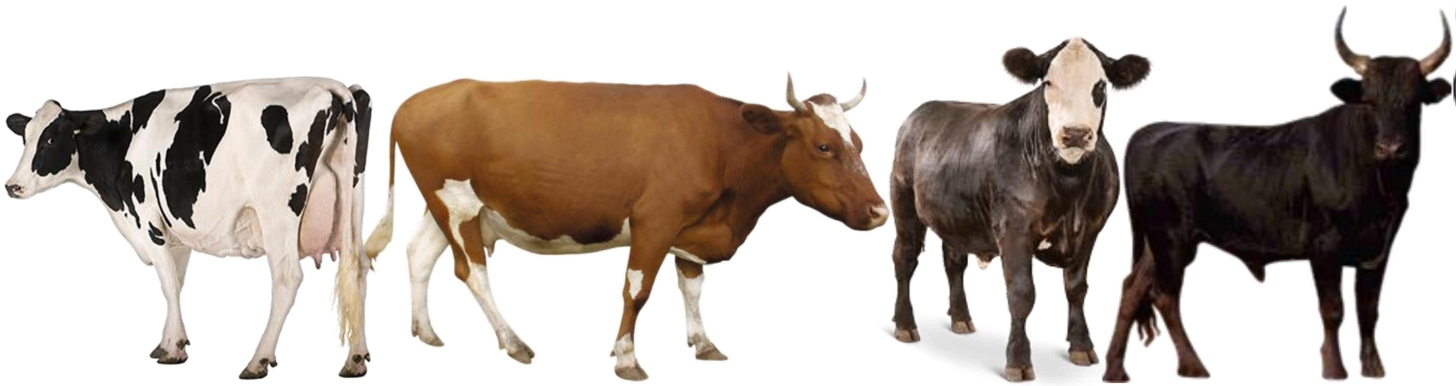
ITU KALEIDOSCOPE
SANTA FE 2018
Machine learning for a 5G future

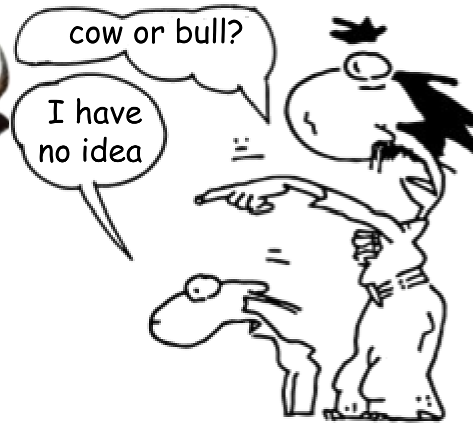
Pattern recognition

Juan Pablo Martín
Universidad Tecnológica Nacional
jpmartin@gicom.com.ar

26-28 November
Santa Fe, Argentina

Objective





© Fontanarrosa

Real World

Decision rule

Detection Problem

The general problem can be written as:

$$\begin{cases} \omega_0 : x = b & \text{hypothesis “noise alone”} \\ \omega_1 : x = b + s & \text{hypothesis “signal + noise”} \end{cases}$$

© Fontanarrosa

The aim is to build a classifier *d*, or a decision rule *d* which minimizes a criteria (e.g. the error probability)

$$P_e(d) = P(d(\boldsymbol{X}) \neq Y)$$

where $\boldsymbol{X}$ is an observation and is Y the associated hypothesis.

The strategy to design a solution to this problem depends on the nature of the available information on the problem

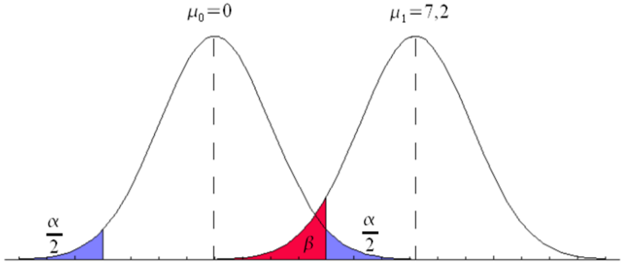
Resolution Approaches

Rule based

- If $\boldsymbol{X}$ is grey and $\text{weight}(\boldsymbol{X}) > \mathbf{1000}$, then
  - $\boldsymbol{X}$ is an elephant
- else
  - $\boldsymbol{X}$ is a mouse
- Needs and expert to translate knowledge into **rules**; are complex, long, not reliable.

Hypothesis testing

$$\begin{cases} H_0: X \in \omega_0 & X \sim p(X|\omega_0) \\ H_1: X \in \omega_1 & X \sim p(X|\omega_1) \end{cases}$$



Bayes test

$$\begin{cases} H_0: X \in \omega_0 & X \sim p(X|\omega_0) \\ H_1: X \in \omega_1 & X \sim p(X|\omega_1) \end{cases}$$

$$P_e = P(D_0|\omega_1)P(\omega_1) + P(D_1|\omega_0)P(\omega_0)$$

minimize

$$P(x|\omega_1)P(\omega_1) \underset{D_0}{\overset{D_1}{\gtrless}} P(x|\omega_0)P(\omega_0)$$

$$\underbrace{\frac{P(x|\omega_1)}{P(x|\omega_0)}}_{q(x)} \underset{D_0}{\overset{D_1}{\gtrless}} \underbrace{\frac{P(\omega_0)}{P(\omega_1)}}_{threshold}$$

Likelihood ratio

Bayes test – Example: Gaussian case

Linear function!

$$\underbrace{V^T X}_{q(x)} + \underbrace{U}_{constant\ threshold} \underset{D_0}{\overset{D_1}{\gtrless}} 0$$

