Support vector machine

Ngô Minh Nhựt

2021

Part 1

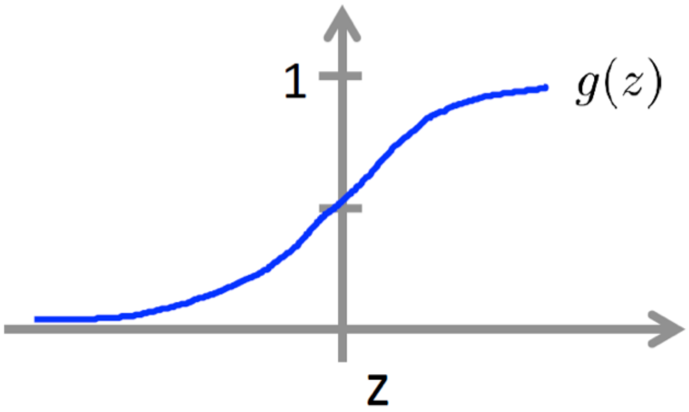
Large margin classifier

Logistic regression

- Hypothesis:
  - $h_\theta(x) = g(\theta^T x)$
  - $h_\theta(x) = P(y = 1|x; \theta)$

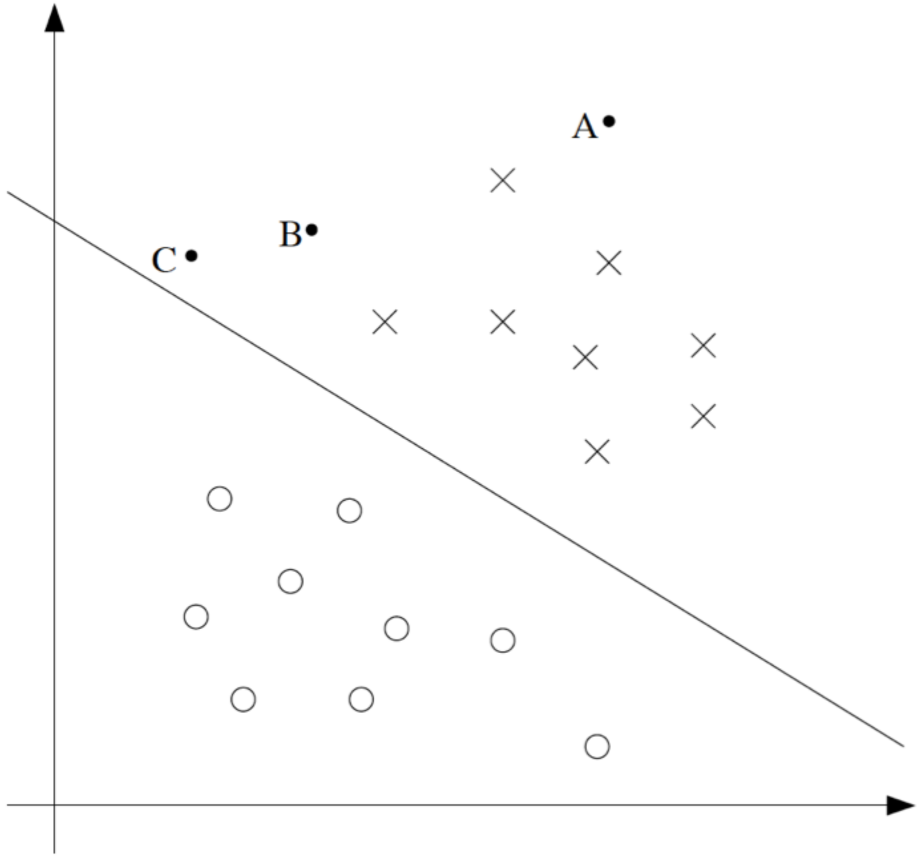
$$g(z) = \frac{1}{1 + e^{-z}}$$

- $\theta^T x \gg 0$
  - Possibility y = 1 is higher
- $\theta^T x \ll 0$
  - Possibility y = 0 is higher



