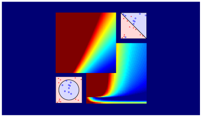
Machine Learning Foundations
(機器學習基石)



Lecture 12: Nonlinear Transformation

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 11: Linear Models for Classification

binary classification via **(logistic) regression**;
multiclass via **OVA/OVO decomposition**

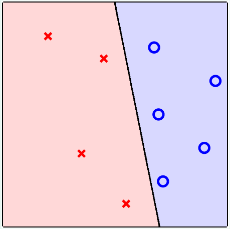
Lecture 12: Nonlinear Transformation

- Quadratic Hypotheses
- Nonlinear Transform
- Price of Nonlinear Transform
- Structured Hypothesis Sets

4. How Can Machines Learn Better?

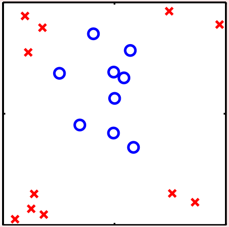
Linear Hypotheses

up to now: linear hypotheses



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

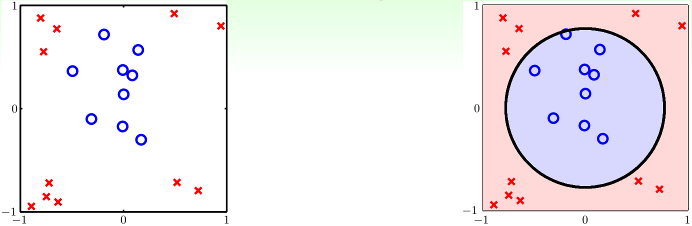
but limited . . .



- 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



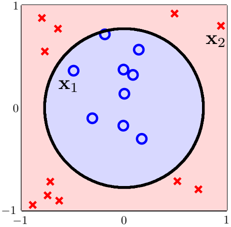
- $\mathcal{D}$ not linear separable
- but **circular separable** by a circle of radius $\sqrt{0.6}$ centered at origin:

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

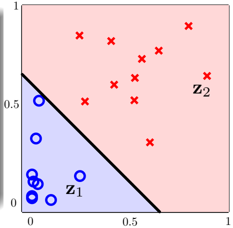
re-derive **Circular**-PLA, **Circular**-Regression, blahblah . . . all over again? **:-)**

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? 反之不成立

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 (○ inside)
- $(-0.6, +1, +1)$: circle (○ outside)
- $(0.6, -1, -2)$: ellipse 椭圆
- $(0.6, -1, +2)$: hyperbola 双曲线
- $(0.6, +1, +2)$: **constant ○ :-)**

lines in $\mathcal{Z}$-space

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

所以Z空间里的线对应着X空间中各式各样的形状（圆，椭圆，双曲线，等等），且只有特殊的二次曲线

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

Fun Time

Using the transform $\mathbf{\Phi}_2(\mathbf{x}) = (1, x_1, x_2, x_1^2, x_1 x_2, x_2^2)$, which of the following weights $\tilde{\mathbf{w}}^T$ in the $\mathcal{Z}$-space implements the parabola $2x_1^2 + x_2 = 1$?

1. $[-1, 2, 1, 0, 0, 0]$
2. $[0, 2, 1, 0, -1, 0]$
3. $[-1, 0, 1, 2, 0, 0]$
4. $[-1, 2, 0, 0, 0, 1]$

Fun Time

Using the transform $\mathbf{\Phi}_2(\mathbf{x}) = (1, x_1, x_2, x_1^2, x_1 x_2, x_2^2)$, which of the following weights $\tilde{\mathbf{w}}^T$ in the $\mathcal{Z}$-space implements the parabola $2x_1^2 + x_2 = 1$?

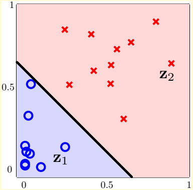
1. $[-1, 2, 1, 0, 0, 0]$
2. $[0, 2, 1, 0, -1, 0]$
3. $[-1, 0, 1, 2, 0, 0]$
4. $[-1, 2, 0, 0, 0, 1]$

Reference Answer: 3

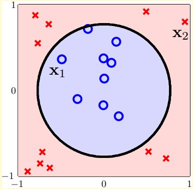
Too simple, uh? :-) Flexibility to implement arbitrary quadratic curves opens new possibilities for minimizing E_{in}!