Real World Problems

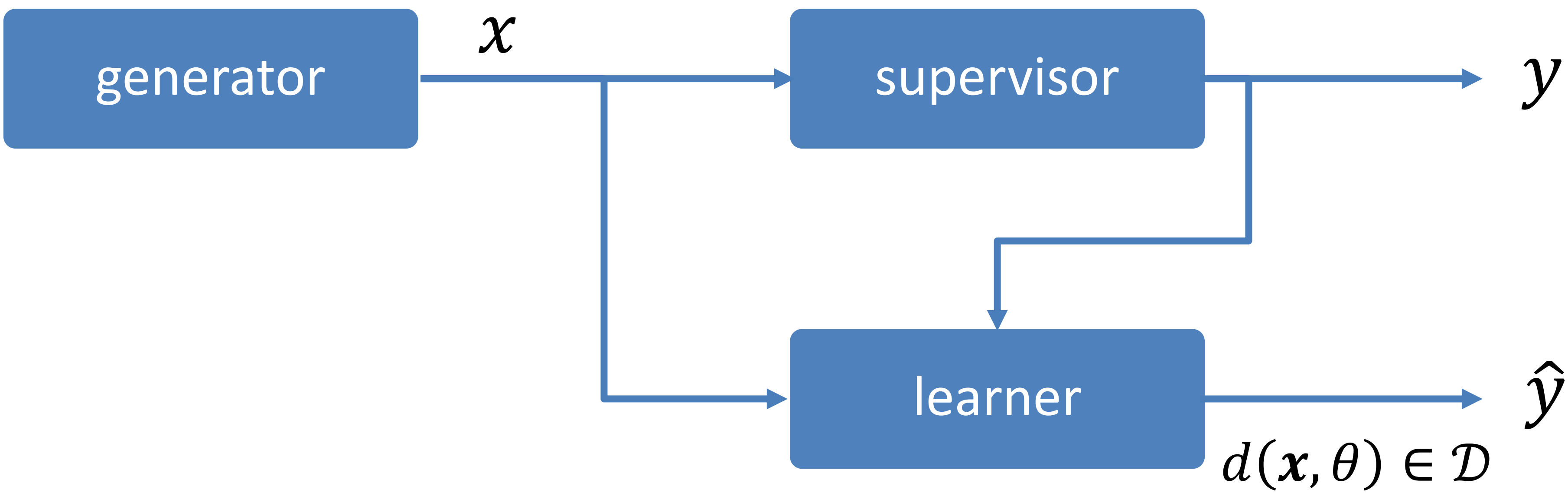
Knowing the distribution function

- $p(X|\omega_0)$ and $p(X|\omega_1)$ has to be known …

If that is not the case

- Can asume that
  - $p(X|\omega_0)$ and $p(X|\omega_1) \subset \mathcal{F}_\theta$
- then estimate θ based on data and plugin the estimator in the decision rule

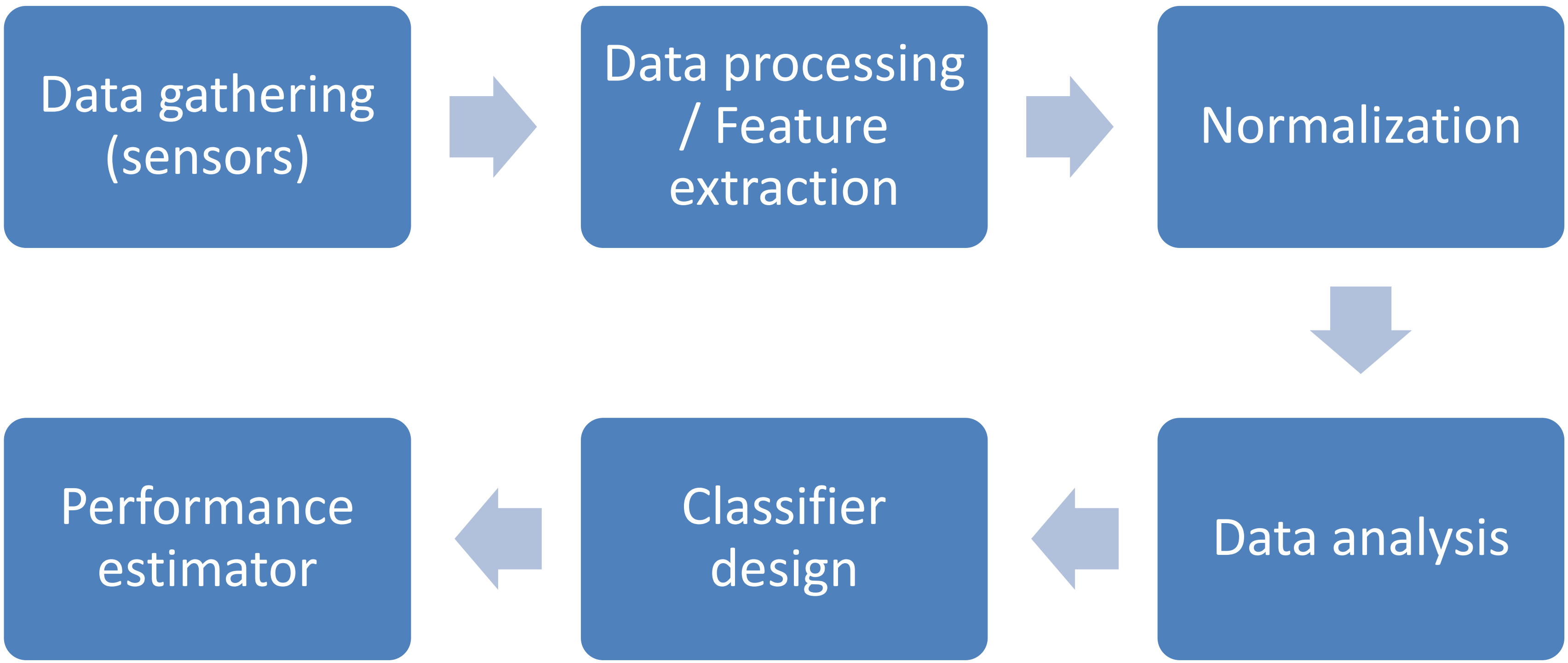
Learning Model



The knowledge of the probabilistic model is replaced by

$A_n = \{(\boldsymbol{x}_1, y_1), (\boldsymbol{x}_2, y_2), \dots, (\boldsymbol{x}_n, y_n)\}$ $\leftarrow$ training set of data

