# Machine Learning
# (機器學習)

## Lecture 06: Beyond Basic Linear Models

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)

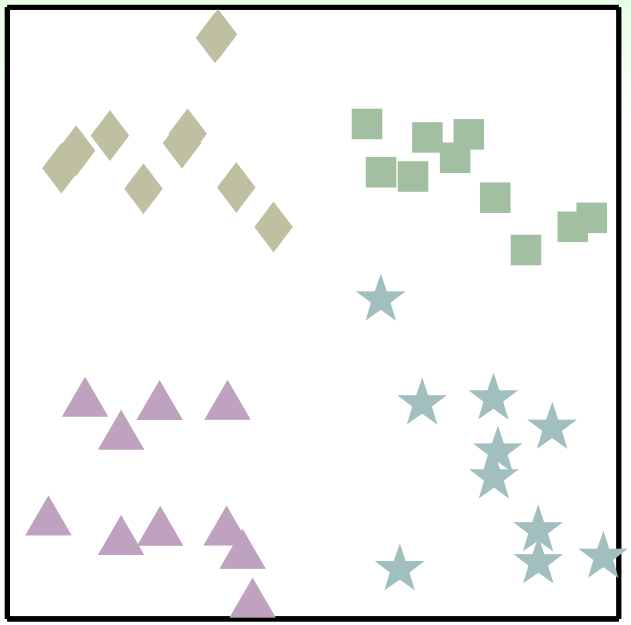
# Roadmap

1. When Can Machines Learn?
2. Why Can Machines Learn?
3. **How** Can Machines Learn?

## Lecture 06: Beyond Basic Linear Models

- Multiclass via Logistic Regression
- Multiclass via Binary Classification
- Quadratic Hypotheses
- Nonlinear Transform
- Price of Nonlinear Transform
- Structured Hypothesis Sets

# Multiclass Classification



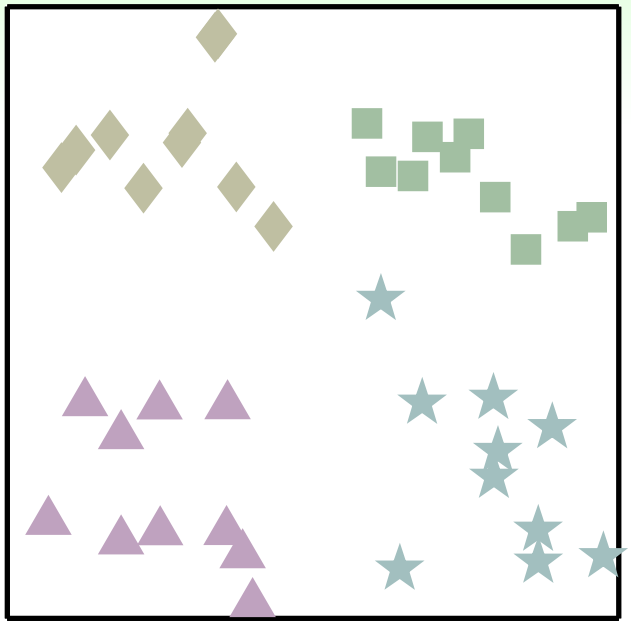
變選擇題

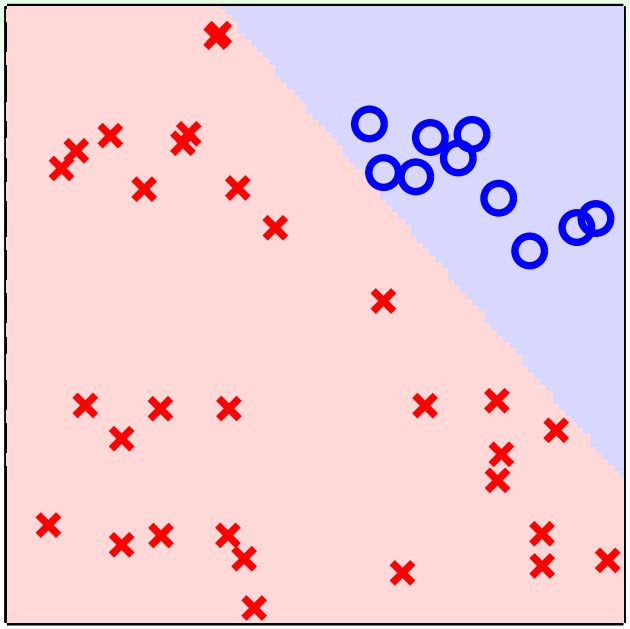
- $\mathcal{Y} = \{\square, \lozenge, \triangle, \star\}$ (4-class classification)
- **many applications** in practice, especially for 'recognition'

Binary classification

next: use **tools for** $\{\times, \circ\}$ **classification** to $\{\square, \lozenge, \triangle, \star\}$ classification

# One Class at a Time

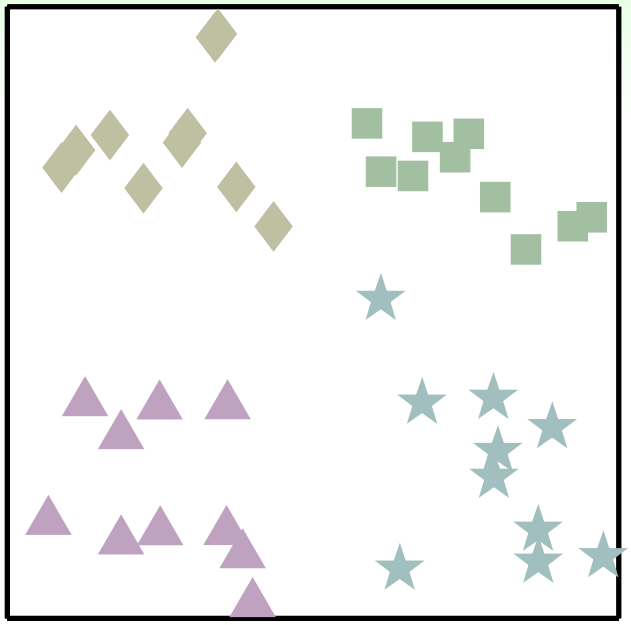


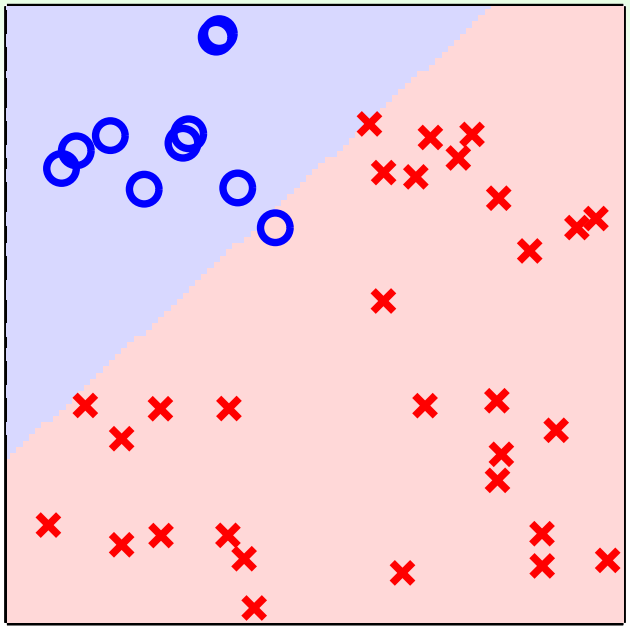


□ or not? $\{\square = \circ, \lozenge = \times, \triangle = \times, \star = \times\}$

把口跟其他人分开

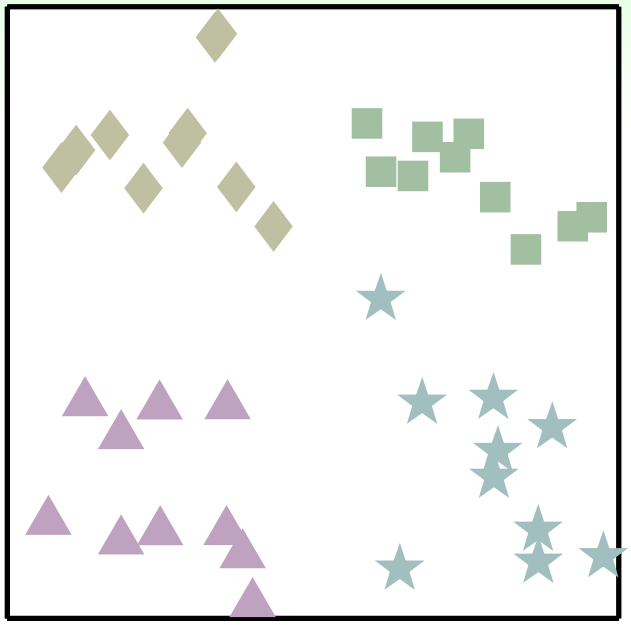
# One Class at a Time

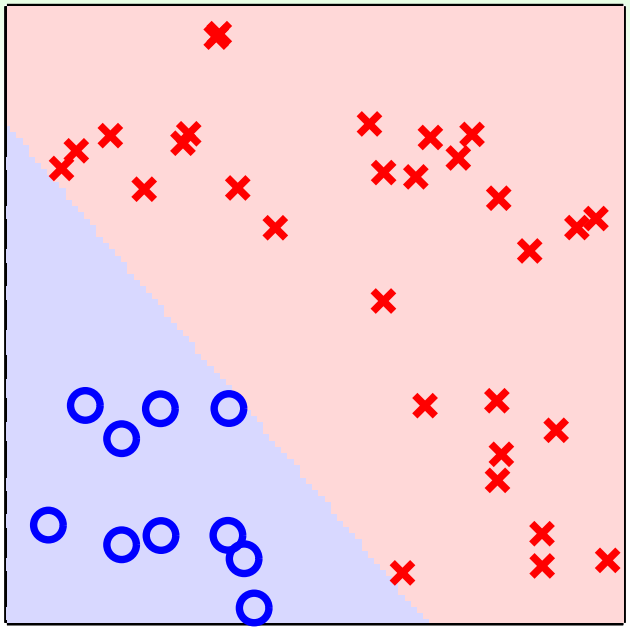




$\Diamond$ or not? $\{\square = \times, \Diamond = \circ, \triangle = \times, \star = \times\}$

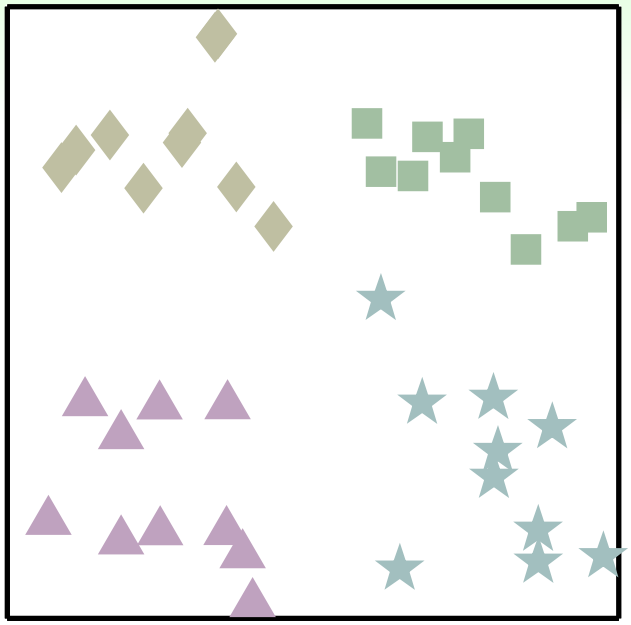
# One Class at a Time

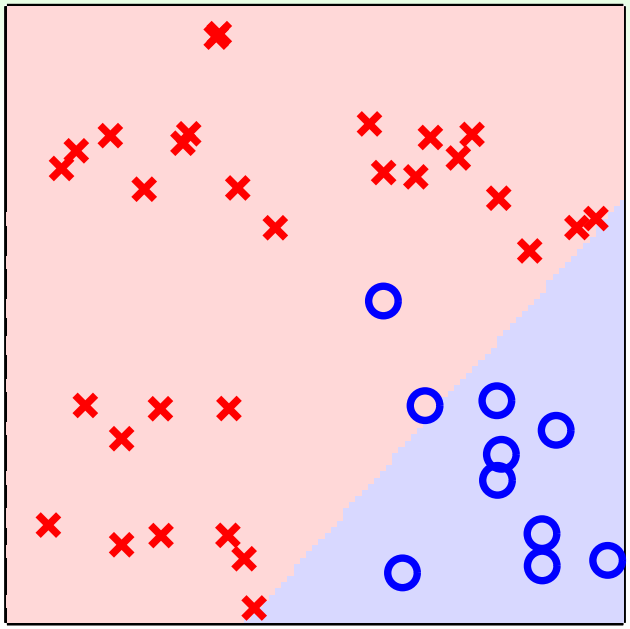




△ or not? $\{\square = \times, \Diamond = \times, \triangle = \circ, \star = \times\}$

