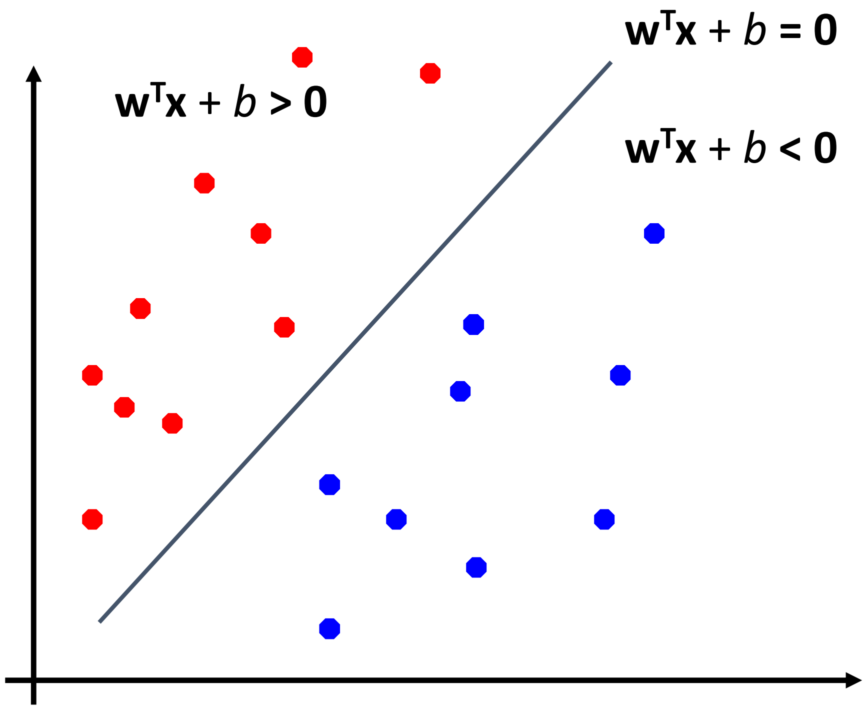
Support Vector Machines

Linear Separators

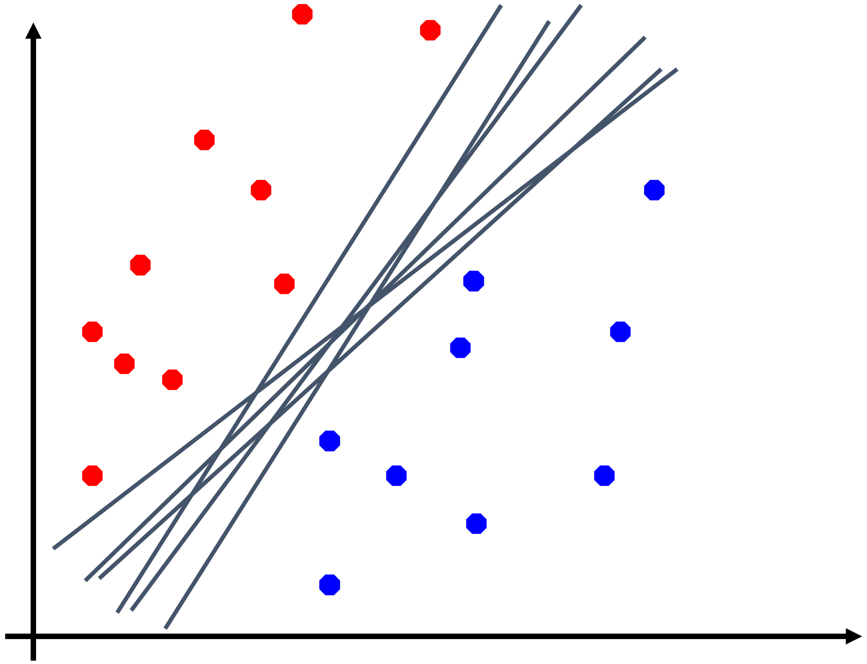
- Binary classification can be viewed as the task of separating classes in feature space:



$f(\mathbf{x}) = \text{sign}(\mathbf{w}^T\mathbf{x} + b)$

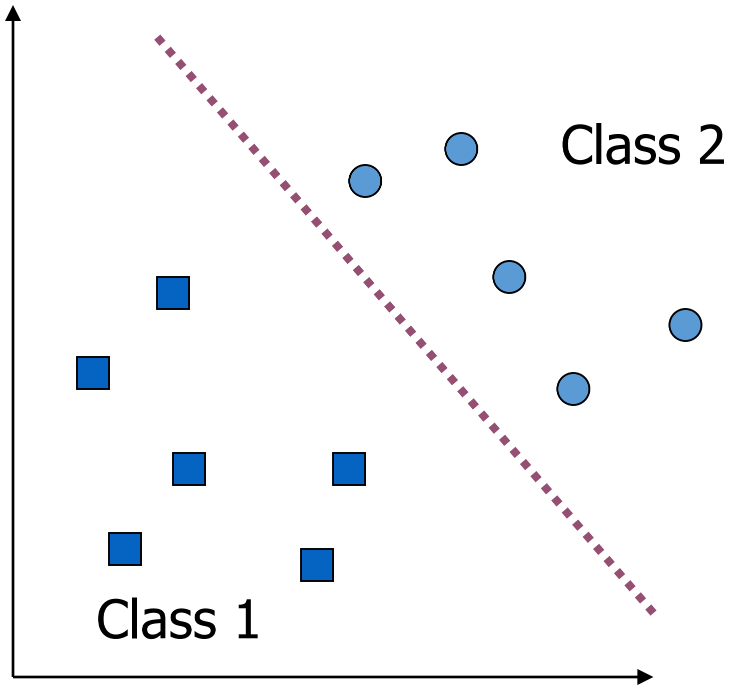
Linear Separators

- Which of the linear separators is optimal?

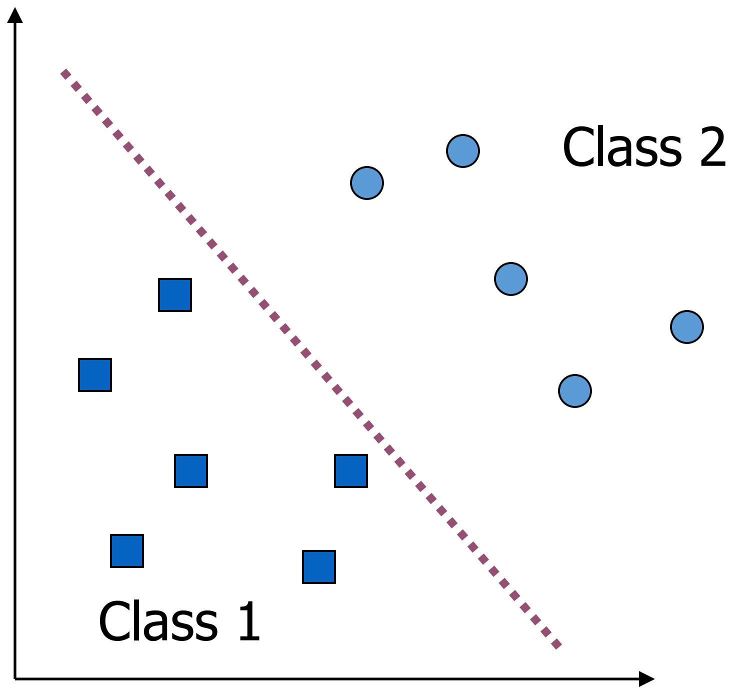


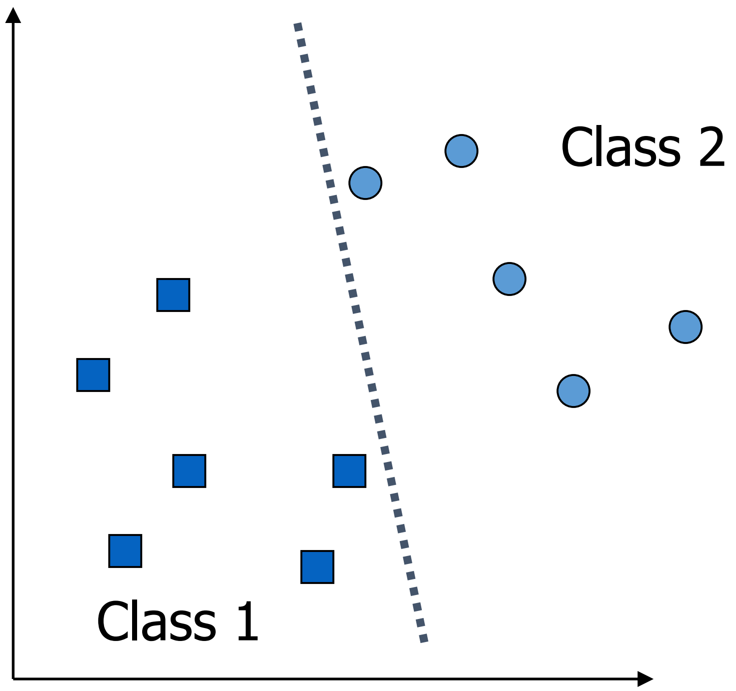
What is a good Decision Boundary?

- Many decision boundaries!
  - The Perceptron algorithm can be used to find such a boundary
- Are all decision boundaries equally good?

