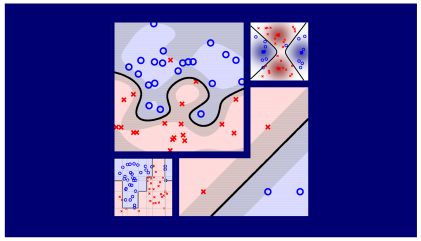
# Machine Learning Techniques
# (機器學習技法)



## Lecture 1: Linear Support Vector Machine


Hsuan-Tien Lin (林軒田)

htlin@csie.ntu.edu.tw

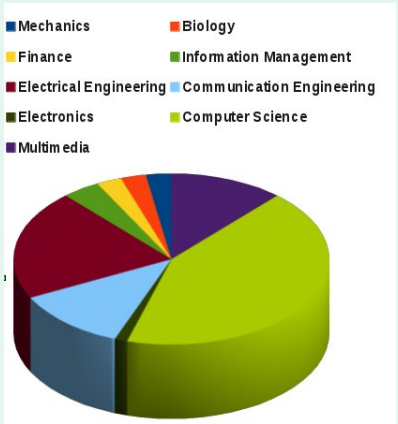
Department of Computer Science
& Information Engineering

National Taiwan University
(國立台灣大學資訊工程系)


# Course History

## NTU Version

- 15-17 weeks (2+ hours)
- highly-praised with **English and blackboard teaching**




## Coursera Version

- 8 weeks of 'foundations' (previous course) + 8 weeks of 'techniques' (**this course**)
- **Mandarin teaching** to reach more audience in need
- **slides teaching** improved with Coursera's quiz and homework mechanisms

goal: **try** making Coursera version even better than NTU version

# Course Design

## from Foundations to Techniques

- mixture of philosophical illustrations, key theory, core algorithms, usage in practice, and hopefully jokes **:-)**
- three major techniques surrounding **feature transforms**:
  - Embedding Numerous Features: how to **exploit** and **regularize** numerous features?
    —inspires **Support Vector Machine** (SVM) model
  - Combining Predictive Features: how to **construct** and **blend** predictive features?
    —inspires **Adaptive Boosting** (AdaBoost) model
  - Distilling Implicit Features: how to **identify** and **learn** implicit features?
    —inspires **Deep Learning** model

allows students to **use ML professionally**

# Fun Time

## Which of the following description of this course is true?

1. the course will be taught in Taiwanese
2. the course will tell me the techniques that create the android Lieutenant Commander Data in Star Trek
3. the course will be 16 weeks long
4. the course will focus on three major techniques

# Fun Time

## Which of the following description of this course is true?

1. the course will be taught in Taiwanese
2. the course will tell me the techniques that create the android Lieutenant Commander Data in Star Trek
3. the course will be 16 weeks long
4. the course will focus on three major techniques

## Reference Answer: 4

1. no, my Taiwanese is unfortunately not good enough for teaching (yet)
2. no, although what we teach may serve as building blocks
3. no, unless you have also joined the previous course
4. yes, **let's get started**!

# Roadmap

1. Embedding Numerous Features: Kernel Models

**Lecture 1: Linear Support Vector Machine**

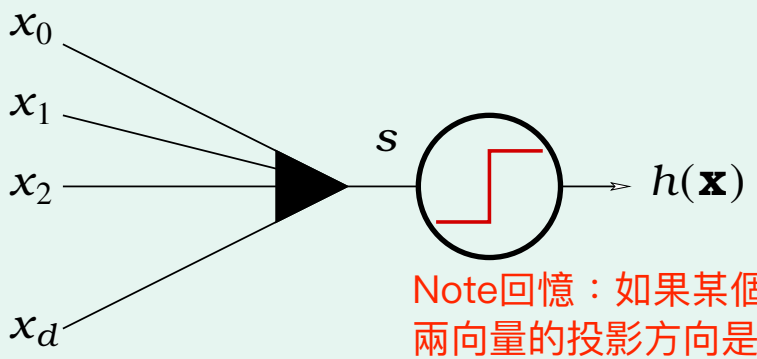
- Course Introduction
- Large-Margin Separating Hyperplane
- Standard Large-Margin Problem
- Support Vector Machine
- Reasons behind Large-Margin Hyperplane

2. Combining Predictive Features: Aggregation Models
3. Distilling Implicit Features: Extraction Models

# Linear Classification Revisited

## PLA/pocket

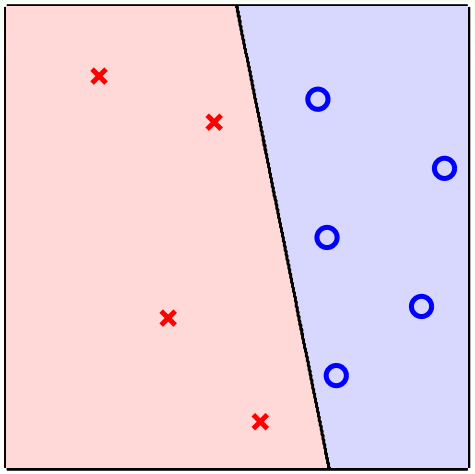
$$h(\mathbf{x}) = \text{sign}(s)$$




plausible err = 0/1

(small flipping noise)

minimize **specially**



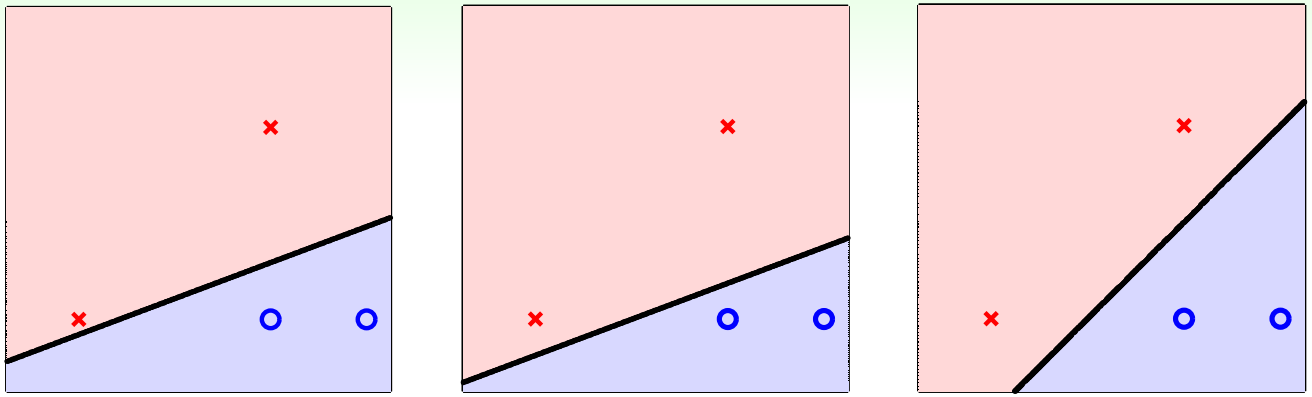
(linear separable)