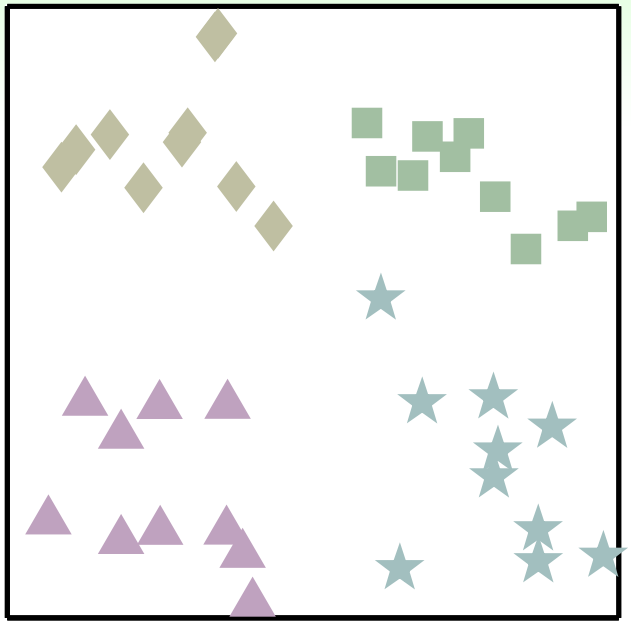
# One Class at a Time

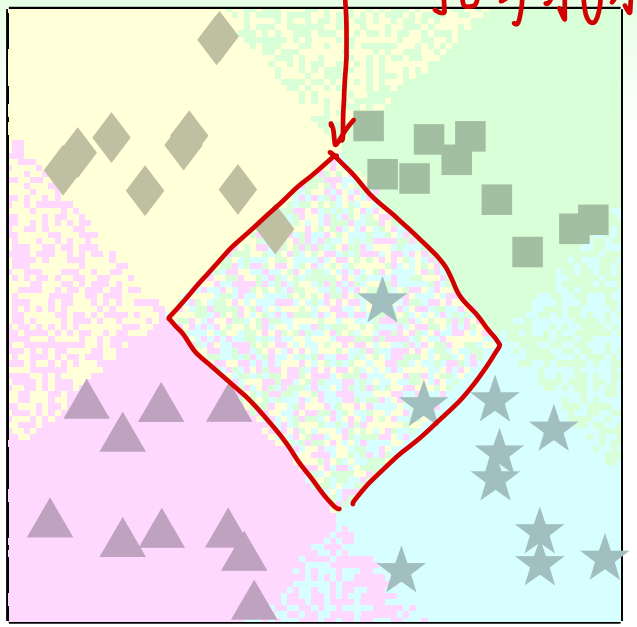


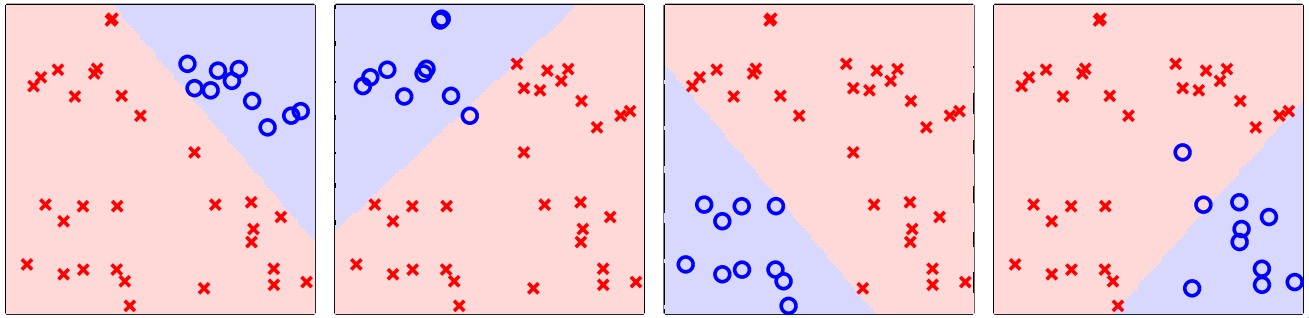


$\star$ or not? $\{\square = \times, \lozenge = \times, \triangle = \times, \star = \circ\}$

# Multiclass Prediction: Combine Binary Classifiers



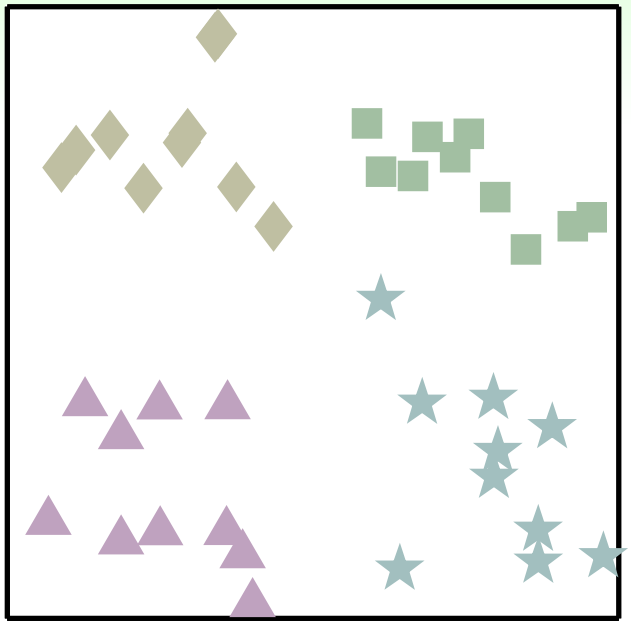


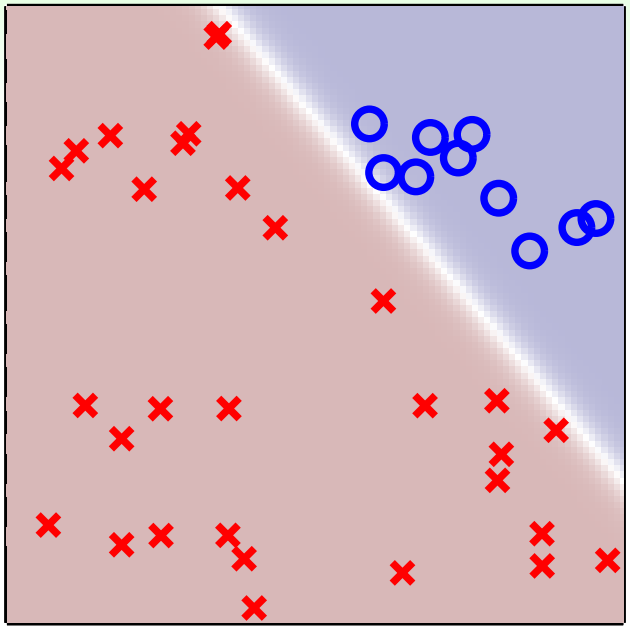



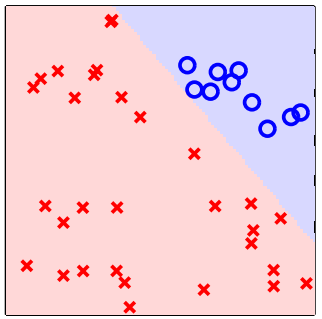
四个分类器？

but **ties**? :-)

# One Class at a Time **Softly**



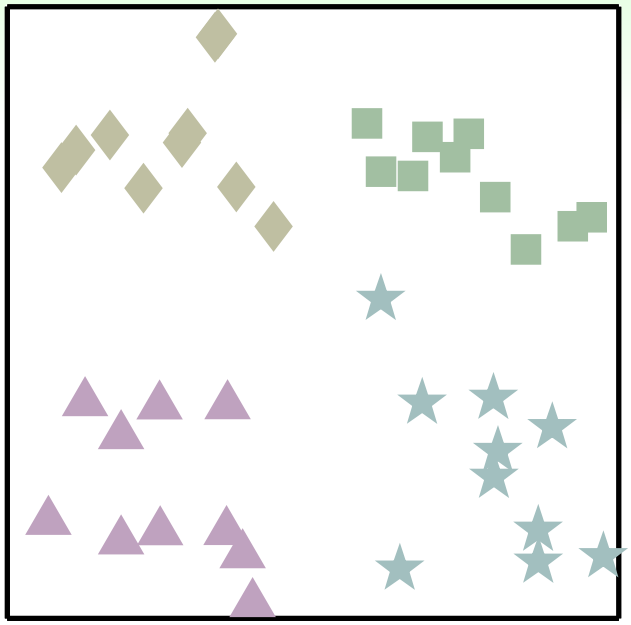


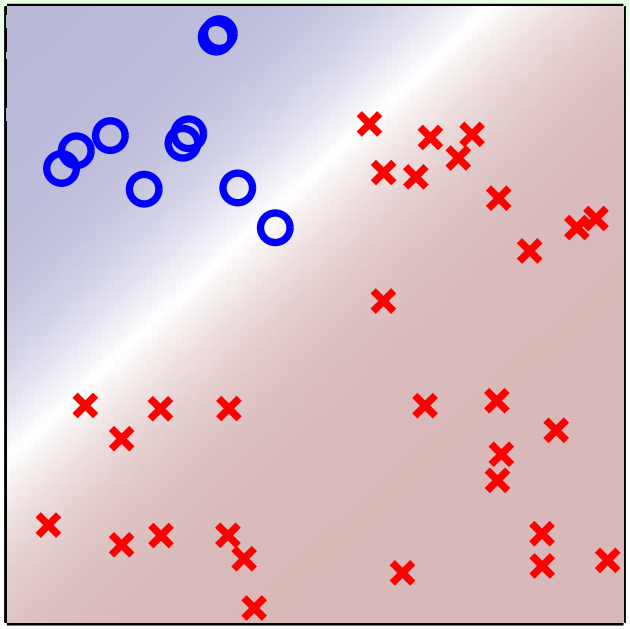


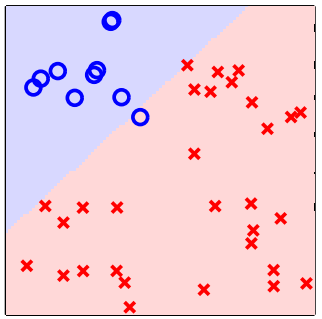
logistic regression 來估計

$$P(\square|\mathbf{x})? \ \{\square = \circ, \diamond = \times, \triangle = \times, \star = \times\}$$