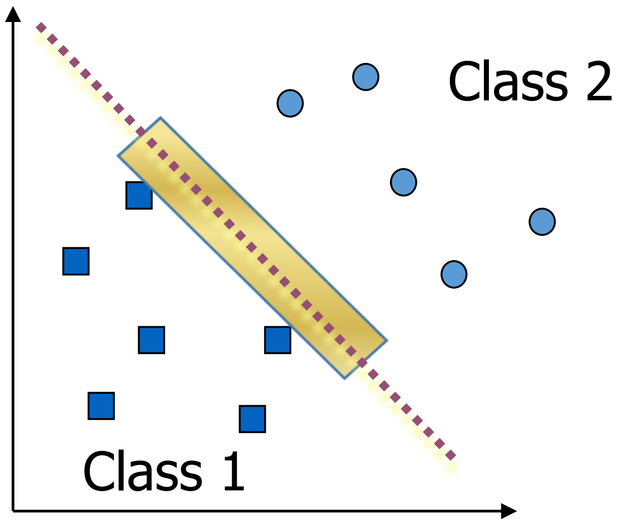


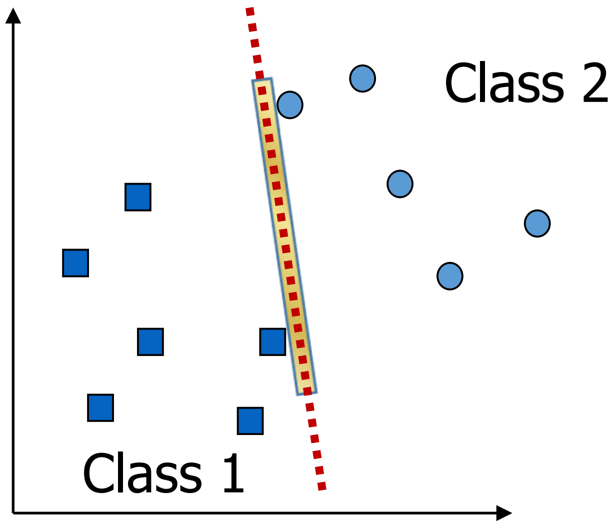
Examples of Bad Decision Boundaries

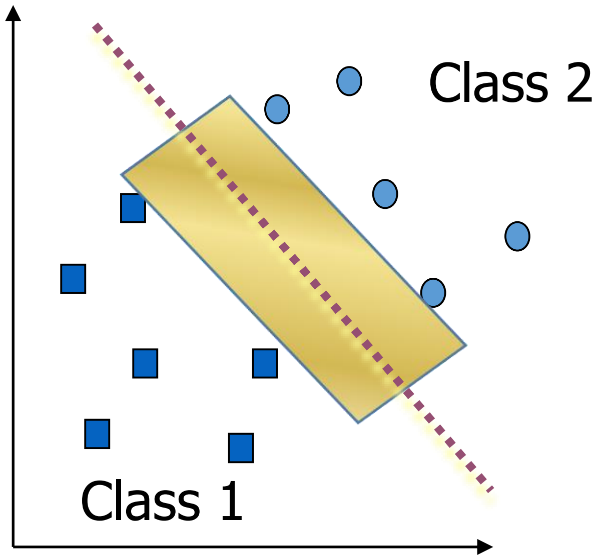




Better Linear Separation

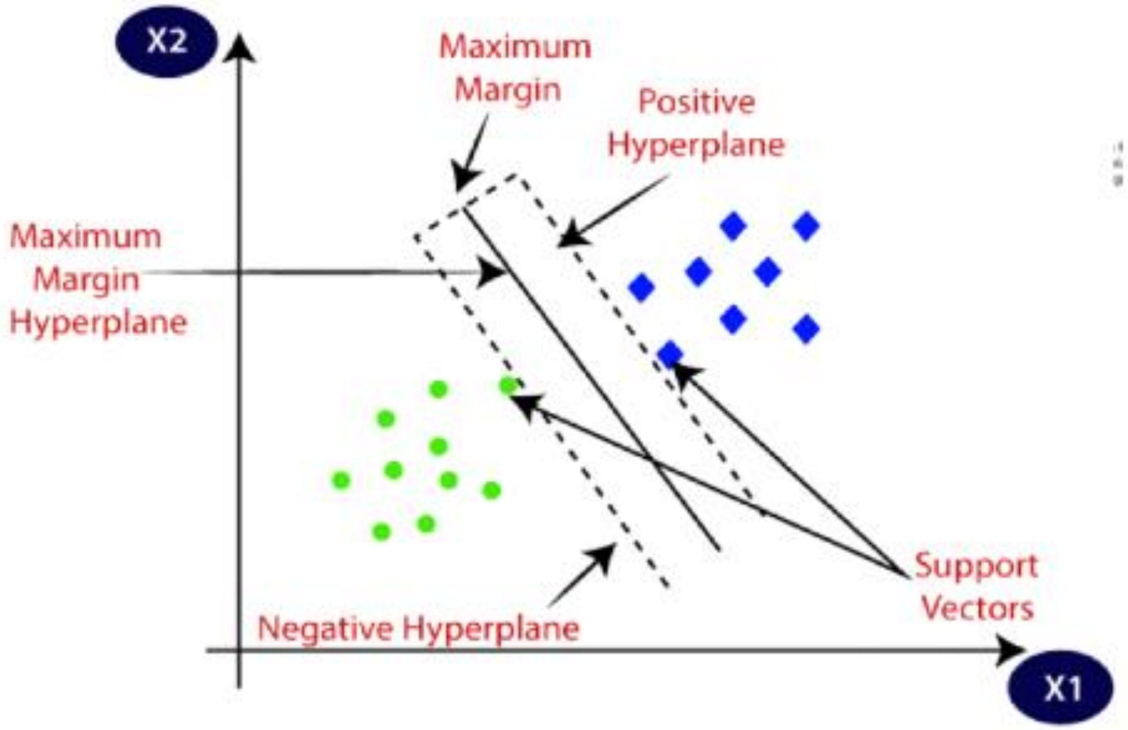




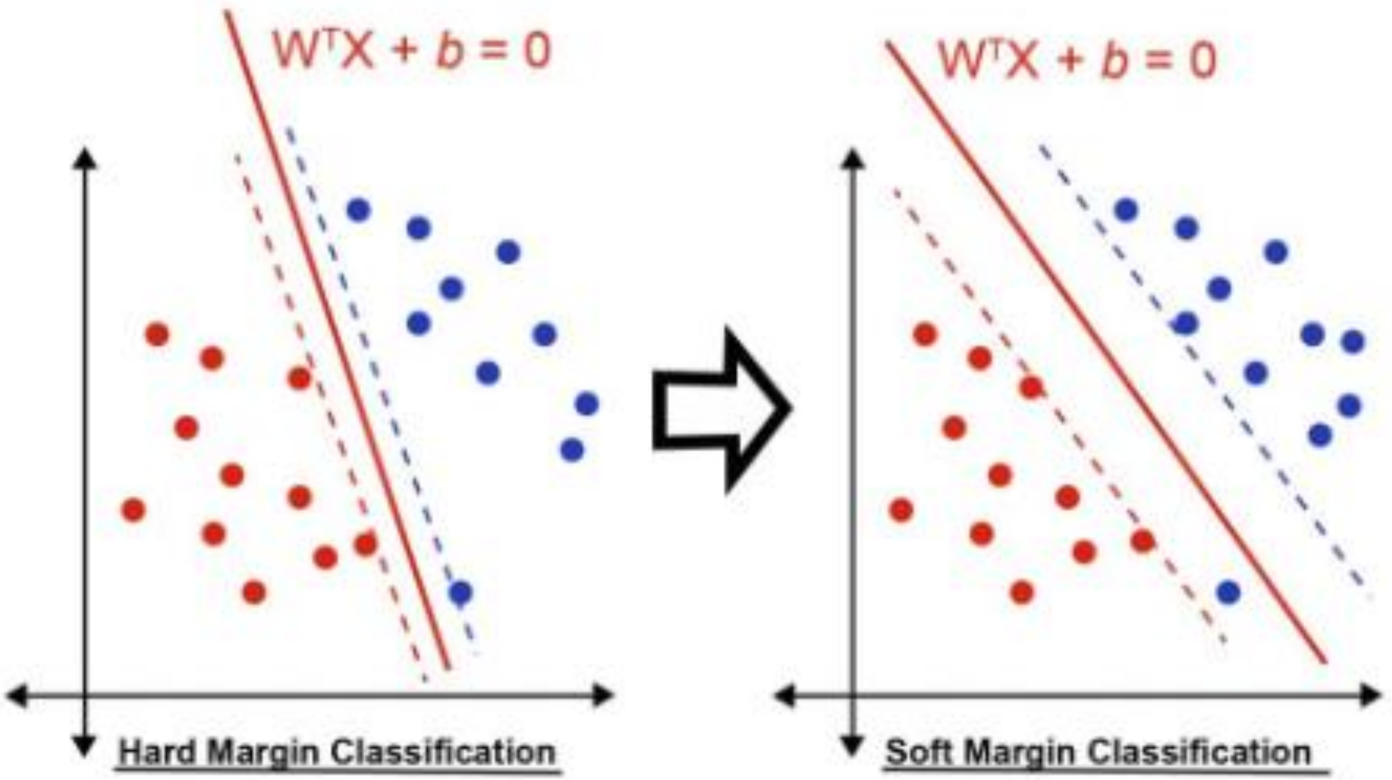


1. Why is bigger margin better?
2. Which **w** maximizes the margin?

SVM Feature



- Support Vectors
- Hyper plane
- Marginal Distance



$W^TX + b = 0$
Hard Margin Classification
$W^TX + b = 0$
Soft Margin Classification

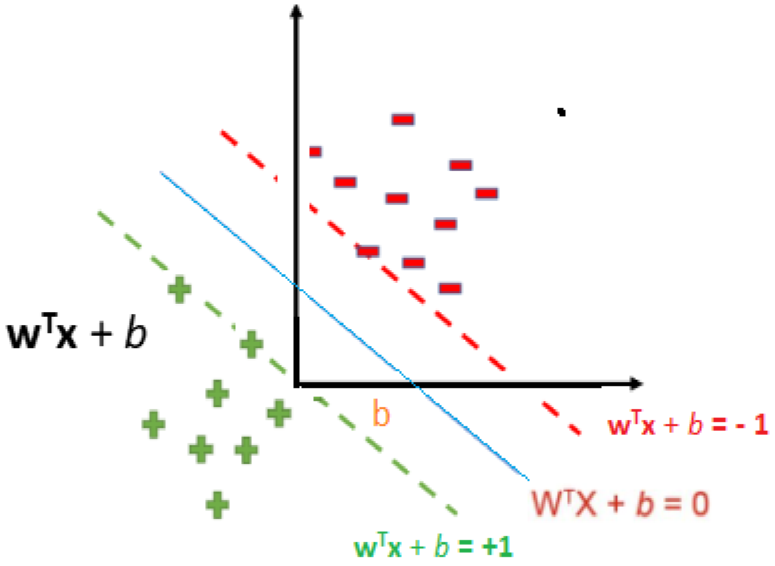
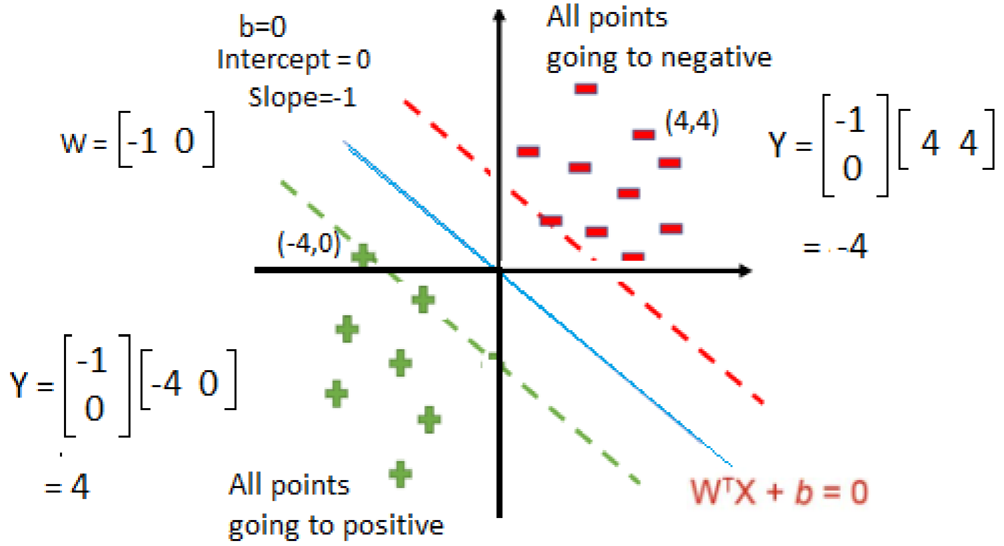