Note回憶：如果某個向量與線之法向量內積為正，代表此兩向量的投影方向是同方向，也就是指向法向量的那一邊;反之內積為負代表是指反法向量的那一邊

linear (hyperplane) classifiers:

$$h(\mathbf{x}) = \text{sign}(\mathbf{w}^T\mathbf{x})$$

# Which Line Is Best?



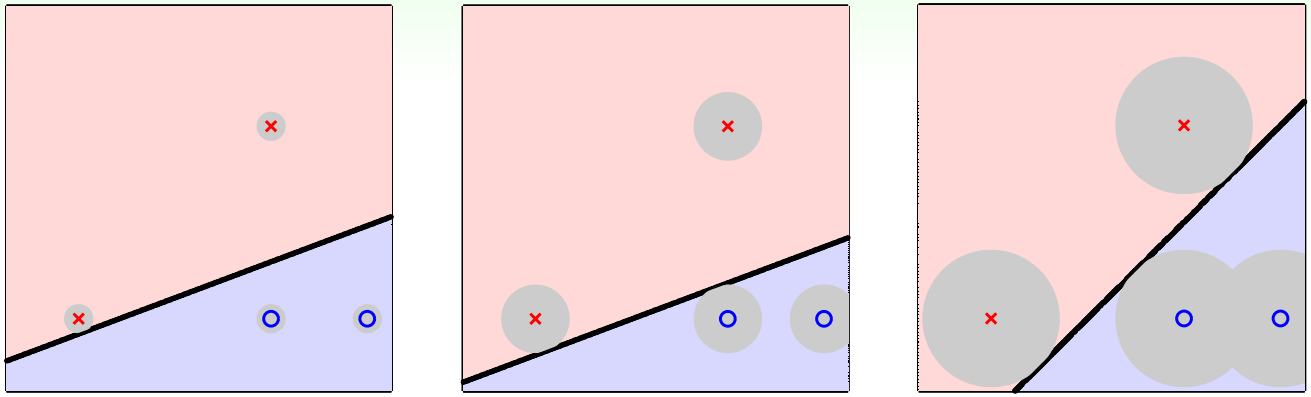
- PLA? depending on randomness
- VC bound? whichever you like!

$$E_{\text{out}}(\mathbf{w}) \leq \underbrace{E_{\text{in}}(\mathbf{w})}_{0} + \underbrace{\Omega(\mathcal{H})}_{d_{\text{VC}}=d+1}$$

You? **rightmost one, possibly :-)**

# Why Rightmost Hyperplane?

想要找一條可以長得最「胖」的線



## informal argument

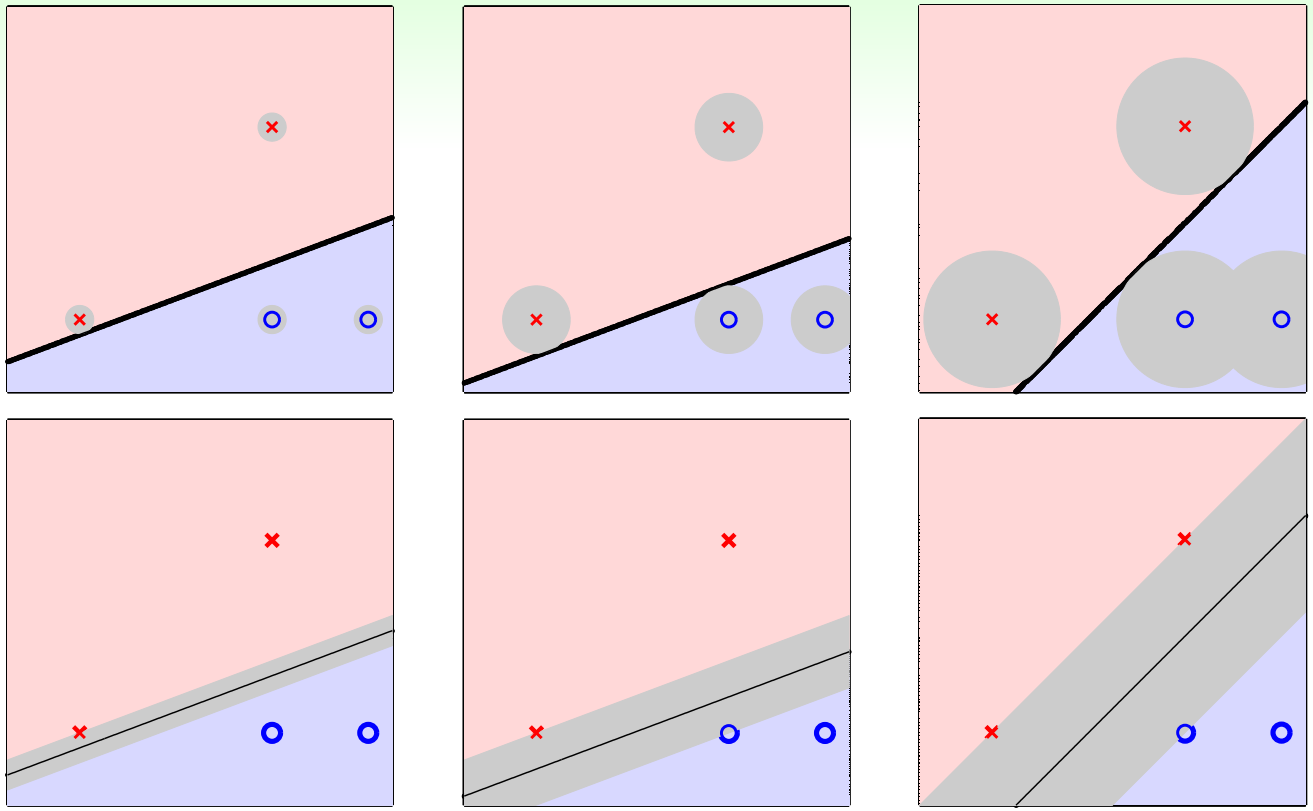
if (Gaussian-like) noise on future $\mathbf{x} \approx \mathbf{x}_n$:

| | |
|---|---|
| $\mathbf{x}_n$ further from hyperplane | distance to closest $\mathbf{x}_n$ |
| $\Longleftrightarrow$ tolerate more noise | $\Longleftrightarrow$ amount of noise tolerance |
| $\Longleftrightarrow$ more robust to overfitting | $\Longleftrightarrow$ robustness of hyperplane |

rightmost one: **more robust**

because of **larger distance to closest $\mathbf{x}_n$**

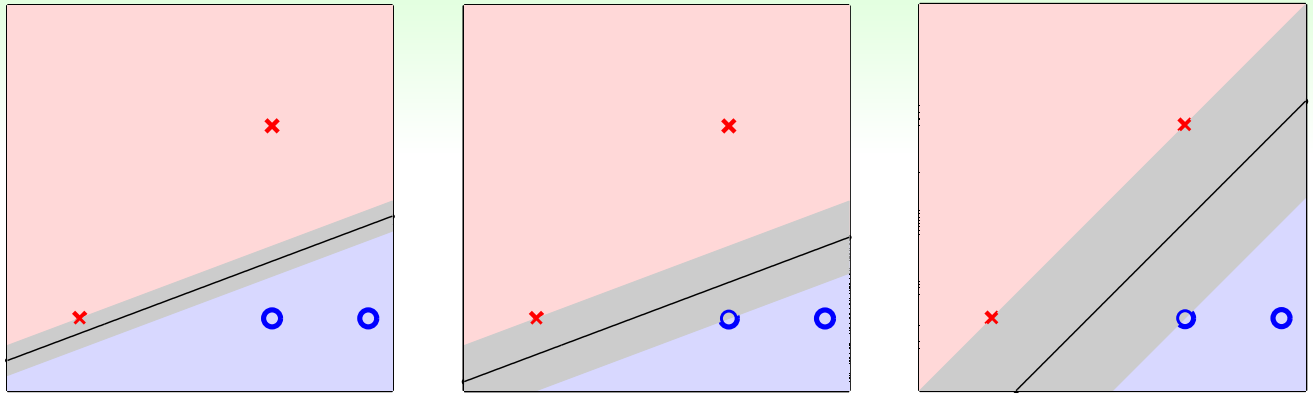
# Fat Hyperplane



- robust separating hyperplane: **fat**
  —far from both sides of examples
- robustness $\equiv$ **fatness**: distance to closest $\mathbf{x}_n$

goal: find **fattest** separating hyperplane

# Large-Margin Separating Hyperplane

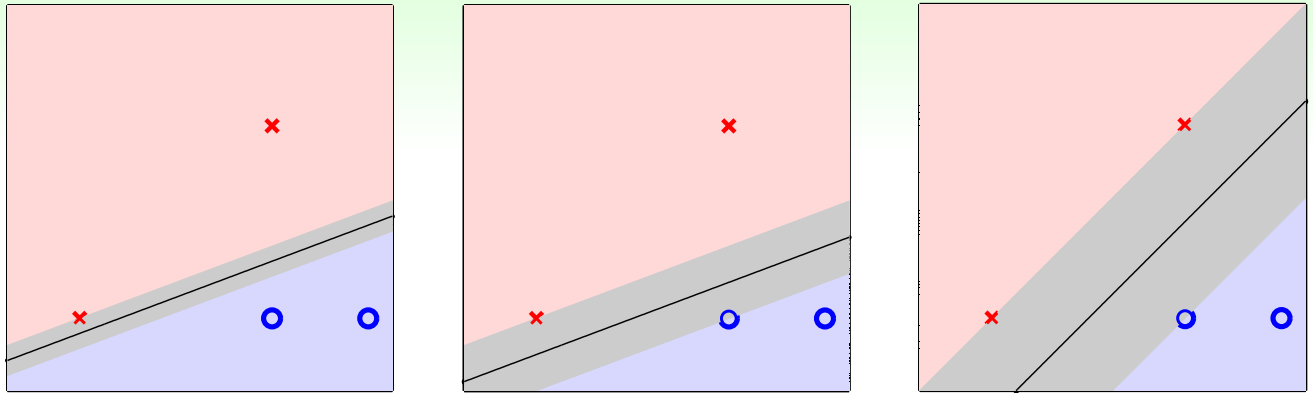


$$\begin{aligned} \max_{\mathbf{w}} \quad & \text{fatness}(\mathbf{w}) \\ \text{subject to} \quad & \mathbf{w} \text{ classifies every } (\mathbf{x}_n, y_n) \text{ correctly} \\ & \text{fatness}(\mathbf{w}) = \min_{n=1,\dots,N} \text{distance}(\mathbf{x}_n, \mathbf{w}) \end{aligned}$$

- fatness: formally called **margin**
- correctness: $y_n = \text{sign}(\mathbf{w}^T \mathbf{x}_n)$ W^T跟X_n同號就代表兩個相乘>0

goal: find **largest-margin** **separating** hyperplane

# Large-Margin Separating Hyperplane



$$\max_{\mathbf{w}} \quad \textbf{margin}(\mathbf{w})$$

$$\text{subject to} \quad \text{every } y_n \mathbf{w}^T \mathbf{x}_n > 0$$

$$\textbf{margin}(\mathbf{w}) = \min_{n=1,\dots,N} \text{distance}(\mathbf{x}_n, \mathbf{w})$$

- fatness: formally called **margin**
- correctness: $y_n = \text{sign}(\mathbf{w}^T \mathbf{x}_n)$

goal: find **largest-margin** **separating** hyperplane

# Fun Time

Consider two examples $(\mathbf{v}, +1)$ and $(-\mathbf{v}, -1)$ where $\mathbf{v} \in \mathbb{R}^2$ (without padding the $v_0 = 1$). Which of the following hyperplane is the largest-margin separating one for the two examples? You are highly encouraged to visualize by considering, for instance, $\mathbf{v} = (3, 2)$.

