Machine Learning

Support Vector Machines
SVM

Lesson 6

Data Classification problem

Training set: $D = \{(x_i, y_i), \ldots, (x_N, y_N)\}$

- x_i : ***input data sample***
- $y_i \in \{1, \ldots, K\}$: ***class or label*** of input

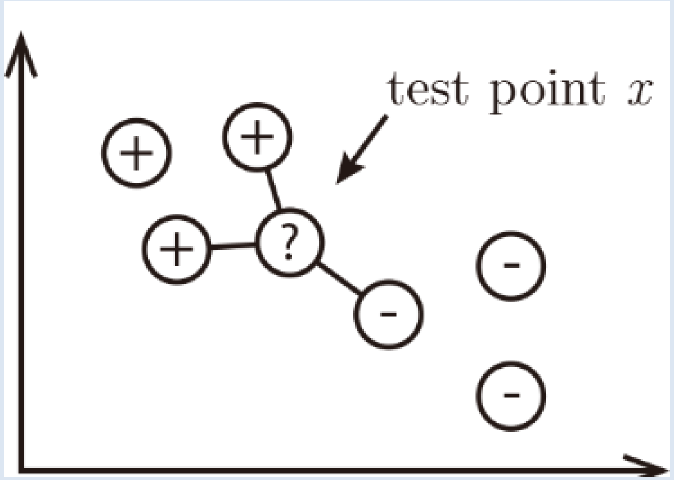
- **Target: Construct function** $f : X \to Y$

$$f(x_i) = y_i \quad \forall \ (x_i, y_i) \in D$$

- **Prediction** of class for any unknown input

$$y^* = f(x^*)$$

Nearest Neighbor classifier



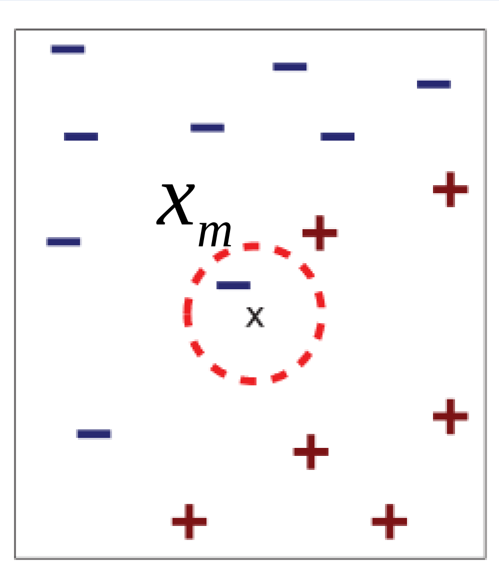
- **The simplest classification method**
- **Assumption:** data belongs to the same category are neighbors
- **Classification rule:** Classify according to the neighbor(s)

Nearest Neighbor Classifier

Classification

- Find the nearest neighbor (according to a **distance function**)

$$x_m : \min_{n=1,\ldots,N} \{dist(x^*, x_n)\}$$

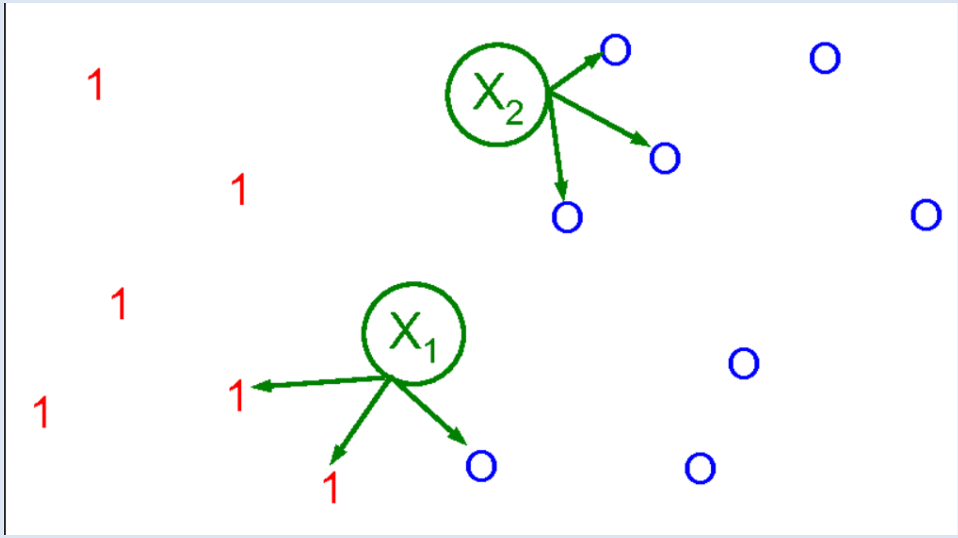


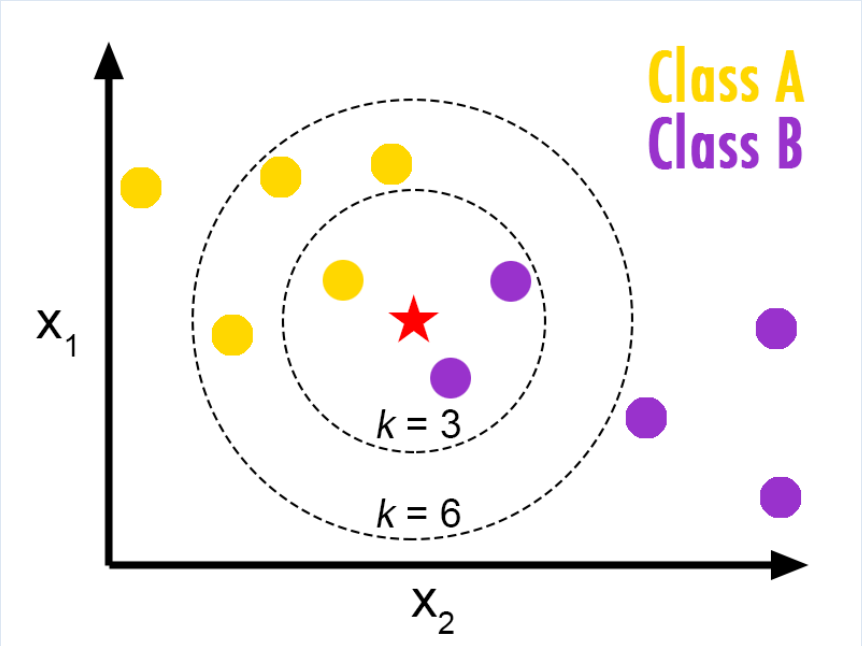
- Class of unknown x^* is similar to its neighbor

$$y^* = y_m$$

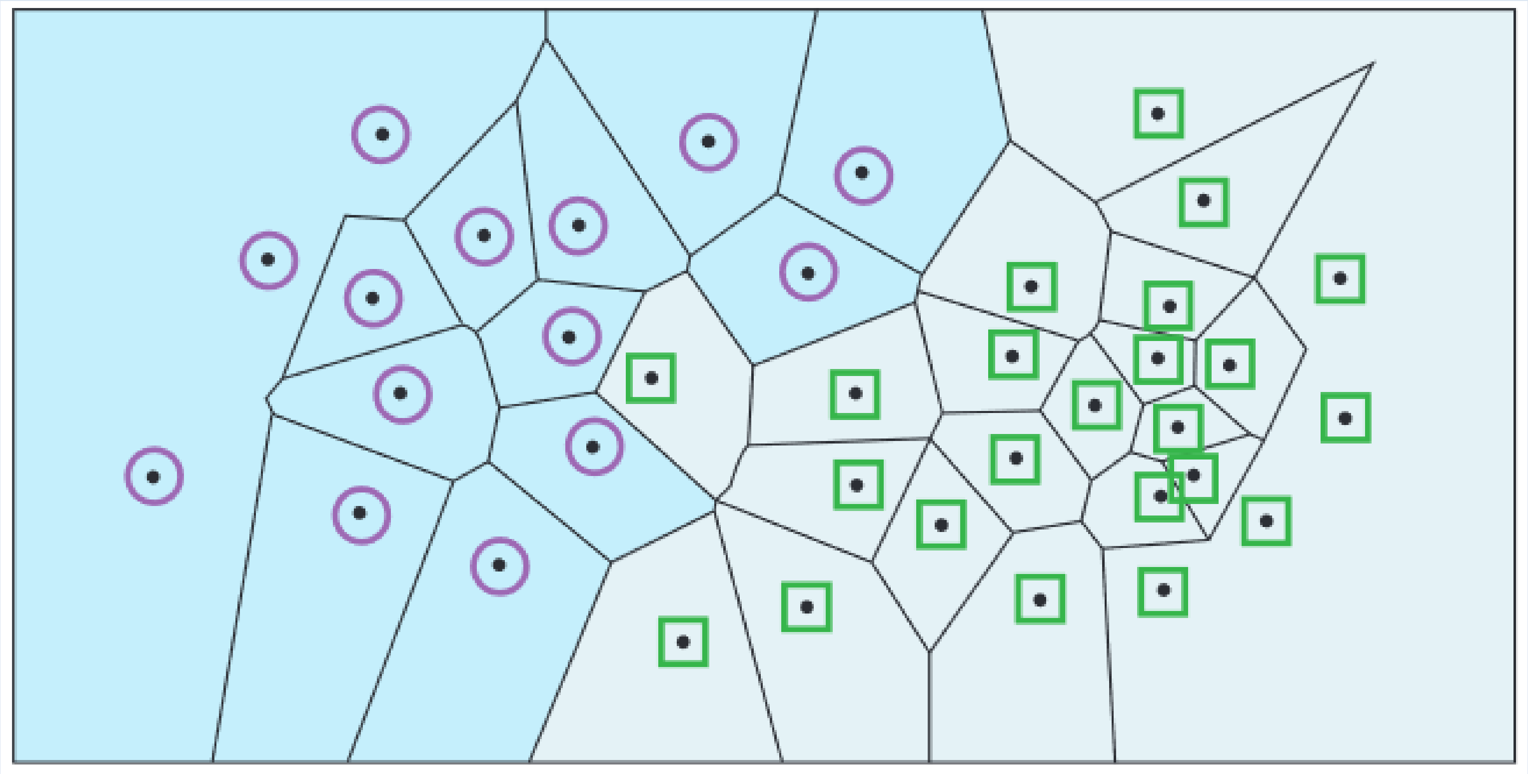
Extension to *k*-NN

- Find k>1 neighbors
- Classify according to the **class majority**





- Voronoi diagram

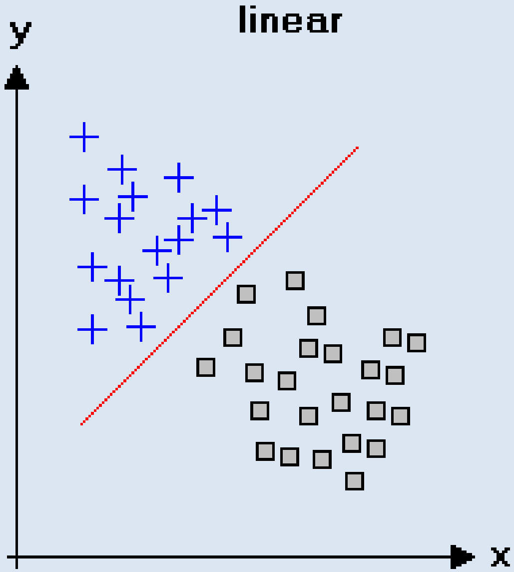


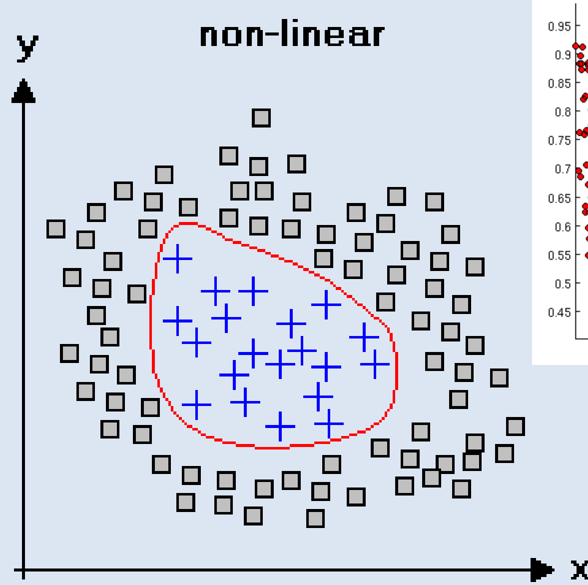
Linear Classifiers

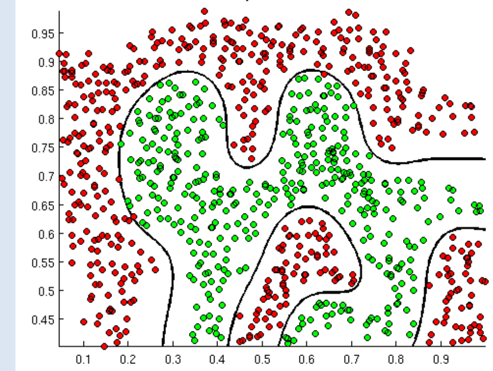
- ***K=2*** classes Ω_1 , Ω_2
- **Target:** Construction of a hyperplane $f(x,w)$ between data of 2 classes
- Decision boundaries:

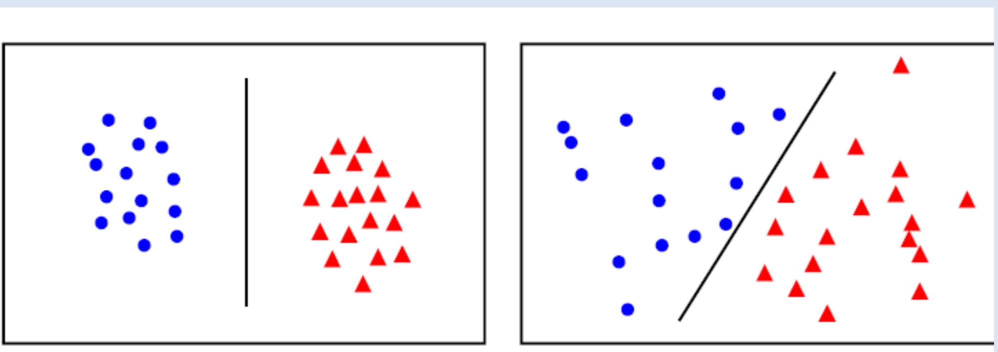
$$\begin{aligned}&if \quad f(x,w) \geq 0 \quad then \quad x \in \Omega_1\\&else\\&if \quad f(x,w) < 0 \quad then \quad x \in \Omega_2\end{aligned}$$

