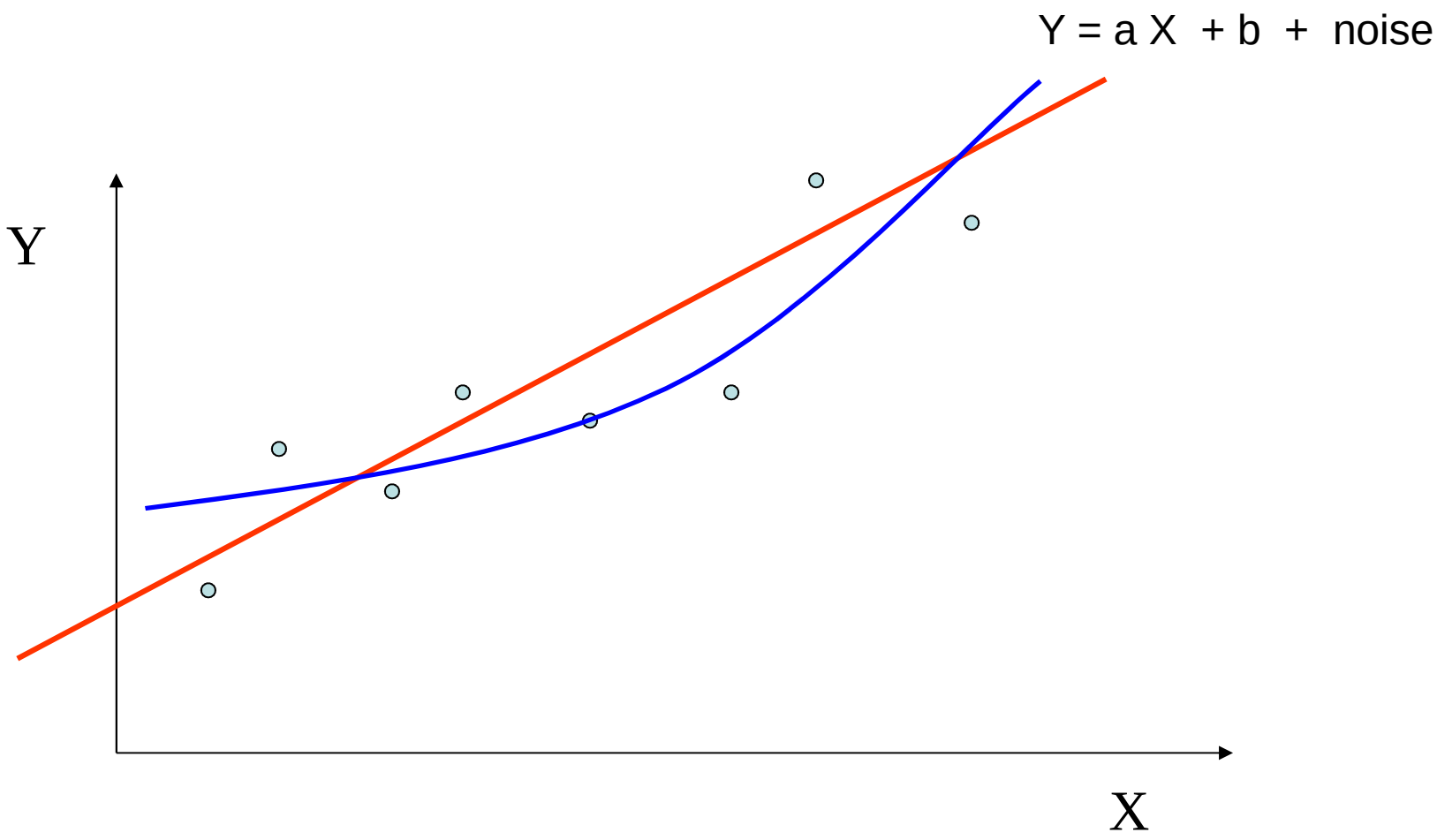
Example



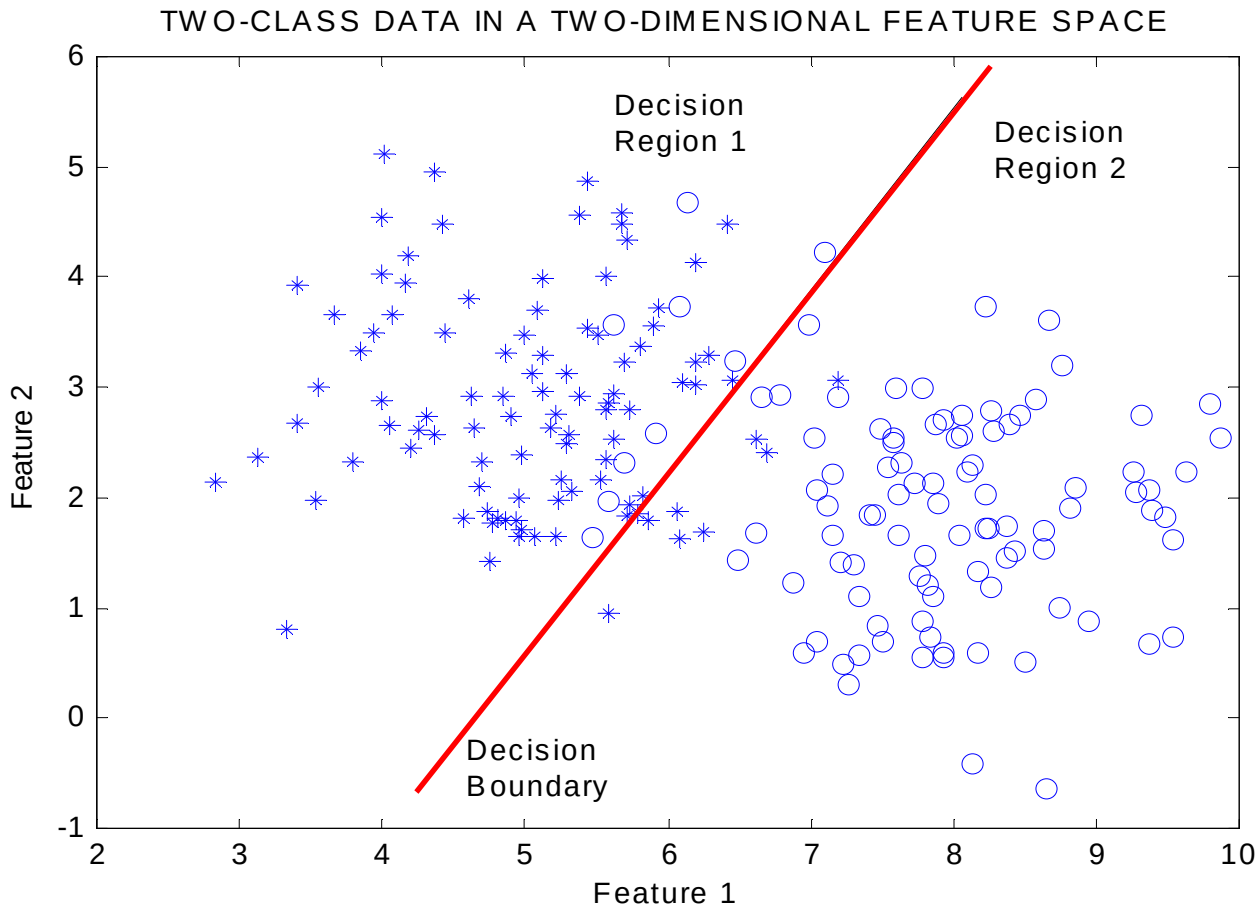
Model too complex



Simple model



Example

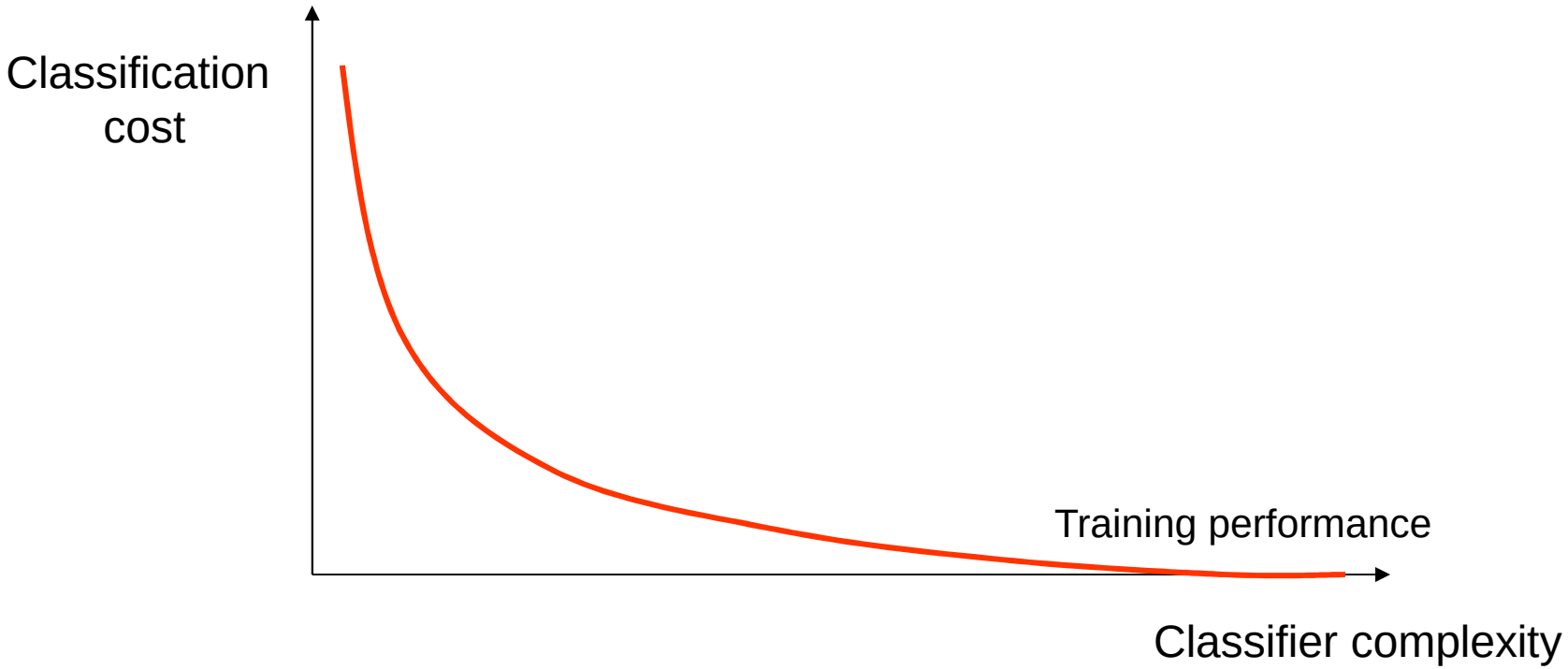

TWO-CLASS DATA IN A TWO-DIMENSIONAL FEATURE SPACE
Decision
Region 1
Decision
Region 2
Decision
Boundary
Feature 2
Feature 1
6
5
4
3
2
1
0
-1
2
3
4
5
6
7
8
9
10