1. $x_1 = 0$
2. $x_2 = 0$
3. $v_1 x_1 + v_2 x_2 = 0$
4. $v_2 x_1 + v_1 x_2 = 0$

# Fun Time

Consider two examples $(\mathbf{v}, +1)$ and $(-\mathbf{v}, -1)$ where $\mathbf{v} \in \mathbb{R}^2$ (without padding the $v_0 = 1$). Which of the following hyperplane is the largest-margin separating one for the two examples? You are highly encouraged to visualize by considering, for instance, $\mathbf{v} = (3, 2)$.

1. $x_1 = 0$
2. $x_2 = 0$
3. $v_1 x_1 + v_2 x_2 = 0$
4. $v_2 x_1 + v_1 x_2 = 0$

## Reference Answer: 3

Here the largest-margin separating hyperplane (line) must be a perpendicular bisector of the line segment between $\mathbf{v}$ and $-\mathbf{v}$. Hence $\mathbf{v}$ is a normal vector of the largest-margin line. The result can be extended to the more general case of $\mathbf{v} \in \mathbb{R}^d$.

V剛好就是那條最胖的線的法向量！

# Distance to Hyperplane: Preliminary 初步想法

$$\max_{\mathbf{w}} \quad \text{margin}(\mathbf{w})$$

$$\text{subject to} \quad \text{every } y_n \mathbf{w}^T \mathbf{x}_n > 0$$

$$\text{margin}(\mathbf{w}) = \min_{n=1,\dots,N} \text{distance}(\mathbf{x}_n, \mathbf{w})$$

## 'shorten' **x** and **w**

distance needs $w_0$ and $(w_1, \dots, w_d)$ differently (to be derived)

b在這邊就是之前墊高的向量，當作截距項

$$b = w_0 \qquad \cancel{x_0 = 1}$$

$$\begin{bmatrix} | \\ \mathbf{w} \\ | \end{bmatrix} = \begin{bmatrix} w_1 \\ \vdots \\ w_d \end{bmatrix} \quad ; \quad \begin{bmatrix} | \\ \mathbf{x} \\ | \end{bmatrix} = \begin{bmatrix} x_1 \\ \vdots \\ x_d \end{bmatrix}$$

為了推導距離公式，我們這邊不把x向量墊高、改用原本拆開的方式

for this part: $h(\mathbf{x}) = \text{sign}(\mathbf{w}^T \mathbf{x} + b)$

# Distance to Hyperplane

want: distance($\mathbf{x}, b, \mathbf{w}$), with hyperplane $\mathbf{w}^T\mathbf{x}' + b = 0$

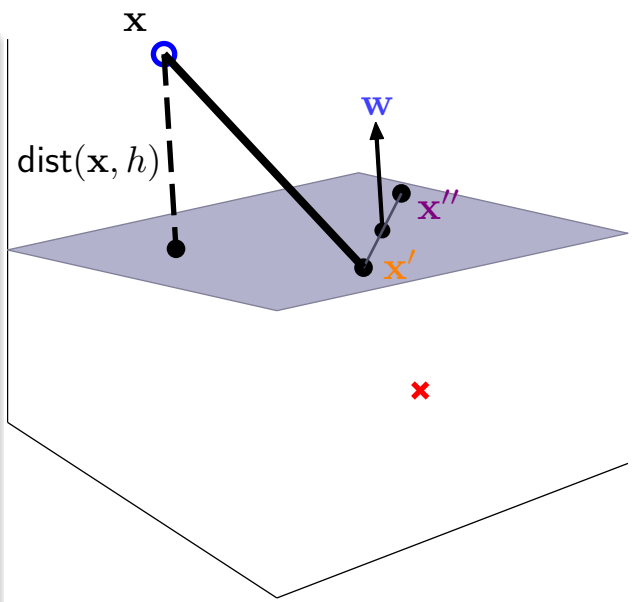
先考慮在超平面上的兩個點x', x'':

consider $\mathbf{x}'$, $\mathbf{x}''$ on hyperplane

1. $\mathbf{w}^T\mathbf{x}' = -b$, $\mathbf{w}^T\mathbf{x}'' = -b$
2. $\mathbf{w} \perp$ hyperplane: W其實就是這個超平面的法向量

$$\left( \mathbf{w}^T \quad \underbrace{(\mathbf{x}'' - \mathbf{x}')}_{\text{vector on hyperplane}} \right) = 0$$

3. distance = project $(\mathbf{x} - \mathbf{x}')$ to $\perp$ hyperplane




所以距離就可以看成是x到x'上對法向量w的投影

$$\text{distance}(\mathbf{x}, b, \mathbf{w}) = \left| \frac{\mathbf{w}^T}{\|\mathbf{w}\|}(\mathbf{x} - \mathbf{x}') \right| \overset{①}{=} \frac{1}{\|\mathbf{w}\|}|\mathbf{w}^T\mathbf{x} + b|$$

線性代數：a向量在b向量上的投影長度，就是ab內積後除以b的長度（可由內積定義推導）

# Distance to **Separating** Hyperplane

$$\text{distance}(\mathbf{x}, b, \mathbf{w}) = \frac{1}{\|\mathbf{w}\|}|\mathbf{w}^T\mathbf{x} + b|$$