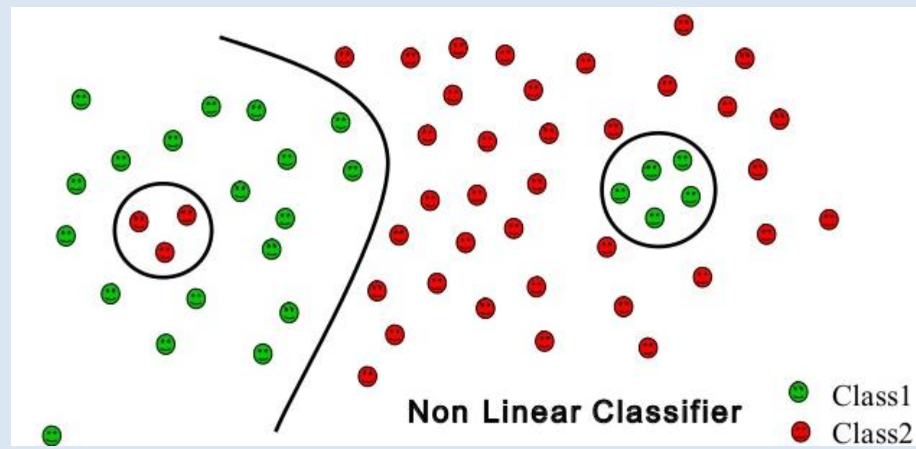
- ***w*** are the unknown parameters











linear classification ***nonlinear classification***

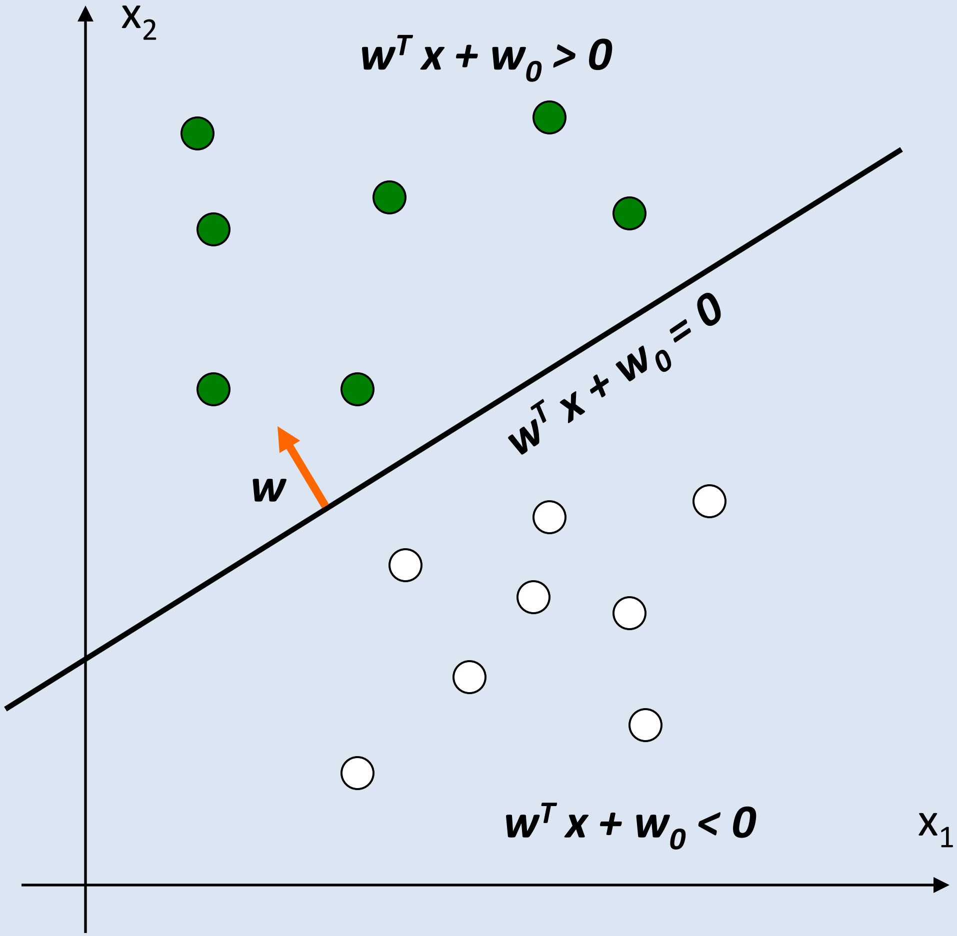
- **Training Set**

$$D = \{(x_i, y_i)\}_{i=1}^{N}$$

- ***f(x)* linear function:**

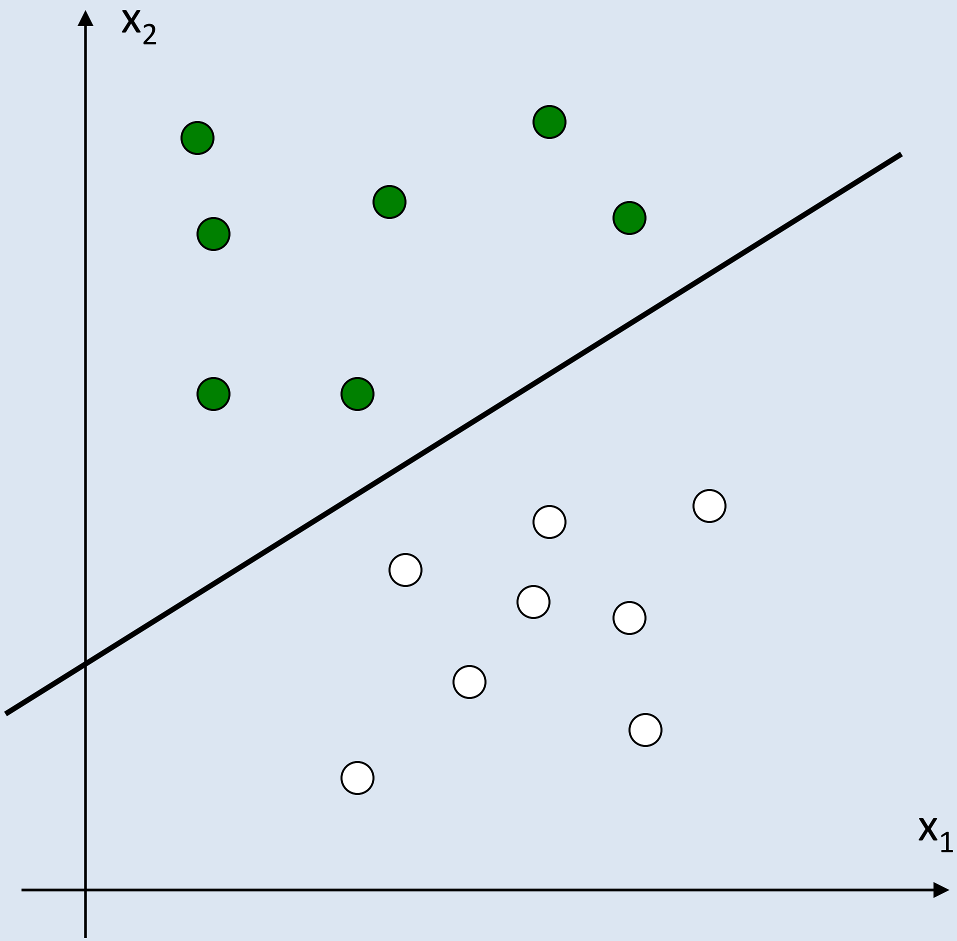
$$f(x) = w^T x + w_0$$

- Define a separating hyperplane between two classes



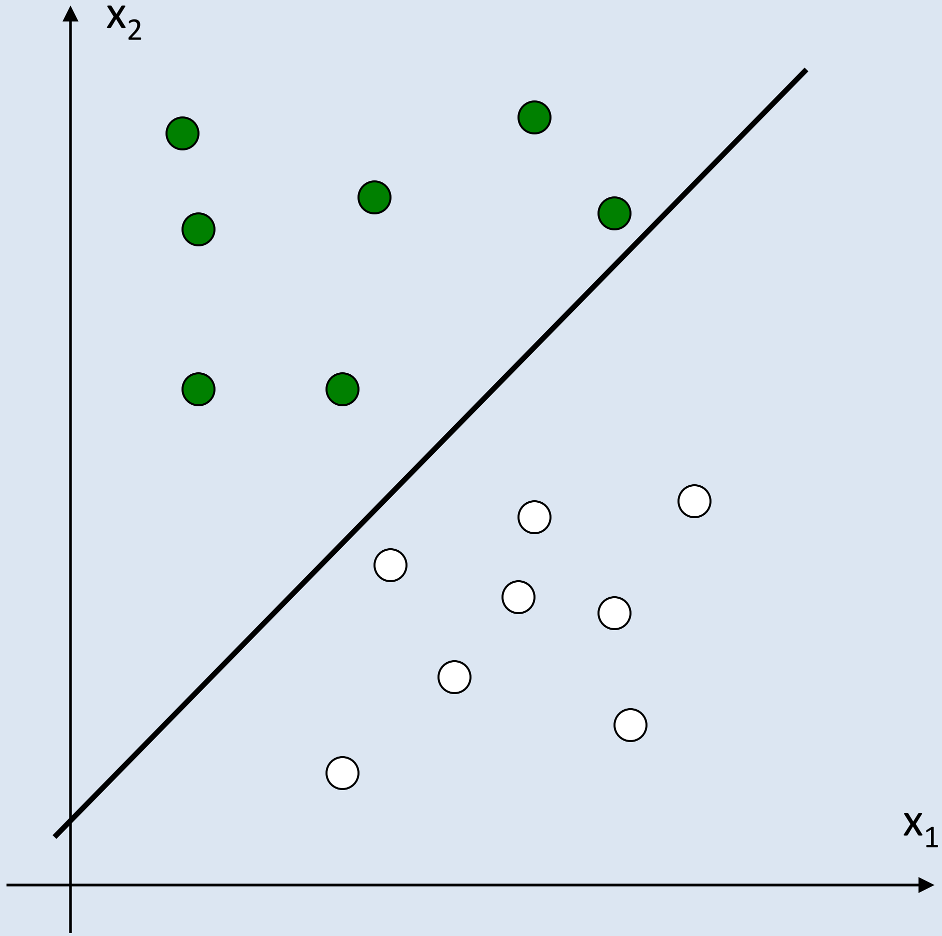
- **Question:**

  Which is the **optimum hyperplane** that separates better two classes?



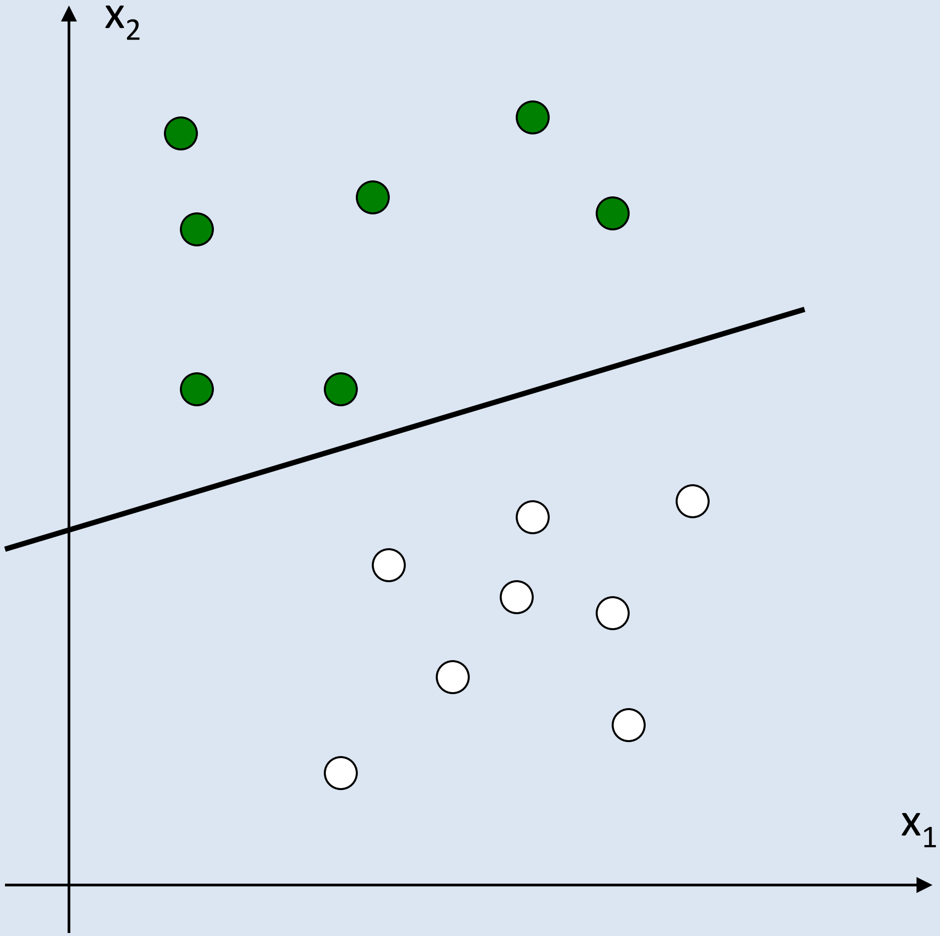
- **Question:**

  Which is the **optimum hyperplane** that separates better two classes?



- **Question:**

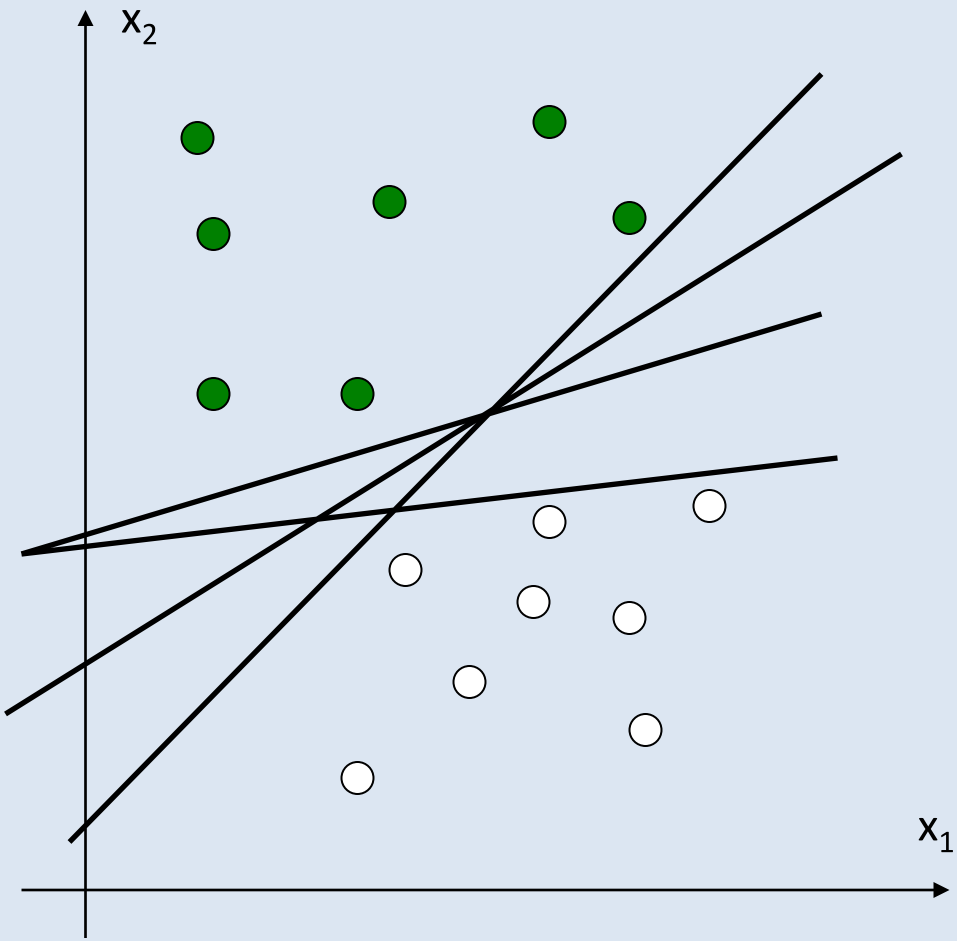
  Which is the **optimum hyperplane** that separates better two classes?



- **Question:**

  Which is the **optimum hyperplane** that separates better two classes?

