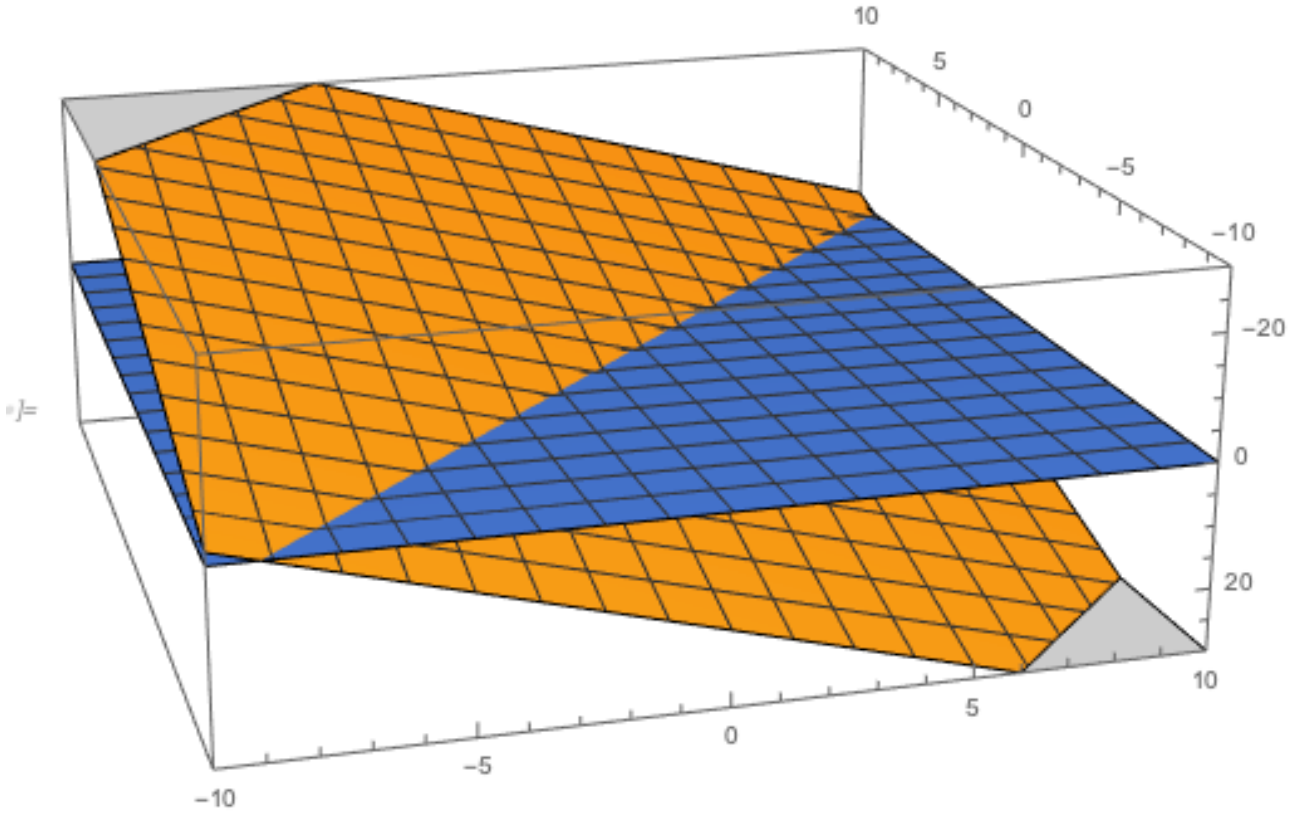
# Lecture 11:Support Vector Machines

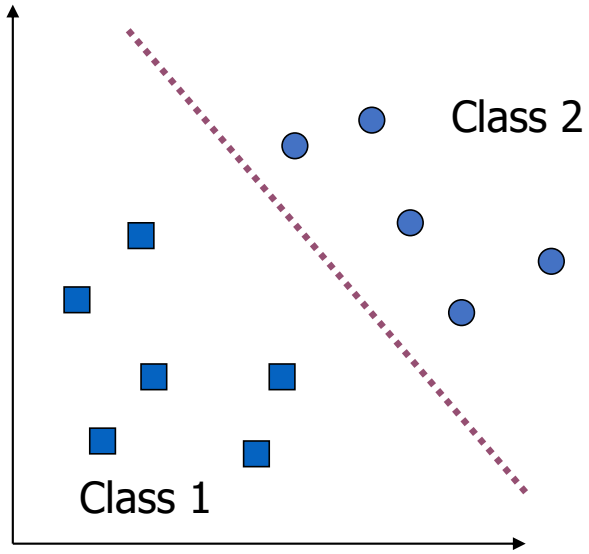
Andreas Wichert
Department of Computer Science and Engineering
Técnico Lisboa

# Linearly separable patterns

$w_0+w_1x_1+w_2x_2=0$

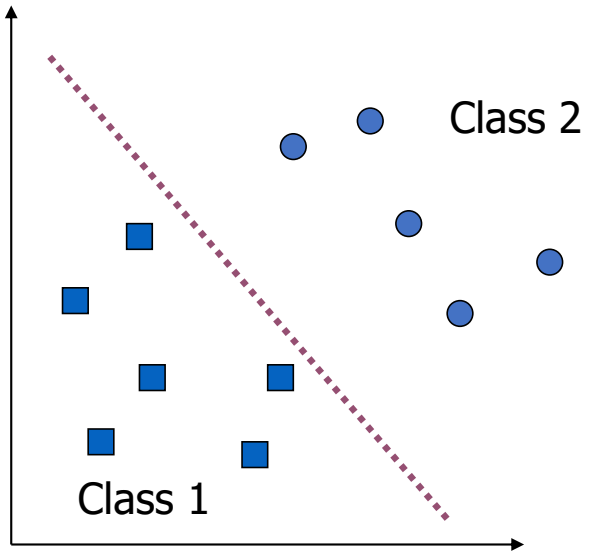


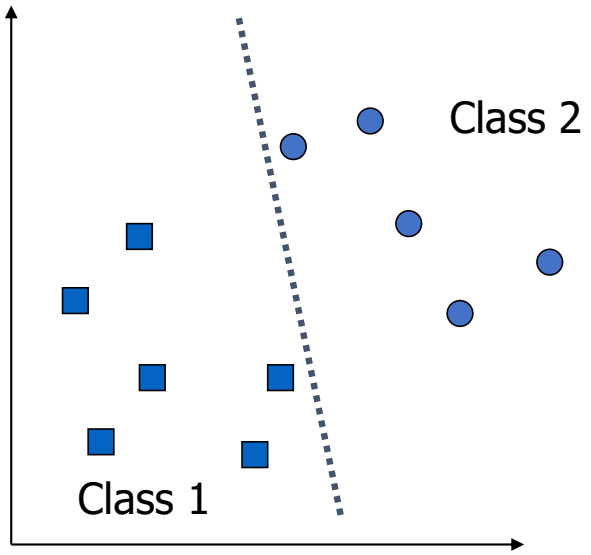
# Two Class Problem: Linear Separable Case



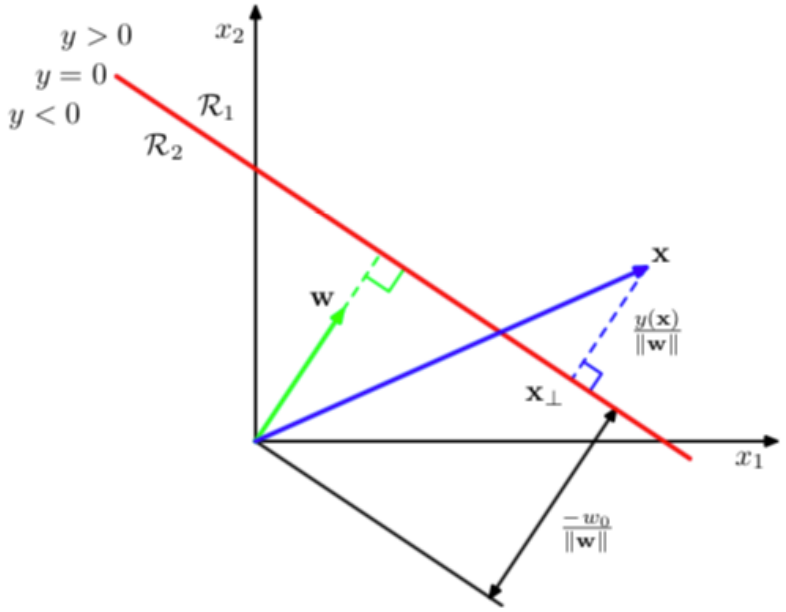

- Many decision boundaries can separate these two classes
- Which one should we choose?

