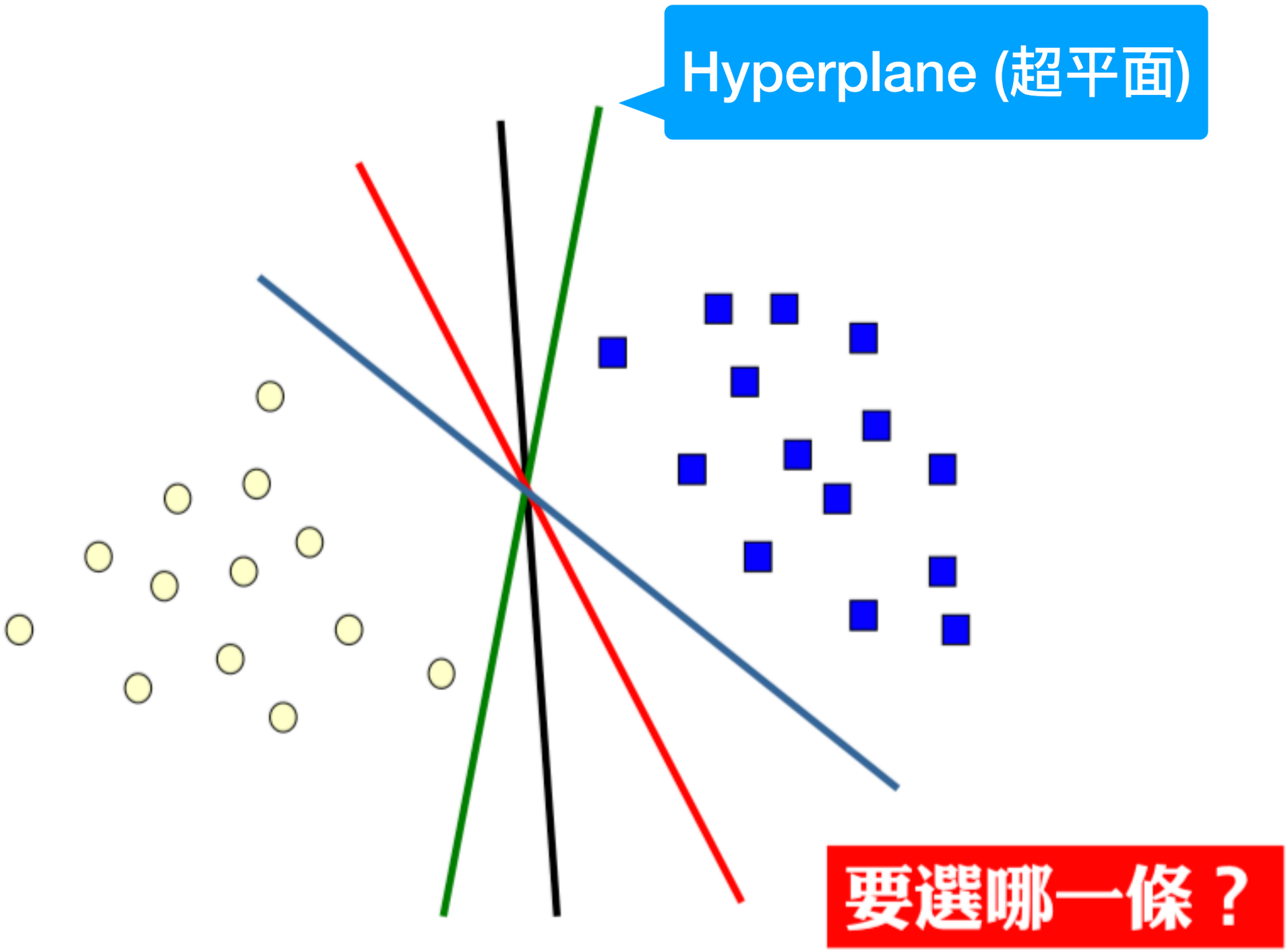
機器學習

Lecture 8 Support Vector Machines (SVM)

