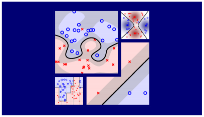
Machine Learning Techniques
(機器學習技法)



Lecture 1: Linear Support Vector Machine

Hsuan-Tien Lin (林軒田)
htlin@csie.ntu.edu.tw

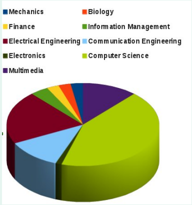
Department of Computer Science
& Information Engineering

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

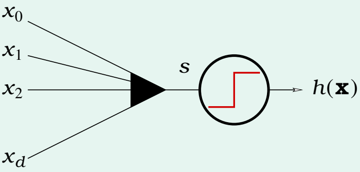
- Course Introduction
- Large-Margin Separating Hyperplane
- Standard Large-Margin Problem
- Support Vector Machine
- Reasons behind Large-Margin Hyperplane

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

Linear Classification Revisited

PLA/pocket

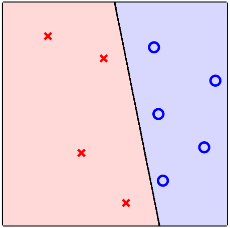
$$h(\mathbf{x}) = \text{sign}(s)$$



plausible err = 0/1

(small flipping noise)

minimize **specially**

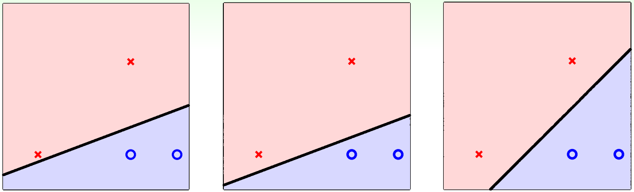


(linear separable)

linear (hyperplane) classifiers:

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

Which Line Is Best?

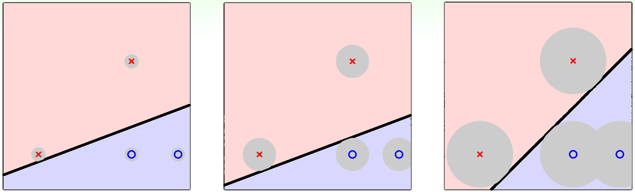


- PLA? depending on randomness
- VC bound? whichever you like!

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

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

Why Rightmost Hyperplane?



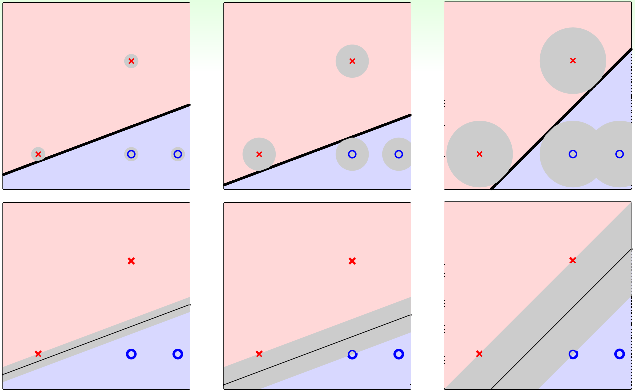
informal argument

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 $\mathbf{x}_n$**

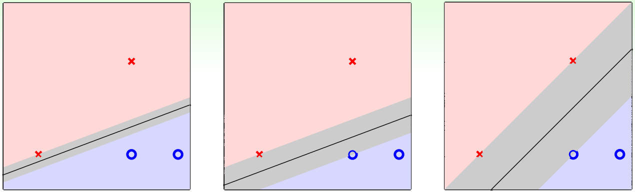
Fat Hyperplane



- 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



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

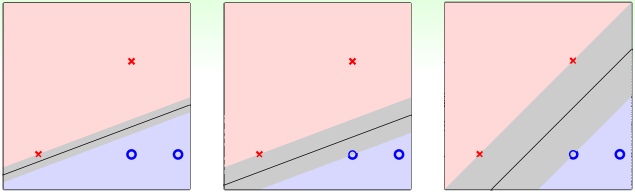
$$\text{subject to} \quad \mathbf{w} \text{ classifies every } (\mathbf{x}_n, y_n) \text{ correctly}$$

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

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

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

Large-Margin Separating Hyperplane



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

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

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

- 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 $\mathbf{v} \in \mathbb{R}^d$.

Distance to Hyperplane: Preliminary

$$\max_{\mathbf{w}} \quad \text{margin}(\mathbf{w})$$
$$\text{subject to} \quad \text{every } y_n \mathbf{w}^T \mathbf{x}_n > 0$$
$$\text{margin}(\mathbf{w}) = \min_{n=1,\ldots,N} \text{distance}(\mathbf{x}_n, \mathbf{w})$$

'shorten' x and w

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

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

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

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

Distance to Hyperplane

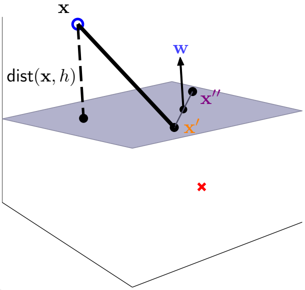
want: distance($\mathbf{x}$, b, $\mathbf{w}$), with hyperplane $\mathbf{w}^T\mathbf{x}' + b = 0$

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:

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

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



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

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:

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

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

Margin of Special Separating Hyperplane

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

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

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

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

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

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

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

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

Standard Large-Margin Hyperplane Problem

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

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

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

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

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

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?