# Example of Bad Decision Boundaries



# Discriminant Functions




$$net = y(\mathbf{x}) = \sum_{j=1}^{D} w_j \cdot x_j + w_0 = \langle \mathbf{w} | \mathbf{x} \rangle = \mathbf{w}^T \cdot \mathbf{x} + w_0$$

The projection of **x** on **w** defined as

$$proj_{\mathbf{w}}\mathbf{x} = \frac{\mathbf{w}^T \cdot \mathbf{x}}{\|\mathbf{w}\|}$$

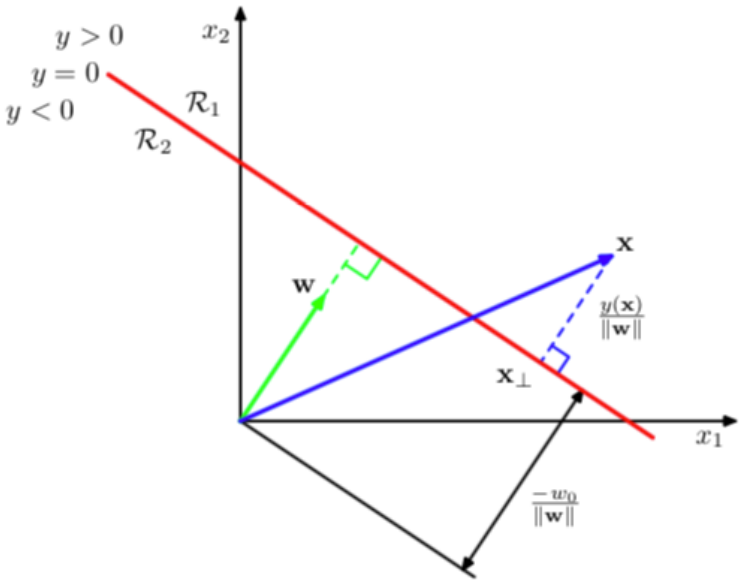
Since for the points on the decision surface

$$\mathbf{w}^T \cdot \mathbf{x} + w_0 = 0$$

$$\mathbf{w}^T \cdot \mathbf{x} = -w_0$$

we put it into the distance formula and we get

$$\frac{\mathbf{w}^T \cdot \mathbf{x}}{\|\mathbf{w}\|} = -\frac{w_0}{\|\mathbf{w}\|}$$



- *y(**x**)* gives the perpendicular signed distance of the point x from the decision surface. We represent ***x*** as

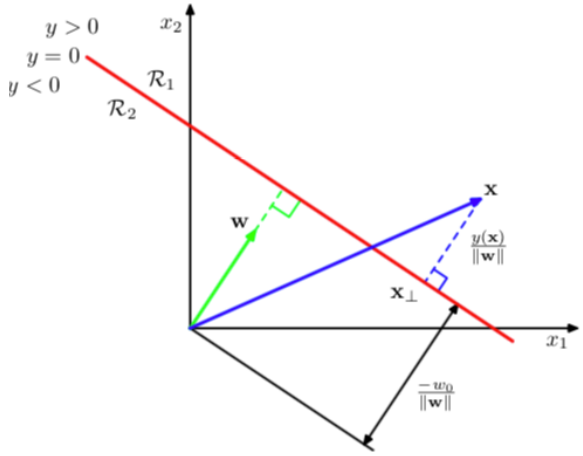
$$\mathbf{x} = \mathbf{x}_p + r \cdot \frac{\mathbf{w}}{\|\mathbf{w}\|}$$

Since $y(\mathbf{x}_p) = 0$

$$y(\mathbf{x}) = \mathbf{w}^T \cdot \mathbf{x}_p + w_0 + r \cdot \mathbf{w}^T \cdot \frac{\mathbf{w}}{\|\mathbf{w}\|} = 0 + r \cdot \|\mathbf{w}\|$$

or

$$r = \frac{y(\mathbf{x})}{\|\mathbf{w}\|}$$



# Optimal Hyperplane for Linear Separable Patterns

Given a training set $\{\mathbf{x}_i, t_i\}_{i=1}^{N}$, with $t_i \in \{-1, +1\}$ of linear separable patterns we have

$$\mathbf{w}^T \cdot \mathbf{x}_i + w_0 \geq 0, \quad for \;\; t_i = +1$$

$$\mathbf{w}^T \cdot \mathbf{x}_i + w_0 < 0, \quad for \;\; t_i = -1$$

with a hyperplane

$$\mathbf{w}^T \cdot \mathbf{x} + w_0 = \mathbf{w}^T \cdot \mathbf{x} + b = 0$$

We want to find the parameters $\mathbf{w}_{opt}$ and $b_{opt}$

$$\mathbf{w}_{opt}^T \cdot \mathbf{x}_i + b_{opt} \geq 1, \quad for \;\; t_i = +1$$

$$\mathbf{w}_{opt}^T \cdot \mathbf{x}_i + b_{opt} \leq -1, \quad for \;\; t_i = -1$$

with a hyperplane

$$\mathbf{w}_{opt}^{T} \cdot \mathbf{x}_i + b_{opt} = 0$$

Since the patterns are linearly separable we can always rescale $\mathbf{w}_{opt}$ and $b_{opt}$ correspondingly.

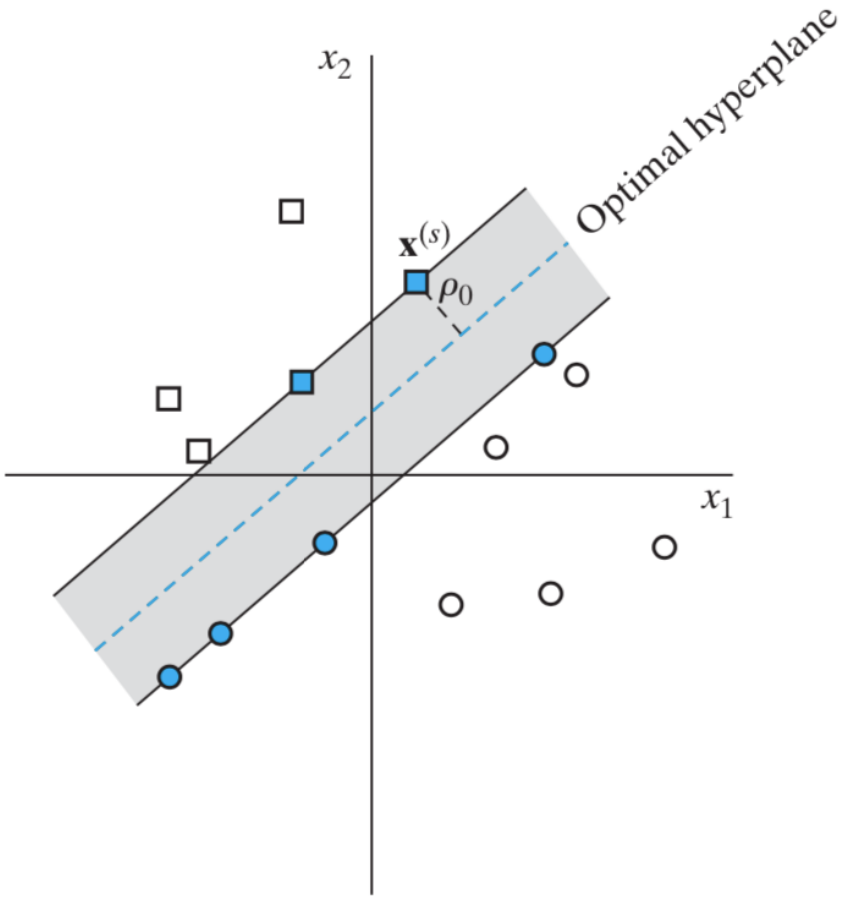
The data points$\{\mathbf{x}_i, t_i\}$ for which

$$\mathbf{w}_{opt}^{T} \cdot \mathbf{x}_i + b_{opt} = \pm 1$$

are called support vectors $\mathbf{x}^{(s)}$ , that why the algorithm is called support vector machine.

All the remaining examples in the training sample are completely irrelevant.

$$y(\mathbf{x}^{(s)}) = \mathbf{w}_{opt}^{T} \cdot \mathbf{x}^{(s)} + b_{opt} = \pm 1, \qquad for \;\; t^{(s)} = \pm 1$$


$x_2$
Optimal hyperplane
$\mathbf{x}^{(s)}$
$\rho_0$
$x_1$

All the remaining examples in the training sample are completely irrelevant.

$$y(\mathbf{x}^{(s)}) = \mathbf{w}_{opt}^T \cdot \mathbf{x}^{(s)} + b_{opt} = \pm 1, \qquad for \;\; t^{(s)} = \pm 1$$

and

$$y(\mathbf{x}^{(s)}) = +r \cdot \|\mathbf{w}_{opt}\| = \pm 1$$

or

$$r = \frac{y(\mathbf{x}^{(s)})}{\|\mathbf{w}_{opt}\|} = \begin{cases} \frac{1}{\|\mathbf{w}_{opt}\|} & \text{if} \quad t^{(s)} = +1 \\ -\frac{1}{\|\mathbf{w}_{opt}\|} & \text{if} \quad t^{(s)} = -1 \end{cases}$$

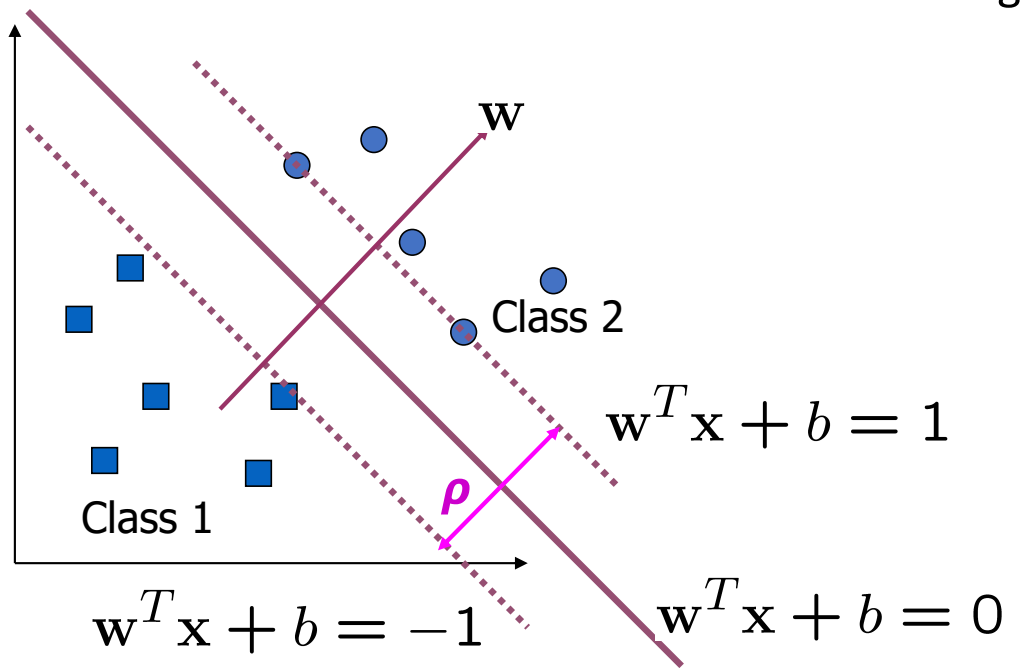
The optimum value $\rho$ of the margin of separation between the two classes that constitute the training sample