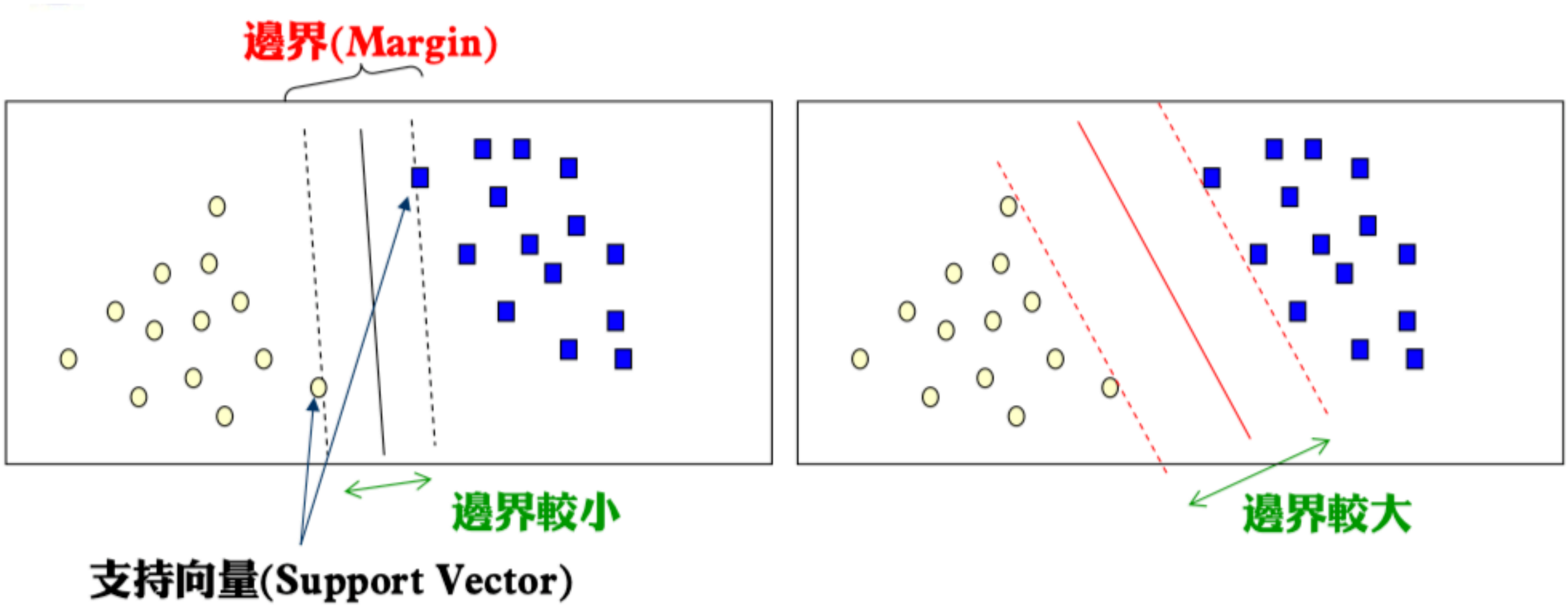
Support Vector Machines



Support Vector Machines

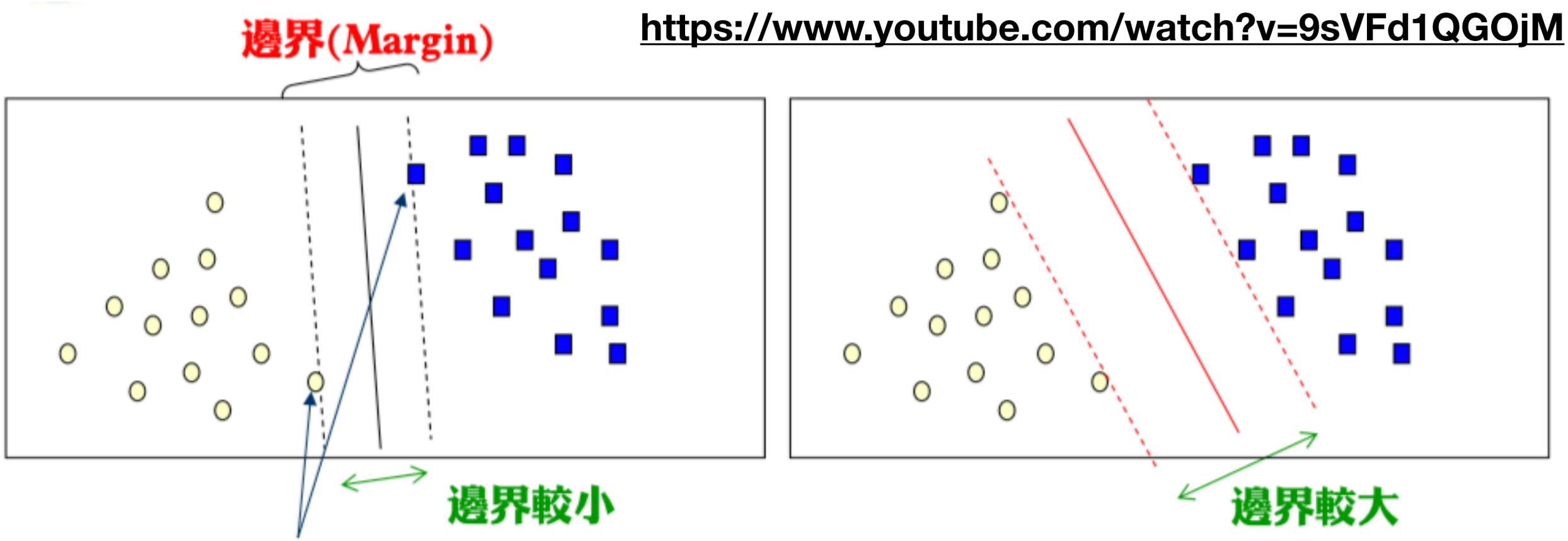
想找到最佳分界線，必須先找出距離另一組資料最近的邊緣資料點

