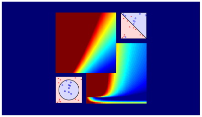
Machine Learning Foundations
(機器學習基石)



Lecture 13: Hazard of Overfitting

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 12: Nonlinear Transform

nonlinear □ via **nonlinear feature transform Φ**
plus **linear** □ with price of **model complexity**

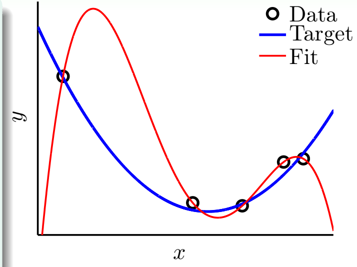
4. How Can Machines Learn **Better**?

Lecture 13: Hazard of Overfitting

- What is Overfitting?
- The Role of Noise and Data Size
- Deterministic Noise
- Dealing with Overfitting

Bad Generalization

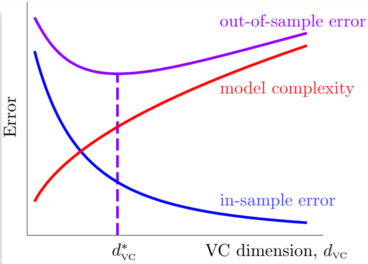
- regression for $x \in \mathbb{R}$ with $N = 5$ examples
- target $f(x)$ = 2nd order polynomial
- label $y_n = f(x_n) +$ very small noise
- linear regression in $\mathcal{Z}$-space + $\mathbf{\Phi}$ = 4th order polynomial
- unique solution passing all examples $\Longrightarrow E_{\text{in}}(g) = 0$
- $E_{\text{out}}(g)$ **huge**



bad generalization: low E_{in}, high E_{out}

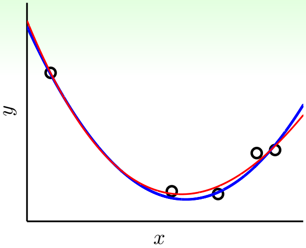
Bad Generalization and Overfitting

- take $d_{\text{VC}} = 1126$ for learning: bad generalization —(E_{out} - E_{in}) large
- switch from $d_{\text{VC}} = d^*_{\text{VC}}$ to $d_{\text{VC}} = 1126$: **overfitting** —$E_{\text{in}} \downarrow$, $E_{\text{out}} \uparrow$
- switch from $d_{\text{VC}} = d^*_{\text{VC}}$ to $d_{\text{VC}} = 1$: **underfitting** —$E_{\text{in}} \uparrow$, $E_{\text{out}} \uparrow$



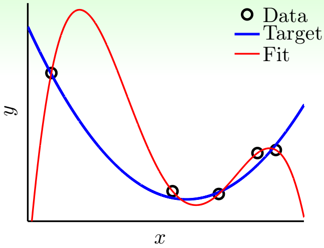
bad generalization: low E_{in}, high E_{out};
overfitting: low**er** E_{in}, high**er** E_{out}

Cause of Overfitting: A Driving Analogy



'good fit'

⟹



overfit

learning	driving
overfit	commit a car accident
use excessive d_{VC}	'drive too fast'
noise	bumpy road
limited data size N	limited observations about road condition

next: how does **noise** & **data size** affect overfitting?

Fun Time

Based on our discussion, for data of fixed size, which of the following situation is relatively of the lowest risk of overfitting?

1. small noise, fitting from small d_{VC} to median d_{VC}
2. small noise, fitting from small d_{VC} to large d_{VC}
3. large noise, fitting from small d_{VC} to median d_{VC}
4. large noise, fitting from small d_{VC} to large d_{VC}

Fun Time

Based on our discussion, for data of fixed size, which of the following situation is relatively of the lowest risk of overfitting?

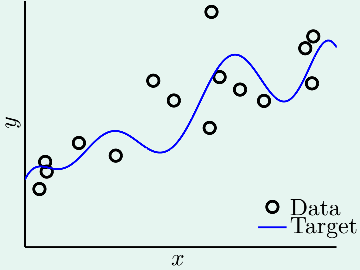
1. small noise, fitting from small d_{VC} to median d_{VC}
2. small noise, fitting from small d_{VC} to large d_{VC}
3. large noise, fitting from small d_{VC} to median d_{VC}
4. large noise, fitting from small d_{VC} to large d_{VC}

Reference Answer: (1)

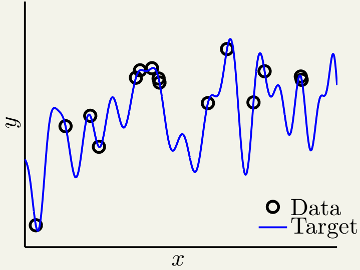
Two causes of overfitting are noise and excessive d_{VC}. So if both are relatively 'under control', the risk of overfitting is smaller.