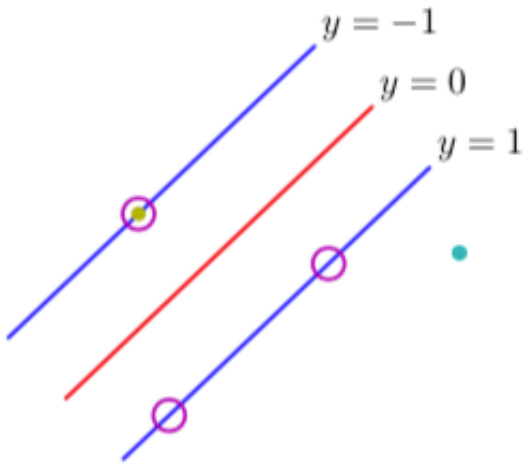
$$\rho = \frac{1}{\|\mathbf{w}_{opt}\|} + \left| -\frac{1}{\|\mathbf{w}_{opt}\|} \right| = \frac{2}{\|\mathbf{w}_{opt}\|} = 2 \cdot r$$

Maximising the margin of separation between binary classes is equivalent to minimising the Euclidean norm of the weight vector $\mathbf{w}$

# Good Decision Boundary: Margin Should Be Large

- The decision boundary should be as far away from the data of both classes as possible
  - We should maximize the margin, $\rho$






The optimal hyperplaneis unique in the sense that the optimum weight vector $\mathbf{w}_{opt}$ provides the maximum possible separation between positive and negative examples.

# Quadratic Optimization for Finding the Optimal Hyperplane

For a training set $\{\mathbf{x}_i, t_i\}_{i=1}^{N}$, with $t_i \in \{-1, +1\}$ of linear separable patterns we want find the optimum values of the weight vector $\mathbf{w}$ and bias $b$ such that they satisfy the constraints

$$\mathbf{w}^T \cdot \mathbf{x}_i + b \geq 1, \qquad for \quad t_i = +1$$

$$\mathbf{w}^T \cdot \mathbf{x}_i + b \leq -1, \qquad for \quad t_i = -1$$

expressed in a single line as

$$t_i \cdot \left(\mathbf{w}^T \cdot \mathbf{x}_i + b\right) \geq 1$$

and the weight vector $\mathbf{w}$ minimizes the cost function

$$\Phi(\mathbf{w}) = \frac{1}{2} \cdot \mathbf{w}^T \mathbf{w} = \frac{1}{2} \cdot \|\mathbf{w}\|^2$$

with scaling factor $\frac{1}{2}$ that is introduced for convenience.

This constrained optimisation problem is called the primal problem and it is described as

- The cost function $\Phi(\mathbf{w})$ is a convex function of $\mathbf{w}$
- The constraints are linear in $\mathbf{w}$

We may solve the constrained optimisation problem by using the method of Lagrange multipliers with

$$J(\mathbf{w}, b, \alpha) = \frac{1}{2} \cdot \mathbf{w}^T \mathbf{w} - \sum_{i=1}^{N} \alpha_i \cdot \left( t_i \cdot \left( \mathbf{w}^T \cdot \mathbf{x}_i + b \right) - 1 \right)$$

where $\alpha_i$ are called Lagrange multipliers.

Lagrange multipliers with

$$J(\mathbf{w}, b, \alpha) = \frac{1}{2} \cdot \mathbf{w}^T \mathbf{w} - \sum_{i=1}^{N} \alpha_i \cdot \left( t_i \cdot \left( \mathbf{w}^T \cdot \mathbf{x}_i + b \right) - 1 \right)$$

where $\alpha_i$ are called Lagrange multipliers.

The solution to the constrained optimisation problem is determined by the saddle point of the Lagrangian function $J(\mathbf{w}, b, \alpha)$,
The saddle point has to be minimised with respect to $\mathbf{w}$ and $b$ and maximised with respect to $\alpha$

We get two conditions

$$\frac{\partial(\mathbf{w}, b, \alpha)}{\partial \mathbf{w}} = \mathbf{0}, \qquad \frac{\partial(\mathbf{w}, b, \alpha)}{\partial b} = 0$$

leading to

$$\mathbf{w} = \sum_{i=1}^{N} \alpha_i \cdot t_i \cdot \mathbf{x}_i, \qquad \sum_{i=1}^{N} \alpha_i \cdot t_i = 0$$

This solution is unique by virtue of the convexity of the Lagrangian, but no with respect of the Lagrange multipliers $\alpha_i$.

The constraints

$$t_i \cdot \left(\mathbf{w}^T \cdot \mathbf{x}_i + b\right) - 1 \neq 0$$

that are not satisfied as equalities, the corresponding multiplier $\alpha_i$ must be zero, the condition must be satisfied

$$\alpha_i \cdot \left(t_i \cdot \left(\mathbf{w}^T \cdot \mathbf{x}_i + b\right) - 1\right) = 0$$

(Karush-Kuhn-Tucker conditions)

- It is possible to construct another problem called the dual problem that has the same optimal value as the primal problem
- If the primal problem has an optimal solution, the dual problem also has an optimal solution, and the corresponding optimal values are equal.
- In order for $\mathbf{w}_{opt}$ to be an optimal primal solution and $\alpha_{opt}$ to be an optimal dual solution, it is necessary and sufficient that $\mathbf{w}_{opt}$ is feasible for the primal problem, and

$$\Phi(\mathbf{w}_{opt}) = J(\mathbf{w}_{opt}, b_{opt}, \alpha_{opt}) = \min_{\mathbf{w}} J(\mathbf{w}, b, \alpha)$$

$$J(\mathbf{w}, b, \alpha) = \frac{1}{2} \cdot \mathbf{w}^T \mathbf{w} - \sum_{i=1}^{N} \alpha_i \cdot \left(t_i \cdot \left(\mathbf{w}^T \cdot \mathbf{x}_i + b\right) - 1\right)$$

We expand

$$J(\mathbf{w}, b, \alpha) = \frac{1}{2} \cdot \mathbf{w}^T \mathbf{w} - \sum_{i=1}^{N} \alpha_i \cdot t_i \cdot \mathbf{w}^T \mathbf{x}_i - b \cdot \sum_{i=1}^{N} \alpha_i \cdot t_i + \sum_{i=1}^{N} \alpha_i$$

we have because of

$$\mathbf{w} = \sum_{i=1}^{N} \alpha_i \cdot t_i \cdot \mathbf{x}_i$$

$$\mathbf{w}^T \mathbf{w} = \sum_{i=1}^{N} \alpha_i \cdot t_i \cdot \mathbf{w}^T \mathbf{x}_i = \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_i \cdot \alpha_j \cdot t_i \cdot t_j \cdot \mathbf{x}_i^T \mathbf{x}_j$$

$$J(\mathbf{w}, b, \alpha) = \frac{1}{2} \cdot \sum_{i=1}^{N} \alpha_i \cdot t_i \cdot \mathbf{w}^T \mathbf{x}_i - \sum_{i=1}^{N} \alpha_i \cdot t_i \cdot \mathbf{w}^T \mathbf{x}_i - b \cdot \sum_{i=1}^{N} \alpha_i \cdot t_i + \sum_{i=1}^{N} \alpha_i$$

$$J(\mathbf{w}, b, \alpha) = -b \cdot \sum_{i=1}^{N} \alpha_i \cdot t_i + \sum_{i=1}^{N} \alpha_i - \frac{1}{2} \cdot \sum_{i=1}^{N} \alpha_i \cdot t_i \cdot \mathbf{w}^T \mathbf{x}_i$$

$$J(\mathbf{w}, b, \alpha) = -b \cdot \sum_{i=1}^{N} \alpha_i \cdot t_i + \sum_{i=1}^{N} \alpha_i - \frac{1}{2} \cdot \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_i \cdot \alpha_j \cdot t_i \cdot t_j \cdot \mathbf{x}_i^T \mathbf{x}_j$$

Setting

$$Q(\alpha) = J(\mathbf{w}, b, \alpha)$$

with

$$\sum_{i=1}^{N} \alpha_i \cdot t_i = 0$$

we reformulate the constraint.

$$Q(\alpha) = \sum_{i=1}^{N} \alpha_i - \frac{1}{2} \cdot \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_i \cdot \alpha_j \cdot t_i \cdot t_j \cdot \mathbf{x}_i^T \mathbf{x}_j$$

with $\alpha_i$ being non negative.