因為最佳分界線由這些邊緣資料點所「支撐」，所以他們被稱為支持向量（Support Vectors）



Support Vector Machines

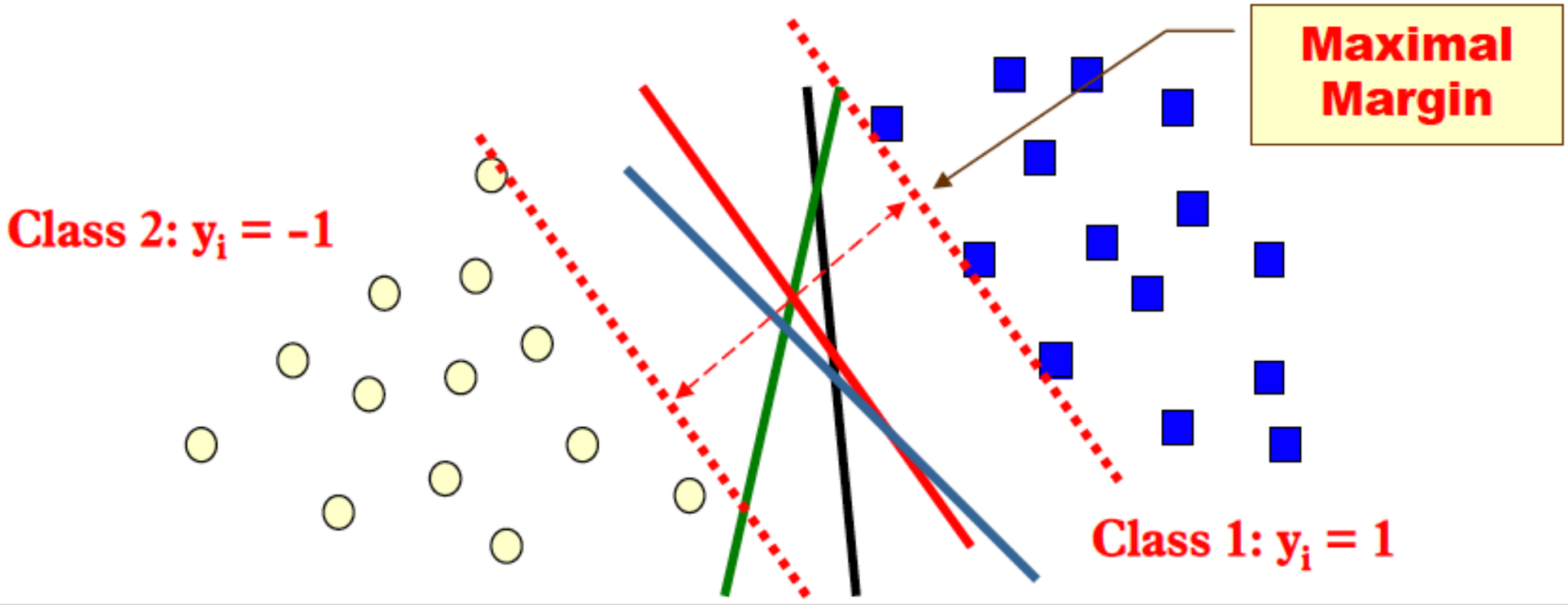
https://www.youtube.com/watch?v=9sVFd1QGOjM



