Tutorial 1

Support Vector Machines (SVMs)

Optimal hyperplane for linearly separable patterns

Two-class linearly separable task

- training set

$$\{(\mathbf{x}_i, d_i)\}_{i=1}^{N}$$

where $\mathbf{x}_i$ is the input pattern vector for the *i*th example and d_i is the corresponding desired response

- for patterns from $\omega_1 \rightarrow d_i = +1$

  $\omega_2 \rightarrow d_i = -1$

- classes are linearly separable:

$$\mathbf{w}^T\mathbf{x} + w_{n+1} = 0$$

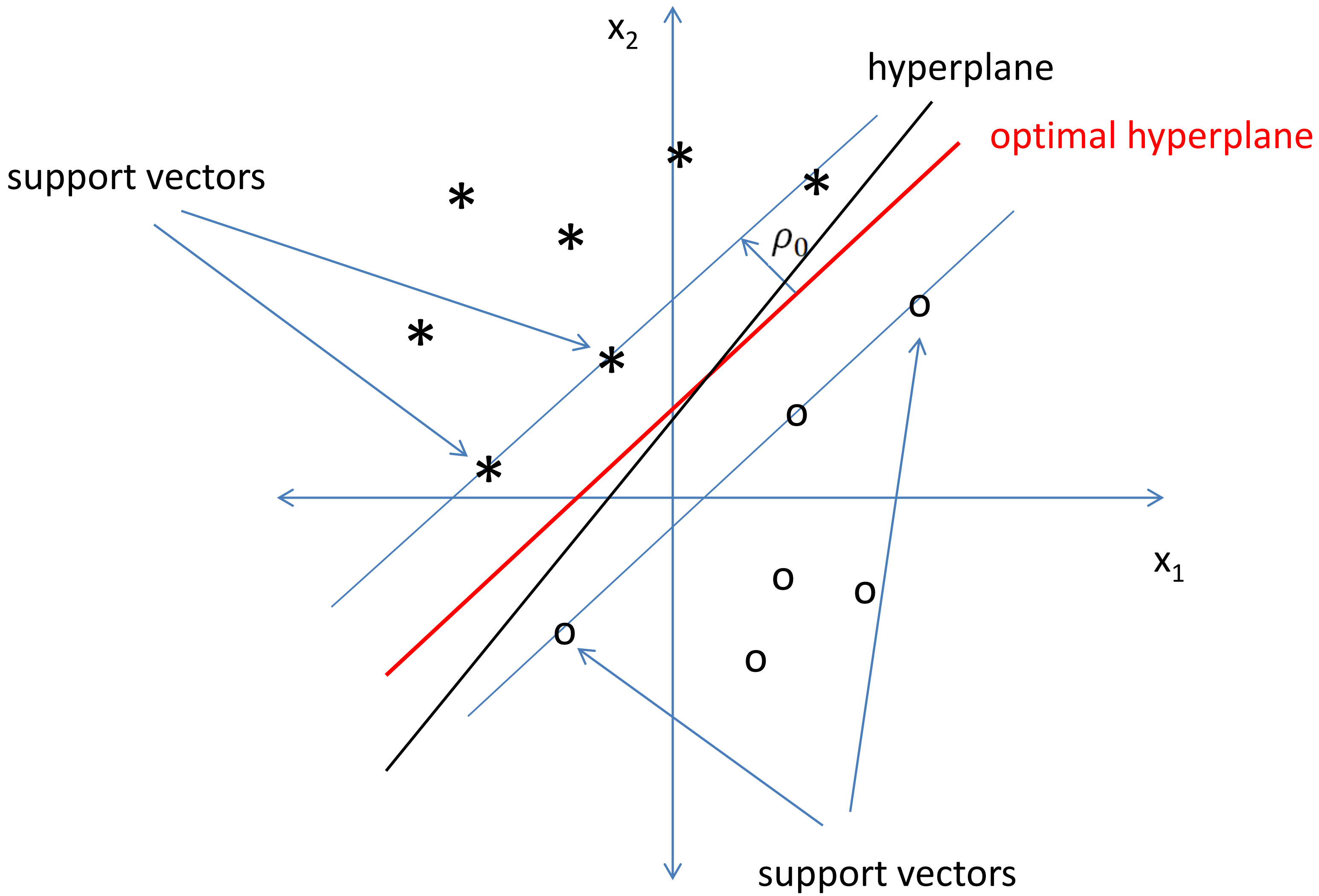
-where **x** is an input vector, **w** is weight vector and w_{n+1} is a bias /let denote w_{n+1} by *b*/

- we may write

$$\mathbf{w}^T\mathbf{x}_i + b \geq 0 \quad \text{for } d_i = +1$$
$$\mathbf{w}^T\mathbf{x}_i + b < 0 \quad \text{for } d_i = -1$$

- for given weight vector **w** and bias *b,* the separation ρ between the hyperplane (defined by $\mathbf{w}^T\mathbf{x}_i + b = 0$) and the closest data point is called the margin of separation ρ

- the goal of a support vector machine is to find the hyperplane for which the margin of separation is maximized



x_2
hyperplane
optimal hyperplane
support vectors
ρ_0
x_1
support vectors

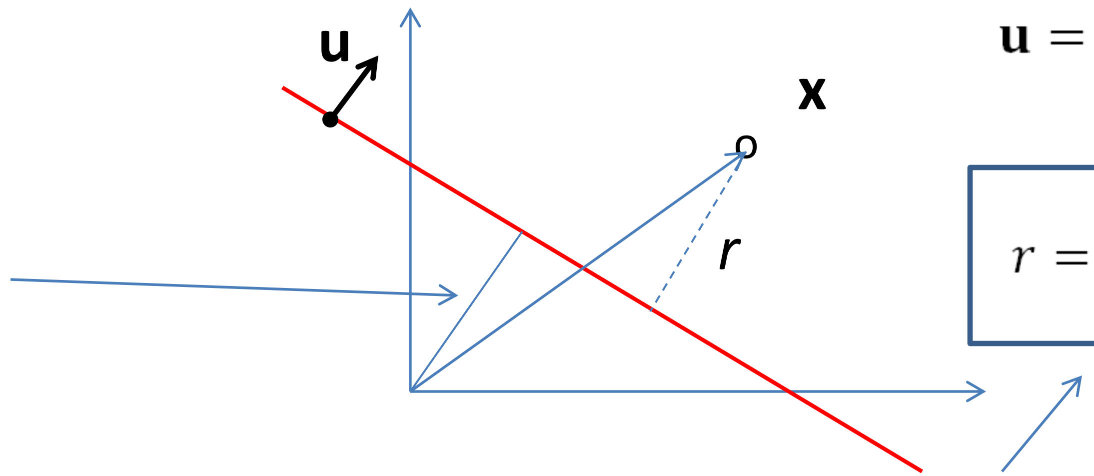
- let $\mathbf{w}_0$ and b_0 denote the optimum values of the weight vector and bias
- optimal hyperplane:

$$\mathbf{w}_0^T\mathbf{x} + b_0 = 0$$

- decision function: $g(\mathbf{x}) = \mathbf{w}_0^T\mathbf{x} + b_0$

$$\mathbf{u} = \frac{\mathbf{w}_0}{\|\mathbf{w}_0\|}$$

$$\frac{b_0}{\|\mathbf{w}_0\|}$$



$$\boxed{r = \frac{g(\mathbf{x})}{\|\mathbf{w}_0\|}}$$

- decision function gives an algebraic measure of distance from **x** and the optimal hyperplane

- each hyperplane is determined within a scaling factor

- we can scale $\mathbf{w_0}$, b_0 so that the value of $g(\mathbf{x})$, at the nearest points in ω_1, ω_2 is equal to 1 for ω_1 and, thus, equal to -1 for ω_2.