Case Study (1/2)

10-th order target function **+ noise**



50-th order target function **noiselessly**



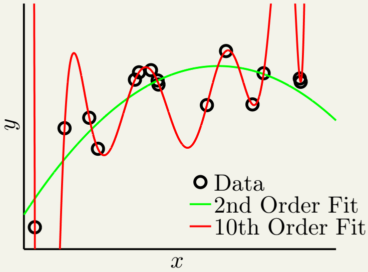
overfitting from best $g_2 \in \mathcal{H}_2$ to best $g_{10} \in \mathcal{H}_{10}$?

Case Study (2/2)

10-th order target function **+ noise**

	$g_2 \in \mathcal{H}_2$	$g_{10} \in \mathcal{H}_{10}$
E_{in}	0.050	0.034
E_{out}	0.127	**9.00**

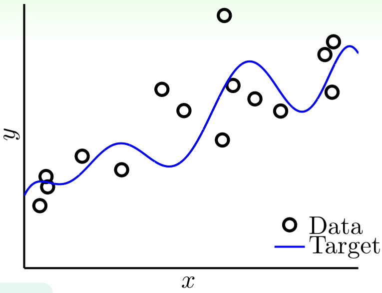
50-th order target function **noiselessly**



	$g_2 \in \mathcal{H}_2$	$g_{10} \in \mathcal{H}_{10}$
E_{in}	0.029	0.00001
E_{out}	0.120	**7680**

overfitting from g_2 to g_{10}? **both yes!**

Irony of Two Learners

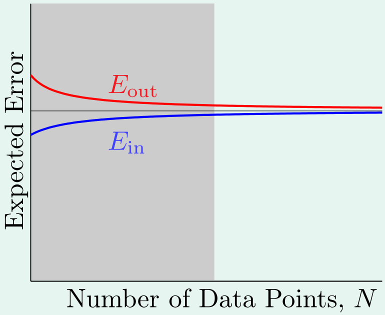


- learner *Overfit*: pick $g_{10} \in \mathcal{H}_{10}$
- learner *Restrict*: pick $g_2 \in \mathcal{H}_2$
- when both **know that target = 10th**
  —*R* 'gives up' ability to fit

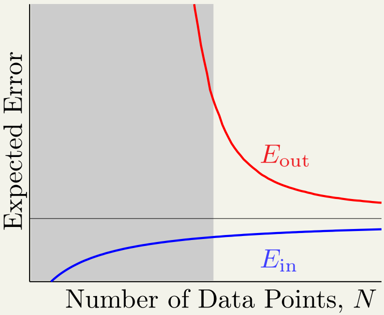
but *R* **wins in** E_{out} a lot!
philosophy: concession for **advantage**? :-)

Learning Curves Revisited

$\mathcal{H}_2$



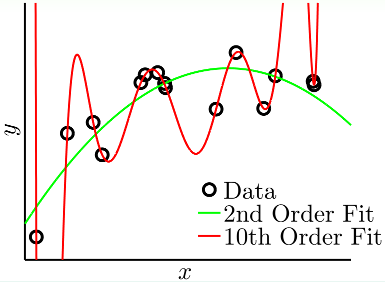
$\mathcal{H}_{10}$

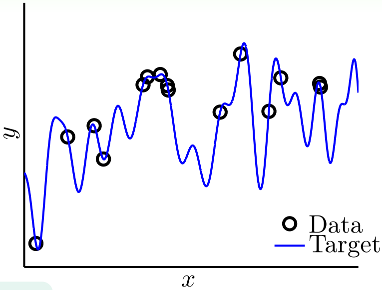


- $\mathcal{H}_{10}$: lower $\overline{E_{\text{out}}}$ when $N \to \infty$, but much larger generalization error for small N
- gray area: O overfits! ($\overline{E_{\text{in}}} \downarrow$, $\overline{E_{\text{out}}} \uparrow$)

R always **wins in** $\overline{E_{\text{out}}}$ if N small!

The 'No Noise' Case





- learner *Overfit*: pick $g_{10} \in \mathcal{H}_{10}$
- learner *Restrict*: pick $g_2 \in \mathcal{H}_2$
- when both **know that there is no noise** —*R* still wins

is there really **no noise**?
'target complexity' acts like noise

Fun Time

When having limited data, in which of the following case would learner *R* perform better than learner *O*?

1. limited data from a 10-th order target function with some noise
2. limited data from a 1126-th order target function with no noise
3. limited data from a 1126-th order target function with some noise
4. all of the above

Fun Time

When having limited data, in which of the following case would learner *R* perform better than learner *O*?