- 選擇**邊界較大**者。
  - 邊界較大者，其推論錯誤率優於邊界較小者，這是因為邊界如果太小，那麼任何些微的變動都會引起顯著的影響。

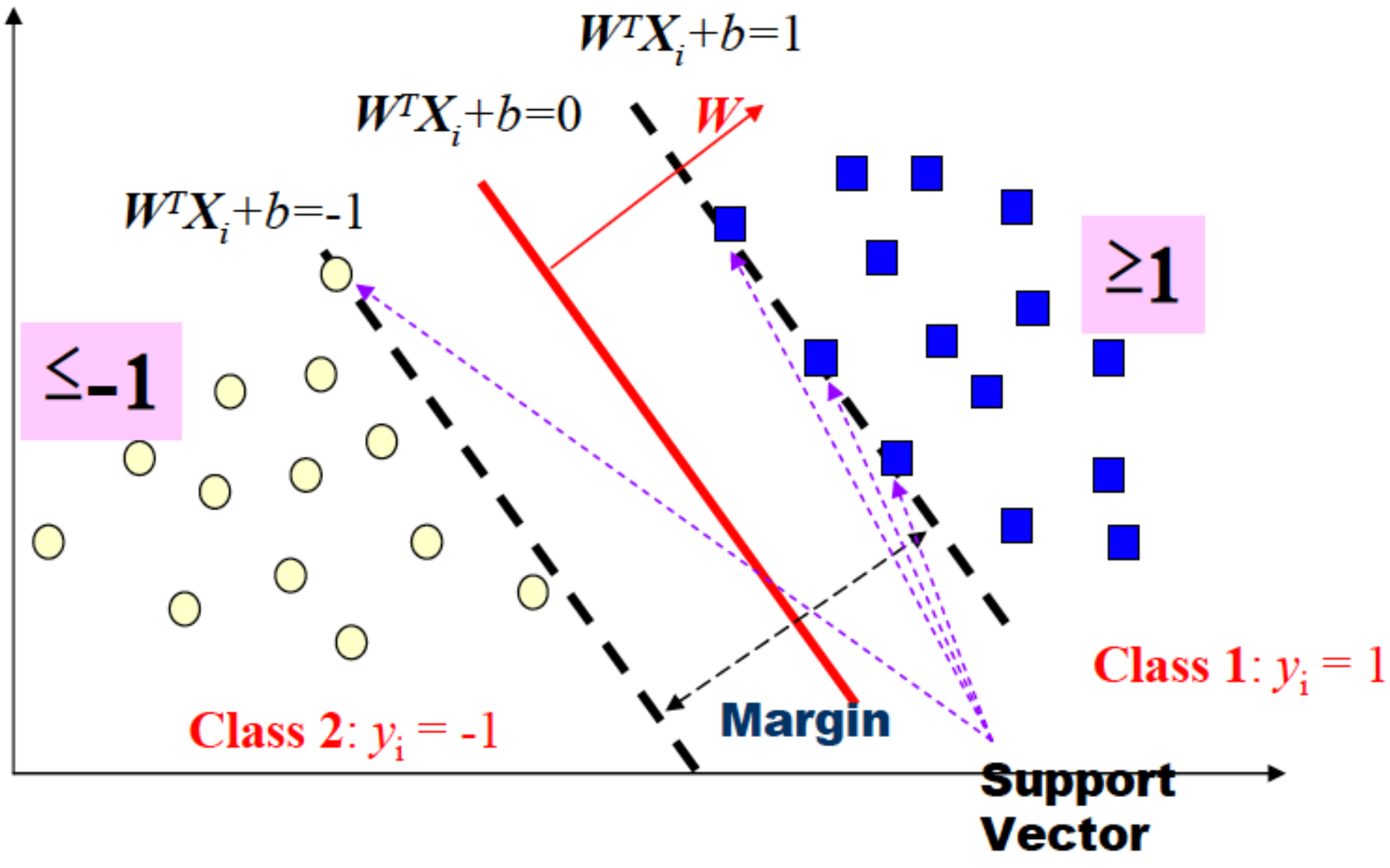
SVM - Linearly Separable

- **我們希望能夠找到一條直線，能將所有** $y_i = -1$ **及** $y_i = 1$ **的訓練資料分別落在此直線的兩側。**
  - **此直線稱為分割超平面** (Separating Hyperplane)。
  - Separating Hyperplane**往兩側延伸，可以得到兩條平行的直線邊界，稱為**Support Hyperplane (**支持超平面**)。**兩個平行的**Support Hyperplane**間之距離稱為**Margin。Margin**愈大愈好**。
  - **具最大**Margin**之分割超平面稱為**Optimal Separating Hyperplane



SVM - Linearly Separable

- **目標：Max(Margin)**
  - **由於法向量W和截距b皆會變動。因此希望找出相對應的法向量W和截距b，所構成的直線$W^TX_i+b=0$可使得Margin最大**

