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)

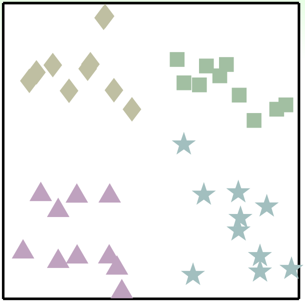
Roadmap

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

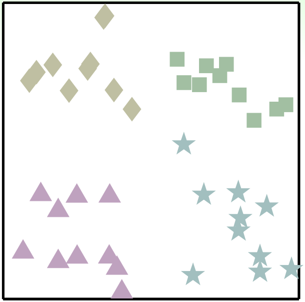
Multiclass Classification

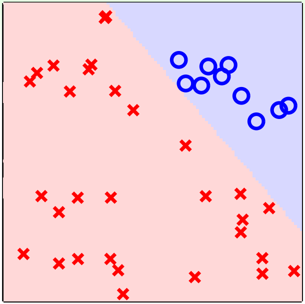


- $\mathcal{Y} = \{\square, \Diamond, \triangle, \star\}$ (4-class classification)
- **many applications** in practice, especially for 'recognition'

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

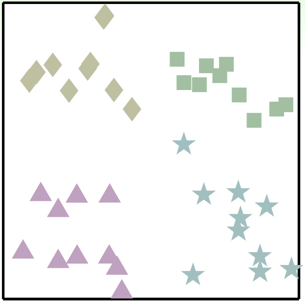
One Class at a Time

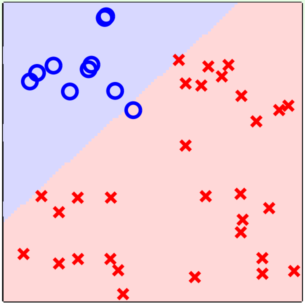




$\square$ or not? $\{\square = \circ, \lozenge = \times, \triangle = \times, \star = \times\}$

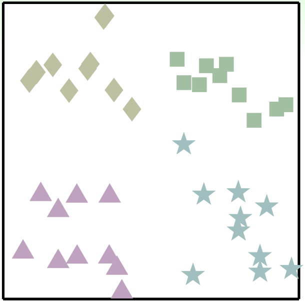
One Class at a Time

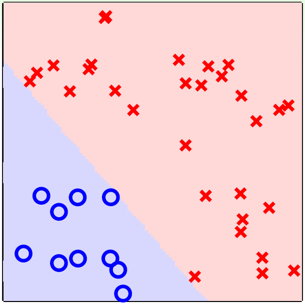




$\Diamond$ or not? $\{\Box = \times, \Diamond = \circ, \triangle = \times, \star = \times\}$

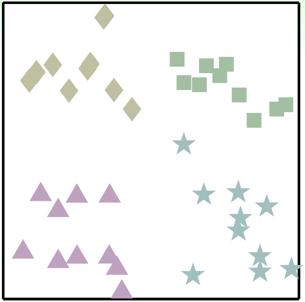
One Class at a Time

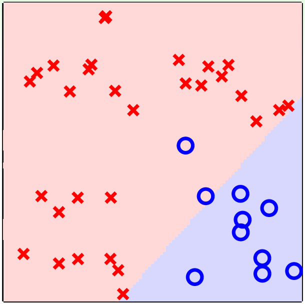




$\triangle$ or not? $\{\square = \times, \diamond = \times, \triangle = \circ, \star = \times\}$

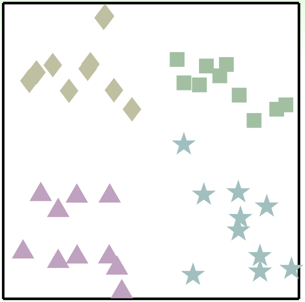
One Class at a Time

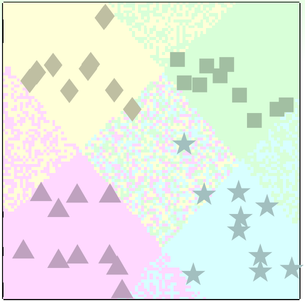


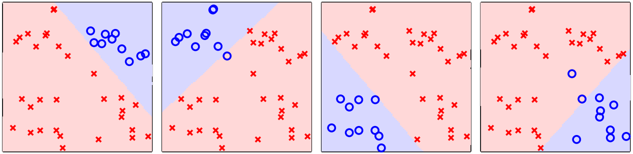


$\star$ or not? $\{\square = \times, \lozenge = \times, \triangle = \times, \star = \circ\}$