- the margin is:

$$\rho = 2r = \frac{1}{\|\mathbf{w}_0\|} + \frac{1}{\|\mathbf{w}_0\|} = \frac{2}{\|\mathbf{w}_0\|}$$

- the pair ($\mathbf{w}_0$, b_0) must satisfy the constraint:

$$\mathbf{w}_0{}^T\mathbf{x} + b_0 \ \geq 1 \quad \text{for } d_i = \ +1$$

$$\mathbf{w}_0{}^T\mathbf{x} + b_0 \ \leq -1 \quad \text{for } d_i = \ -1$$

for all $\{(\mathbf{x}_i, d_i)\}_{i=1}^{N}$

- the particular data points $(\mathbf{x}_i, d_i)$ for which $\mathbf{w}_0{}^T\mathbf{x}_i + b_0 = 1$ for d_i = +1 (class ω_1) and $\mathbf{w}_0{}^T\mathbf{x}_i + b_0 = -1$ for d_i = -1 (class ω_2) are called supports vectors
- supports vectors play a prominent role in the learning decision functions
- the support vectors are those data points that lie closest to the decision surface and are the most difficult to classify

- a support vector $\mathbf{x}^{(s)}$ for which $d^{(s)} = \pm 1$:

$$g(\mathbf{x}^{(s)}) = \mathbf{w}_0{}^T \mathbf{x}^{(s)} + b_0 = \pm 1 \quad \text{for } d^{(s)} = \pm 1$$

-algebraic distance from the support vector $\mathbf{x}^{(s)}$ to the optimal hyperplane is:

$$r = \frac{g(\mathbf{x}^{(s)})}{\|\mathbf{w}_0\|} = \begin{cases} \dfrac{1}{\|\mathbf{w}_0\|} & \text{if } d^{(s)} = +1 \\ -\dfrac{1}{\|\mathbf{w}_0\|} & \text{if } d^{(s)} = -1 \end{cases}$$

- let ρ denote the optimal value of the margin of separation between two classes that constitute the training set $\{(\mathbf{x}_i, d_i)\}_{i=1}^{N}$

$$\rho = 2r = \frac{1}{\|\mathbf{w}_0\|} + \frac{1}{\|\mathbf{w}_0\|} = \frac{2}{\|\mathbf{w}_0\|}$$

$$\boxed{\rho = 2r = \frac{1}{\|\mathbf{w}_0\|} + \frac{1}{\|\mathbf{w}_0\|} = \frac{2}{\|\mathbf{w}_0\|}}$$

- maximizing the margin of separation between classes is equivalent to minimizing the Euclidian norm of the weight vector $\mathbf{w_0}$

- vector $\mathbf{w}_0$ provides the maximum possible separation between positive (from ω_1) and negative examples (from ω_2)

Quadratic optimization for finding the optimal hyperplane

- goal is to develop a computationally efficient procedure for using the training set of samples $\{(\mathbf{x}_i, d_i)\}_{i=1}^{N}$ to find optimal hyperplane, subject to the constrains:

$$d_i(\mathbf{w}^T\mathbf{x}_i + b) \geq 1 \quad \text{for } i = 1, 2, \ldots, N$$

- the constrained optimization problem (primal problem):

Given the training set $\{(\mathbf{x}_i, d_i)\}_{i=1}^{N}$, find the optimum value of the weight vector **w** and bias *b* such that they satisfy the constrains

$d_i(\mathbf{w}^T\mathbf{x}_i + b) \geq 1 \quad \text{for } i = 1, 2, \ldots, N$ and the

weight vector **w** minimize the cost function $J(\mathbf{w}) = \frac{1}{2}\mathbf{w}^T\mathbf{w}$

Cost function:

$$J(\mathbf{w}) = \frac{1}{2}\,\mathbf{w}^T\mathbf{w} \qquad\qquad \mathbf{w}^T\mathbf{w} = \|\mathbf{w}\|^2$$

- scaling factor $\frac{1}{2}$ included here for convenience of presentation
- the cost function *J(***w***)* is a convex function of **w**
- the constrains are linear in **w**

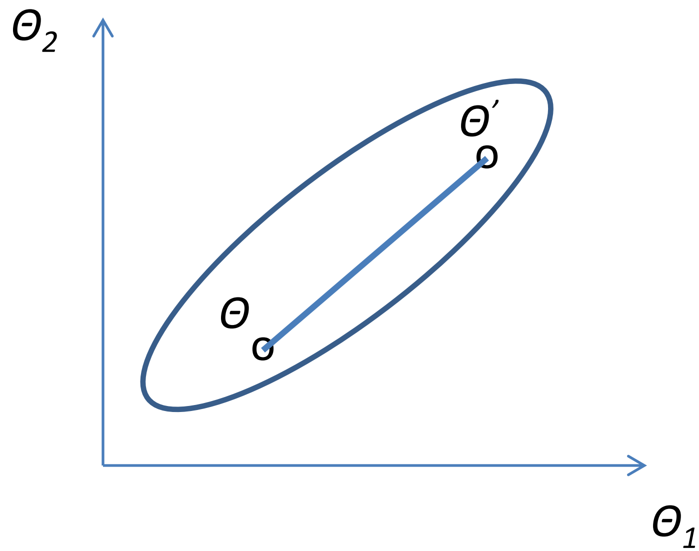
This is a nonlinear (quadratic) optimization task subject to a set of linear inequality constrains

$$J(\mathbf{w}) = \frac{1}{2}\,\mathbf{w}^T\mathbf{w}$$

$$d_i(\mathbf{w}^T\mathbf{x}_i + b) \geq 1 \quad \text{for } i = 1, 2, \ldots, N$$

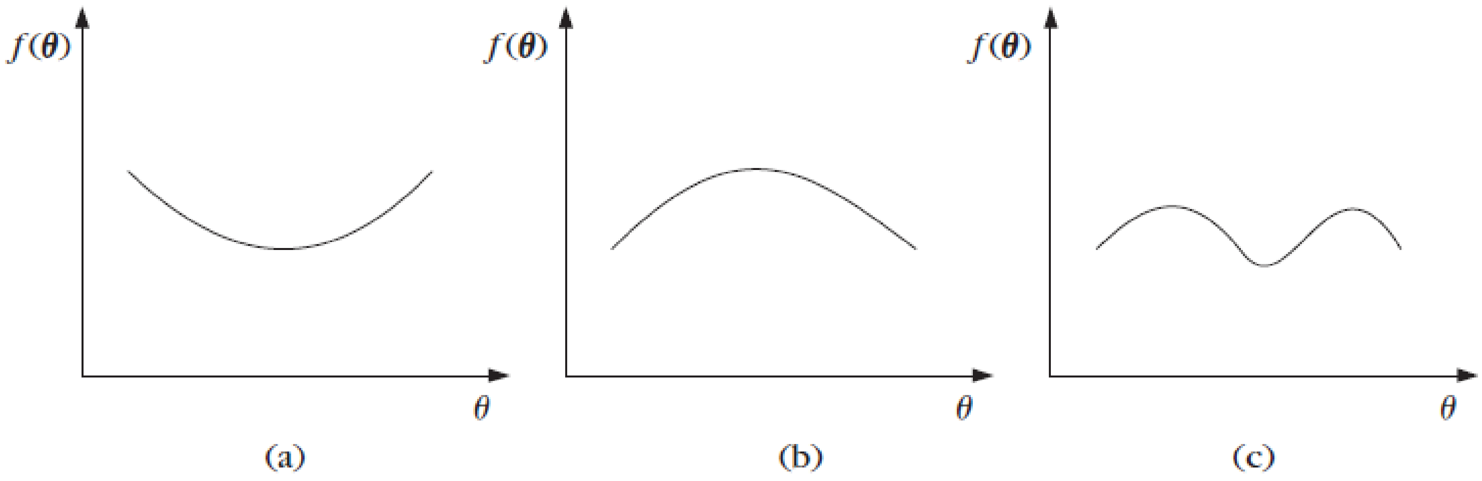
To solve the constrained optimization problem → the method of Lagrange multipliers

Convex set: A set S $\subseteq R^l$ is called convex, if for every pair of points Θ, $\Theta' \in$ S, the line segment joining these points also belongs to the set.