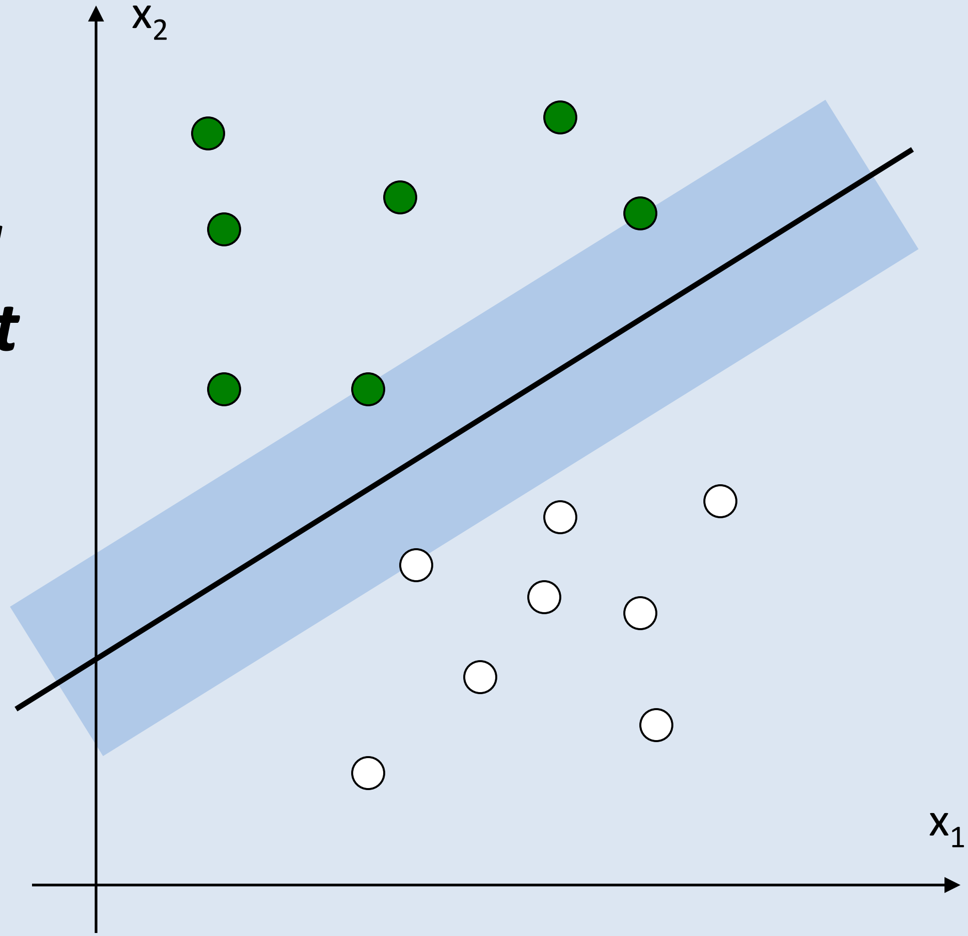
➢ **Infinite number of solutions!**



Solution: Marginal Maximization

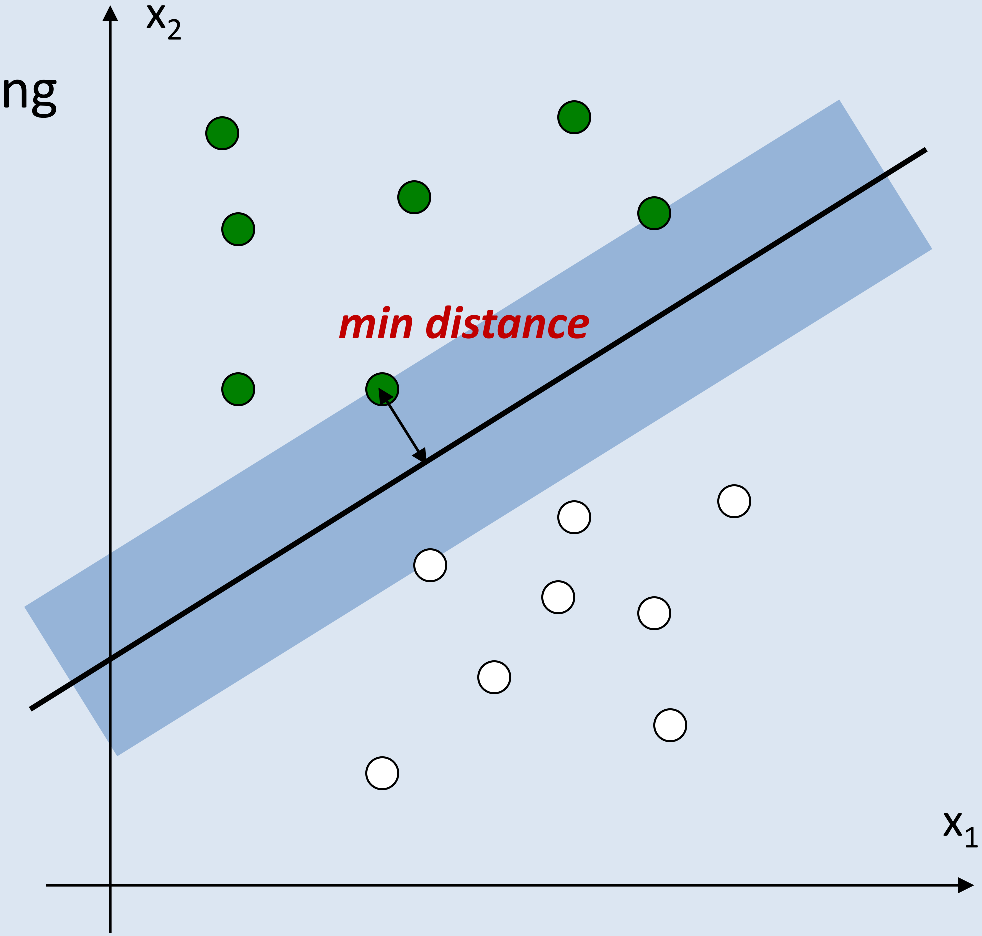
[Boser, Guyon, Vapnik '92],
[Cortes & Vapnik '95]

- ***The optimal separating hyperplane is the one that gives the maximum margin width***



Marginal Maximization

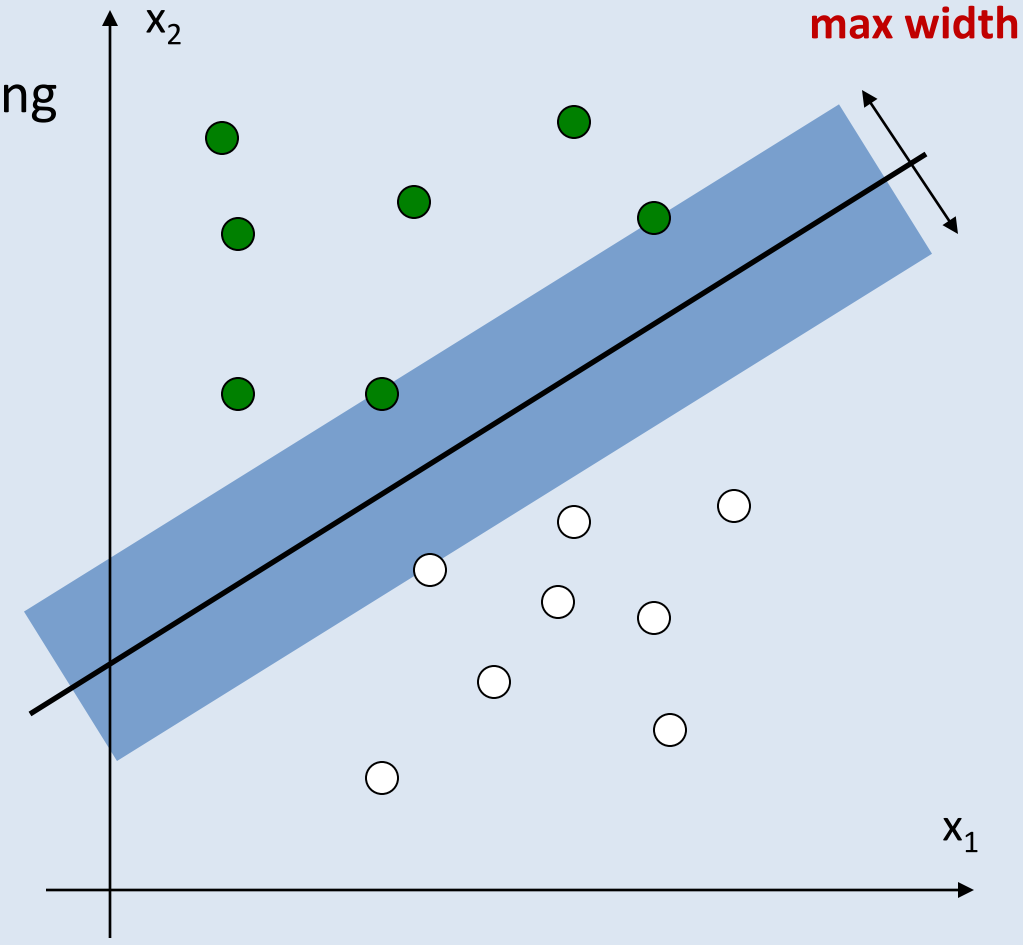
- **Definition 1**: **Margin** is the minimum distance of N training samples to the hyperplane



Marginal Maximization

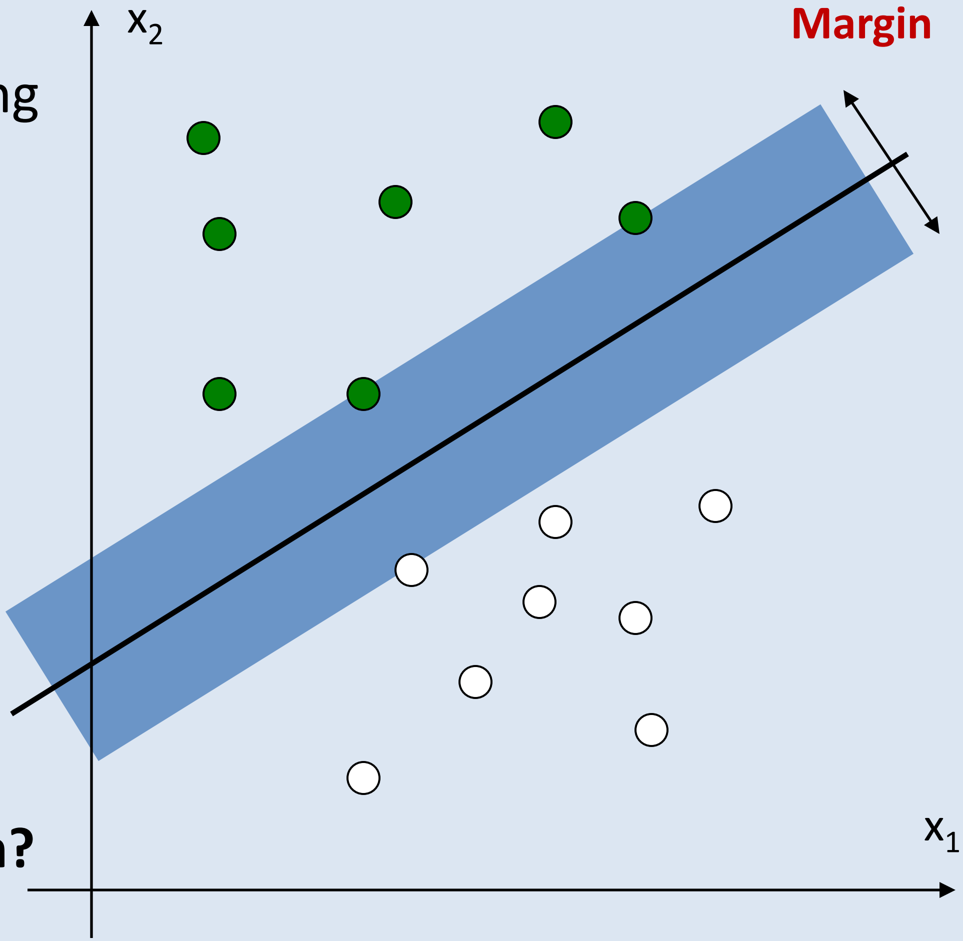
- **Definition 1**: **Margin** is the minimum distance of N training samples to the hyperplane

- **Definition 2**: **Margin** is the maximum width of boundary around the separating hyperplane without covering any sample



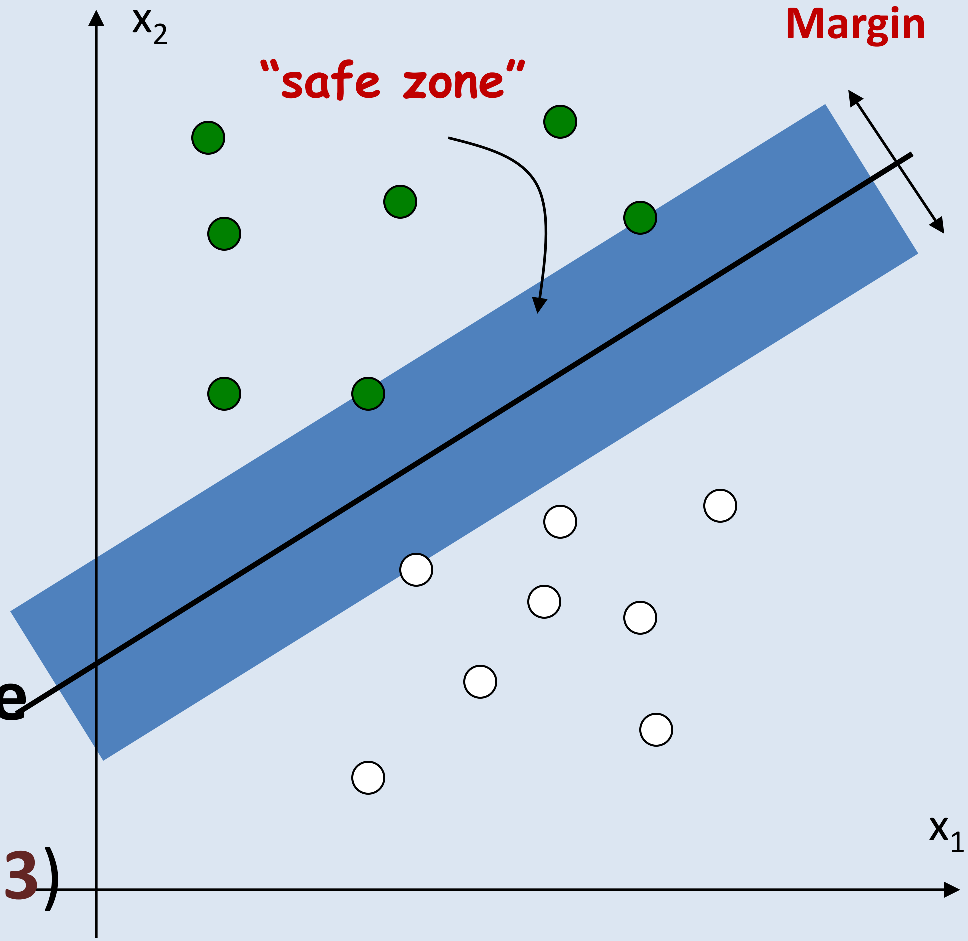
Marginal Maximization

- **Definition 1**: **Margin** is the minimum distance of N training samples to the hyperplane
- **Definition 2**: **Margin** is the maximum width of boundary around the separating hyperplane without covering any sample
- **Why is the optimum solution?**