Multiclass Prediction: Combine Binary Classifiers

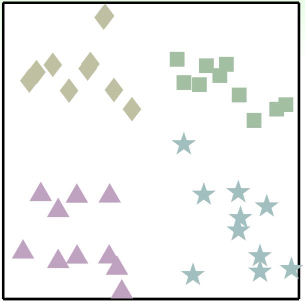


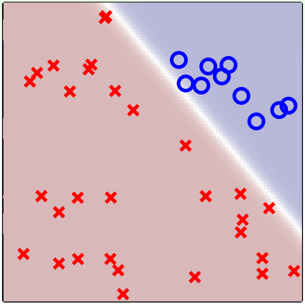


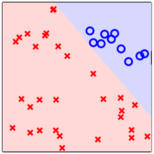


but **ties? :-)**

One Class at a Time **Softly**

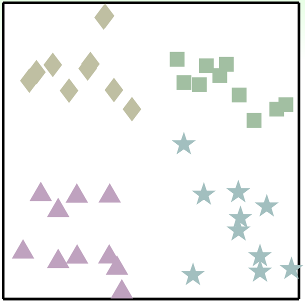


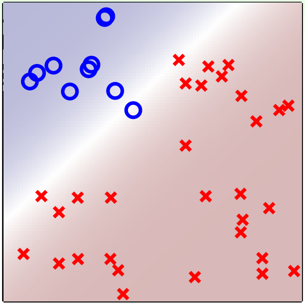


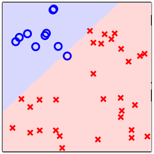


$$P(\square|\mathbf{x})?\ \{\square = \circ, \lozenge = \times, \triangle = \times, \star = \times\}$$

One Class at a Time **Softly**

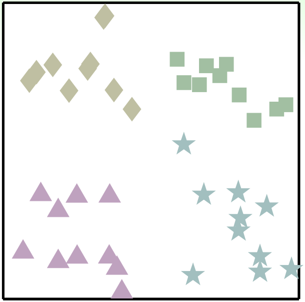


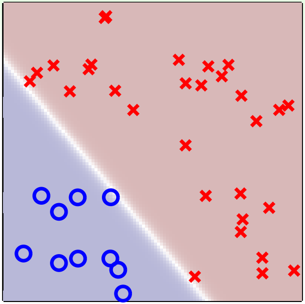


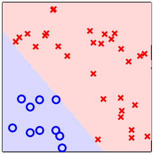


$$P(\Diamond|\mathbf{x})?\ \{\square = \times, \Diamond = \circ, \triangle = \times, \star = \times\}$$

One Class at a Time **Softly**

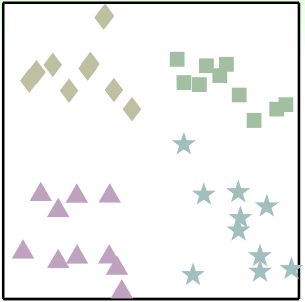


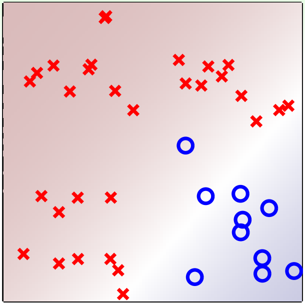


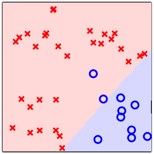


$$P(\triangle|\mathbf{x})?\ \{\square = \times, \Diamond = \times, \triangle = \circ, \star = \times\}$$

