Decision surfaces.
Support Vector Machines

V. Mygdalis, F. Fotopoulos, Prof. Ioannis Pitas
Aristotle University of Thessaloniki
pitas@csd.auth.gr
www.aiia.csd.auth.gr
Version 4.2

Artificial Intelligence &
Information Analysis Lab

Outline

- Decision surfaces
- Hyperplanes
- Non-linear Decision Surfaces
- 2^{nd} degree polynomial surfaces
- Hyperellipsoid/Hyperparaboloid
- Support Vector Machines

Decision surfaces

- ***Classification***:
  - Two class $(m = 2)$ and multiple class $(m > 2)$ classification.
  - Example: ***Face detection*** (two classes), ***face recognition*** (many classes).
- Two class $\mathcal{C}_1, \mathcal{C}_2$ (binary) classification of sample $\mathbf{x} \in \mathbb{R}^n$:
  - One (binary) hypothesis to be tested:

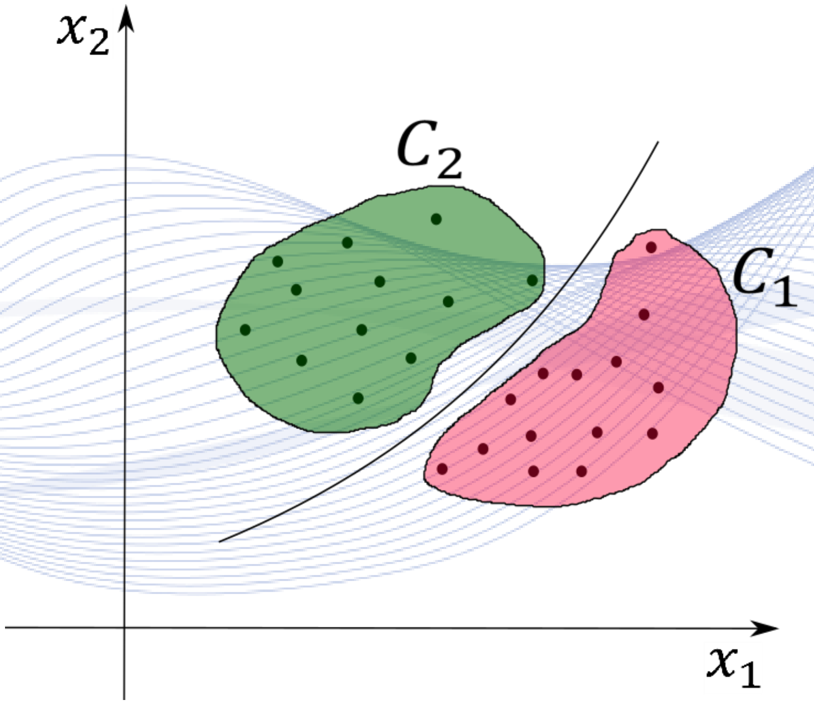
$$\mathcal{H}_1: \ \mathbf{x} \in \mathcal{C}_1, \qquad \mathcal{H}_2: \ \mathbf{x} \in \mathcal{C}_2.$$

  - Use one ***decision surface*** to separate two classes.

Two Class Classification

Two class $\mathcal{C}_1$, $\mathcal{C}_2$ (***binary***) classification of sample $\mathbf{x} \in \mathbb{R}^n$:

- A binary hypothesis to be tested: $\mathbf{x}$ is either in $\mathcal{C}_1$ or in $\mathcal{C}_2$.
- Find a decision surface to separate two classes.



Hyperplanes

- ***Hyperplane*** $\mathbb{H}$ is described by a linear equation having parameters w_0 , $\mathbf{w} = [w_1, \dots, w_n]^T$:

$$\sum_{j=1}^{n} w_j x_j + w_0 = \mathbf{w}^T \mathbf{x} + w_0 = 0, \qquad \mathbf{x} = [x_1, \dots, x_n]^T.$$

- Distance of a point $\mathbf{x}$ from hyperplance $\mathbb{H}$:

$$d(\mathbf{x}, \mathbb{H}) = \frac{|\mathbf{w}^T \mathbf{x} + w_0|}{||\mathbf{w}||}.$$

Artificial Intelligence & Information Analysis Lab

Hyperplanes

Linear discriminant function:

$$g(\mathbf{x}) = \mathbf{w}^T\mathbf{x} + b.$$

- $\mathbf{w}$ is the weight vector and b (or w_0) is the bias (or threshold weight).
- ***Decision rule***:
- If $g(\mathbf{x}) > 0$ then $\mathbf{x}$ is assigned in $\mathcal{C}_1$ class.
- Otherwise, if $g(\mathbf{x}) < 0$, it is assigned in $\mathcal{C}_2$ class.
- The ***decision surface*** $g(\mathbf{x}) = 0$ separates points assigned to $\mathcal{C}_1$ from points assigned to $\mathcal{C}_2$.