Figure 1



$$w^Tx_- + b = -1$$
$$w^Tx_+ + b = +1$$
$$\text{-------------------}$$
$$w^T(x_+ - x_-) = 2$$
$$\frac{w^T}{||w||}(x_+ - x_-) = \frac{2}{||w||}$$

remove W^T need to divided by Norm of W

Find out (W, b) to Max $\frac{2}{||W||}$

$$Y_i \begin{cases} 1 & w^Tx + b >= +1 \\ -1 & w^Tx + b <= -1 \end{cases} \qquad Y_i(w^Tx + b) >= 1$$

Finding the Decision Boundary

- The decision boundary should classify all points correctly $\Rightarrow$

$$y_i(\mathbf{w}^T\mathbf{x}_i + b) \geq 1, \qquad \forall i$$

- The decision boundary can be found by solving the following constrained optimization problem

$$\text{Minimize } \frac{1}{2}||\mathbf{w}||^2$$
$$\text{subject to } y_i(\mathbf{w}^T\mathbf{x}_i + b) \geq 1 \qquad \forall i$$

- This is a constrained optimization problem. Solving it requires to use Lagrange multipliers

Finding the Decision Boundary

$$\text{Minimize } \frac{1}{2}||\mathbf{w}||^2$$

$$\text{subject to } 1 - y_i(\mathbf{w}^T\mathbf{x}_i + b) \leq 0 \qquad \text{for } i = 1, \ldots, n$$

- The Lagrangian is

$$\mathcal{L} = \frac{1}{2}\mathbf{w}^T\mathbf{w} + \sum_{i=1}^{n} \alpha_i \left(1 - y_i(\mathbf{w}^T\mathbf{x}_i + b)\right)$$

- $\alpha_i \geq 0$
- Note that $||\mathbf{w}||^2 = \mathbf{w}^T\mathbf{w}$

Prerequisite

Optimization Problems using Subject to Constraint

$$MAX_{xy}\, Z \quad where \;\; [z = x^2 y] \qquad STC\; x^2 + y^2 = 1$$

Lagrange Multiplier

$$L(h,s,\lambda) = f(h,s) - \lambda(\mathrm{H}(\mathrm{h},\mathrm{s}))$$

more condtions

$$L(h,s,\lambda) = f(h,s) - \lambda 1(\mathrm{H1}(\mathrm{h},\mathrm{s})) - \lambda 2(\mathrm{H2}(\mathrm{h},\mathrm{s}))$$

Example

$$MAX_{hs}\; 200h^{2/3}s^{1/3}\; f(h,s)$$

$$20h + 170s = 20000\; H(h,s) \;\; [Equality\ condition]$$

$$L(h,s,\lambda) = 200h^{2/3}s^{1/3} - \lambda(20h + 170s - 20000)$$

$$\frac{\partial L}{\partial h} = 200\frac{2}{3}h^{-1/3}s^{1/3} - 20\lambda = 0$$

$$\frac{\partial L}{\partial s} = 200\frac{1}{3}h^{2/3}s^{-2/3} - 170\lambda = 0$$

$$\frac{\partial L}{\partial \lambda} = -20h - 170s + 20000 = 0$$

$$h = 666.66, s = 39.12, \lambda = 2.59$$

$$\max f(hs) = 51777$$

Prerequisite

Karush Kuhn Tucker

KKT Conditions

1. *Convert to Lagrange funtions, partially derive variables and equals to 0*
2. $\lambda_i h^i = 0$
3. $h^i \leq 0$
4. $\lambda_i \geq 0$

$$Max\ -x_1^{\,2} - x_2^{\,2} - x_3^{\,2} + 4x_1 + 6x_2$$

$$STC\ \ x_1 + x_2 \leq 2$$

$$2x_1 + 3x_2 \leq 12$$

$$x_1, x_2 \geq 0$$

Conditions 1:

$$\mathrm{L}(x_1, x_2, x_3, \lambda_1, \lambda_2) = \ -x_1^{\,2} - x_2^{\,2} - x_3^{\,2} + 4x_1 + 6x_2 - \lambda_1(x_1 + x_2 - 2) - \ \lambda_2(2x_1 + 3x_2 - 12)$$

$$\frac{\partial L}{\partial x_1} = 2x_1 + 4 - \ \lambda_1 - 2\lambda_2 = 0\ \ \ldots..(1a)$$

$$\frac{\partial L}{\partial x_2} = 2x_2 + 6 - \ \lambda_1 - 3\lambda_2 = 0\ \ \ldots\ldots(1b)$$

$$\frac{\partial L}{\partial x_3} = 2x_3 = 0\qquad i.e:\ x_3 = 0$$

Conditions 2:

$$\lambda_1(x_1 + x_2 - 2) = 0\ \ \ldots\ldots(2a)$$

$$\lambda_2(2x_1 + 3x_2 - 12) = 0\ \ \ldots\ldots(2b)$$

Conditions 3:

$$x_1 + x_2 - 2\ \leq 0\ \ \ldots\ldots(3a)$$

$$2x_1 + 3x_2 - 12\ \leq 0\ \ \ldots..(3b)$$

Conditions 4:

$$\lambda_1 \geq 0\,, \lambda_2 \geq 0$$

Prerequisite

Karush Kuhn Tucker

Case 1: $\lambda_1 = 0\,,\ \lambda_2 = 0$

Substitute 1a, 1b → $x_1 = 2\,,\ x_2 = 3$

Substitute $x_1\ x_2\ in\ 3a, 3b$ $\quad x_1 + x_2 - 2 \le 0$

$5 - 2 \le 0$

$3 \le 0$ X

$2x_1 + 3x_2 - 12 \le 0$

$1 \le 0$ X

Case 2: $\lambda_1 \neq 0\,,\ \lambda_2 \neq 0$

Means from condition 2

$x_1 + x_2 - 2 = 0, 2x_1 + 3x_2 - 12 = 0\ \ by\ solving\ \ x_2 = 8\,, x_1 = -6$

Substitute in 1a, 1b → Solve $\lambda_1\,,\ \lambda_2$

$\lambda_2 = -26$ X

Case 3: $\lambda_1 = 0\,,\ \lambda_2 \neq 0$

Substitute in 1a, 1b

$-2x_1 + 4 - 2\lambda_2 = 0$

$-2x_1 + 6 - 3\lambda_2 = 0$, solving $x_1 = \frac{2}{3}x_2$

$\lambda_2 \neq 0\ , so$

$2x_1 + 3x_2 - 12 = 0$

$\frac{4}{3}x_1 + 3x_2 - 12 = 0$

$x_1 = 2\,,\ x_2 = 3$

$x_1 + x_2 - 2 \le 0$

$5 - 2 \le 0$ X

$2x_1 + 3x_2 - 12 \le 0$

$4 + 9 - 12 \le 0$ X

Case 4: $\lambda_1 \neq 0\,,\ \lambda_2 = 0$

$\lambda_1 = 3\,,\ \lambda_2 = 0 \quad , \ x_1 = \frac{1}{2}\ , x_2 = {}^{3}/_{2}$ ✓

$x_1 + x_2 - 2 \le 0\quad (0 \le 0)\quad 2x_1 + 3x_2 - 12\ (-13 \le 0)$

Prerequisite

Primal and dual problem for understanding support vector machine:

$M_{in} f(w)$

STC $g_i(w) \leq 0 \quad i = 1 \ldots\ldots k$

$h_i(w) = 0 \quad i = 1 \ldots\ldots l$

Generalized Lagrange function:

L $(w, \alpha, \beta) = f(w) + \sum_{i=1}^{k} \alpha_i g_i(w) + \sum_{i=1}^{l} \beta_i h_i(w)$

Define: $\theta_p(w) = Max_{\alpha\,\beta, \alpha \leq 0}\, L(w, \alpha, \beta)$

$\theta_p(w) = Max_{\alpha\,\beta, \alpha \leq 0}\, f(w) + \sum_{i=1}^{k} \alpha_i g_i(w) + \sum_{i=1}^{l} \beta_i h_i(w)$

If $g_i(w) > 0$ $[violates\ condition]$ $\quad \theta_p(w) = \infty$

If $h_i(w) \neq 0$ $[violates\ condition]$ $\quad \theta_p(w) = \infty$

If $g_i(w), h_i(w)$ $[satisfies\ condition]$ $\quad \theta_p(w) = f(w)$

So, $\theta_p(w) = \begin{cases} f(w) \rightarrow satisfies \\ \infty \rightarrow violates \end{cases}$

Primal problem:

$p^* = min_w\, \theta_p(w)$

$p^* = min_w Max_{\alpha\,\beta, \alpha \leq 0}\, L(w, \alpha, \beta)$

Dual problem:

$d^* = Max_{\alpha\,\beta, \alpha \leq 0}\, min_w\, L(w, \alpha, \beta)$

$= Max_{\alpha\,\beta, \alpha \leq 0}\, \theta_d(\alpha, \beta)$

$d^* \leq p^*$ But under some conditions $d^* = p^*$

$\exists\ w^* \alpha^* \beta^*$

Where w^* solution to Primal,

$\alpha^* \beta^*$ Solution to Dual,

$d^* = p^*$,

$w^* \alpha^* \beta^*$ Satisfy KKT conditions,

1) Derivative w.r.t variable =0
2) $\alpha_i g_i(w) = 0$
3) $g_i(w) \leq 0$
4) $\alpha_i \geq 0$

Fact: $MaxMinf(x) \leq MinMaxf(x)$ $\quad Example:\ MaxMin\,Sin(x+y) \leq MinMaxSin(x+y)$

