LINEAR CLASSIFIERS

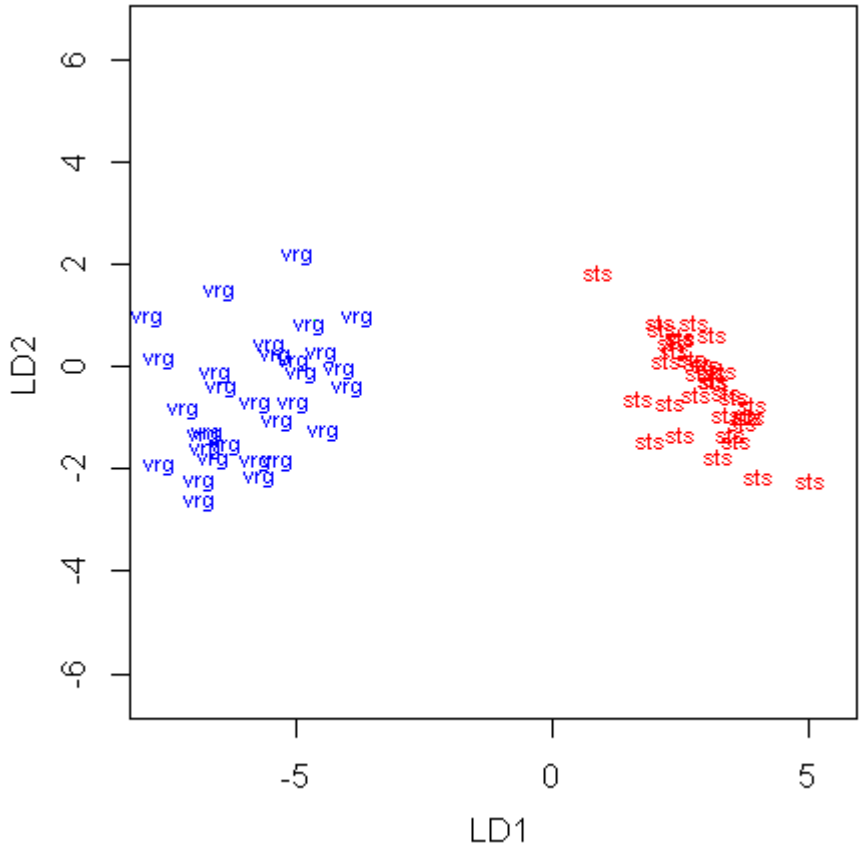
A. LINEAR DISCRIMINATION FUNCTIONS AND DECISION HYPERPLANES.