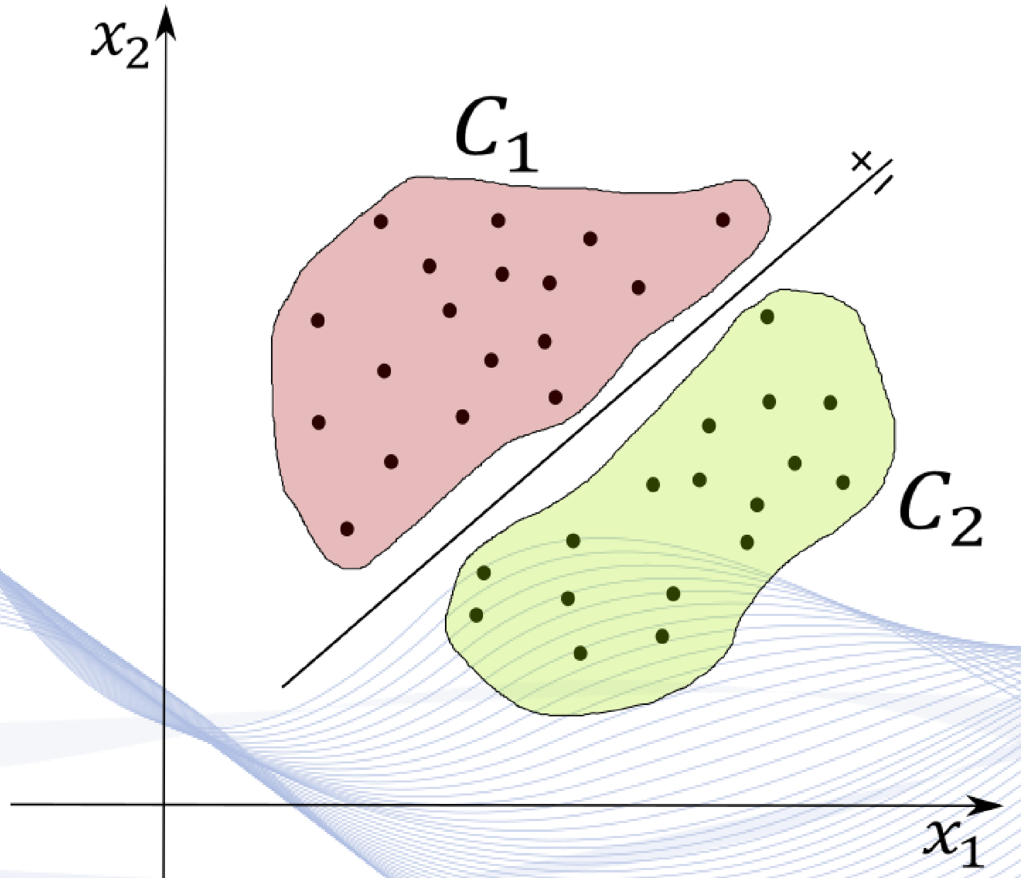
Hyperplanes

- If $g(\mathbf{x})$ is **linear**, the decision surface is a ***hyperplane*** $\mathbb{H}$.
- It divides the feature space into two half-spaces, decision region $\mathcal{R}_1$ for $\mathcal{C}_1$ and region $\mathcal{R}_2$ for $\mathcal{C}_2$.
- We usually consider any $\mathbf{x}$ point in $\mathcal{R}_1$ to be on the ***positive side*** $g(\mathbf{x}) > 0$ and, respectively, any point in $\mathcal{R}_2$ to be on the ***negative side*** $g(\mathbf{x}) < 0$.
- $\mathbf{x}$ can also be expressed by its distance d from the hyperplane:

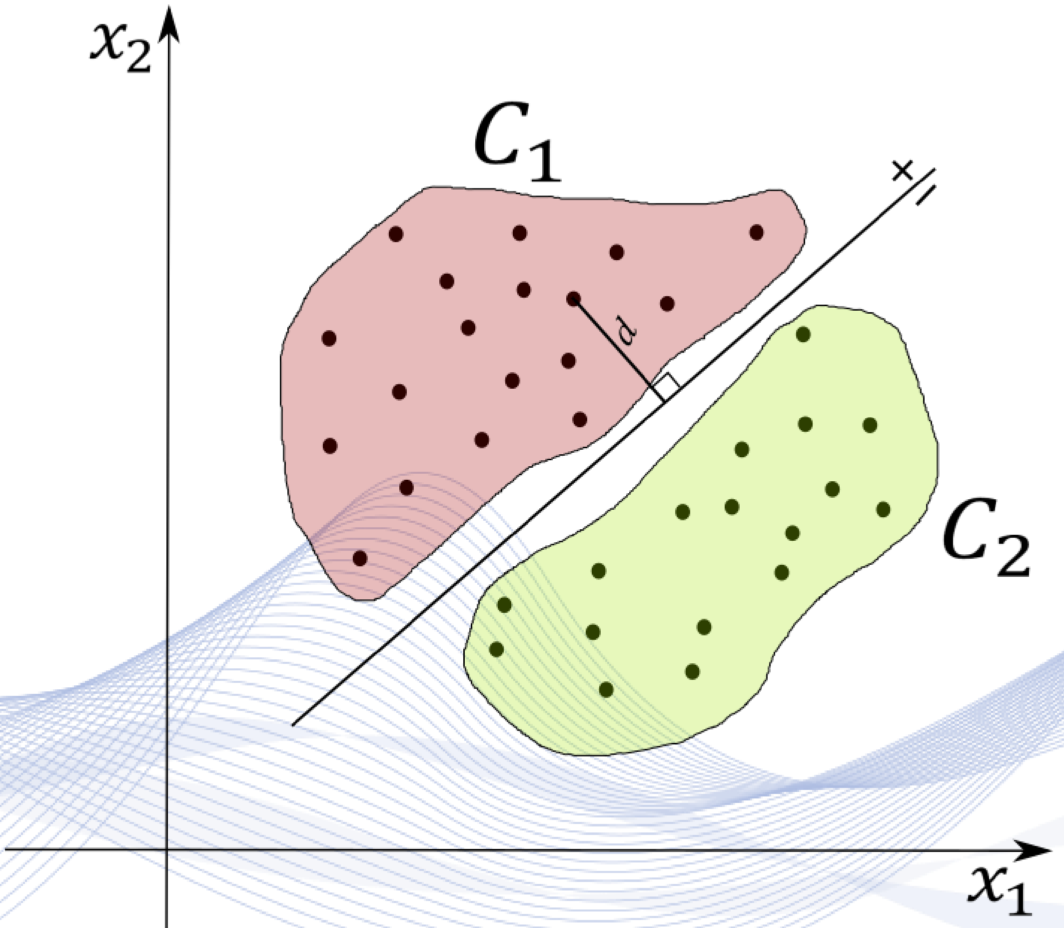
$$\mathbf{x} = \mathbf{x}_p + d\frac{\mathbf{w}}{||\mathbf{w}||},$$

- $\mathbf{x}_p$ is the normal projection of $\mathbf{x}$ onto $\mathbb{H}$.

Hyperplanes (Line)



a) Linear Decision Line.



b) Distance of a point from a line.