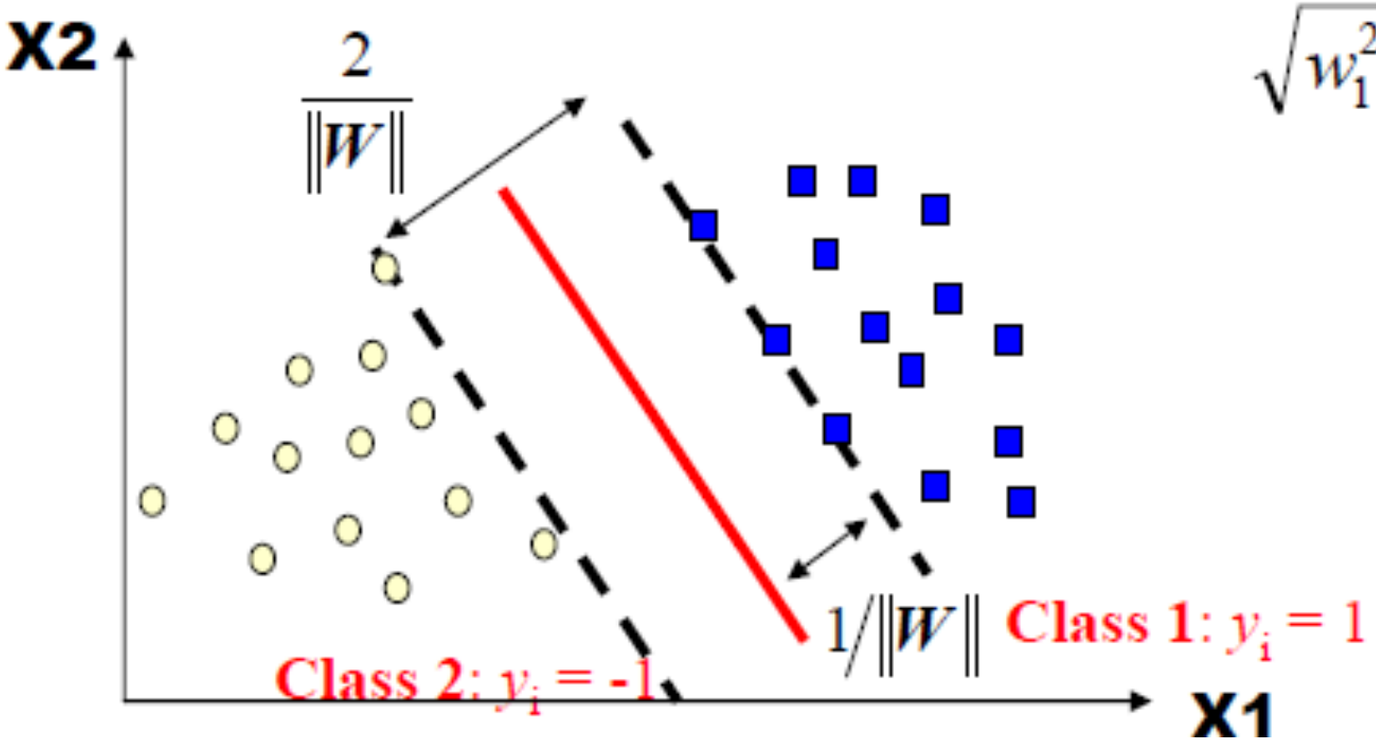


SVM - Linearly Separable

- **設計原始問題的目標函數：**
  - **找出兩個邊界之Margin最大化。即求上述兩個公式之等式的距離 (假設資料位於二維空間)：**
    - **公式1的等式：** $W^T X_i + b = 1 \Rightarrow w_1 x_1 + w_2 x_2 + (b-1) = 0$
    - **公式2的等式：** $W^T X_i + b = -1 \Rightarrow w_1 x_1 + w_2 x_2 + (b+1) = 0$
  - **上述兩平行直線公式的距離為：** $\dfrac{|(b-1)-(b+1)|}{\sqrt{w_1^2 + w_2^2}} = \dfrac{2}{\sqrt{w_1^2 + w_2^2}}$

$$= \frac{2}{\sqrt{W^T W}}$$

$$= \frac{2}{\|W\|}$$



SVM - Linearly Separable

- 我們要求上述距離$2/\|W\|$最大化，就類似於將下述公式最小化：

$$f(\boldsymbol{W}) = \frac{\|\boldsymbol{W}\|^2}{2}$$

- **所以，原始的最佳化問題模式如下所示：**

$$\text{Min.} \quad \frac{\|\boldsymbol{W}\|^2}{2}$$

$$\text{Subject to} \quad y_i\,(\boldsymbol{W}^T\boldsymbol{X}_i + b) \geq +1,\ i = 1 \ \ldots\ N$$

這就是SVM所要解決的主要問題(Primal Problem)。

可用 Karush-Kuhn-Tucker (KKT) Conditions Method
求出最好的 W* 和 b*

logistic regression v.s. linear SVM

- 實務上，linear logistic regression 和 linear SVM 常會產生相似的結果

Logistic Regression

- 觀察 $\hat{y} = P(y = 1 \mid x)$
  - 若 y = 1: 希望 $P(y = 1 \mid x)$ 越大越好
  - 若 y = 0: 希望 $P(y = 0 \mid x)$ 越大越好

x	y	機率
(80, 150)	0	希望 $P(y = 0 \mid x)$ 越大越好
(35, 130)	0	希望 $P(y = 0 \mid x)$ 越大越好
(160, 20)	1	希望 $P(y = 1 \mid x)$ 越大越好
(125, 30)	1	希望 $P(y = 1 \mid x)$ 越大越好