Logistic regression

- Margin

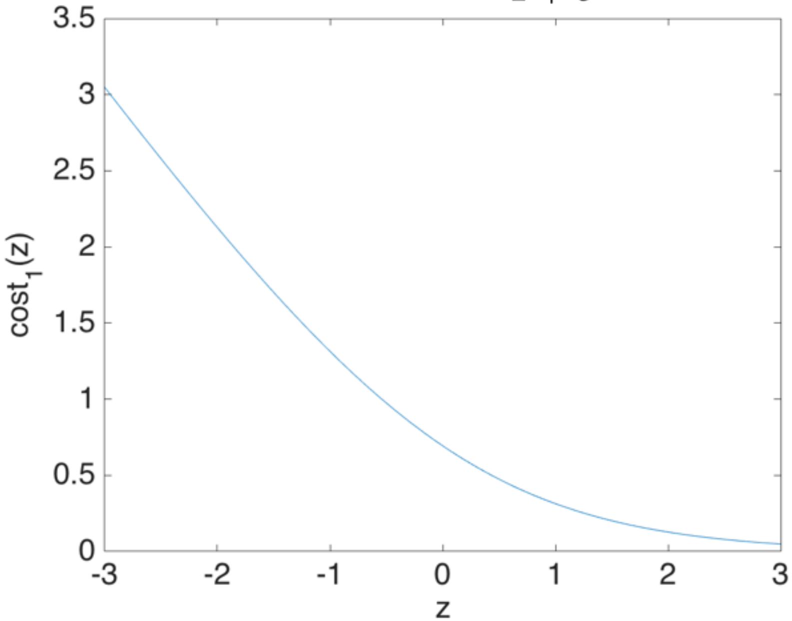


Logistic regression

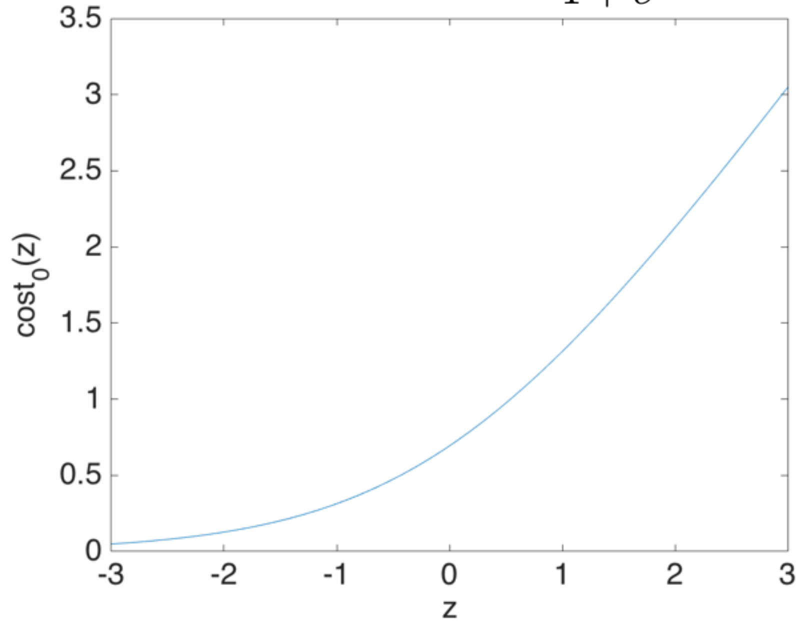
- Cost function for a sample

$$-y\log\frac{1}{1+e^{-\theta^T x}} - (1-y)\log(1-\frac{1}{1+e^{-\theta^T x}})$$

$$y=1, \mathrm{cost} = -\log\frac{1}{1+e^{-\theta^T x}}$$



$$y=0, \mathrm{cost} = -\log(1-\frac{1}{1+e^{-\theta^T x}})$$

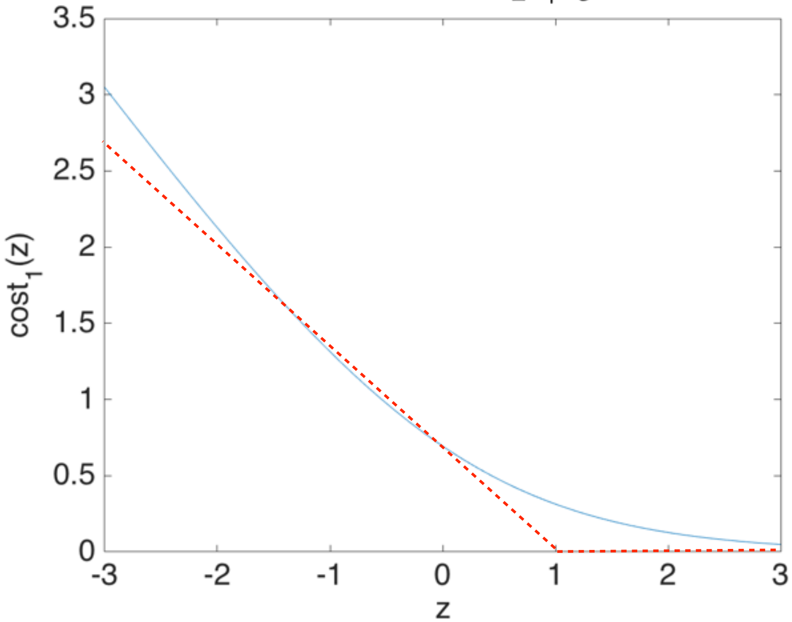


Logistic regression