# One Class at a Time **Softly**

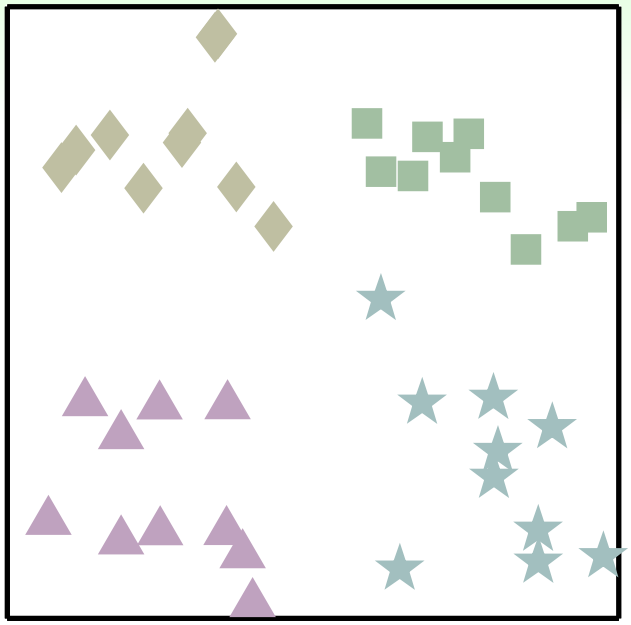


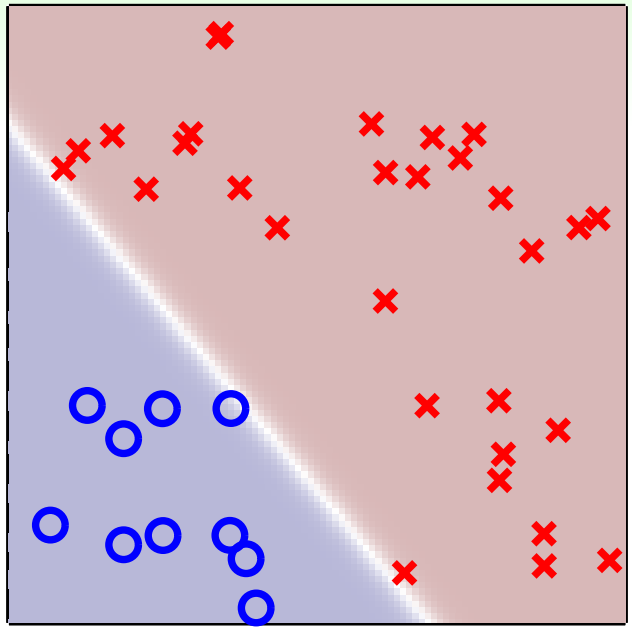


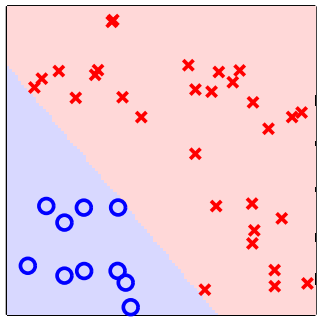


$P(\diamondsuit|\mathbf{x})$? $\{\square = \times, \diamondsuit = \circ, \triangle = \times, \star = \times\}$

# One Class at a Time **Softly**

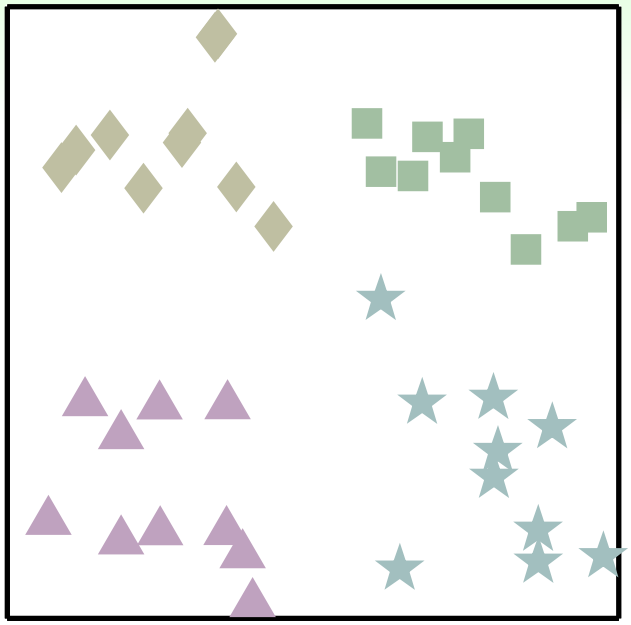


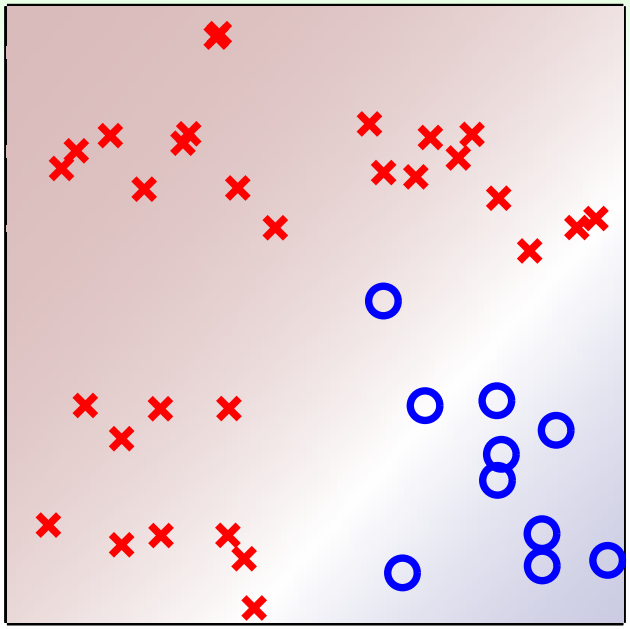


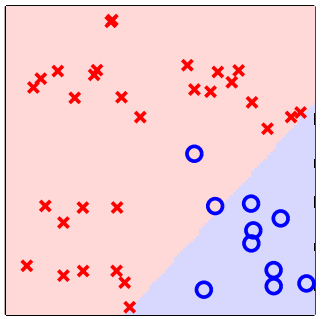


$$P(\triangle|\mathbf{x})? \ \{\square = \times, \lozenge = \times, \triangle = \circ, \star = \times\}$$

# One Class at a Time **Softly**

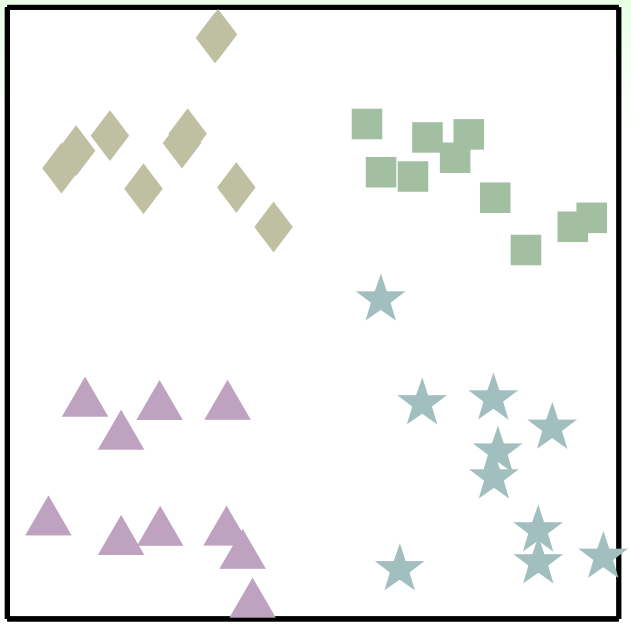


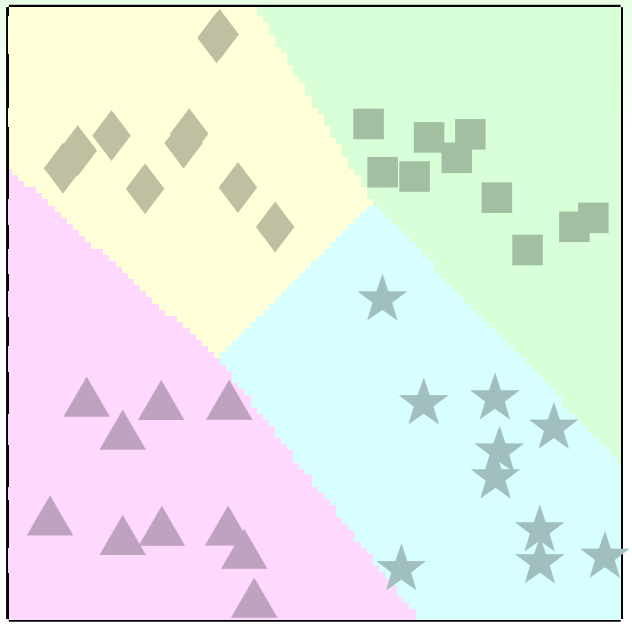


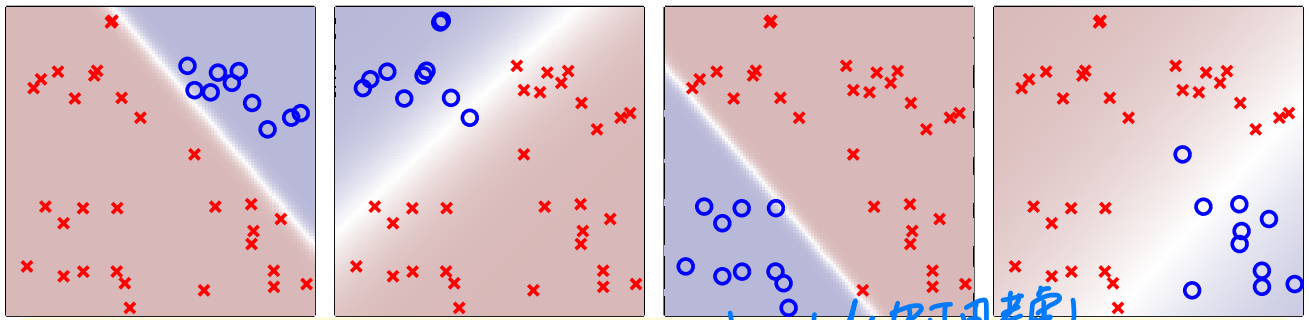


$P(\star|\mathbf{x})$? $\{\square = \times, \Diamond = \times, \triangle = \times, \star = \circ\}$

# Multiclass Prediction: Combine Soft Classifiers







$$g(\mathbf{x}) = \text{argmax}_{k \in \mathcal{Y}} \theta\left(\mathbf{w}_{[k]}^T \mathbf{x}\right)$$

sigmoid(也可用softmax)

选可能性最大者

# One-Versus-All (OVA) Decomposition

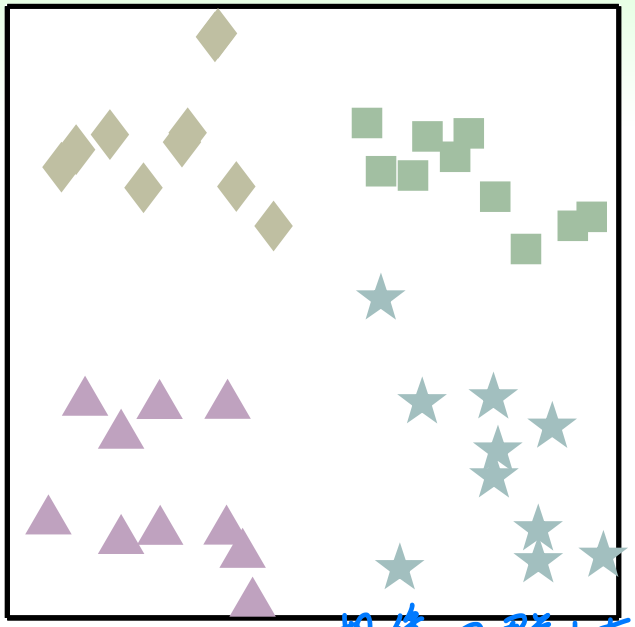
1. for $k \in \mathcal{Y}$
   obtain $\mathbf{w}_{[k]}$ by running logistic regression on
   $$\mathcal{D}_{[k]} = \{(\mathbf{x}_n, y'_n = 2[\![y_n = k]\!] - 1)\}_{n=1}^{N}$$
2. return $g(\mathbf{x}) = \text{argmax}_{k \in \mathcal{Y}} \left(\mathbf{w}_{[k]}^T \mathbf{x}\right)$