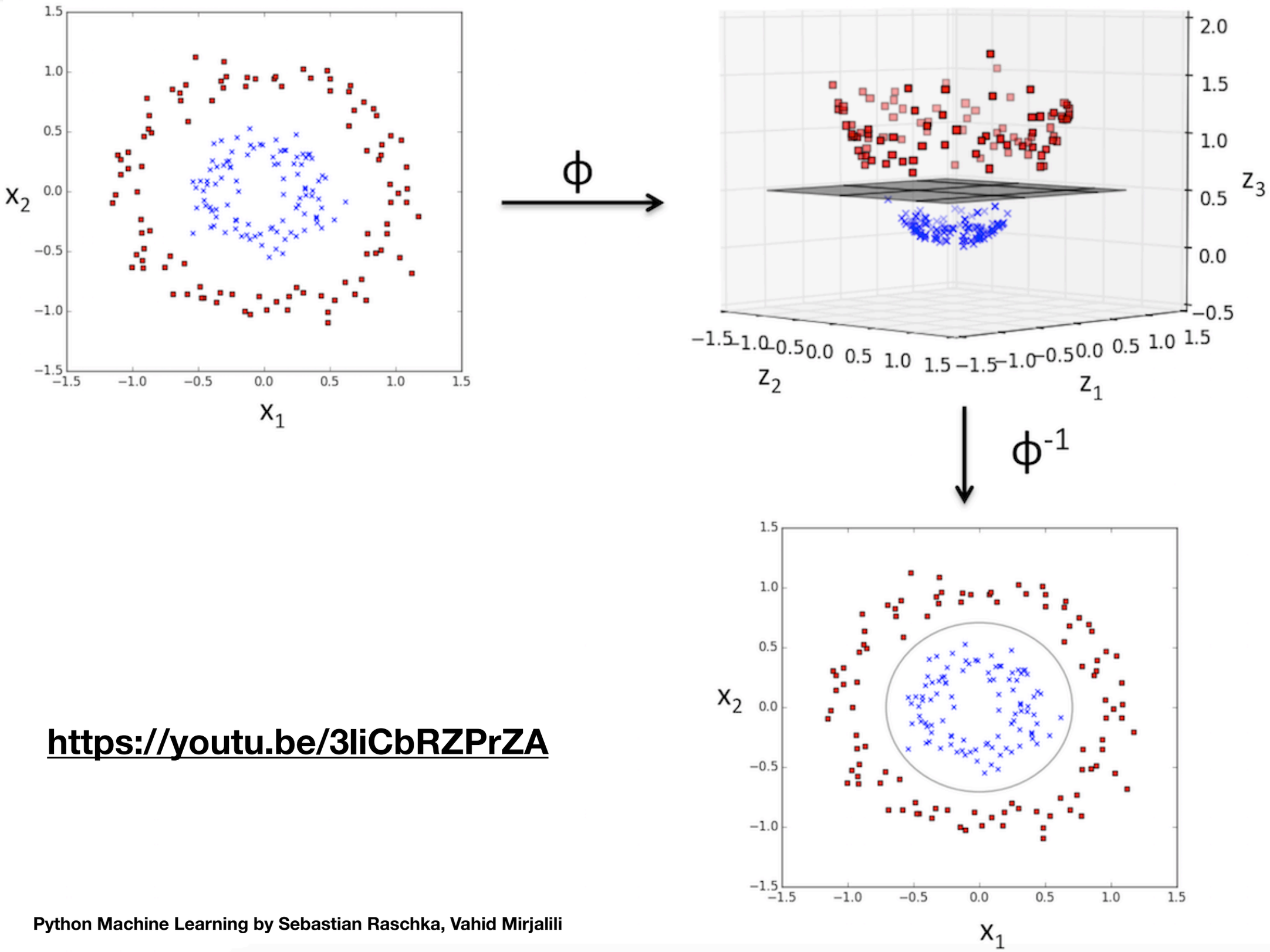
Goal: Maximize 每個點都有考慮進去—>較易受離群值影響

$$P(y^{(1)} = 0 \mid x^{(1)})P(y^{(2)} = 0 \mid x^{(2)})P(y^{(3)} = 1 \mid x^{(3)})P(y^{(4)} = 1 \mid x^{(4)})$$

logistic regression v.s. linear SVM

- 實務上，linear logistic regression 和 linear SVM 常會產生相似的結果
- 但因 logistic regression 試圖最大化訓練數據集的條件概似(conditional likelihood)，所以較容易受離群值影響；而SVM 主要在意那些非常接近決策邊界的點(支援向量)

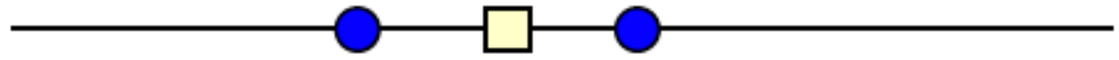
Kernel SVM: 非線性分類問題



https://youtu.be/3liCbRZPrZA

Kernel methods

- **有三個一維資料：** $X_1 = 0, X_2 = 1, X_3 = 2$**，這三個資料在一維空間** R^1**之下是線性不可分的** (Nonseparable)