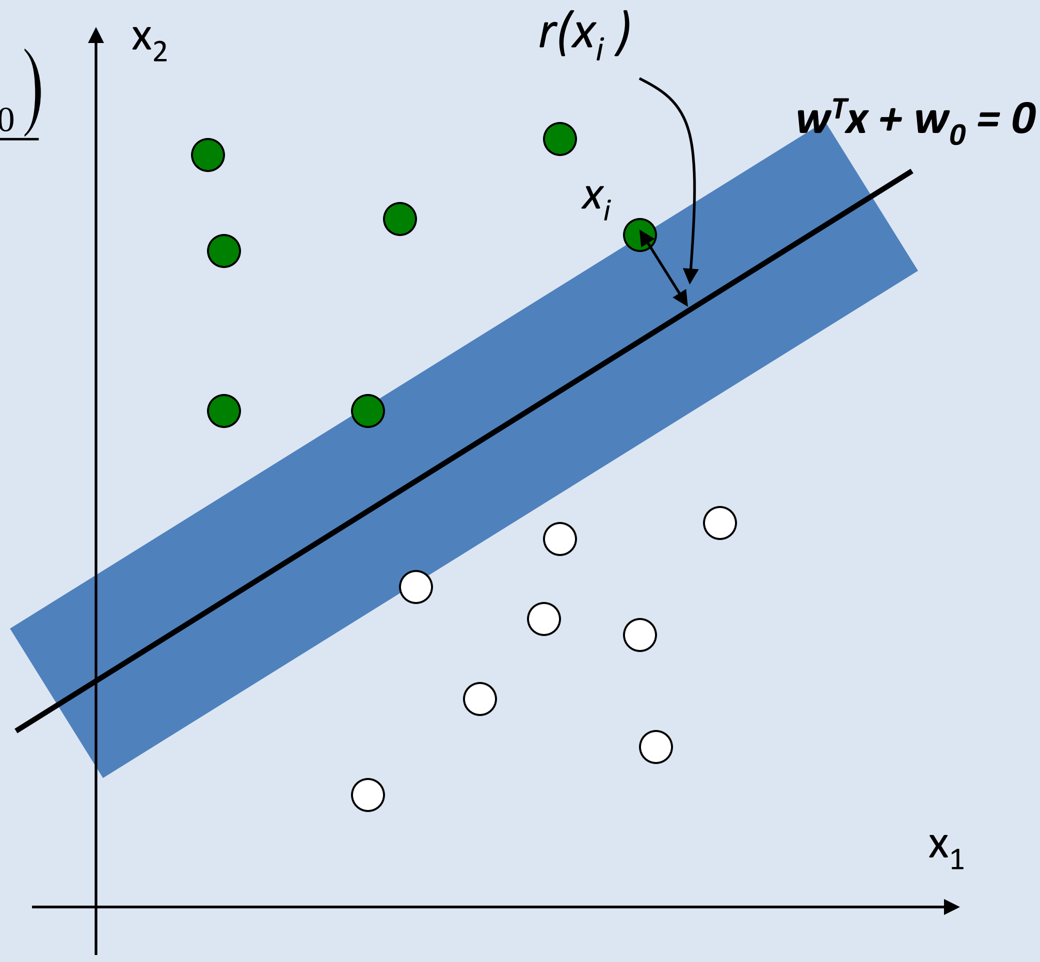
Marginal Maximization

- ✓ **Solution:** Find the hyperplane that **maximizes the margin** between two classes.
- ✓ This will **minimize the risk** of classifier's decision.
- ✓ Also, it will **increase the generalization** of classifier (**Vapnick, 1963**)



- **Distance of any point x_i**

$$r(x_i) = \frac{\left|w^T x_i + w_0\right|}{\|w\|} = \frac{y_i\left(w^T x_i + w_0\right)}{\|w\|}$$



- **Margin:**

$$\text{margin} = \min_{x_i}\left\{2\,\frac{y_i\left(w^T x_i + w_0\right)}{\|w\|}\right\}$$

$$= \frac{2}{\|w\|}\min_{x_i}\left\{y_i\left(w^T x_i + w_0\right)\right\}$$

Marginal Maximization Problem

$$\{\hat{w}, \hat{w}_0\} : \max_{w, w_0} \left\{ \frac{2}{\|w\|} \min_{x_i} \left\{ y_i \left(w^T x_i + w_0 \right) \right\} \right\}$$

Marginal Maximization Problem

$$\{\hat{w}, \hat{w}_0\} : \max_{w, w_0} \left\{ \frac{2}{\|w\|} \min_{x_i} \left\{ y_i \left(w^T x_i + w_0 \right) \right\} \right\}$$

- **Solution: Use a** scaling factor k:

$$k \min_{x_i} \left\{ y_i \left(w^T x_i + w_0 \right) \right\} = 1$$

Marginal Maximization Problem

$$\{\hat{w}, \hat{w}_0\} : \max_{w, w_0} \left\{ \frac{2}{\|w\|} \min_{x_i} \left\{ y_i \left(w^T x_i + w_0 \right) \right\} \right\}$$

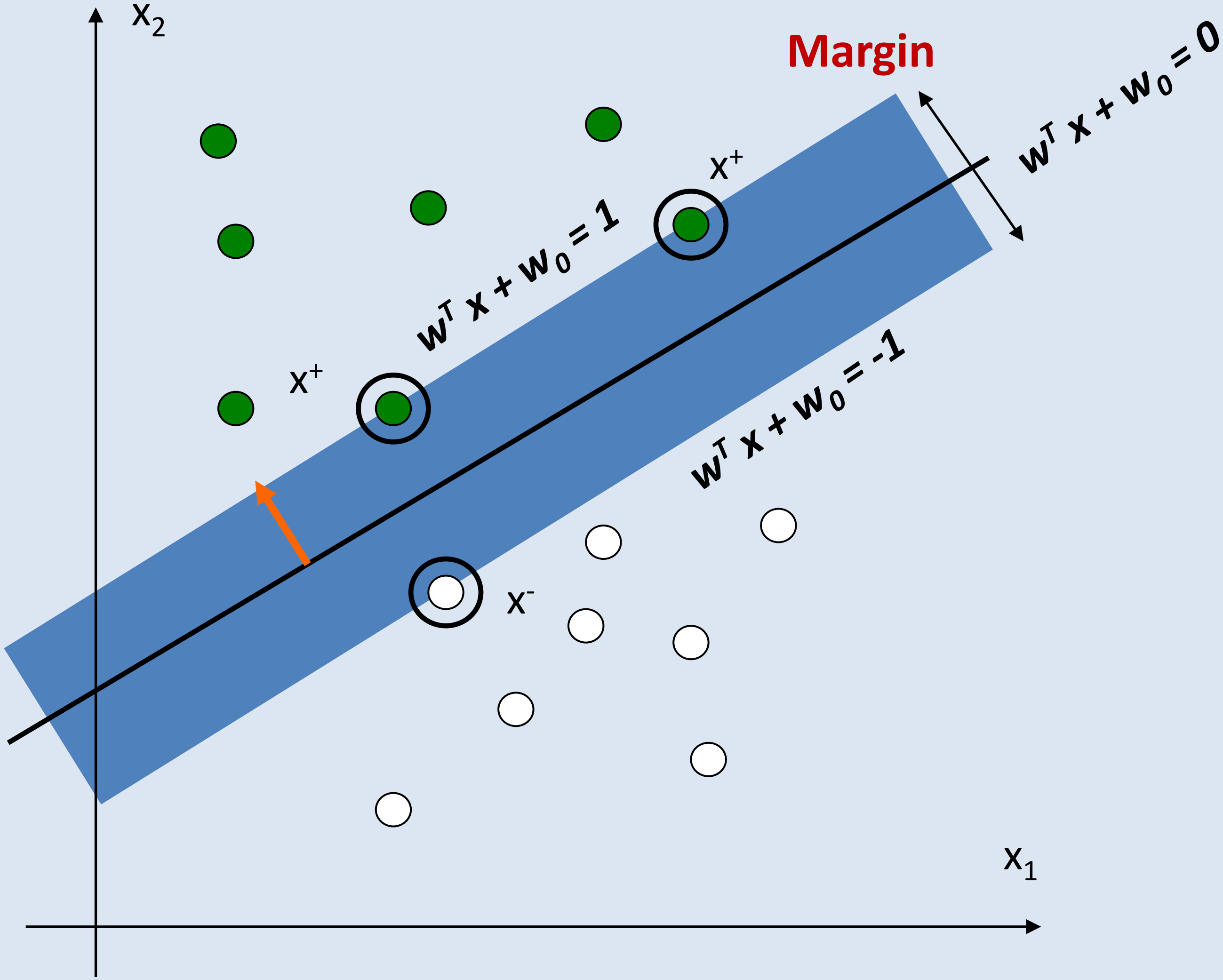
- **Solution: Use a** scaling factor k:

$$k \min_{x_i} \left\{ y_i \left(w^T x_i + w_0 \right) \right\} = 1$$

- Thus **margin** becomes:

$$\frac{2}{\|w\|} \min_{x_i} \left\{ y_i \left(w^T x_i + w_0 \right) \right\} = \frac{2}{\|w\|}$$

- **Therefore:** $\forall x_i \in D : y_i\left(w^T x_i + w_0\right) \geq 1$



The objective function

We need to optimize $\|w\|^{-1}$ which is the same as **minimizing** $\|w\|^2$ subject to the **margin requirements**

$$\{\hat{w}, \hat{w}_0\} : \max_{w, w_0} \left\{ \frac{2}{\|w\|} \right\} \quad \text{s.t.} \quad y_i \left(w^T x_i + w_0 \right) \geq 1 \ \forall i$$

$$\Updownarrow$$

$$\{\hat{w}, \hat{w}_0\} : \min_{w, w_0} \left\{ \frac{1}{2} \|w\|^2 \right\} \quad \text{s.t.} \quad y_i \left(w^T x_i + w_0 \right) \geq 1 \ \forall i$$

Quadratic Optimization Problem: minimize a quadratic function subject to a set of linear inequality constraints

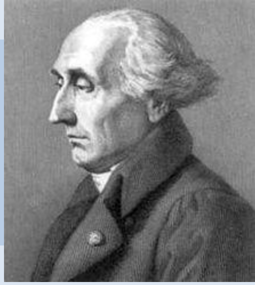
SVM Training Methodology

- Training is formulated as an optimization problem
  - **Dual problem** reduces computational complexity
  - **Kernel trick** is used to reduce computation

- Determination of the model parameters corresponds to a **convex optimization problem**.
  - Solution is straightforward (local solution is the global optimum)

- Makes use of **Lagrange multipliers**

Joseph-Louis Lagrange (1736-1813)



- Optimization problem with linear inequality constraints

$$\min_{\theta}\{f(\theta)\} \text{ s.t. } g(\theta) \geq c \Rightarrow g(\theta) - c \geq 0$$

- Lagrange function:

$$L(\theta, \lambda) = f(\theta) - \lambda(g(\theta) - c)$$

✓ **Karush-Khun-Tucker (KKT) conditions**:

$$\boxed{\begin{gathered} \lambda \geq 0 \\ g(\theta) - c \geq 0 \\ \lambda(g(\theta) - c) = 0 \end{gathered}}$$

Solving the Optimization Problem

- **Minimization Problem**:

$$\boxed{\{\hat{w}, \hat{w}_0\}: \min_{w, w_0}\left\{\frac{1}{2}\|w\|^2\right\} \text{ s.t. } y_i\left(w^T x_i + w_0\right) \geq 1 \;\; \forall i}$$

- **Lagrange function**:

$$L(w, w_0, a) = \frac{1}{2}\|w\|^2 - \sum_{i=1}^{N} a_i\left(y_i\left(w^T x_i + w_0\right) - 1\right)$$

KKT conditions

$$\begin{aligned} \forall i \qquad & a_i \geq 0 \\ & y_i\left(w^T x_i + w_0\right) - 1 \geq 0 \\ & a_i\left(y_i\left(w^T x_i + w_0\right) - 1\right) = 0 \end{aligned}$$

a_i *Lagrange multipliers*

Dual Optimization Problem

$$\text{minimize} \;\; L(w, w_0, a) = \frac{1}{2}\|w\|^2 - \sum_{i=1}^{N} a_i \left(y_i \left(w^T x_i + w_0 \right) - 1 \right)$$

$$\frac{\partial L}{\partial w} = 0 \Rightarrow \hat{w} = \sum_{i=1}^{N} a_i y_i x_i$$

$$\frac{\partial L}{\partial w_0} = 0 \Rightarrow \sum_{i=1}^{N} a_i y_i = 0$$

Prime problem

$$\text{minimize} \quad L(w, w_0, a) = \frac{1}{2}\|w\|^2 - \sum_{i=1}^{N} a_i \left(y_i \left(w^T x_i + w_0 \right) - 1 \right)$$

Prime problem

$$\boxed{\text{minimize}\ \ L(w, w_0, a) = \frac{1}{2}\|w\|^2 - \sum_{i=1}^{N} a_i \left(y_i \left(w^T x_i + w_0 \right) - 1 \right)}$$

$$\sum_{i=1}^{N} a_i y_i = 0 \quad \Updownarrow \quad \hat{w} = \sum_{i=1}^{N} a_i y_i x_i$$

Prime problem

$$\text{minimize}\ \ L(w, w_0, a) = \frac{1}{2}\|w\|^2 - \sum_{i=1}^{N} a_i\left(y_i\left(w^T x_i + w_0\right) - 1\right)$$

$$\sum_{i=1}^{N} a_i y_i = 0 \qquad \Updownarrow \qquad \hat{w} = \sum_{i=1}^{N} a_i y_i x_i$$

Dual problem

$$\text{maximize}\ L_D(a) = \sum_{i=1}^{N} a_i - \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N} a_i a_j y_i y_j x_i^T x_j$$

$$\textbf{s.t.}\quad a_i \geq 0 \quad , \quad \sum_{i=1}^{N} a_i y_i = 0$$

Important Remarks

1. The **Prime problem** has ***d+1* unknown** parameters that must be tuned. These are the linear coefficients $\{\boldsymbol{w}, w_0\}$, where d is the data dimension.

   The **Dual problem** has ***N* unknown** parameters which are the Lagrange multipliers $\{ a_i \; i=1,\ldots, N\}$, where N is the number of training samples.

   This is valuable and convenient for multi-dimensional data, where $d \gg N$, since the dual search space is significantly lower in comparison with the prime search space.

2. **The** **decision rule** for choosing the class of an unknown sample ***x*** becomes:

$$f(x) = w^T x + w_0 \qquad \Rightarrow \quad f(x) = \sum_{i=1}^{N} a_i y_i x_i^T x + w_0$$

$$\hat{w} = \sum_{i=1}^{N} a_i y_i x_i$$

which is a **linear combination of dot products** of ***x*** with all training samples $\boldsymbol{x_i}$, where each one has a unique weight equal to the Langrange multiplier $\boldsymbol{a_i}$.

3. **According to the KKT conditions** we have:

$$a_i \geq 0$$
$$y_i\left(w^T x_i + w_0\right) - 1 \geq 0$$
$$a_i\left(y_i\left(w^T x_i + w_0\right) - 1\right) = 0$$

Thus:

$$a_i = 0 \quad \textbf{and} \quad y_i\left(w^T x_i + w_0\right) > 1$$

or

$$y_i\left(w^T x_i + w_0\right) - 1 = 0 \quad \textbf{and} \quad a_i > 0$$

3. **According to the KKT conditions** we have:

$$a_i \geq 0$$
$$y_i\left(w^T x_i + w_0\right) - 1 \geq 0$$
$$a_i\left(y_i\left(w^T x_i + w_0\right) - 1\right) = 0$$

Thus:

$$a_i = 0$$

Training samples of D with zero weight outside the margin

or

$$y_i\left(w^T x_i + w_0\right) - 1 = 0$$

3. **According to the KKT conditions** we have:

$$a_i \geq 0$$
$$y_i\left(w^T x_i + w_0\right) - 1 \geq 0$$
$$a_i\left(y_i\left(w^T x_i + w_0\right) - 1\right) = 0$$

Thus:

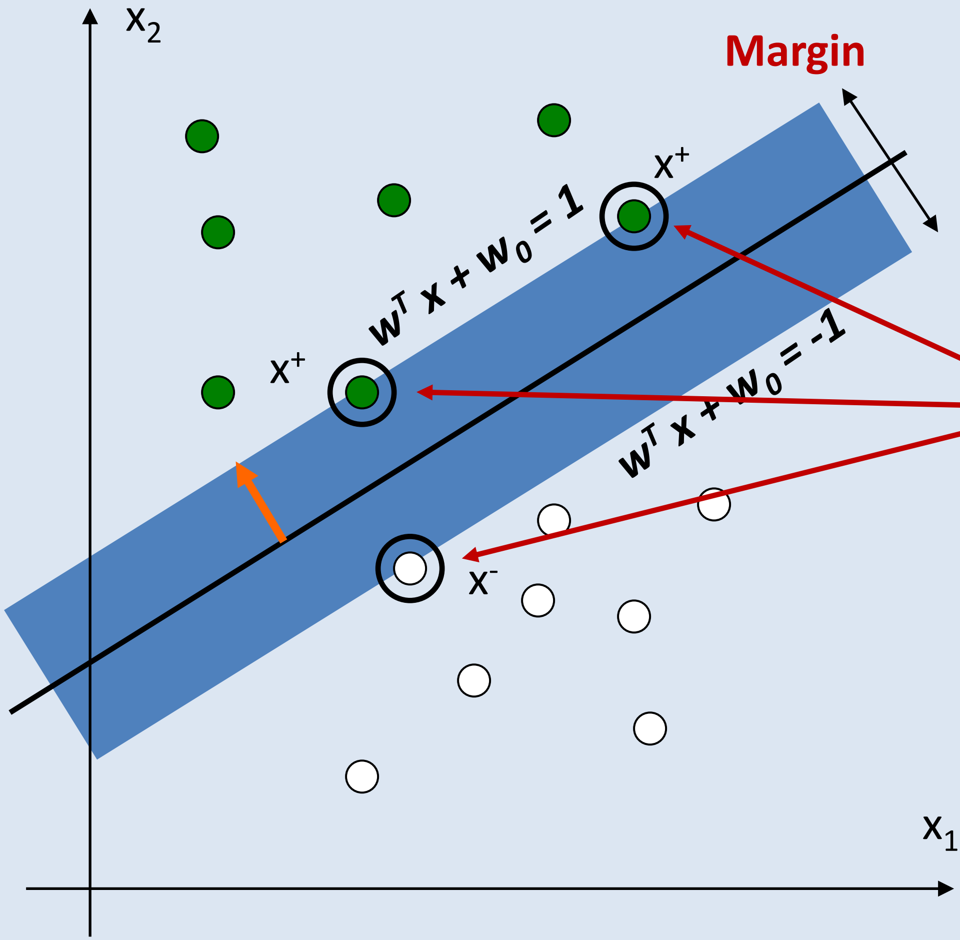
$$a_i = 0$$

or

$$y_i\left(w^T x_i + w_0\right) - 1 = 0$$

Training samples of D which are found on the margin

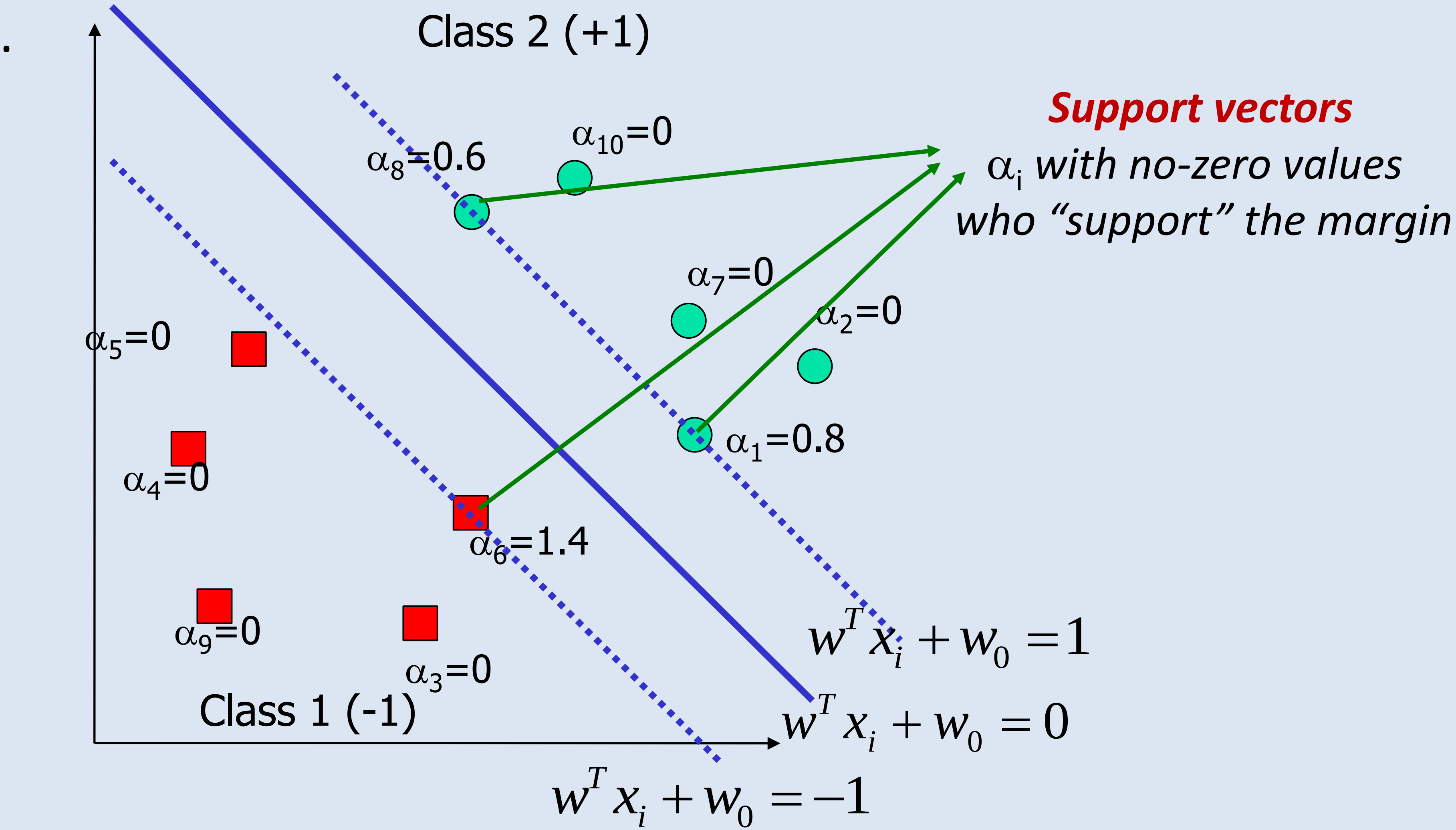
- All training samples **outside the margin** have ***a_i=0*** and they do not play any significant role to the decision.
- Training samples over the margin hold:



$$y_i\left(w^T x_i + w_0\right) - 1 = 0$$

and they have ***a_i >0***. These are called **support vectors** and they play important role to the decision.

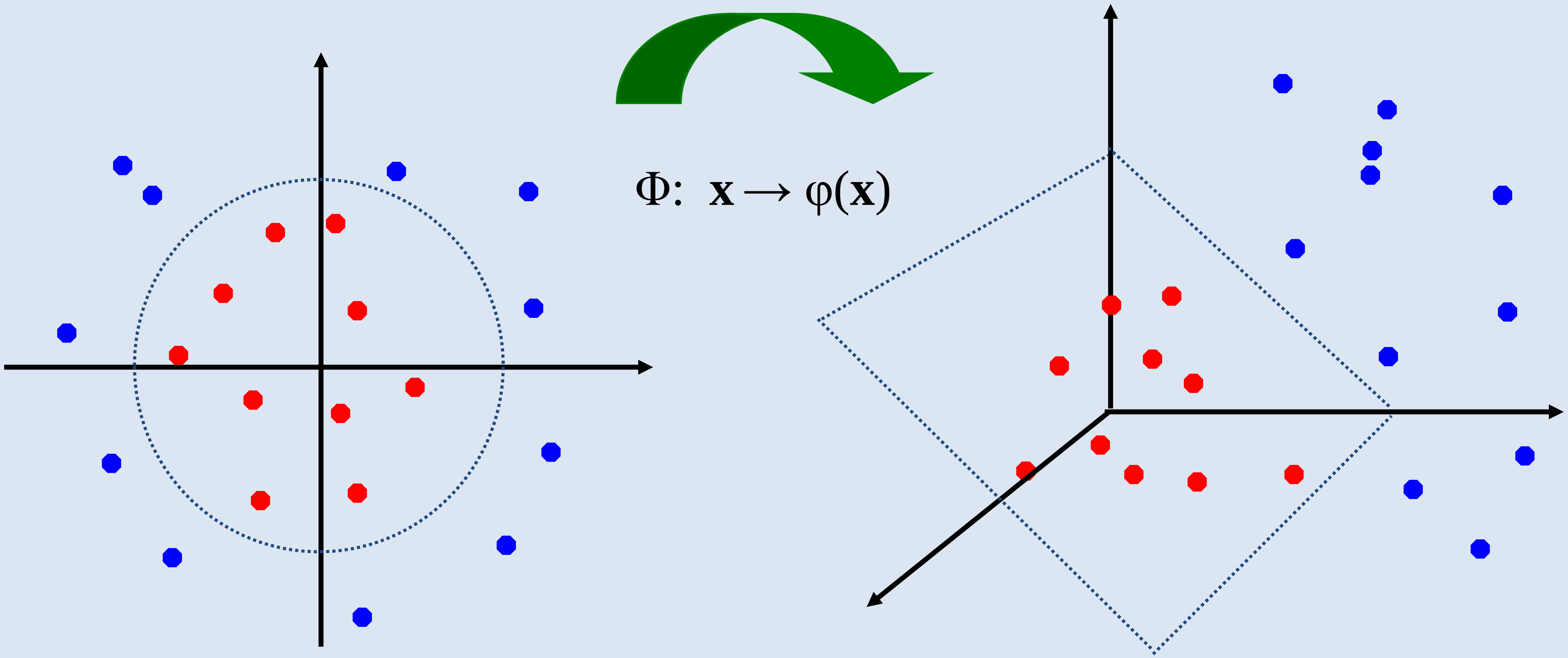
An example



Class 2 (+1)
α_{10}=0
α_8=0.6
Support vectors
α_i with no-zero values
who "support" the margin
α_7=0
α_2=0
α_5=0
α_1=0.8
α_4=0
α_6=1.4
α_9=0
α_3=0
Class 1 (-1)
$w^T x_i + w_0 = 1$
$w^T x_i + w_0 = 0$
$w^T x_i + w_0 = -1$

4. **Kernel trick:** Use a particular representation **φ(x)**

   Idea: The original feature space is transformed into a (usually) larger feature space which increases the likelihood of being linear separable.



- In the new space all dot products become:

$$x_i^T x_j \quad \rightarrow \quad \phi(x_i)^T \phi(x_j) \equiv K(x_i, x_j)$$

  which is called **kernel function** and specifies similarity

- The new decision rule can be written as:

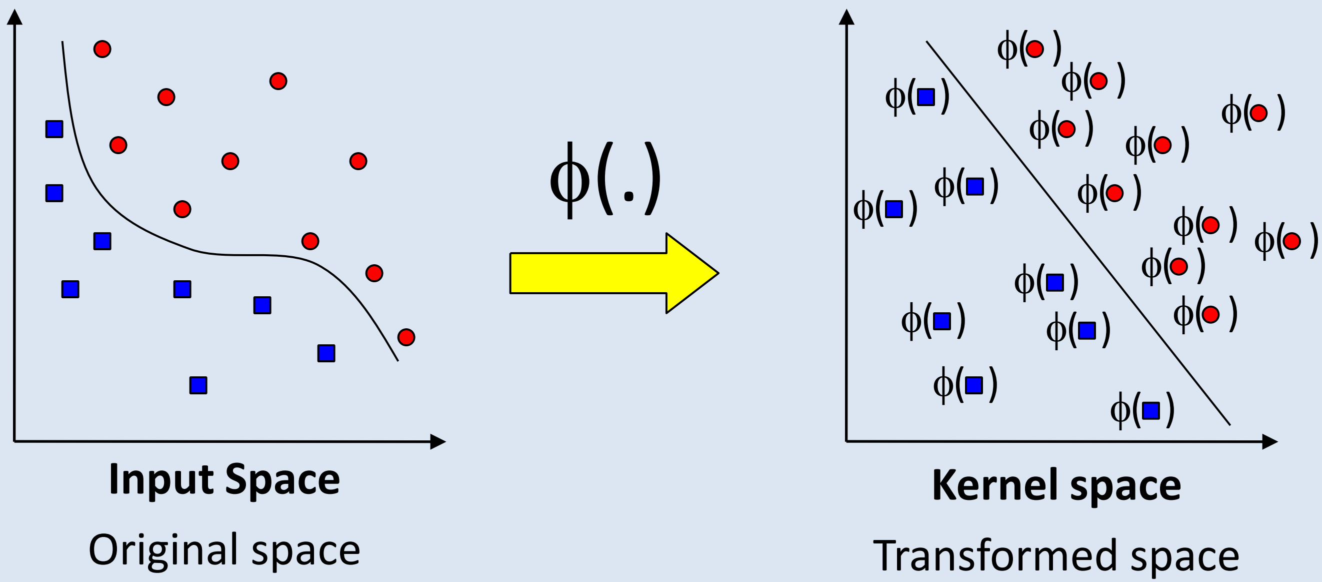
$$f(x) = \sum_{i=1}^{N} a_i y_i x_i^T x + w_0 \quad \rightarrow \quad f(x) = \sum_{i=1}^{N} a_i y_i \phi(x_i)^T \phi(x) + w_0$$

$$f(x) = \sum_{i=1}^{N} a_i y_i K(x_i, x) + w_0$$

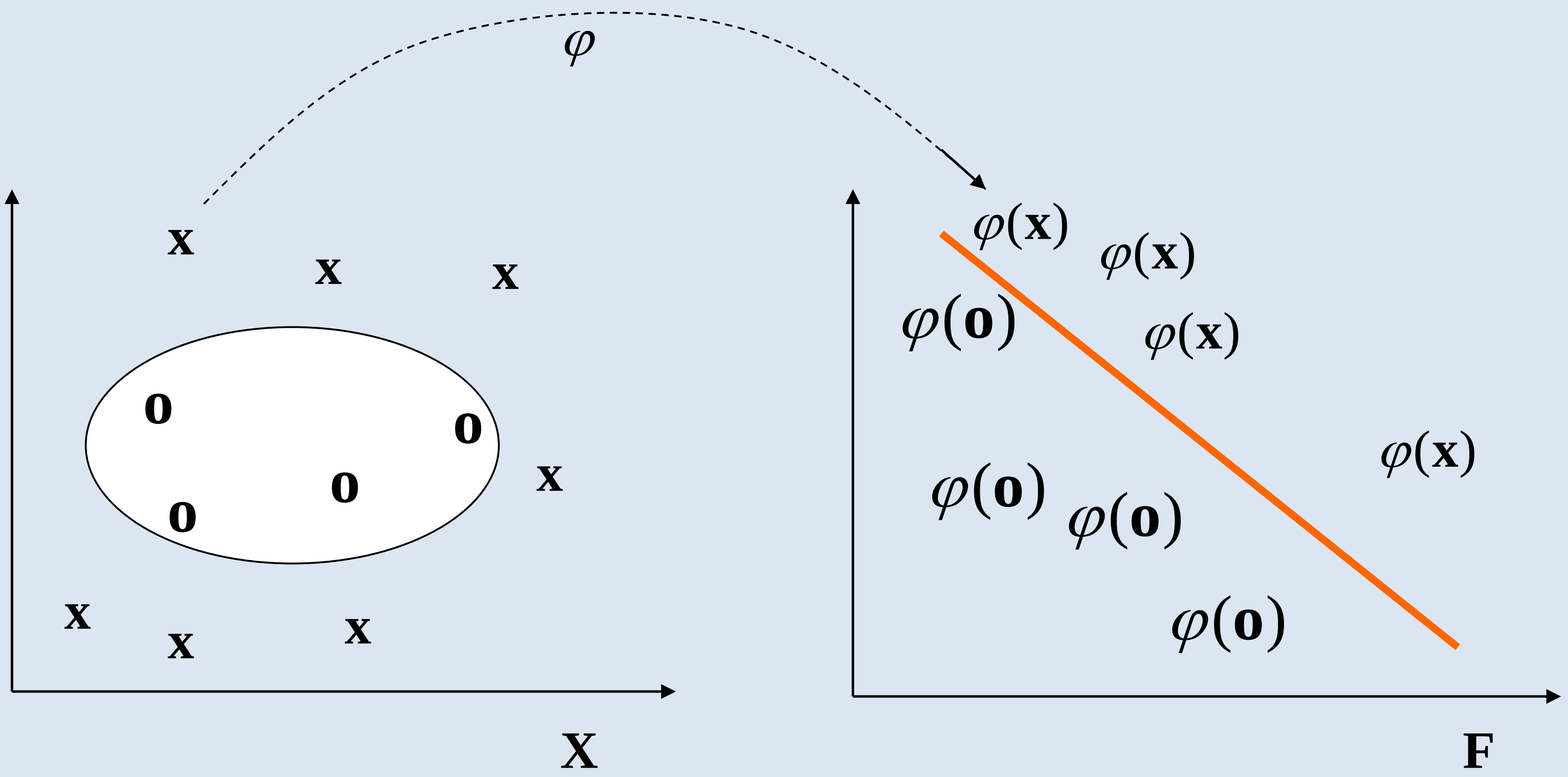
Examples of kernel functions

- Linear Kernel $$K(x_i, x_j) = x_i^T x_j$$
- Polynomial Kernel $$K(x_i, x_j) = (x_i^T x_j + 1)^p$$
- Gaussian ή RBF Kernel $$K(x_i, x_j) = e^{-\frac{\|x_i - x_j\|^2}{2\sigma^2}}$$
- Cosine $$K(x_i, x_j) = \frac{x_i^T x_j}{\|x_i\|^2 \|x_j\|^2}$$
- Sigmoid $$K(x_i, x_j) = \frac{1}{1 + e^{-(\beta_1 x_i^T x_j + \beta_0)}}$$
- ……