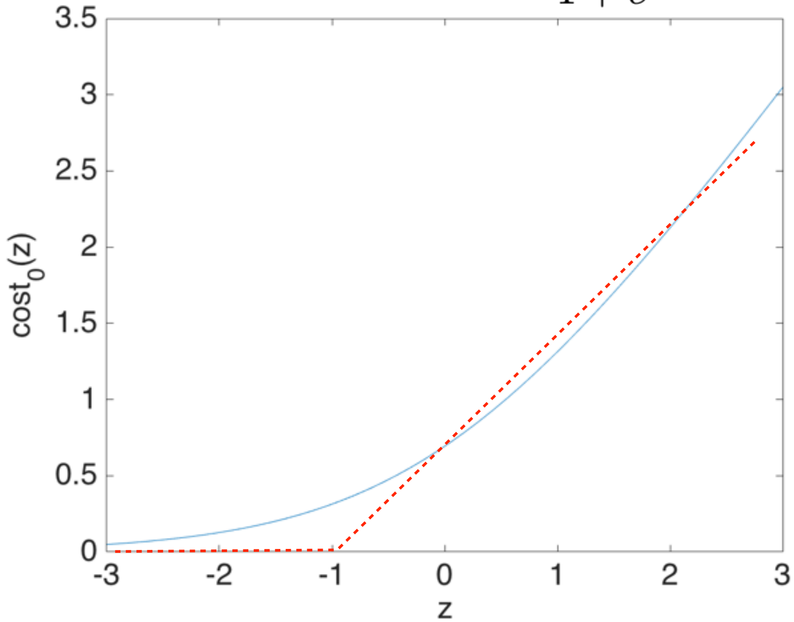
- Cost function for a sample

$$-y\log\frac{1}{1+e^{-\theta^T x}}-(1-y)\log(1-\frac{1}{1+e^{-\theta^T x}})$$

$$y=1, \text{cost}=-\log\frac{1}{1+e^{-\theta^T x}}$$



$$y=0, \text{cost}=-\log(1-\frac{1}{1+e^{-\theta^T x}})$$



Support vector machine

- Cost function

$$J(\theta) = C\sum_{i=1}^{m}\left[y^{(i)}\text{cost}_1(\theta^T x^{(i)}) + (1-y^{(i)})\text{cost}_0(\theta^T x^{(i)})\right] + \frac{1}{2}\sum_{j=1}^{n}\theta_j^2$$