B. PERCEPTRON ALGORITHM.

D. LEAST SQUARE METHODS.

E. SUPPORT VECTOR MACHINES.

LINEAR DISCRIMINANT



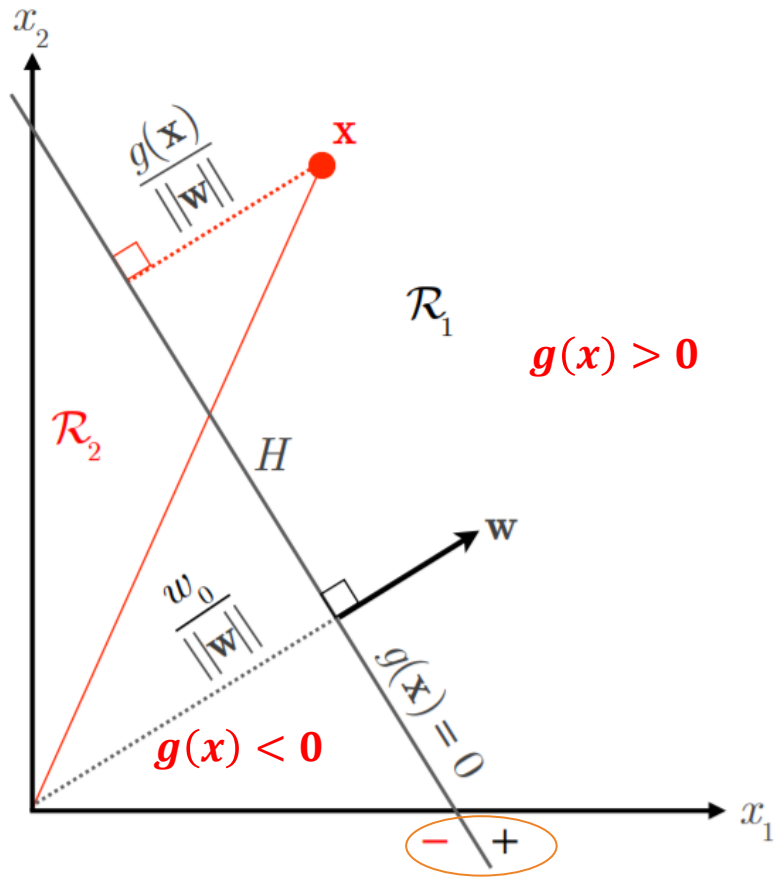
$$g(x) = w^T x + w_0$$

$\boldsymbol{w}$ = weight vector, orthogonal to the decision hyperplane

$\boldsymbol{x}$ = Feature vector

w_o = Threshold

LINEAR DISCRIMINANT



$w^T = [w_1, w_2]$
Orthogonal to the decision plane

$$g(x) = w^T x + w_0$$

➔ $|g(x)|$ is a measure of the Euclidean distance of the point x from the decision hyperplane

PERCEPTRON ALGORITHM

PERCEPTRON ALGORITHM

Problem: Define values for $w_i, i = 1,2, \dots, l$

Assumption: C1 and C2 are classes linearly separable.

→ There exists a hyperplane, $\boldsymbol{w^T x}$, such that:

$$w^T x > 0 \qquad x \in C1$$

$$w^T x \leq 0 \qquad x \in C2$$

For a $(l + 1)$ dimensional space

$$x' \equiv [x^T, 1]^T$$

$$w' \equiv [w, w_0]^T$$

$$w^T x + w_0 = w'^T x'$$

COST FUNCTION

$J(w) = \sum_{x \in Y}(\delta_x w^T x)$ (I)

Y Subset of misclassified training vectors.

δ_x -1 if $x \in C1$

1 if $x \in C2$

→ $J(w)$ always > 0

0 iff Y={0}

GRADIENT DESCENT METHOD

$$\theta_{new} = \theta_{old} + \Delta\theta$$

$$w(t+1) = w(t) - \rho_t \left.\frac{\partial J(w)}{\partial w}\right|_{w=w(t)} \qquad (2)$$

$w(t)$ vector w at the *th* iteration

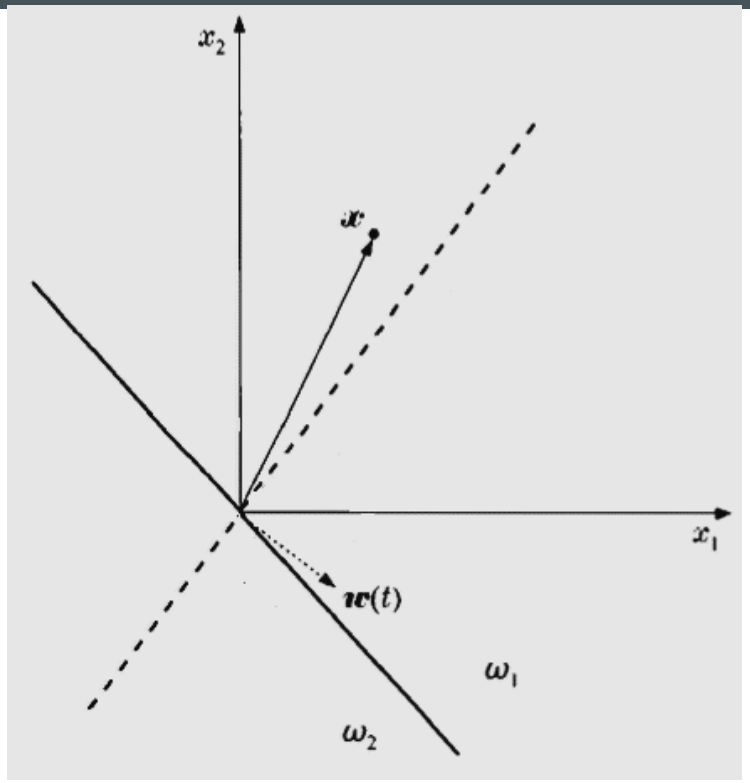
ρ_t Sequence of real numbers (e.g. cte/t)

from (1) and for the values that is defined:

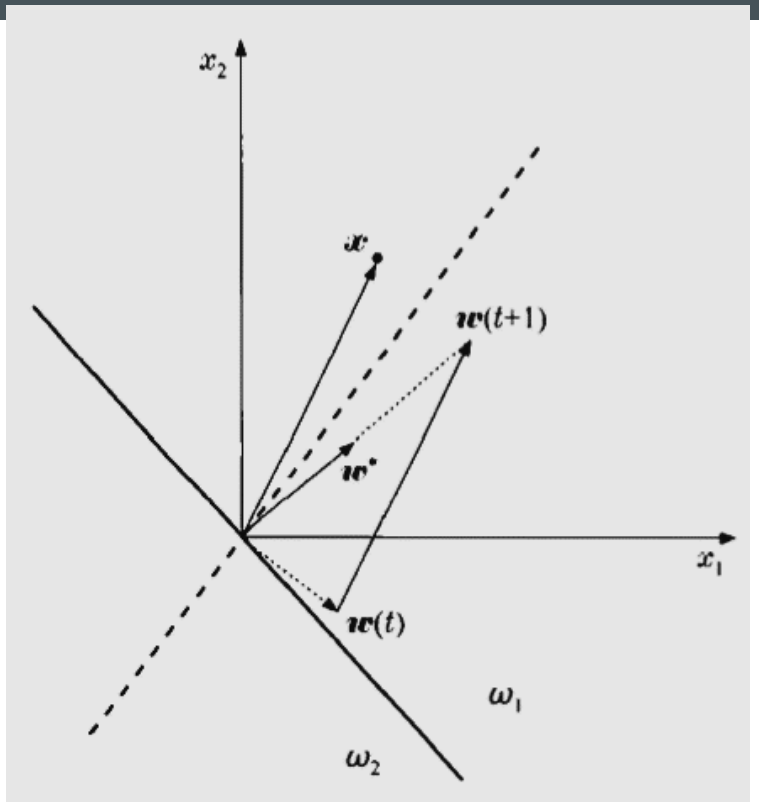
$$\frac{\partial J(w)}{\partial w} = \sum_{x \in Y} (\delta_x x) \qquad (3)$$

→ $w(t+1) = w(t) - \rho_t \sum_{x \in Y} (\delta_x x)$

GRADIENT DESCENT METHOD



First iteration



Second iteration

PSEUDO ALGORITHM

Define: $\boldsymbol{w(0)}$, $\boldsymbol{\rho_0}$

Repeat:

$Y = 0$

$i = 1{:}N$

if $\delta_i w^T x_i > 0$ → $Y = Y \cup \{x_i\}$

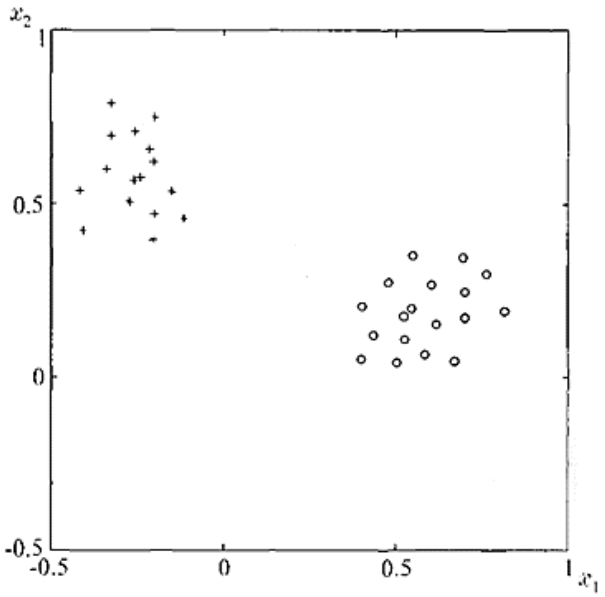
$$w(t+1) = w(t) - \rho_0 \sum_{x \in Y} (\delta_x x)$$

adjust ρ_t

t=t+1

Until Y={0}

EXAMPLE



The dashed line given by

$$x_1 + x_2 - 0.5 = 0$$

Corresponding to the weight vector *w*, which has been computed from the latest iteration step of the perceptron algorithm with ρ=0.7.

A) Draw this plane and indicate what Class 1 is and which class 2.

The line classifies correctly all the vectors except:

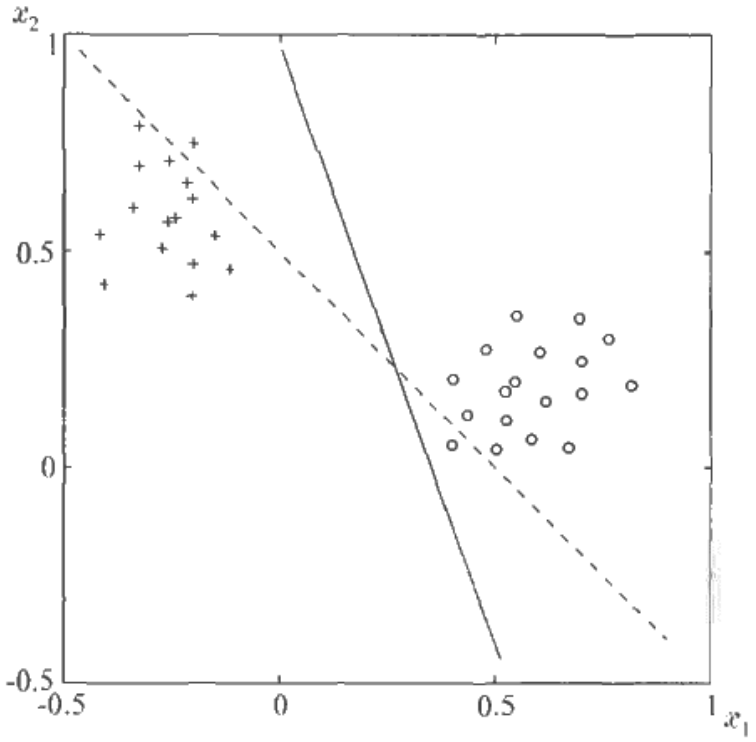
$[0.4, 0.05]^T$

$[-0.2, 0.75]^T$

B) Compute the resulting new line.

C) Draw the new line and indicate if all vectors are correctly classified.

EXAMPLE



EXERCISE TO PRACTICE

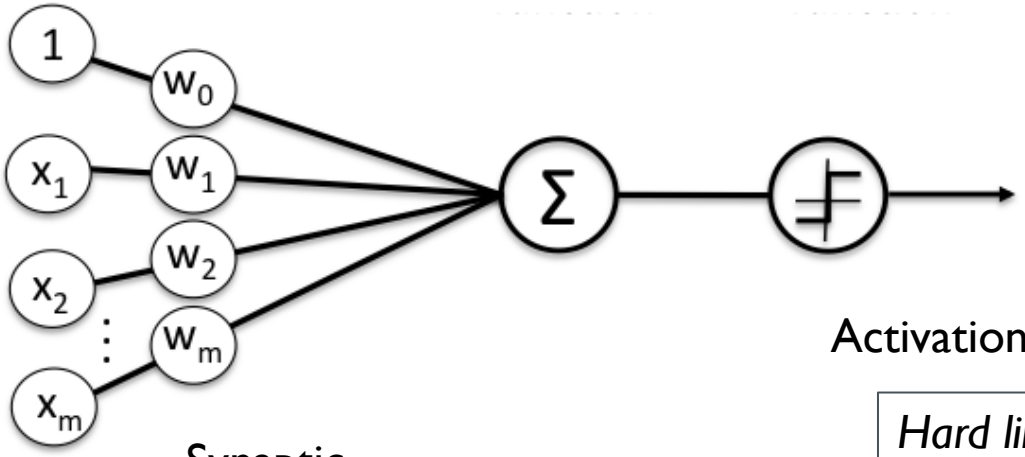
Using the perceptron algorithm, get the hyperplane for:

* AND function

* OR function

PERCEPTRON/ NEURON

Inputs



Synaptic weights

Activation function

Hard limit
-1 if $w^T x$<0
1 if $w^T x$>0

PERCEPTRON ALGORITHM CONVERGENCE

- If the data is linearly separable, the algorithm will converge into a finite number of iterations.
- There are practical limitations:
  - Convergence can be slow
  - If the data is not linearly separable, the algorithm will never converge
  - The solution is not unique,

LEAST SQUARES METHOD
MEAN SQUARE ERROR ESTIMATION

Given a vector x, the output of the classifier will be $w^T x$

$y(x) = \pm 1$ denotes the desired output

Compute the weight vector *w* so that the mean square error is minimized.

$$J(w) = E[|y - x^T w|^2] \qquad (1)$$

$$\hat{w} = argmin\ J(w) \qquad (2)$$

$$\frac{\partial J(w)}{\partial w} = 2E[x(y - x^T w)] = 0 \qquad (3)$$

$$\hat{w} = R_x^{-1} E(xy)$$

MULTICLASS GENERALIZATION

$$g_i(x) = w_i^T x$$

$$y_i = 1 \quad if \quad x \in C_1$$
$$y_i = 0 \quad if \quad x \in C_2$$

Define:

$\boldsymbol{y}^T = [y_1, y_2, \dots, y_M]$ For a vector $\boldsymbol{x}$ and a matrix $W = [w_1, w_2, \dots, w_M]$

$$\widehat{W} = \arg minE[|\boldsymbol{y} - W^T x|^2]$$

$$\widehat{W} = \arg minE\left[\sum_{i=1}^{M} (y_i - w_i^T x)^2\right]$$

STOCHASTIC APPROXIMATION AND THE LMS ALGORITHM

$$E[F(x_k, w)]=0 \qquad (1)$$

$$\hat{w}(k) = \hat{w}(k-1) + \rho_k F\left(x_k, \hat{w}(k-1)\right) \qquad (2)$$

$$\sum_{k=1}^{\infty} \rho_k \to \infty, \qquad \sum_{k=1}^{\infty} {\rho_k}^2 < \infty \quad \rightarrow \quad \rho_k \to 0$$

$$\lim_{k\to\infty} \text{prob}\{\hat{w}(k) = w\} = 1$$

$$\boldsymbol{\lim_{k\to\infty} E[||\hat{w}(k) - w||^2] = 0}$$

WIDROW HOFF ALGORITHM

Considering $E[x_k - w] = 0, \quad \rho_k = 1/k$, eq (2) becomes:

$$\hat{w}(k) = \hat{w}(k-1) + \frac{1}{k}[x_k - \hat{w}(k-1)]$$
$$= \frac{(k-1)}{k}\hat{w}(k-1) + \frac{1}{k}x_k$$

For large values of k

$$\hat{w}(k) = \frac{1}{k}\sum_{r=1}^{k} x_r$$

* Adaline Neurone (adaptive linear element)

Substituting the previous value in (2)

$$\hat{w}(k) = \hat{w}(k-1) + \rho_k x_k (y_k - x_k^T \hat{w}(k-1))$$

y_k and x_k are the desired output(± 1) and input pairs

SUM OF ERROR SQUARES ESTIMATION

$$J(w) = \sum_{i=1}^{N} (y_i - x_i^T w)^2 \equiv \sum_{i=1}^{N} e_i^2$$

Minimizing

$$\sum_{i=1}^{N} x_i (y_i - x_i^T \hat{w}) = 0 \Rightarrow \left(\sum_{i=1}^{N} x_i x_i^T\right)\hat{w} = \sum_{i}^{N} (x_i y_i)$$

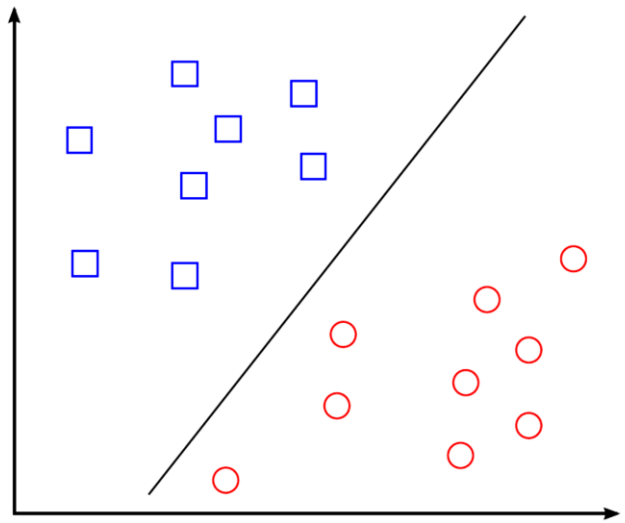
$$X = \begin{bmatrix} x_1^T \\ x_2^T \\ x_N^T \end{bmatrix} \qquad Y = \begin{bmatrix} y_1 \\ y_2 \\ y_N \end{bmatrix}$$

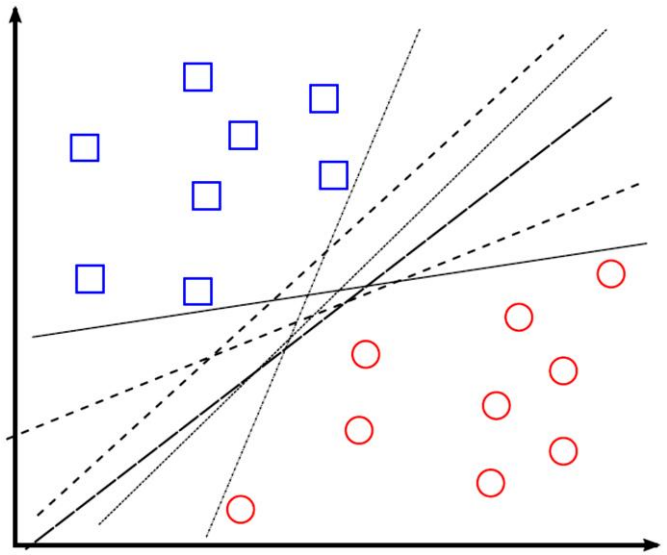
$$\left(X^T X\right)\hat{w} = X^T Y \Rightarrow \hat{w} = (X^T X)^{-1} X^T Y$$

