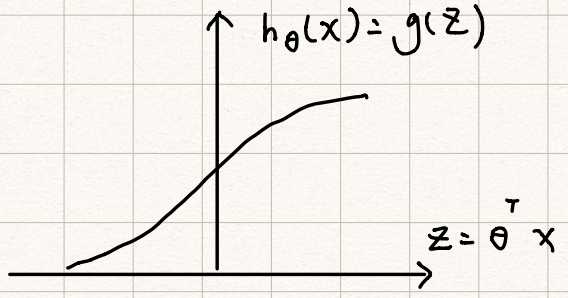
Model 表現狀況取決於：
特徵量選擇 and 選擇正則化參數

## Support Vector Machines (SVM)
支持向量機

## Optimization Object

## Alternative view of logistic regression

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



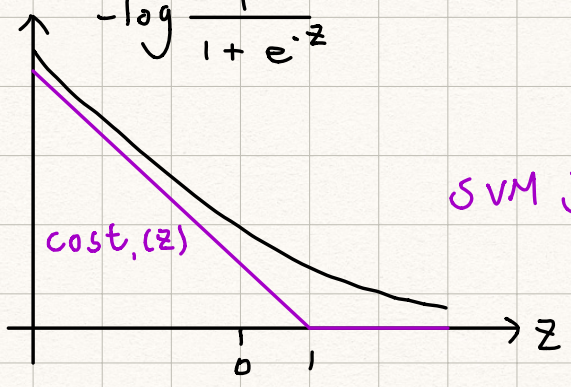

if $y=1$, we want $h_\theta(x) \approx 1$, $\theta^T x >> 0$
if $y=0$, we want $h_\theta(x) \approx 0$, $\theta^T x << 0$

## Alternative view of logistic regression

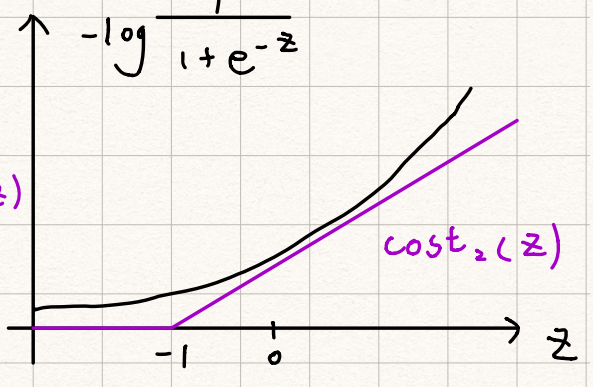
Cost of example: $-(y \log h_\theta(x) + (1-y) \log(1-h_\theta(x)))$

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

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




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




SVM $J(z)$

## Support vector machine

Logistic regression:

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

Support vector Machine:

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

if $C = \frac{1}{\lambda}$，則以上兩條算式算出來的值會相等

↓ 正則化因子

SVM Hypothesis: ← SVM在預測結果時所用的 hypothesis

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

SVM 用於擬合高項次方程式有諸多幫助

## Large Margin Intuition

## SVM Decision Boundary

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

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

↳ 如此一來，cost才會等於 0

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

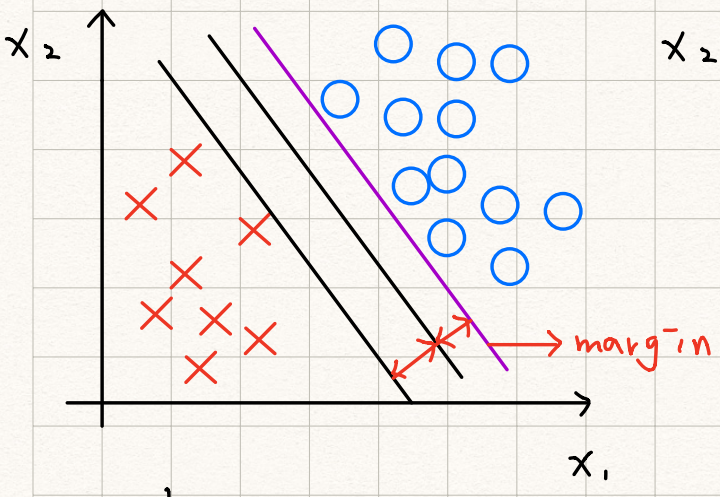
whenever $y^{(i)} = 0$: $\theta^T x^{(i)} \leq -1$

$$\min_\theta C \sum_{i=1}^{m} \underbrace{\left[ y^{(i)} \text{cost}_1(\theta^T x^{(i)}) + (1-y^{(i)}) \text{cost}_0(\theta^T x^{(i)}) \right]}_{=0} + \frac{1}{2} \sum_{j=1}^{n} \theta_j^2$$