1. limited data from a 10-th order target function with some noise
2. limited data from a 1126-th order target function with no noise
3. limited data from a 1126-th order target function with some noise
4. all of the above

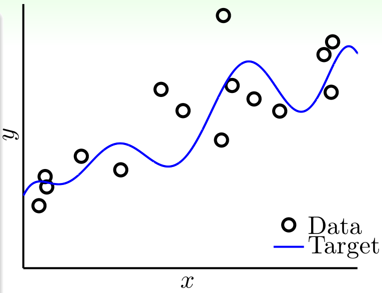
Reference Answer: (4)

We discussed about (1) and (2), but you shall be able to **'generalize' :-)** that *R* also wins in the more difficult case of (3).

A Detailed Experiment

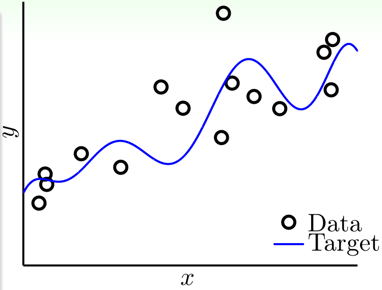
$$y = f(x) + \epsilon$$
$$\sim \text{Gaussian}\Bigg(\underbrace{\sum_{q=0}^{Q_f} \alpha_q x^q}_{f(x)}, \sigma^2\Bigg)$$

- Gaussian iid noise ϵ with level σ^2
- some 'uniform' distribution on $f(x)$ with complexity level Q_f
- data size N



goal: **'overfit level'** for different (N, σ^2) and (N, Q_f)?

The Overfit Measure

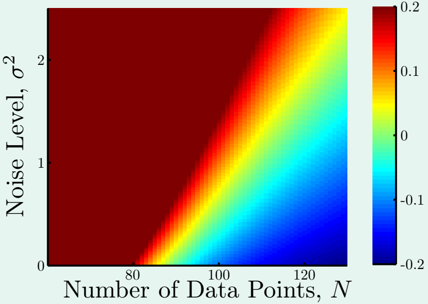


- $g_2 \in \mathcal{H}_2$
- $g_{10} \in \mathcal{H}_{10}$
- $E_{\text{in}}(g_{10}) \leq E_{\text{in}}(g_2)$ for sure

overfit measure $E_{\text{out}}(g_{10}) - E_{\text{out}}(g_2)$

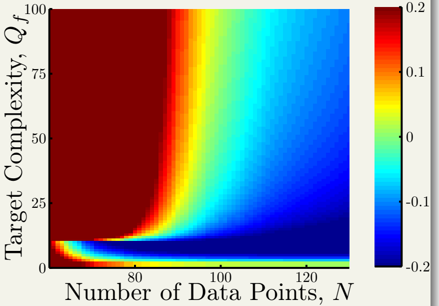
The Results

impact of σ^2 versus N



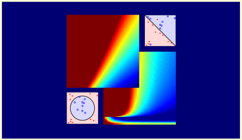
fixed $Q_f = 20$

impact of Q_f versus N



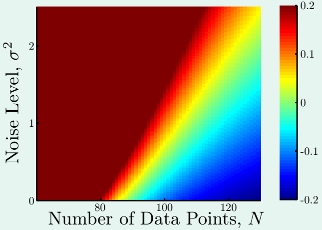
fixed $\sigma^2 = 0.1$

ring a bell? :-)

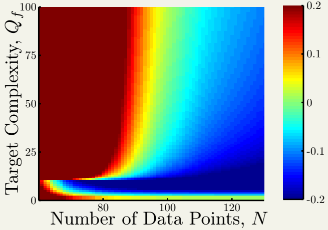


Impact of Noise and Data Size

impact of σ^2 versus N: **stochastic noise**



impact of Q_f versus N: **deterministic noise**



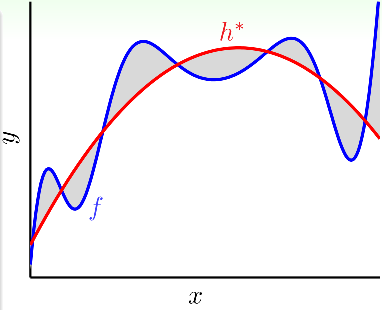
four reasons of serious overfitting:

data size $N \downarrow$	overfit $\uparrow$
stochastic noise $\uparrow$	overfit $\uparrow$
deterministic noise $\uparrow$	overfit $\uparrow$
excessive power $\uparrow$	overfit $\uparrow$

overfitting 'easily' happens

Deterministic Noise

- if $f \notin \mathcal{H}$: something of f cannot be captured by $\mathcal{H}$
- deterministic noise : difference between best $h^* \in \mathcal{H}$ and f
- acts like 'stochastic noise'—not new to CS: pseudo-random generator
- difference to stochastic noise:
  - depends on $\mathcal{H}$
  - fixed for a given $\mathbf{x}$



philosophy: when teaching a kid, perhaps better not to use examples from a complicated target function? **:-)**

Fun Time

Consider the target function being $\sin(1126x)$ for $x \in [0, 2\pi]$. When x is uniformly sampled from the range, and we use all possible linear hypotheses $h(x) = w \cdot x$ to approximate the target function with respect to the squared error, what is the level of deterministic noise for each x?