SUPPORT VECTOR MACHINES (SVM)

$x_i, \quad i = 1, 2, 3, \ldots, N \quad y_i \in \{+1, -1\}$

$$g(x) = w^T x + w_0 = 0$$



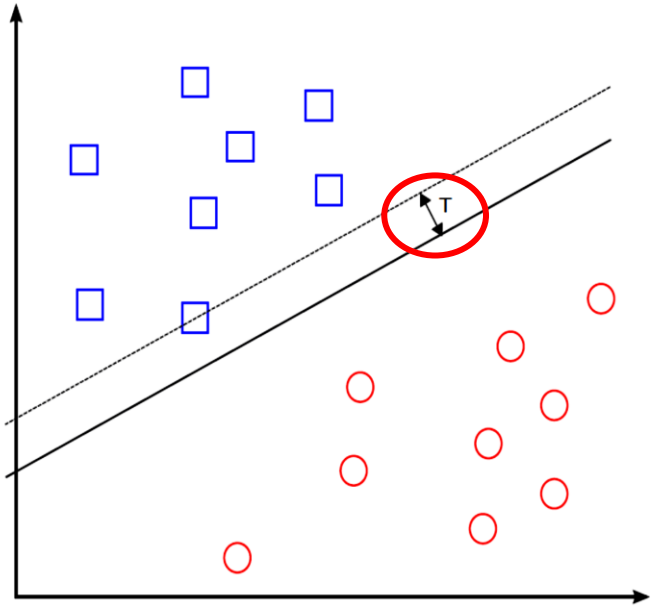


Distance between the hyperplane ***D(x)*** and ***x***

$$\frac{|D(x')|}{||w||}$$

$\forall \boldsymbol{x_i}$ it must be fulfilled that:

$$\frac{y_i D(x_i)}{||w||} \geq \tau$$

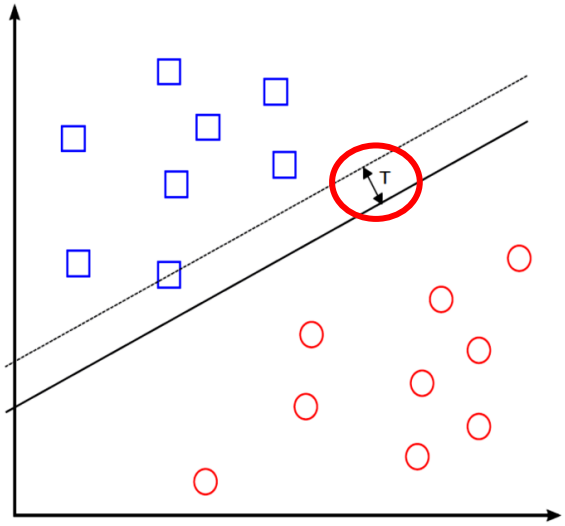


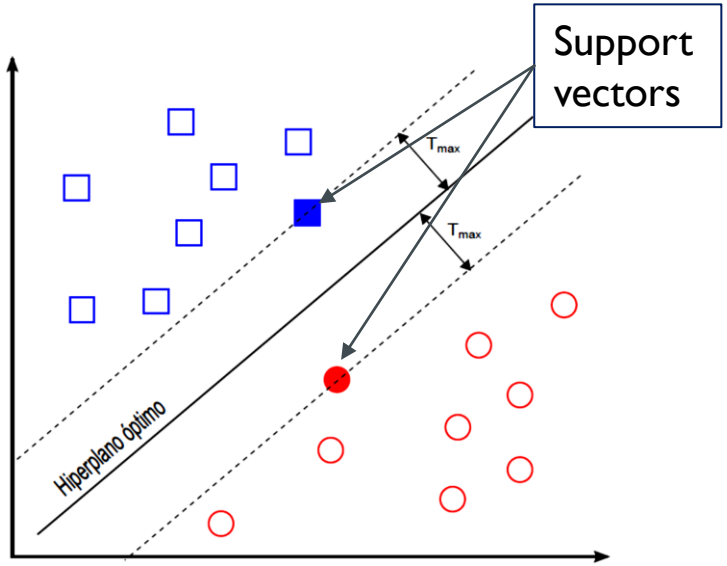
Distance between the hyperplane ***D(x)*** and ***x***

$$\frac{|D(x')|}{||w||}$$

$\forall \boldsymbol{x_i}$ it must be fulfilled that:

$$\frac{y_i D(x_i)}{||w||} \geq \tau$$





The above equation can be expressed as:

$$y_i D(x_I) \geq \tau||w|| \qquad \tau||w|| = 1$$

$$y_i D(x_i) \geq 1$$

LINEARLY SEPARABLE CLASSES

$$\min \quad f(w) = \frac{1}{2}||w||^2$$

Subject to $y_i(\boldsymbol{w}^T \boldsymbol{x}_i + w_0) \geq 1$
i=1, 2, …, N

$$\boldsymbol{w} = \sum_{i=1}^{N_s} \lambda_i y_i \boldsymbol{x_i}$$

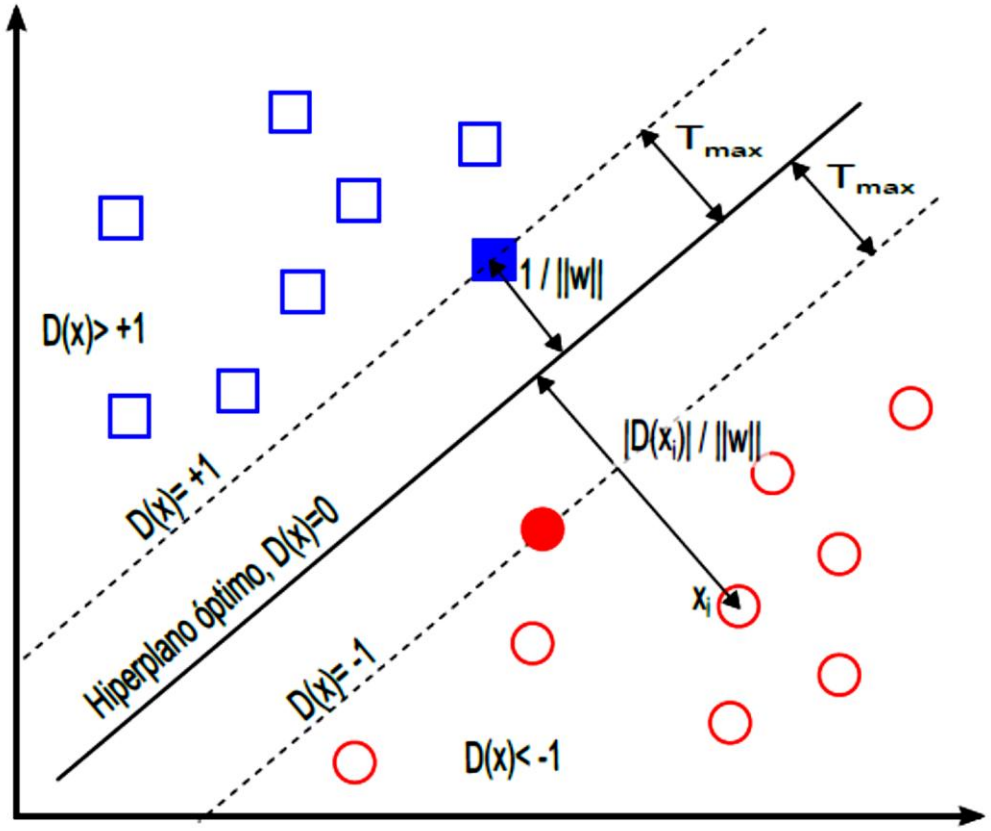
$$N_s < N$$

$$\boldsymbol{w} = \sum_{i=1}^{N} \lambda_i y_i \boldsymbol{x_i}$$

$$\boldsymbol{w^T x} + w_0 = \pm 1$$

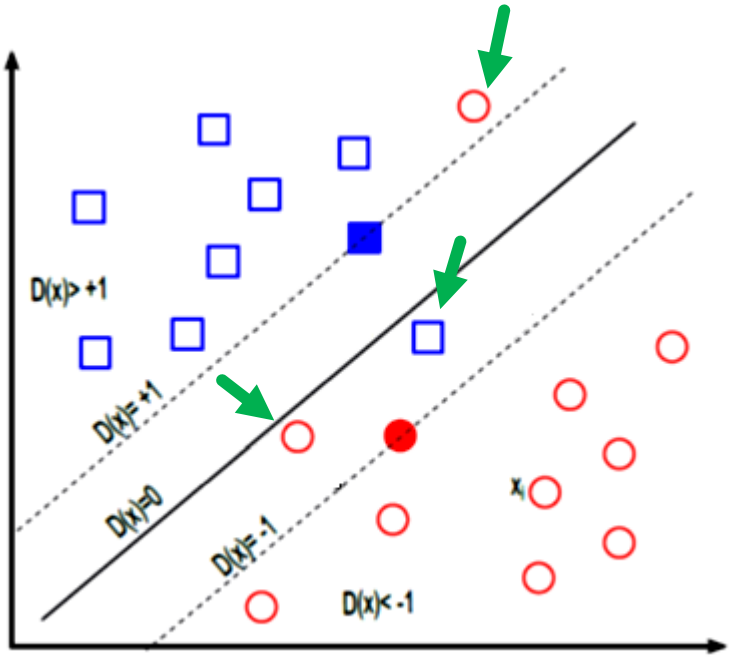
$$\lambda_i \geq 0, \qquad i = 1, 2, \ldots, N$$
$$\lambda_i[y_i(\boldsymbol{w}^T \boldsymbol{x_i} + w_0) - 1] = 0$$