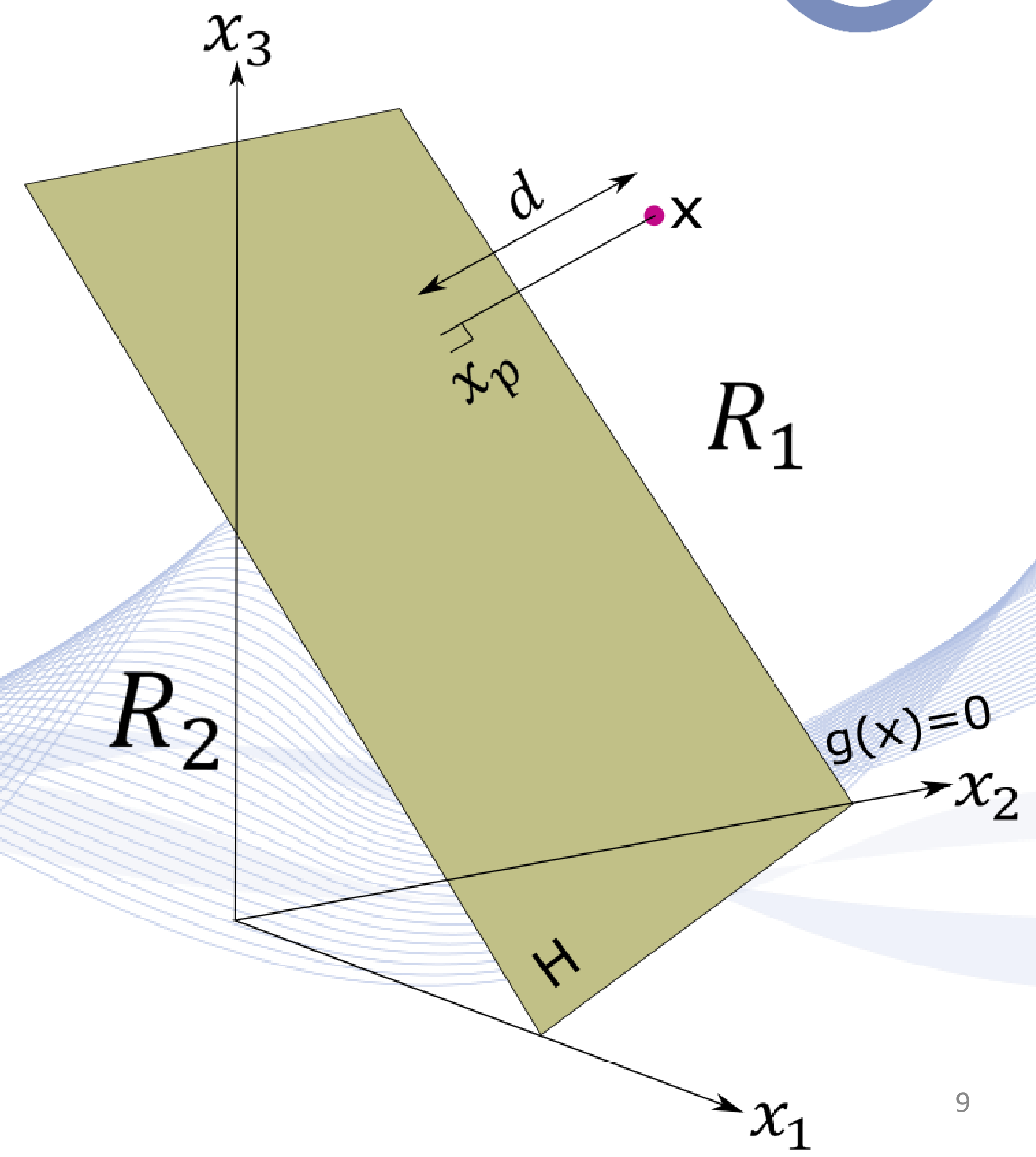
Hyperplanes (Plane)

The linear decision boundary $\mathbb{H}$:

$$g(\mathbf{x}) = \mathbf{w}^T\mathbf{x} + b = 0$$

separates the feature space into 2 half-spaces:

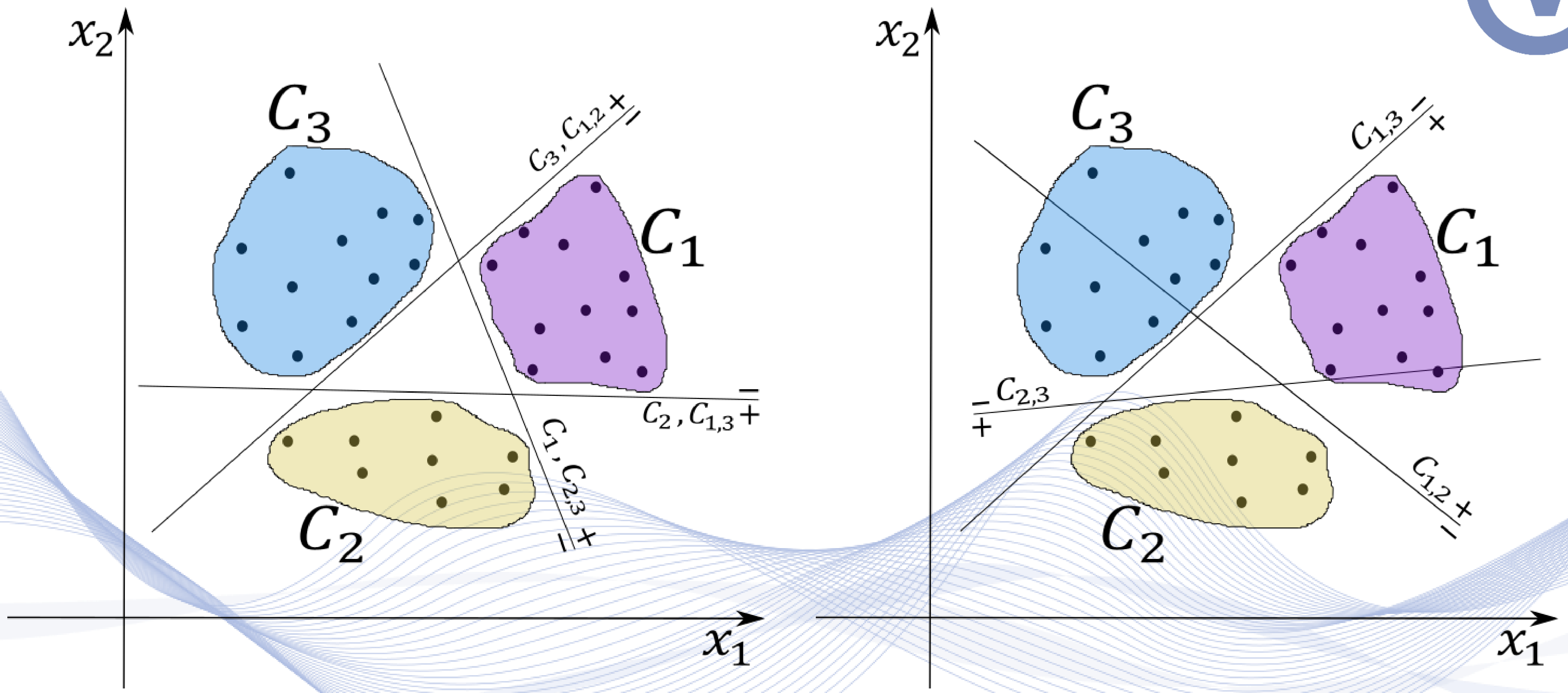
- $\mathcal{R}_1$ (where $g(\mathbf{x}) > 0$) and
- $\mathcal{R}_2$ (where $g(\mathbf{x}) < 0$).



Decision surfaces

Multiclass Classification $(m > 2)$**:**

- Binary hypothesis testing:
  - ***One class against all***: m binary hypotheses.
    - m decision surfaces must be found.
  - ***Pair-wise class comparisons*** (one-against-one):
    - $m(m-1)/2$ binary hypotheses
    - $m(m-1)/2$ decision surfaces must be found.



a) One-against-all multi-class classification; b) Pairwise multi-class classification.