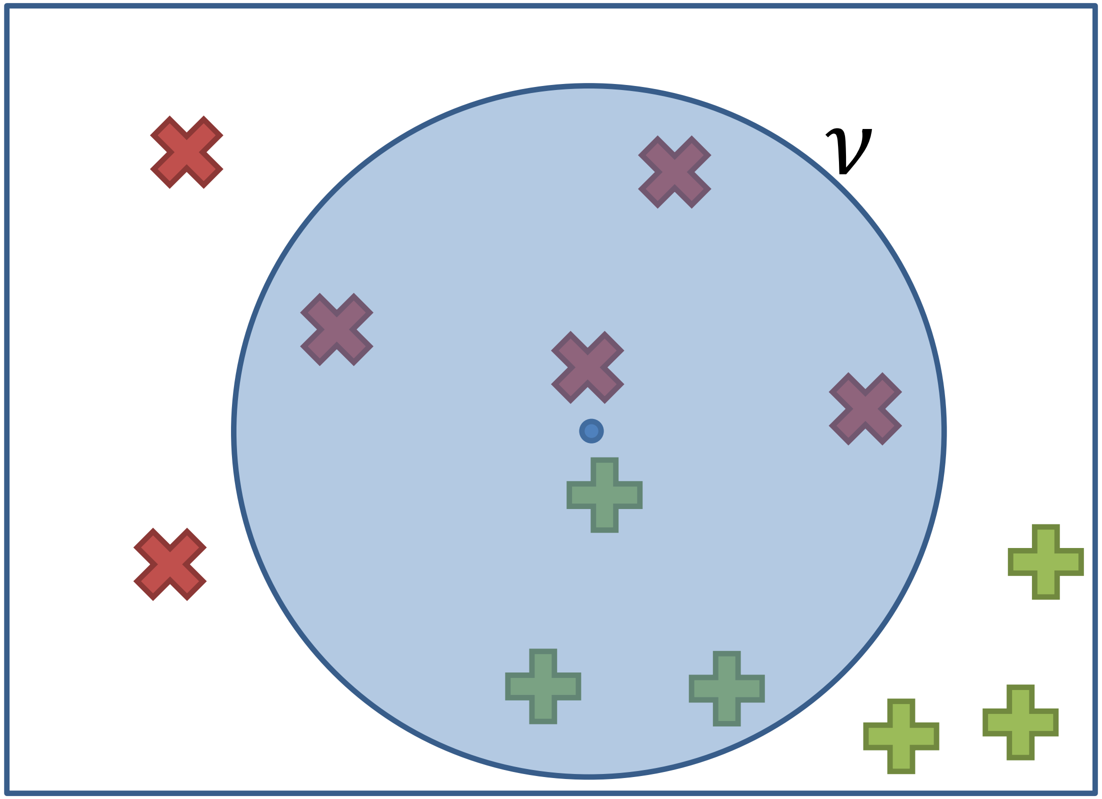
Non Parametric Methods



Non Parametric Methods

- Parzen density estimator
- k-nearest neighbor estimator

kNN Method

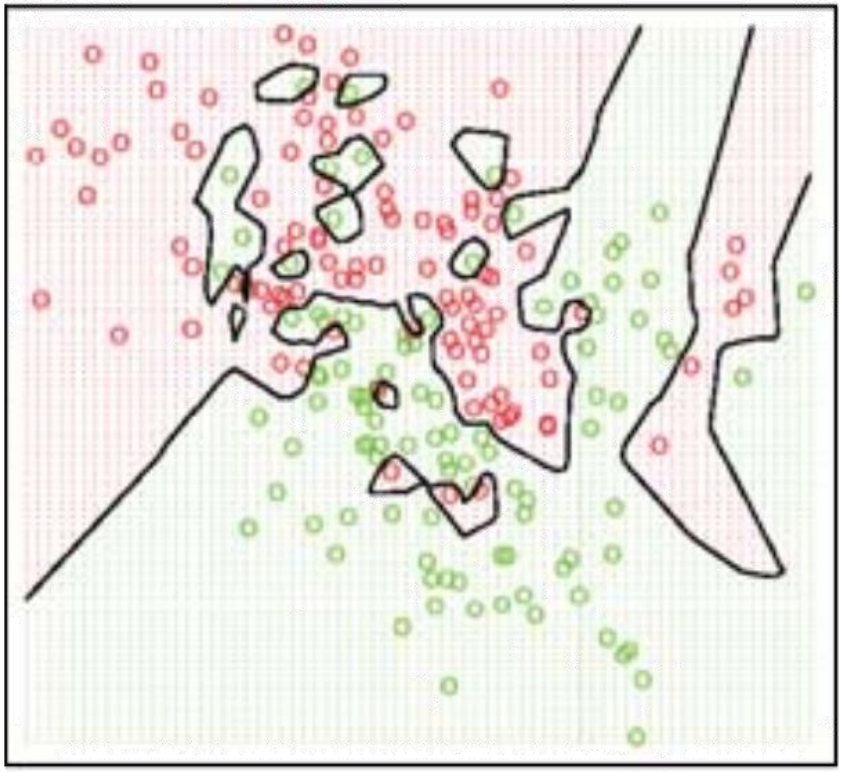


- Centered at the unlabeled sample a spherical volumen ν is enlarged
- when k samples fall inside the volumen
- the proportion is counted
- the new sample **is labeled** with the most occurent class.

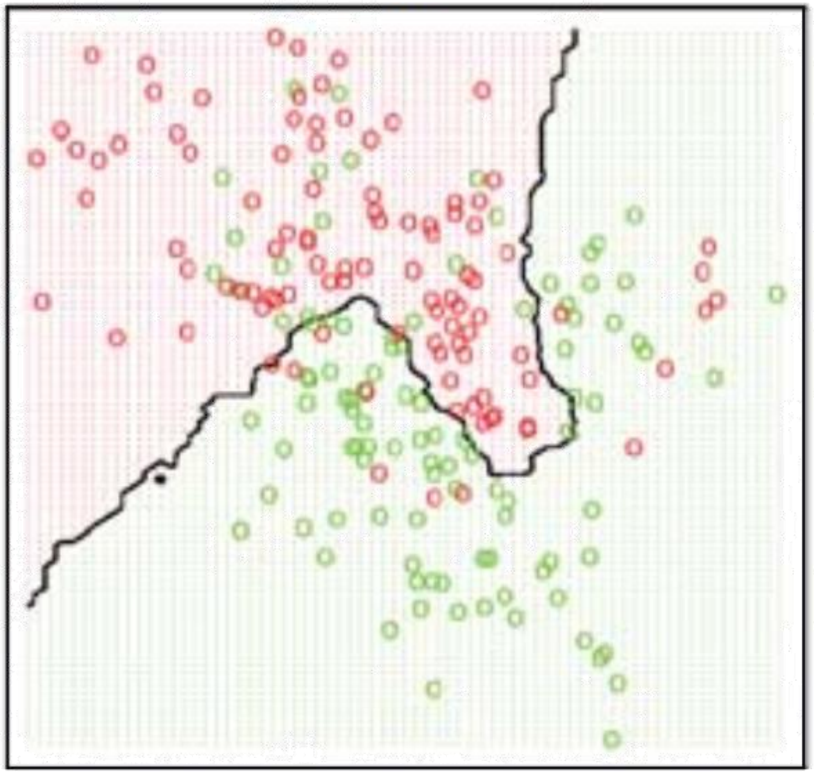
✓ For the two classes, k should be odd.

Examples

1NN



15NN



Support Vector Machines

Problem of Functional Learning

Find within $\mathcal{D} = \{d(\boldsymbol{x}, \theta) : \theta \in \Theta\}$, the function which gives the best approximation of y according to a risk functional

$$J(d) = \int Q(\underbrace{d(\boldsymbol{x}, \theta)}_{\hat{y}}, y) p(\boldsymbol{x}, y)\, d\boldsymbol{x}\, dy$$

where Q expresses the cost associated with each couple $(\boldsymbol{x}, y)$.