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**



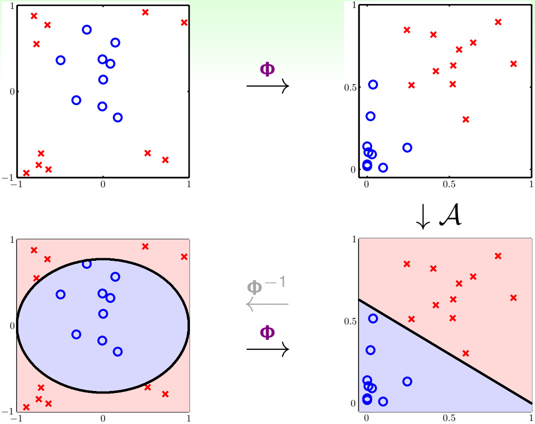
$\Longleftrightarrow$



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

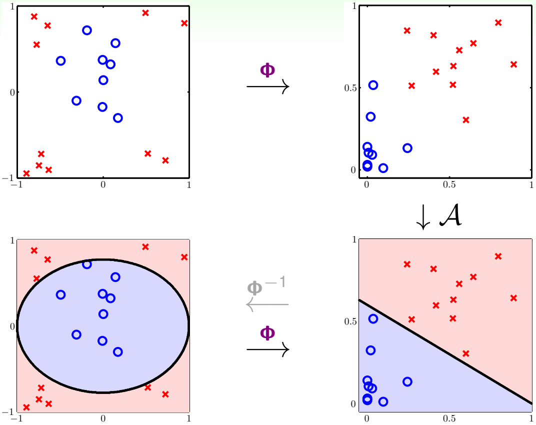
todo: get **good perceptron** in $\mathcal{Z}$-space with data $\{(\mathbf{z}_n = \mathbf{\Phi}_2(\mathbf{x}_n), y_n)\}$

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 **Φ** + Linear Models



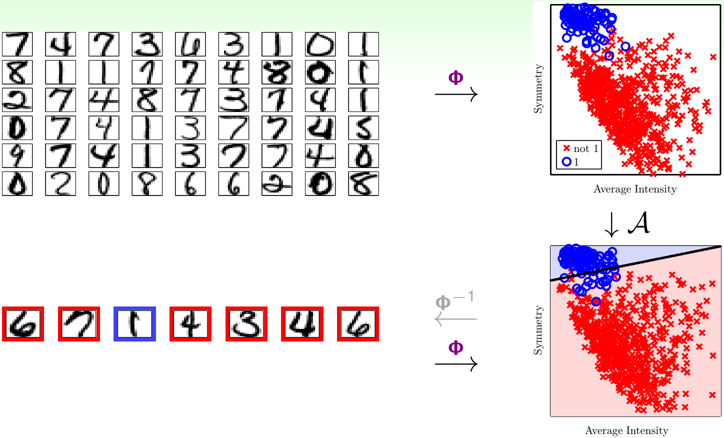
two choices:

- feature transform **Φ**
- 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? :-)

Fun Time

Consider the quadratic transform $\mathbf{\Phi}_2(\mathbf{x})$ for $\mathbf{x} \in \mathbb{R}^d$ instead of in $\mathbb{R}^2$. The transform should include all different quadratic, linear, and constant terms formed by $(x_1, x_2, \ldots, x_d)$. What is the number of dimensions of $\mathbf{z} = \mathbf{\Phi}_2(\mathbf{x})$?

1. d
2. $\frac{d^2}{2} + \frac{3d}{2} + 1$
3. $d^2 + d + 1$
4. 2^d

Fun Time

Consider the quadratic transform $\mathbf{\Phi}_2(\mathbf{x})$ for $\mathbf{x} \in \mathbb{R}^d$ instead of in $\mathbb{R}^2$. The transform should include all different quadratic, linear, and constant terms formed by $(x_1, x_2, \ldots, x_d)$. What is the number of dimensions of $\mathbf{z} = \mathbf{\Phi}_2(\mathbf{x})$?