Non-linear Decision Surfaces

- ***Linear discriminant function*** $g(\mathbf{x})$:

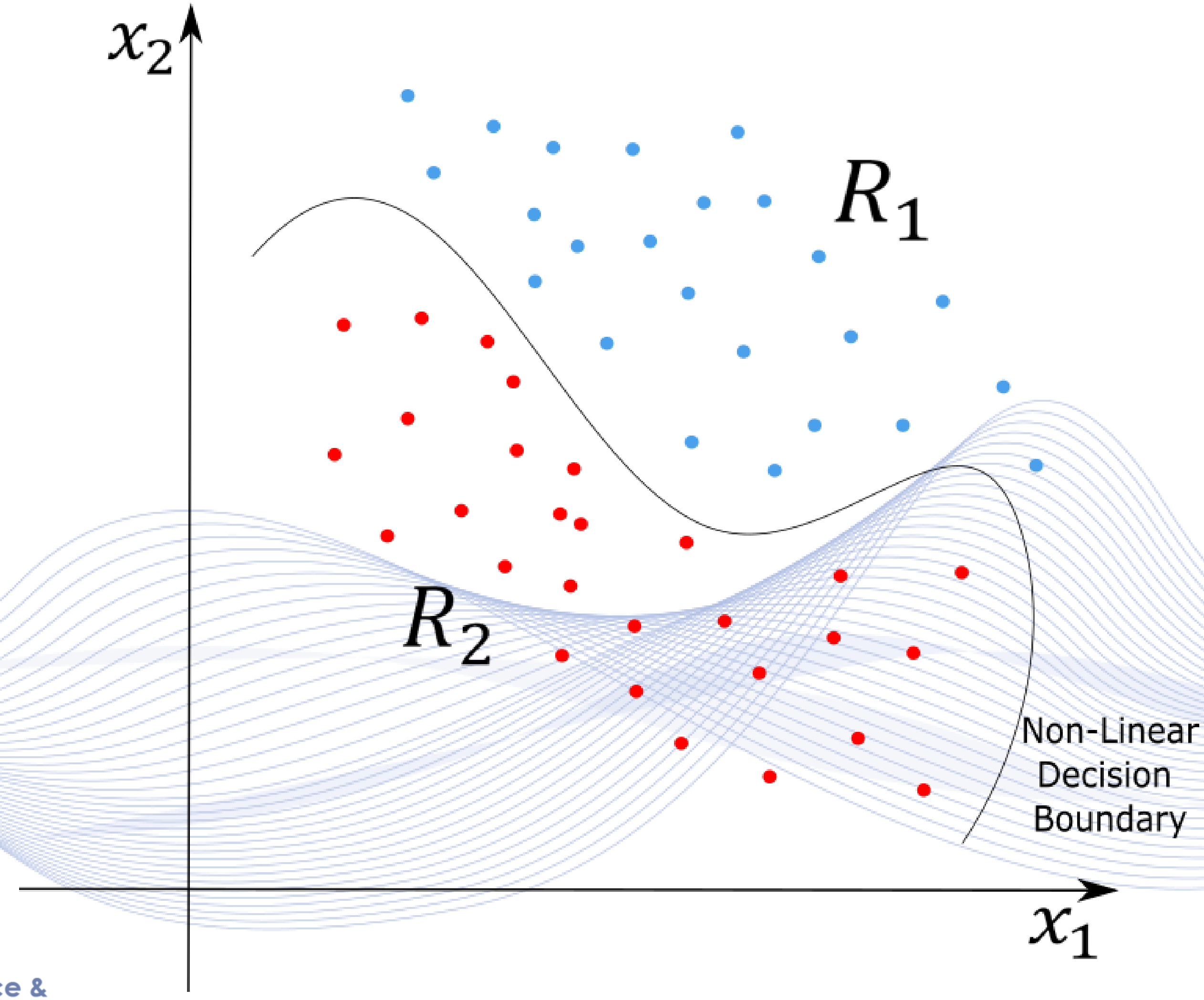
$$g(\mathbf{x}) = w_0 + \sum_{i=1}^{n} w_i x_i \,,$$

- coefficients w_i are the components of the weight vector $\mathbf{w}$.

- A general ***nonlinear discriminant function***: $g(\mathbf{x}) = f(\mathbf{x}; \mathbf{w})$ defines a decision surface $\mathbb{S}$.
- Distance of a point $\mathbf{x}$ from $\mathbb{S}$:

$$d(\mathbf{x}, \mathbb{S}) = \min_{\mathbf{z} \in \mathbb{S}} d(\mathbf{x}, \mathbf{z}).$$

Non-linear Decision Surfaces



Quadratic Decision Surfaces

- ***Polynomial discriminant function***:

$$g(\mathbf{x}) = w_0 + \sum_{i=1}^{n} w_i x_i + \sum_{i=1}^{n}\sum_{j=1}^{n} w_{ij} x_i x_j + \cdots + \sum_{i_1=1}^{n} \ldots \sum_{i_n=1}^{n} w_{i_1 \ldots i_n} x_{i_1} \ldots x_{i_n}$$

- The ***quadratic discriminant function*** is a second degree multivariate polynomial function:

$$g(\mathbf{x}) = w_0 + \sum_{i=1}^{n} w_i x_i + \sum_{i=1}^{n}\sum_{j=1}^{n} w_{ij} x_i x_j .$$

Quadratic Decision Surfaces

Special cases of ***quadratic decision surfaces***:

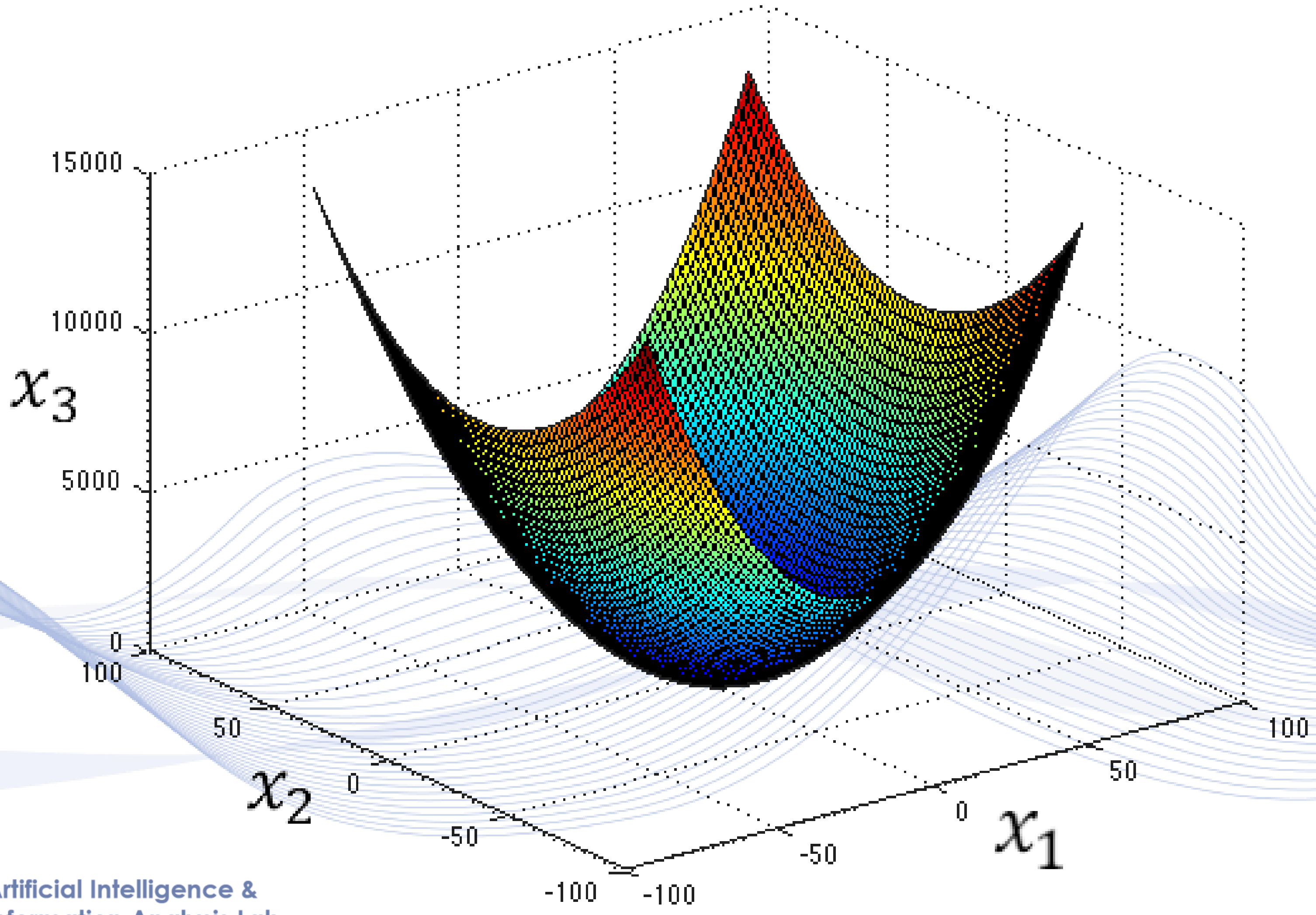
- ***Hypersphere*** equation having parameters $\mathbf{c}, r$ (hypersphere center, radius):

$$g(\mathbf{x}) = (\mathbf{x} - \mathbf{c})^T(\mathbf{x} - \mathbf{c}) - r^2.$$

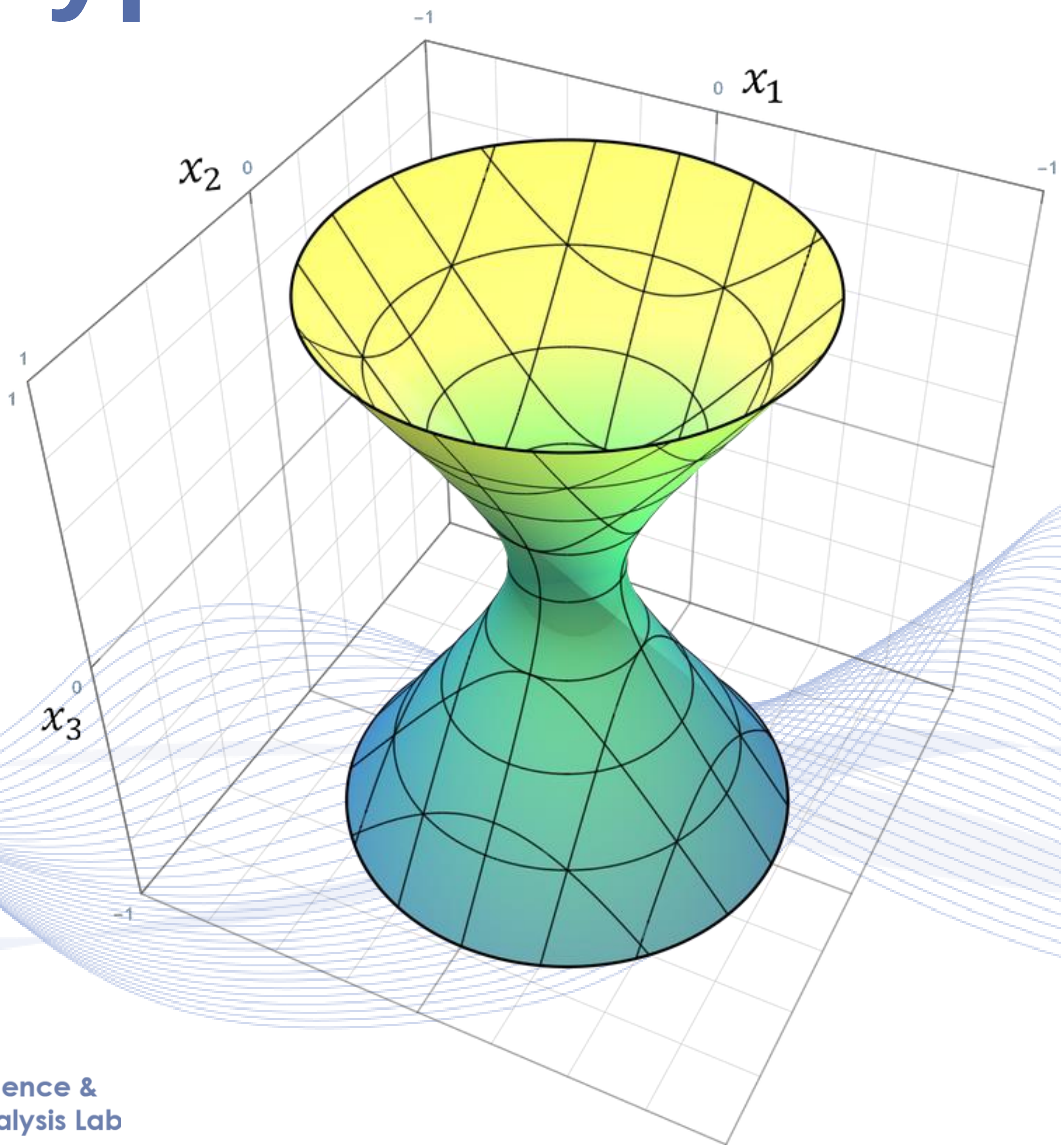
- ***Hyperellipsoid*** equation having parameters $\mathbf{A}, \mathbf{c}, r$:

$$g(\mathbf{x}) = (\mathbf{x} - \mathbf{c})^T\mathbf{A}(\mathbf{x} - \mathbf{c}) - r^2 .$$