Gradient with respect to *w* and *b*

- Setting the gradient of $\mathcal{L}$ w.r.t. **w** and b to zero, we have

$$L = \frac{1}{2} w^T w + \sum_{i=1}^{n} \alpha_i \left(1 - y_i \left(w^T x_i + b\right)\right) =$$

$$= \frac{1}{2} \sum_{k=1}^{m} w^k w^k + \sum_{i=1}^{n} \alpha_i \left(1 - y_i \left(\sum_{k=1}^{m} w^k x_i{}^k + b\right)\right)$$

n: no of examples, m: dimension of the space

$$\begin{cases} \dfrac{\partial L}{\partial w^k} = 0, \forall k \\ \dfrac{\partial L}{\partial b} = 0 \end{cases}$$

$$\mathbf{w} + \sum_{i=1}^{n} \alpha_i (-y_i) \mathbf{x}_i = \mathbf{0} \quad \Rightarrow \quad \mathbf{w} = \sum_{i=1}^{n} \alpha_i y_i \mathbf{x}_i$$

$$\sum_{i=1}^{n} \alpha_i y_i = 0$$

The Dual Problem

- If we substitute $\mathbf{w} = \sum_{i=1}^{n} \alpha_i y_i \mathbf{x}_i$, we have $\mathcal{L}$

$$\mathcal{L} = \frac{1}{2}\sum_{i=1}^{n} \alpha_i y_i \mathbf{x}_i^T \sum_{j=1}^{n} \alpha_j y_j \mathbf{x}_j + \sum_{i=1}^{n} \alpha_i \left(1 - y_i(\sum_{j=1}^{n} \alpha_j y_j \mathbf{x}_j^T \mathbf{x}_i + b)\right)$$

$$= \underline{\frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j \mathbf{x}_i^T \mathbf{x}_j} + \sum_{i=1}^{n} \alpha_i - \underline{\sum_{i=1}^{n} \alpha_i y_i \sum_{j=1}^{n} \alpha_j y_j \mathbf{x}_j^T \mathbf{x}_i} - b\sum_{i=1}^{n} \alpha_i y_i$$

$$= -\frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j \mathbf{x}_i^T \mathbf{x}_j + \sum_{i=1}^{n} \alpha_i$$

Since $\sum_{i=1}^{n} \alpha_i y_i = 0$

- This is a function of α_i only

The Dual Problem

- The new objective function is in terms of α_i only
- It is known as the dual problem: if we know **w**, we know all α_i; if we know all α_i, we know **w**
- The original problem is known as the primal problem
- The objective function of the dual problem needs to be maximized (comes out from the KKT theory)
- The dual problem is therefore:

$$\max.\ W(\boldsymbol{\alpha}) = \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1,j=1}^{n} \alpha_i \alpha_j y_i y_j \mathbf{x}_i^T \mathbf{x}_j$$

$$\text{subject to } \alpha_i \geq 0, \qquad \sum_{i=1}^{n} \alpha_i y_i = 0$$

Properties of α_i when we introduce the Lagrange multipliers

The result when we differentiate the original Lagrangian w.r.t. b

The Dual Problem

$$\text{max. } W(\boldsymbol{\alpha}) = \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1,j=1}^{n} \alpha_i \alpha_j y_i y_j \mathbf{x}_i^T \mathbf{x}_j$$

$$\text{subject to } \alpha_i \geq 0, \sum_{i=1}^{n} \alpha_i y_i = 0$$

- This is a quadratic programming (QP) problem
  - A global maximum of α_i can always be found

- **w** can be recovered by

$$\mathbf{w} = \sum_{i=1}^{n} \alpha_i y_i \mathbf{x}_i$$

QP Solver provides us α

Solution: $\boldsymbol{\alpha} = \alpha_1, \alpha_2, \ldots, \alpha_N$

- Note: **w** need not be formed explicitly

$$\mathbf{w} = \sum_{n=1}^{N} \alpha_n y_n \mathbf{x}_n$$

KKT Condition: For n = 1, 2, …, N

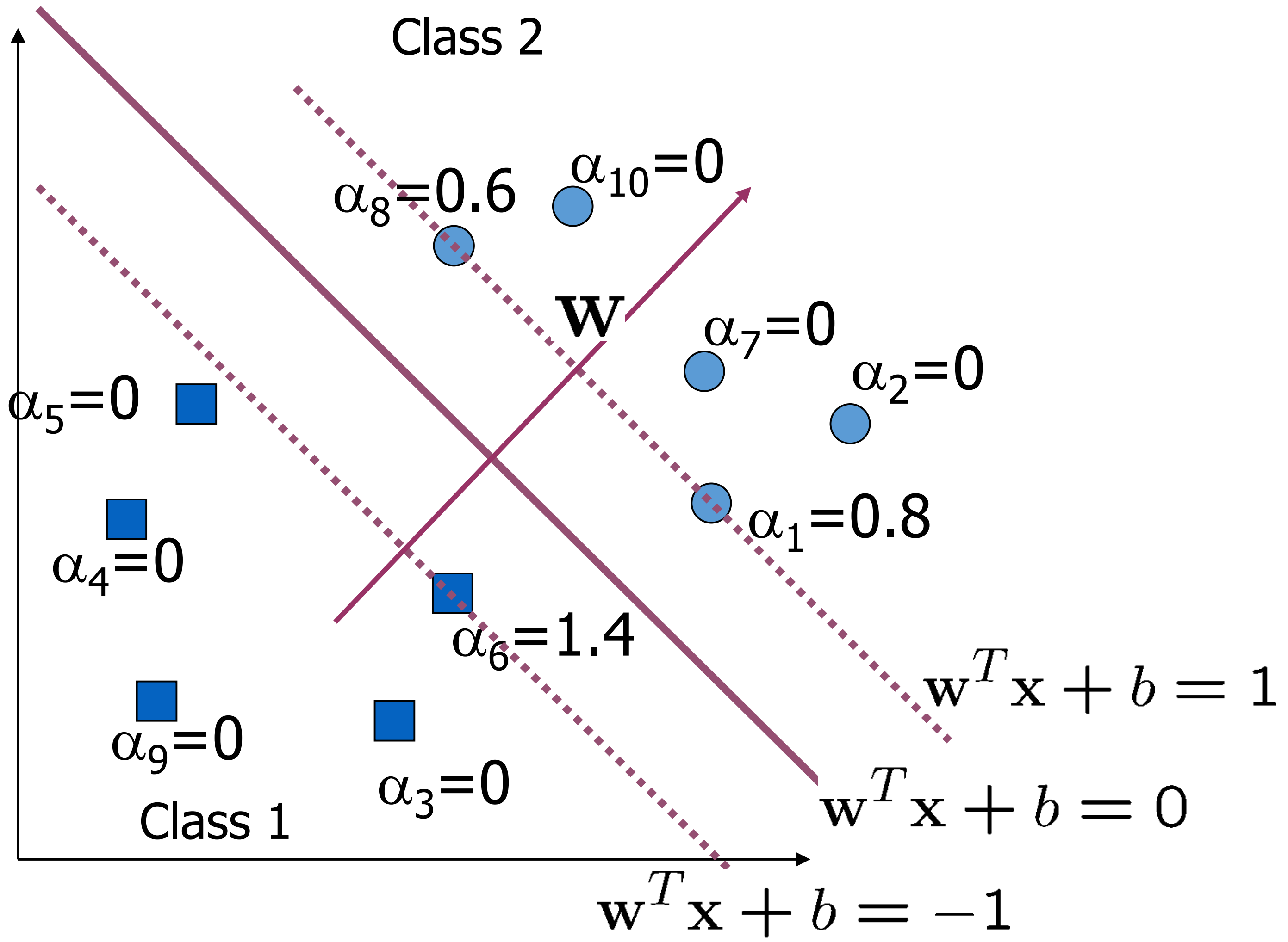
$$\alpha_i\left(1 - y_i\left(w^T x_i + b\right)\right) = 0$$

$\alpha_n > 0 \Rightarrow \mathbf{x_n}$ is **support vector**

Characteristics of the Solution

- For testing with a new data **z**
  - Compute $\mathbf{w}^T\mathbf{z} + b = \sum_{j=1}^{s} \alpha_{t_j} y_{t_j} (\mathbf{x}_{t_j}^T \mathbf{z}) + b$ and classify **z** as class 1 if the sum is positive, and class 2 otherwise
  - Note: **w** need not be formed explicitly

A Geometrical Interpretation



SVM Classifier to Find Hyperplane – Solved Example

X1	X2	Class
2	2	-1 √
4	5	+1
7	4	+1

- $N = 3$
- $\vec{x}_1 = (2, 2)$
- $\vec{x}_2 = (4, 5)$
- $\vec{x}_3 = (7, 4)$
- $y_1 = -1$
- $y_2 = +1$
- $y_3 = +1$

$$f(\vec{x}) = \vec{w} \cdot \vec{x} - b$$

- $\vec{\alpha} = (\alpha_1, \alpha_2, \alpha_3)$
- subject to the conditions
- $\sum_{i=1}^{N} \alpha_i y_i = -\alpha_1 + \alpha_2 + \alpha_3 = 0$
- $\alpha_1 > 0, \alpha_2 > 0, \alpha_3 > 0$

SVM Classifier to Find Hyperplane – Solved Example – Step 1

$$\phi(\vec{\alpha}) = \sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i=1,j=1}^{N} \alpha_i \alpha_j y_i y_j (\vec{x}_i \cdot \vec{x}_j)$$

$$= \sum_{i=1}^{3} \alpha_i - \frac{1}{2} \sum_{i=1,j=1}^{3} \alpha_i \alpha_j y_i y_j (\vec{x}_i \cdot \vec{x}_j)$$

$$(\vec{x}_1 \cdot \vec{x}_1) = 08, \quad (\vec{x}_1 \cdot \vec{x}_2) = 18, \quad (\vec{x}_1 \cdot \vec{x}_3) = 22$$

$$(\vec{x}_2 \cdot \vec{x}_1) = 18, \quad (\vec{x}_2 \cdot \vec{x}_2) = 41, \quad (\vec{x}_2 \cdot \vec{x}_3) = 48,$$

$$(\vec{x}_3 \cdot \vec{x}_1) = 22, \quad (\vec{x}_3 \cdot \vec{x}_2) = 48, \quad (\vec{x}_3 \cdot \vec{x}_3) = 65$$

$$\phi(\vec{\alpha}) = (\alpha_1 + \alpha_2 + \alpha_3) - \frac{1}{2}[8\alpha_1^2 + 41\alpha_2^2 + 65\alpha_3^2 - 36\alpha_1\alpha_2 - 44\alpha_1\alpha_3 + 96\alpha_2\alpha_3]$$

$$\phi(\vec{\alpha}) = 2(\alpha_2 + \alpha_3) - \frac{1}{2}(13\alpha_2^2 + 32\alpha_2\alpha_3 + 29\alpha_3^2)$$

$$N = 3$$

$$\vec{x}_1 = (2, 2)$$

$$\vec{x}_2 = (4, 5)$$

$$\vec{x}_3 = (7, 4)$$

$$y_1 = -1$$

$$y_2 = +1$$

$$y_3 = +1$$

$$-\alpha_1 + \alpha_2 + \alpha_3 = 0$$

SVM Classifier to Find Hyperplane – Solved Example – Step 1

- Find values of α_1, α_2 and α_3 which maximizes

$$\phi(\vec{\alpha}) = 2(\alpha_2 + \alpha_3) - \frac{1}{2}(13\alpha_2^2 + 32\alpha_2\alpha_3 + 29\alpha_3^2)$$