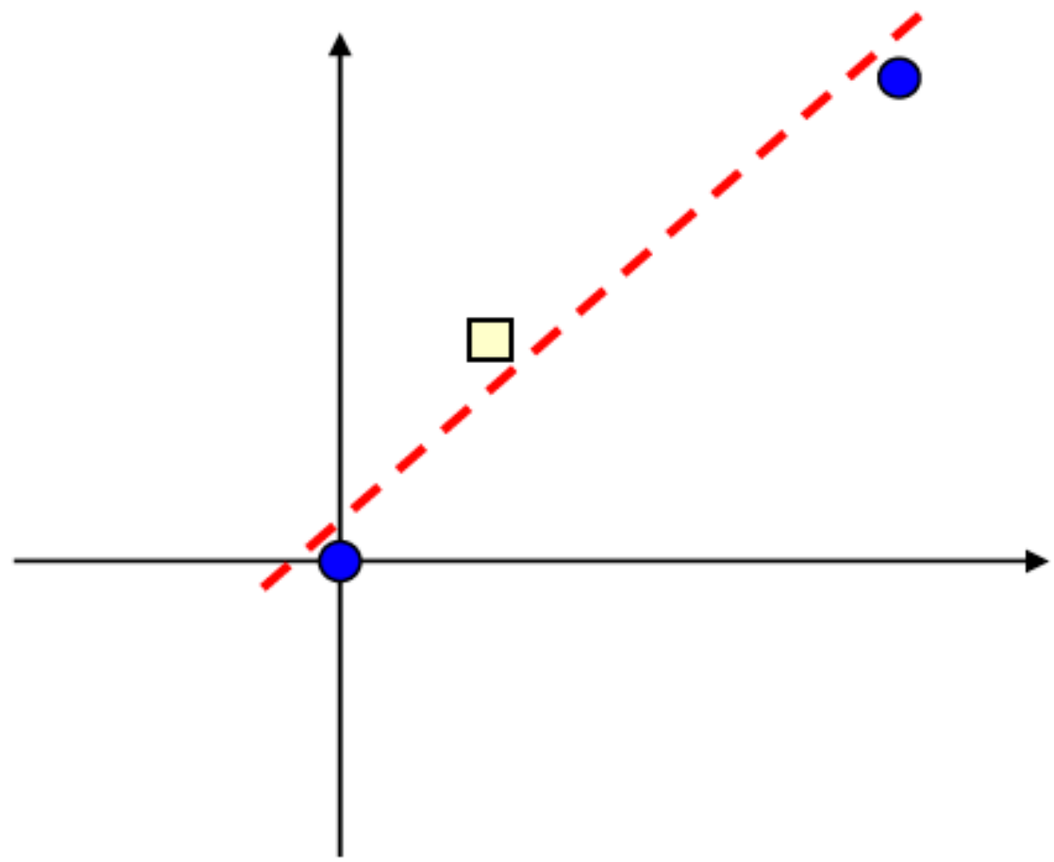


- **將這三個資料透過一個函數** $\Phi(X) = (X^2, \sqrt{2}X)$ **轉換到二維空間** R^2**中，即為線性可分割的**(Separable)