# Dual Problem

Given the training sample $\{\mathbf{x}_i, t_i\}_{i=1}^N$, with $t_i \in \{-1, +1\}$ of linear separable patterns find the Lagrange multipliers $\{\alpha_i\}_{i=1}^N$ that maximise

$$Q(\alpha) = \sum_{i=1}^{N} \alpha_i - \frac{1}{2} \cdot \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_i \cdot \alpha_j \cdot t_i \cdot t_j \cdot \mathbf{x}_i^T \mathbf{x}_j$$

subject to constraints

$$\sum_{i=1}^{N} \alpha_i \cdot t_i = 0$$

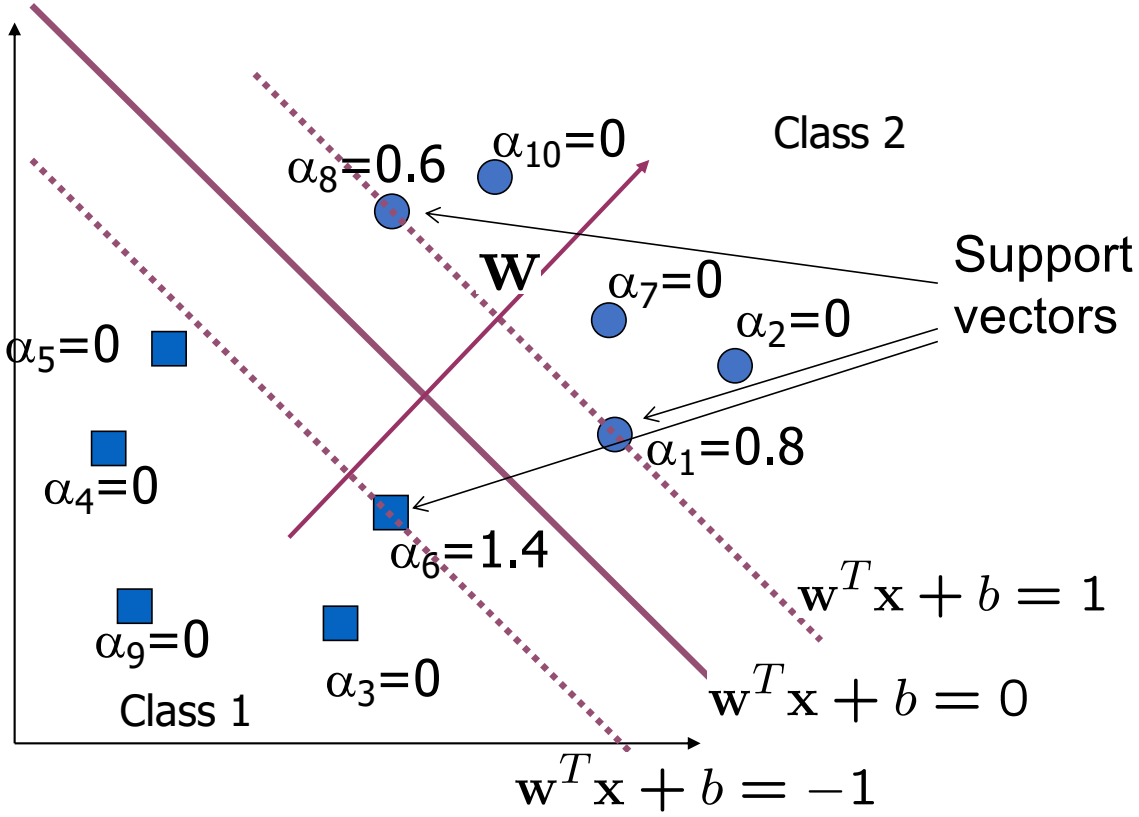
$$\alpha_i \geq 0, \quad i = 1, 2, \cdots, N$$

The dual problem is cast entirely in terms of the training data.

The support vectors constitute a subset of the training sample, which means that the solution vector is sparse

The dual problem is satisfied with the inequality sign for all the support vectors for which the $\alpha$'s are nonzero, and with the equality sign for all the other data points in the training sample, for which the $\alpha$'s are all zero.

# A Geometrical Interpretation



Having determined the optimum Lagrange multipliers

$$\alpha_{opt,i}$$

we may recover

$$\mathbf{w}_{opt} = \sum_{i=1}^{N} \alpha_{opt,i} \cdot t_i \cdot \mathbf{x}_i,$$

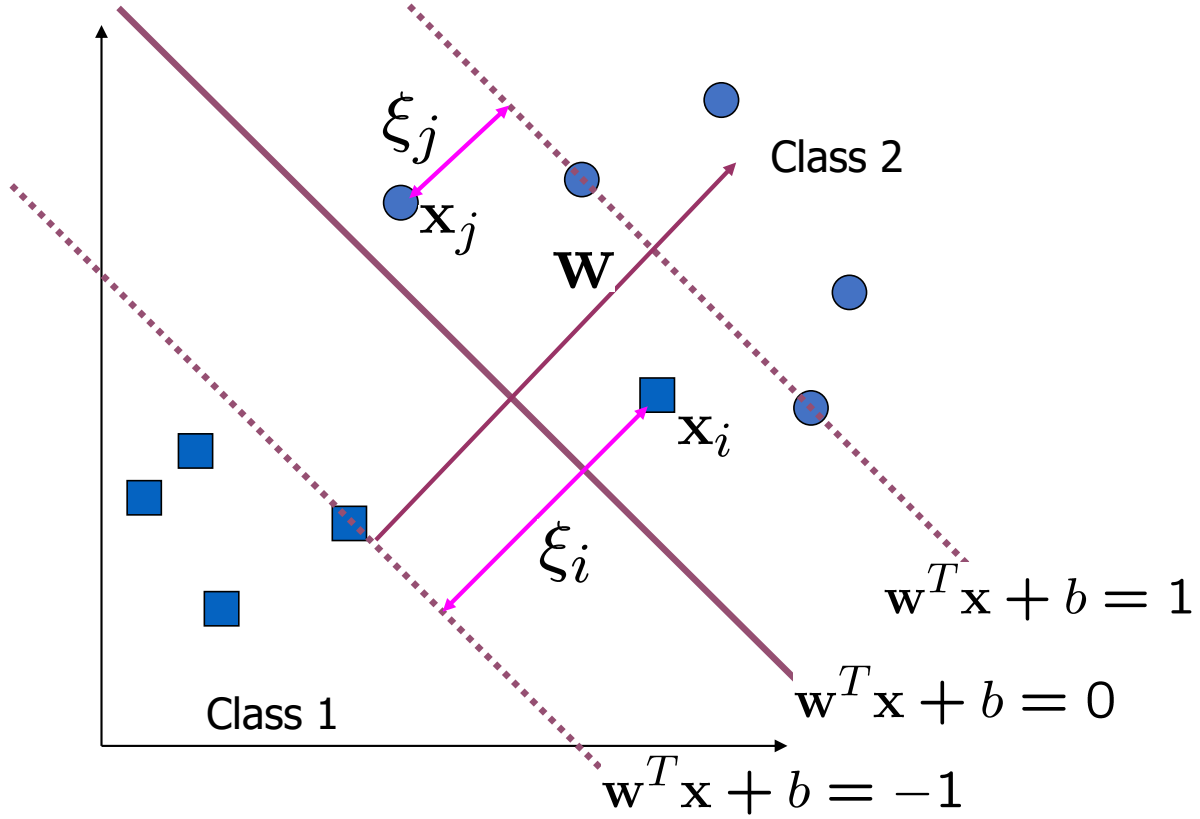
and

$$b_{opt} = 1 - \mathbf{w}_{opt}^T \cdot \mathbf{x}^{(s)}, \quad for \;\; t^{(s)} = 1$$

$$b_{opt} = 1 - \sum_{i=1}^{N} \alpha_{opt,i} \cdot t_i \cdot \mathbf{x}_i \cdot \mathbf{x}^{(s)}, \quad for \;\; t^{(s)} = 1$$

# How About Not Linearly Separable

- We allow "error" $\xi_i$ in classification



# Optimal Hyperplane for Nonseparable Patterns

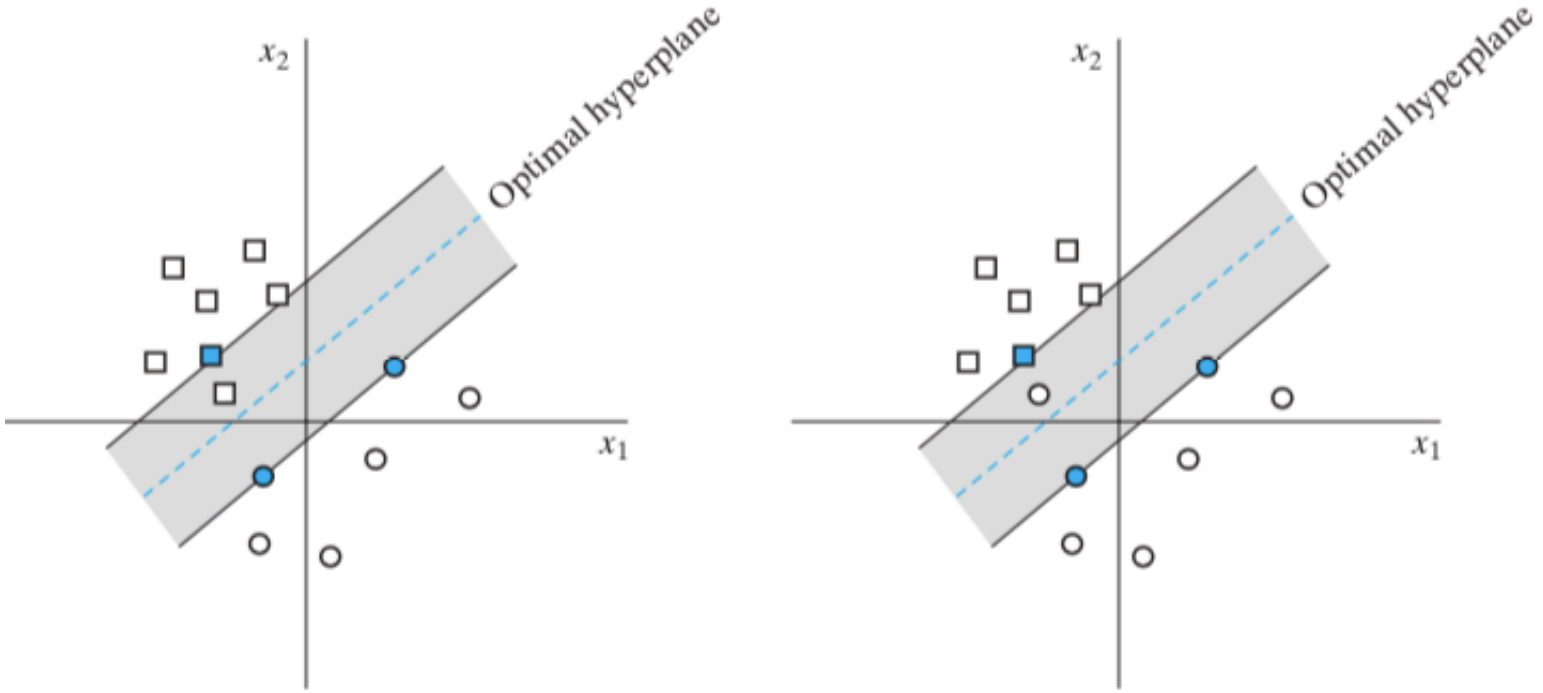
$$t_i \cdot \left(\mathbf{w}^T \cdot \mathbf{x}_i + b\right) \geq 1 - \xi_i, \qquad for \;\; i = 1, 2, \cdots, N$$