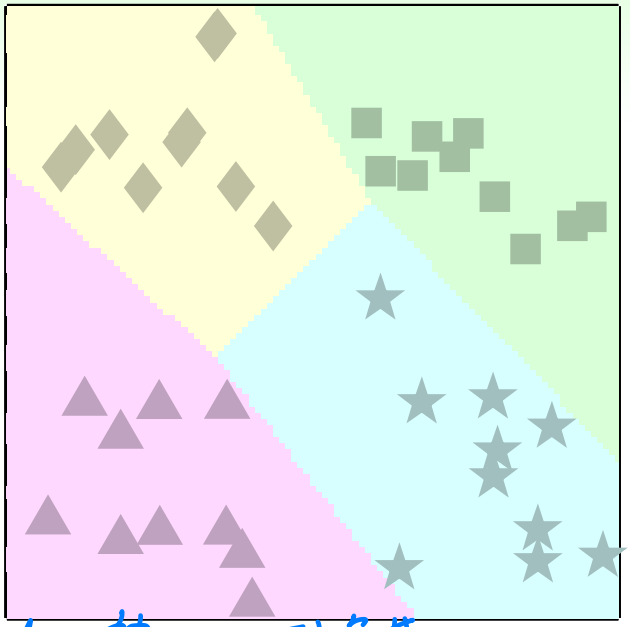
- pros: efficient,
  can be coupled with any logistic regression-like approaches
- cons: often **unbalanced** $\mathcal{D}_{[k]}$ when $K$ large
- extension: **multinomial ('coupled') logistic regression**

OVA: a simple multiclass **meta-algorithm**
to keep in your toolbox

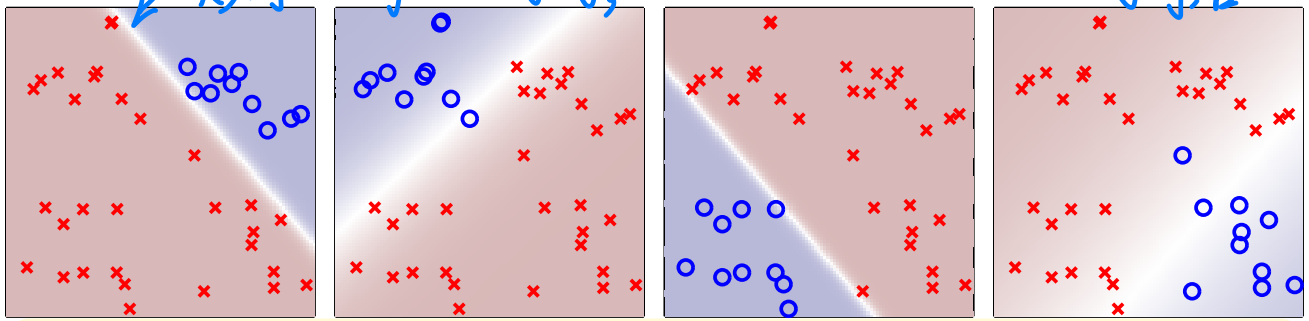
# Questions?

# Source of **Unbalance**: One versus **All**





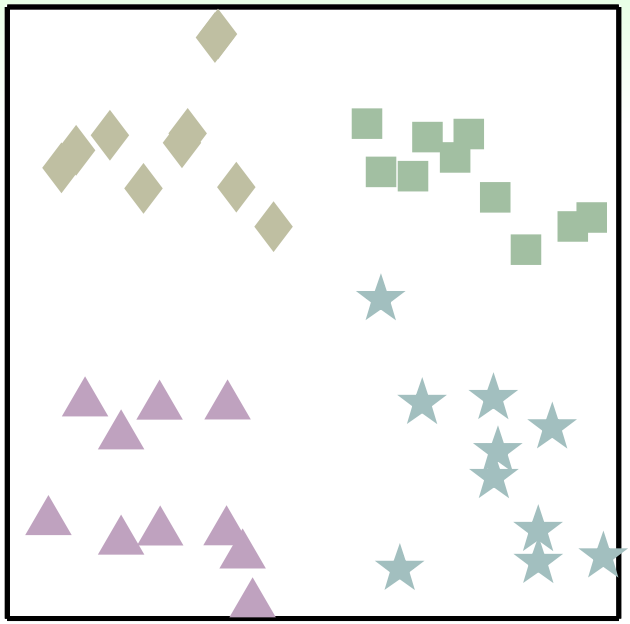
想像 2群人在对打, unbalance data 被corner到角落

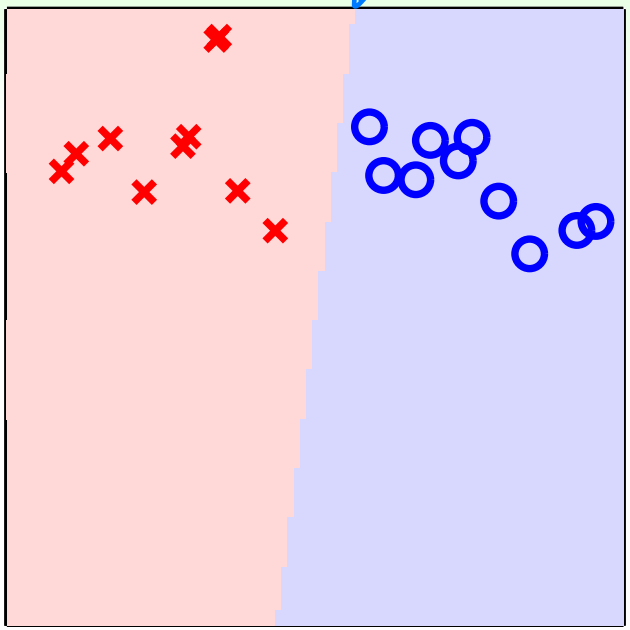


idea: make binary classification problems more **balanced** by one versus **one**

# One versus One at a Time

所以选2个class 出来对打

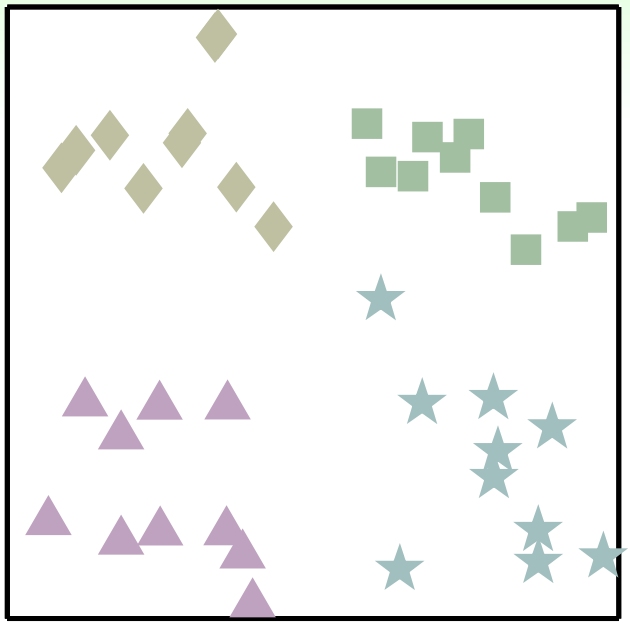


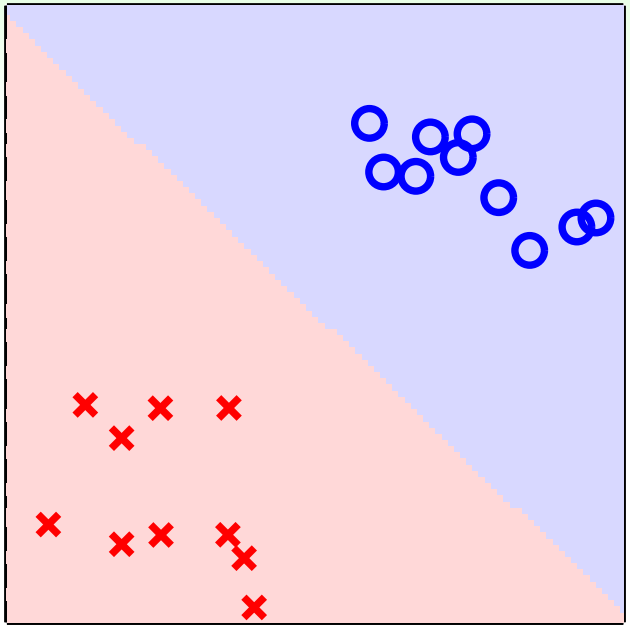


□ or ◊? {□ = ○, ◊ = ×, △ = nil, ⋆ = nil}

只用2个classes

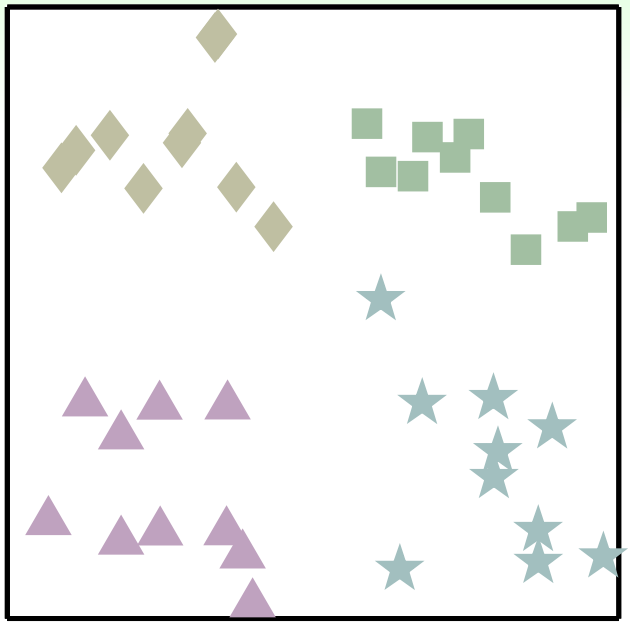
# One versus One at a Time

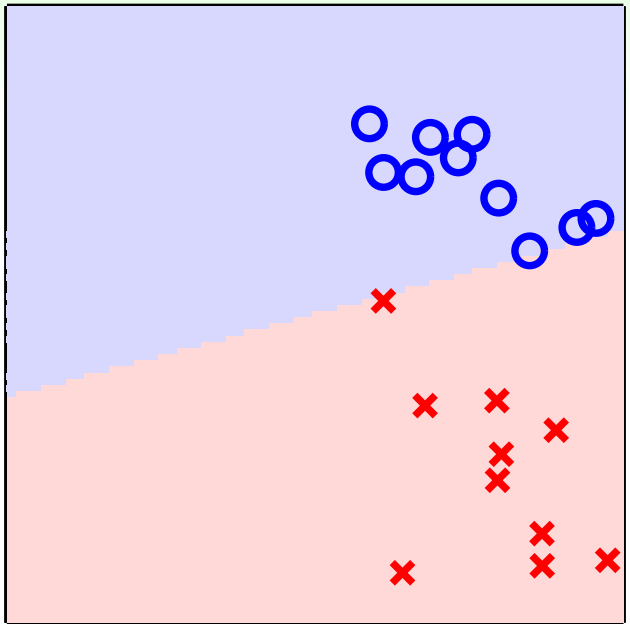




$\square$ or $\triangle$? $\{\square = \circ, \lozenge = \text{nil}, \triangle = \times, \star = \text{nil}\}$