Example 1: Construct a linear feature space using $\phi(x)$



Example 2



φ
x
x
x
0
0
0
0
x
x
x
x
X
$\varphi(\mathbf{x})$
$\varphi(\mathbf{x})$
$\varphi(\mathbf{o})$
$\varphi(\mathbf{x})$
$\varphi(\mathbf{x})$
$\varphi(\mathbf{o})$
$\varphi(\mathbf{o})$
$\varphi(\mathbf{o})$
F

5. Estimate the constant term w_0

- Set of support vectors $S = \{x_i : y_i(w^T\phi(x_i) + w_0) = 1\}$
- Substituting $\hat{w} = \sum_{j=1}^{N} a_j y_j \phi(x_j)$

  we take: $$y_i\left(\sum_{x_j \in S} a_j y_j \phi(x_j)^T \phi(x_i) + w_0\right) = 1 \quad \forall x_i \in S$$

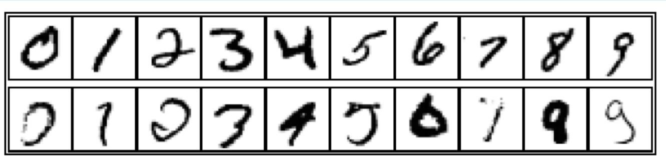
- Summing all:

$$\sum_{x_i \in S} y_i\left(\sum_{x_j \in S} a_j y_j K(x_j, x_i) + w_0\right) = N_s = |S|$$

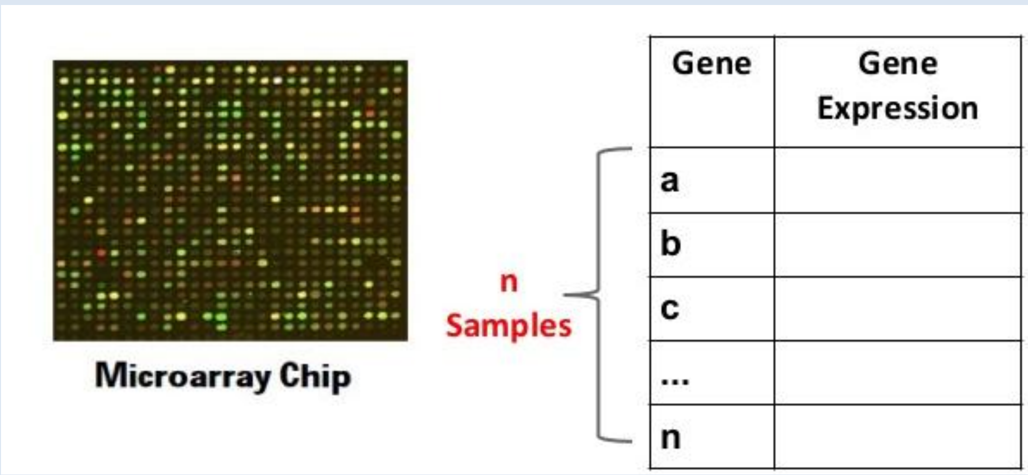
size of S

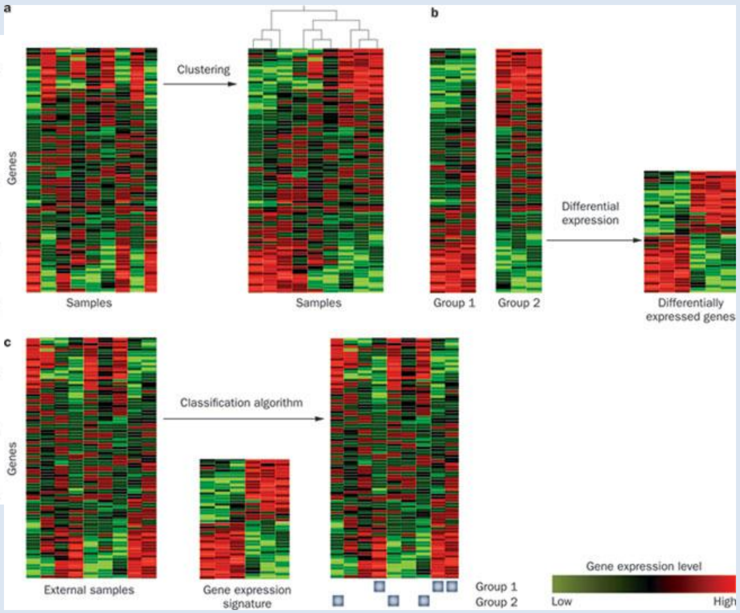
Applications

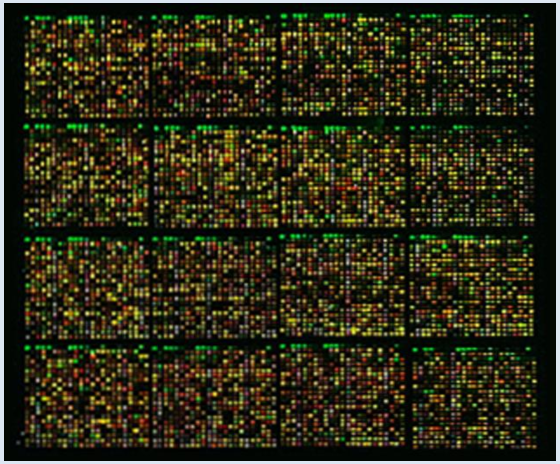
- Bioinformatics
- Text categorization – mining
- Handwritten character recognition
- Computer Vision
- Time series analysis
- …..



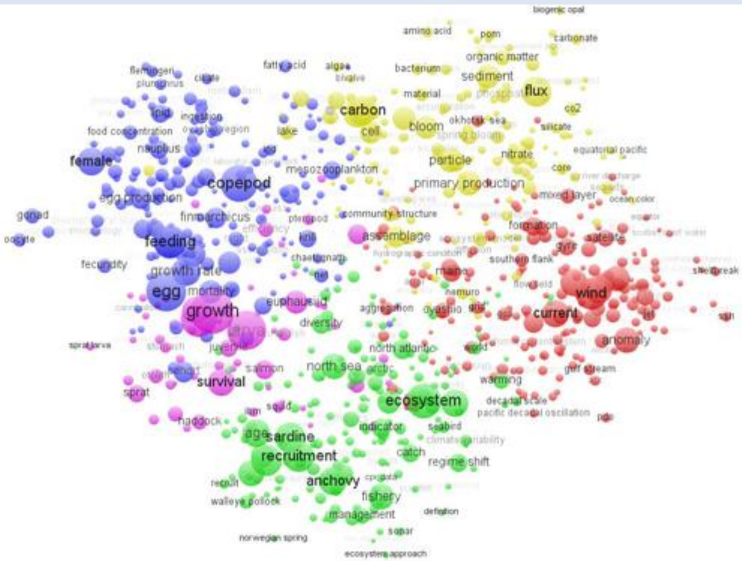
- Bioinformatics – gene expression data

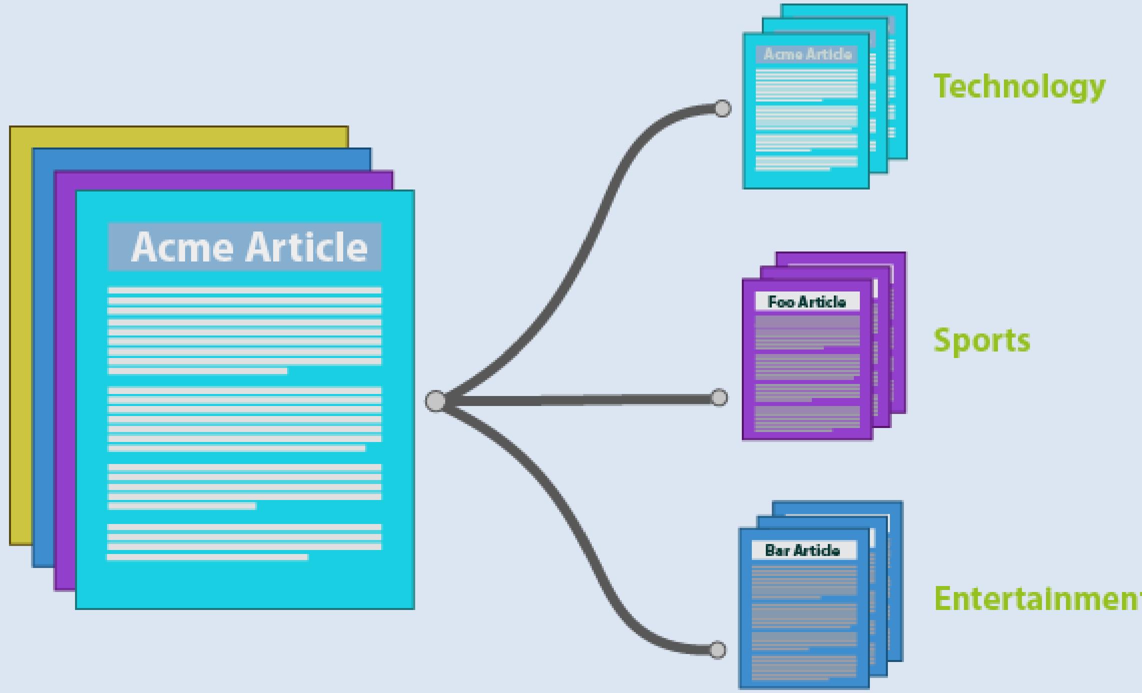






- Text categorization – mining

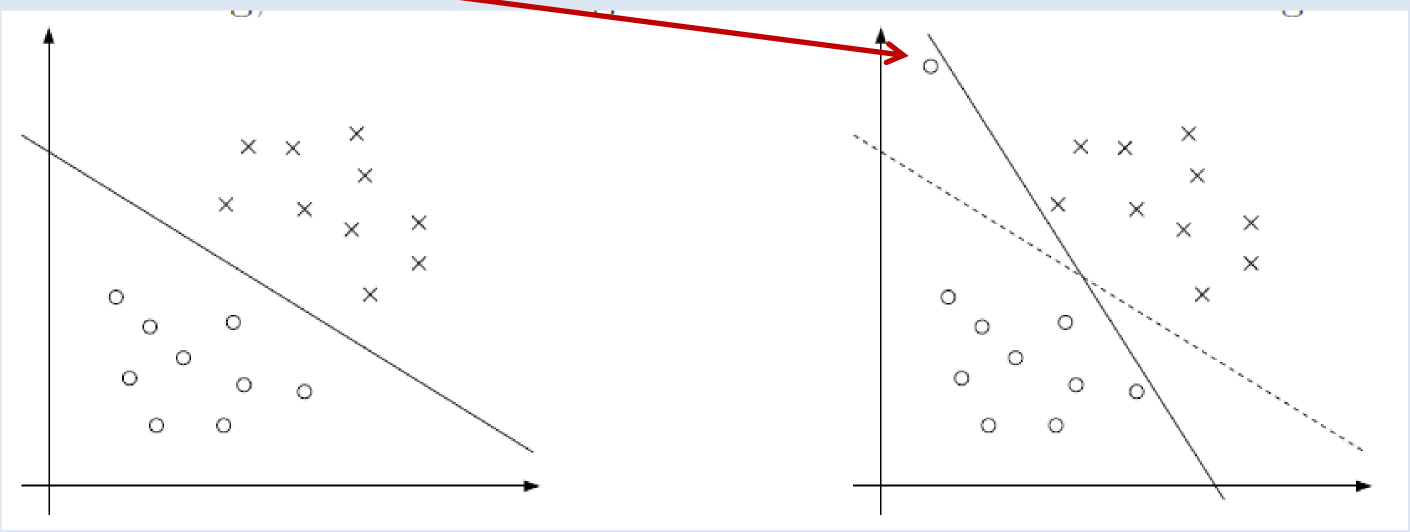




Bag of words (lexicon)

Nonlinear SVM
The non-separable case

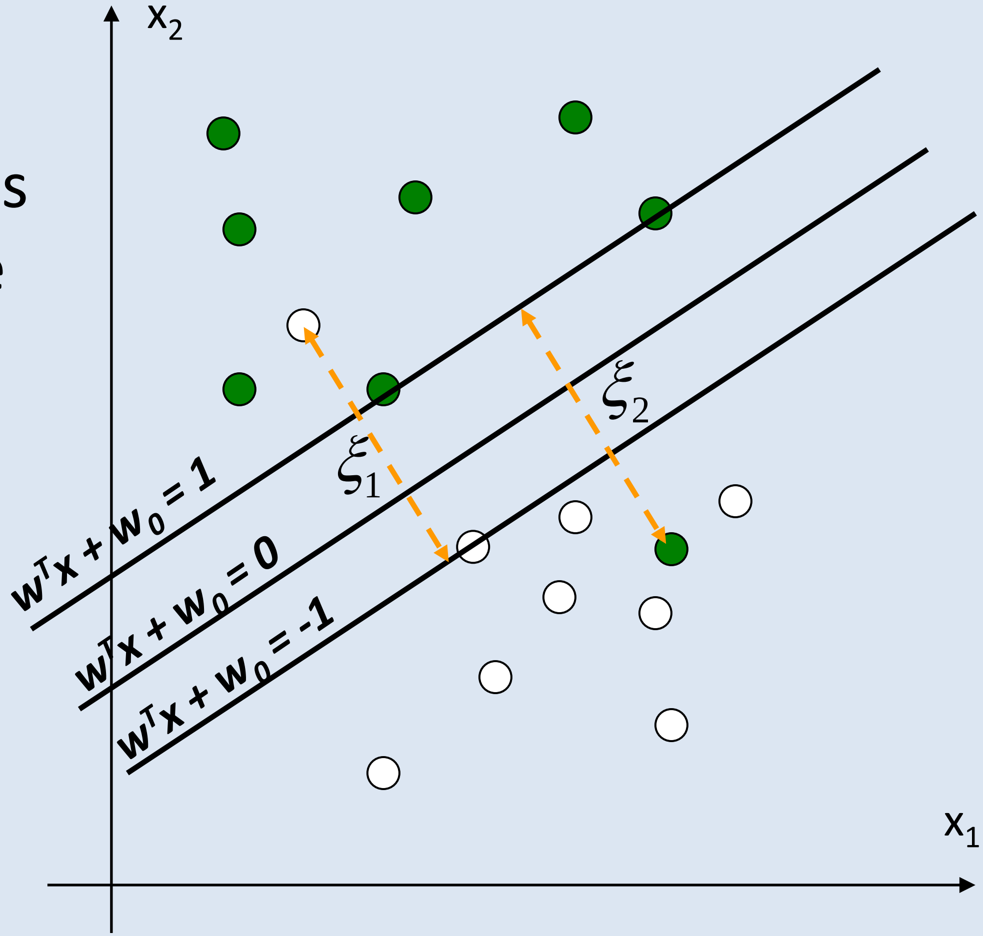
- ✓ Mapping data to a high dimensional space, via ϕ(x), increase the likelihood the data be separable.
- ✓ However, this cannot be guaranteed.
- ✓ Also, separating hyperplane might be **susceptible to outliers.**



