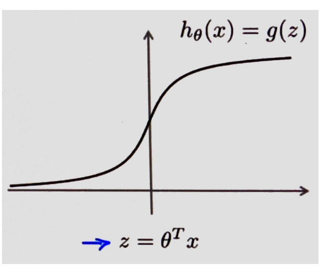
Support Vector Machines

2019年8月21日 星期三 下午4:48

- Optimization objective
  - Alternative view of logistic regression



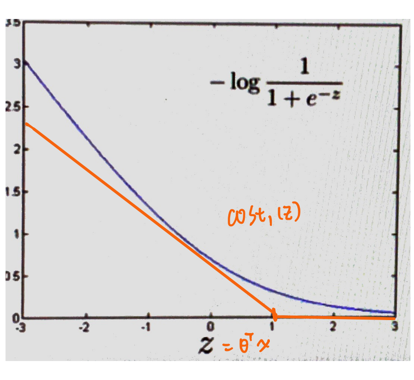
$h_\theta(x) = \frac{1}{1+e^{-\theta^T x}}$

if $y=1$, we want $h_\theta(x) \approx 1$, $\theta^T x >> 0$

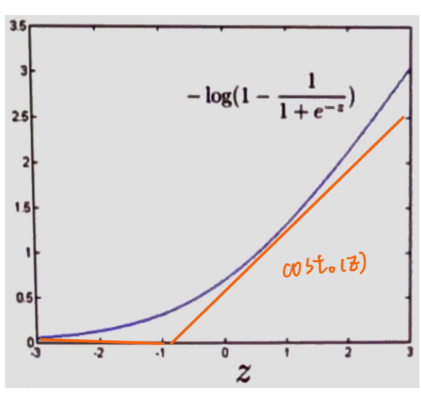
$y=0$, $h_\theta(x) \approx 0$, $\theta^T x << 0$

Cost of example: $-(y \log(h_\theta(x)) + (1-y)\log(1-h_\theta(x)))$

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



if $y=1$ (want $\theta^T x >> 0$)



if $y=0$ (want $\theta^T x << 0$)

- Support vector machine

Logistic regression (classification):

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

($-\log h_\theta(x^{(i)})$ → $cost_1(\theta^T x^{(i)})$; $(-\log(1-h_\theta(x))$ → $cost_0(\theta^T x^{(i)})$)

Support vector machine:

$\min_\theta \frac{1}{m} C \sum_{i=1}^{m} y^{(i)} cost_1(\theta^T x^{(i)}) + (1-y^{(i)}) cost_0(\theta^T x^{(i)}) + \frac{1 \lambda}{2 m}\sum_{i=0}^{n} \theta_j^2$

$A + \lambda B$ 當λ固定

$CA + B$

$\to \frac{1}{\lambda}$

SVM hypothesis

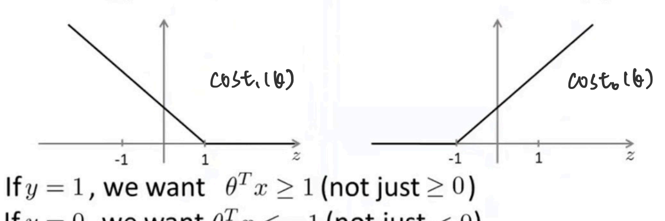
$$\min_\theta C \sum_{i=1}^{m} \left[y^{(i)} cost_1(\theta^T x^{(i)}) + (1-y^{(i)}) cost_0(\theta^T x^{(i)}) \right] + \frac{1}{2}\sum_{i=1}^{n} \theta_j^2$$

Hypothesis:

$h_\theta(x) \begin{cases} 1 & \text{if } \theta^T x \geq 0 \\ 0 & \text{otherwise} \end{cases}$

- Large Margin Intuition
  - Support Vector Machine

$$\min_\theta C \sum_{i=1}^{m} \left[y^{(i)} cost_1(\theta^T x^{(i)}) + (1-y^{(i)}) cost_0(\theta^T x^{(i)}) \right] + \frac{1}{2}\sum_{i=1}^{n} \theta_j^2$$



If $y=1$, we want $\theta^T x \geq 1$ (not just ≥ 0)

If $y=0$, we want $\theta^T x \leq -1$ (not just < 0)

- SVM Decision Boundary

if $C = 10^6$ (非常大)

$$\min_\theta C \sum_{i=1}^{m} \left[y^{(i)} cost_1(\theta^T x^{(i)}) + (1-y^{(i)}) cost_0(\theta^T x^{(i)}) \right] + \frac{1}{2}\sum_{i=1}^{n} \theta_j^2$$

(bracketed term $= 0$)

$\min C \times 0 + \frac{1}{2}\sum_{i=1}^{n} \theta_j^2$

$\theta^T x^{(i)} \geq 1$ if $y^{(i)} = 1$

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

SVM decision boundary

large margin classifier

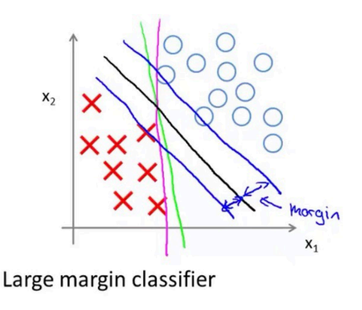
$\min \frac{1}{2}\sum_{i=1}^{n} \theta_j^2 = \|\theta\|^2$