Convex function: Let S be convex set. A function $f(\Theta): S \subseteq R^l \rightarrow R$ is called convex in S, if for every Θ and $\Theta' \in$ S

$$f(\lambda \Theta + (1-\lambda)\Theta') \leq \lambda f(\Theta) + (1-\lambda) f(\Theta') \text{ for every } \lambda \in [0, 1]$$



a) convex function
b) concave function
c) neither convex nor concave

Constrain optimization – an example

Find the optimum values of the x and y such that they satisfied the constrain φ (x, y) = 0 and maximize the function z= f(x, y).

Method of Lagrange multipliers

1. Construct the Lagrangian function F:

F (x, y, λ) = *f*(x, y) + λ φ (x, y), where λ is *Lagrange multiplier*

2. Differentiating F (x, y, λ) with respect to x and y and setting the results equal zero, the following conditions of optimality is obtained:

$$\frac{\partial F}{\partial x} = 0 \quad \text{and} \quad \frac{\partial F}{\partial y} = 0 \qquad \varphi(x, y) = 0$$

3. From system of equations $\frac{\partial F}{\partial x} = 0$ $\frac{\partial F}{\partial y} = 0$ and φ (x, y) = 0 find x, y and λ

4. If second-order total differentials

$d^2F < 0$ for x and y - the function z= f(x, y) has a maximum
$d^2F > 0$ for x and y - the function z= f(x, y) has a minimum

Example:
We are looking for the extreme of the function z= x + 2y with following constraint $x^2 + y^2 = 5$.

1. $F(x, y, \lambda) = x + 2y + \lambda(x^2 + y^2 - 5)$
2. $\frac{\partial F}{\partial x} = 1 + 2x\lambda = 0$ and $\frac{\partial F}{\partial y} = 2 + 2y\lambda = 0$
3. From $1 + 2x\lambda = 0$
$2 + 2y\lambda = 0$
$x^2 + y^2 - 5 = 0$

$x = -\frac{1}{2\lambda}$ and $y = -\frac{1}{\lambda}$ substitute in $x^2 + y^2 - 5 = 0$:

$$\frac{1}{4\lambda^2} + \frac{1}{\lambda^2} - 5 = 0$$

$$\frac{1}{4\lambda^2} + \frac{1}{\lambda^2} - 5 = 0 \; /4\lambda^2$$

$$5(1 - 4\lambda^2) = 0 \qquad \lambda_{1,2} = \pm\sqrt{\frac{1}{4}} \qquad \lambda_1 = +\frac{1}{2}$$

$$\lambda_2 = -\frac{1}{2}$$

for $\lambda_1 = +\frac{1}{2}$ $x_1 = -1$ and $y_1 = -2$

for $\lambda_2 = -\frac{1}{2}$ $x_2 = 1$ and $y_2 = 2$

$$d^2F = \frac{\partial^2 F}{\partial x^2}\, dx^2 + 2\frac{\partial^2 F}{\partial x \partial y}\, dxdy + \frac{\partial^2 F}{\partial y^2}\, dy^2$$

$$\frac{\partial^2 F}{\partial x^2} = F_{xx} = 2\lambda \qquad \frac{\partial^2 F}{\partial y^2} = F_{yy} = 2\lambda \qquad \frac{\partial^2 F}{\partial x \partial y} = F_{xy} = 0$$

$$d^2F = 2\lambda dx^2 + 2\lambda dy^2 = 2\lambda(dx^2 + dy^2)$$

For $\lambda_1 = +\frac{1}{2}$ $\quad d^2F > 0$ minimum $f(x, y)$

For $\lambda_2 = -\frac{1}{2}$ $\quad d^2F < 0$ maximum $f(x, y)$

$$z(x, y) = x + 2y = (-1) + \ 2(-2) = \ -5$$

$$z(x, y) = x + 2y = (+1) + \ 2(+2) = \ 5$$

Method of Lagrange multipliers

i) Construct the Lagrangian function

$$J(\mathbf{w}, b, \boldsymbol{\lambda}) = \frac{1}{2}\mathbf{w}^T\mathbf{w} - \sum_{i=1}^{N} \lambda_i [d_i (\mathbf{w}^T\mathbf{x}_i + b) - 1]$$

where $\boldsymbol{\lambda}$ is a vector of Lagrange multipliers (nonnegative variables) λ_i, $i = 1, 2, \ldots, N$

ii) The Karush-Kuhn-Tucker (KKT) conditions

a) $$\frac{\partial J(\mathbf{w}, b, \boldsymbol{\lambda})}{\partial \mathbf{w}} = \mathbf{0}$$

b) $$\frac{\partial J(\mathbf{w}, b, \boldsymbol{\lambda})}{\partial b} = 0$$

c) $\lambda_i \geq 0$, $i = 1, 2, \ldots, N$

c) $\lambda_i [d_i(\mathbf{w}^{\mathbf{T}}\mathbf{x}_i + b) - 1] = 0, \qquad i = 1, 2, \ldots, N$

$$\frac{\partial J(\mathbf{w}, b, \boldsymbol{\lambda})}{\partial \mathbf{w}} = \mathbf{w} - \sum_{i=1}^{N} \lambda_i \, d_i \mathbf{x}_i = \mathbf{0}$$

$$\boxed{\mathbf{w} = \sum_{i=1}^{N} \lambda_i \, d_i \mathbf{x}_i}$$

$$\frac{\partial J(\mathbf{w}, b, \boldsymbol{\lambda})}{\partial b} = -\sum_{i=1}^{N} \lambda_i \, d_i = 0$$

$$\boxed{\sum_{i=1}^{N} \lambda_i \, d_i = 0}$$

$$\boxed{\mathbf{w} = \sum_{i=1}^{N} \lambda_i \, d_i \mathbf{x}_i}$$

- the Lagrange multipliers can be either zero or positive
- weight vector **w** of the optimal solution is a linear combination of $N_s \leq N$ feature vectors that are associated with $\lambda_i \neq 0$:

$$\mathbf{w} = \sum_{i=1}^{N_s} \lambda_i \, d_i \mathbf{x}_i$$

- feature vectors that are associated with $\lambda_i \neq 0$ are support vectors
- due to the set of constrains

$$\lambda_i [d_i (\mathbf{w}^{\mathbf{T}} \mathbf{x}_i + \ b) - 1] = 0, \qquad i = 1, 2, \ldots, N$$

support vectors lie in two hyperplanes:

$$\mathbf{w}^T \mathbf{x} + b = \ \pm 1$$

- support vectors are the training vectors that are closest to the linear classifier and they constitute the critical elements of the training set
- feature vectors corresponding to $\lambda_i = 0$ can lie outside the class separation band or can also lie on one of these hyperplanes

$$\mathbf{w}^T\mathbf{x} + b = \pm 1$$

but the resulting hyperplane classifier is insensitive to the number and position of such feature vectors