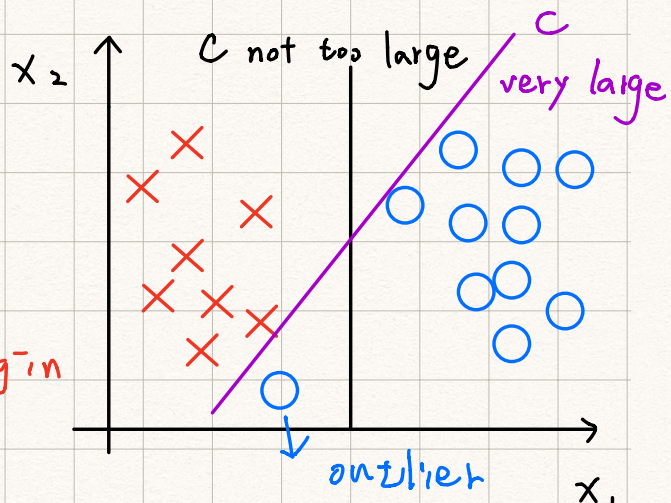
$$\min_\theta \quad C \times 0 + \frac{1}{2} \sum_{j=1}^{n} \theta_j^2$$

產生大間距分類器

## SVM Decision Boundary : Linearly separable case




Large margin classifier



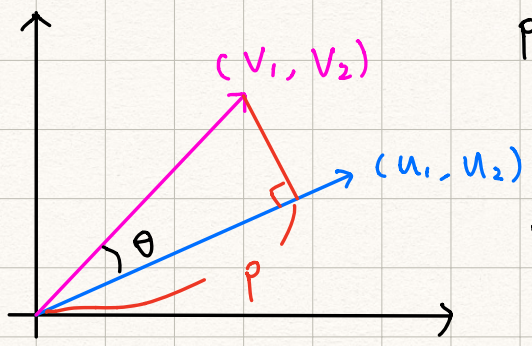

$C = \frac{1}{\lambda}$

$C \uparrow, \lambda \downarrow$, overfitting

## Mathematics Behind Large Margin Classification

本部分選學課程，利用內積理解 SVM

$$u = \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} \quad v = \begin{bmatrix} v_1 \\ v_2 \end{bmatrix} \qquad |u| = \sqrt{u_1^2 + u_2^2}$$




$p$ = length of projection of $v$ onto $u$

$$u^T v = |u|\,|v| \cos\theta = p \cdot |u| = u_1 v_1 + u_2 v_2$$

# SVM Decision Boundary

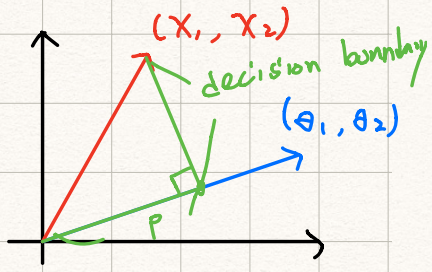
$$\min_\theta C \sum_{i=1}^{m} \left[ y^{(i)} \text{cost}_1(\theta^T x^{(i)}) + (1-y^{(i)}) \text{cost}_0(\theta^T x^{(i)}) \right]$$

視為零

$$\min_\theta \frac{1}{2} \sum_{j=1}^{n} \theta_j^2 = \frac{1}{2}(\theta_1^2 + \theta_2^2) = \frac{1}{2}\left(\sqrt{\theta_1^2 + \theta_2^2}\right)^2 = \frac{1}{2}\|\theta\|^2$$