LINEARLY SEPARABLE CLASSES



Only support vectors will have $\lambda>0$

QUASI-LINEARLY SEPARABLE CLASSES



When classes are quasi-linearly separable:

- The point falls within the margin but on the right side.

$$0 \leq y_i(\boldsymbol{w}^T \boldsymbol{x} + w_0) < 1$$

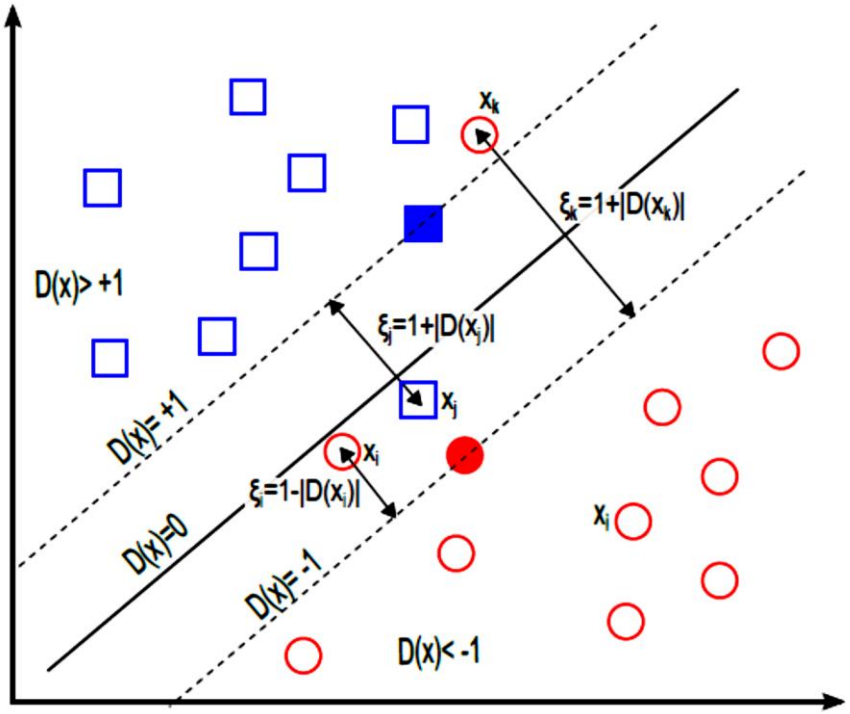
- The point falls from across the border.

$$y_i(\boldsymbol{w}^T \boldsymbol{x} + w_0) < 0$$

ξ represents the deviation of the case linearly separable, the following conditions may occur:

$\xi = 0$	Separable variables
$0 < \xi \leq 1$	Non-separable variables
$\xi > 1$	Non-separable and poorly classified variables

$$y_i([\boldsymbol{w}^T \boldsymbol{x} + w_0]) \geq 1 - \xi i$$



$$J(\boldsymbol{w}, w_o, \xi) = \frac{1}{2}||w||^2 + C\sum_{i=!}^{N} \xi_i$$

Subject to:

$$y_i([\boldsymbol{w}^T \boldsymbol{x_i} + w_0]) \geq 1 - \xi_i$$

$$\max_{\lambda}\left(\sum_{i=1}^{N}\lambda i - 1/2\sum_{i,j}\lambda_i\lambda_j y_i y_j \boldsymbol{x}_i^T \boldsymbol{x}_j\right)$$

Subject to:

$$0 \leq \lambda_I \leq C\ ,$$

$$\sum_{i=1}^{N}\lambda_i y_i = 0$$

EXERCISE.

- Consider a two-class database with the following features:
  - The data follow a normal distribution
  - The means for each of the classes are: $\mu_1^T = [2, 2], \mu_2^T = [0,0]$
  - Equal variance for both classes $\sigma_1^2 = \sigma_2^2 = 2$
- Using the perceptron algorithm, least squares and SVM, compare the performance of the three configurations for the following cases:
  - 50% Training data, 50% Test data
  - 30% Training data, 50% Test data
  - 10% Training data, 50% Test data