One Class at a Time Softly

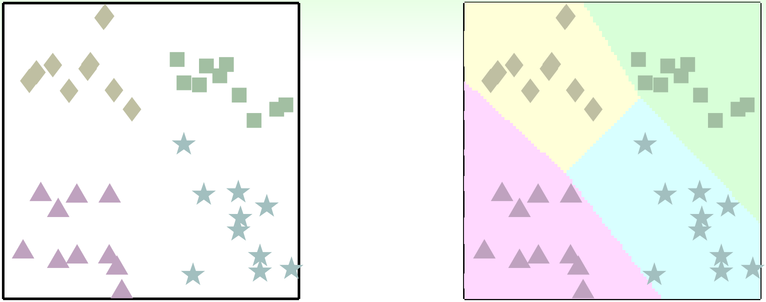


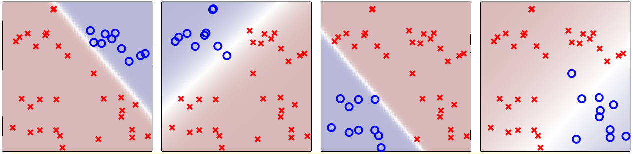




$$P(\star|\mathbf{x})?\ \{\square = \times, \Diamond = \times, \triangle = \times, \star = \circ\}$$

Multiclass Prediction: Combine **Soft** Classifiers





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

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

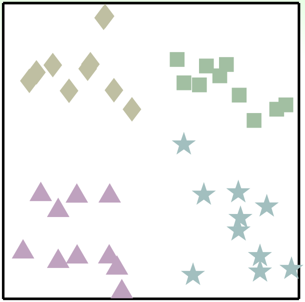
- pros: efficient,

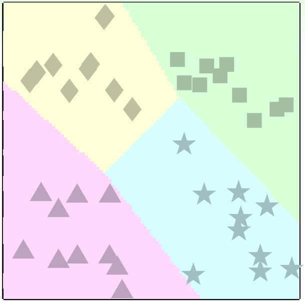
  can be coupled with any logistic regression-like approaches
- cons: often **unbalanced** $\mathcal{D}_{[k]}$ when K large
- extension: **multinomial ('coupled') logistic regression**

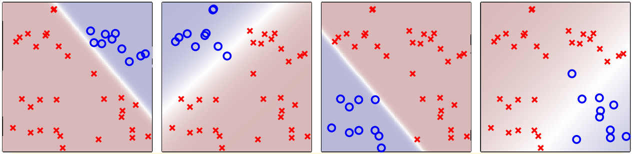
OVA: a simple multiclass **meta-algorithm**
to keep in your toolbox

Questions?

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

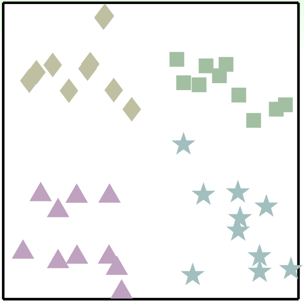


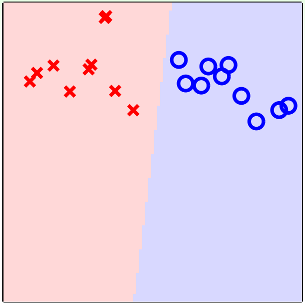




idea: make binary classification problems
more **balanced** by one versus **one**

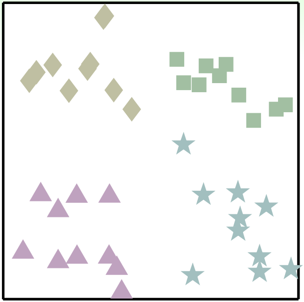
One versus One at a Time

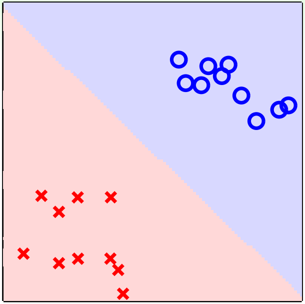




$\square$ or $\Diamond$? $\{\square = \circ, \Diamond = \times, \triangle = \text{nil}, \star = \text{nil}\}$