Example of a cost function: error probability

$$P_e(d) = \int \mathbb{1}_{d(\boldsymbol{x},\theta) \neq y}\, p(\boldsymbol{x}, y)\, d\boldsymbol{x}\, dy$$

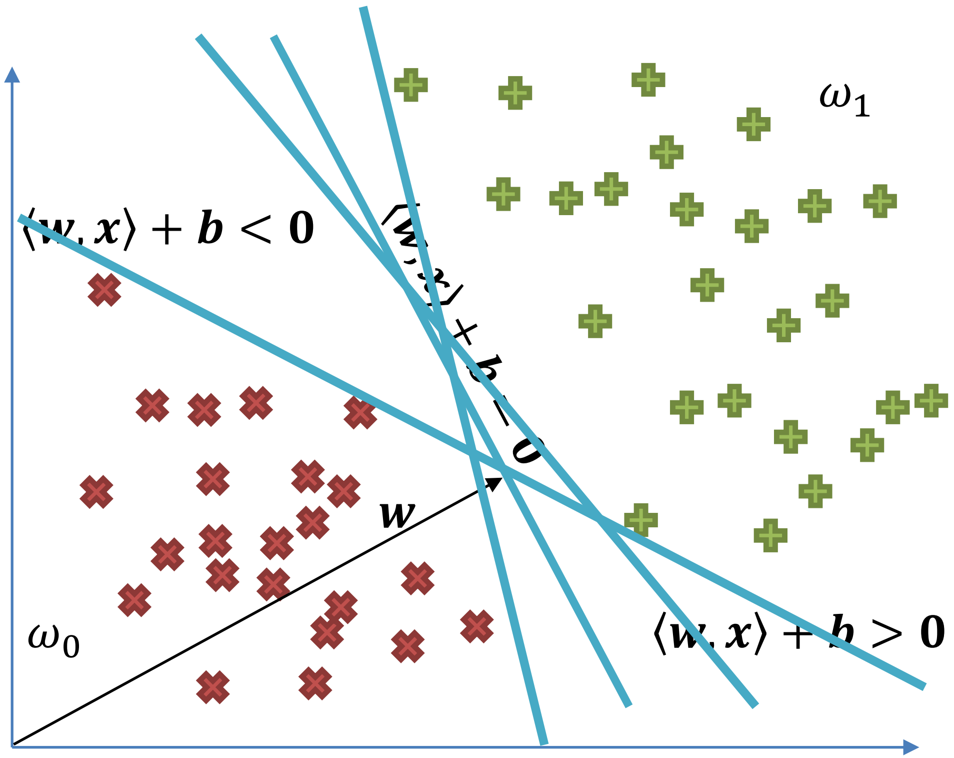
Problem of Functional Learning

As the density $p(\boldsymbol{x}, y)$ is unknown, the minimization of $J(d)$ is done by plugging an estimator; the empirical risk

$$J_{emp}(d) = \frac{1}{n}\sum_{k=1}^{n} Q(d(\boldsymbol{x}_k, \theta), y_k)$$

$$P_{emp}(d) = \frac{1}{2n}\sum_{k=1}^{n} |y_k - \underbrace{d(\boldsymbol{x}_k, \boldsymbol{w}, \mathrm{b})}_{\hat{y}}|$$

Induction principle

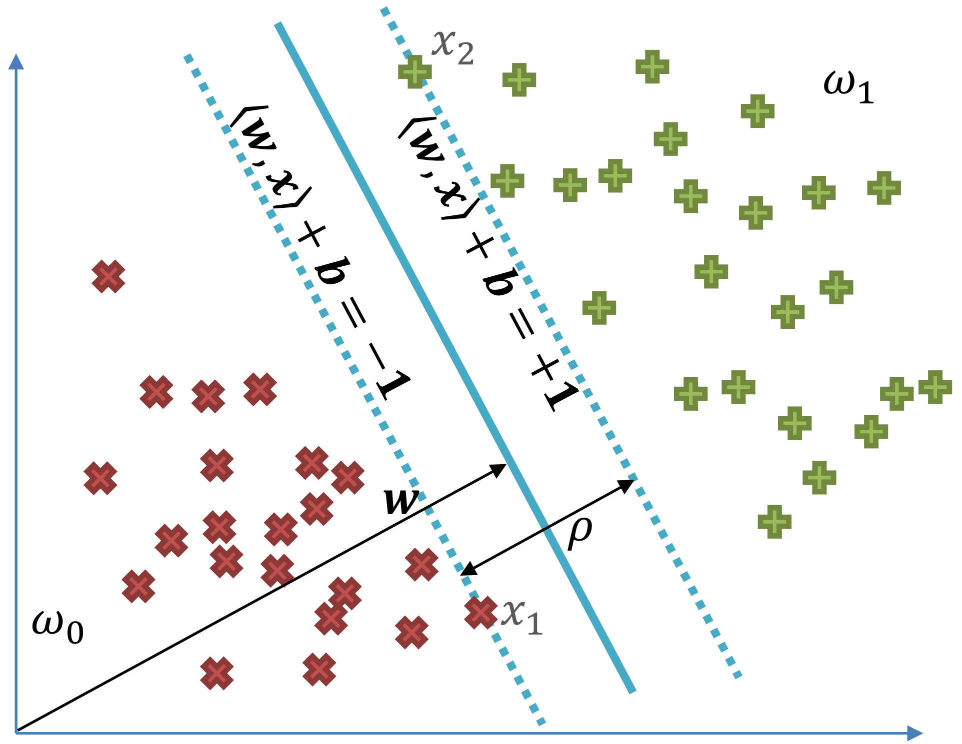


Which separating hyperplane is the best choise?

Should be the one that maximizes the margin!

- Vapnik (1965, 1992)

Margin Calculation



$$\rho = \left\langle \frac{w}{||w||}, x_2 - x_1 \right\rangle = \frac{2}{||w||}$$

Maximizing the margin, is equivalent to:

- minimizing $\frac{1}{2}||w||^2$
- under the constrains
  $y_i(\langle \boldsymbol{w}, \boldsymbol{x_i} \rangle + b) \geq 1,$
  $1 \leq i \leq n$

Only valid for linerarly separable classes

Minimization (Lagrange Multipliers)

Minimizing a convex function $f(x)$ under the constrains $g_i(x) \leq 0, i = 1, \dots, n$ is equivalent to finding the saddle point of the Lagrangian:

$$L(\boldsymbol{x}, \boldsymbol{\alpha}) = f(\boldsymbol{x}) + \sum_{i=1}^{n} \alpha_i g_i(x)$$

Optimality conditions made with respect to the Lagrangian:

$$L(\boldsymbol{w}, b, \boldsymbol{\alpha}) = \frac{1}{2}\|\boldsymbol{w}\|^2 + \sum_{i=1}^{n} \alpha_i \{y_i(\langle \boldsymbol{w}, \boldsymbol{x_i} \rangle + b) - 1\}, \qquad \alpha_i \geq 0$$