$\xi_i$ are called the slack variables; they measure the deviation of a data point from the ideal condition of pattern separability

For $0 < \xi_i \leq 1$ the data point falls inside the region of separation, but on the correct side of the decision surface,

For $\xi_i > 1$ falls on the wrong side of the separating hyperplane

Our goal is to find a separating hyperplane for which the misclassification error, averaged over the training sample, is minimized


$x_2$
Optimal hyperplane
$x_1$
$x_2$
Optimal hyperplane
$x_1$

We minimize

$$\Phi(\xi) = \sum_{i=1}^{N} I(\xi - 1)$$

with respect to the weight vector $\mathbf{w}$ as before. The indicator function $I(\xi)$ is defined as

$$I(\xi) = \begin{cases} 0 & \text{if } \xi \leq 0 \\ 1 & \text{if } \xi > 0 \end{cases}$$

To simplify we approximate the function by

$$\Phi(\xi) = \sum_{i=1}^{N} \xi_i$$

We simplify more by

$$\Phi(\xi) = \frac{1}{2} \cdot \mathbf{w}^T \cdot \mathbf{w} + C \cdot \sum_{i=1}^{N} \xi_i$$

We simplify more by

$$\Phi(\xi) = \frac{1}{2} \cdot \mathbf{w}^T \cdot \mathbf{w} + C \cdot \sum_{i=1}^{N} \xi_i$$

with $C$ controls the tradeoff between complexity of the machine and the number of non separable points

$C$ may be viewed as an inverse regularisation parameter

Large value indicates high confidence in the quality of the training sample

Small value indicates noisy training set, and less emphasis should therefore be placed on it.

Given the training sample $\{\mathbf{x}_i, t_i\}_{i=1}^N$, with $t_i \in \{-1, +1\}$ of linear separable patterns find find the parameters $\mathbf{w}$ and $b$ with the constraint

$$t_i \cdot \left(\mathbf{w}^T \cdot \mathbf{x}_i + b\right) \geq 1 - \xi_i, \qquad for \quad i = 1, 2, \cdots, N$$

$$\xi_i \geq 0$$

and such that the weight vector $\mathbf{w}$ and the slack variables $\xi_i$ minimise the cost functional

$$\Phi(\xi) = \frac{1}{2} \cdot \mathbf{w}^T \cdot \mathbf{w} + C \cdot \sum_{i=1}^{N} \xi_i$$

where $C$ is a user specified positive parameter

# Dual Problem

Given the training sample $\{\mathbf{x}_i, t_i\}_{i=1}^N$, with $t_i \in \{-1, +1\}$ of linear separable patterns find the Lagrange multipliers $\{\alpha_i\}_{i=1}^N$ that maximise

$$Q(\alpha) = \sum_{i=1}^{N} \alpha_i - \frac{1}{2} \cdot \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_i \cdot \alpha_j \cdot t_i \cdot t_j \cdot \mathbf{x}_i^T \mathbf{x}_j$$

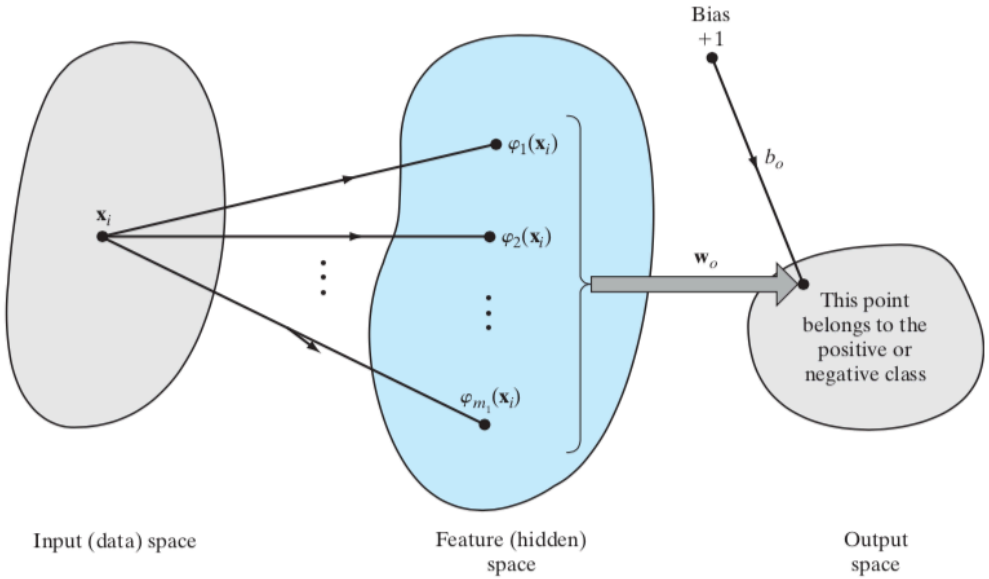
subject to constraints

$$\sum_{i=1}^{N} \alpha_i \cdot t_i = 0$$

$$0 \leq \alpha_i \leq C, \quad i = 1, 2, \cdots, N$$

where $C$ is a user specified positive parameter

# Philosophy of a Support Vector Machine




- Nonlinear mapping of an input vector into a high-dimensional feature space that is hidden from both the input and output
- Construction of an optimal hyperplane for separating the features that were discovered before

# Support Vector Machine as a Kernel Machine

Let $\mathbf{x}$ be a vector from the input space of the dimension $D$

Let $\{\phi_j(\mathbf{x})\}_{j=1}^{\infty}$ be a set of nonlinear functions, from $D$ dimension to infinite dimension.

The hyperplane is defined as

$$\sum_{j=1}^{\infty} w_j \cdot \phi_j(\mathbf{x}) = 0$$

Using matrix noation

$$\mathbf{w}^T \cdot \Phi(\mathbf{x}) = 0$$

We can represent now the weights vector as

$$\mathbf{w} = \sum_{i=1}^{N_s} \alpha_i \cdot t_i \cdot \Phi(\mathbf{x}_i)$$

wit $N_s$ being the number of support vectors with the feature vector

$$\Phi(\mathbf{x}_i) = \begin{pmatrix} \phi_1(\mathbf{x}_i) \\ \phi_2(\mathbf{x}_i) \\ \vdots \end{pmatrix}$$

We get the decision surface as

$$\sum_{i=1}^{N_s} \alpha_i \cdot t_i \cdot \Phi^T(\mathbf{x}_i)\Phi(\mathbf{x}) = 0$$

We see that $\Phi^T(\mathbf{x}_i)\Phi(\mathbf{x})$ represents an inner product $\langle \Phi(\mathbf{x}_i) | \Phi(\mathbf{x}) \rangle$

$$k(\mathbf{x}, \mathbf{x}_i) = \Phi^T(\mathbf{x}_i)\Phi(\mathbf{x}) = \langle \Phi(\mathbf{x}_i) | \Phi(\mathbf{x}) \rangle$$

$$k(\mathbf{x}, \mathbf{x}_i) = \langle \Phi(\mathbf{x}_i) | \Phi(\mathbf{x}) \rangle = \sum_{j=1}^{\infty} \phi(\mathbf{x}_i) \cdot \phi(\mathbf{x})$$

with $k(\mathbf{x}, \mathbf{x}_i)$ being the inner-product kernel

Property 1: The function $k(\mathbf{x}, \mathbf{x}_i)$ is symmetric about the center point $\mathbf{x}_i$ that is,

$$k(\mathbf{x}, \mathbf{x}_i) = k(\mathbf{x}_i, \mathbf{x})$$

and it attains its maximum value at the point $\mathbf{x} = \mathbf{x}_i$

Property 2: The total volume under the surface of the function $k(\mathbf{x}, \mathbf{x}_i)$ is a constant.

# Kernel Trick

Specifying the kernel $k(\mathbf{x}, \mathbf{x}_i)$ is sufficient, we need never explicitly compute the weight vector $\mathbf{w}_{opt}$

$$\sum_{j=1}^{\infty} w_j \cdot \phi_j(\mathbf{x}) = \sum_{i=1}^{N_s} \alpha_i \cdot t_i \cdot k(\mathbf{x}, \mathbf{x}_i) = 0$$

Even though we assumed that the feature space could be of infinite dimensionality, t, defining the optimal hyperplane, consists of a finite number of terms that is equal to the number of training patterns used in the classifier.

The support vector machine is also referred to as a kernel machine.