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

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

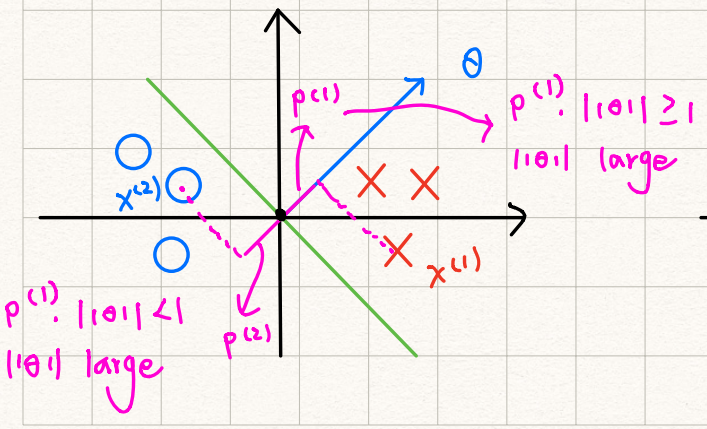


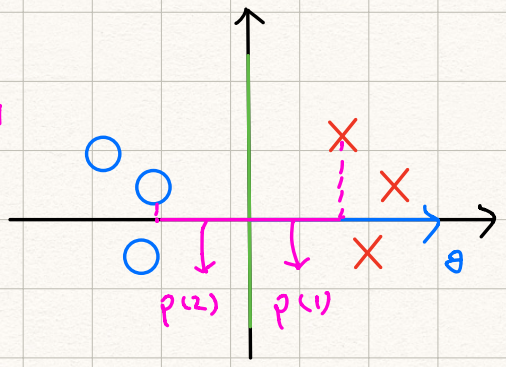

s.t. $p^{(i)} \cdot \|\theta\| \geq 1$ if $y^{(i)} = 1$

$p^{(i)} \cdot \|\theta\| \leq -1$ if $y^{(i)} = 0$

where $p^{(i)}$ is the projection of $x^{(i)}$ onto the vector $\theta$.

Simplification: $\theta_0 = 0$ → 因此 decision boundary 必通過原點



目標是
最小化 $\min_\theta \frac{1}{2}\sum_{j=1}^{n}\theta_j^2$，但由此圖分析得之，$\|\theta\|$ is large，故此條 decision boundary 非最佳化

$p^{(1)}$ 和 $p^{(2)}$ 比左圖的大得多。
因此，
$\|\theta\|$ can be smaller

**Kernel** 核密度函數

透過此函數，可以訓練出複雜的非線性 Decision Boundary

通過標記點和相似性函數定義新的特徵變量

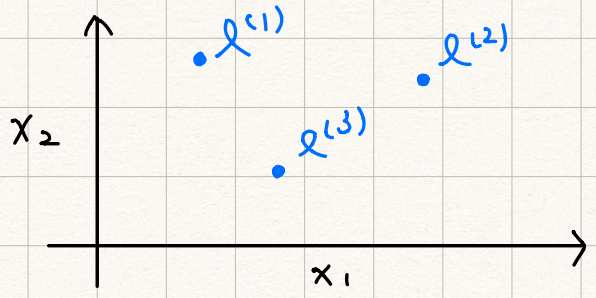
**Non-linear Decision Boundary**

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

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

Is there a different/better choice of the features $f_1, f_2, f_3, \ldots$?

## Kernel




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

Given $x$:

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

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

$$f_3 = \text{similarity}(x, l^{(3)}) = \exp(\dots)$$

kernel (Gaussian kernels) = $k(x, l^{(i)})$

## Kernels and Similarity

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

$$= \exp\left(-\frac{\sum_{j=1}^{n}(x_j - l_j^{(1)})^2}{2\sigma^2}\right)$$

$x \approx l^{(1)}$, $f_1 \approx \exp\left(-\frac{0}{2\sigma^2}\right) \approx 1$