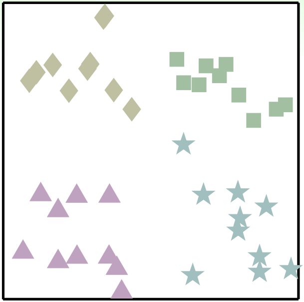
One versus One at a Time

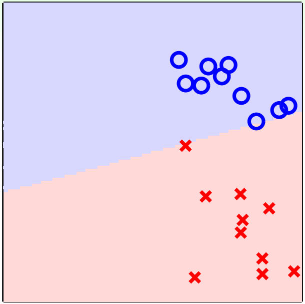




$\square$ or $\triangle$? $\{\square = \circ, \Diamond = \text{nil}, \triangle = \times, \star = \text{nil}\}$

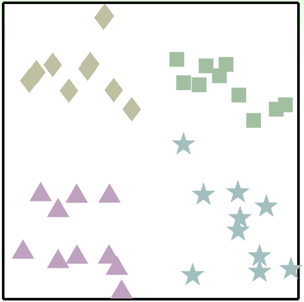
One versus One at a Time

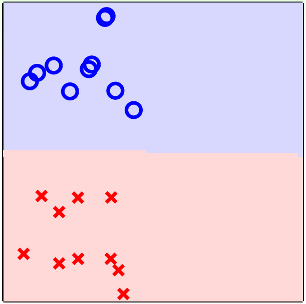




$\square$ or $\star$? $\{\square = \circ, \Diamond = \text{nil}, \triangle = \text{nil}, \star = \times\}$

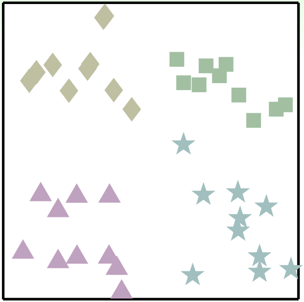
One versus One at a Time

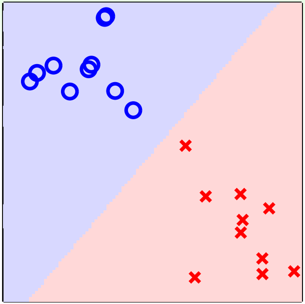




$\Diamond$ or $\triangle$? $\{\Box = \text{nil}, \Diamond = \circ, \triangle = \times, \star = \text{nil}\}$

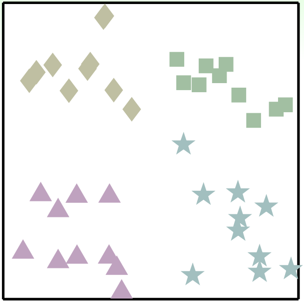
One versus One at a Time

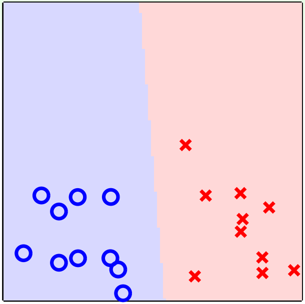




$\diamondsuit$ or $\star$? $\{\square = \text{nil}, \diamondsuit = \circ, \triangle = \text{nil}, \star = \times\}$

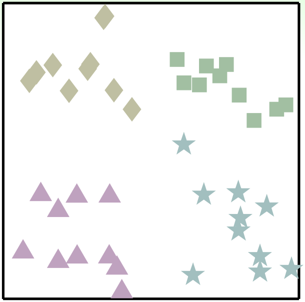
One versus One at a Time

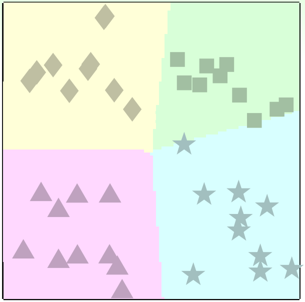


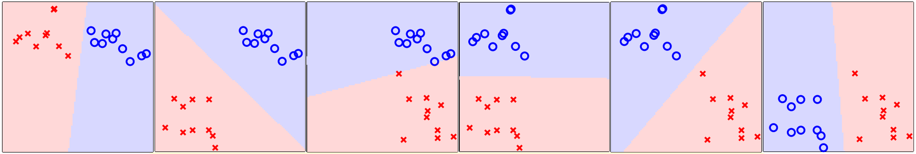


$\triangle$ or $\star$? $\{\square = \text{nil}, \Diamond = \text{nil}, \triangle = \circ, \star = \times\}$