- **separating** hyperplane: for every $n$

$$y_n(\mathbf{w}^T\mathbf{x}_n + b) > 0$$

- distance to **separating** hyperplane:

$$\text{distance}(\mathbf{x}_n, b, \mathbf{w}) = \frac{1}{\|\mathbf{w}\|} y_n(\mathbf{w}^T\mathbf{x}_n + b)$$

yn不是+1就是–1，所以沒差

因為完美分隔線滿足上述yn(w^Txn+b)>0，所以這裡就可以把絕對值給脫掉！

$$\begin{aligned} \max_{b,\mathbf{w}} \quad & \text{margin}(b, \mathbf{w}) \\ \text{subject to} \quad & \text{every } y_n(\mathbf{w}^T\mathbf{x}_n + b) > 0 \\ & \text{margin}(b, \mathbf{w}) = \min_{n=1,\dots,N} \frac{1}{\|\mathbf{w}\|} y_n(\mathbf{w}^T\mathbf{x}_n + b) \end{aligned}$$

再進行簡化問題~

# Margin of Special Separating Hyperplane

$$\max_{b,\mathbf{w}} \quad \text{margin}(b,\mathbf{w})$$

$$\text{subject to} \quad \text{every } y_n(\mathbf{w}^T\mathbf{x}_n + b) > 0$$

$$\text{margin}(b,\mathbf{w}) = \min_{n=1,\dots,N} \frac{1}{\|\mathbf{w}\|} y_n(\mathbf{w}^T\mathbf{x}_n + b)$$

係數的放縮不影響我們對這個分隔平面的表示方式

- $\mathbf{w}^T\mathbf{x} + b = 0$ same as $3\mathbf{w}^T\mathbf{x} + 3b = 0$: scaling does not matter
- **special** scaling: only consider separating $(b,\mathbf{w})$ such that

$$\min_{n=1,\dots,N} y_n(\mathbf{w}^T\mathbf{x}_n + b) = 1 \Longrightarrow \text{margin}(b,\mathbf{w}) = \frac{1}{\|\mathbf{w}\|}$$

$$\max_{b,\mathbf{w}} \quad \frac{1}{\|\mathbf{w}\|}$$

$$\text{subject to} \quad \text{every } y_n(\mathbf{w}^T\mathbf{x}_n + b) > 0$$

$$\min_{n=1,\dots,N} \quad y_n(\mathbf{w}^T\mathbf{x}_n + b) = 1$$

假設我們要找的是 wTx + b = 0 這個超平面，我們對這個超平面進行縮放其實是沒有任何影響的，現在我們也將 wTx + b 進行放縮，讓它跟 yn 相乘會是 1，也就是 yn(wTXn + b) = 1，這樣原本的 margin(b,w) 就是可以轉換成 1 除以 w 的長度，我們只要求讓這個值最大的平面就可以了。

等於1一定會大於0，所以被隱含了

# Standard Large-Margin Hyperplane Problem

再簡化問題一次！因為最大化問題裡又有一個最小化的限制條件，想要將其放鬆！

$$\max_{b,\mathbf{w}} \quad \frac{1}{\|\mathbf{w}\|} \quad \text{subject to} \quad \min_{n=1,\dots,N} \; y_n(\mathbf{w}^T\mathbf{x}_n + b) = 1$$

接下來想要放鬆這個限制條件

比較鬆的條件

necessary constraints: $y_n(\mathbf{w}^T\mathbf{x}_n + b) \geq 1$ for all $n$

比較緊的條件！

original constraint: $\min_{n=1,\dots,N} \; y_n(\mathbf{w}^T\mathbf{x}_n + b) = 1$

want: optimal $(b, \mathbf{w})$ **here (inside)**

if optimal $(b, \mathbf{w})$ outside, e.g. $y_n(\mathbf{w}^T\mathbf{x}_n + b) > 1.126$ for all $n$

—can scale $(b, \mathbf{w})$ to "more optimal" $(\frac{b}{1.126}, \frac{\mathbf{w}}{1.126})$ **(contradiction!)**

假設有一個解是大於1(say 1.126)的，那我們可以同時scale b and w 使得限制式仍滿足，可是會使得scale後的解更optimal(contradict to the original optimal solution)

final change: max $\Longrightarrow$ min, remove $\sqrt{\ }$, add $\frac{1}{2}$

後面會講為什麼要加1/2這個scale

$$\min_{b,\mathbf{w}} \quad \frac{1}{2}\mathbf{w}^T\mathbf{w}$$

透過不等式把原先限制式中的min拿掉

$$\text{subject to} \quad y_n(\mathbf{w}^T\mathbf{x}_n + b) \geq 1 \text{ for all } n$$

# Fun Time

Consider three examples $(\mathbf{x}_1, +1)$, $(\mathbf{x}_2, +1)$, $(\mathbf{x}_3, -1)$, where $\mathbf{x}_1 = (3, 0)$, $\mathbf{x}_2 = (0, 4)$, $\mathbf{x}_3 = (0, 0)$. In addition, consider a hyperplane $x_1 + x_2 = 1$. Which of the following is not true?

1. the hyperplane is a separating one for the three examples
2. the distance from the hyperplane to $\mathbf{x}_1$ is 2
3. the distance from the hyperplane to $\mathbf{x}_3$ is $\frac{1}{\sqrt{2}}$
4. the example that is closest to the hyperplane is $\mathbf{x}_3$

# Fun Time

Consider three examples $(\mathbf{x}_1, +1)$, $(\mathbf{x}_2, +1)$, $(\mathbf{x}_3, -1)$, where $\mathbf{x}_1 = (3, 0)$, $\mathbf{x}_2 = (0, 4)$, $\mathbf{x}_3 = (0, 0)$. In addition, consider a hyperplane $x_1 + x_2 = 1$. Which of the following is not true?