- In comparison with logistic regression

$$J(\theta) = \frac{1}{m}\sum_{i=1}^{m}\left[y^{(i)}(-\log h_\theta(x^{(i)})) + (1-y^{(i)})(-\log(1-h_\theta(x^{(i)})))\right] + \frac{\lambda}{2m}\sum_{j=1}^{n}\theta_j^2$$

SVM model

- Hypothesis
  - $y = 1 \quad \text{if} \quad \theta^T x \geq 1$
  - $y = 0 \quad \text{if} \quad \theta^T x \leq -1$
- Cost function

$$J(\theta) = C \sum_{i=1}^{m} \left[y^{(i)} \text{cost}_1(\theta^T x^{(i)}) + (1 - y^{(i)}) \text{cost}_0(\theta^T x^{(i)}) \right] + \frac{1}{2} \sum_{j=1}^{n} \theta_j^2$$

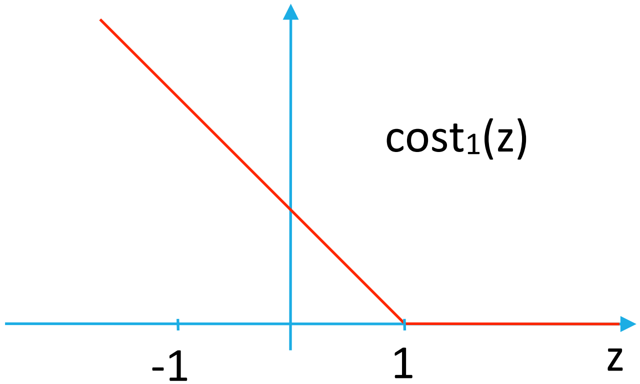
- Learning:

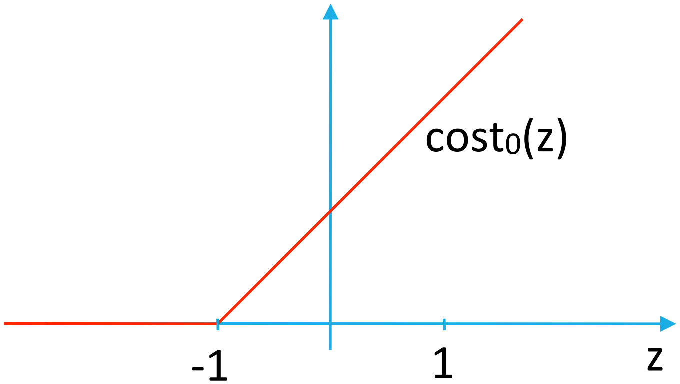
$$\min_{\theta} J(\theta)$$

Learning

- Training:

$$\min_{\theta} C \sum_{i=1}^{m} \left[y^{(i)} \text{cost}_1(\theta^T x^{(i)}) + (1 - y^{(i)}) \text{cost}_0(\theta^T x^{(i)}) \right] + \frac{1}{2} \sum_{j=1}^{n} \theta_j^2$$





- If y = 1, we need $\theta^T x \geq 1$
- If y = 0, we need $\theta^T x \leq -1$

Learing

- Goal

$$\min_\theta C\sum_{i=1}^{m}\left[y^{(i)}\text{cost}_1(\theta^T x^{(i)}) + (1-y^{(i)})\text{cost}_0(\theta^T x^{(i)})\right] + \frac{1}{2}\sum_{j=1}^{n}\theta_j^2$$

  - If y = 1, we need $\theta^T x \geq 1$
  - If y = 0, we need $\theta^T x \leq -1$

➡ $$\min_\theta \frac{1}{2}\sum_{j=1}^{n}\theta_j^2$$

$$\text{s.t.}\quad \theta^T x^{(i)} \geq 1 \quad \text{if} \quad y^{(i)} = 1$$
$$\theta^T x^{(i)} \leq -1 \quad \text{if} \quad y^{(i)} = 0$$

Why large margin

- Dot product

$$u = \begin{bmatrix} u_1 \\ u_2 \end{bmatrix}, \qquad v = \begin{bmatrix} v_1 \\ v_2 \end{bmatrix}$$

$$u^T v = u_1 v_1 + u_2 v_2$$

- Magnitude of vector u: $\|u\| = \sqrt{u_1^2 + u_2^2}$
- Projection of v on u: p
- Then: $u^T v = p\|u\|$