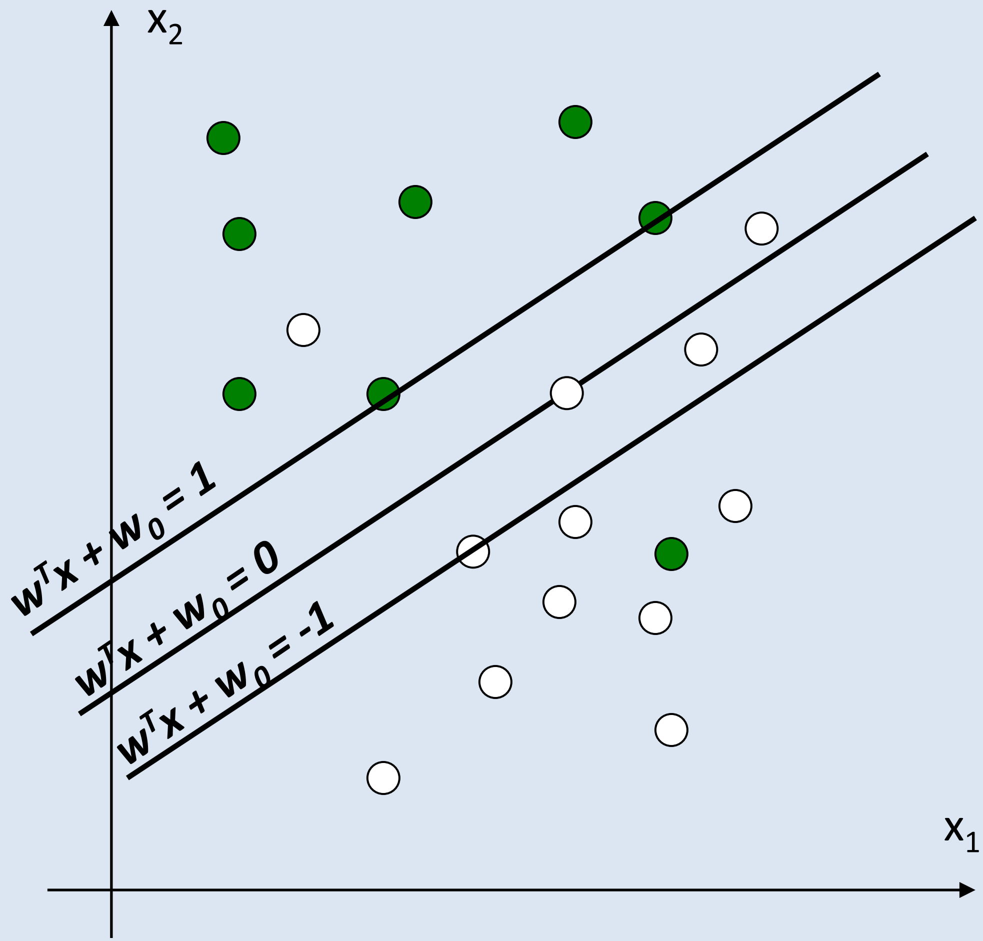
Nonlinear SVM
The non-separable case

- Need to make the algorithm work for non-linearly separable cases, as well as to be less sensitive to outliers.

➢ Introduction of **auxiliary variables ξ_i** which allow errors, i.e. samples being in erroneous side of margin.

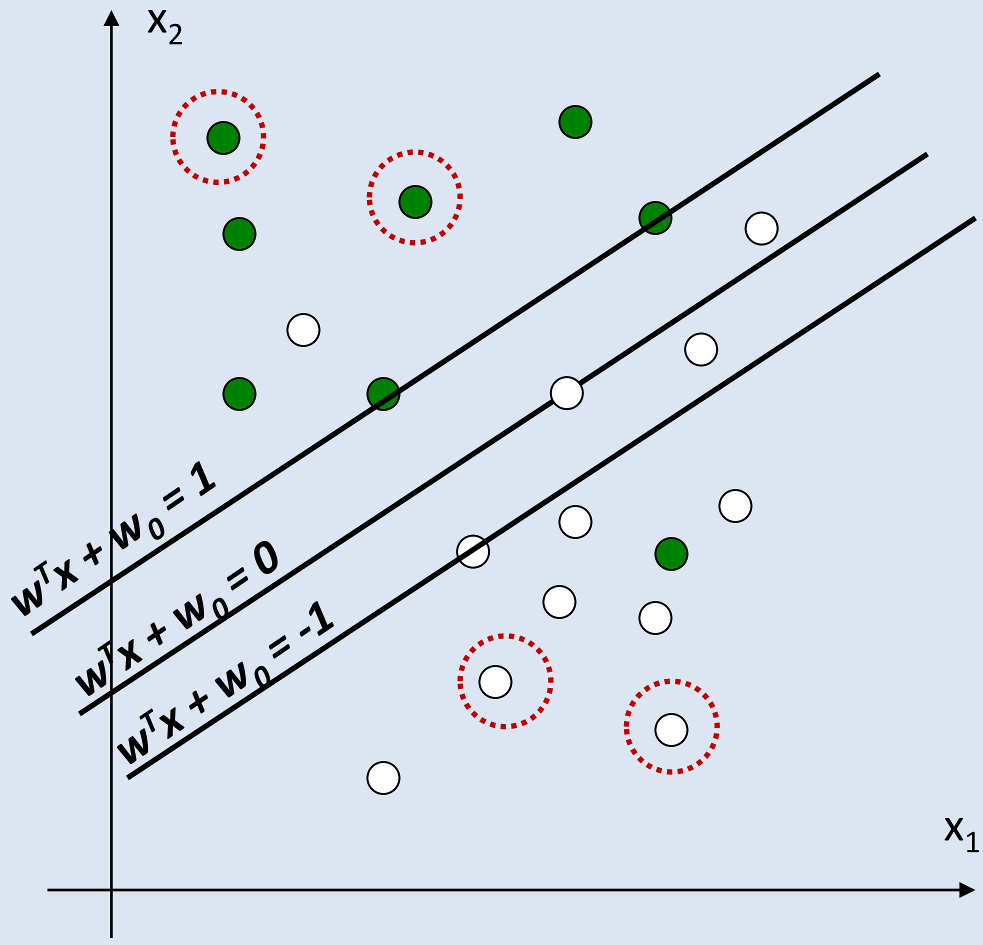


For any sample x_i : $\xi_i = \left| y_i - f\left(x_i\right) \right|$



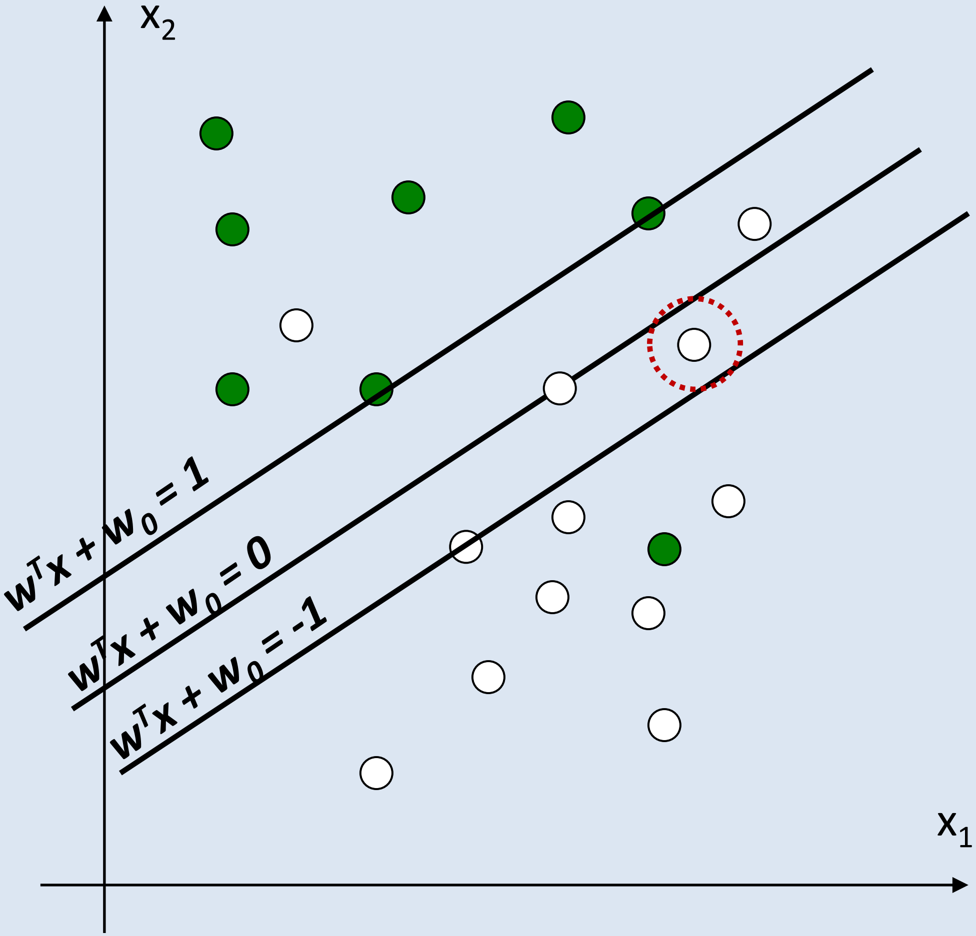
For any sample x_i : $\xi_i = |y_i - f(x_i)|$

- If x_i found in the right side (**no error**), then $\boldsymbol{\xi_i = 0}$.



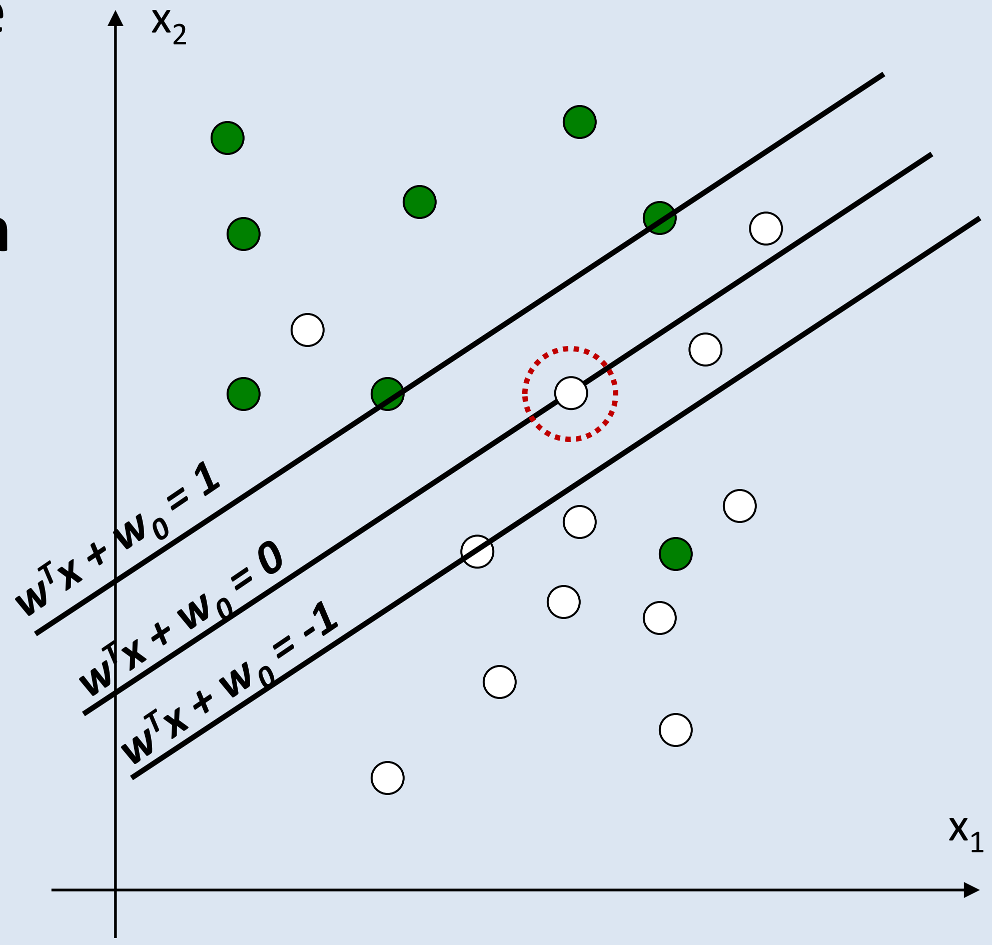
For any sample x_i : $\xi_i = |y_i - f(x_i)|$

- If x_i found in the right side (**no error**), then $\boldsymbol{\xi_i = 0}$.
- If found **inside the margin** but in the **right side** $\boldsymbol{\xi_i < 1}$



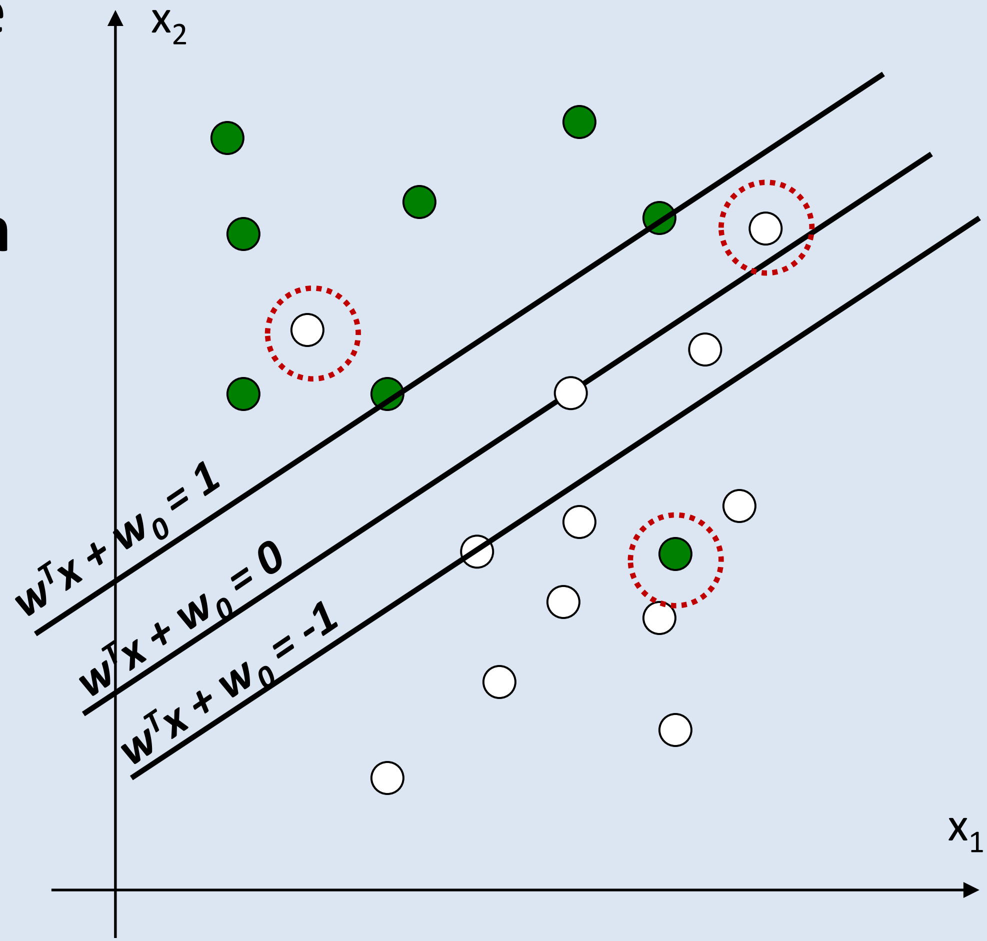
For any sample x_i : $\xi_i = |y_i - f(x_i)|$

- If x_i found in the right side (**no error**), then $\boldsymbol{\xi_i = 0}$.
- If found **inside the margin** but in the **right side** $\boldsymbol{\xi_i < 1}$
- If found exactly in the **hyperplane** where $w^T x + w_0 = 0$ then $\boldsymbol{\xi_i = 1}$