For pattern classification, the machine is parameterised by an N-dimensional vector whose $i$th term is defined by the product

$$\alpha_i \cdot t_i$$

We may view $k(\mathbf{x}_i, \mathbf{x}_j)$ as the $ij$-th element of the $N \times N$ matrix

$$K = \begin{pmatrix} k(\mathbf{x}_1, \mathbf{x}_1) & k(\mathbf{x}_1, \mathbf{x}_2) & k(\mathbf{x}_1, \mathbf{x}_3) & \cdots & k(\mathbf{x}_1, \mathbf{x}_N) \\ k(\mathbf{x}_2, \mathbf{x}_1) & k(\mathbf{x}_2, \mathbf{x}_2) & k(\mathbf{x}_2, \mathbf{x}_3) & \cdots & k(\mathbf{x}_2, \mathbf{x}_N) \\ \vdots & \vdots & \ddots & \vdots & \\ k(\mathbf{x}_N, \mathbf{x}_1) & k(\mathbf{x}_N, \mathbf{x}_2) & k(\mathbf{x}_N, \mathbf{x}_3) & \cdots & k(\mathbf{x}_N, \mathbf{x}_N) \end{pmatrix}$$

The matrix $K$ is a nonnegative definite matrix called the kernel matrix; it is also referred to simply as the Gram.

$$\mathbf{a}^T K \mathbf{a} \geq 0$$

# Mercer's theorem

James Mercer (1883-1932)

- Kernel $k(\boldsymbol{x}_i, \boldsymbol{x}_j)$ needs to satisfy a technical condition, Mercer condition specified by Mercer's theorem
- Mercer's theorem tells us only whether a candidate kernel is actually an inner-product kernel in some space and therefore admissible for use in a support vector machine.
- It says nothing about how to construct the functions

# Design of Support Vector Machine

Given the training sample $\{\mathbf{x}_i, t_i\}_{i=1}^N$, with $t_i \in \{-1, +1\}$ of linear separable patterns find the Lagrange multipliers $\{\alpha_i\}_{i=1}^N$ that maximise

$$Q(\alpha) = \sum_{i=1}^{N} \alpha_i - \frac{1}{2} \cdot \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_i \cdot \alpha_j \cdot t_i \cdot t_j \cdot k(\mathbf{x}_i, \mathbf{x}_j)$$

subject to constraints

$$\sum_{i=1}^{N} \alpha_i \cdot t_i = 0$$

$$0 \leq \alpha_i \leq C, \quad i = 1, 2, \cdots, N$$

where $C$ is a user specified positive parameter

In order to classify data points, we evaluate the sign of $\mathbf{w}^T \cdot \Phi(\mathbf{x}) + b$

$$o = sgn\left(\mathbf{w}^T \cdot \Phi(\mathbf{x}) + b\right)$$

by substituting for $\mathbf{w}$

$$\mathbf{w} = \sum_{i=1}^{N_s} \alpha_i \cdot t_i \cdot \Phi(\mathbf{x}_i)$$

in terms of parameters $\alpha_i$

$$o = sgn\left(\sum_{i=1}^{N_s} \alpha_i \cdot t_i \cdot \Phi(\mathbf{x}_i)^T \Phi(\mathbf{x}) + b\right)$$

and the kernel function

$$o = sgn\left(\sum_{i=1}^{N_s} \alpha_i \cdot t_i \cdot k(\mathbf{x}, \mathbf{x}_i) + b\right) = sgn\left(\sum_{i=1}^{N} \alpha_i \cdot t_i \cdot k(\mathbf{x}, \mathbf{x}_i) + b\right)$$

with the bias

$$b = \frac{1}{N_s} \sum_{i=1}^{N_s} \left(t_i - \sum_{j=1}^{N_s} \alpha_j \cdot t_j \cdot k(\mathbf{x}_i, \mathbf{x}_j)\right)$$

# Classify Data Points

$$o = sgn\left(\sum_{i=1}^{N_s} \alpha_i \cdot t_i \cdot k(\mathbf{x}, \mathbf{x}_i) + b\right)$$

with the bias

$$b = \frac{1}{N_s} \sum_{i=1}^{N_s} \left(t_i - \sum_{j=1}^{N_s} \alpha_j \cdot t_j \cdot k(\mathbf{x}_i, \mathbf{x}_j)\right)$$

Polynomial learning machine

$$k(\mathbf{x}, \mathbf{x}_i) = (\mathbf{x}^T \mathbf{x}_i + 1)^p = (\langle \mathbf{x} | \mathbf{x}_i \rangle + 1)^p$$

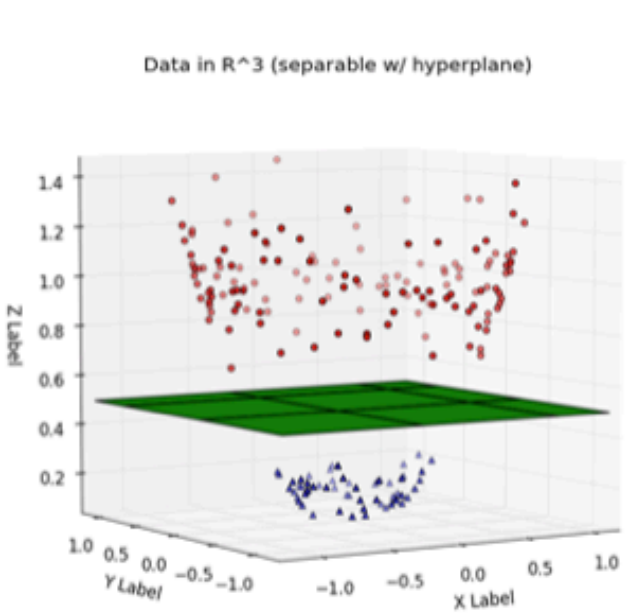
Radial-basis-function network support vector machine

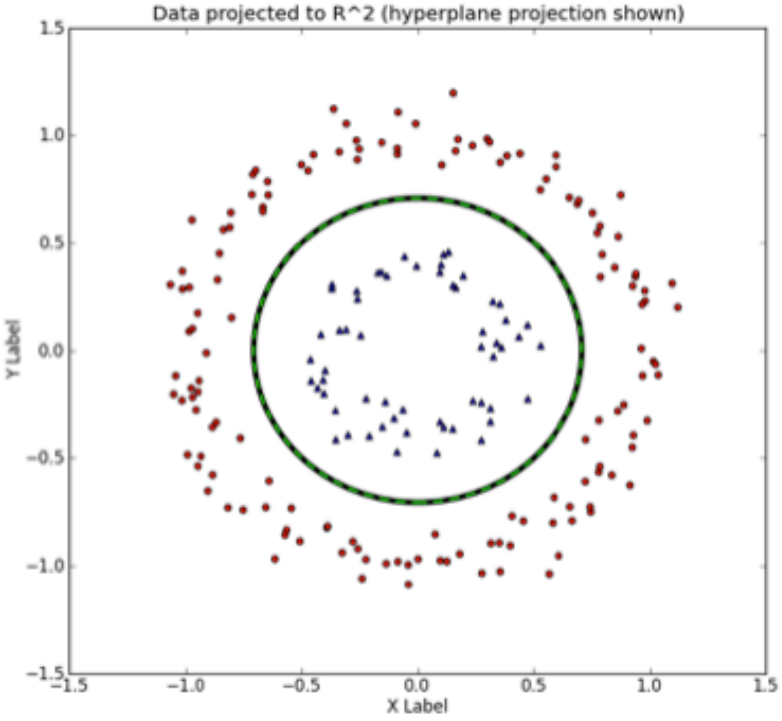
$$k(\mathbf{x}, \mathbf{x}_i) = \exp\left(-\frac{\|\mathbf{x} - \mathbf{x}_i\|^2}{2 \cdot \sigma^2}\right)$$

Two-layer perceptron

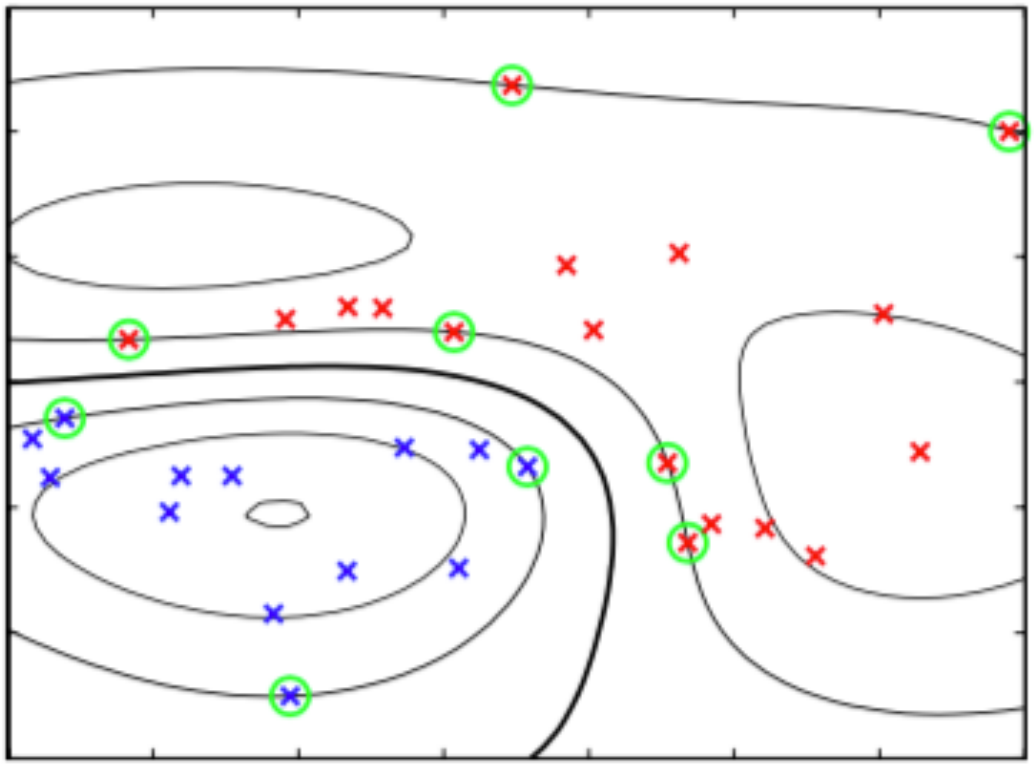
$$k(\mathbf{x}, \mathbf{x}_i) = \tanh\left(\beta_0 \langle \mathbf{x} | \mathbf{x}_i \rangle + \beta_1\right)$$

# Polynomial kernel with degree 2



# Gaussian Kernel



- Two-layer perceptron type of support vector machine is somewhat restricted, this is due to the fact that the determination of whether a given kernel satisfies Mercer's theorem can indeed be a difficult matter.
- For all three machine types, the dimensionality of the feature space is determined by the number of support vectors extracted from the training data by the solution to the constrained-optimization problem.