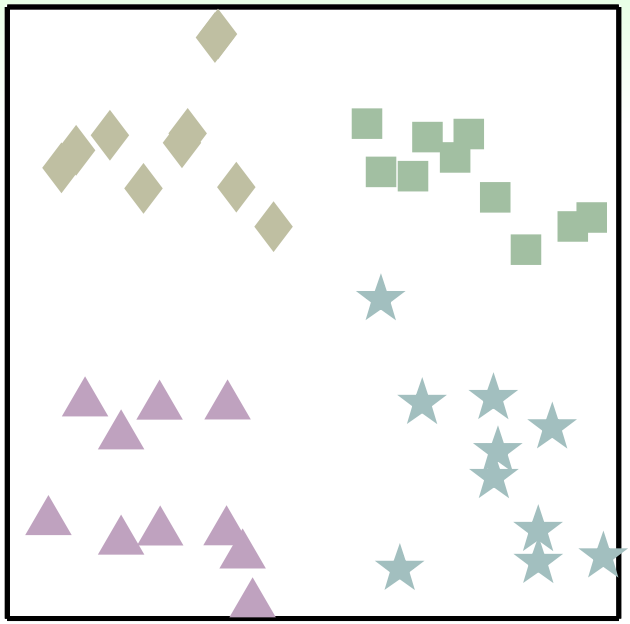
# One versus One at a Time

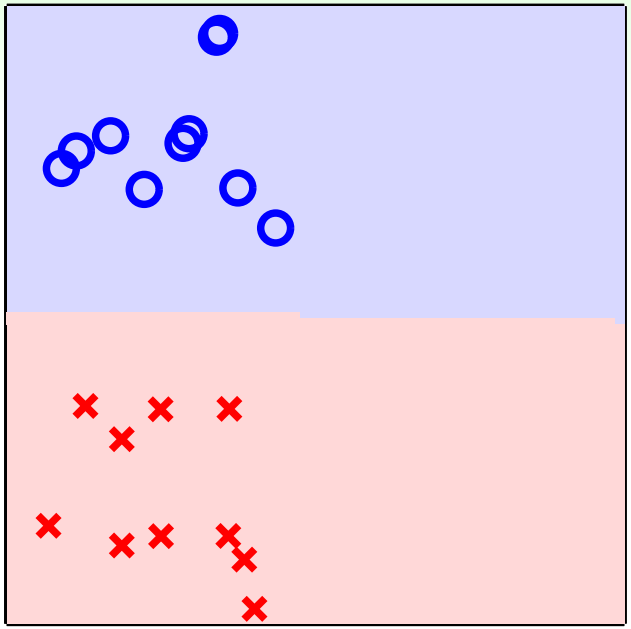




$\square$ or $\star$? $\{\square = \circ, \lozenge = \text{nil}, \triangle = \text{nil}, \star = \times\}$

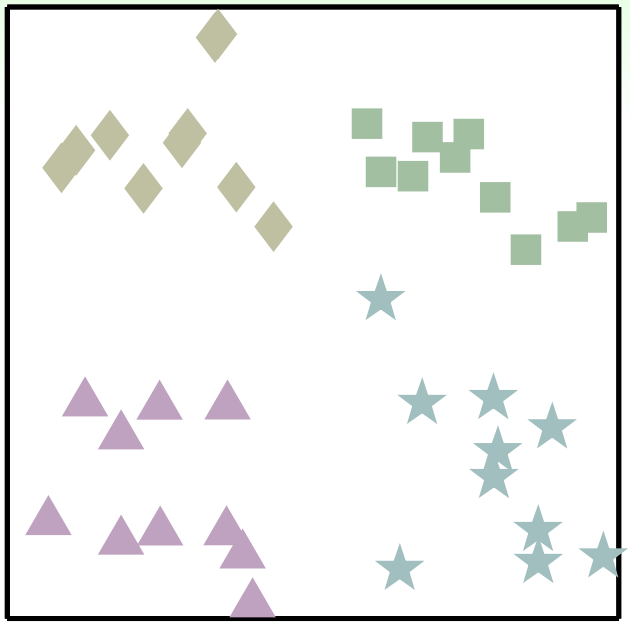
# One versus One at a Time

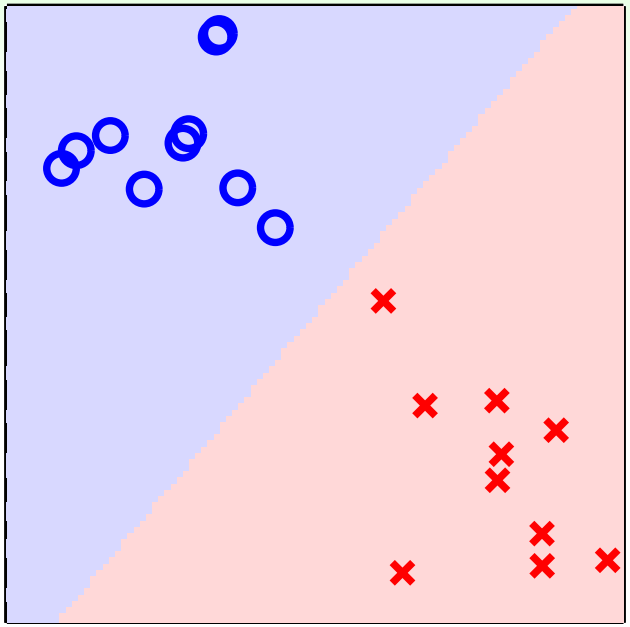




$\diamondsuit$ or $\triangle$? $\{\square = \text{nil}, \diamondsuit = \circ, \triangle = \times, \star = \text{nil}\}$

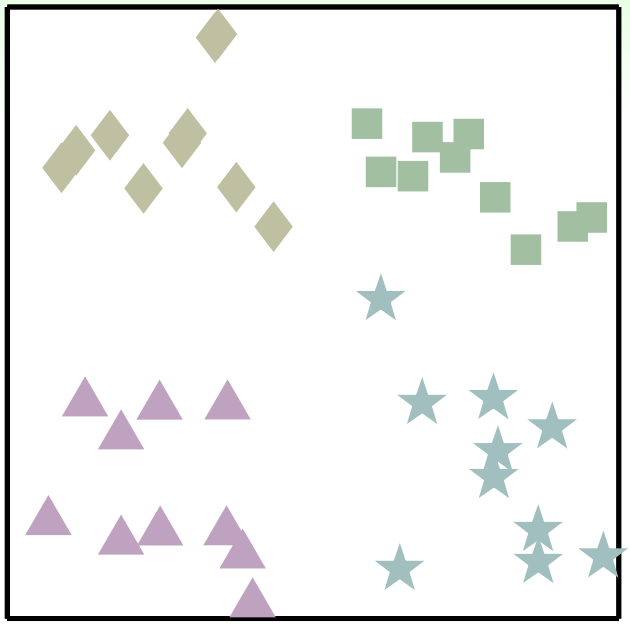
# One versus One at a Time

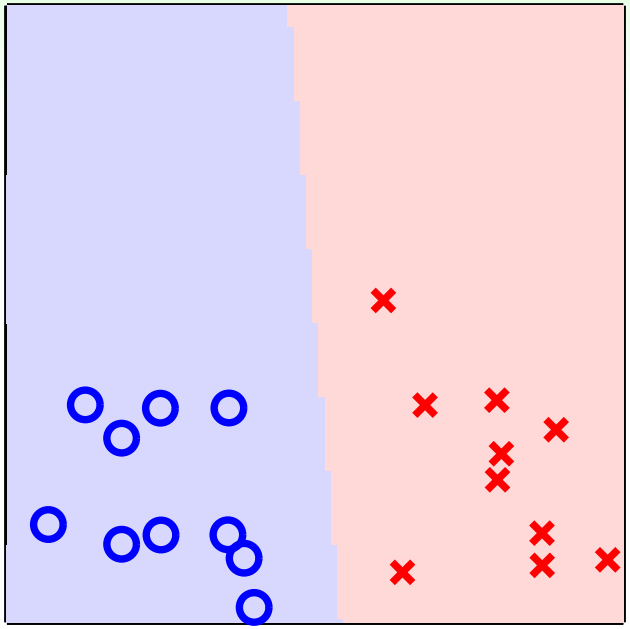




$\diamond$ or $\star$? $\{\square = \text{nil}, \diamond = \circ, \triangle = \text{nil}, \star = \times\}$

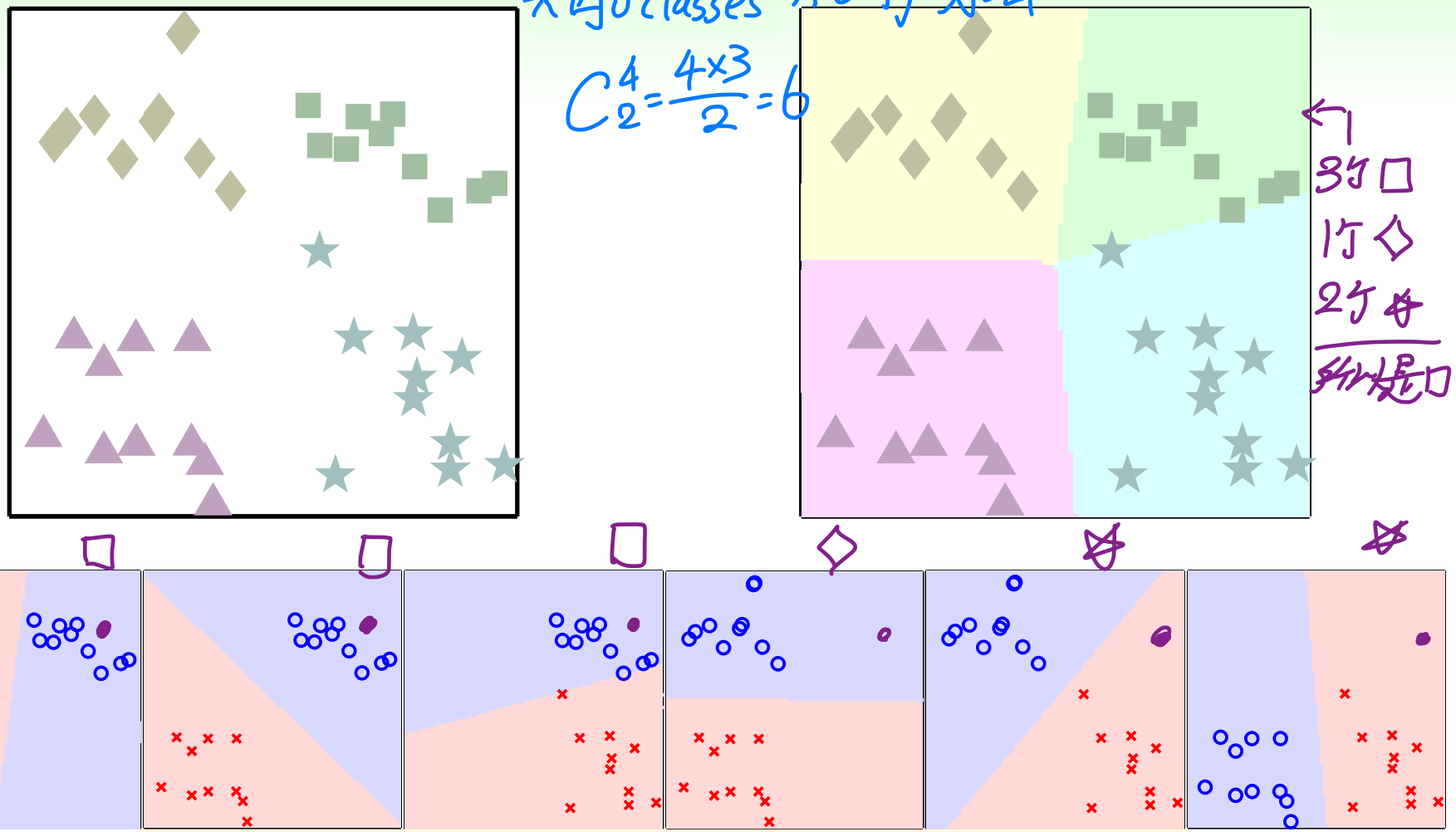
# One versus One at a Time





$\triangle$ or $\star$? $\{\square = \text{nil}, \lozenge = \text{nil}, \triangle = \circ, \star = \times\}$