For any sample x_i : $\xi_i = \left| y_i - f\left(x_i\right) \right|$

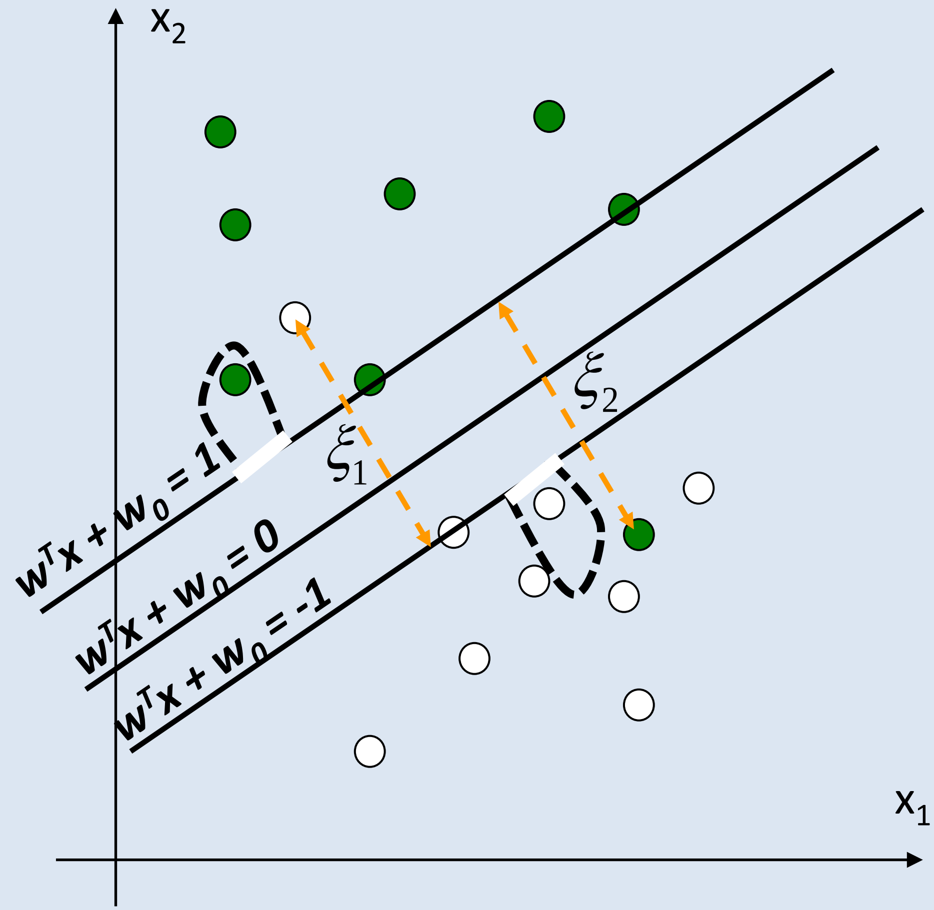
- If x_i found in the right side (**no error**), then $\boldsymbol{\xi_i = 0}$.
- If found **inside the margin** but in the **right side** $\boldsymbol{\xi_i < 1}$
- If found exactly in the **hyperplane** where $w^T x+w_0=0$ then $\boldsymbol{\xi_i = 1}$
- If it is **wrong classified** then $\boldsymbol{\xi_i > 1}$



- We allow margin be less than 1

$$\forall i \quad y_i\left(w^T x_i + w_0\right) \geq 1 - \xi_i$$

- $\boldsymbol{\xi_i}$ plays to role of **error tolerance** for every sample x_i and sets up the **local margin** which allows margin to enter the space of other class.



Nonlinear SVM

- Objective function:
- $\sum_{i=1}^{N} \xi_i$ is the total error tolerance of training set
- **Problem:**

$$\min_{w, w_0, \xi} \left\{ \frac{1}{2} \|w\|^2 + C \sum_{i=1}^{N} \xi_i \right\}$$

$$\textbf{s.t.}\ \ y_i \left(w^T x_i + w_0 \right) \geq 1 - \xi_i \ \ \forall i$$

$$\xi_i \geq 0$$

Nonlinear SVM

- **Problem:**

$$\min_{w,w_0,\xi}\left\{\frac{1}{2}\|w\|^2 + C\sum_{i=1}^{N}\xi_i\right\}$$

$$\textbf{s.t. } y_i\left(w^T x_i + w_0\right) \geq 1 - \xi_i \quad \forall i$$

$$\xi_i \geq 0$$

Lagrange function

Lagrange multipliers (≥ 0)

$$L(w, w_0, \xi) = \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{N}\xi_i - \sum_{i=1}^{N} a_i\left(y_i\left(w^T x_i + w_0\right) - 1 + \xi_i\right) - \sum_{i=1}^{N}\mu_i \xi_i$$

The dual form of the problem

$$\text{minimize} \;\; L(w, w_0, \xi) = \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{N}\xi_i - \sum_{i=1}^{N} a_i\left(y_i\left(w^T x_i + w_0\right) - 1 + \xi_i\right) - \sum_{i=1}^{N}\mu_i\xi_i$$

KKT conditions

$$a_i \geq 0$$

$$y_i\left(w^T x_i + w_0\right) - 1 + \xi_i \geq 0$$

$$a_i\left(y_i\left(w^T x_i + w_0\right) - 1 + \xi_i\right) = 0$$

or

$$a_i = 0 \qquad y_i\left(w^T x_i + w_0\right) - 1 + \xi_i = 0$$

$$\mu_i \geq 0$$

$$\xi_i \geq 0$$

$$\mu_i \xi_i = 0$$

or

$$\mu_i = 0 \qquad \xi_i = 0$$

The dual form of the problem

$$\text{minimize} \;\; L(w, w_0, \xi) = \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{N}\xi_i - \sum_{i=1}^{N} a_i\left(y_i\left(w^T x_i + w_0\right) - 1 + \xi_i\right) - \sum_{i=1}^{N}\mu_i\xi_i$$

Partial derivatives

$$\frac{\partial L}{\partial w} = 0 \Rightarrow \hat{w} = \sum_{i=1}^{N} a_i y_i x_i$$

$$\frac{\partial L}{\partial w_0} = 0 \Rightarrow \sum_{i=1}^{N} a_i y_i = 0$$

$$\frac{\partial L}{\partial \xi_i} = 0 \Rightarrow a_i = C - \mu_i$$

The dual form of the problem

$$\text{minimize} \;\; L(w, w_0, \xi) = \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{N}\xi_i - \sum_{i=1}^{N} a_i\left(y_i\left(w^T x_i + w_0\right) - 1 + \xi_i\right) - \sum_{i=1}^{N}\mu_i \xi_i$$

Dual form of the problem

$$\text{maximize} \;\; L_D(a) = \sum_{i=1}^{N} a_i - \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N} a_i a_j y_i y_j x_i^T x_j$$

s.t. $0 \le a_i \le C$, $\quad \sum_{i=1}^{N} a_i y_i = 0$

The dual form of the problem

$$\text{maximize} \;\; L_D(a) = \sum_{i=1}^{N} a_i - \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N} a_i a_j y_i y_j x_i^T x_j$$

$$\textbf{s.t.} \quad 0 \le a_i \le C \;\;,\;\; \sum_{i=1}^{N} a_i y_i = 0$$

- If $\boldsymbol{a_i > 0}$ **then** x_i are **support vectors**:

$$y_i\left(w^T x_i + w_0\right) - 1 + \xi_i = 0$$

- If $\boldsymbol{a_i < C}$ **then** $\mu_i > 0$ and $\boldsymbol{\xi_i = 0}$. It holds:

$$y_i\left(w^T x_i + w_0\right) - 1 = 0$$

The dual form of the problem

$$\text{maximize } L_D(a) = \sum_{i=1}^{N} a_i - \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N} a_i a_j y_i y_j x_i^T x_j$$

$$\textbf{s.t.} \quad 0 \le a_i \le C \quad , \quad \sum_{i=1}^{N} a_i y_i = 0$$

- If $\boldsymbol{a_i = C}$ **then** $\mu_i = 0$ and $\boldsymbol{\xi_i > 0}$. Sample x_i **is inside the margin**
  - If $\boldsymbol{\xi_i \le 1}$ then x_i is **right classified**,
  - If $\boldsymbol{\xi_i > 1}$ then x_i is **wrong classified**

The dual form of the problem

$$\text{maximize} \quad L_D(a) = \sum_{i=1}^{N} a_i - \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N} a_i a_j y_i y_j x_i^T x_j$$

$$\textbf{s.t.} \quad 0 \le a_i \le C \quad , \quad \sum_{i=1}^{N} a_i y_i = 0$$

- If ***a_i = C*** **then** $\mu_i = 0$ and **$\xi_i > 0$**. Sample x_i **is inside the margin**
  - If ***$\xi \le 1$*** then x_i is **right classified**,
  - If ***$\xi_i > 1$*** then x_i is **wrong classified**

The SMO algorithm

J. Platt, Fast Training of Support Vector Machines using Sequential Minimal Optimization, MIT Press (1998).

- **Sequential Minimal Optimization (SMO)**

- **Solving the dual problem**

$$\text{maximize} \;\; L_D(a) = \sum_{i=1}^{N} a_i - \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} a_i a_j y_i y_j x_i^T x_j$$

$$\textbf{s.t.} \quad 0 \le a_i \le C \quad , \quad \sum_{i=1}^{N} a_i y_i = 0$$

SMO algorithmic structure

- SMO breaks this problem into a **series of smallest possible sub-problems**, which are then solved sequentially.
- The smallest problem involves **two such multipliers :**

$$0 \le a_1, a_2 \le C \qquad \text{and} \qquad a_1 y_1 + a_2 y_2 = -\sum_{i=3}^{N} a_i y_i = \zeta$$

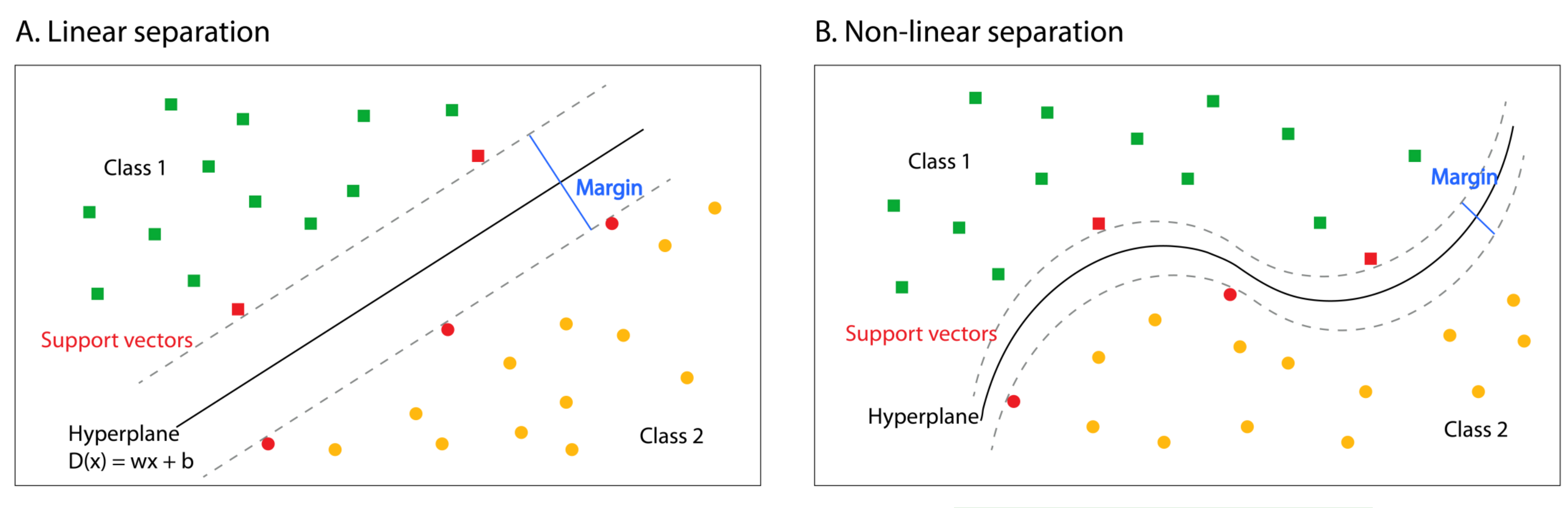
- This reduced problem can be solved analytically:

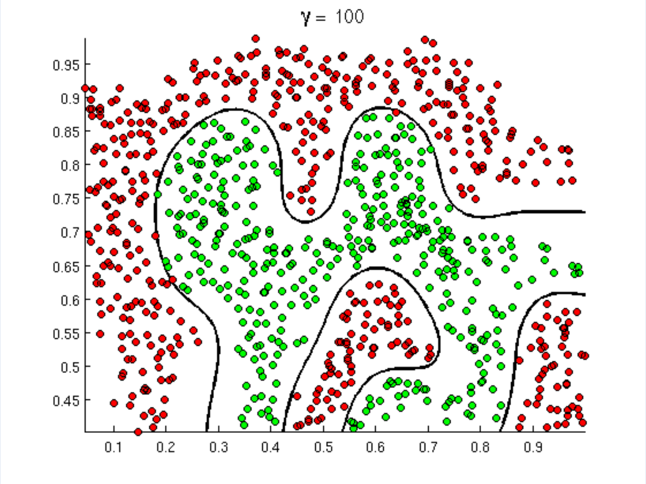
$$a_1 = y_1(\zeta - a_2 y_2) \qquad \hat{a}_2 : \max_{a_2} \{L_D(a)\}\Big|_{a_1 = y_1(\zeta - a_2 y_2)}$$

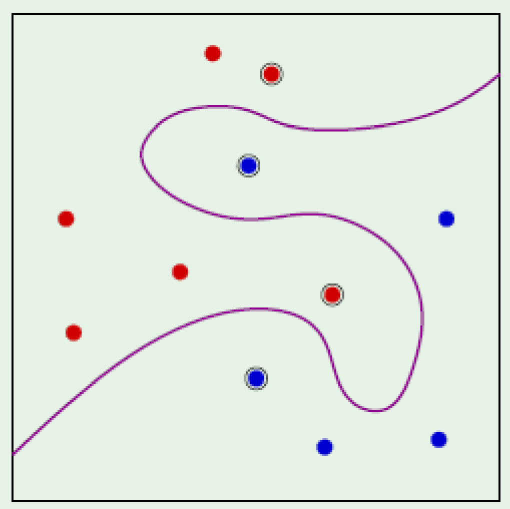
$$a_2^{(new)} = \begin{cases} C & \text{if } \hat{a}_2 > C \\ \hat{a}_2 & \text{if } 0 \le \hat{a}_2 \le C \\ 0 & \text{if } \hat{a}_2 < 0 \end{cases}$$

$$\hat{a}_1 = y_1(\zeta - \hat{a}_2 y_2)$$

Examples of non-linear svm classification







Multi-class Classification
Working with more than 2 classes

Two general schemes

- **one vs. all classifiers**
- **Pairwise Classifiers**

One vs. All Classifiers

- One classifier for every class $j = 1,\ldots,K$
- Samples of examined class are positive (label +1), while rest samples from all other K-1 classes are negative examples with label -1.
- Training the K different classifiers and construct functions:

$$f_j(x) = \varphi\left(w_{j0} + \sum_{i=1}^{d} w_{ji} x_i\right)$$

- **Decision rule:** Classify an unknown sample x to the class with the maximum function value:

$$c(x) = \arg\max_{j=1,\ldots,K} f_j(x)$$

Pairwise Classifiers

- One classifier for every pair of classes (j, k)

- Training the K*(K-1) classifiers and construct separating functions for every pair:

$$f_{jk}(x) = \varphi\left(w_0^{(j,k)} + \sum_{j=1}^{d} w_j^{(j,k)} x_j \right)$$

- **Decision rule:** Classify an unknown sample x to the class with the most votes among all classifiers.
- In case of **equivalence** use the functions' values for taking the decision.