1. $|\sin(1126x)|$
2. $|\sin(1126x) - x|$
3. $|\sin(1126x) + x|$
4. $|\sin(1126x) - 1126x|$

Fun Time

Consider the target function being $\sin(1126x)$ for $x \in [0, 2\pi]$. When x is uniformly sampled from the range, and we use all possible linear hypotheses $h(x) = w \cdot x$ to approximate the target function with respect to the squared error, what is the level of deterministic noise for each x?

1. $|\sin(1126x)|$
2. $|\sin(1126x) - x|$
3. $|\sin(1126x) + x|$
4. $|\sin(1126x) - 1126x|$

Reference Answer: 1

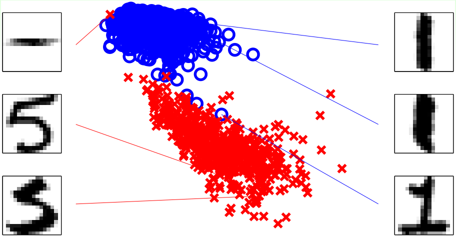
You can try a few different w and convince yourself that the best hypothesis h^* is $h^*(x) = 0$. The deterministic noise is the difference between f and h^*.

Driving Analogy Revisited

learning	driving
overfit	commit a car accident
use excessive d_{VC}	'drive too fast'
noise	bumpy road
limited data size N	limited observations about road condition
start from simple model	drive slowly
data cleaning/pruning	use more accurate road information
data hinting	exploit more road information
regularization	put the brakes
validation	monitor the dashboard

all very **practical** techniques
to combat overfitting

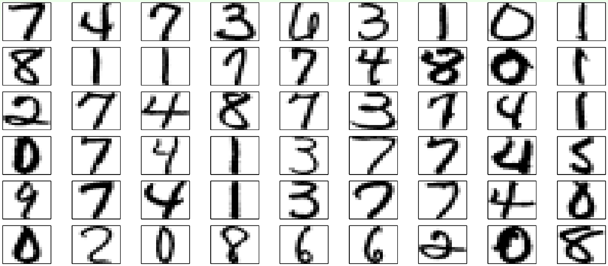
Data Cleaning/Pruning



- if ‘detect’ the outlier 5 at the top by
  - too close to other ○, or too far from other ×
  - wrong by current classifier
  - . . .
- possible action 1: correct the label (**data cleaning**)
- possible action 2: remove the example (**data pruning**)

possibly helps, but **effect varies**

Data Hinting



- slightly shifted/rotated digits carry the same meaning
- possible action: add **virtual examples** by shifting/rotating the given digits (**data hinting**)

possibly helps, but **watch out**
—**virtual example not** $\stackrel{iid}{\sim} P(\mathbf{x}, y)$!

Fun Time

Assume we know that $f(x)$ is symmetric for some 1D regression application. That is, $f(x) = f(-x)$. One possibility of using the knowledge is to consider symmetric hypotheses only. On the other hand, you can also generate virtual examples from the original data $\{(x_n, y_n)\}$ as hints. What virtual examples suit your needs best?

1. $\{(x_n, -y_n)\}$
2. $\{(-x_n, -y_n)\}$
3. $\{(-x_n, y_n)\}$
4. $\{(2x_n, 2y_n)\}$

Fun Time

Assume we know that $f(x)$ is symmetric for some 1D regression application. That is, $f(x) = f(-x)$. One possibility of using the knowledge is to consider symmetric hypotheses only. On the other hand, you can also generate virtual examples from the original data $\{(x_n, y_n)\}$ as hints. What virtual examples suit your needs best?

1. $\{(x_n, -y_n)\}$
2. $\{(-x_n, -y_n)\}$
3. $\{(-x_n, y_n)\}$
4. $\{(2x_n, 2y_n)\}$

Reference Answer: ③

We want the virtual examples to encode the invariance when $x \to -x$.

Summary

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

Lecture 12: Nonlinear Transform

4. How Can Machines Learn **Better**?

Lecture 13: Hazard of Overfitting

- What is Overfitting?
  lower E_{in} but higher E_{out}
- The Role of Noise and Data Size
  overfitting 'easily' happens!
- Deterministic Noise
  what $\mathcal{H}$ cannot capture acts like noise
- Dealing with Overfitting
  data cleaning/pruning/hinting, and more

- **next: putting the brakes with regularization**