Example


TWO-CLASS DATA IN A TWO-DIMENSIONAL FEATURE SPACE
Decision
Region 1
Decision
Region 2
Decision
Boundary
Feature 2
Feature 1
6
5
4
3
2
1
0
-1
2
3
4
5
6
7
8
9
10

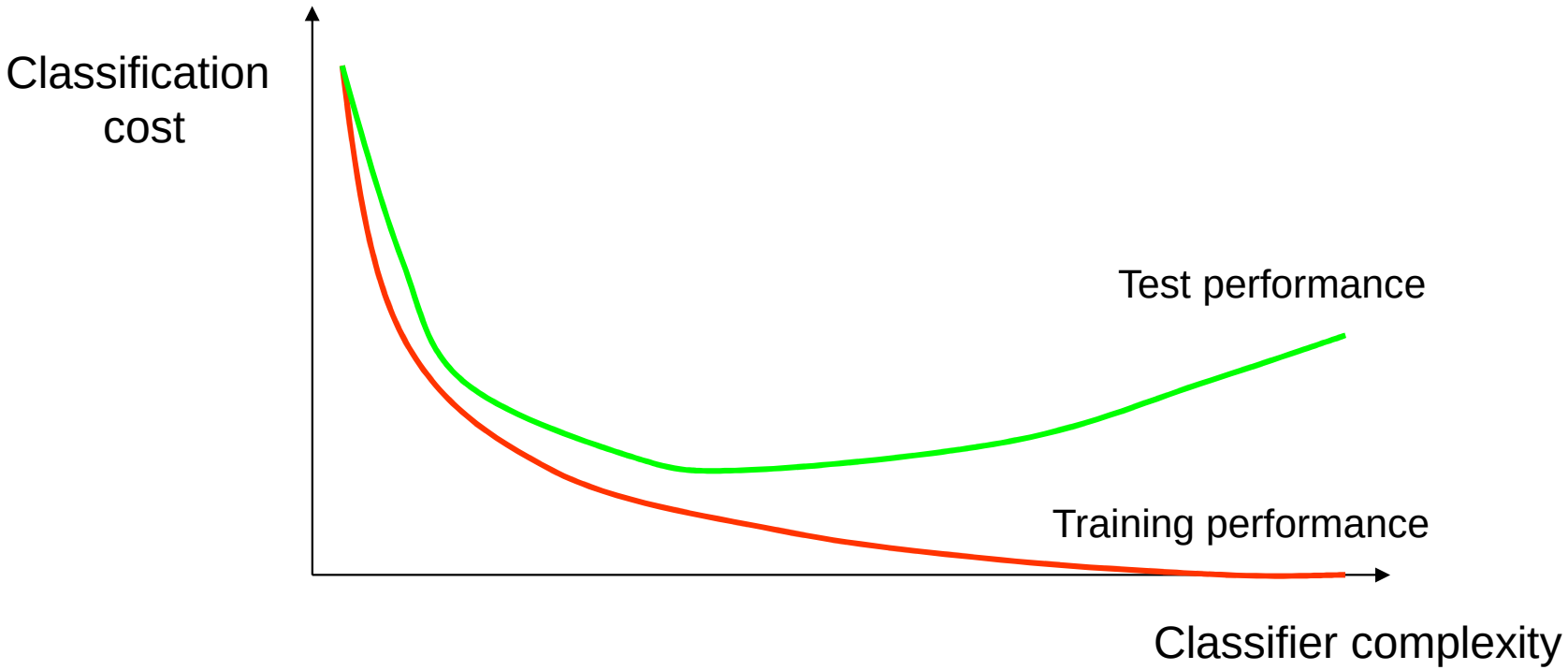
Overfitting

How the overfitting impacts the performance of classifier?