Multiclass Prediction: Combine **Pairwise** Classifiers







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

(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

   $$\mathcal{D}_{[k,\ell]} = \{(\mathbf{x}_n, y_n' = 2\,[\![y_n = k]\!] - 1) \colon y_n = k \text{ or } y_n = \ell\}$$

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

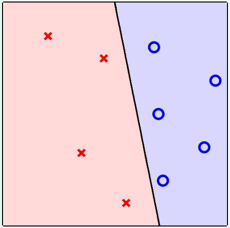
- pros: efficient ('smaller' training problems), stable,
  can be coupled with any binary classification approaches
- cons: use $O(K^2)$ $\mathbf{w}_{[k,\ell]}$
  —**more space, slower prediction, more training**

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

Questions?

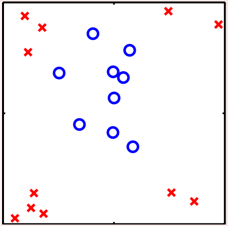
Linear Hypotheses

up to now: linear hypotheses



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

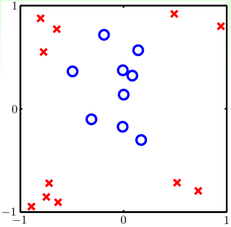
but limited . . .

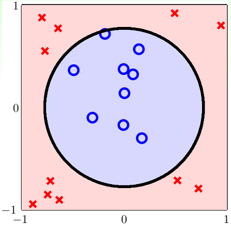


- theoretically: d_{VC} **under control :-)**
- practically: on some $\mathcal{D}$, **large** E_{in} for every line **:-(**

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

Circular Separable





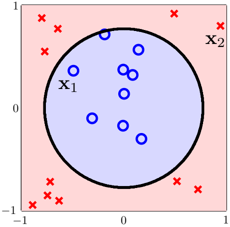
- $\mathcal{D}$ not linear separable
- but **circular separable** by a circle of radius $\sqrt{0.6}$ centered at origin:

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

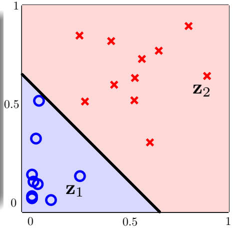
re-derive **Circular**-PLA, **Circular**-Regression, blahblah . . . all over again? **:-)**

Circular Separable and Linear Separable

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



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



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

vice versa?

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

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

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

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

- $(0.6, -1, -1)$: circle ($\circ$ inside)
- $(-0.6, +1, +1)$: circle ($\circ$ outside)
- $(0.6, -1, -2)$: ellipse
- $(0.6, -1, +2)$: hyperbola
- $(0.6, +1, +2)$: **constant $\circ$:-)**

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

General Quadratic Hypothesis Set

a 'bigger' $\mathcal{Z}$-space with $\mathbf{\Phi}_2(\mathbf{x}) = (1, x_1, x_2, x_1^2, x_1 x_2, x_2^2)$

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

$$\mathcal{H}_{\mathbf{\Phi}_2} = \left\{ h(\mathbf{x}) \colon h(\mathbf{x}) = \tilde{h}(\mathbf{\Phi}_2(\mathbf{x})) \text{ for some linear } \tilde{h} \text{ on } \mathcal{Z} \right\}$$

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

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

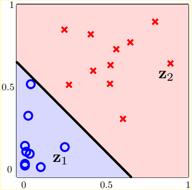
  $\Longleftarrow \tilde{\mathbf{w}}^T = [33, -20, -4, 3, 2, 3]$
- include **lines and constants as degenerate cases**

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

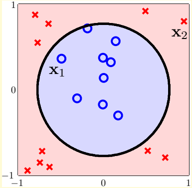
Questions?