- For $\phi(\vec{\alpha})$ to be maximum we must have

$$\frac{\partial \phi}{\partial \alpha_2} = 0, \quad \frac{\partial \phi}{\partial \alpha_3} = 0$$

- That is,

$$2 - 13\alpha_2 - 16\alpha_3 = 0, \quad 2 - 16\alpha_2 - 29\alpha_3 = 0$$

- Solving these, we get

$$\alpha_2 = \frac{26}{121}, \quad \alpha_3 = -\frac{6}{121} \quad \alpha_1 = \frac{20}{121}$$

$N = 3$

$\vec{x}_1 = (2, 2)$

$\vec{x}_2 = (4, 5)$

$\vec{x}_3 = (7, 4)$

$y_1 = -1$

$y_2 = +1$

$y_3 = +1$

$-\alpha_1 + \alpha_2 + \alpha_3 = 0$

SVM Classifier to Find Hyperplane – Solved Example – Step 2

$$\vec{w} = \sum_{i=1}^{N} \alpha_i y_i \vec{x}_i$$

$$= \frac{20}{121}(-1)(2,2) + \frac{26}{121}(+1)(4,5) - \frac{6}{121}(+1)(7,4)$$

$$= \left(\frac{2}{11}, \frac{6}{11}\right)$$

$$\alpha_1 = \frac{20}{121}$$

$$\alpha_2 = \frac{26}{121}$$

$$\alpha_3 = -\frac{6}{121}$$

$$N = 3$$

$$\vec{x}_1 = (2,2)$$

$$\vec{x}_2 = (4,5)$$

$$\vec{x}_3 = (7,4)$$

$$y_1 = -1$$

$$y_2 = +1$$

$$y_3 = +1$$

$$-\alpha_1 + \alpha_2 + \alpha_3 = 0$$

SVM Classifier to Find Hyperplane – Solved Example – Step 3

$$b = \frac{1}{2}\left(\min_{i:y_i=+1}(\vec{w}\cdot\vec{x}_i) + \max_{i:y_i=-1}(\vec{w}\cdot\vec{x}_i)\right)$$

$$= \frac{1}{2}\left(\min\{(\vec{w}\cdot\vec{x}_2),(\vec{w}\cdot\vec{x}_3)\} + \max\{(\vec{w}\cdot\vec{x}_1)\}\right)$$

$$= \frac{1}{2}\left(\min\{\tfrac{38}{11},\tfrac{38}{11}\} + \max\{\tfrac{16}{11}\}\right)$$

$$= \frac{1}{2}\left(\frac{38}{11} + \frac{16}{11}\right)$$

$$= \frac{27}{11}$$

$$\alpha_1 = \frac{20}{121}$$

$$\alpha_2 = \frac{26}{121}$$

$$\alpha_3 = -\frac{6}{121}$$

$$\vec{w} = \left(\frac{2}{11}, \frac{6}{11}\right)$$

$$N = 3$$

$$\vec{x}_1 = (2,2)$$

$$\vec{x}_2 = (4,5)$$

$$\vec{x}_3 = (7,4)$$

$$y_1 = -1$$

$$y_2 = +1$$

$$y_3 = +1$$

$$-\alpha_1 + \alpha_2 + \alpha_3 = 0$$

SVM Classifier to Find Hyperplane – Solved Example – Step 4

- The SVM classifier function is given by

$$f(\vec{x}) = \vec{w} \cdot \vec{x} - b$$

- Where,

- $\vec{x} = (x_1, x_2)$

$$= \frac{2}{11}x_1 + \frac{6}{11}x_2 - \frac{27}{11}$$

- The equation of the maximal margin hyperplane is

$$f(\vec{x}) = 0 \qquad f(\vec{x}) = \frac{2}{11}x_1 + \frac{6}{11}x_2 - \frac{27}{11}$$