3D ellipsoid



3D hyperboloid



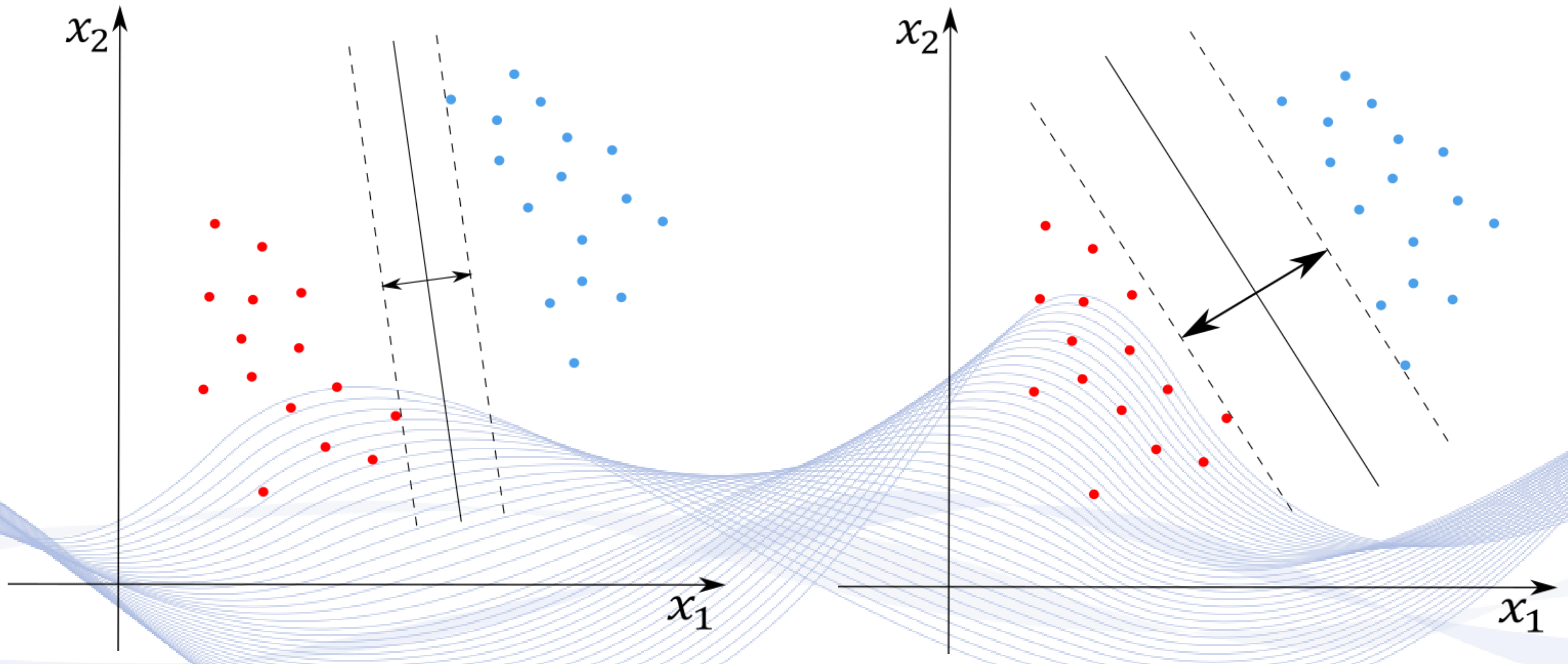
Introduction to SVMs

- Support Vector Machines is a **supervised** learning algorithm originally introduced in order to solve the **binary classification** problem.
- Its main objective is to find a ***hyperplane*** in the n-dimensional space (n**:** number of features) that separates the classes with the ***maximum margin*** (i.e., the maximum distance between samples of both classes).

Introduction to SVMs

- The derived hyperplane is a weighted, linear combination of the training set.
- ***Support Vectors*** are the training samples that lie ***closer*** to the hyperplane and have the ***biggest influence*** on its position and orientation.

Support Vector Machines

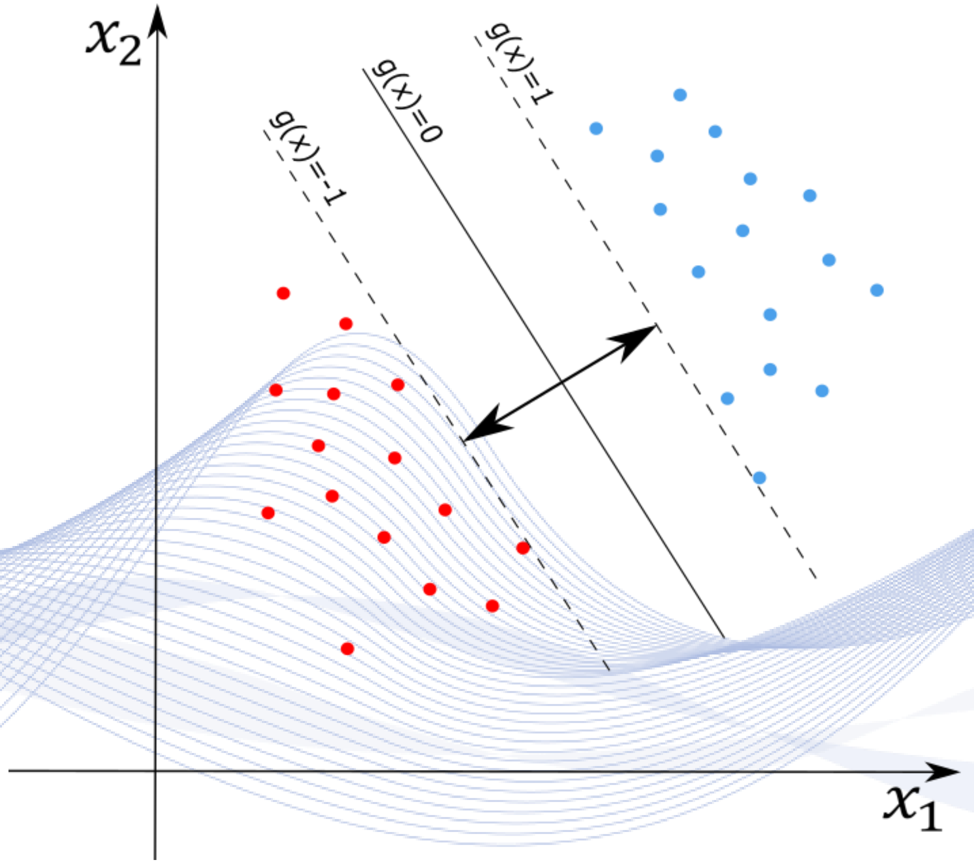


a) Small Margin. b) Optimal Margin.