- **w** is explicitly given, *b* can be implicitly obtained by any of the conditions

$$\lambda_i[d_i(\mathbf{w}^\mathbf{T}\mathbf{x}_i + b) - 1] = 0, \qquad i = 1, 2, \ldots, N$$

- in practice *b* is computed as an average value obtained using all conditions of above type

- the primal problem deals with a convex cost function and linear constrains
- it is possible to construct another problem – the dual problem
- duality theorem

a) If the primal problem has an optimal solution, the dual problem also has an optimal solution, and the corresponding optimal values are equal;
b) In order for $\mathbf{w}_0$ to be an optimal solution and λ_0 to be an optimal solution, it is necessary and sufficient that $\mathbf{w}_0$ is feasible for the primal problem

Optimization task

- minimize $J(\mathbf{w})$ with constrains: $\varphi_i(\mathbf{w}) \geq 0$, $i=1, 2, \dots, N$

$$J(\mathbf{w}, \boldsymbol{\lambda}) = J(\mathbf{w}) - \sum_{i=1}^{N} \lambda_i \varphi_i(\mathbf{w})$$

- the maximum value of $J(\mathbf{w}, \boldsymbol{\lambda})$ is when $\lambda_i = 0$ for $i = 1, 2, \dots, N$ or $\varphi_i(\mathbf{w}) = 0$ (or both) – maximum value of $J(\mathbf{w}, \boldsymbol{\lambda})$ is $J(\mathbf{w})$
- the minimal value of $J(\mathbf{w}, \boldsymbol{\lambda})$ is when $\lambda_i > 0$ for $i = 1, 2, \dots N$ and $\varphi_i(\mathbf{w}) \geq 0$
- the original problem is equivalent to:

$$\min_{\mathbf{w}} J(\mathbf{w}) = \min_{\mathbf{w}} \max_{\boldsymbol{\lambda}} J(\mathbf{w}, \boldsymbol{\lambda})$$

The dual problem:

$$\max_{\boldsymbol{\lambda}\geq \mathbf{0}} \underbrace{min_{\mathbf{w}} J(\mathbf{w},\boldsymbol{\lambda})}$$

solution of this part is

$$\mathbf{w} = \sum_{i=1}^{N} \lambda_i\, d_i \mathbf{x}_i$$ and

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

- by substituting $\mathbf{w} = \sum_{i=1}^{N} \lambda_i\, d_i \mathbf{x}_i$ into Lagrangian function

$$J(\mathbf{w}, b, \boldsymbol{\lambda})$$

it becomes independent of **w** and *b*

$$J(\mathbf{w}, b, \boldsymbol{\lambda}) = \frac{1}{2}\mathbf{w}^T\mathbf{w} - \sum_{i=1}^{N} \lambda_i \left[d_i(\mathbf{w}^T\mathbf{x}_i + b) - 1\right]$$

$$J(\mathbf{w}, b, \boldsymbol{\lambda}) = \frac{1}{2}\,\mathbf{w}^T\mathbf{w} - \sum_{i=1}^{N} \lambda_i\, d_i \mathbf{w}^T\mathbf{x}_i - b \underbrace{\sum_{i=1}^{N} \lambda_i\, d_i}_{0} + \sum_{i=1}^{N} \lambda_i$$

$$\mathbf{w} = \sum_{i=1}^{N} \lambda_i\, d_i \mathbf{x}_i \qquad\qquad \mathbf{w}^T\mathbf{w} = \sum_{i=1}^{N} \lambda_i\, d_i \mathbf{w}^T\mathbf{x}_i$$

$$\mathbf{w}^T\mathbf{w} = \sum_{i=1}^{N} \lambda_i\, d_i \mathbf{w}^T\mathbf{x}_i = \sum_{i=1}^{N}\sum_{j=1}^{N} \lambda_i \lambda_j d_i d_j \mathbf{x}_i^T\mathbf{x}_j$$

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

$$max_{\boldsymbol{\lambda}}\, Q(\boldsymbol{\lambda}) = \sum_{i=1}^{N} \lambda_i - \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N} \lambda_i \lambda_j d_i\, d_j \mathbf{x}_i^T\mathbf{x}_j$$

The dual problem (Wolfe dual representation):

Given the training sample $\{(\mathbf{x}_i, d_i)\}_{i=1}^{N}$ find the Lagrange multipliers $\{\lambda_i\}_{i=1}^{N}$ that maximize the objective (cost) function

$$Q(\boldsymbol{\lambda}) = \sum_{i=1}^{N} \lambda_i - \frac{1}{2}\sum_{i=1}^{N}\sum\nolimits_{j=1}^{N} \lambda_i \lambda_j d_i d_j \mathbf{x}_i^T \mathbf{x}_j$$

subject to the constrains

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

ii) $\lambda_i \geq 0$ for $i = 1, 2, \dots, N$

The function $Q(\boldsymbol{\lambda})$ to be maximized depends only on input pattern

in a form of a set of dot products $\{\mathbf{x}_i^T \mathbf{x}_j\}_{(i,j)}^{N}$

The dual problem:

Multiply $Q(\boldsymbol{\lambda})$ by (-1): $N(\boldsymbol{\lambda}) = Q(\boldsymbol{\lambda}) \times (-1)$

$$N(\boldsymbol{\lambda}) = \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N} \lambda_i \lambda_j d_i d_j \mathbf{x}_i^T \mathbf{x}_j - \sum_{i=1}^{N} \lambda_i$$

and for given the training sample $\{(\mathbf{x}_i, d_i)\}_{i=1}^{N}$ find the Lagrange multipliers $\{\lambda_i\}_{i=1}^{N}$ that minimize the objective (cost) function $N(\boldsymbol{\lambda})$

subject to the constrains:

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

ii) $\lambda_i \geq 0$ for $i = 1, 2, \dots, N$

Let us define a new Lagrangian function:

$$N_L(\boldsymbol{\lambda}, \mu) = \frac{1}{2}\sum_{i=1}^{N}\sum\nolimits_{j=1}^{N} \lambda_i \lambda_j d_i d_j \mathbf{x}_i^T \mathbf{x}_j - \sum_{i=1}^{N} \lambda_i + \mu \sum_{i=1}^{N} \lambda_i d_i$$

ii) The Karush-Kuhn-Tucker (KKT) conditions

$$\frac{\partial N_L(\boldsymbol{\lambda}, \mu)}{\partial \boldsymbol{\lambda}} = \mathbf{0}$$

$$\frac{\partial N_L(\boldsymbol{\lambda}, \mu)}{\partial \mu} = 0$$

$$\mu \geq 0$$

- having determined the optimum Lagrange multipliers $\lambda_{o,i}$, the optimum weight vector $\mathbf{w}_0$ may be computed:

$$\boxed{\mathbf{w}_0 = \sum_{i=1}^{N} \lambda_{0,i}\, d_i \mathbf{x}_i}$$

- the optimum bias b_0 is computed based on the positive support vector ($d^{(s)}$ = +1) from:

$$\mathbf{w}_0^T\,\mathbf{x}^{(s)} + b_0 = 1 \text{ for } d^{(s)} = +1$$

$$\boxed{b_0 = 1 - \mathbf{w}_0^T\,\mathbf{x}^{(s)}}$$

Example:

Consider the two-class classification task that consists of the following training patterns:

$\omega_1 = \{\mathbf{x}_1=(0, 0)^T, \mathbf{x}_2=(0, 1)^T\}$ and

$\omega_2 = \{\mathbf{x}_3=(1, 0)^T, \mathbf{x}_4=(1, 1)^T\}$

using the SVM approach (dual problem) find optimal hyperplane (line)!