1. d
2. $\frac{d^2}{2} + \frac{3d}{2} + 1$
3. $d^2 + d + 1$
4. 2^d

Reference Answer: ②

Number of different quadratic terms is $\binom{d}{2} + d$;
number of different linear terms is d;
number of different constant term is 1.

Computation/Storage Price

Q-th order polynomial transform: $\mathbf{\Phi}_Q(\mathbf{x}) = \left(\begin{array}{l} 1, \\ x_1, x_2, \ldots, x_d, \\ x_1^2, x_1 x_2, \ldots, x_d^2, \\ \ldots, \\ x_1^Q, x_1^{Q-1} x_2, \ldots, x_d^Q \end{array} \right)$

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

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

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

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

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

Model Complexity Price

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

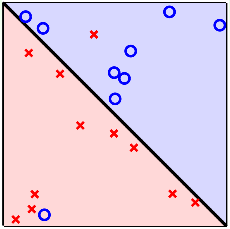
$\underbrace{1}_{\tilde{w}_0} + \underbrace{\tilde{d}}_{\text{others}}$ dimensions = $O\left(Q^d\right)$

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

$$\begin{aligned} & \text{any } \tilde{d} + 2 \text{ inputs not shattered in } \mathcal{Z} \\ \Longrightarrow\ & \text{any } \tilde{d} + 2 \text{ inputs not shattered in } \mathcal{X} \end{aligned}$$

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

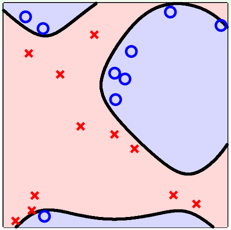
Generalization Issue



$\mathbf{\Phi}_1$ (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?

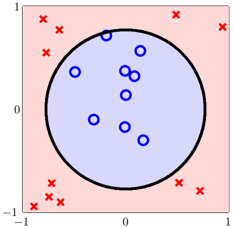
Danger of Visual Choices

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

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

- full $\mathbf{\Phi}_2$: $\mathbf{z} = (1, x_1, x_2, x_1^2, x_1 x_2, x_2^2)$, $d_{\mathrm{VC}} = 6$
- or $\mathbf{z} = (1, x_1^2, x_2^2)$, $d_{\mathrm{VC}} = 3$, **after visualizing**?
- or better $\mathbf{z} = (1, x_1^2 + x_2^2)$, $d_{\mathrm{VC}} = 2$?
- or even better $\mathbf{z} = \left(\mathrm{sign}(0.6 - x_1^2 - x_2^2)\right)$?

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



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

Fun Time

Consider the Q-th order polynomial transform $\mathbf{\Phi}_Q(\mathbf{x})$ for $\mathbf{x} \in \mathbb{R}^2$. Recall that $\tilde{d} = \binom{Q+2}{2} - 1$. When $Q = 50$, what is the value of $\tilde{d}$?

1. 1126
2. 1325
3. 2651
4. 6211

Fun Time

Consider the Q-th order polynomial transform $\mathbf{\Phi}_Q(\mathbf{x})$ for $\mathbf{x} \in \mathbb{R}^2$. Recall that $\tilde{d} = \binom{Q+2}{2} - 1$. When $Q = 50$, what is the value of $\tilde{d}$?

1. 1126
2. 1325
3. 2651
4. 6211

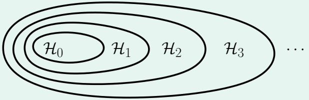
Reference Answer: ②

It's just a simple calculation, but shows you how $\tilde{d}$ becomes hundreds of times of $d = 2$ after the transform.