$$\Phi(X_1) = (0, 0)$$

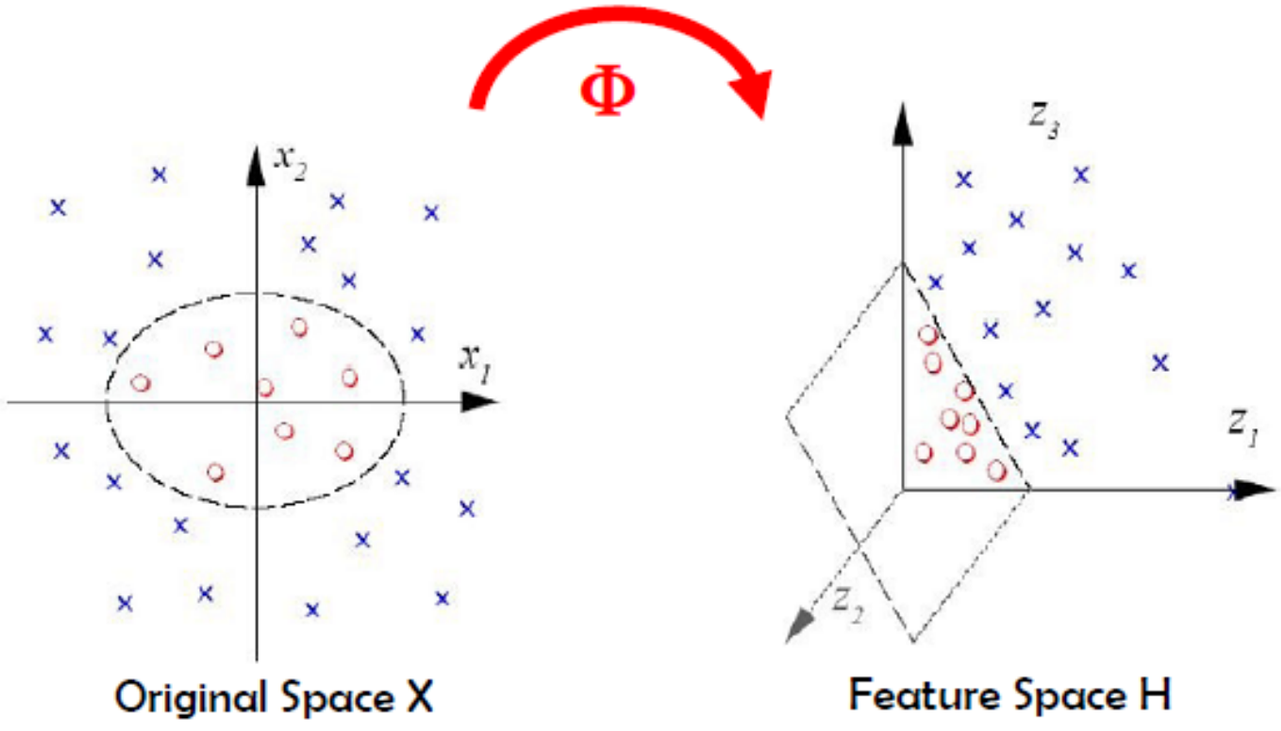
$$\Phi(X_2) = (1, \sqrt{2})$$

$$\Phi(X_3) = (4, 2\sqrt{2})$$



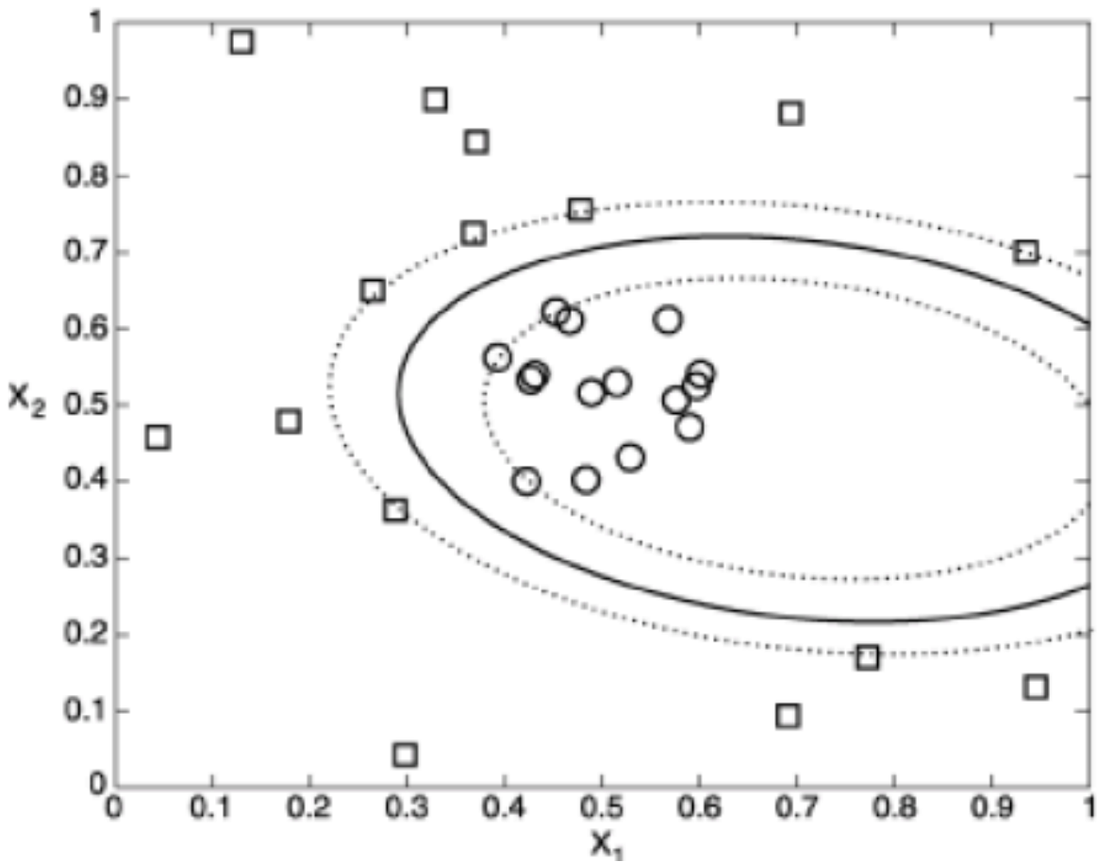
Kernel methods

- **精神：**
  - **在原始空間 X (Original Space) 不好區分的資料(無法線性區分)，可以用一個非線性的映射函數Φ，將這些資料轉換到另一個空間 H (特徵空間；Feature Space)，或許比較好區分 (可線性區分)。**
  - **特徵空間不一定是更高維的空間，但通常愈高維，資料可以分得愈開。**



Kernel Trick

- Kernel trick **是一個在特徵空間中使用原始屬性集合來進行內積計算的方法。即** $k(X_i, X_j) = <\Phi(X_i), \Phi(X_j)>$
- **可用在非線性的支援向量機的問題上：**
  - **不用知道正確的映射函數。**
  - **使用核函數計算內積，比起使用轉換後的屬性集合來得容易。**
  - **因在原始的空間中進行計算，可避免維度過高所造成的問題。**
- **使用多項式核函數的非線性決策界限**



常用的核函數

- **Linear Kernel (線性核函數)**
  - $k(X_i, X_j) = <X_i, X_j>$
- **Polynomial Kernel (多項式核函數)**
  - $k(X_i, X_j) = (<X_i, X_j>+1)^p$, $P \in Z^+$
- **Gaussian Kernel (高斯核函數)**
  - $k(X_i, X_j) = e^{[-\|X_i - X_j\|^2/(2\sigma^2)]}$

Radial Basis Function kernel (RBF)

- **上述多項式核函數中有一個參數P，而高斯核函數有一個參數σ。若使用這些函數時，參數設定值不同，則所對應到的特徵空間也會不同!!哪一個比較好，就是一個重要的研究課題。**

SVM的優缺點

- 優點
  - SVM對不同類型的數據集都有不錯的表現

- 缺點
  - 當樣本數太大時，SVM很耗費時間
  - 對資料預處理以及參數調節的要求很高
    - ★三個重要的參數
      1. 核函數
      2. 核函數的參數 （ex. RBF的gamma值）
      3. 正規化的參數 （C）