- for patterns from ω_1: $d_1 = +1$, $d_2 = +1$
- for patterns from ω_2: $d_3 = -1$, $d_4 = -1$

Dual problem:

Find the Lagrange multipliers λ_i, $i = 1, 2, 3, 4$ that minimise the objective function:

$$N(\boldsymbol{\lambda}) = \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}\lambda_i \lambda_j d_i d_j \mathbf{x}_i^T \mathbf{x}_j - \sum_{i=1}^{N}\lambda_i$$

subject to the constrains

i) $\sum_{i=1} \lambda_i d_i = 0$

ii) $\lambda_i \geq 0$ for $i = 1, 2, \ldots, N$

$$N(\boldsymbol{\lambda}) = \frac{1}{2}\sum_{i=1}^{4}\sum\nolimits_{j=1}^{4} \lambda_i \lambda_j d_i \, d_j \mathbf{x}_i^T \mathbf{x}_j - \sum_{i=1}^{4} \lambda_i$$

$$\sum_{i=1}^{4} \lambda_i \, d_i = 0$$

$\lambda_i \geq 0$; $i = 1, 2, 3, 4$;

For training vectors the function $N(\boldsymbol{\lambda})$ is obtained:

$$N(\boldsymbol{\lambda}) = \frac{1}{2}(\lambda_2^2 - 2\lambda_2\lambda_4 + 2\lambda_3\lambda_4 + \lambda_3^2 + 2\lambda_4^2) - \lambda_1 - \lambda_2 - \lambda_3 - \lambda_4$$

$$\sum_{i=1}^{4} \lambda_i \, d_i = 0 \qquad\qquad \lambda_1 + \lambda_2 - \lambda_3 - \lambda_4 = 0$$

Form the Lagrangian function

$$N_L(\boldsymbol{\lambda},\mu) = \frac{1}{2}\lambda_2^2 - \lambda_2\lambda_4 + \lambda_3\lambda_4 + \frac{1}{2}\lambda_3^2 + \lambda_4^2 - \lambda_1 - \lambda_2 - \lambda_3 - \lambda_4 + \mu(\lambda_1 + \lambda_2 - \lambda_3 - \lambda_4)$$

where μ is Lagrange multiplier;

from

$$\frac{\partial N_L(\boldsymbol{\lambda},\mu)}{\partial \lambda_1} = -1 + \mu = 0$$

$$\frac{\partial N_L(\boldsymbol{\lambda},\mu)}{\partial \lambda_2} = \lambda_2 - \lambda_4 - 1 + \mu = 0$$

$$\frac{\partial N_L(\boldsymbol{\lambda},\mu)}{\partial \lambda_3} = \lambda_4 + \lambda_3 - 1 - \mu = 0$$

$$\frac{\partial N_L(\boldsymbol{\lambda},\mu)}{\partial \lambda_4} = -\lambda_2 + \lambda_3 + 2\lambda_4 - 1 - \mu = 0$$

$$\lambda_1 + \lambda_2 - \lambda_3 - \lambda_4 = 0$$

are obtained: $\mu = 1, \quad \lambda_1 = \lambda_2 = \lambda_3 = \lambda_4 = 1$

$$\mathbf{w}_{0}= \sum_{i=1}^{N} \lambda_{0,i}\, d_i \mathbf{x}_i \quad ; N^{(s)} = N$$

$$\mathbf{w}_{0}= \sum_{i=1}^{4} \lambda_{0,i}\, d_i \mathbf{x}_i$$

$\omega_1 = \{\mathbf{x}_1=(0, 0)^T, \mathbf{x}_2=(0, 1)^T\}$
$\omega_2 = \{\mathbf{x}_3=(1, 0)^T, \mathbf{x}_4=(1, 1)^T\}$

$$\mathbf{w}_0 = \begin{bmatrix} 0 \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} - \begin{bmatrix} 1 \\ 0 \end{bmatrix} - \begin{bmatrix} 1 \\ 1 \end{bmatrix} = \begin{bmatrix} -2 \\ 0 \end{bmatrix}$$

$$b_0 = 1 - \mathbf{w}_0 \mathbf{x}^{(s)} \ \text{ for } d^{(s)} = 1$$

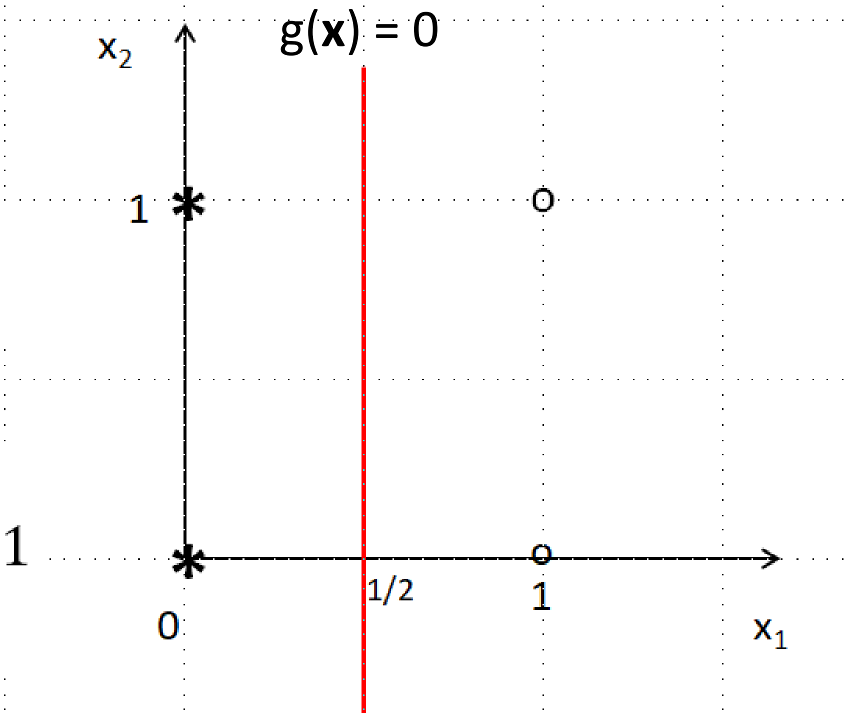
$$g(\mathbf{x}) = [-2, 0] \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + b_0 = -2x_1 + 1$$

$\omega_1 = \{\mathbf{x}_1=(0, 0)^T, \mathbf{x}_2=(0, 1)^T\}$
$\omega_2 = \{\mathbf{x}_3=(1, 0)^T, \mathbf{x}_4=(1, 1)^T\}$

$$g(\mathbf{x}) = -2x_1 + 1 = 0$$

- the margin separation:

$$\rho = \frac{2}{\|w_0\|} = \frac{2}{\sqrt{(-2)^2}} = 1$$



SVMs: The nonlinear case

- how to find the optimal hyperplane for linearly nonseparable patterns?
- Direct approach

mapping:

$$\mathbf{x} \in R^l \longrightarrow \mathbf{y} \in R^k \qquad k > l$$

- Implicit or "hidden" approach

- it can be achieved in two steps:

i) Nonlinear mapping of an input vector (pattern) into high-dimensional feature space that is hidden from both the input and output

ii) Construction of an optimal hyperplane for separating features discovered in step i)