Results in null derivatives with respecto to the primal and dual variables:

$$\frac{\partial}{\partial \boldsymbol{w}} L(\boldsymbol{w}, b, \boldsymbol{\alpha}) = 0 \text{ and } \frac{\partial}{\partial \boldsymbol{b}} L(\boldsymbol{w}, b, \boldsymbol{\alpha}) = 0$$

Dual problem to solve:

$$\sum_{i=1}^{n} \alpha_i^* y_i = 0 \text{ and } \boldsymbol{w}^* = \sum_{i=1}^{n} \alpha_i^* y_i x_i$$

Support Vectors

The optimum separator is:

$$\boldsymbol{w}^* = \sum_{i=1}^{n} \alpha_i^* y_i x_i$$

According to the Karush-Kuhn-Tucker conditions:

$$\alpha_i^* \{ y_i(\langle w^*, x_i \rangle + b^*) - 1 \} = 0, \qquad \forall i$$

Two possible cases:

1. $y_i(\langle w^*, x_i \rangle + b^*) > 1$
   Then $\alpha_i^* = 0$, meaning that x_i is not used to calculate $\boldsymbol{w}^*$.

2. $y_i(\langle w^*, x_i \rangle + b^*) = 1$
   Then $\alpha_i^* \neq 0$ and x_i is on the margin.

Linearly Separable Solution

The optimum separator is:

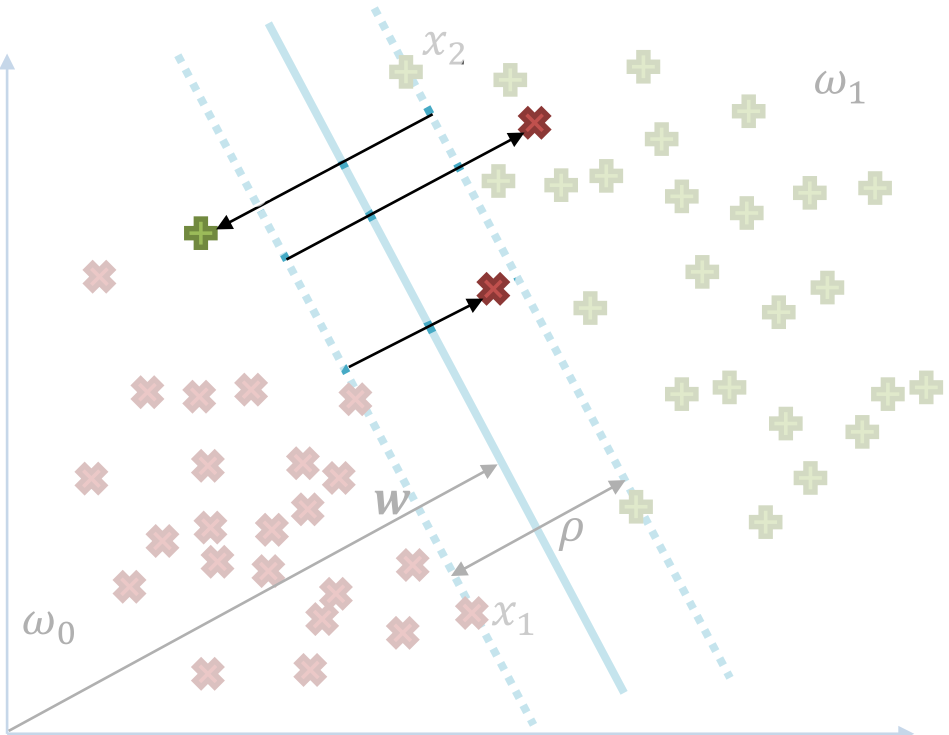
$$\boldsymbol{w}^* = \sum_{sv} \alpha_i^* y_i x_i$$

The decisión rule is:

$$d(\boldsymbol{x}, \boldsymbol{\alpha}^*, b^*) = sign\left(\sum_{sv} \langle \boldsymbol{x}, \boldsymbol{w}^* \rangle + b^*\right)$$

$$d(\boldsymbol{x}, \boldsymbol{\alpha}^*, b^*) = sign\left(\sum_{sv} \alpha_i^* y_i \langle \boldsymbol{x}, \boldsymbol{x}_i \rangle + b^*\right)$$

Classes not linearly separable



- The problem formulation has to be modified.
- Missclassified data is penalized.
  - A cost related to the distance from the sample to the margin is considered.
- A new $f(x)$ is minimized

$$\frac{1}{2}\|w\|^2 + C\underbrace{\sum_{i=1}^{n}\xi_i}_{\text{Penalizes missclassified samples}}, \qquad C \geq 0$$

Non Separable Solution

- A new $f(x)$ is minimized

$$\frac{1}{2}||w||^2 + C\sum_{i=1}^{n}\xi_i\,, \qquad C \geq 0$$

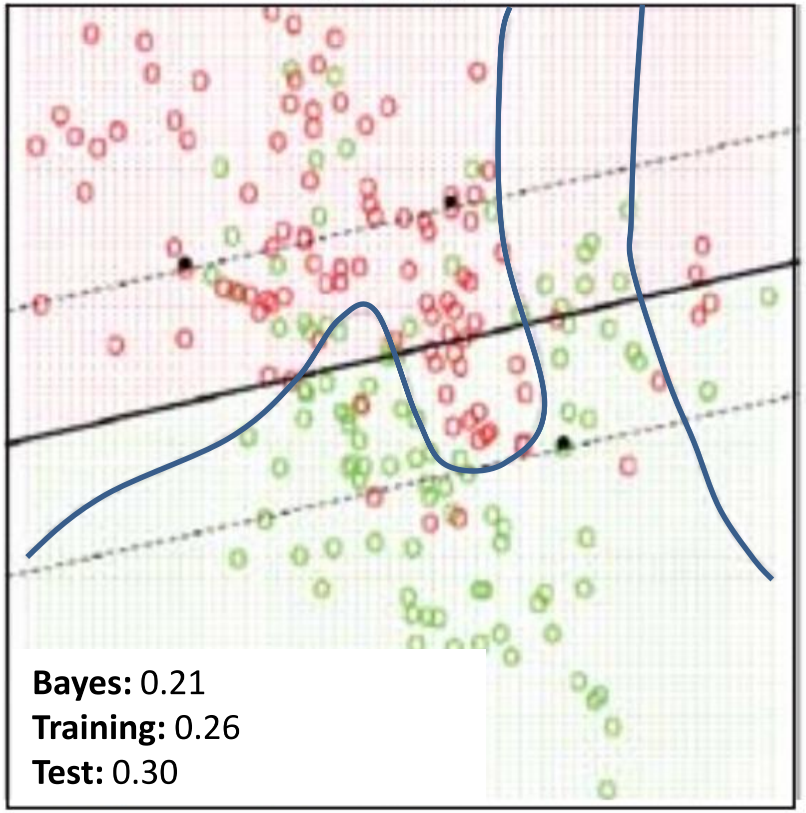
- Under the constraints

$$y_i(\langle \mathbf{w}, \mathbf{x}_i \rangle + b) \geq 1 - \xi_i, \qquad \xi_i \geq 0, 1 \leq i \leq n$$