- The underlying theory of a support vector machine avoids the need for heuristics often used in the design of conventional radial-basis-function networks and multilayer Perceptron.
- In RBF networks the number of basis functions and their centers are determined automatically by the number of support vectors and their values, respectively.

**Example: XOR Problem**

The XOR Problem is described by four vectors, instead of 0 we will use (-1)

$$\mathbf{x}_1 = \begin{pmatrix} -1 \\ -1 \end{pmatrix}, \mathbf{x}_2 = \begin{pmatrix} 1 \\ 1 \end{pmatrix}, \mathbf{x}_3 = \begin{pmatrix} 1 \\ -1 \end{pmatrix}, \mathbf{x}_4 = \begin{pmatrix} 1 \\ 1 \end{pmatrix},$$

and the corresponding target of the two classes is indicated as

$$t_1 = -1, t_2 = 1, t_3 = 1, t_4 = -1.$$

We will use a polynomial kernel

$$k(\mathbf{x}_i, \mathbf{x}_j) = (1 + \mathbf{x}_i^T \mathbf{x}_j)^2$$

with

$$k(\mathbf{x}_i, \mathbf{x}_j) = 1 + x_{i1}^2 \cdot x_{j1}^2 + 2 \cdot x_{i1} \cdot x_{i2} \cdot x_{j1} \cdot x_{j2} + x_{i2}^2 \cdot x_{j2}^2 + 2 \cdot x_{i1} \cdot x_{j1} + 2 \cdot x_{i2} \cdot x_{j2}$$

$$k(\mathbf{x}_i, \mathbf{x}_j) = 1 + x_{i1}^2 \cdot x_{j1}^2 + 2 \cdot x_{i1} \cdot x_{i2} \cdot x_{j1} \cdot x_{j2} + x_{i2}^2 \cdot x_{j2}^2 + 2 \cdot x_{i1} \cdot x_{j1} + 2 \cdot x_{i2} \cdot x_{j2}$$

with the feature vectors (not required, indeed for certain kernels the vector can have an infinite dimension)

$$\Phi(\mathbf{x}_i) = \begin{pmatrix} 1 \\ x_{i1}^2 \\ \sqrt{2} \cdot x_{i1} \cdot x_{i2} \\ x_{i2}^2 \\ \sqrt{2} \cdot x_{i1} \\ \sqrt{2} \cdot x_{i2} \end{pmatrix}, \quad \Phi(\mathbf{x}_j) = \begin{pmatrix} 1 \\ x_{j1}^2 \\ \sqrt{2} \cdot x_{j1} \cdot x_{j2} \\ x_{j2}^2 \\ \sqrt{2} \cdot x_{j1} \\ \sqrt{2} \cdot x_{j2} \end{pmatrix}$$

We obtain the Gram

$$K = \begin{pmatrix} k(\mathbf{x}_1, \mathbf{x}_1) & k(\mathbf{x}_1, \mathbf{x}_2) & k(\mathbf{x}_1, \mathbf{x}_3) & k(\mathbf{x}_1, \mathbf{x}_4) \\ k(\mathbf{x}_2, \mathbf{x}_1) & k(\mathbf{x}_2, \mathbf{x}_2) & k(\mathbf{x}_2, \mathbf{x}_3) & k(\mathbf{x}_2, \mathbf{x}_4) \\ k(\mathbf{x}_3, \mathbf{x}_1) & k(\mathbf{x}_3, \mathbf{x}_3) & k(\mathbf{x}_3, \mathbf{x}_3) & k(\mathbf{x}_4, \mathbf{x}_3) \\ k(\mathbf{x}_4, \mathbf{x}_1) & k(\mathbf{x}_4, \mathbf{x}_4) & k(\mathbf{x}_4, \mathbf{x}_3) & k(\mathbf{x}_4, \mathbf{x}_4) \end{pmatrix} = \begin{pmatrix} 9 & 1 & 1 & 1 \\ 1 & 9 & 1 & 1 \\ 1 & 1 & 9 & 1 \\ 1 & 1 & 1 & 9 \end{pmatrix}$$

The objective function for the dual form of optimization is

$$Q(\alpha) = \sum_{i=1}^{4} \alpha_i - \frac{1}{2} \cdot \sum_{i=1}^{4} \sum_{j=1}^{4} \alpha_i \cdot \alpha_j \cdot t_i \cdot t_j \cdot k(\mathbf{x}_i, \mathbf{x}_j)$$

$$Q(\alpha) = \alpha_1 + \alpha_2 + \alpha_3 + \alpha_4 - \frac{1}{2} \cdot (9\alpha_1^2 - 2\alpha_1\alpha_2 - 2\alpha_1\alpha_3 + 2\alpha_1\alpha_4$$
$$9\alpha_2^2 + 2\alpha_2\alpha_3 - 2\alpha_2\alpha_4 + 9\alpha_3^2 - 2\alpha_3\alpha_4 + 9\alpha_4^2)$$

We maximize the objective function $Q(\alpha)$ by determining the partial derivatives

$$\frac{\partial Q(\alpha)}{\partial \alpha_1} = 1 - 9 \cdot \alpha 1 + \alpha 2 + \alpha 3 - \alpha 4 = 0$$

$$\frac{\partial Q(\alpha)}{\partial \alpha_2} = 1 - + \alpha 1 - 9 \cdot \alpha 2 - \alpha 3 + \alpha 4 = 0$$

$$\frac{\partial Q(\alpha)}{\partial \alpha_3} = 1 + \alpha 1 - \alpha 2 - 9 \cdot \alpha 3 + \alpha 4 = 0$$

$$\frac{\partial Q(\alpha)}{\partial \alpha_4} = 1 - \alpha 1 + \alpha 2 + \alpha 3 - 9 \cdot \alpha 4 = 0$$

that lead to four equation that can be solve by

$$\begin{pmatrix} 9 & -1 & -1 & 1 \\ -1 & 9 & 1 & -1 \\ -1 & 1 & 9 & -1 \\ 1 & -1 & -1 & 9 \end{pmatrix} \cdot \begin{pmatrix} \alpha_1 \\ \alpha_2 \\ \alpha_3 \\ \alpha_4 \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \end{pmatrix}$$

$$\begin{pmatrix} 9 & -1 & -1 & 1 \\ -1 & 9 & 1 & -1 \\ -1 & 1 & 9 & -1 \\ 1 & -1 & -1 & 9 \end{pmatrix} \cdot \begin{pmatrix} \alpha_1 \\ \alpha_2 \\ \alpha_3 \\ \alpha_4 \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \end{pmatrix}$$

with the optimum values of the Lagrange multipliers

$$\alpha_{opt,1} = \alpha_{opt,2} = \alpha_{opt,3} = \alpha_{opt,4} = \frac{1}{8}$$

with all input vectors being support vectors.

To compute the output

1. we compute the bias

$$b = \frac{1}{4}\sum_{i=1}^{4}\left(t_i - \sum_{j=1}^{4}\alpha_j \cdot t_j \cdot k(\mathbf{x}_i, \mathbf{x}_j)\right), \tag{10.58}$$

$$b = \frac{1}{4}\left(\cdot\left(-1 - \frac{1}{8}\cdot(-9+1+1-1)\right) + \left(1 - \frac{1}{8}\cdot(-1+9+1-1)\right) + \left(1 - \frac{1}{8}\cdot(-1+1+9-1)\right) + \left(-1 - \frac{1}{8}\cdot(-1+1+1-9)\right)\right) = 0.$$

In this case we had an optimum Lagrange multipliers $\alpha_{opt,i}$, se we could as well use the form

$$b_{opt} = 1 - \sum_{i=1}^{N}\alpha_{opt,i} \cdot t_i \cdot \cdot k(\mathbf{x}_i, \mathbf{x}^{(s)}), \quad for \quad t^{(s)} = 1.$$

2. then the output

$$o = sgn\left(\sum_{i=1}^{4}\alpha_i \cdot t_i \cdot k(\mathbf{x}, \mathbf{x}_i) + b\right). \tag{10.59}$$