# Multiclass Prediction: Combine **Pairwise** Classifiers




$$g(\mathbf{x}) = \underline{\text{tournament champion}}\left\{\mathbf{w}_{[k,\ell]}^T\mathbf{x}\right\}$$

**(voting of classifiers)**

# One-versus-one (OVO) Decomposition

1. for $(k, \ell) \in \mathcal{Y} \times \mathcal{Y}$
   obtain $\mathbf{w}_{[k,\ell]}$ by running linear binary classification on
   $$\mathcal{D}_{[k,\ell]} = \{(\mathbf{x}_n, y'_n = 2[\![y_n = k]\!] - 1) : y_n = k \text{ or } y_n = \ell\}$$
   2个classes
2. return $g(\mathbf{x}) = \text{tournament champion}\left\{\mathbf{w}_{[k,\ell]}^T \mathbf{x}\right\}$
   投票最高者

- pros: efficient ('smaller' training problems), stable, can be coupled with any binary classification approaches
- cons: use $O(K^2)$ $\mathbf{w}_{[k,\ell]}$ —**more space, slower prediction, more training**
  雖然要train多个classifier,但每个classifier training data很少

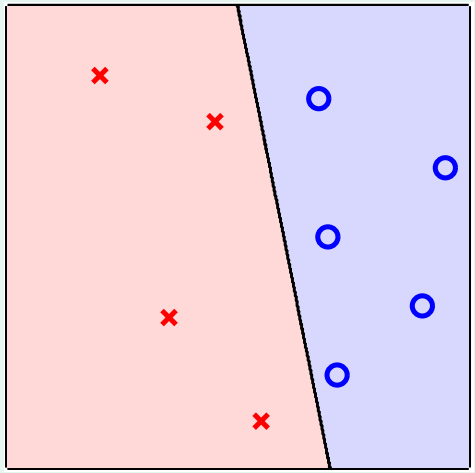
常用在K不大的情形下.

OVO: another simple multiclass **meta-algorithm** to keep in your toolbox

# Questions?

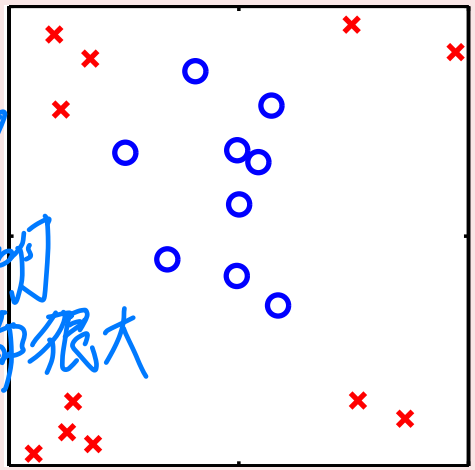
# Linear Hypotheses

## up to now: linear hypotheses



- visually: **'line'-like** boundary
- mathematically: linear scores $s = \mathbf{w}^T\mathbf{x}$

## but limited . . .

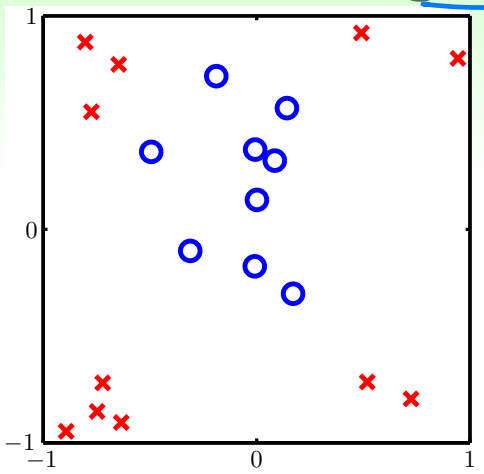



- theoretically: $d_{VC}$ **under control :-)**
- practically: on some $\mathcal{D}$, **large $E_{in}$ for every line :-(**

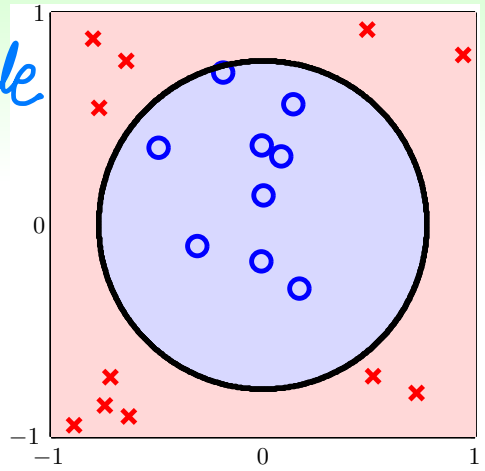
好處是VC dimension很小, Ein, Eout不會差太多

how to **break the limit** of linear hypotheses

# Circular Separable




不是 linear separable




- $\mathcal{D}$ not linear separable 同圈可分
- but **circular separable** by a circle of radius $\sqrt{0.6}$ centered at origin: circle formular

$$h_{\text{SEP}}(\mathbf{x}) = \text{sign}\left(-x_1^2 - x_2^2 + 0.6\right)$$

到圆心的距离

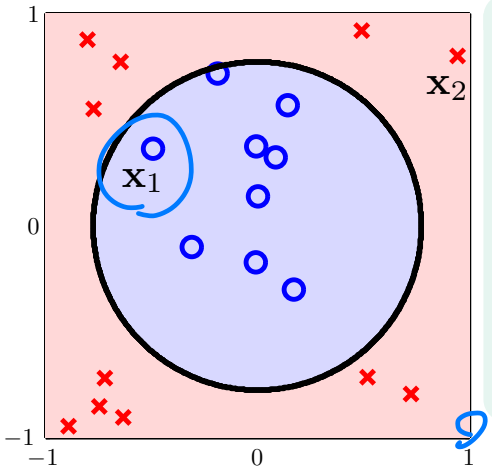
re-derive **Circular**-PLA, **Circular**-Regression, blahblah . . . all over again? **:-)**

# Circular Separable and Linear Separable

$$h(\mathbf{x}) = \text{sign}\left(\underbrace{0.6}_{\tilde{w}_0}\cdot\underbrace{1}_{z_0} + \underbrace{(-1)}_{\tilde{w}_1}\cdot\underbrace{x_1^2}_{z_1} + \underbrace{(-1)}_{\tilde{w}_2}\cdot\underbrace{x_2^2}_{z_2}\right) = \text{sign}\left(\tilde{\mathbf{w}}^T\mathbf{z}\right)$$

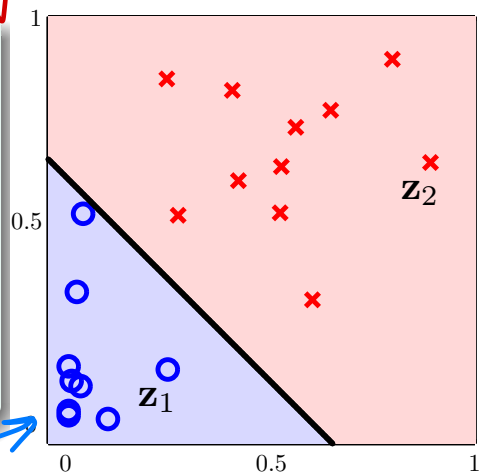
半徑²  data.

圓圈可分 轉到z-space 變線性可分




- $\{(\mathbf{x}_n, y_n)\}$ circular separable $\Longrightarrow$ $\{(\mathbf{z}_n, y_n)\}$ linear separable
- $\mathbf{x} \in \mathcal{X} \xrightarrow{\Phi} \mathbf{z} \in \mathcal{Z}$: **(nonlinear) feature transform** $\Phi$

data $\mathcal{X} \xrightarrow{\Phi} \mathcal{Z}$




轉換到z-space.

circular separable in $\mathcal{X}$ $\Longrightarrow$ linear separable in $\mathcal{Z}$

**vice versa?**

# Linear Hypotheses in $\mathcal{Z}$-Space

$$(z_0, z_1, z_2) = \mathbf{z} = \Phi(\mathbf{x}) = (1, x_1^2, x_2^2)$$

$$h(\mathbf{x}) = \tilde{h}(\mathbf{z}) = \text{sign}\left(\tilde{\mathbf{w}}^T \Phi(\mathbf{x})\right) = \text{sign}\left(\tilde{w}_0 + \tilde{w}_1 x_1^2 + \tilde{w}_2 x_2^2\right)$$

## $\tilde{\mathbf{w}} = (\tilde{w}_0, \tilde{w}_1, \tilde{w}_2)$

- $(0.6, -1, -1)$: circle ($\circ$ inside)
- $(-0.6, +1, +1)$: circle ($\circ$ outside)
- $(0.6, -1, -2)$: ellipse. $\text{sign}(0.6 - x_1^2 - 2x_2^2)$
- $(0.6, -1, +2)$: hyperbola
- $(0.6, +1, +2)$: **constant** $\circ$ :-) $\text{sign}(0.6 + x_1^2 + x_2^2)$ ← 永遠是正的

lines in $\mathcal{Z}$-space
$\Longleftrightarrow$ **special** quadratic curves in $\mathcal{X}$-space

但圓心一定要過原点!?

# General Quadratic Hypothesis Set

a 'bigger' $\mathcal{Z}$-space with $\Phi_2(\mathbf{x}) = (1, x_1, x_2, x_1^2, x_1x_2, x_2^2)$

perceptrons in $\mathcal{Z}$-space $\Longleftrightarrow$ quadratic hypotheses in $\mathcal{X}$-space

$$\mathcal{H}_{\Phi_2} = \left\{ h(\mathbf{x}) : h(\mathbf{x}) = \tilde{h}(\Phi_2(\mathbf{x})) \text{ for some linear } \tilde{h} \text{ on } \mathcal{Z} \right\}$$

- can implement **all possible quadratic curve boundaries**: circle, ellipse, **rotated** ellipse, hyperbola, parabola, ...

ellipse $2(x_1 + x_2 - 3)^2 + (x_1 - x_2 - 4)^2 = 1$

$\Longleftarrow \tilde{\mathbf{w}}^T = [33, -20, -4, 3, 2, 3]$

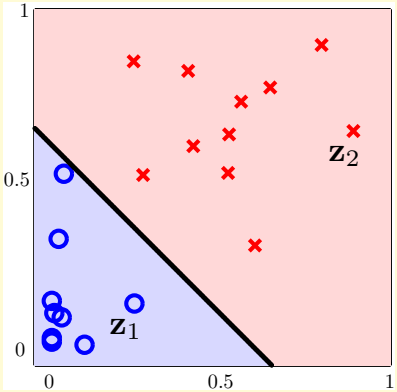
- include **lines and constants as degenerate cases**

next: **learn** a good quadratic hypothesis $g$

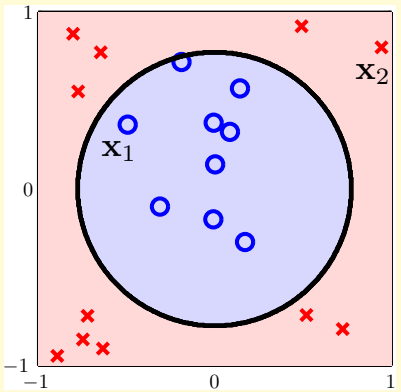
# Questions?