1. the hyperplane is a separating one for the three examples
2. the distance from the hyperplane to $\mathbf{x}_1$ is 2
3. the distance from the hyperplane to $\mathbf{x}_3$ is $\frac{1}{\sqrt{2}}$
4. the example that is closest to the hyperplane is $\mathbf{x}_3$

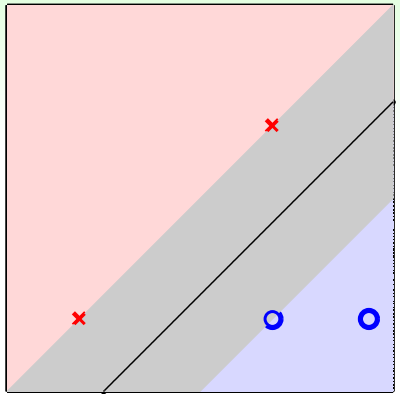
## Reference Answer: 2

The distance from the hyperplane to $\mathbf{x}_1$ is $\frac{1}{\sqrt{2}}(3 + 0 - 1) = \sqrt{2}$.

# Solving a Particular Standard Problem

要找最胖的那條線之標準問題：

$$\begin{array}{ll} \min\limits_{b,\mathbf{w}} & \frac{1}{2}\mathbf{w}^T\mathbf{w} \\ \text{subject to} & y_n(\mathbf{w}^T\mathbf{x}_n + b) \geq 1 \text{ for all } n \end{array}$$



每一組的點都對應
到四個不同的條件

$$\mathrm{X} = \begin{bmatrix} 0 & 0 \\ 2 & 2 \\ 2 & 0 \\ 3 & 0 \end{bmatrix} \quad \mathbf{y} = \begin{bmatrix} -1 \\ -1 \\ +1 \\ +1 \end{bmatrix}$$

$$\begin{array}{rrrl} & & -b \geq 1 & (i) \\ -2w_1 & -2w_2 & -b \geq 1 & (ii) \\ 2w_1 & & +b \geq 1 & (iii) \\ 3w_1 & & +b \geq 1 & (iv) \end{array}$$

- $\left.\begin{array}{lll} (i) & \& & (iii) \implies w_1 \geq +1 \\ (ii) & \& & (iii) \implies w_2 \leq -1 \end{array}\right\} \implies \frac{1}{2}\mathbf{w}^T\mathbf{w} \geq 1$

W^T*W = w1^2 + w2^2 >= 2

- $(w_1 = 1, w_2 = -1, b = -1)$ at **lower bound** and satisfies $(i) - (iv)$

X1 – X2 –1 = 0 這條線是解 (W1x1–W2x2+b=0)

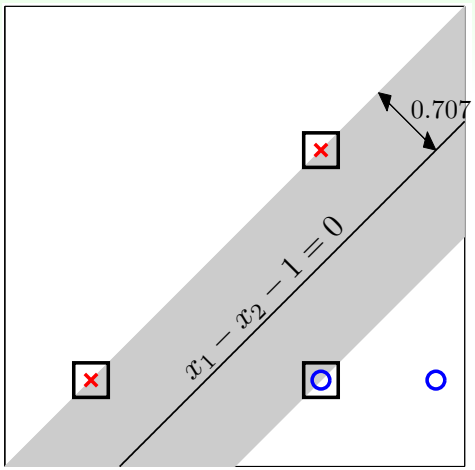
這裡的g_SVM就是那條可以長到最胖的線，也就是我們所稱的“支撐向量機”

$$g_{\text{SVM}}(\mathbf{x}) = \text{sign}(x_1 - x_2 - 1)\text{: SVM? :-)}$$

# Support Vector Machine (SVM)

optimal solution: $(w_1 = 1, w_2 = -1, b = -1)$

margin$(b, \mathbf{w})$ $= \frac{1}{\|\mathbf{w}\|} = \frac{1}{\sqrt{2}}$




- examples on boundary: 'locates' fattest hyperplane
  other examples: **not needed**
- call boundary example **support vector** (candidate)

那些座落於邊界上的點，支撐了整個margin，所以稱他們為之稱向量。

**support vector** machine (SVM):
learn **fattest hyperplanes**
(with help of **support vectors** )

# Solving General SVM

$$\min_{b,\mathbf{w}} \quad \frac{1}{2}\mathbf{w}^T\mathbf{w}$$

$$\text{subject to} \quad y_n(\mathbf{w}^T\mathbf{x}_n + b) \geq 1 \text{ for all } n$$

- **not easy manually, of course :-)**
- gradient descent? **not easy with constraints**
- luckily:
  - (convex) quadratic objective function of $(b, \mathbf{w})$
  - linear constraints of $(b, \mathbf{w})$

—**quadratic programming** 二次規劃（線性規劃的進階版）

**quadratic programming** (QP):
'easy' optimization problem

# Quadratic Programming

optimal $(b, \mathbf{w}) = ?$

$$\min_{b,\mathbf{w}} \quad \frac{1}{2}\mathbf{w}^T\mathbf{w}$$

$$\text{subject to} \quad y_n(\mathbf{w}^T\mathbf{x}_n + b) \geq 1, \quad \text{for } n = 1, 2, \ldots, N$$

optimal $\mathbf{u} \leftarrow \text{QP}(Q, \mathbf{p}, A, \mathbf{c})$

一次項係數放在p矩陣裡面

$$\min_{\mathbf{u}} \quad \frac{1}{2}\mathbf{u}^T Q\mathbf{u} + \mathbf{p}^T\mathbf{u}$$

二次項係數放在Q矩陣裡面

$$\text{subject to} \quad \mathbf{a}_m^T\mathbf{u} \geq c_m, \quad \text{for } m = 1, 2, \ldots, M$$

限制式要線性的，係數放在a裡面　常數放在c裡面

因為objective function裡面我們只注重w向量

$$\text{objective function:} \quad \mathbf{u} = \begin{bmatrix} b \\ \mathbf{w} \end{bmatrix}; Q = \begin{bmatrix} 0 & \mathbf{0}_d^T \\ \mathbf{0}_d & \mathrm{I}_d \end{bmatrix}; \mathbf{p} = \mathbf{0}_{d+1}$$

$$\text{constraints:} \quad \mathbf{a}_n^T = y_n \begin{bmatrix} 1 & \mathbf{x}_n^T \end{bmatrix}; c_n = 1; M = N$$