$$\alpha_1 = \frac{20}{121}$$

$$\alpha_2 = \frac{26}{121}$$

$$\alpha_3 = -\frac{6}{121}$$

$$\vec{w} = \left(\frac{2}{11}, \frac{6}{11}\right)$$

$$b = \frac{27}{11}$$

$$N = 3$$

$$\vec{x}_1 = (2, 2)$$

$$\vec{x}_2 = (4, 5)$$

$$\vec{x}_3 = (7, 4)$$

$$y_1 = -1$$

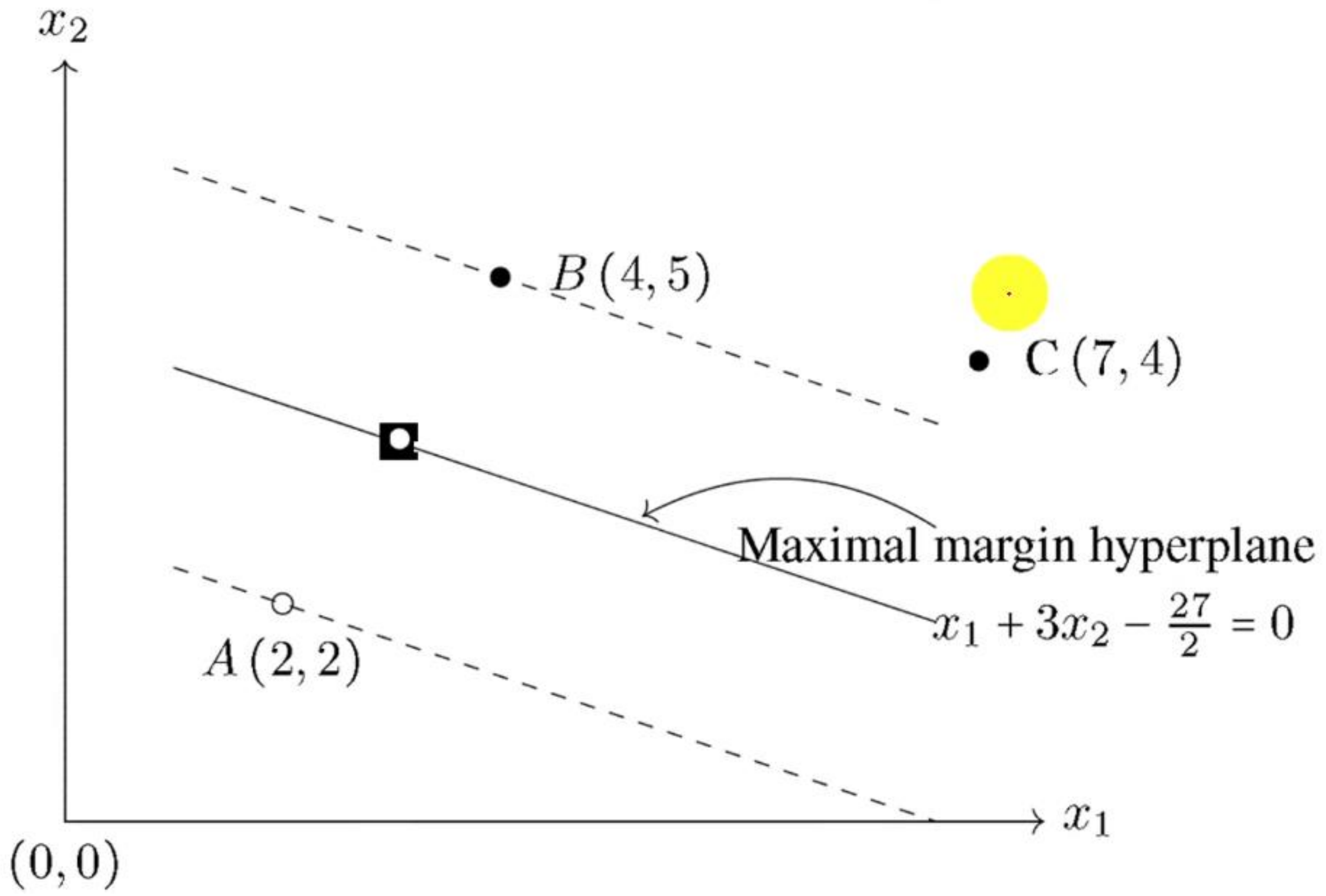
$$y_2 = +1$$

$$y_3 = +1$$

$$-\alpha_1 + \alpha_2 + \alpha_3 = 0$$

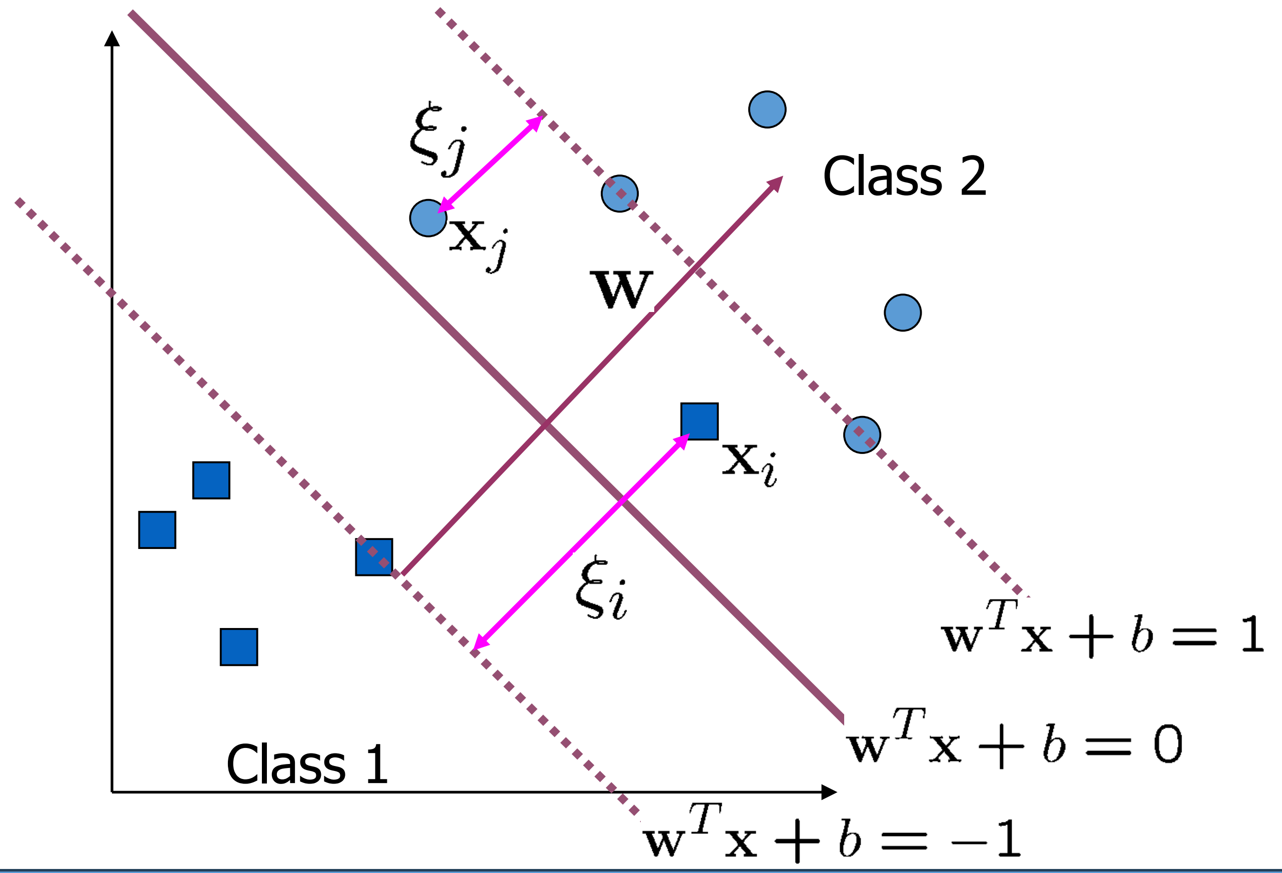
SVM Classifier to Find Hyperplane – Solved Example – Step 5

$$x_1 + 3x_2 - \frac{27}{2} = 0$$



Non-linearly Separable Problems

- We allow "error" ξ_i in classification; it is based on the output of the discriminant function $\mathbf{w}^T\mathbf{x}$+b
- ξ_i approximates the number of misclassified samples



Soft Margin Hyperplane

- The new conditions become

$$\begin{cases} \mathbf{w}^T\mathbf{x}_i + b \geq 1 - \xi_i & y_i = 1 \\ \mathbf{w}^T\mathbf{x}_i + b \leq -1 + \xi_i & y_i = -1 \\ \xi_i \geq 0 & \forall i \end{cases}$$

  - ξ_i are "slack variables" in optimization
  - Note that ξ_i=0 if there is no error for $\mathbf{x}_i$
  - ξ_i is an upper bound of the number of errors
- We want to minimize

$$\frac{1}{2}\|w\|^2 + C\sum_{i=1}^{n}\xi_i$$

$$\text{subject to } y_i(\mathbf{w}^T\mathbf{x}_i + b) \geq 1 - \xi_i, \quad \xi_i \geq 0$$

- *C* : tradeoff parameter between error and margin

The Optimization Problem

$$L = \frac{1}{2} w^T w + C \sum_{i=1}^{n} \xi_i + \sum_{i=1}^{n} \alpha_i \left(1 - \xi_i - y_i \left(w^T x_i + b\right)\right) - \sum_{i=1}^{n} \mu_i \xi_i$$

With α and μ Lagrange multipliers, POSITIVE

$$\frac{\partial L}{\partial w_j} = w_j - \sum_{i=1}^{n} \alpha_i y_i x_{ij} = 0$$

$$\vec{w} = \sum_{i=1}^{n} \alpha_i y_i \vec{x}_i$$

$$\frac{\partial L}{\partial \xi_j} = C - \alpha_j - \mu_j = 0$$

$$\frac{\partial L}{\partial b} = \sum_{i=1}^{n} y_i \alpha_i = 0$$

The Dual Problem

$$L = \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\alpha_i\alpha_j y_i y_j \vec{x}_i^{\,T}\vec{x}_j + C\sum_{i=1}^{n}\xi_i +$$

$$+\sum_{i=1}^{n}\alpha_i\left(1-\xi_i - y_i\left(\sum_{j=1}^{n}\alpha_j y_j x_j^{\,T} x_i + b\right)\right) - \sum_{i=1}^{n}\mu_i\xi_i$$

With $\sum_{i=1}^{n} y_i\alpha_i = 0$ $\qquad C = \alpha_j + \mu_j$

$$L = -\frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\alpha_i\alpha_j y_i y_j \vec{x}_i^{\,T}\vec{x}_j + \sum_{i=1}^{n}\alpha_i$$

The Optimization Problem

- The dual of this new constrained optimization problem is

$$\text{max. } W(\boldsymbol{\alpha}) = \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1,j=1}^{n} \alpha_i \alpha_j y_i y_j \mathbf{x}_i^T \mathbf{x}_j$$

$$\text{subject to } C \geq \alpha_i \geq 0, \sum_{i=1}^{n} \alpha_i y_i = 0$$

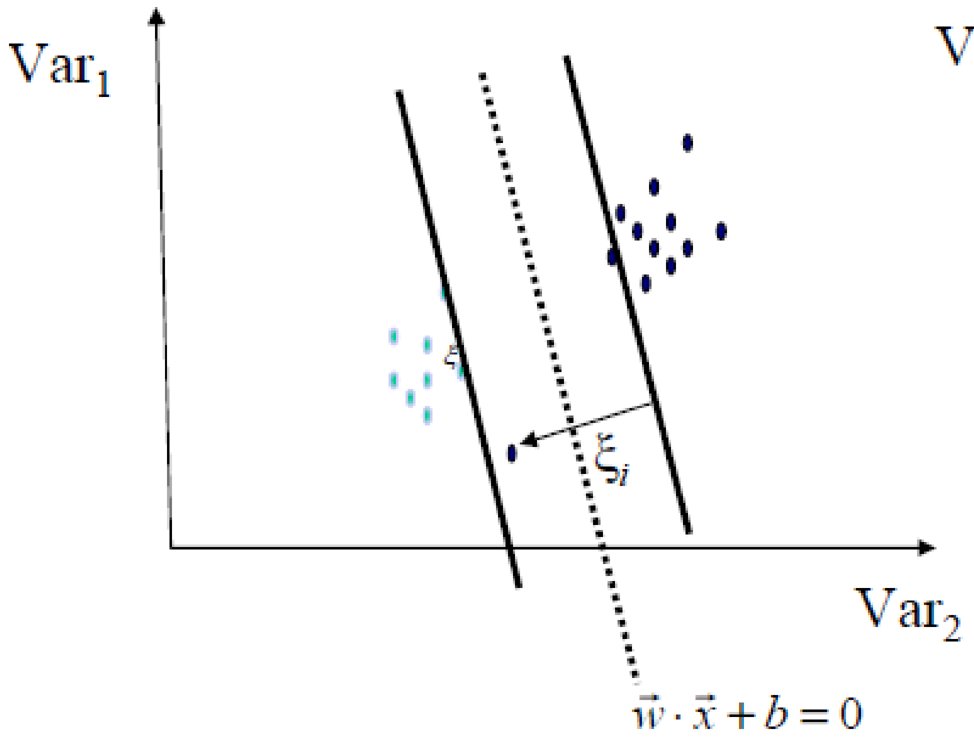
- New constrains derive from $C = \alpha_j + \mu_j$ since μ and α are positive.
- **w** is recovered as $\mathbf{w} = \sum_{j=1}^{s} \alpha_{t_j} y_{t_j} \mathbf{x}_{t_j}$
- This is very similar to the optimization problem in the linear separable case, except that there is an upper bound *C* on α_i now
- Once again, a QP solver can be used to find α_i

$$\frac{1}{2}\|w\|^2 + C\sum_{i=1}^{n}\xi_i$$

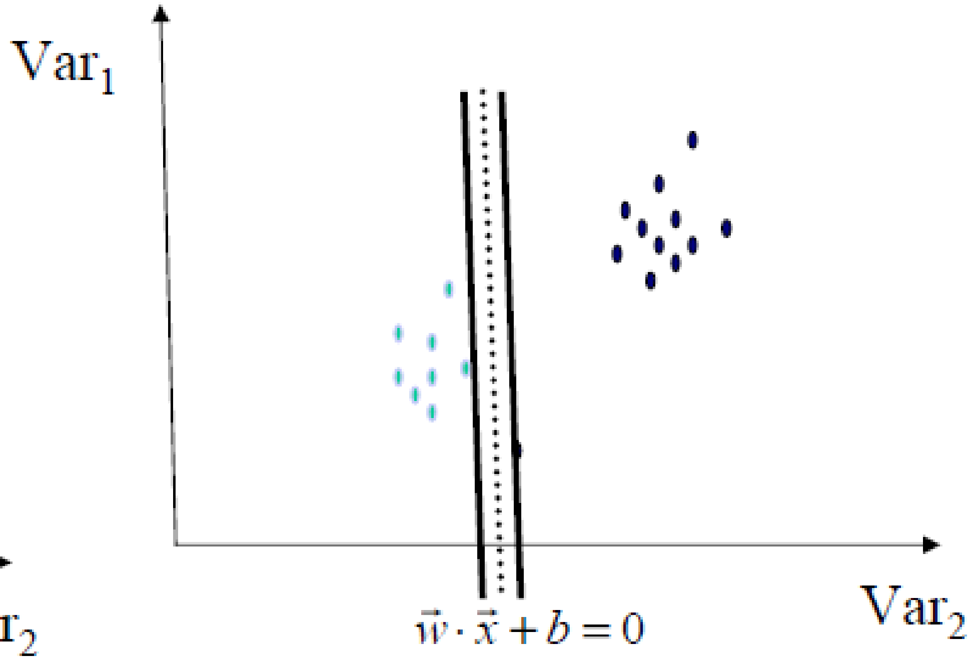
- The algorithm try to keep ξ null, maximising the margin
- The algorithm does not minimise the number of error. Instead, it minimises the sum of distances fron the hyperplane

- When C increases the number of errors tend to lower. At the limit of C tending to infinite, the solution tend to that given by the hard margin formulation, with 0 errors