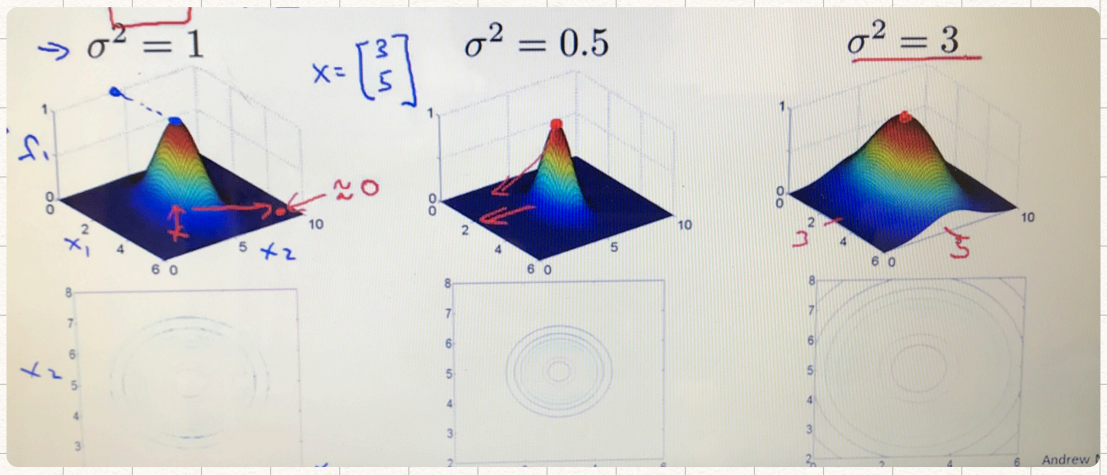
$x$ far from $l^{(1)}$, $f_1 \approx \exp\left(-\frac{\text{large num}}{2\sigma^2}\right) \approx 0$

Example

$$f_1 = \exp\left(-\frac{|x - l^{(1)}|^2}{2\sigma^2}\right)$$

$$l^{(1)} = \begin{bmatrix} 3 \\ 5 \end{bmatrix}$$

$x_1$, $x_2$, $f_1$ 的函數隨 $\sigma^2$ 而變

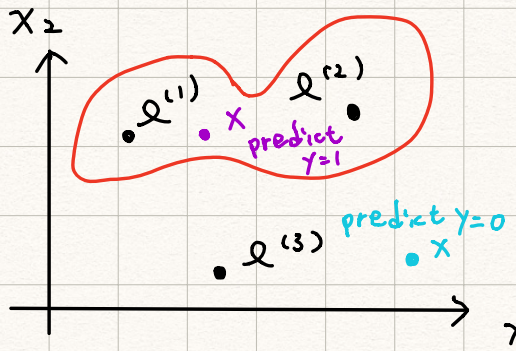



依上圖所示，$\sigma^2$ 越大，核函數坡度越緩

$\sigma^2$ 越小，核函數坡度越陡

Predict '1' when $\theta_0 + \theta_1 f_1 + \theta_2 f_2 + \theta_3 f_3 \geq 0$

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




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

$\theta_0 + \theta_1 \times 1 + \theta_2 \times 0 + \theta_3 \times 0 \geq 0$

$f_1, f_2, f_3 \approx 0$

$\rightarrow \theta_0 + 0 + 0 \approx -0.5 < 0$

## Choosing the landmarks

Where to get landmark, $l^{(1)}, l^{(2)} \dots$ ?

直接將樣本點轉為 landmark

## SVM with Kernels

Given $(x^{(1)}, y^{(1)}), (x^{(2)}, y^{(2)}), \dots (x^{(m)}, y^{(m)})$

choose $l^{(1)} = x^{(1)}, l^{(2)} = x^{(2)}, \dots l^{(m)} = x^{(m)}$

- example $x$:

  $f_1 = similarity(x, l^{(1)})$

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

  ...

  $f = \begin{pmatrix} f_0 \\ f_1 \\ \vdots \\ f_m \end{pmatrix}$ $f_0 = 1$

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

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

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

  $\vdots$

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

  $x^{(i)} = l^{(i)}$

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

  $f = \begin{pmatrix} f_0 \\ f_1 \\ \vdots \\ f_m \end{pmatrix}$ $f_0^{(i)} = 1$

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

Predict "$y=1$" if $\theta^T f \geq 0$

Training:

$$\min_\theta C \sum_{i=1}^{m} y^{(i)} \text{cost}_1(\theta^T f^{(i)}) + (1-y^{(i)}) \text{cost}_0(\theta^T f^{(i)}) + \frac{1}{2} \sum_{j=1}^{m} \theta_j^2$$