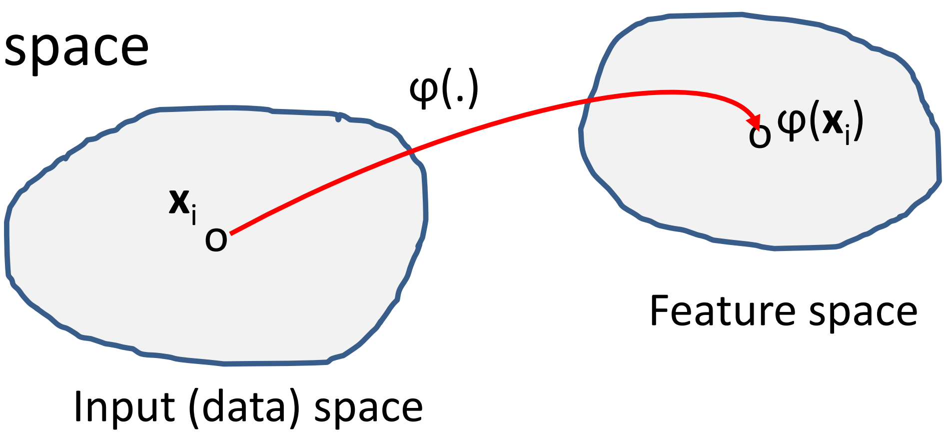
- the step i) is based on Cover's theorem on the separability of patterns (1965):

A complex pattern-classification problem cast in a high-dimensional space nonlinearly is more likely to be linearly separable than in a low-dimensional space

Two conditions have to be satisfied:

i) transformation is nonlinear
ii) the dimensionality of the feature space has to be enough high

Nonlinear mapping of an input vector into a high-dimensional feature space



x is a l-dimensional vector drawn from the input space

$\{\varphi_j(\mathbf{x})\}_{j=1}^{k}$ - a set of nonlinear transformations from the input space to the feature space

- it is assumed that $\{\varphi_j(\mathbf{x})\}_{j=1}^{k}$ is defined a priori for all *j*

- a hyperplane (decision surface) is:

$$\sum_{j=1}^{k} w_j \, \varphi_j(\mathbf{x}) + b = 0$$, where $\{w_j\}_{j=1}^{k}$ is a set of linear weights and *b* is a bias

we can write $$\sum_{j=0}^{k} w_j \, \varphi_j(\mathbf{x}) = 0$$, where $\varphi_0(\mathbf{x}) = 1$ for all **x**

and w_0 denotes the bias *b*

- define the vector:

$$\boldsymbol{\varphi}(\mathbf{x}) = [\varphi_0(\mathbf{x}), \varphi_1(\mathbf{x}), \dots, \varphi_k(\mathbf{x})]^T$$

- vector $\boldsymbol{\varphi}(\mathbf{x})$ represents the "image" induced in the feature space due to the input vector **x**
- the decision surface

$$\mathbf{w}^T \boldsymbol{\varphi}(\mathbf{x}) = 0 \qquad (*)$$

-we now seek linear separability of features
- from SVM we have the following result:

$$\mathbf{w} = \sum_{i=1}^{N} \lambda_i d_i \mathbf{x}_i \quad \rightarrow \quad \mathbf{w} = \sum_{i=1}^{N} \lambda_i d_i \boldsymbol{\varphi}(\mathbf{x}_i) \quad (**)$$

- substitute (**) in (*):

$$\sum_{i=1}^{N} \lambda_i d_i \boldsymbol{\varphi}^{\mathrm{T}}(\mathbf{x}_i) \boldsymbol{\varphi}(\mathbf{x}) = 0$$

$\boldsymbol{\varphi}^{\mathrm{T}}(\mathbf{x}_i)\boldsymbol{\varphi}(\mathbf{x})$ - represents the *inner product* of two vectors induced in the feature space by the input vector **x** and vector $\mathbf{x}_i$

- the inner-product kernel is defined as:

$$K(\mathbf{x}, \mathbf{x}_i) = \boldsymbol{\varphi}^{\mathrm{T}}(\mathbf{x})\boldsymbol{\varphi}(\mathbf{x}_i)$$

$$K(\mathbf{x}, \mathbf{x}_i) = \sum_{j=0}^{k} \varphi_j(\mathbf{x})\varphi_j(\mathbf{x}_i) \quad \text{for all } i = 1, 2, \ldots, N$$

-the inner-product kernel is a symmetric function of its arguments

$$K(\mathbf{x}, \mathbf{x}_i) = K(\mathbf{x}_i, \mathbf{x}) \quad \text{for all } i$$

Important:
We may use the inner-product kernel $K(\mathbf{x}, \mathbf{x}_i)$ to construct the optimal hyperplane in the feature space without having to consider the feature space itself in explicit form

- the optimal hyperplane is:

$$\sum_{i=1}^{N} \lambda_i \, d_i \boldsymbol{\varphi}^{\mathrm{T}}(\mathbf{x}_i)\boldsymbol{\varphi}(\mathbf{x}) = 0 \;\rightarrow\; \sum_{i=1}^{N} \lambda_i \, d_i K(\mathbf{x}, \mathbf{x}_i) = 0$$

Optimum design of a SVM

- dual form for constrained optimization of SVM

Given the training sample $\{(\mathbf{x}_i, d_i)\}_{i=1}^{N}$ find the Lagrange multipliers $\{\lambda_i\}_{i=1}^{N}$ that maximize the objective (cost) function

$$Q(\boldsymbol{\lambda}) = \sum_{i=1}^{N} \lambda_i - \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} \lambda_i \lambda_j d_i d_j K(\mathbf{x}_i, \mathbf{x}_j)$$

subject to the constrains

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

ii) $0 \leq \lambda_i \leq C \quad \text{for } i = 1,2, \dots, N$

C is a user-specified positive parameter

$$Q(\boldsymbol{\lambda}) = \sum_{i=1}^{N} \lambda_i - \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N} \lambda_i \lambda_j d_i d_j \mathbf{x}_i^T \mathbf{x}_j$$

$$Q(\boldsymbol{\lambda}) = \sum_{i=1}^{N} \lambda_i - \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N} \lambda_i \lambda_j d_i d_j K(\mathbf{x}_i, \mathbf{x}_j)$$

- the inner-product kernel $K(\mathbf{x}_i, \mathbf{x}_j)$ may be viewed as the *ij*-th element of a symmetric *N-by-N* matrix **K**

$$\mathbf{K} = \{K(\mathbf{x}_i, \mathbf{x}_j)\}_{(i,j)=1}^{N}$$

- having found the optimal values of the Lagrange multipliers $\lambda_{0,i}$ we may determine $\mathbf{w}_0$:

$$\mathbf{w}_0 = \sum_{i=1}^{N} \lambda_{o,i}\, d_i \boldsymbol{\varphi}(\mathbf{x}_i)$$

where the first component of $\mathbf{w}_o$ (i.e. w_o) represents the optimum bias b_o

- the inner-product kernel can be any continuous symmetric function defined on the closed interval (Marcer’s theorem, 1908)

- three common types of the inner-product kernels are used for SVMs:

1. polynomial

$(\mathbf{x}^T\mathbf{x}_i + 1)^p$ power p is specified a priori by the user

2. radial-basis

$$\exp\left(-\frac{1}{2\sigma^2}\|\mathbf{x} - \mathbf{x}_i\|^2\right)$$

3. sigmoid

$$\tanh\,(\beta_0\mathbf{x}^T\mathbf{x}_i + \beta_1)$$

Marcer's theorem is satisfied only for some values of β_0 and β_1 (e.g. $\beta_0 = 2$ and $\beta_1 = 1$)