Soft margin is more robust



Soft Margin SVM



Hard Margin SVM

Extension to Non-linear Decision Boundary

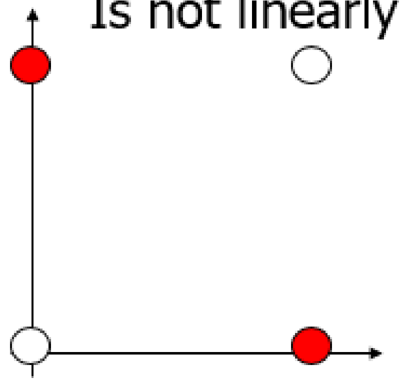
- So far, we have only considered large-margin classifier with a linear decision boundary
- How to generalize it to become nonlinear?
- Key idea: transform $\mathbf{x}_i$ to a higher dimensional space to “make life easier”
  - Input space: the space the point $\mathbf{x}_i$ are located
  - Feature space: the space of $\phi(\mathbf{x}_i)$ after transformation
- Why transform?
  - Linear operation in the feature space is equivalent to non-linear operation in input space
  - Classification can become easier with a proper transformation. In the XOR problem, for example, adding a new feature of x_1x_2 make the problem linearly separable

XOR

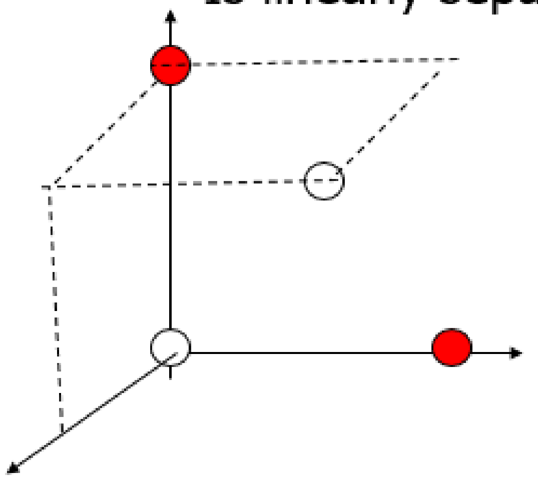
X	Y	
0	0	0
0	1	1
1	0	1
1	1	0

X	Y	XY	
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

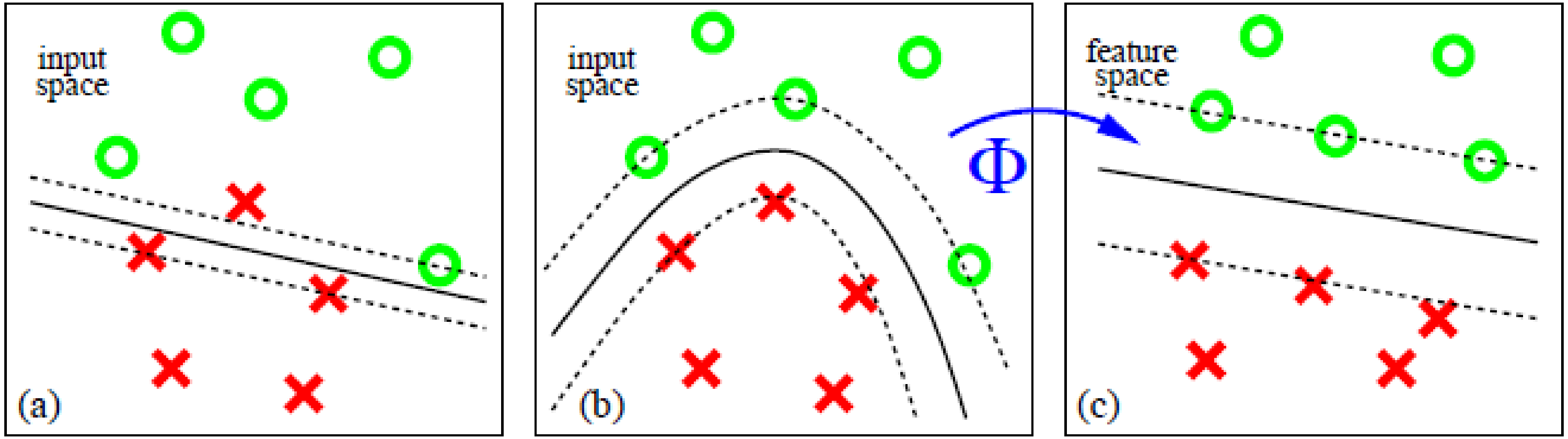
Is not linearly separable



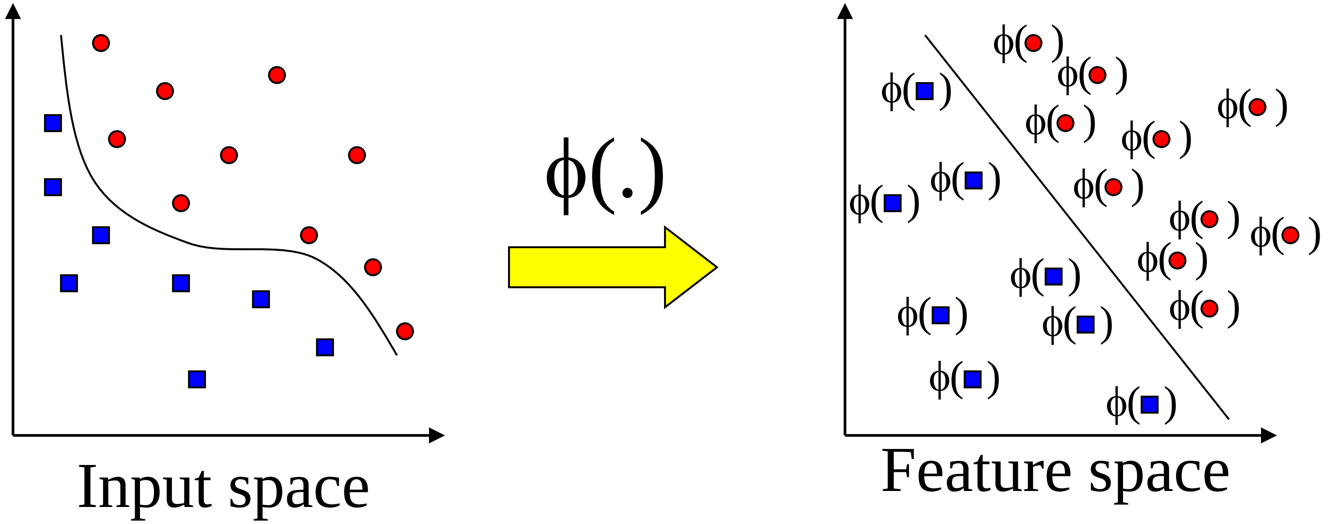
Is linearly separable



Find a feature space



Transforming the Data



Note: feature space is of higher dimension than the input space in practice

- Computation in the feature space can be costly because it is high dimensional
  - The feature space is typically infinite-dimensional!
- The kernel trick comes to rescue

The Kernel Trick

- Recall the SVM (

$$\max.\ W(\boldsymbol{\alpha}) = \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1,j=1}^{n} \alpha_i \alpha_j y_i y_j \mathbf{x}_i^T \mathbf{x}_j$$

$$\text{subject to } C \geq \alpha_i \geq 0,\ \sum_{i=1}^{n} \alpha_i y_i = 0$$

- The data points only appear as inner product
- As long as we can calculate the inner product in the feature space, we do not need the mapping explicitly
- Many common geometric operations (angles, distances) can be expressed by inner products
- Define the kernel function *K* by $K(\mathbf{x}_i, \mathbf{x}_j) = \phi(\mathbf{x}_i)^T \phi(\mathbf{x}_j)$

An Example for ϕ(.) and K(.,.)

- Suppose ϕ(.) is given as follows

$$\phi(\begin{bmatrix} x_1 \\ x_2 \end{bmatrix}) = (1, \sqrt{2}x_1, \sqrt{2}x_2, x_1^2, x_2^2, \sqrt{2}x_1x_2)$$

- An inner product in the feature space is

$$\langle \phi(\begin{bmatrix} x_1 \\ x_2 \end{bmatrix}), \phi(\begin{bmatrix} y_1 \\ y_2 \end{bmatrix}) \rangle = (1 + x_1y_1 + x_2y_2)^2$$

- So, if we define the kernel function as follows, there is no need to carry out ϕ(.) explicitly

$$K(\mathbf{x}, \mathbf{y}) = (1 + x_1y_1 + x_2y_2)^2$$

- This use of kernel function to avoid carrying out ϕ(.) explicitly is known as the kernel trick

Kernels

- Given a mapping: $\mathbf{x} \to \boldsymbol{\varphi}(\mathbf{x})$

a kernel is represented as the inner product

$$K(\mathbf{x},\mathbf{y}) \to \sum_i \varphi_i(\mathbf{x})\varphi_i(\mathbf{y})$$

A kernel must satisfy the Mercer's condition:

$$\forall g(\mathbf{x}) \text{ such that } \int g^2(\mathbf{x})d\mathbf{x} \geq 0 \Rightarrow \int K(\mathbf{x},\mathbf{y})g(\mathbf{x})g(\mathbf{y})d\mathbf{x}d\mathbf{y} \geq 0$$

Modification Due to Kernel Function

- Change all inner products to kernel functions
- For training,

Original

$$\text{max. } W(\boldsymbol{\alpha}) = \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1,j=1}^{n} \alpha_i \alpha_j y_i y_j \mathbf{x}_i^T \mathbf{x}_j$$

$$\text{subject to } C \geq \alpha_i \geq 0, \sum_{i=1}^{n} \alpha_i y_i = 0$$

With kernel function

$$\text{max. } W(\boldsymbol{\alpha}) = \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1,j=1}^{n} \alpha_i \alpha_j y_i y_j K(\mathbf{x}_i, \mathbf{x}_j)$$

$$\text{subject to } C \geq \alpha_i \geq 0, \sum_{i=1}^{n} \alpha_i y_i = 0$$

Modification Due to Kernel Function

- For testing, the new data **z** is classified as class 1 if *f* ≥0, and as class 2 if *f* <0

Original

$$\mathbf{w} = \sum_{j=1}^{s} \alpha_{t_j} y_{t_j} \mathbf{x}_{t_j}$$

$$f = \mathbf{w}^T \mathbf{z} + b = \sum_{j=1}^{s} \alpha_{t_j} y_{t_j} \mathbf{x}_{t_j}^T \mathbf{z} + b$$