↓ 跟正則化類似，不對下標為0的數進行計算 ignore $\theta_0$

$\sum_j \theta_j^2 \neq \theta^T \theta$

$= \theta^T M \theta$

SVM是這樣實現的，多乘一個 $M$ 使SVM更有效進行運算

另，核函數(kernel)是特地為SVM設計的，用在其他算法如logistic regression會使之運算緩慢。

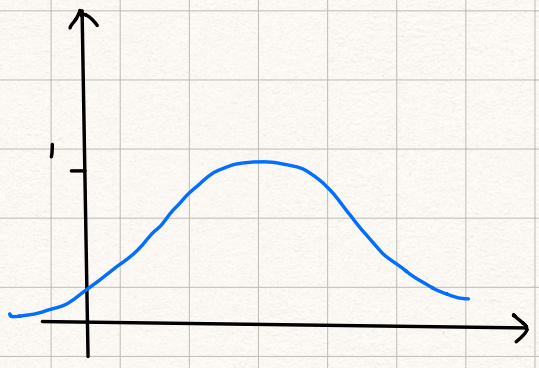
## SVM Parameters

- $C (= \frac{1}{\lambda})$. large $C$: Lower bias, high variance
  small $C$: Higher bias, low variance

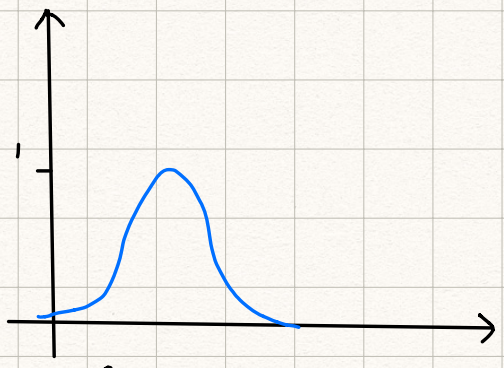
C和λ成反比

- $\sigma^2$. Large $\sigma^2$: Features $f_i$ more smoothly.
  Higher bias, lower variance

  Small $\sigma^2$: Features $f_i$ less smoothly.
  Lower bias, higher variance.




Large $\sigma^2$




Small $\sigma^2$

## Using an SVM

Use SVM software package (e.g. liblinear, libsvm, ...) to solve for parameters $\theta$.

- 不建議自己寫算法，調用現有函式庫即可

Need to specify:

Choice of parameter C

Choice of kernel (Similarity function)

↓

① No kernel ("linear kernel")

Predict "$y=1$" if $\theta^T x \geq 0$

使用時機：特徵數量多但資料量少的時候

即以線性預測之，而不造成 overfitting

② Gaussian kernel:

Need to choose $\sigma^2$.

Kernel (similarity) functions

function $f$ = kernel $(x_1, x_2)$

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

$$x \to \begin{matrix} f_1 \\ f_2 \\ \vdots \\ f_m \end{matrix}$$

return

Note: Do perform feature scaling before using the Gaussian kernel.

↓

因為各特徵的資料大小有極大的差別

## Other choices of kernel

Note: Not all similarity functions similarity$(x, \ell)$ make valid kernels.

Need to satisfy technical condition called "Mercer's Theorem" to make sure SVM Packages' optimizations run correctly, and do not diverge.

Many off-the-shelf kernel available:

- Polynomial kernel
- More esoteric: String kernel, chi-square kernel histogram intersection kernel.

## Multi-class classification

Many SVM packagages already have build-in multi-class classfication functionality.

Pick class $i$ with largest $(\theta^{(i)})^T x$

## Logistic regression vs. SVMs

n是特徵變量的大小，m是訓練集的數量

- n is large relative to m:

Use logistic regression or SVM without kernel

$n \geq m$, $n = 10000$, $m = 10 \cdots 1000$

- n is small, m is intermediate:

Use SVM with Gaussian Kernel

$n = 1\text{-}1000$, $m = 10\text{-}10000$

- if n is small, m is large:

Create / Add more features, then use logistic regression or SVM without a kernel

$n = 1\text{-}1000$, $m = 50000$

NN likely to work well for most of these settings, but slower to train.

對於SVM，是凸優化問題，不必擔心找到局部最小值，SVM之解通常為全局最小值