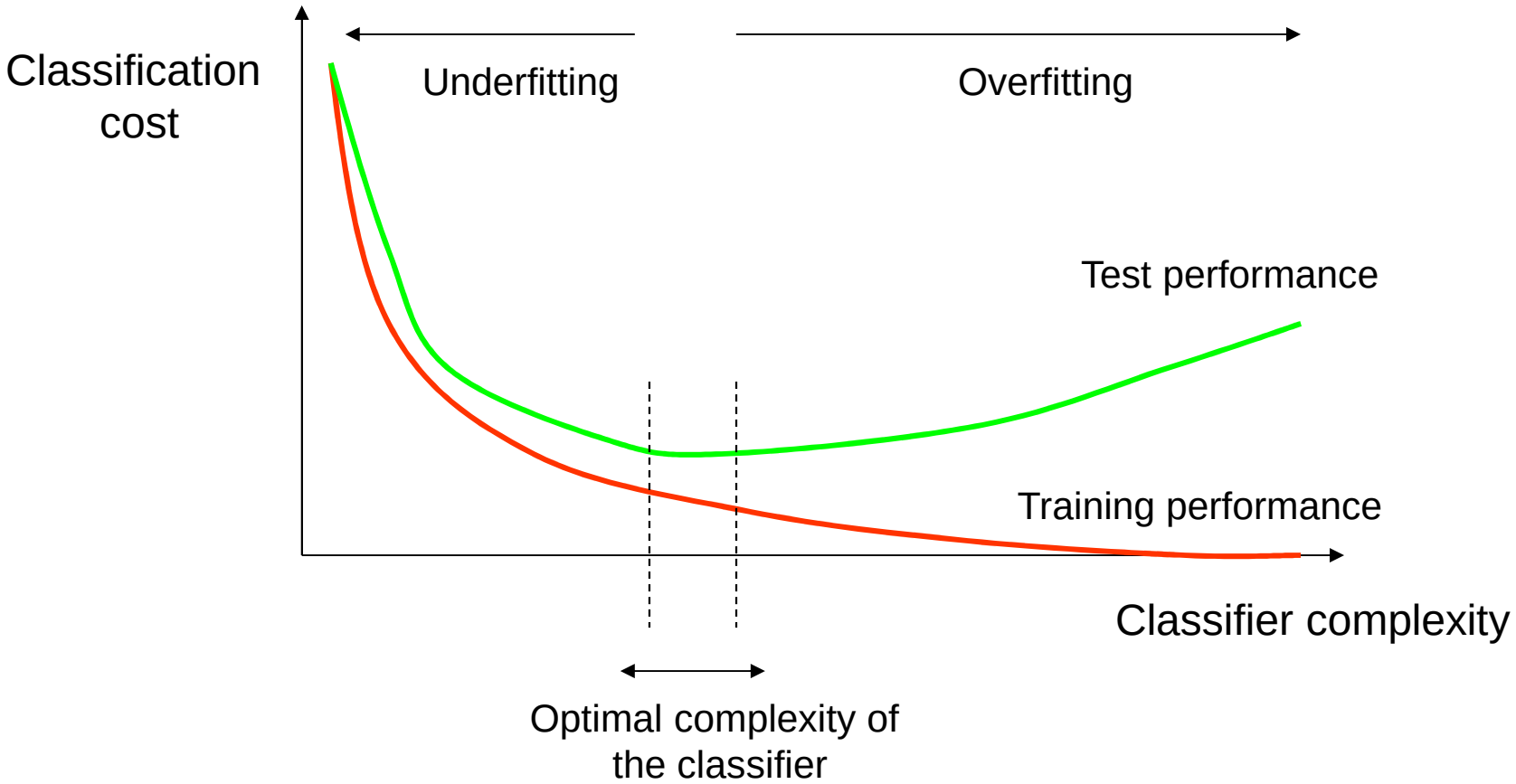
Overfitting

How the overfitting impacts the performance of classifier?



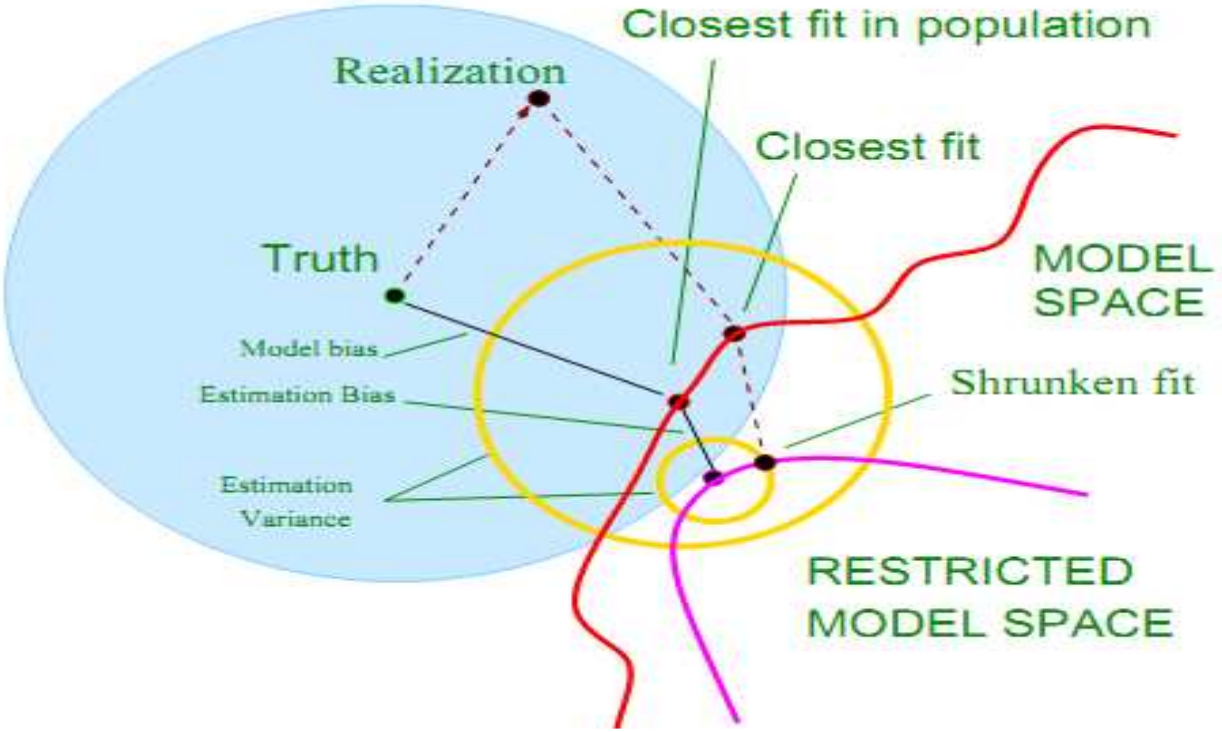
Overfitting

How the overfitting impacts the performance of classifier?