Polynomial Transform Revisited

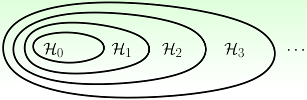
$$\begin{aligned}
\boldsymbol{\Phi}_0(\mathbf{x}) = \Big(1\Big), \boldsymbol{\Phi}_1(\mathbf{x}) &= \Big(\boldsymbol{\Phi}_0(\mathbf{x}), \quad x_1, x_2, \ldots, x_d\Big) \\
\boldsymbol{\Phi}_2(\mathbf{x}) &= \Big(\boldsymbol{\Phi}_1(\mathbf{x}), \quad x_1^2, x_1x_2, \ldots, x_d^2\Big) \\
\boldsymbol{\Phi}_3(\mathbf{x})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

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



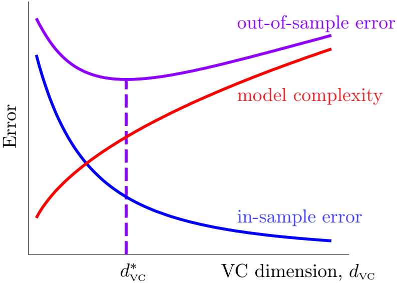
structure: **nested** $\mathcal{H}_i$

Structured Hypothesis Sets



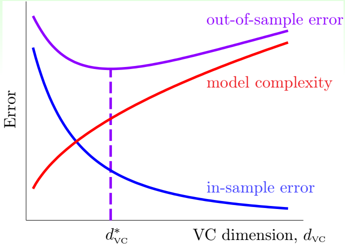
Let $g_i = \text{argmin}_{h \in \mathcal{H}_i} E_{\text{in}}(h)$:

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



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!**

Fun Time

Consider two hypothesis sets, $\mathcal{H}_1$ and $\mathcal{H}_{1126}$, where $\mathcal{H}_1 \subset \mathcal{H}_{1126}$. Which of the following relationship between $d_{\mathrm{VC}}(\mathcal{H}_1)$ and $d_{\mathrm{VC}}(\mathcal{H}_{1126})$ is not possible?

1. $d_{\mathrm{VC}}(\mathcal{H}_1) = d_{\mathrm{VC}}(\mathcal{H}_{1126})$
2. $d_{\mathrm{VC}}(\mathcal{H}_1) \neq d_{\mathrm{VC}}(\mathcal{H}_{1126})$
3. $d_{\mathrm{VC}}(\mathcal{H}_1) < d_{\mathrm{VC}}(\mathcal{H}_{1126})$
4. $d_{\mathrm{VC}}(\mathcal{H}_1) > d_{\mathrm{VC}}(\mathcal{H}_{1126})$

Fun Time

Consider two hypothesis sets, $\mathcal{H}_1$ and $\mathcal{H}_{1126}$, where $\mathcal{H}_1 \subset \mathcal{H}_{1126}$. Which of the following relationship between $d_{\text{VC}}(\mathcal{H}_1)$ and $d_{\text{VC}}(\mathcal{H}_{1126})$ is not possible?

1. $d_{\text{VC}}(\mathcal{H}_1) = d_{\text{VC}}(\mathcal{H}_{1126})$
2. $d_{\text{VC}}(\mathcal{H}_1) \neq d_{\text{VC}}(\mathcal{H}_{1126})$
3. $d_{\text{VC}}(\mathcal{H}_1) < d_{\text{VC}}(\mathcal{H}_{1126})$
4. $d_{\text{VC}}(\mathcal{H}_1) > d_{\text{VC}}(\mathcal{H}_{1126})$

Reference Answer: 4

Every input combination that $\mathcal{H}_1$ shatters can be shattered by $\mathcal{H}_{1126}$, so d_{VC} cannot decrease.

Summary

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

Lecture 11: Linear Models for Classification

Lecture 12: Nonlinear Transformation

- Quadratic Hypotheses

  linear hypotheses on quadratic-transformed data
- Nonlinear Transform

  happy linear modeling after $\mathcal{Z} = \boldsymbol{\Phi}(\mathcal{X})$
- Price of Nonlinear Transform

  computation/storage/[model complexity]
- Structured Hypothesis Sets

  linear/simpler model first

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

4. How Can Machines Learn Better?