Why large margin

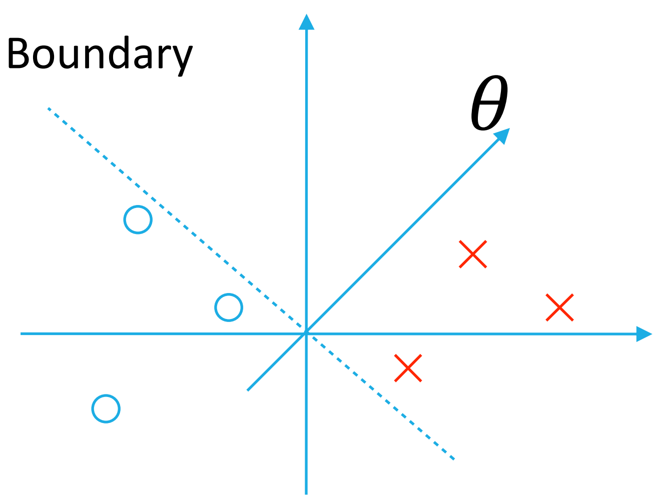
- $$\min_{\theta} \frac{1}{2}\sum_{j=1}^{n}\theta_j^2$$
  $$\text{s.t.} \quad \theta^T x^{(i)} \geq 1 \quad \text{if} \quad y^{(i)} = 1$$
  $$\theta^T x^{(i)} \leq -1 \quad \text{if} \quad y^{(i)} = 0$$
- Dot product:

$$\begin{aligned}\theta^T x^{(i)} &= p^{(i)}\|\theta\| \\ &= \theta_1 x_1^{(i)} + \theta_2 x_2^{(i)}\end{aligned}$$

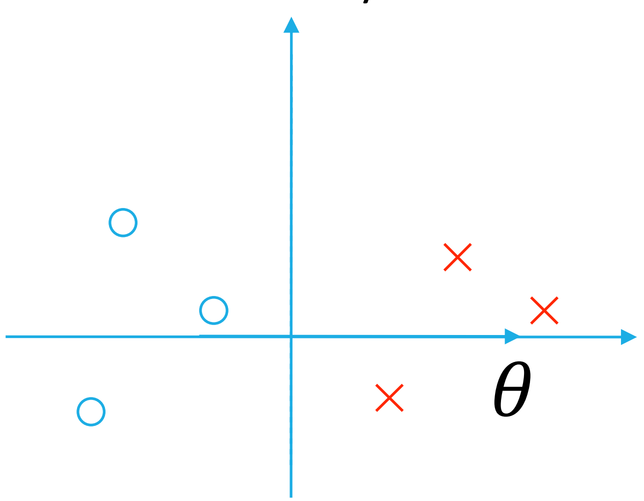
Why large margin

- $$\min_{\theta} \frac{1}{2} \sum_{j=1}^{n} \theta_j^2$$
  $$\text{s.t.} \quad p^{(i)} \|\theta\| \geq 1 \quad \text{if} \quad y^{(i)} = 1$$
  $$p^{(i)} \|\theta\| \leq -1 \quad \text{if} \quad y^{(i)} = 0$$
- Given $p^{(i)}$ to be length of projection of $x^{(i)}$ on θ
- $p^{(i)}$ is distance between sample and boundary
- Do $p^{(i)} \|\theta\| \geq 1$
  - $p^{(i)}$ is smaller, length of θ is larger
  - $p^{(i)}$ is large, length of θ can be small