Bias-variance decomposition

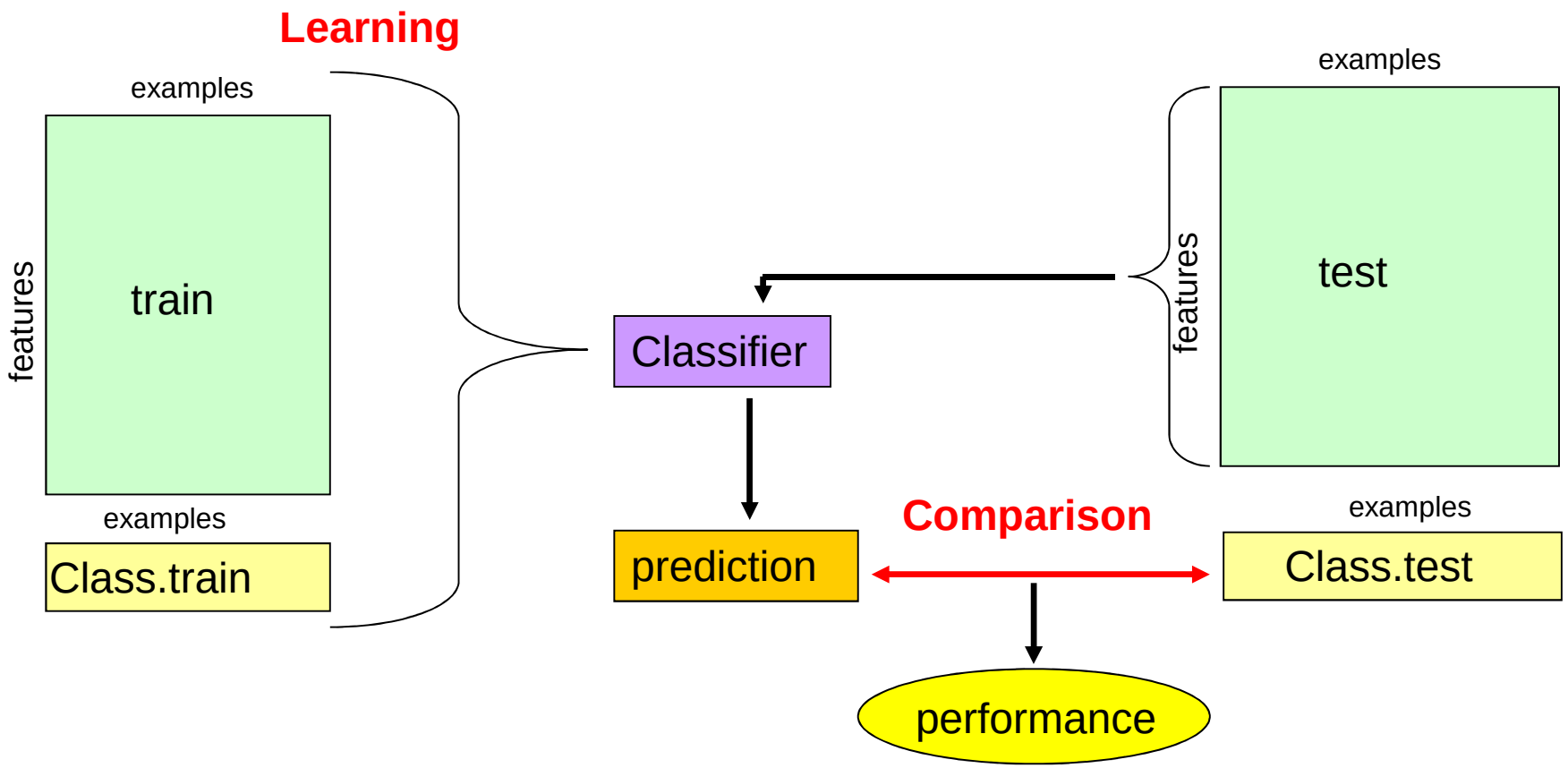
$$
\begin{aligned}
\mathrm{Err}(x_0) &= E[(Y - \hat{f}(x_0))^2 | X = x_0] \\
&= \sigma_\varepsilon^2 + [E\hat{f}(x_0) - f(x_0)]^2 + E[\hat{f}(x_0) - E\hat{f}(x_0)]^2 \\
&= \sigma_\varepsilon^2 + \mathrm{Bias}^2(\hat{f}(x_0)) + \mathrm{Var}(\hat{f}(x_0)) \\
&= \text{Irreducible Error} + \mathrm{Bias}^2 + \text{Variance}.
\end{aligned}
$$



Training and test datatset

- Training datatset
  - The classes are known
  - Use for model learning
- Test datatset
  - The classes are unknown
  - Not use in the model learning
  - Use to estimate the performance of the classifier

Training and test datatset



Decision function

Decision function of the classes

$$g(x) = P(\omega_1 \mid x) - P(\omega_2 \mid x)$$

$$= \ln \frac{P(x \mid \omega_1)}{P(x \mid \omega_2)} + \ln \frac{P(\omega_1)}{P(\omega_2)}$$