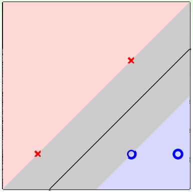
1. the hyperplane is a separating one for the three examples
2. the distance from the hyperplane to $\mathbf{x}_1$ is 2
3. the distance from the hyperplane to $\mathbf{x}_3$ is $\frac{1}{\sqrt{2}}$
4. the example that is closest to the hyperplane is $\mathbf{x}_3$

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

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



$$\mathrm{X} = \begin{bmatrix} 0 & 0 \\ 2 & 2 \\ 2 & 0 \\ 3 & 0 \end{bmatrix} \quad \mathbf{y} = \begin{bmatrix} -1 \\ -1 \\ +1 \\ +1 \end{bmatrix} \quad \begin{array}{rrrl} & & -b \geq 1 & (i) \\ -2w_1 & -2w_2 & -b \geq 1 & (ii) \\ 2w_1 & & +b \geq 1 & (iii) \\ 3w_1 & & +b \geq 1 & (iv) \end{array}$$

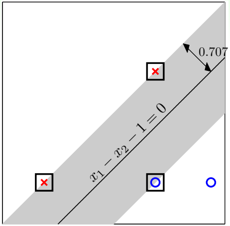
- $\left\{ \begin{array}{lclcl} (i) & \& & (iii) & \Longrightarrow & w_1 \geq +1 \\ (ii) & \& & (iii) & \Longrightarrow & w_2 \leq -1 \end{array} \right\} \Longrightarrow \frac{1}{2}\mathbf{w}^T\mathbf{w} \geq 1$
- $(w_1 = 1, w_2 = -1, b = -1)$ at **lower bound** and satisfies $(i) - (iv)$

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

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)

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**

quadratic programming (QP):
'easy' optimization problem

Quadratic Programming

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

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

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

for $n = 1, 2, \ldots, N$

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

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

subject to $\quad \mathbf{a}_m^T\mathbf{u} \geq c_m,$

for $m = 1, 2, \ldots, M$

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

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. $\mathrm{Q} = \begin{bmatrix} 0 & \mathbf{0}_d^T \\ \mathbf{0}_d & \mathrm{I}_d \end{bmatrix}$; $\mathbf{p} = \mathbf{0}_{d+1}$; $\mathbf{a}_n^T = y_n \begin{bmatrix} 1 & \mathbf{x}_n^T \end{bmatrix}$; $c_n = 1$
2. $\begin{bmatrix} b \\ \mathbf{w} \end{bmatrix} \leftarrow \mathrm{QP}(\mathrm{Q}, \mathbf{p}, \mathrm{A}, \mathbf{c})$
3. return b & $\mathbf{w}$ as g_{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 $\mathbf{u}$, 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. $\mathbf{a}_1^T = [-1, 0, 0]$, $\mathbf{a}_2^T = [-1, -2, -2]$, $\mathbf{a}_3^T = [1, 2, 0]$, $\mathbf{a}_4^T = [1, 3, 0]$

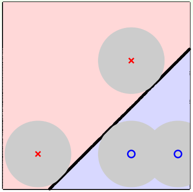
Reference Answer: 4

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

Why Large-Margin Hyperplane?

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

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



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

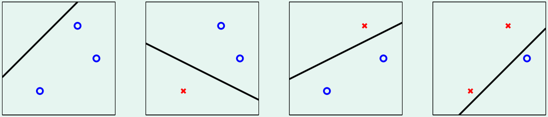
SVM (large-margin hyperplane):
'weight-decay regularization' within $E_{\text{in}} = 0$

Large-Margin Restricts Dichotomies

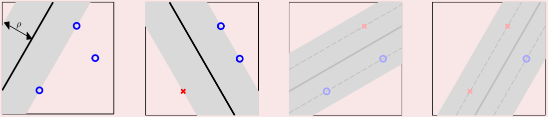
consider 'large-margin algorithm' $\mathcal{A}_\rho$:

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



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

VC Dimension of Large-Margin Algorithm

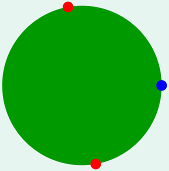
fewer dichotomies $\Longrightarrow$ smaller **'VC dim.'**

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

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

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

- $\rho = 0$: just perceptrons ($d_{\mathrm{VC}} = 3$)
- $\rho > \frac{\sqrt{3}}{2}$: cannot shatter any 3 inputs ($d_{\mathrm{VC}} < 3$)
  —some inputs must be of **distance** $\leq \sqrt{3}$



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

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

Benefits of Large-Margin Hyperplanes

	large-margin hyperplanes	hyperplanes	hyperplanes + feature transform $\mathbf{\Phi}$
#	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 $\mathbf{\Phi}$
#	**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

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