With kernel function

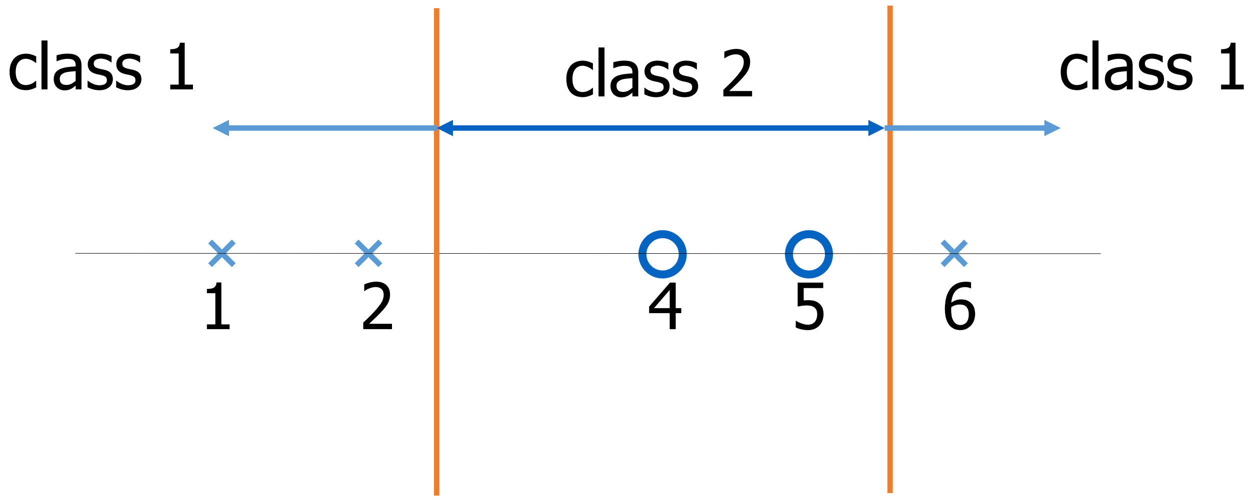
$$\mathbf{w} = \sum_{j=1}^{s} \alpha_{t_j} y_{t_j} \phi(\mathbf{x}_{t_j})$$

$$f = \langle \mathbf{w}, \phi(\mathbf{z}) \rangle + b = \sum_{j=1}^{s} \alpha_{t_j} y_{t_j} K(\mathbf{x}_{t_j}, \mathbf{z}) + b$$

Example

- Suppose we have 5 1D data points
  - $x_1=1$, $x_2=2$, $x_3=4$, $x_4=5$, $x_5=6$, with 1, 2, 6 as class 1 and 4, 5 as class 2 $\Rightarrow$ $y_1=1$, $y_2=1$, $y_3=-1$, $y_4=-1$, $y_5=1$

Example



Example

- We use the polynomial kernel of degree 2
  - $K(x,y) = (xy+1)^2$
  - C is set to 100   first find α_i (i=1, …, 5) by

$$\text{max.} \quad \sum_{i=1}^{5} \alpha_i - \frac{1}{2} \sum_{i=1}^{5} \sum_{j=1}^{5} \alpha_i \alpha_j y_i y_j (x_i x_j + 1)^2$$

$$\text{subject to } 100 \geq \alpha_i \geq 0, \sum_{i=1}^{5} \alpha_i y_i = 0$$

Example

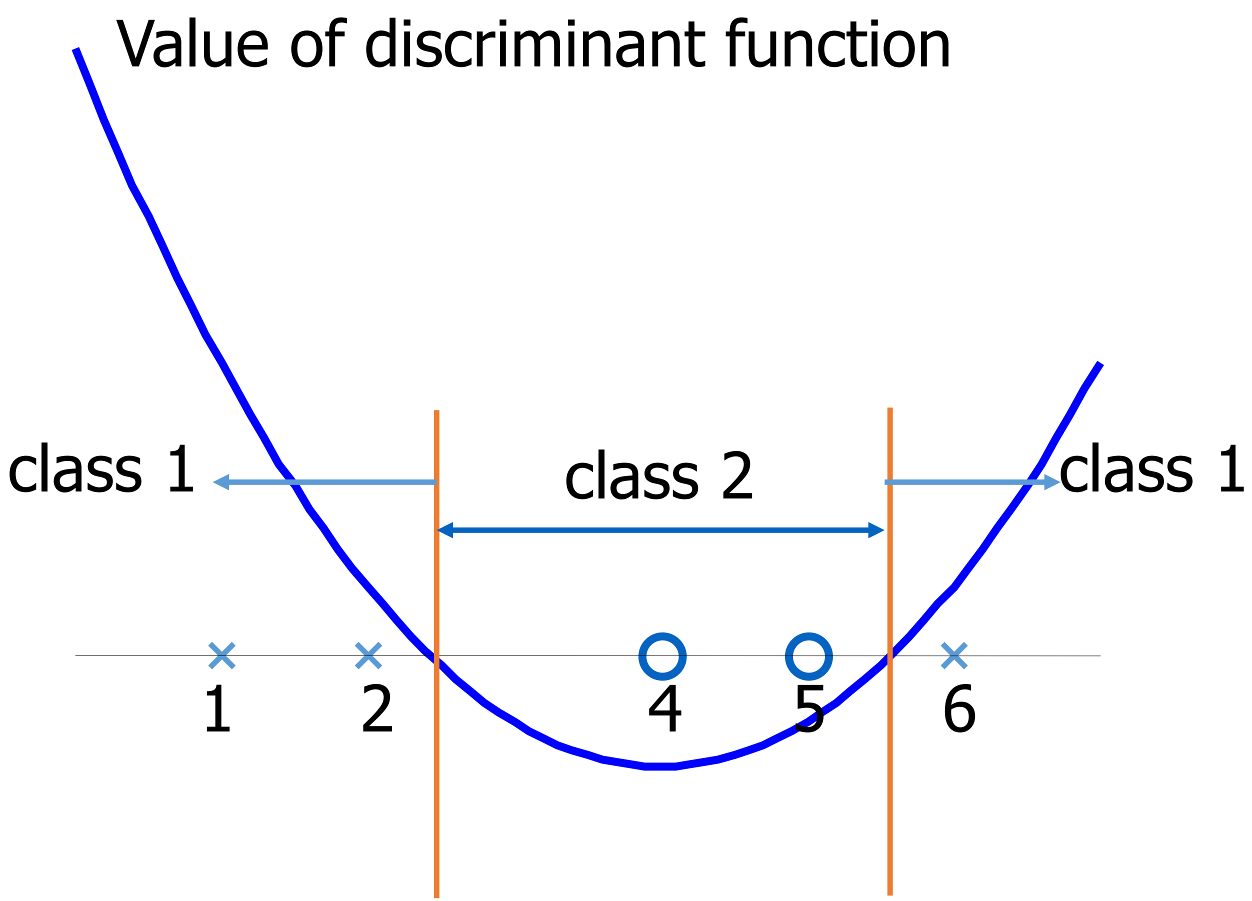
- By using a QP solver, we get
  - α_1=0, α_2=2.5, α_3=0, α_4=7.333, α_5=4.833
  - Note that the constraints are indeed satisfied
  - The support vectors are {x_2=2, x_4=5, x_5=6}
- The discriminant function is

$$
\begin{aligned}
&f(z)\\
&= 2.5(1)(2z+1)^2 + 7.333(-1)(5z+1)^2 + \overset{\alpha_5}{4.833}(\overset{y_5}{1})\overset{K(z,x_5)}{(6z+1)^2} + b\\
&= 0.6667z^2 - 5.333z + b
\end{aligned}
$$

- *b* is recovered by solving f(2)=1 or by f(5)=-1 or by f(6)=1,
- All three give b=9

$$\Rightarrow \quad f(z) = 0.6667z^2 - 5.333z + 9$$

Example



Value of discriminant function
class 1
class 2
class 1
1
2
4
5
6

Kernel Functions

- In practical use of SVM, the user specifies the kernel function; the transformation $\phi(.)$ is not explicitly stated
- Given a kernel function $K(\mathbf{x}_i, \mathbf{x}_j)$, the transformation $\phi(.)$ is given by its eigenfunctions (a concept in functional analysis)
  - Eigenfunctions can be difficult to construct explicitly
  - This is why people only specify the kernel function without worrying about the exact transformation
- Another view: kernel function, being an inner product, is really a similarity measure between the objects

A kernel is associated to a transformation

- Given a kernel, in principle it should be recovered the transformation in the feature space that originates it.

- $K(x,y) = (xy+1)^2 = x^2y^2+2xy+1$

It corresponds the transformation

$$x \rightarrow \begin{pmatrix} x^2 \\ \sqrt{2}x \\ 1 \end{pmatrix}$$

Examples of Kernel Functions

- Polynomial kernel up to degree *d*

$$K(\mathbf{u}, \mathbf{v}) = (\mathbf{u} \cdot \mathbf{v})^d$$

- Polynomial kernel up to degree *d*

$$K(\mathbf{x}, \mathbf{y}) = (\mathbf{x}^T\mathbf{y} + 1)^d$$

- Radial basis function kernel with width σ

$$K(\mathbf{x}, \mathbf{y}) = \exp(-||\mathbf{x} - \mathbf{y}||^2/(2\sigma^2))$$

  - The feature space is infinite-dimensional
- Sigmoid with parameter κ and θ

$$K(\mathbf{x}, \mathbf{y}) = \tanh(\kappa\mathbf{x}^T\mathbf{y} + \theta)$$

  - It does not satisfy the Mercer condition on all κ and θ

Summary: Steps for Classification

- Prepare the pattern matrix
- Select the kernel function to use
- Select the parameter of the kernel function and the value of *C*
  - You can use the values suggested by the SVM software, or you can set apart a validation set to determine the values of the parameter
- Execute the training algorithm and obtain the α_i
- Unseen data can be classified using the α_i and the support vectors

Strengths and Weaknesses of SVM

- Strengths
  - Training is relatively easy
    - No local optimal, unlike in neural networks
  - It scales relatively well to high dimensional data
  - Tradeoff between classifier complexity and error can be controlled explicitly
  - Non-traditional data like strings and trees can be used as input to SVM, instead of feature vectors
- Weaknesses
  - Need to choose a “good” kernel function.

Conclusion

- SVM is a useful alternative to neural networks
- Two key concepts of SVM: maximize the margin and the kernel trick
- Many SVM implementations are available on the web for you to try on your data set!

Thank You!

Original PPT Made by

Prof. Dinabandhu Bhandari
Computer Science and Engineering
Heritage Institute of Technology, Kolkata
Dinabandhu.bhandari@gmail.com