$$g_i(x) = \ln P(x \mid \omega_i) + \ln P(\omega_i)$$

The distribution of the classes is generally unknown

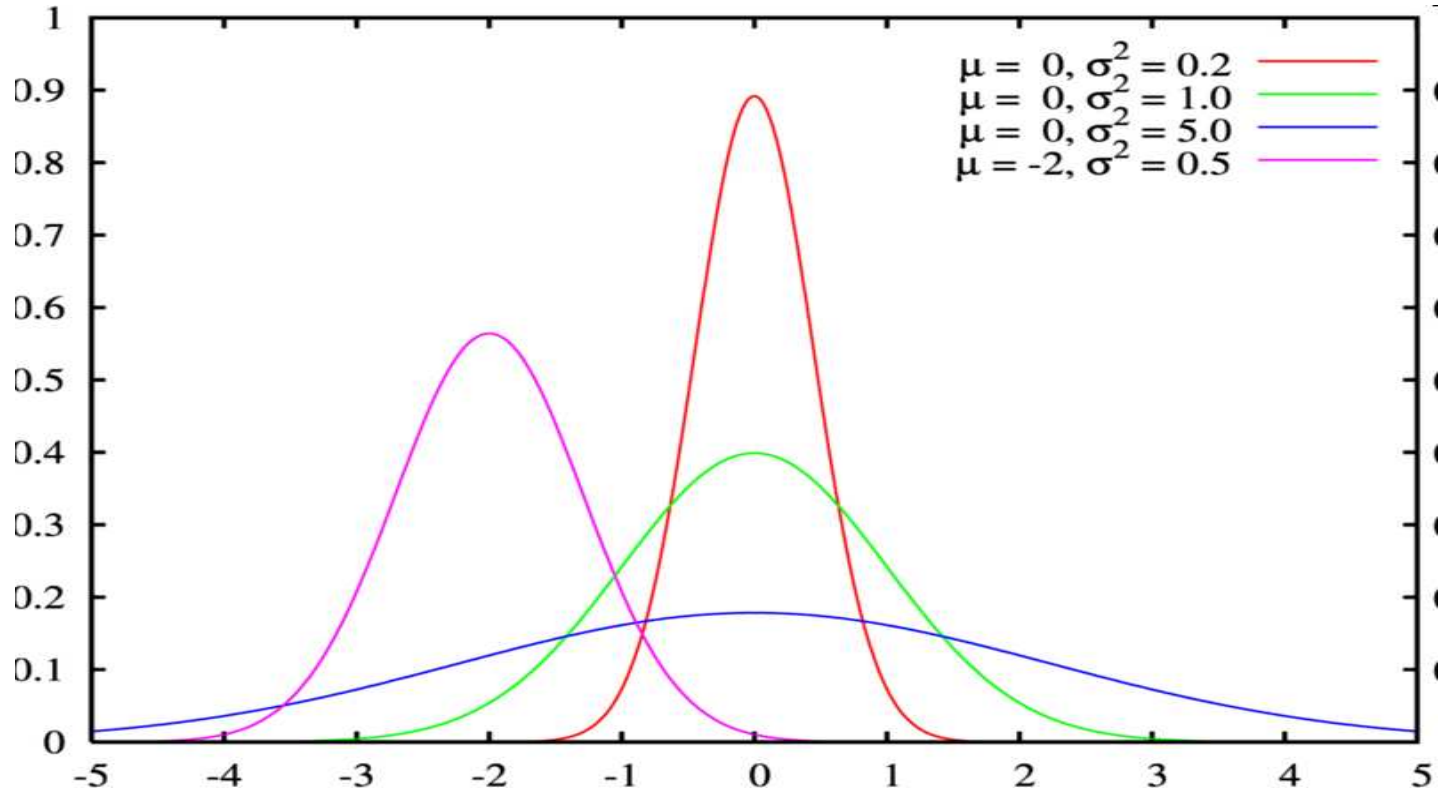
We could put some assumptions on this distribution.
-> Gaussian assumption is the simplest and most used

Gaussian distribution

2 parametrs

- μ : mean
- σ : standart deviation

$$p(x) = \frac{1}{\sqrt{2\pi\sigma}} \exp\left[-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)\right]$$



Multivariate Gaussian distribution

Parameters:

- μ : mean vector
- Σ : covariance matrix

$$f_\theta(\boldsymbol{x}) = \frac{1}{(2\pi)^{N/2}\left|\boldsymbol{\Sigma}\right|^{1/2}} e^{-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1}(\boldsymbol{x}-\boldsymbol{\mu})}$$

Decision Fonction

The decision function is:

$$g_i(x) = -\frac{1}{2}\log\left|\Sigma_i\right| - \frac{1}{2}(x-\mu_k)^T \Sigma_k^{-1}(x-\mu_k)_i + \ln \pi_i$$

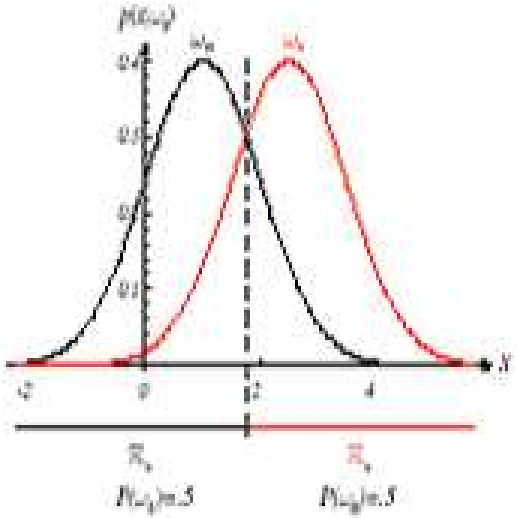
Parameters : $\mu_1, \Sigma_1, \mu_2, \Sigma_2$