Why large margin



Narrow margin



Large margin

Part 2

Feature mapping with Kernel

Learning algorithm

$$\min_{\theta} \frac{1}{2}\sum_{j=1}^{n}\theta_j^2$$

$$\text{s.t.}\quad \theta^T x^{(i)} \geq 1 \quad \text{if} \quad y^{(i)} = 1$$

$$\theta^T x^{(i)} \leq -1 \quad \text{if} \quad y^{(i)} = 0$$

Given y = -1 for negative class

$$\Rightarrow \quad \min_{\theta} \frac{1}{2}\sum_{j=1}^{n}\theta_j^2$$

$$\text{s.t.}\quad y^{(i)}(\theta^T x^{(i)}) \geq 1, i = 1, 2, ..., m$$

Learning algorithm

- Given $g_i(\theta) = -y^{(i)}(\theta^T x^{(i)}) + 1$

$$\Rightarrow \quad \min_{\theta} \frac{1}{2} \sum_{j=1}^{n} \theta_j^2$$

$$\text{s.t.} \quad g_i(\theta) = -y^{(i)}(\theta^T x^{(i)}) + 1 \leq 0$$

Switch to Lagrange problem:

$$\mathcal{L}(\theta, \alpha) = \frac{1}{2} \sum_{j=1}^{n} \theta_j^2 - \sum_{i=1}^{m} \alpha_i [y^{(i)}(\theta^T x^{(i)}) - 1]$$

Goal: $\min_{\theta,\alpha} \mathcal{L}(\theta, \alpha)$

Learning algorithm

$$\mathcal{L}(\theta, \alpha) = \frac{1}{2}\sum_{j=1}^{n}\theta_j^2 - \sum_{i=1}^{m}\alpha_i[y^{(i)}(\theta^T x^{(i)}) - 1]$$

Goal: $$\min_{\theta,\alpha} \mathcal{L}(\theta, \alpha)$$

Conditions Karush-Kuhn-Tucker (KKT)

Suppose θ^*, α^* to be solution

(1) $$\frac{\partial}{\partial\theta_j}\mathcal{L}(\theta^*, \alpha^*) = 0, j = 0, ..., n$$

(2) $$\alpha_i^* g_i(\theta^*) = 0, i = 1, ..., m$$

(3) $$g_i(\theta^*) \leq 0, i = 1, ..., m$$

(4) $$\alpha^* \geq 0, i = 1, ..., m$$

Learning algorithm

$$\mathcal{L}(\theta, \alpha) = \frac{1}{2}\sum_{j=1}^{n}\theta_j^2 - \sum_{i=1}^{m}\alpha_i[y^{(i)}(\theta^T x^{(i)}) - 1]$$

Goal: $$\min_{\theta,\alpha} \mathcal{L}(\theta, \alpha)$$

Solution for theta:

(1) $$\nabla_\theta \mathcal{L}(\theta, \alpha) = \theta - \sum_{i=1}^{m}\alpha_i y^{(i)} x^{(i)} = 0$$

➡ $$\theta = \sum_{i=1}^{m}\alpha_i y^{(i)} x^{(i)}$$

Support vector

Conditions Karush-Kuhn-Tucker (KKT)

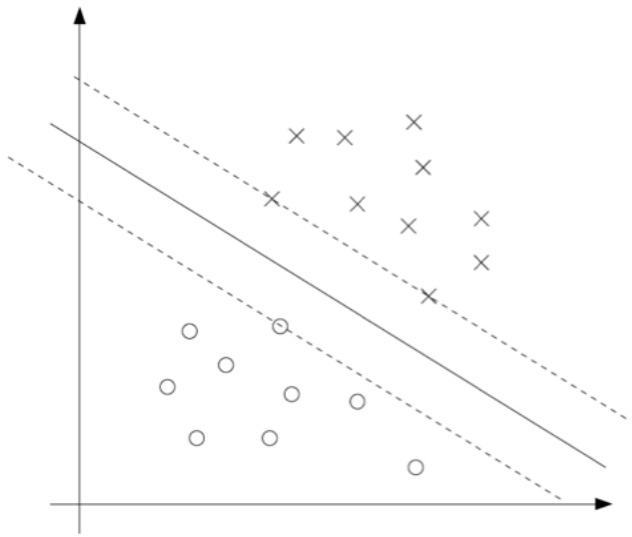
Suppose θ^*, α^* to be solution

(1) $\frac{\partial}{\partial \theta_j} \mathcal{L}(\theta^*, \alpha^*) = 0, j = 0, ..., n$