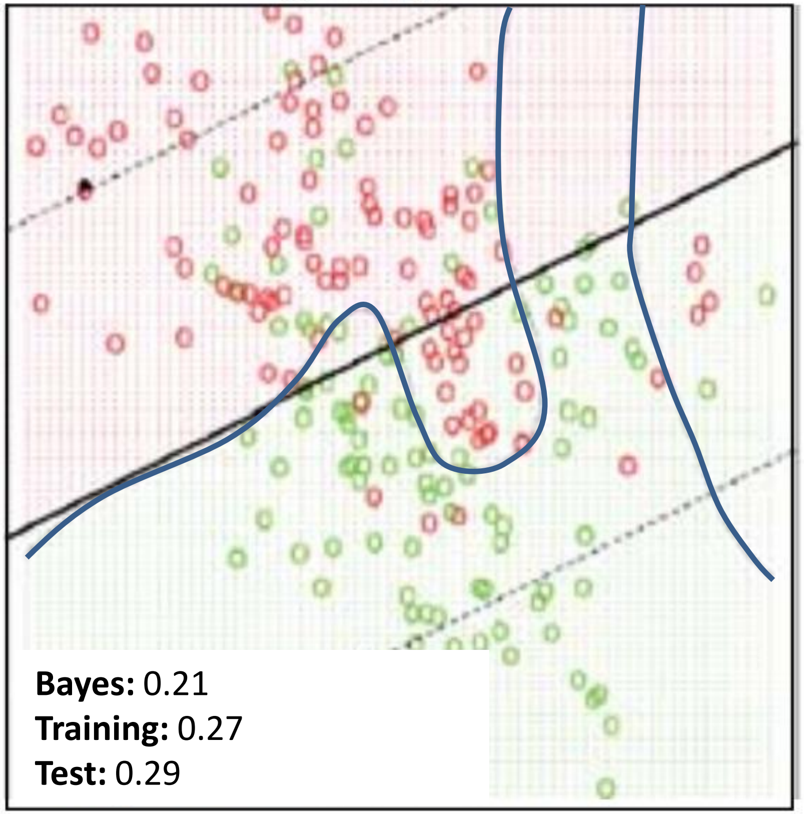
- **C large**: small margin, less training errors.
- **C small**: large margin, more training errors.

Examples

$C = 10^4$



$C = 10^{-2}$



Non-linear Classification

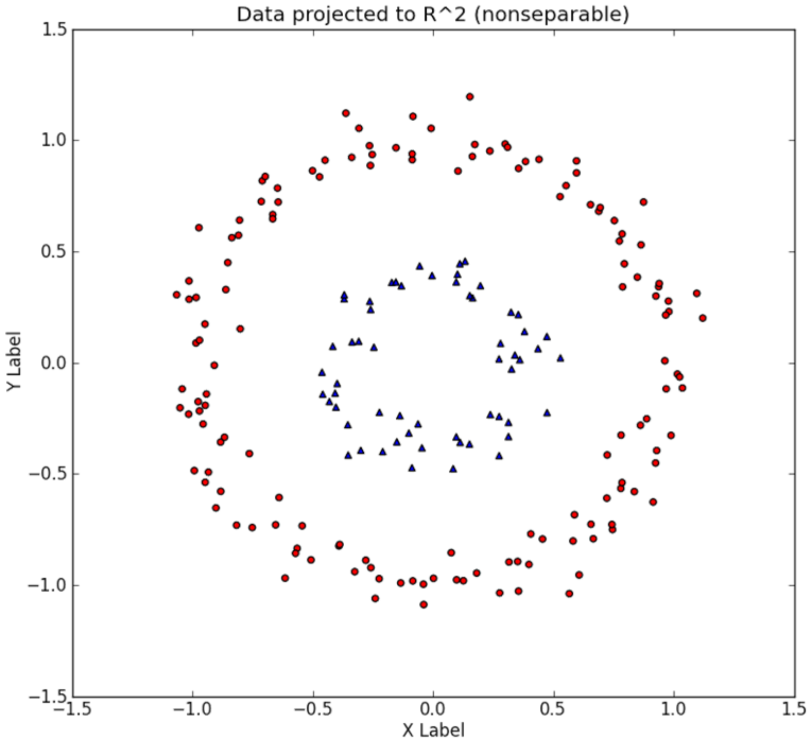
Linear classifiers have limited capabilities.

Can be implemented after a non-linear transformation,

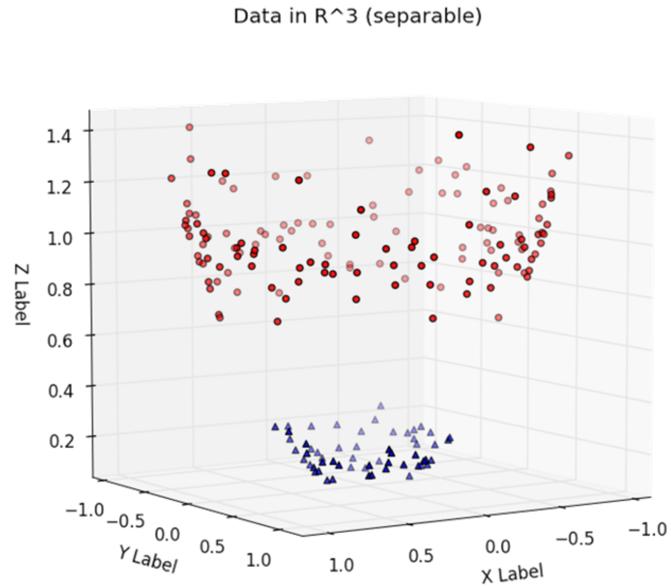
$$\boldsymbol{x} \rightarrow \phi(\boldsymbol{x}) = [\phi_1(\boldsymbol{x}), \phi_2(\boldsymbol{x}), \ldots]^T$$

where $\phi_i(x)$ are non-linear functions.

A linear classifier with respect to $\phi(x)$ is non-linear with respect to x.