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

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

$(\sqrt{\theta_1^2 + \theta_2^2})^2 = \|\theta\|^2$

- SVM Decision Boundary: Linearly separable case

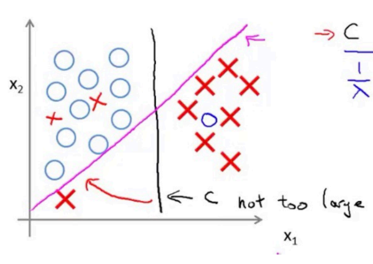


Large margin classifier

黑線與樣本之間擁有最大的最短距離

margin

- Large margin classifier in presence of outliers

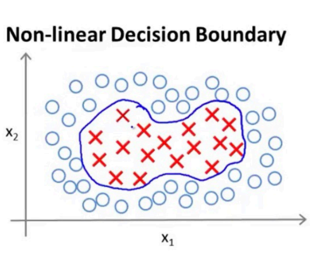


→ C very large

$\frac{1}{\lambda}$

C太大若有一個異常樣本即導致從黑線變為粉線

- Kernels I
  - **Non-linear Decision Boundary**



Predict $y = 1$ if

$\theta_0 + \theta_1 x_1 + \theta_2 x_2 + \theta_3 x_1 x_2 + \theta_4 x_1^2 + \theta_5 x_2^2 + \cdots \geq 0$

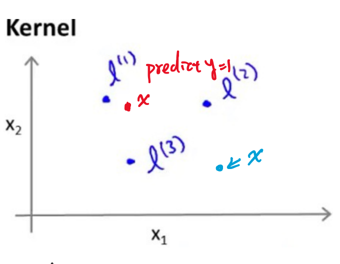
$h_\theta(x) = \begin{cases} 1 & \text{if } \theta_0 + \theta_1 x_1 + \cdots \geq 0 \\ 0 & \text{otherwise} \end{cases}$

Kernel trick在ML的角色就是希望當不同類別的資料在原始空間中無法被線性分類器區隔開時，經由非線性投影後的資料能在更高維度的空間中區隔開.

$\theta_0 + \theta_1 f_1 + \theta_2 f_2 + \theta_3 f_3 + \cdots$

$f_1 = x_1$, $f_2 = x_2$, $f_3 = x_1 x_2$, $f_4 = x_1^2$, $f_5 = x_2^2$, ...

  - **Kernel**



Given x, compute new feature depending on proximity to landmarks $l^{(1)}, l^{(2)}, l^{(3)}$

Given x: $f_1 = \text{similarity}(x, l^{(1)}) = \exp(-\frac{\|x - l^{(1)}\|^2}{2\sigma^2})$

$f_2 = \text{similarity}(x, l^{(2)}) = \exp(-\frac{\|x - l^{(2)}\|^2}{2\sigma^2})$

kernal (Gaussica kernal)

if $x \approx l^{(1)}$:

$f_1 \approx \exp(-\frac{0}{2\sigma^2}) \approx 1$

if x is far from $l^{(1)}$:

$f_1 \approx \exp(-\frac{\infty}{2\sigma^2}) \approx 0$ ← $(\text{largenumber})^2$

$l^{(1)} \to f_1$, $l^{(2)} \to f_2$ ...

Predict "1" when

$\theta_0 + \theta_1 f_1 + \theta_2 f_2 + \theta_3 f_3 > 0$

$\theta_0 = -0.5$, $\theta_1 = 1$, $\theta_2 = 1$, $\theta_3 = 0$

$f_1 \approx 1$, $f_2 \approx 0$, $f_3 \approx 0$

$-0.5 + 1 = 0.5 \geq 0$

$f_1, f_2, f_3 \approx 0$

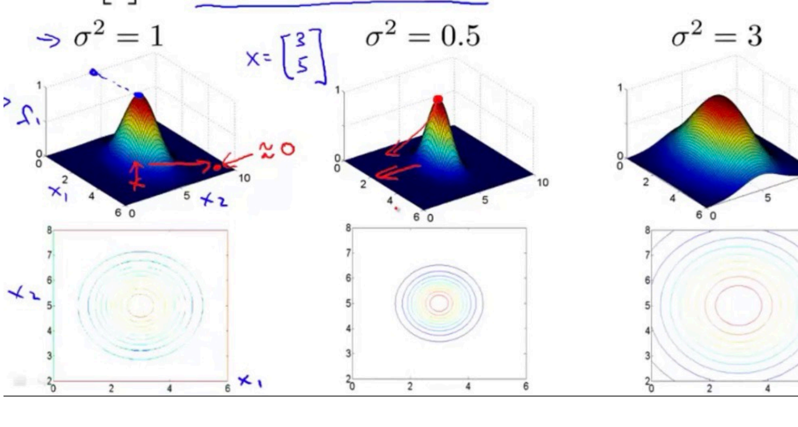
$\theta_0 + \cdots = -0.5 \leq 0$

  - σ越大收斂越慢

Example:

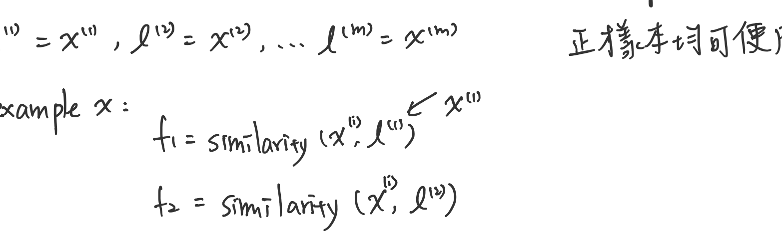
$l^{(1)} = \begin{bmatrix} 3 \\ 5 \end{bmatrix}$, $f_1 = \exp\left(-\frac{\|x - l^{(1)}\|^2}{2\sigma^2}\right)$

→ $\sigma^2 = 1$ $x = \begin{bmatrix} 3 \\ 5 \end{bmatrix}$ $\sigma^2 = 0.5$ $\sigma^2 = 3$



- Kernel II
  - Choosing the landmarks

Where to get $l^{(1)}, l^{(2)}, l^{(3)}, \ldots$?



$l^{(1)}$
$l^{(2)}$
⋮
$l^{(m)}$

  - SVM with Kernels

Given $(x^{(1)}, y^{(1)}), (x^{(2)}, y^{(2)}), \ldots (x^{(m)}, y^{(m)})$ ← training set, cross validation set, test set

choose $l^{(1)} = x^{(1)}$, $l^{(2)} = x^{(2)}$, ... $l^{(m)} = x^{(m)}$ 正樣本時可使用

Given example x:

$f_1 = \text{similarity}(x^{(i)}, l^{(1)})$ ← $x^{(i)}$

$f_2 = \text{similarity}(x^{(i)}, l^{(2)})$

...

For training example $(x^{(i)}, y^{(i)})$

$x^{(i)} \Rightarrow f_1^{(i)} = \text{sim}(x^{(i)}, l^{(1)})$

$f_2^{(i)} = \text{sim}(x^{(i)}, l^{(2)})$

⋮ ← $f_i^{(i)} = \text{sim}(x^{(i)}, l^{(i)}) = \exp(-\frac{0}{2\sigma^2}) = 1$ ← $x^{(i)}$

$f_m^{(i)} = \text{sim}(x^{(i)}, l^{(m)})$

$x^{(i)} \in R^{n+1}$ (or R^n)

$f^{(i)} = \begin{bmatrix} f_0^{(i)} \\ f_1^{(i)} \\ \vdots \\ f_m^{(i)} \end{bmatrix}$ ($f_0^{(i)}$ ← 0)

Hypothesis: Given x, compute features $f \in R^{m+1}$ $\theta \in R^{n+1}$

Predict "$y=1$" if $\theta^T f \geq 0$ ← $\theta_0 f_0 + \theta_1 f_1 + \ldots \theta_m f_m$

Training: $\min_\theta C \sum_{i=1}^{m} y^{(i)} cost_1(\theta^T f^{(i)}) + (1-y^{(i)}) cost_0(\theta^T f^{(i)}) + \frac{1}{2}\sum_{j=1}^{n=m} \theta_j^2 = \theta^T\theta$ (ignor θ_0)

$= \theta^T M \theta$ 優化計算效率

  - SVM parameters

$C = \frac{1}{\lambda}$ Large C: Lower bias, high variance (small λ)

Small C: Higher bias, low variance (large λ)

σ^2 Large σ^2: Features f_i very more smoothly

Higher bias, lower variance

$\exp(-\frac{\|x - l^{(i)}\|^2}{2\sigma^2})$

Small σ^2: Features f_i very less smoothly

Lower bias, higher variance

- Using an SVM
  - software package (eg. liblinear, libsvm, ...) to solve for parameters θ

Need to specify:

Choice of parameter C

Choice of kernel (similarity function):

e.g. No kernel ("linear kernel") n large, m small

Predict "$y=1$" if $\theta^T x > 0$ $\theta_0 + \theta_1 x_1 + \ldots \theta_n x_n \geq 0$

Gaussian kernel: n small, m large

$f_i = \exp(-\frac{\|x - l^{(i)}\|^2}{2\sigma^2})$, where $l^{(i)} = x^{(i)}$

Need to choose σ^2

  - Kernel (similarity) functions:

function f = kernel (x_1, x_2)

$f = \exp(-\frac{\|x_1 - x_2\|^2}{2\sigma^2})$

return

Note: Do perform feature scaling before using Gaussian kernel

  - Logistic regression V.S. SVMs

If n is large (relative to m): (n=10,000, m=10-1000)

Use logistic regression, or SVM without a kernel ("Linear kernel")

If n is small, m is intermediate: (n=1-1000, m=10-10,000)

Use SVM with Gaussian kernel

If n is small, m is large (n=1-1000, m=50,000+)

Create / add more features, then use logistic regression, or SVM without a kernel ("Linear kernel")