(2) $\alpha_i^* g_i(\theta^*) = 0, i = 1, ..., m$

(3) $g_i(\theta^*) \leq 0, i = 1, ..., m$

(4) $\alpha_i^* \geq 0, i = 1, ..., m$



From (2), (3) and (4):

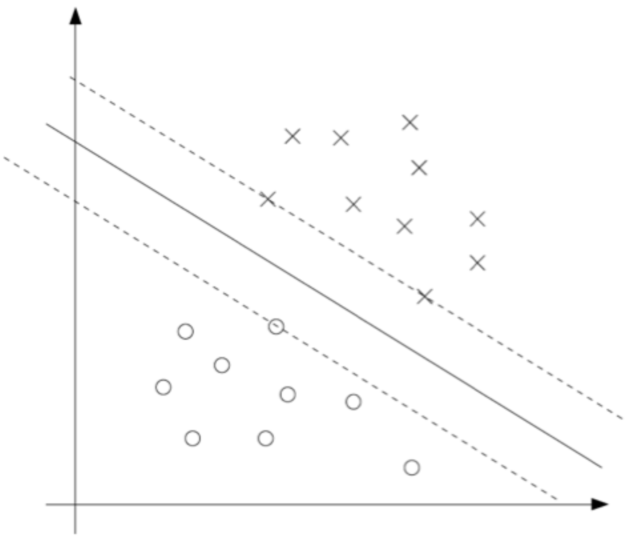
$$\text{if} \quad g_i(\theta^*) < 0, \text{then} \quad \alpha_i^* = 0, i = 1, ..., m$$

Samples outside margins: $\alpha_i^* = 0$

Support vector

Samples outside margins: $\alpha_i^* = 0$

$$\theta = \sum_{i=1}^{m} \alpha_i y^{(i)} x^{(i)}$$



Predict:

$$\theta^T x = (\sum_{i=1}^{m} \alpha_i y^{(i)} x^{(i)})^T x$$

$$= \sum_{i=1}^{m} \alpha_i y^{(i)} \langle x^{(i)}, x \rangle$$

Only calculated with data points on margins ➡ support vector

Feature mapping

- Given a function

$$\phi(x) = [x \quad x^2 \quad x^3]^T$$

Predict:

$$\theta^T x = (\sum_{i=1}^{m} \alpha_i y^{(i)} x^{(i)})^T x$$
$$= \sum_{i=1}^{m} \alpha_i y^{(i)} \langle x^{(i)}, x \rangle$$

Replace $\langle x^{(i)}, x \rangle$ by $\langle \phi(x^{(i)}), \phi(x) \rangle$

Kernel

- Given

$$K(x^{(i)}, x) = \langle \phi(x^{(i)}), \phi(x) \rangle$$

Predict:

$$\theta^T x = (\sum_{i=1}^{m} \alpha_i y^{(i)} x^{(i)})^T x$$

$$= \sum_{i=1}^{m} \alpha_i y^{(i)} \langle x^{(i)}, x \rangle$$

Replace $\langle x^{(i)}, x \rangle$ by $K(x^{(i)}, x)$

Kernel

- Example: $K(x,z) = (x^T z)^2$ O(n)

Rewrite $$K(x,z) = \left(\sum_{i=1}^{n} x_i z_i\right)\left(\sum_{j=1}^{n} x_j z_j\right)$$
$$= \sum_{i=1}^{n}\sum_{j=1}^{n} x_i x_j z_i z_j$$
$$= \phi(x)^T \phi(z)$$ O(n^2)

Với $\phi(x) = [x_1x_1 \; x_1x_2 \; x_1x_3 \; x_2x_1 \; x_2x_2 \; x_2x_3 \; x_3x_1 \; x_3x_2 \; x_3x_3]^T$

Calculation with kernel is faster than that with feature mapping

Kernel

- Gaussian kernel

$$K(x,z) = \exp\left(-\frac{\|x-z\|^2}{2\sigma^2}\right)$$