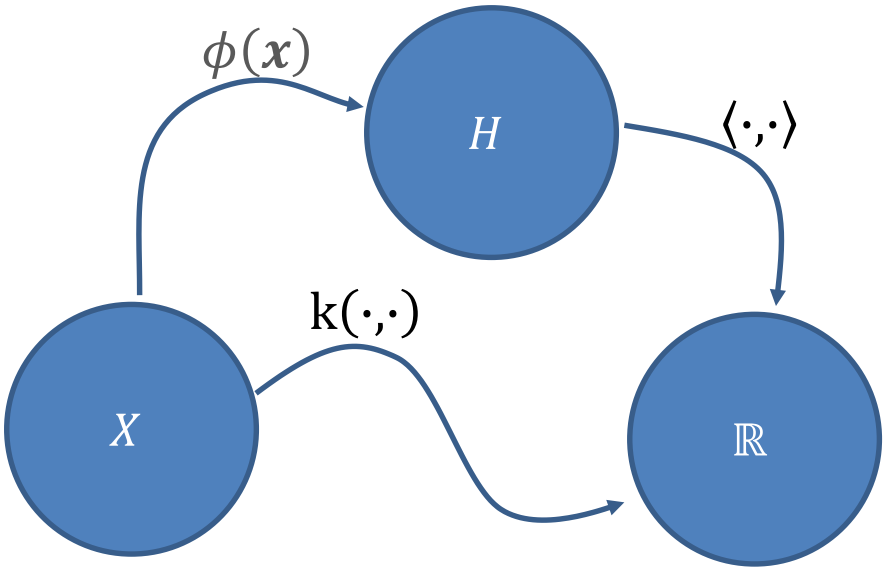


$z = x^2 + y^2$



$$d(\boldsymbol{x}, \boldsymbol{\alpha}^*, b^*) = sign\left(\sum_{sv} \boldsymbol{\alpha}_i^* y_i \langle \boldsymbol{x}, \boldsymbol{x}_i \rangle + b^*\right) = sign\left(\sum_{sv} \boldsymbol{\alpha}_i^* y_i \langle \phi(\boldsymbol{x}), \phi(\boldsymbol{x}_i) \rangle + b^*\right)$$

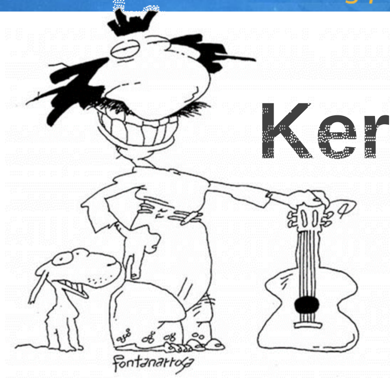
Kernel Trick



$$\forall \boldsymbol{x}, \boldsymbol{x}' \in \boldsymbol{X}$$

$$\mathrm{k}(x, x') = \langle \phi(\boldsymbol{x}), \phi(\boldsymbol{x}') \rangle_{\mathcal{H}}$$

- Almost no conditions on $\boldsymbol{X}$
- No need for dot product
- No nned to know $\phi(\boldsymbol{x})$.



Kernel Trick

$$d(\boldsymbol{x}, \boldsymbol{\alpha}^*, b^*) = sign\left(\sum_{sv} \boldsymbol{\alpha}_i^* y_i k(\boldsymbol{x}, \boldsymbol{x}_i) + b^*\right)$$

Projective kernels	
Monomial of degree q	$\langle \boldsymbol{x}, \boldsymbol{x}' \rangle^q$
Polynomial of degree q	$(1 + \langle \boldsymbol{x}, \boldsymbol{x}' \rangle)^q$
Sigmoidal	$\frac{1}{\eta_0} tanh(\beta_0 \langle \boldsymbol{x}, \boldsymbol{x}' \rangle - \alpha_0)$

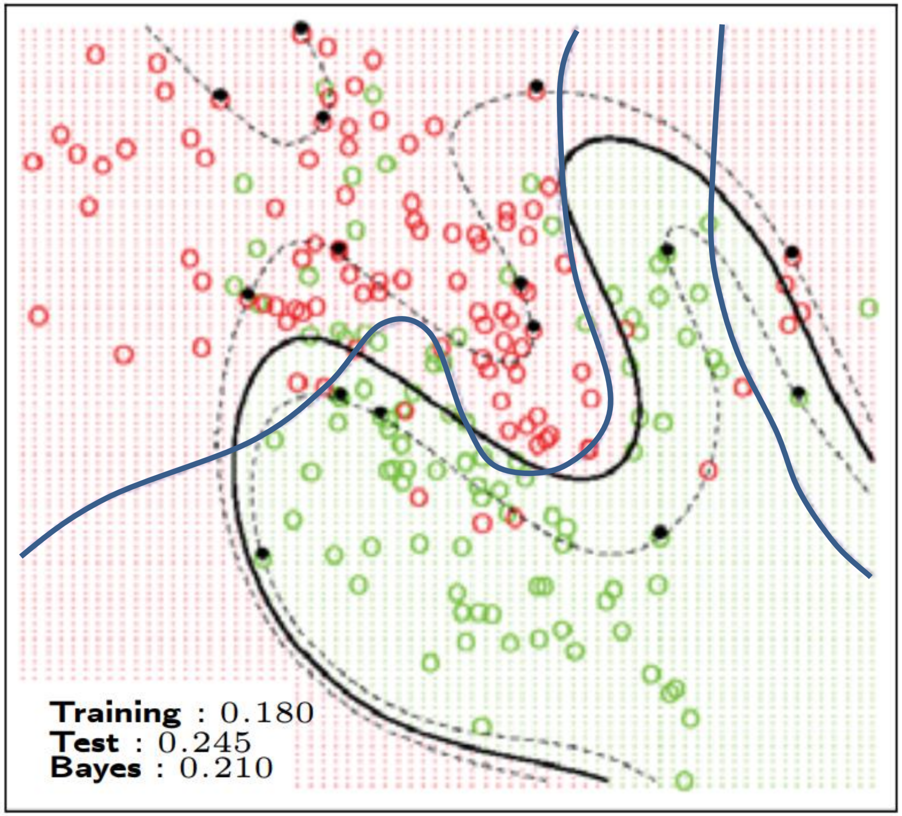
… and also:

- $k_1(x, x') + k_2(x, x')$
- $k_1(x, x').k_2(x, x')$
- …

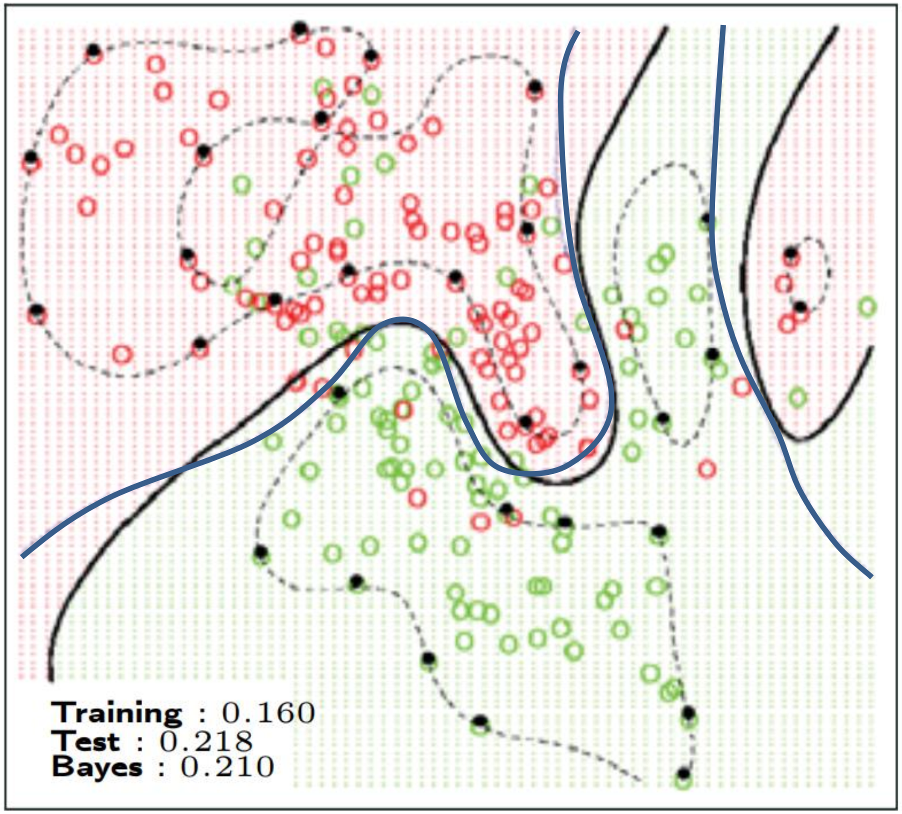
Radial kernels	
Gaussian	$exp\left(-\frac{1}{2\sigma_0^2}\lVert\boldsymbol{x}-\boldsymbol{x}'\rVert^2\right)$
Exponential	$exp\left(-\frac{1}{2\sigma_0^2}\lVert\boldsymbol{x}-\boldsymbol{x}'\rVert\right)$
Uniform	$\frac{1}{\eta_0}\mathbb{1}_{\lVert\boldsymbol{x}-\boldsymbol{x}'\rVert \le \beta_0}$
Epanechnikov	$\frac{1}{\eta_0}(\beta_0^2 - \lVert\boldsymbol{x} - \boldsymbol{x}'\rVert^2)\mathbb{1}_{\lVert\boldsymbol{x}-\boldsymbol{x}'\rVert \le \beta_0}$
Cauchy	$\frac{1}{\eta_0}\frac{1}{1+\lVert\boldsymbol{x}-\boldsymbol{x}'\rVert^2/\beta_0^2}$

Examples

polynome



gaussian kernel



Parameter Tunning

k-fold

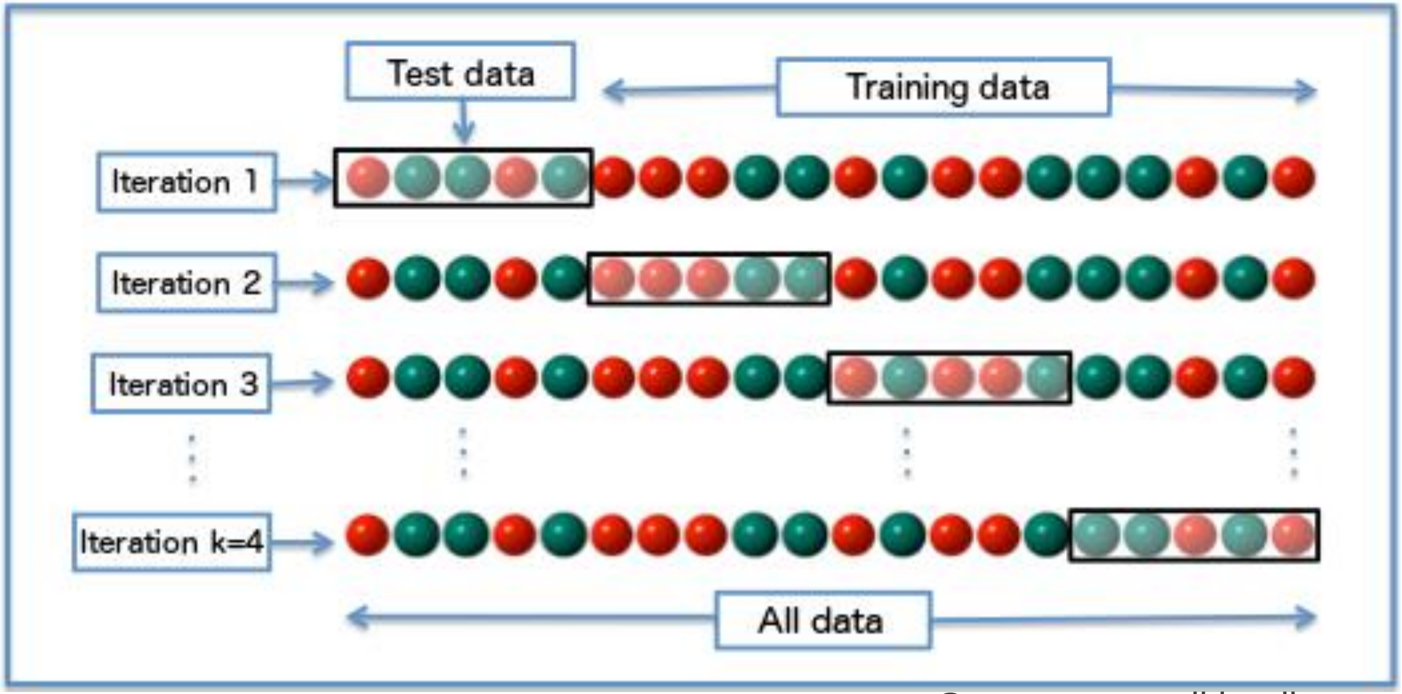
- Regular
- Random

LOOCV

- Leave-one-out

$$E = \frac{1}{k}\sum_{i=1}^{k} E_i$$

k-fold cross-validation



Source: www.wikipedia.org

Conclusions

Do NOT use ML as a black box

Focus on the concept of the method

Aware of the knowledge of the system

Correct selection of the method used

Every problem is unique

Choose wisely

ITUKALEIDOSCOPE
SANTA FE 2018

Thank you