Example:

Training set: $\mathbf{x}_1 = (-1, -1)^T$ $d_1 = -1$; $\mathbf{x}_1 \in \omega_2$

$\mathbf{x}_2 = (-1, +1)^T$ $d_2 = +1$; $\mathbf{x}_2 \in \omega_1$

$\mathbf{x}_3 = (+1, -1)^T$ $d_3 = +1$; $\mathbf{x}_3 \in \omega_1$

$\mathbf{x}_4 = (+1, +1)^T$ $d_4 = -1$; $\mathbf{x}_4 \in \omega_2$

-let us select $K(\mathbf{x}, \mathbf{x}_i) = (1 + \mathbf{x}^T\mathbf{x}_i)^2$ where $\mathbf{x} = (x_1, x_2)^T$ and

$\mathbf{x}_i=(x_{i1}, x_{i2})^T$

-inner-product kernel

$$K(\mathbf{x},\mathbf{x}_i) = (1 + \begin{bmatrix} x_1, & x_2 \end{bmatrix} \begin{bmatrix} x_{i1} \\ x_{i2} \end{bmatrix})^2 = (1 + (x_1x_{i1} + x_2x_{i2}))^2$$

$$K(\mathbf{x},\mathbf{x}_i) = 1 + x_1^2x_{i1}^2 + 2x_1x_2x_{i1}x_{i2} + x_2^2x_{i2}^2 + 2x_1x_{i1} + 2x_2x_{i2}$$

$$K(\mathbf{x},\mathbf{x}_i) = \boldsymbol{\varphi}^T(\mathbf{x})\boldsymbol{\varphi}(\mathbf{x}_i)$$

$$K(\mathbf{x}, \mathbf{x}_i) = 1 + x_1^2 x_{i1}^2 + 2x_1 x_2 x_{i1} x_{i2} + x_2^2 x_{i2}^2 + 2x_1 x_{i1} + 2x_2 x_{i2}$$

$$K(\mathbf{x}, \mathbf{x}_i) = \boldsymbol{\varphi}^T(\mathbf{x}) \boldsymbol{\varphi}(\mathbf{x}_i)$$

$$\boldsymbol{\varphi}(\mathbf{x}) = [1, x_1^2, \sqrt{2} x_1 x_2, x_2^2, \sqrt{2} x_1, \sqrt{2} x_2]^{\mathrm{T}}$$

$$\boldsymbol{\varphi}(\mathbf{x}_i) = [1, x_{i1}^2, \sqrt{2} x_{i1} x_{i2}, x_{i2}^2, \sqrt{2} x_{i1}, \sqrt{2} x_{i2}]^{\mathrm{T}}$$

$\mathbf{K} = \{K(\mathbf{x}_i, \mathbf{x}_j)\}_{(i,j)=1}^{N}$ is an *N*-by-*N* matrix, where *ij*-th represents K($\mathbf{x}_i$, $\mathbf{x}_j$)

$$\mathbf{x}_1 = (-1, -1)^T$$

$$\boldsymbol{\varphi}(\mathbf{x}_1) = [1, 1, \sqrt{2}, 1, -\sqrt{2}, -\sqrt{2}]^{\mathrm{T}}$$

$$K(\mathbf{x}_1, \mathbf{x}_1) = [1, 1, \sqrt{2}, 1, -\sqrt{2}, -\sqrt{2}][1, 1, \sqrt{2}, 1, -\sqrt{2}, -\sqrt{2}]^{\mathrm{T}}$$

$$K(\mathbf{x}_1, \mathbf{x}_1) = (1 + 1 + 2 + 1 + 2 + 2) = 9$$

$$\mathbf{x}_2 = (-1, +1)^T$$

$$\boldsymbol{\varphi}(\mathbf{x}_2) = [1, 1, -\sqrt{2}, 1, -\sqrt{2}, +\sqrt{2}]^{\mathrm{T}}$$

$$K(\mathbf{x}_1, \mathbf{x}_2) = \left[1, 1, \sqrt{2}, 1, -\sqrt{2}, -\sqrt{2}\right]\left[1, 1, -\sqrt{2}, 1, -\sqrt{2}, +\sqrt{2}\right]^{\mathrm{T}}$$

$$K(\mathbf{x}_1, \mathbf{x}_2) = (1 + 1 + 2 + 1 - 2 - 2) = 1$$

$$\mathbf{x}_3 = (+1, -1)^{\mathrm{T}}$$

$$\mathbf{x}_4 = (+1, +1)^{\mathrm{T}}$$

$$\boldsymbol{\varphi}(\mathbf{x}_3) = [1, 1, -\sqrt{2}, 1, +\sqrt{2}, -\sqrt{2}]^{\mathrm{T}}$$

$$\boldsymbol{\varphi}(\mathbf{x}_4) = [1,1, \sqrt{2}, 1, \sqrt{2}, \sqrt{2}]^{\mathrm{T}}$$

$$K(\mathbf{x}_3, \mathbf{x}_4) = [1, 1, -\sqrt{2}, 1, +\sqrt{2}, -\sqrt{2}]\ [1, 1, \sqrt{2}, 1, \sqrt{2}, \sqrt{2}]^{\mathrm{T}}$$

$$K(\mathbf{x}_3, \mathbf{x}_4) = (1 + 1 - 2 + 1 + 2 - 2) = 1$$

$$\mathbf{K} = \begin{bmatrix} 9 & 1 & 1 & 1 \\ 1 & 9 & 1 & 1 \\ 1 & 1 & 9 & 1 \\ 1 & 1 & 1 & 9 \end{bmatrix}$$

$$Q(\boldsymbol{\lambda}) = \sum_{i=1}^{N} \lambda_i - \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} \lambda_i \lambda_j d_i d_j K(\mathbf{x}_i, \mathbf{x}_j)$$

$$Q(\boldsymbol{\lambda}) = (\lambda_1 + \lambda_2 + \lambda_3 + \lambda_4) - \frac{1}{2}(9\lambda_1^2 - 2\lambda_1\lambda_2 - 2\lambda_1\lambda_3 + 2\lambda_1\lambda_4 +$$

$$+9\lambda_2^2 + 2\lambda_2\lambda_3 - 2\lambda_2\lambda_4 + 9\lambda_3^2 - 2\lambda_3\lambda_4 + 9\lambda_4^2)$$

$$\lambda_1\lambda_1 d_1 d_1 \, K(\mathbf{x}_1, \mathbf{x}_1) = \lambda_1^2 \, (-1)^2 \, 9 = 9\lambda_1^2$$

Lagrangin function $L(\boldsymbol{\lambda},\mu) = Q(\boldsymbol{\lambda}) + \mu\sum_{i=1}^{4}\lambda_i d_i$

$$L(\boldsymbol{\lambda},\mu) = (\lambda_1+\lambda_2+\lambda_3+\lambda_4) - \frac{1}{2}(9\lambda_1^2 - 2\lambda_1\lambda_2 - 2\lambda_1\lambda_3 + 2\lambda_1\lambda_4 +$$

$$+9\lambda_2^2 + 2\lambda_2\lambda_3 - 2\lambda_2\lambda_4 + 9\lambda_3^2 - 2\lambda_3\lambda_4 + 9\lambda_4^2) + \mu(-\lambda_1+\lambda_2+\lambda_3-\lambda_4)$$

$$\frac{\partial L(\lambda,\mu)}{\partial\lambda_1} = 1 - 9\lambda_1 + \lambda_2 + \lambda_3 - \lambda_4 - \mu = 0$$

$$\frac{\partial L(\lambda,\mu)}{\partial\lambda_2} = 1 - 9\lambda_2 + \lambda_1 - \lambda_3 + \lambda_4 + \mu = 0$$

$$\frac{\partial L(\lambda,\mu)}{\partial\lambda_3} = 1 + \lambda_1 - \lambda_2 - 9\lambda_3 + \lambda_4 + \mu = 0$$

$$\frac{\partial L(\lambda,\mu)}{\partial\lambda_4} = 1 - \lambda_1 + \lambda_2 + \lambda_3 - 9\lambda_4 - \mu = 0$$