SVM with general QP solver:
easy **if you've read the manual :-)**

# SVM with QP Solver

## Linear Hard-Margin SVM Algorithm

1. $Q = \begin{bmatrix} 0 & \mathbf{0}_d^T \\ \mathbf{0}_d & I_d \end{bmatrix}$; $\mathbf{p} = \mathbf{0}_{d+1}$; $\mathbf{a}_n^T = y_n \begin{bmatrix} 1 & \mathbf{x}_n^T \end{bmatrix}$; $c_n = 1$
2. $\begin{bmatrix} b \\ \mathbf{w} \end{bmatrix} \leftarrow \text{QP}(Q, \mathbf{p}, A, \mathbf{c})$
3. return $b$ & $\mathbf{w}$ as $g_{\text{SVM}}$

- **hard-margin**: nothing violate 'fat boundary'
- **linear**: $\mathbf{x}_n$

want **non-linear**?
$\mathbf{z}_n = \mathbf{\Phi}(\mathbf{x}_n)$—**remember? :-)**

# Fun Time

Consider two negative examples with $\mathbf{x}_1 = (0, 0)$ and $\mathbf{x}_2 = (2, 2)$; two positive examples with $\mathbf{x}_3 = (2, 0)$ and $\mathbf{x}_4 = (3, 0)$, as shown on page 17 of the slides. Define $\mathbf{u}$, $\mathrm{Q}$, $\mathbf{p}$, $c_n$ as those listed on page 20 of the slides. What are $\mathbf{a}_n^T$ that need to be fed into the QP solver?

1. $\mathbf{a}_1^T = [-1, 0, 0]$ , $\mathbf{a}_2^T = [-1, 2, 2]$ , $\mathbf{a}_3^T = [-1, 2, 0]$ , $\mathbf{a}_4^T = [-1, 3, 0]$
2. $\mathbf{a}_1^T = [1, 0, 0]$ , $\mathbf{a}_2^T = [1, -2, -2]$ , $\mathbf{a}_3^T = [-1, 2, 0]$ , $\mathbf{a}_4^T = [-1, 3, 0]$
3. $\mathbf{a}_1^T = [1, 0, 0]$ , $\mathbf{a}_2^T = [1, 2, 2]$ , $\mathbf{a}_3^T = [1, 2, 0]$ , $\mathbf{a}_4^T = [1, 3, 0]$
4. $\mathbf{a}_1^T = [-1, 0, 0]$ , $\mathbf{a}_2^T = [-1, -2, -2]$ , $\mathbf{a}_3^T = [1, 2, 0]$ , $\mathbf{a}_4^T = [1, 3, 0]$

# Fun Time

Consider two negative examples with $\mathbf{x}_1 = (0, 0)$ and $\mathbf{x}_2 = (2, 2)$; two positive examples with $\mathbf{x}_3 = (2, 0)$ and $\mathbf{x}_4 = (3, 0)$, as shown on page 17 of the slides. Define $\mathbf{u}$, $Q$, $\mathbf{p}$, $c_n$ as those listed on page 20 of the slides. What are $\mathbf{a}_n^T$ that need to be fed into the QP solver?

1. $\mathbf{a}_1^T = [-1, 0, 0]$ , $\mathbf{a}_2^T = [-1, 2, 2]$ , $\mathbf{a}_3^T = [-1, 2, 0]$ , $\mathbf{a}_4^T = [-1, 3, 0]$
2. $\mathbf{a}_1^T = [1, 0, 0]$ , $\mathbf{a}_2^T = [1, -2, -2]$ , $\mathbf{a}_3^T = [-1, 2, 0]$ , $\mathbf{a}_4^T = [-1, 3, 0]$
3. $\mathbf{a}_1^T = [1, 0, 0]$ , $\mathbf{a}_2^T = [1, 2, 2]$ , $\mathbf{a}_3^T = [1, 2, 0]$ , $\mathbf{a}_4^T = [1, 3, 0]$
4. $\mathbf{a}_1^T = [-1, 0, 0]$ , $\mathbf{a}_2^T = [-1, -2, -2]$ , $\mathbf{a}_3^T = [1, 2, 0]$ , $\mathbf{a}_4^T = [1, 3, 0]$

**Reference Answer: 4**

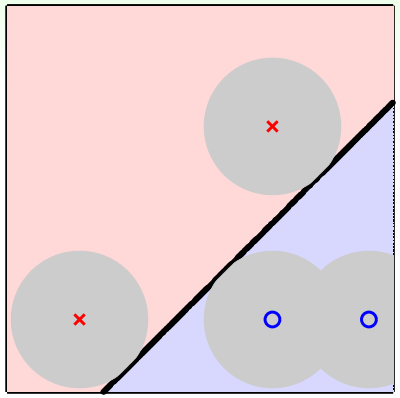
We need $\mathbf{a}_n^T = y_n \begin{bmatrix} 1 & \mathbf{x}_n^T \end{bmatrix}$.

# Why Large-Margin Hyperplane?

為什麼SVM(長得胖胖的線)會做得好？有沒有理論上的保證？

$$\min_{b,\mathbf{w}} \quad \frac{1}{2}\mathbf{w}^T\mathbf{w}$$

$$\text{subject to} \quad y_n(\mathbf{w}^T\mathbf{z}_n + b) \geq 1 \text{ for all } n$$



| | minimize | constraint |
|---|---|---|
| regularization | $E_{\text{in}}$ | $\mathbf{w}^T\mathbf{w} \leq C$ |
| SVM | $\mathbf{w}^T\mathbf{w}$ | $E_{\text{in}} = 0$ [and more] |

SVM (large-margin hyperplane):
**'weight-decay regularization' within $E_{\text{in}} = 0$**

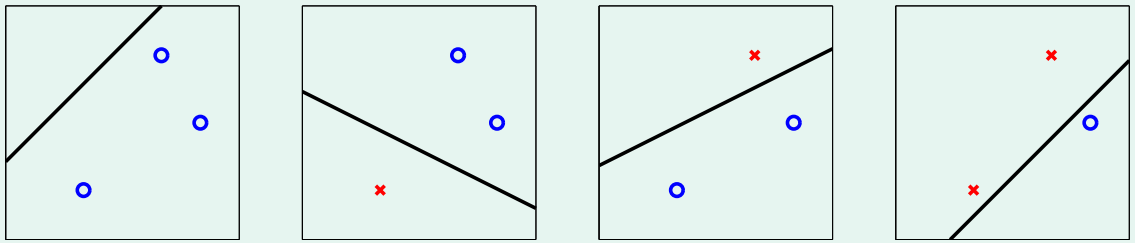
到頭來，SVM也是一種regularization

# Large-Margin Restricts Dichotomies

consider 'large-margin algorithm' $\mathcal{A}_\rho$:

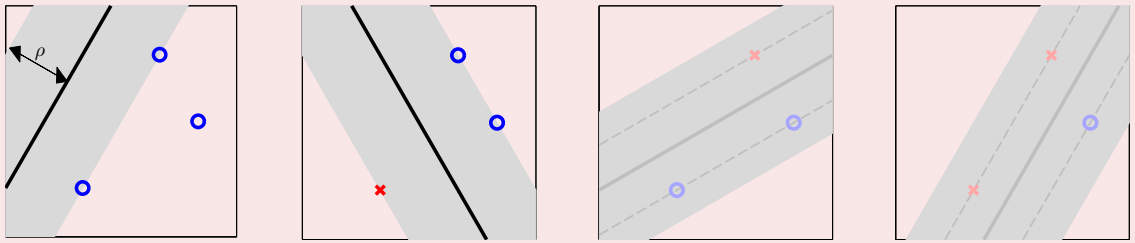
either **returns $g$ with margin$(g) \geq \rho$ (if exists)**, or 0 otherwise

## $\mathcal{A}_0$: like PLA $\Longrightarrow$ shatter 'general' 3 inputs



## $\mathcal{A}_{1.126}$: more strict than SVM $\Longrightarrow$ cannot shatter any 3 inputs

我要找出
比1.126還
要胖的線




fewer dichotomies $\Longrightarrow$ smaller 'VC dim.' $\Longrightarrow$ **better generalization**

# VC Dimension of Large-Margin Algorithm

fewer dichotomies $\Longrightarrow$ smaller **'VC dim.'**

**considers $d_{VC}(\mathcal{A}_\rho)$ [data-dependent, need more than VC]**

instead of $d_{VC}(\mathcal{H})$ [data-independent, covered by VC]

## $d_{VC}(\mathcal{A}_\rho)$ when $\mathcal{X}$ = unit circle in $\mathbb{R}^2$

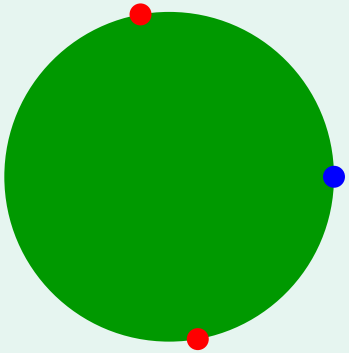
- $\rho = 0$: just perceptrons ($d_{VC} = 3$)

現在我的線要長得胖過二分之根號三

- $\rho > \frac{\sqrt{3}}{2}$: cannot shatter any 3 inputs ($d_{VC} < 3$)

  —some inputs must be of **distance** $\leq \sqrt{3}$

因為單位原上相異的三點，至少有兩點的距離會小於根號三（正三角形的關係），所以這種線沒有可能！



generally, when $\mathcal{X}$ in radius-$R$ hyperball:

$$d_{VC}(\mathcal{A}_\rho) \leq \min\left(\frac{R^2}{\rho^2}, d\right) + 1 \leq \underbrace{d+1}_{d_{VC}(\text{perceptrons})}$$

這個式子告訴我們，透過對線胖瘦的設定，我們可以把VC dimension真的縮減到比原來的還小

# Benefits of Large-Margin Hyperplanes

| | large-margin hyperplanes | hyperplanes | hyperplanes + feature transform $\Phi$ |
|---|---|---|---|
| # | even fewer | **not many** | many |
| boundary | simple | simple | **sophisticated** |

- **not many** good, for $d_{VC}$ and generalization
- **sophisticated** good, for possibly better $E_{in}$

a new possibility: non-linear SVM

| | **large-margin hyperplanes + numerous feature transform $\Phi$** |
|---|---|
| # | **not many** |
| boundary | **sophisticated** |

# Fun Time

Consider running the 'large-margin algorithm' $\mathcal{A}_\rho$ with $\rho = \frac{1}{4}$ on a $\mathcal{Z}$-space such that $\mathbf{z} = \mathbf{\Phi}(\mathbf{x})$ is of 1126 dimensions (excluding $z_0$) and $\|\mathbf{z}\| \leq 1$. What is the upper bound of $d_{\text{VC}}(\mathcal{A}_\rho)$ when calculated by $\min\left(\frac{R^2}{\rho^2}, d\right) + 1$?

1. 5
2. 17
3. 1126
4. 1127

# Fun Time

Consider running the 'large-margin algorithm' $\mathcal{A}_\rho$ with $\rho = \frac{1}{4}$ on a $\mathcal{Z}$-space such that $\mathbf{z} = \mathbf{\Phi}(\mathbf{x})$ is of 1126 dimensions (excluding $z_0$) and $\|\mathbf{z}\| \leq 1$. What is the upper bound of $d_{\text{VC}}(\mathcal{A}_\rho)$ when calculated by $\min\left(\frac{R^2}{\rho^2}, d\right) + 1$?

1. 5
2. 17
3. 1126
4. 1127

**Reference Answer: 2**

By the description, $d = 1126$ and $R = 1$. So the upper bound is simply 17.

# Summary

1. Embedding Numerous Features: Kernel Models
泛指所有使用kernal方法轉換non–linear的model

## Lecture 1: Linear Support Vector Machine

- Course Introduction
  **from foundations to techniques**
- Large-Margin Separating Hyperplane
  **intuitively more robust against noise**
- Standard Large-Margin Problem

**minimize 'length of w' at special separating scale**

- Support Vector Machine
  **'easy' via quadratic programming**
- Reasons behind Large-Margin Hyperplane
  **fewer dichotomies and better generalization**

- **next: solving non-linear Support Vector Machine**

2. Combining Predictive Features: Aggregation Models
3. Distilling Implicit Features: Extraction Models