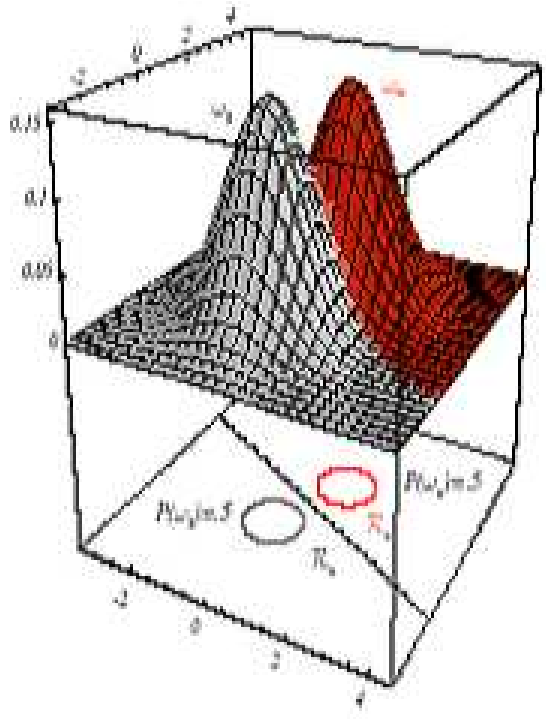
Assumption: The covariance matrix of the classes is identic $\Sigma_i = \Sigma$
i.e. the classes have the same shape

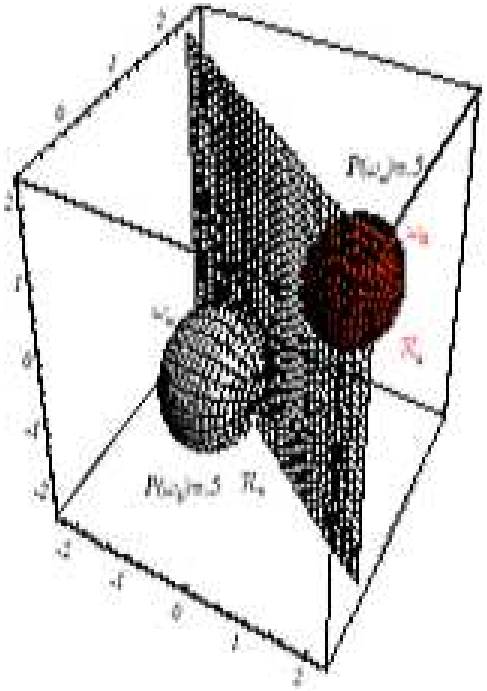
The decision function becomes:

$$g_i(x) = x^T \Sigma^{-1} \mu_i - \frac{1}{2}\mu_i^T \Sigma^{-1} \mu_i + \ln \pi_i$$