# Good Quadratic Hypothesis

| $\mathcal{Z}$-space | | $\mathcal{X}$-space |
|---|---|---|
| perceptrons | $\Longleftrightarrow$ | quadratic hypotheses |
| **good perceptron** | $\Longleftrightarrow$ | **good quadratic hypothesis** |
| **separating perceptron** | $\Longleftrightarrow$ | **separating quadratic hypothesis** |



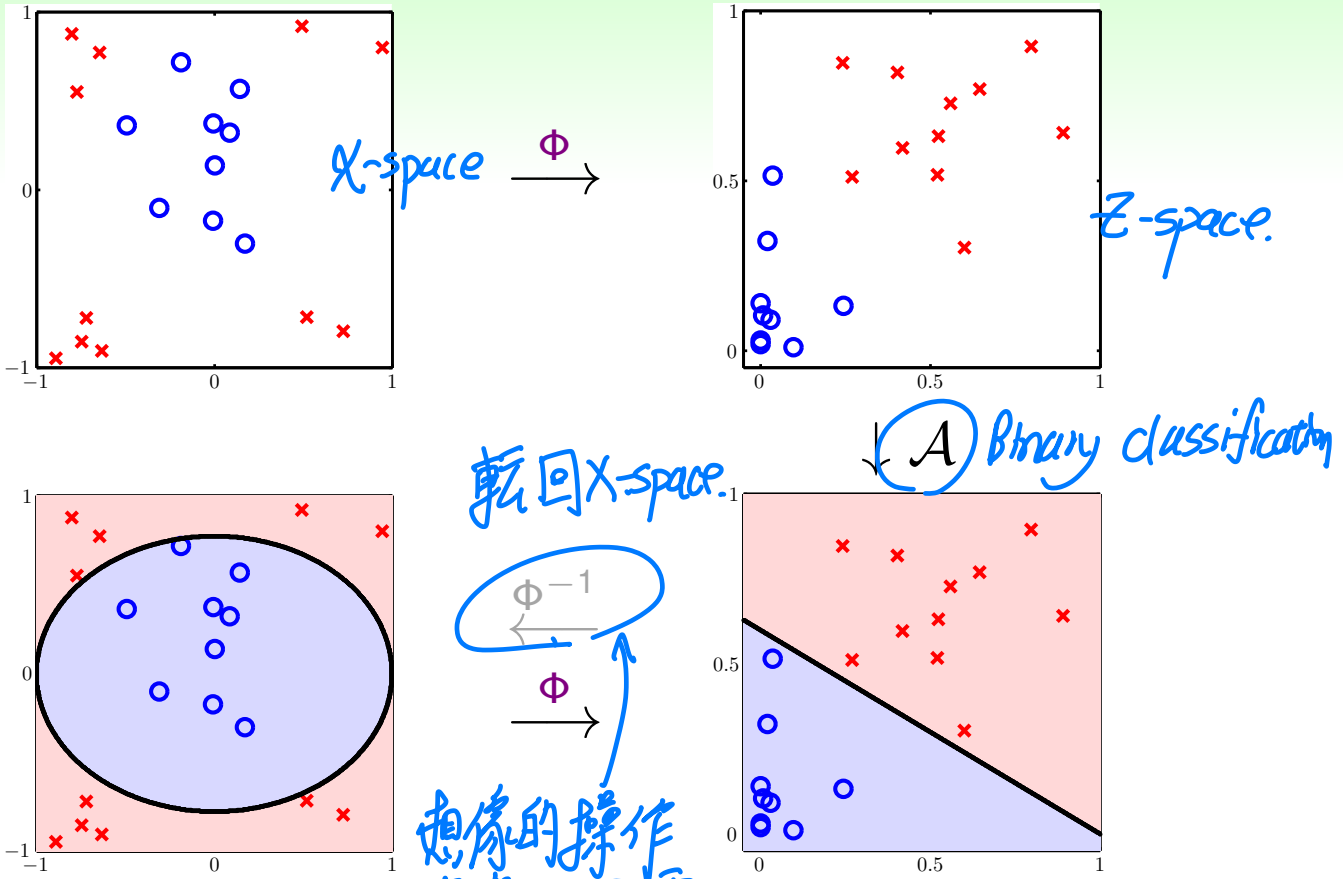

$\Longleftrightarrow$




- want: get **good perceptron** in $\mathcal{Z}$-space ← 在Z space找一條好的 line.
- known: get **good perceptron** in $\mathcal{X}$-space with data $\{(\mathbf{x}_n, y_n)\}$

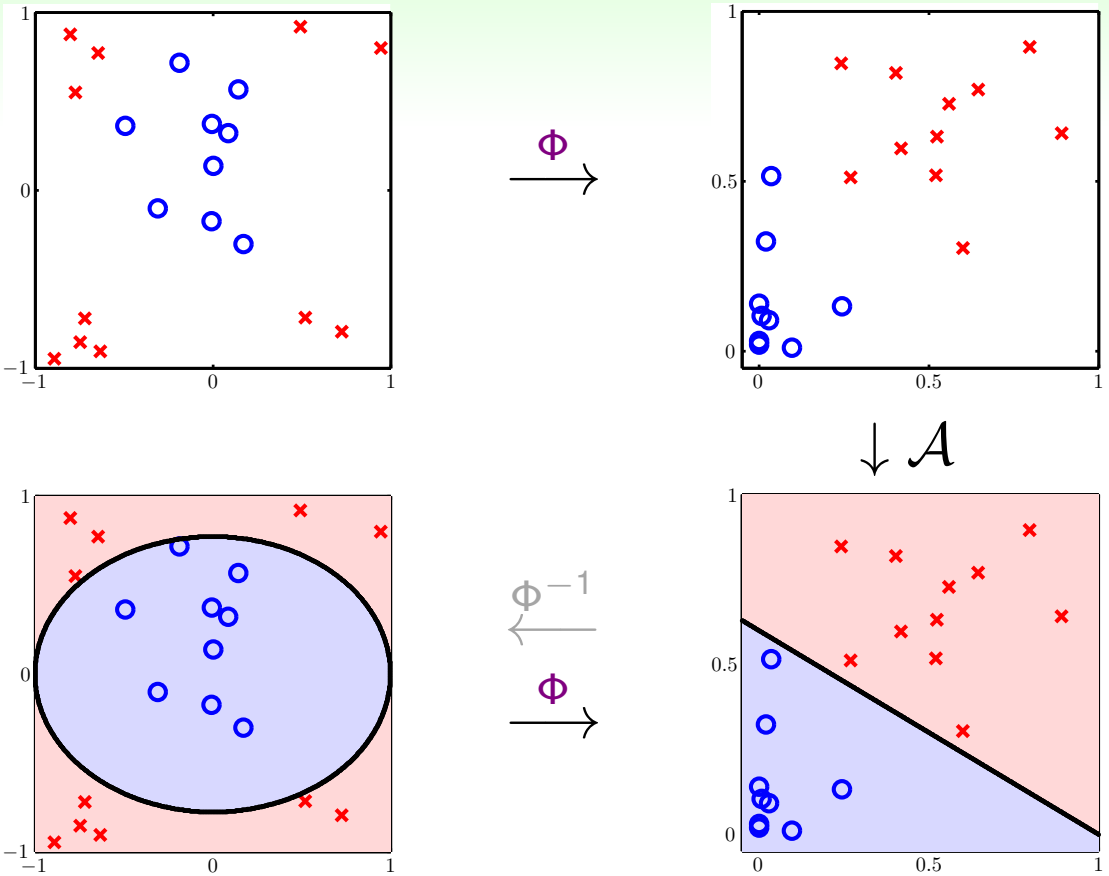
todo: get **good perceptron** in $\mathcal{Z}$-space with data $\{(\mathbf{z}_n = \Phi_2(\mathbf{x}_n), y_n)\}$

# The Nonlinear Transform Steps




1. transform original data $\{(\mathbf{x}_n, y_n)\}$ to $\{(\mathbf{z}_n = \Phi(\mathbf{x}_n), y_n)\}$ by $\Phi$
2. get a good perceptron $\tilde{\mathbf{w}}$ using $\{(\mathbf{z}_n, y_n)\}$ and your favorite linear classification algorithm $\mathcal{A}$
3. return $g(\mathbf{x}) = \text{sign}\left(\tilde{\mathbf{w}}^T \Phi(\mathbf{x})\right)$

# Nonlinear Model via Nonlinear $\Phi$ + Linear Models




two choices:

- feature transform $\Phi$ How to transform.
- linear model $\mathcal{A}$, **not just binary classification**

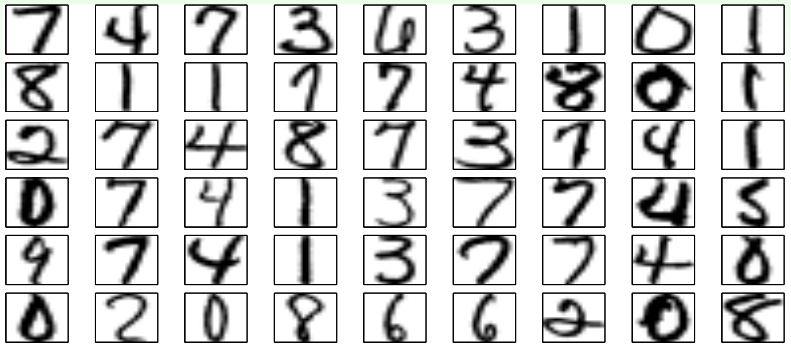
**Pandora's box :-)**: 全部都可以做了

can now freely do **quadratic PLA, quadratic regression, cubic regression, . . ., polynomial regression**

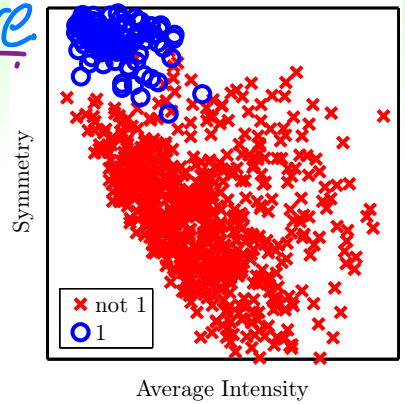
# Feature Transform $\Phi$

it's kind of concert feature

concert and raw feature



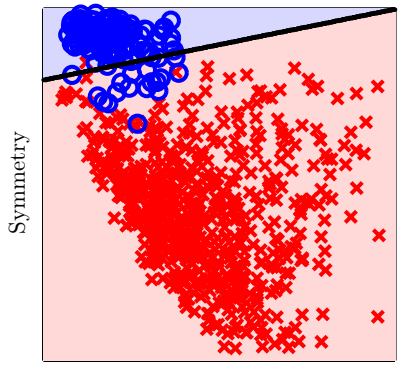
$\xrightarrow{\Phi}$




$\downarrow \mathcal{A}$

$\xleftarrow{\Phi^{-1}}$

$\xrightarrow{\Phi}$




not new, not just polynomial:

raw (pixels) $\xrightarrow{\text{domain knowledge}}$ **concrete (intensity, symmetry)**

原力

the force, too good to be true? :-)

# Questions?

使用 feature transform 的代價

# Computation/Storage Price

$Q$-th order polynomial transform: $\Phi_Q(\mathbf{x}) = \Big($ 1,

$x_1, x_2, \ldots, x_d,$ ← 一次

$x_1^2, x_1x_2, \ldots, x_d^2,$ ← 2次

$\ldots,$

Q次 $x_1^Q, x_1^{Q-1}x_2, \ldots, x_d^Q\Big)$

常數項

$\underbrace{1}_{\tilde{w}_0} + \underbrace{\tilde{d}}_{\text{others}}$ dimensions

d種不同的項乘Q個出來

= # ways of $\leq Q$-combination from $d$ kinds with repetitions

有幾種取法

$= \binom{Q+d}{Q} = \binom{Q+d}{d} = O(Q^d)$

計算幾多 參數要多

= efforts needed for computing/storing $\mathbf{z} = \Phi_Q(\mathbf{x})$ and $\tilde{\mathbf{w}}$

$Q$ large $\Longrightarrow$ **difficult to compute/store**

# Model Complexity Price

$Q$-th order polynomial transform: $\Phi_Q(\mathbf{x}) = \Big($ $1,$ $x_1, x_2, \ldots, x_d,$ $x_1^2, x_1x_2, \ldots, x_d^2,$ $\ldots,$ $x_1^Q, x_1^{Q-1}x_2, \ldots, x_d^Q\Big)$

這是項次的數量(# of term)