Support Vector Machines

- As we have seen, we can use the function $g(\mathbf{x}) = \mathbf{w}^T\mathbf{x} + b$ to define the decision surface (***hyperplane***).
- Then we can divide the training data samples into 2 classes, $\mathcal{C}_1 = \{\mathbf{x}_+\}$ and $\mathcal{C}_2 = \{\mathbf{x}_-\}$ so that:

$$\mathbf{w}^T\mathbf{x}_+ + b \geq 1,$$
$$\mathbf{w}^T\mathbf{x}_- + b \leq -1.$$



Maximize Margin

The ***margin distance*** between $\mathbf{w}^T\mathbf{x} + b = -1$ and $\mathbf{w}^T\mathbf{x} + b = 1$ should be maximized.

- The distance between the decision boundary $\mathbf{w}^T\mathbf{x} + b = 0$ and one of the 2 lines that form the margin (e.g., $\mathbf{w}^T\mathbf{x} + b = 1$) is half of margin distance:

$$\frac{|\mathbf{w}^T\mathbf{x}+b|}{||\mathbf{w}||} = \frac{1}{||\mathbf{w}||},$$

- Thus, the ***margin distance*** is $\frac{2}{||\mathbf{w}||}$.
- In order to ***maximize the margin***, we need to ***minimize*** $||\boldsymbol{w}||$.

Support Vector Machines

- We introduce the parameter y_i, so that:

$$y_i = \begin{cases} 1, & \text{for } \mathbf{x}_+ \text{ samples}, \\ -1, & \text{for } \mathbf{x}_- \text{ samples}. \end{cases}$$

- Thus, ***in both cases***:

$$y_i(\mathbf{w}^T\mathbf{x}_i + b) \geq 1.$$

Primal SVM optimization problem

- The ***primal SVM optimization problem*** is defined as follows:

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

$$s.t.: \quad y_i(\mathbf{w}^T\mathbf{x}_i + b) \leq 1, \qquad i = 1, \ldots, N,$$

- $\mathbf{x}_i, i = 1, \ldots, N$: training samples.

Soft-margin SVM formulation

- The original SVM optimization criteria can never be met, if the data are not linearly separable.
- Therefore, ***soft-margin formulation*** is employed instead:

$$\min_{\mathbf{w},b,\boldsymbol{\xi}} \frac{1}{2}||\mathbf{w}||^2 + c\sum_{i=1}^{N} \xi_i ,$$

$$s.t.: \; y_i(\mathbf{w}^T\mathbf{x}_i + b) \leq 1 - \xi_i, \qquad i = 1, \dots, N.$$

$$\xi_i \geq 0, \qquad i = 1, \dots, N.$$

- $\xi_i, \; i = 1, \dots, N$ are the so-called ***slack variables***.

Artificial Intelligence & Information Analysis Lab

Lagrangian Dual Problem

- $c > 0$ is a hyperparameter that controls the amount of error allowed in the optimization problem.
- $c = 0$ denotes the hard-margin formulation.
- SVM optimization solution is equivalent to finding the ***saddle points*** of the Lagrangian:

$$J_p(\mathbf{w}, b, \boldsymbol{\xi}) = \frac{1}{2}||\mathbf{w}||^2 + c\sum_{i=1}^{N}\xi_i - \sum_{i=1}^{N} a_i[y_i(\mathbf{w}^T\mathbf{x}_i + b) - 1 + \xi_i] + \sum_{i=1}^{N}\beta_i\xi_i \,.$$

- a_i and β_i are ***Lagrange multipliers*** corresponding to the constraints of the primal problem.

Lagrangian Dual Problem

- According to ***Karush–Kuhn–Tucker*** (***KKT***) ***optimality conditions***, we zero the partial derivatives of J_p, with respect to $\mathbf{w}, b, \boldsymbol{\xi}$ and we obtain:

$$\frac{\partial J_p}{\partial w} = 0, \qquad \mathbf{w} = \sum_{i=1}^{N} a_i y_i \mathbf{x}_i \,,$$

$$\frac{\partial J_p}{\partial b} = 0, \qquad \sum_{i=1}^{N} a_i y_i = 0.$$

$$\frac{\partial J_p}{\partial \xi_i} = 0, \qquad \sum_{i=1}^{N} \beta_i = c - \sum_{i=1}^{N} \alpha_i \,.$$

Lagrangian Dual Problem

- By substituting back in J_p, a ***Quadratic Programming*** (***QP***) optimization problem is formed:

$$\max_{a_i} \sum_{i=1}^{N} a_i - \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} a_i a_j y_i y_j \mathbf{x}_i^T \mathbf{x}_j,$$

$$s.t.: 0 \leq a_i \leq c.$$

- This optimization problem can be solved using optimized ***QP-solvers***, e.g., ***Sequential Minimal Optimization*** (***SMO***).
- Note that most of the vector $\mathbf{a}$ entries will turn out to have 0 value.

SVM decision function

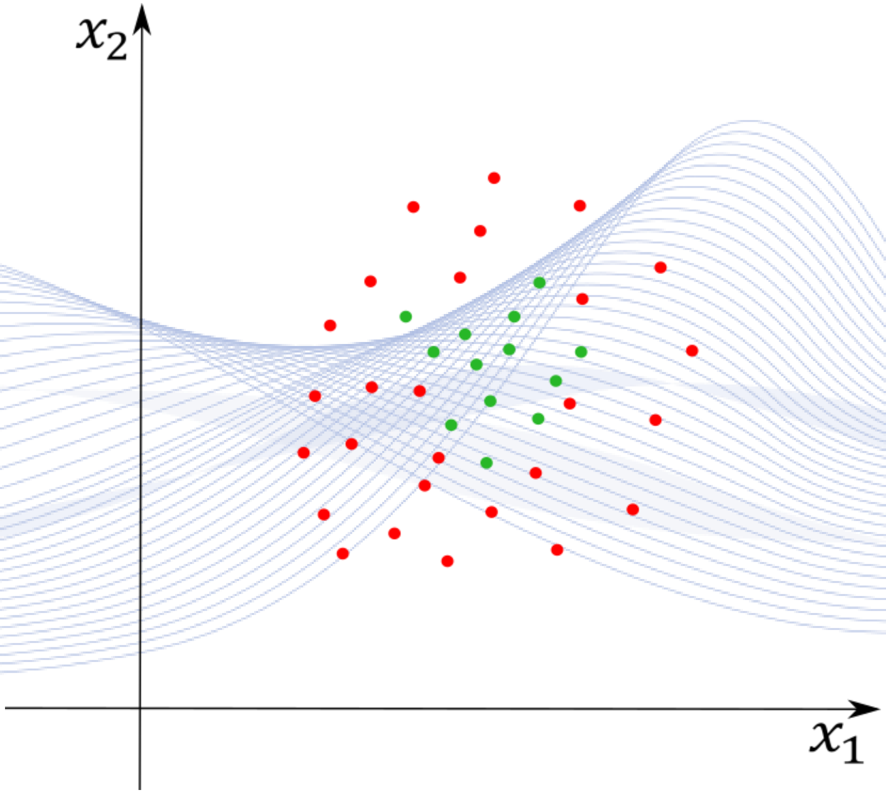
- The non-zero $\mathbf{a}$ entries will correspond to the ***support vectors***.
- Finally, in order to classify a test sample $\mathbf{x}$, we employ the following decision function:

$$g(\mathbf{x}) = \mathbf{w}^T\mathbf{x} + b = \sum_{i=1}^{N} a_i \, \mathbf{x}_i^T\mathbf{x} \, + b,$$