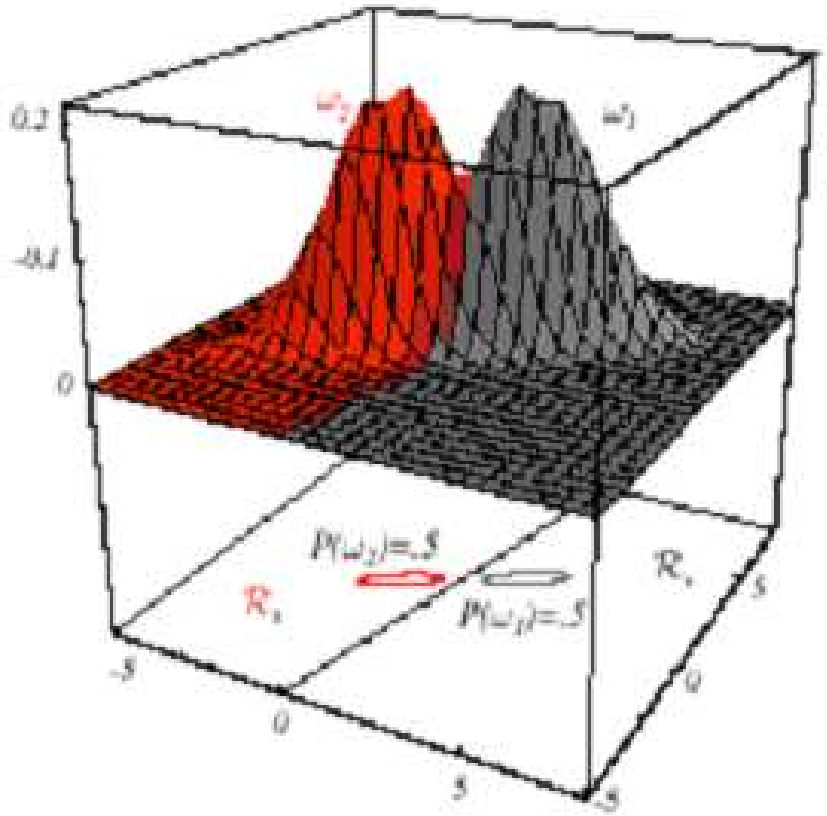
Linear case

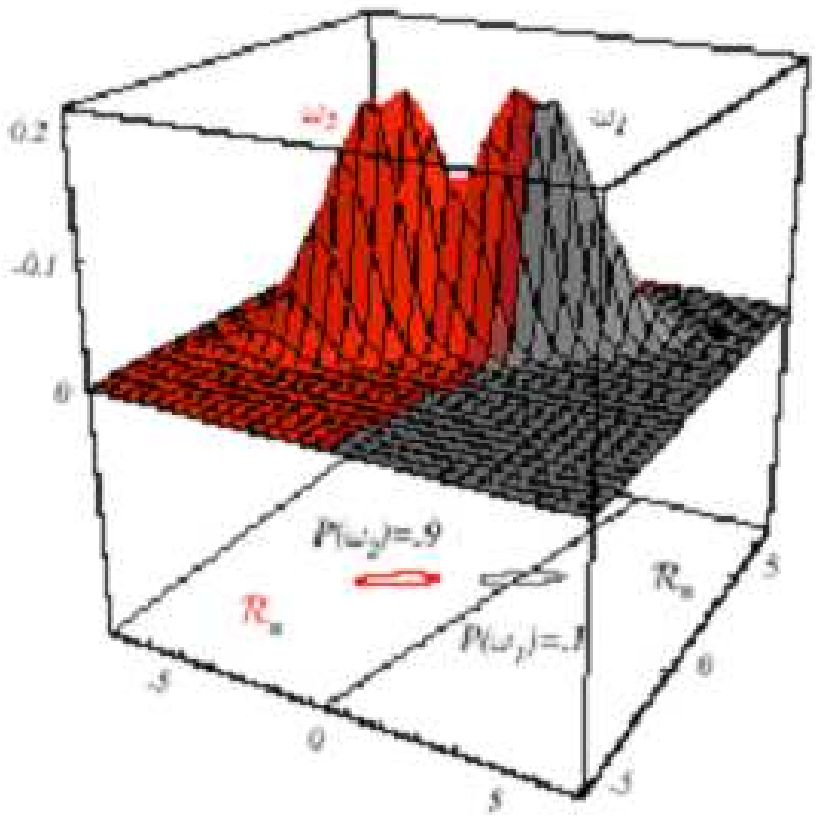




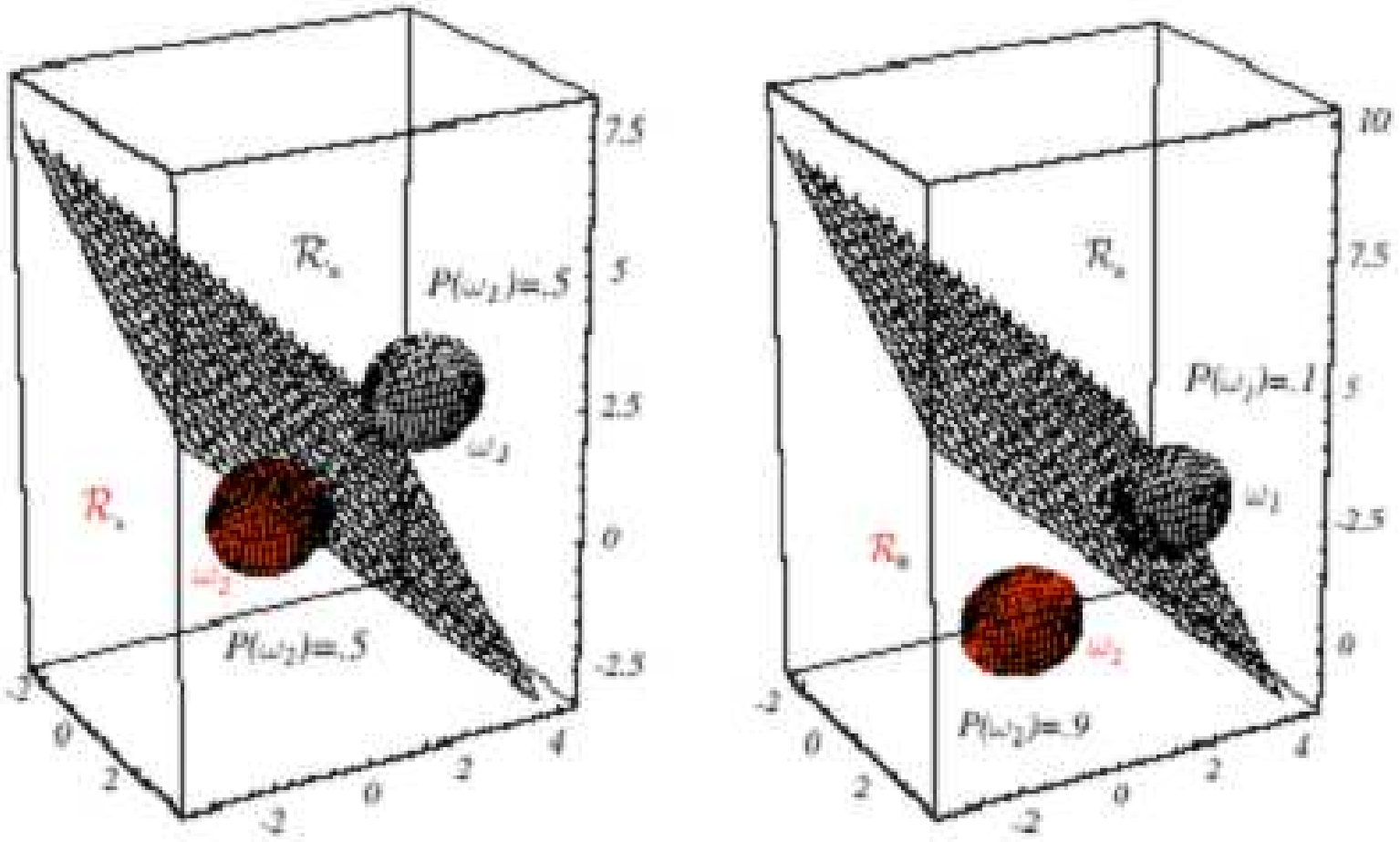


Linear case





Linear case


R_1
$P(\omega_1)=.5$
ω_1
R_2
ω_2
$P(\omega_2)=.5$
7.5
5
2.5
0
-2.5
-2
0
2
-2
0
2
4
R_1
$P(\omega_1)=.1$
ω_1
R_2
ω_2
$P(\omega_2)=.9$
10
7.5
5
2.5
0
-2
0
2
-2
0
2
4

Stage de master

- Laboratoire de recherche: LIPADE
  - Approche méthodologique
  - Sujet applicatif (bioinformatique)

- Société privé
  - ArianaPharma: bioinformatique
  - Orange
  - Thalès