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**



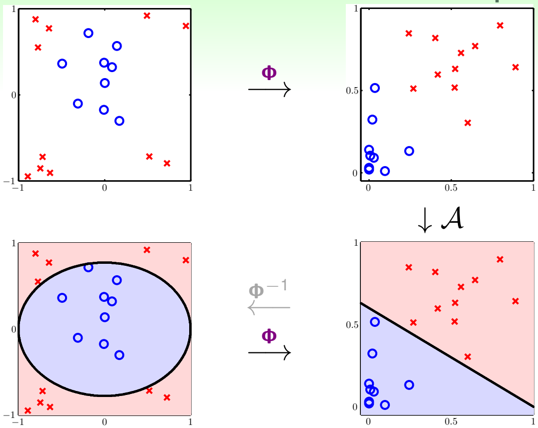
$\Longleftrightarrow$



- want: get **good perceptron** in $\mathcal{Z}$-space
- known: get **good perceptron** in $\mathcal{X}$-space with data $\{(\mathbf{x}_n, y_n)\}$

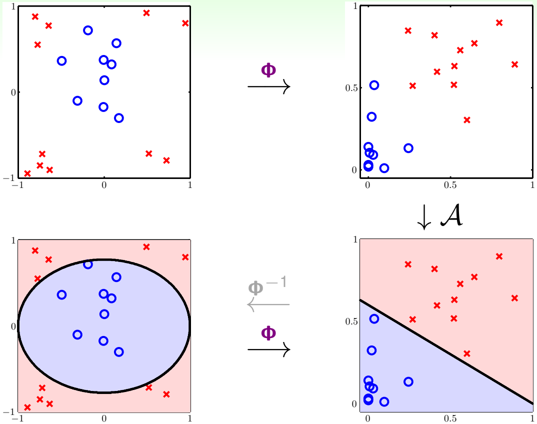
todo: get **good perceptron** in $\mathcal{Z}$-space with data $\{(\mathbf{z}_n = \mathbf{\Phi}_2(\mathbf{x}_n), y_n)\}$

The Nonlinear Transform Steps



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

Nonlinear Model via Nonlinear $\mathbf{\Phi}$ + Linear Models



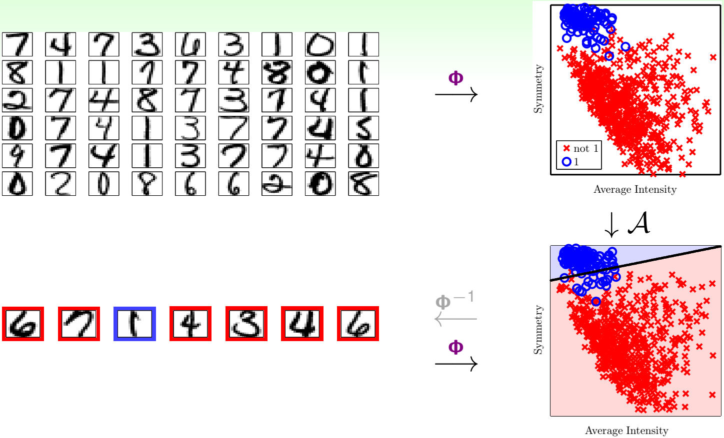
two choices:

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

Pandora's box :-):

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

Feature Transform Φ



not new, not just polynomial:

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

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

Questions?

Computation/Storage Price

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

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

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

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

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

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

Model Complexity Price

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

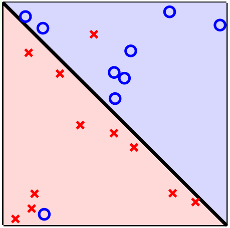
$\underbrace{1}_{\tilde{w}_0} + \underbrace{\tilde{d}}_{\text{others}}$ dimensions = $O(Q^d)$

- 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?

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_{VC}

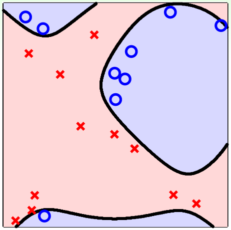
Generalization Issue



$\mathbf{\Phi}_1$ (original $\mathbf{x}$)

which one do you prefer? :-)

- $\mathbf{\Phi}_1$ 'visually' preferred
- $\mathbf{\Phi}_4$: $E_{\text{in}}(g) = 0$ but overkill



$\mathbf{\Phi}_4$

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?