$\underbrace{1}_{\tilde{w}_0} + \underbrace{\tilde{d}}_{\text{others}}$ dimensions $= O(Q^d)$

自由度變大了

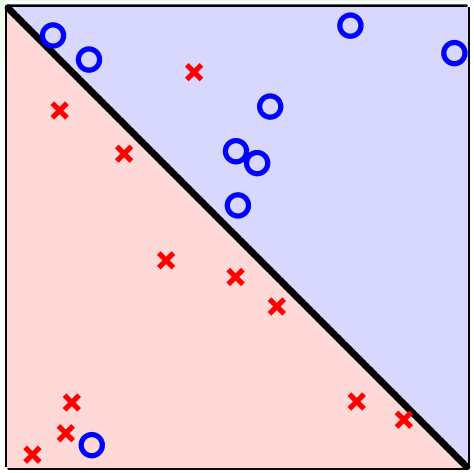
- number of free parameters $\tilde{w}_i = \tilde{d} + 1 \approx d_{\text{VC}}(\mathcal{H}_{\Phi_Q})$
- $d_{\text{VC}}(\mathcal{H}_{\Phi_Q}) \leq \tilde{d} + 1$, why? upper bound.

  any $\tilde{d} + 2$ inputs not shattered in $\mathcal{Z}$

  $\Longrightarrow$ any $\tilde{d} + 2$ inputs not shattered in $\mathcal{X}$

$Q$ large $\Longrightarrow$ **large** $d_{\text{VC}}$

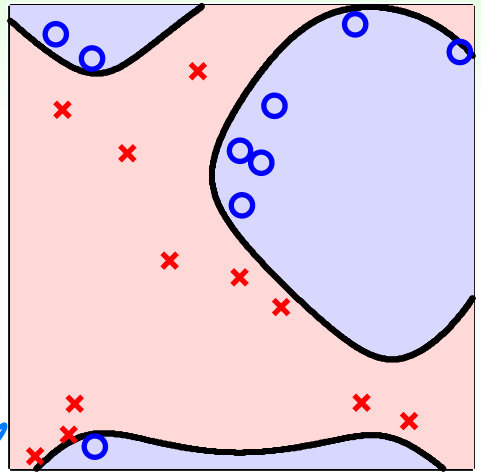
# Generalization Issue



$\Phi_1$ (original $\mathbf{x}$)

**which one do you prefer? :-)**

- $\Phi_1$ 'visually' preferred
- $\Phi_4$: $E_{\text{in}}(g) = 0$ but overkill



overfitting?

$\Phi_4$ Q=4

1. can we make sure that $E_{\text{out}}(g)$ is close enough to $E_{\text{in}}(g)$?
2. can we make $E_{\text{in}}(g)$ small enough?

trade-off:

| $\tilde{d}$ ($Q$) | 1 | 2 |
|---|---|---|
| higher | :-( | **:-D** |
| lower | **:-D** | :-( |

how to pick $Q$? **visually**, maybe?

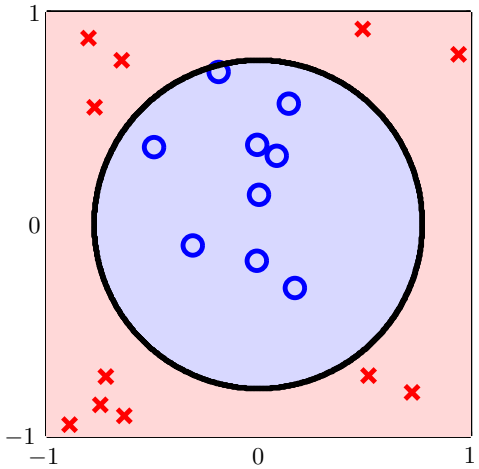
# Danger of Visual Choices

first of all, can you really 'visualize' when $\mathcal{X} = \mathbb{R}^{10}$? **(well, I can't :-))**

## Visualize $\mathcal{X} = \mathbb{R}^2$

- full $\Phi_2$: $\mathbf{z} = (1, x_1, x_2, x_1^2, x_1x_2, x_2^2)$, $d_{\text{VC}} = 6$
- or $\mathbf{z} = (1, x_1^2, x_2^2)$, $d_{\text{VC}} = 3$, **after visualizing**?
- or better $\mathbf{z} = (1, x_1^2 + x_2^2)$ , $d_{\text{VC}} = 2$?
- or even better $\mathbf{z} = (\text{sign}(0.6 - x_1^2 - x_2^2))$?

—careful about **your brain's 'model complexity'**




這是 human learning

做出來 $E_{in}$ 低, $d_{VC}$ 又低, 但真的好嗎?

$d_{VC}=1$?

for VC-safety, $\Phi$ shall be decided **without 'peeking'** data

人類偷看

# Questions?

# Polynomial Transform Revisited

recussive

$$\Phi_0(\mathbf{x}) = \left(1\right), \quad \Phi_1(\mathbf{x}) = \left(\Phi_0(\mathbf{x}), \quad x_1, x_2, \ldots, x_d\right)$$

$$\Phi_2(\mathbf{x}) = \left(\Phi_1(\mathbf{x}), \quad x_1^2, x_1x_2, \ldots, x_d^2\right)$$

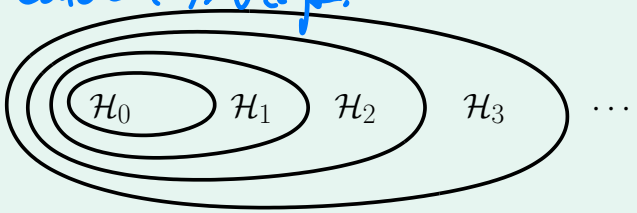
$$\Phi_3(\mathbf{x}) = \left(\Phi_2(\mathbf{x}), \quad x_1^3, x_1^2x_2, \ldots, x_d^3\right)$$

$$\ldots \qquad \ldots$$

$$\Phi_Q(\mathbf{x}) = \left(\Phi_{Q-1}(\mathbf{x}), \quad x_1^Q, x_1^{Q-1}x_2, \ldots, x_d^Q\right)$$

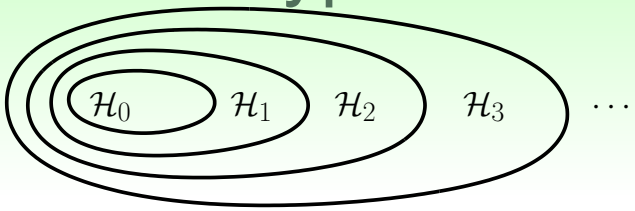
$$\begin{array}{ccccccccccc}
\mathcal{H}_{\Phi_0} & \subset & \mathcal{H}_{\Phi_1} & \subset & \mathcal{H}_{\Phi_2} & \subset & \mathcal{H}_{\Phi_3} & \subset & \ldots & \subset & \mathcal{H}_{\Phi_Q} \\
\| & & \| & & \| & & \| & & & & \| \\
\mathcal{H}_0 & & \mathcal{H}_1 & & \mathcal{H}_2 & & \mathcal{H}_3 & & \ldots & & \mathcal{H}_Q
\end{array}$$

高维必定 Included 低维.



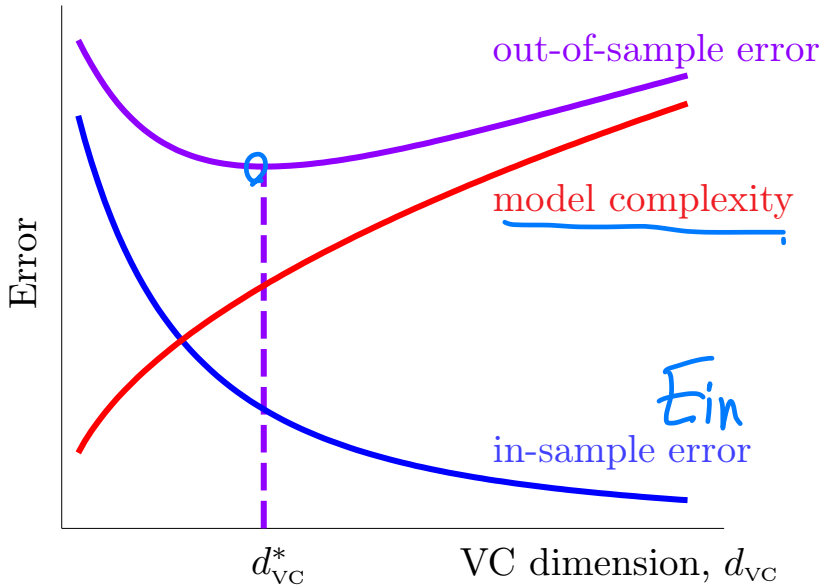

structure: **nested** $\mathcal{H}_i$

# Structured Hypothesis Sets



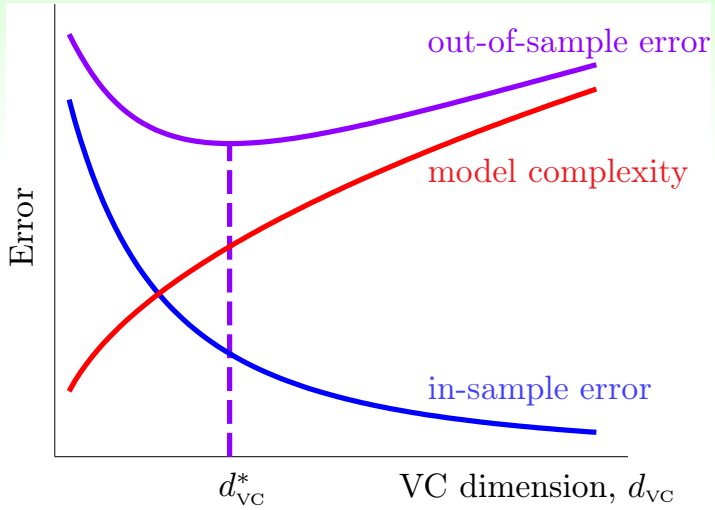

Let $g_i = \text{argmin}_{h \in \mathcal{H}_i} E_{\text{in}}(h)$:

$$\begin{array}{ccccccccc} \mathcal{H}_0 & \subset & \mathcal{H}_1 & \subset & \mathcal{H}_2 & \subset & \mathcal{H}_3 & \subset & \cdots \\ d_{\text{VC}}(\mathcal{H}_0) & \leq & d_{\text{VC}}(\mathcal{H}_1) & \leq & d_{\text{VC}}(\mathcal{H}_2) & \leq & d_{\text{VC}}(\mathcal{H}_3) & \leq & \cdots \\ E_{\text{in}}(g_0) & \geq & E_{\text{in}}(g_1) & \geq & E_{\text{in}}(g_2) & \geq & E_{\text{in}}(g_3) & \geq & \cdots \end{array}$$




use $\mathcal{H}_{1126}$ won't be good! :-(

# Linear Model First




- tempting sin: use $\mathcal{H}_{1126}$, low $E_{\text{in}}(g_{1126})$ to fool your boss
  —**really? :-( a dangerous path of no return**
- safe route: $\mathcal{H}_1$ first
  - if $E_{\text{in}}(g_1)$ good enough, **live happily thereafter :-)**
  - otherwise, move right of the curve
    **with nothing lost except 'wasted' computation**

linear model first:
simple, efficient, **safe**, **and workable!**

# Questions?

# Summary

1. **How** Can Machines Learn?

## Lecture 05: Linear Models

## Lecture 06: Beyond Basic Linear Models

- Multiclass via Logistic Regression
  **predict with maximum estimated $P(k|\mathbf{x})$**
- Multiclass via Binary Classification
  **predict the tournament champion**
- Quadratic Hypotheses

**linear hypotheses on quadratic-transformed data**

- Nonlinear Transform
  **happy linear modeling after $\mathcal{Z} = \Phi(\mathcal{X})$**
- Price of Nonlinear Transform
  **computation/storage/[model complexity]**
- Structured Hypothesis Sets
  **linear/simpler model first**

- **next: dark side of the force :-)**