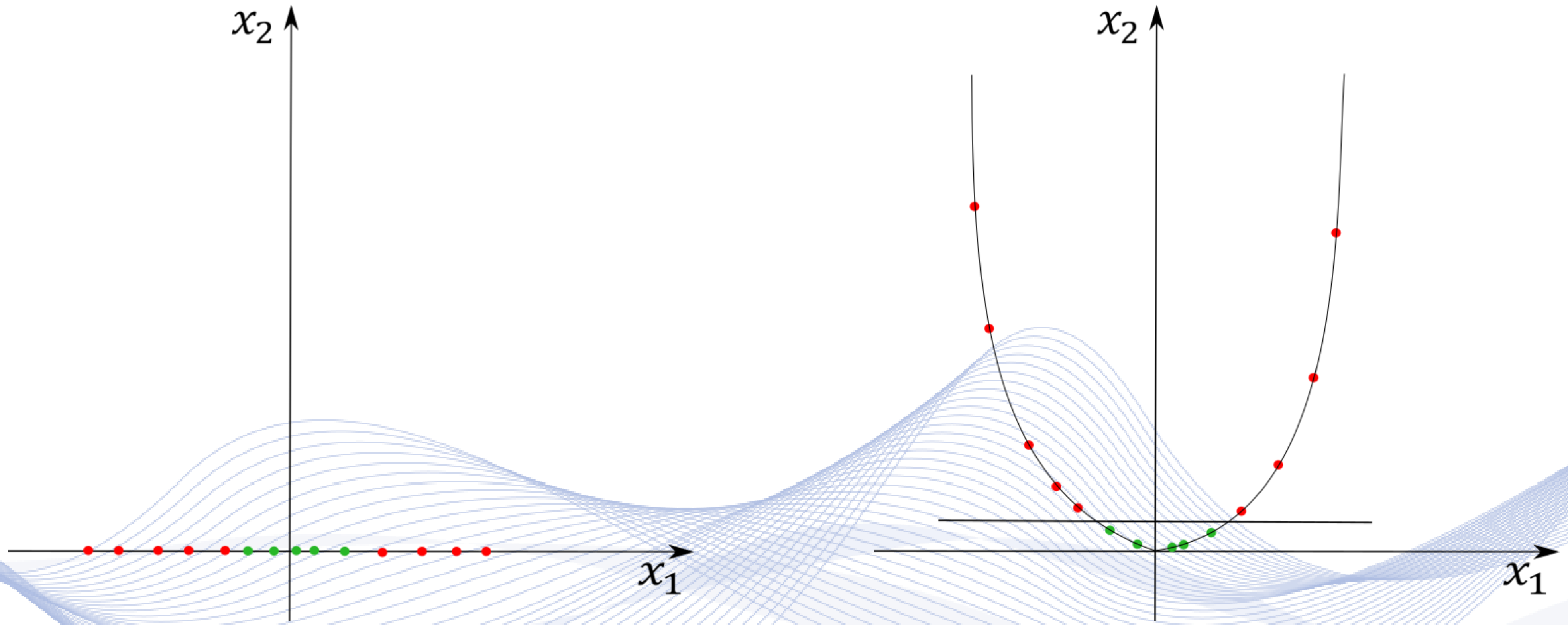
- $\mathbf{x}$ is classified to $\mathcal{C}_1$, if $g(\mathbf{x}) > 0$,
- $\mathbf{x}$ is classified to $\mathcal{C}_2$, otherwise.

Kernel SVMs

If we can not find an acceptable linear decision surface to separate the training data, we can generate a nonlinear one using the ***Kernel Trick***.



Kernel Trick (intuition)



a) data are not linearly separable in the 1D space.

b) If we move to 2D using $f(x) = x^2$, the data become linearly separable.

Artificial Intelligence & Information Analysis Lab

Kernel SVM problem

- In order to obtain non-linear hyperplanes, we assume a mapping function $\varphi(\cdot): \mathbb{R}^n \mapsto \mathcal{H}$ for the training data, where $\mathcal{H}$ is a space of high or even arbitrary dimensionality.

- The linear SVM optimization problem contains inner products $\mathbf{x}_i^T \mathbf{x}_j$ between the training samples.

Kernel SVM problem

- In the non-linear case, this inner product is replaced by any ***Reproducing Kernel Hilbert Space*** (***RKHS***) function:

$$\kappa\left(\mathbf{x}_i, \mathbf{x}_j\right) = \varphi(\mathbf{x}_\mathrm{i})^T \varphi(\mathbf{x}_\mathrm{j}),$$

that expresses data similarity in space $\mathcal{H}$.

- Common choices for $\kappa(\cdot,\cdot)$ include the ***Polynomial, Gaussian, Radial Basis Functions***.

Kernel SVM optimization

- In ***Kernel SVM optimization***, the Lagrangian function takes the following form:

$$\max_{a_i} \sum_{i=1}^{N} a_i - \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N} a_i a_j y_i y_j \kappa(\mathbf{x}_i, \mathbf{x}_j),$$

$$s.t.: \quad 0 \le a_i \le c.$$

- Finally, the decision function requires the same implicit mapping for the test sample as well:

$$g(\mathbf{x}) = \sum_{i=1}^{N} a_i \, \kappa(\mathbf{x}_i, \mathbf{x}) + b.$$

Bibliography

[HAY2009] S. Haykin, *Neural networks and learning machines*, Prentice Hall, 2009.

[BIS2006] C.M. Bishop, Pattern recognition and machine learning, Springer, 2006.

[THEO2011] S. Theodoridis, K. Koutroumbas, Pattern Recognition, Elsevier, 2011.

[ZUR1992] J.M. Zurada, *Introduction to artificial neural systems*. Vol. 8. St. Paul: West publishing company, 1992.

[YEG2009] Yegnanarayana, Bayya. *Artificial neural networks*. PHI Learning Pvt. Ltd., 2009.

Q & A

Thank you very much for your attention!

More material in
http://icarus.csd.auth.gr/cvml-web-lecture-series/

Contact: Prof. I. Pitas
pitas@csd.auth.gr