- the optimal values of the Lagrange multipliers are

$$\lambda_{0,1} = \lambda_{0,2} = \lambda_{0,3} = \lambda_{0,4} = \frac{1}{8} \text{ and } \mu = 0$$

- all four input vectors are support vectors
- the optimum value of Q(**$\boldsymbol{\lambda}$**) is:

$$Q_0(\boldsymbol{\lambda}) = \frac{1}{4}$$

- the optimum weight vector is:

$$\mathbf{w}_0 = \sum_{i=1}^{N_s} \lambda_{0,i}\, d_i \boldsymbol{\varphi}(\mathbf{x}_i)$$

$$\mathbf{w}_0 = \frac{1}{8}[-\boldsymbol{\varphi}(\mathbf{x}_1) + \boldsymbol{\varphi}(\mathbf{x}_2) + \boldsymbol{\varphi}(\mathbf{x}_3) - \boldsymbol{\varphi}(\mathbf{x}_4)]$$

$$\mathbf{w}_0 = \frac{1}{8}\left[-\begin{bmatrix} 1 \\ 1 \\ \sqrt{2} \\ 1 \\ -\sqrt{2} \\ -\sqrt{2} \end{bmatrix} + \begin{bmatrix} 1 \\ 1 \\ -\sqrt{2} \\ 1 \\ -\sqrt{2} \\ \sqrt{2} \end{bmatrix} + \begin{bmatrix} 1 \\ 1 \\ -\sqrt{2} \\ 1 \\ \sqrt{2} \\ -\sqrt{2} \end{bmatrix} - \begin{bmatrix} 1 \\ 1 \\ \sqrt{2} \\ 1 \\ \sqrt{2} \\ \sqrt{2} \end{bmatrix} \right] = \begin{bmatrix} 0 \\ 0 \\ -\sqrt{2} \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

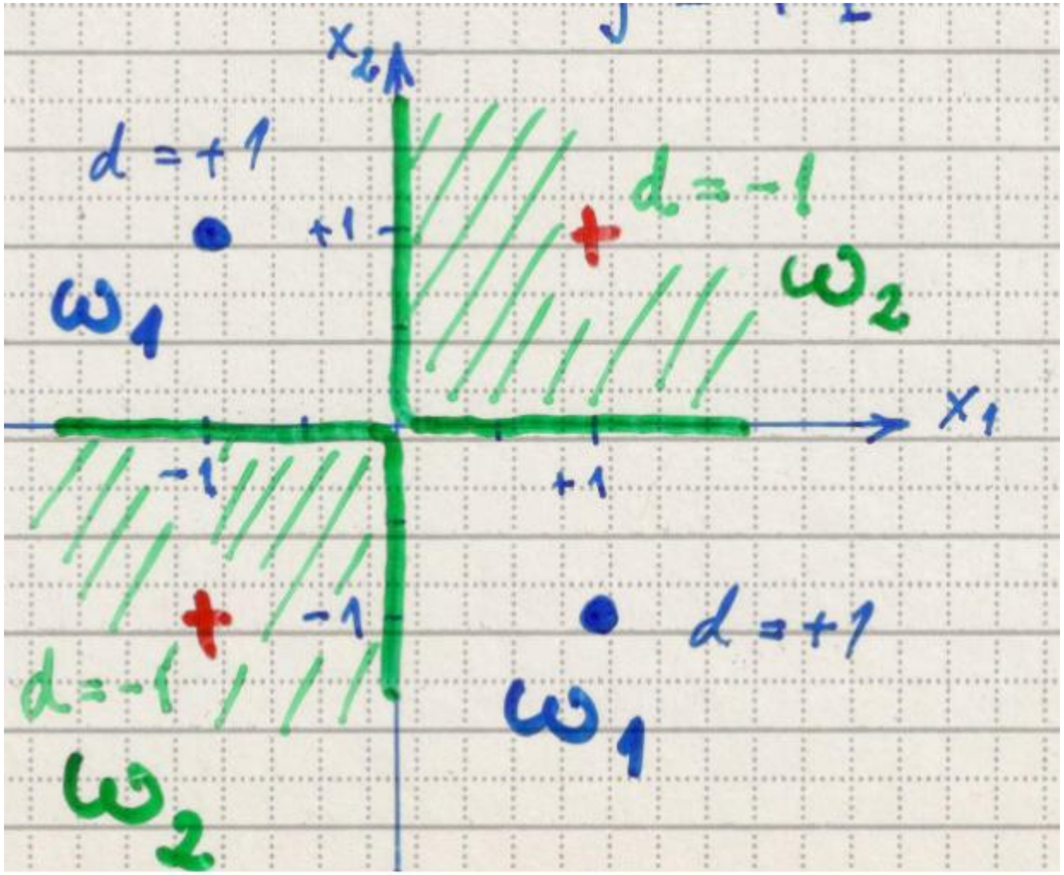
bias *b*

-the optimal hyperplane is: $\mathbf{w}_0^T \boldsymbol{\varphi}(\mathbf{x}) = 0$

$$[0,0,\frac{-\sqrt{2}}{2},0,0,0]\begin{bmatrix} 1 \\ x_1^2 \\ \sqrt{2}x_1x_2 \\ x_2^2 \\ \sqrt{2}x_1 \\ \sqrt{2}x_2 \end{bmatrix} = 0$$

$$-x_1x_2 = 0$$

-for $\mathbf{x}_1$ =(-1, -1)$^T \in \omega_2$ d_1= -1 → -[(-1)(-1)]= -1
-for $\mathbf{x}_2$ =(-1, +1)$^T \in \omega_1$ d_2= +1 → -[(-1)(+1)]= +1
-for $\mathbf{x}_3$ =(+1, -1)$^T \in \omega_1$ d_3= +1 → -[(+1)(-1)]= +1
-for $\mathbf{x}_1$ =(+1, +1)$^T \in \omega_2$ d_4= -1 → -[(+1)(+1)]= -1



x_2
x_1
$d = +1$
$d = -1$
$+1$
-1
ω_1
ω_2
$d = +1$
$d = -1$
ω_1
ω_2

Example:

$\mathbf{x}_1$= $(0, 0)^T$, d_1= +1; $\mathbf{x}_2$ = $(1, 1)^T$, d_2=+1 and
$\mathbf{x}_3$= $(0, 1)^T$, d_3= -1; $\mathbf{x}_4$ = $(1, 0)^T$, d_4 =-1
Inner-product kernel:

$$K(\mathbf{x}, \mathbf{x}_i) = e^{-\alpha\|\mathbf{x} - \mathbf{x}_i\|^2} \quad \alpha = 1$$

Find decision function!

Solution:

$$d(x) = e^{-(x_1^2 + x_2^2)} + e^{-((x_1-1)^2 + (x_2-1)^2)} - e^{-(x_1^2 + (x_2-1)^2)} - e^{-((x_1-1)^2 + x_2^2)} = 0$$