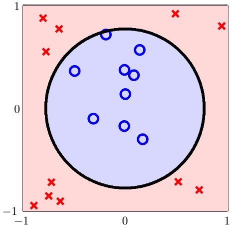
Danger of Visual Choices

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

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

- full $\mathbf{\Phi}_2$: $\mathbf{z} = (1, x_1, x_2, x_1^2, x_1 x_2, x_2^2)$, $d_{\text{VC}} = 6$
- 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 $\mathbf{z} = \left(\text{sign}(0.6 - x_1^2 - x_2^2)\right)$?

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



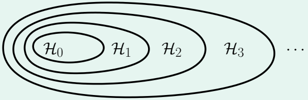
for VC-safety, $\mathbf{\Phi}$ shall be decided **without 'peeking'** data

Questions?

Polynomial Transform Revisited

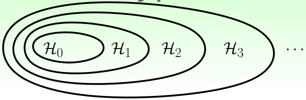
$$
\begin{aligned}
\mathbf{\Phi}_0(\mathbf{x}) = \left(1\right), \mathbf{\Phi}_1(\mathbf{x}) &= \left(\mathbf{\Phi}_0(\mathbf{x}), \quad x_1, x_2, \ldots, x_d\right) \\
\mathbf{\Phi}_2(\mathbf{x}) &= \left(\mathbf{\Phi}_1(\mathbf{x}), \quad x_1^2, x_1x_2, \ldots, x_d^2\right) \\
\mathbf{\Phi}_3(\mathbf{x}) &= \left(\mathbf{\Phi}_2(\mathbf{x}), \quad x_1^3, x_1^2x_2, \ldots, x_d^3\right) \\
&\ldots \quad \ldots \\
\mathbf{\Phi}_Q(\mathbf{x}) &= \left(\mathbf{\Phi}_{Q-1}(\mathbf{x}), \quad x_1^Q, x_1^{Q-1}x_2, \ldots, x_d^Q\right)
\end{aligned}
$$

$$
\begin{array}{ccccccccccc}
\mathcal{H}_{\mathbf{\Phi}_0} & \subset & \mathcal{H}_{\mathbf{\Phi}_1} & \subset & \mathcal{H}_{\mathbf{\Phi}_2} & \subset & \mathcal{H}_{\mathbf{\Phi}_3} & \subset & \ldots & \subset & \mathcal{H}_{\mathbf{\Phi}_Q} \\
\| & & \| & & \| & & \| & & & & \| \\
\mathcal{H}_0 & & \mathcal{H}_1 & & \mathcal{H}_2 & & \mathcal{H}_3 & & \ldots & & \mathcal{H}_Q
\end{array}
$$



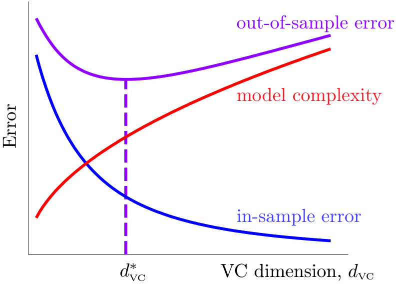
structure: **nested** $\mathcal{H}_i$

Structured Hypothesis Sets



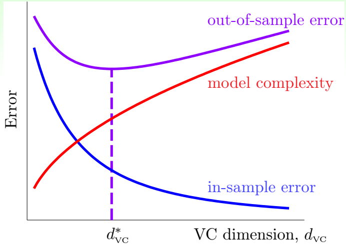
Let $g_i = \text{argmin}_{h \in \mathcal{H}_i} E_{\text{in}}(h)$:

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



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

Linear Model First



- tempting sin: use $\mathcal{H}_{1126}$, low $E_{\mathrm{in}}(g_{1126})$ to fool your boss —**really? :-(a dangerous path of no return**
- safe route: $\mathcal{H}_1$ first
  - if $E_{\mathrm{in}}(g_1)$ 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 $\mathcal{Z} = \mathbf{\Phi}(\mathcal{X})$
- Price of Nonlinear Transform
  computation/storage/[model complexity]
- Structured Hypothesis Sets
  linear/simpler model first

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