For the query vector

$$\mathbf{x}_q = \begin{pmatrix} -1 \\ -1 \end{pmatrix}$$

we get

$$o = sgn\left(\frac{1}{8} \cdot (-9 + 1 + 1 - 1)\right) = sgn(-1) = -1$$

Since the feature vector $\Phi(\mathbf{x}_i)$ has a finite dimension, we can determine the hyperplane (line) by

$$\mathbf{w} = \sum_{i=1}^{4} \alpha_i \cdot t_i \cdot \Phi(\mathbf{x}_i)$$

$$\mathbf{w}_{opt} = \frac{1}{8} \left( -\phi(\mathbf{x}_1) + \phi(\mathbf{x}_2) + \phi(\mathbf{x}_3) - \phi(\mathbf{x}_4) \right)$$

and

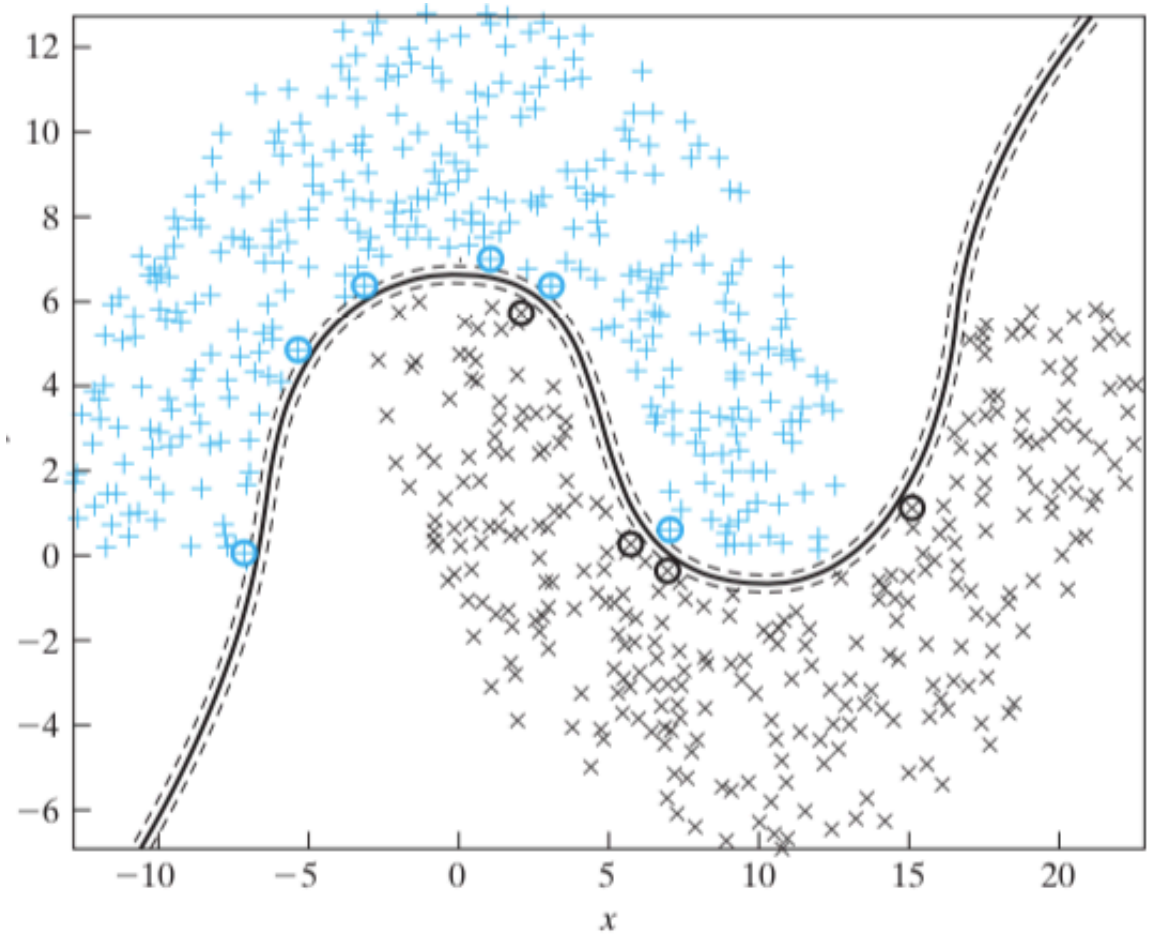
$$\mathbf{w}_{opt} = \begin{pmatrix} 0 \\ 0 \\ -\frac{1}{\sqrt{2}} \\ 0 \\ 0 \\ 0 \end{pmatrix}.$$

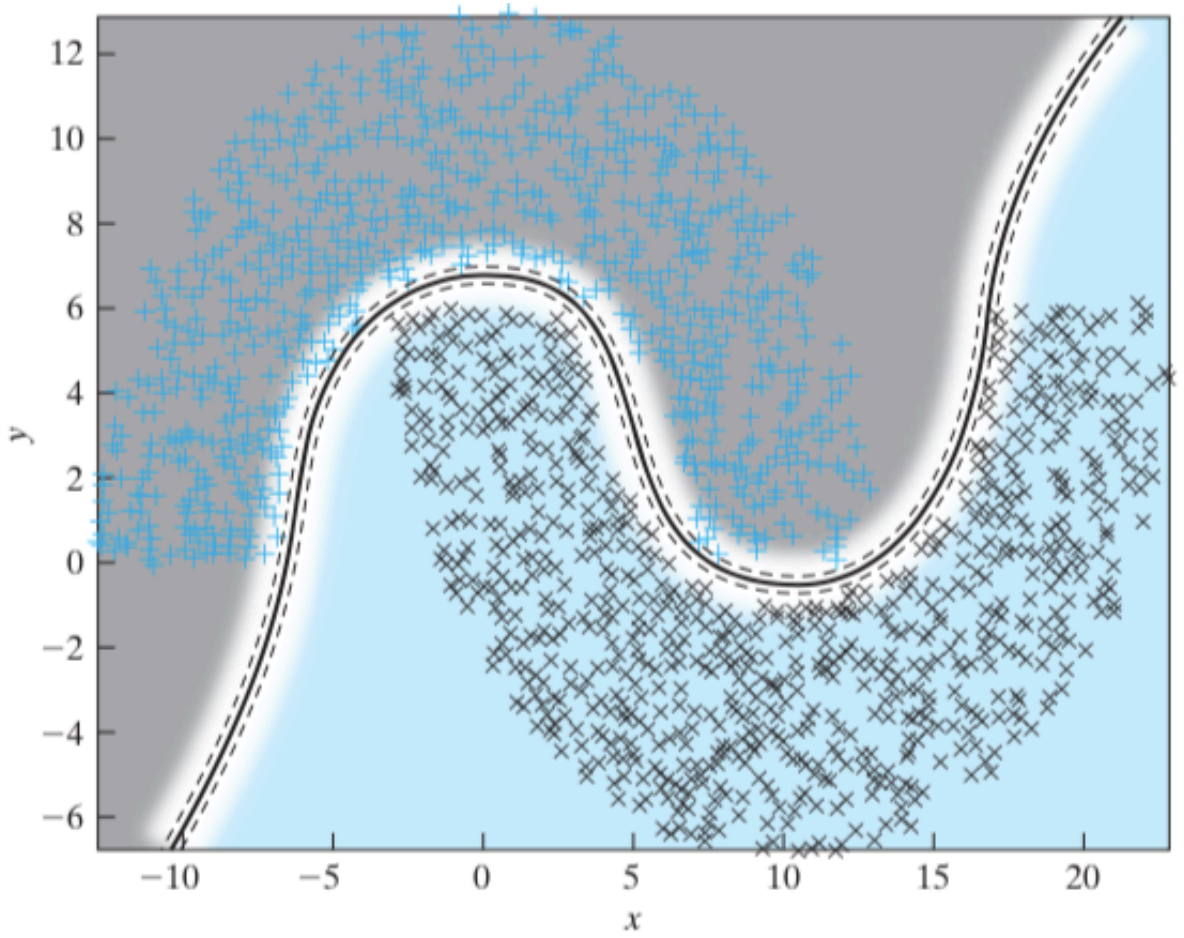
With bias zero

$$\mathbf{w}_{opt}^T \cdot \Phi(\mathbf{x}) + 0 = 0$$

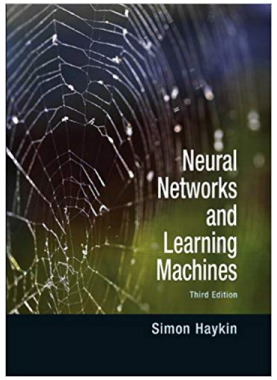
we get the hyperplane (line)

$$(0, 0, -\frac{1}{\sqrt{2}}, 0, 0, 0) \cdot \begin{pmatrix} 1 \\ x_1^2 \\ \sqrt{2} \cdot x_1 \cdot x_2 \\ x_2^2 \\ \sqrt{2} \cdot x_1 \\ \sqrt{2} \cdot x_2 \end{pmatrix} = -x_1 \cdot x_2 = 0.$$

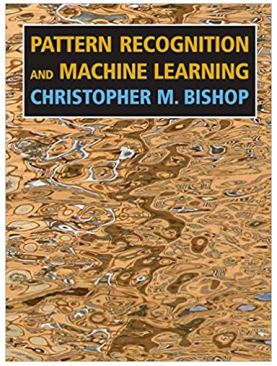

12
10
8
6
4
2
0
−2
−4
−6
−10
−5
0
5
10
15
20
x


12
10
8
6
4
2
0
−2
−4
−6
y
−10
−5
0
5
10
15
20
x

# Literature

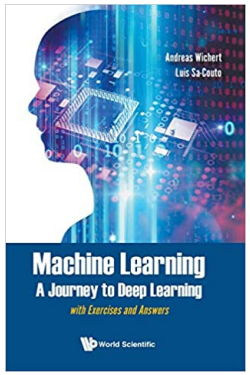



- Simon O. Haykin, Neural Networks and Learning Machine, (3rd Edition), Pearson 2008
  - Chapter 6




- Christopher M. Bishop, Pattern Recognition and Machine Learning (Information Science and Statistics), Springer 2006
  - Chapter 7

# Literature




- Machine Learning - A Journey to Deep Learning, A. Wichert, Luis Sa-Couto, World Scientific, 2021
  - Chapter 11