Machine Learning from Data

Lecture 7: Spring 2021

Today's Lecture

- Approximation Vs Generalization
  - VC Dimension
  - Bias-Variance Analysis
  - Learning Curve

Theory of Learning and its relevance.

- We created a link between E_{out} and E_{in}

$$E_{out} \leq E_{in} + \sqrt{\frac{8}{N} \ln \frac{4(m_{\mathcal{H}}(2N))}{\delta}} \quad \text{(growth function)} \quad \text{— eq}$$

Universe ‖ Sample

Data. ⟷ E_{in}

E_{out} ↑ unknown

δ ↑ see

i) fail or succeed with high probability.

2-step learning process.

i) $E_{in} \simeq 0$ and $E_{out} \simeq E_{in}$ ✓ ideal ✓

ii) $E_{in} >> 0$ and $E_{out} \simeq E_{in}$ Not good

iii) $E_{in} \simeq 0$ $E_{out} >> 0$ ✗ Theory helps us avoid this

i) Time tested

ii) Link between E_{in} & E_{out}

iii) Relevance in practice.

$$\mathbb{P}\left[|E_{\text{in}}(g) - E_{\text{out}}(g)| > \epsilon\right] \leq 4m_{\mathcal{H}}(2N)e^{-\epsilon^2 N/8}, \quad \text{for any } \epsilon > 0.$$

$\mathbb{P}[|E_{\text{in}}(g)-E_{\text{out}}(g)|>\epsilon]\leq 2|\mathcal{H}|e^{-2\epsilon^2 N}$ $\leftarrow$ finite $\mathcal{H}$

$$\mathbb{P}\left[|E_{\text{in}}(g) - E_{\text{out}}(g)| \leq \epsilon\right] \geq 1 - 4m_{\mathcal{H}}(2N)e^{-\epsilon^2 N/8}, \quad \text{for any } \epsilon > 0.$$

$\mathbb{P}[|E_{\text{in}}(g)-E_{\text{out}}(g)|\leq\epsilon]\geq 1-2|\mathcal{H}|e^{-2\epsilon^2 N}$ $\leftarrow$ finite $\mathcal{H}$

$$E_{\text{out}}(g) \leq E_{\text{in}}(g) + \sqrt{\frac{8}{N}\log\frac{4m_{\mathcal{H}}(2N)}{\delta}}, \quad \text{w.p. at least } 1-\delta.$$

$E_{\text{out}}(g)\leq E_{\text{in}}(g)+\sqrt{\frac{1}{2N}\log\frac{2|\mathcal{H}|}{\delta}}$ $\leftarrow$ finite $\mathcal{H}$

$\mathcal{H}, f, P(x), A$

$$m_{\mathcal{H}}(N) \leq \sum_{i=1}^{k-1}\binom{N}{i} \leq N^{k-1}+1$$ (Polynomial)

k is a break point.

2 kinds of $\mathcal{H}$ sets.

1) Bad $m_{\mathcal{H}}(N) = 2^N \quad \forall N \geq 1$, $\Omega = \sqrt{\frac{8}{N} \log \frac{4(2^{2N})}{\delta}} \neq 0$

E_{in} & E_{out} — expand

2) $m_{\mathcal{H}}(N) \longrightarrow$ polynomially

$$m_{\mathcal{H}}(N) \leq \sum_{i=0}^{k-1} \binom{N}{i} \leq N^{k-1} + 1 \simeq N^{k-1}$$

$$\Omega = \sqrt{\frac{8}{N} \log \frac{4(2N)^{k-1}}{\delta}}$$

$$\Omega = \sqrt{\frac{8(k-1)\log(N)}{N}} \longrightarrow 0$$

The VC Dimension

- The VC dimension of a $\mathcal{H}$ is one less than the cheapest breakpoints (smallest)

$$d_{VC} = k_* - 1$$

k_* → smallest

d_{VC} → max. no. of data points that can be shattered.

If $N \leq d_{VC}$ → $\mathcal{H}$ can shatter data

If $N > d_{VC}$ → $\mathcal{H}$ cannot shatter your data

Summarize

$$m_{\mathcal{H}}(N) \sim N^{k-1}$$

The tightest bound is obtained with the smallest break point k^*.

Definition [VC Dimension] $d_{\text{VC}} = k^* - 1$.

The VC dimension is the largest N which can be shattered ($m_{\mathcal{H}}(N) = 2^N$).

- $N \leq d_{\text{VC}}$: $\mathcal{H}$ *could* shatter your data ($\mathcal{H}$ can shatter some N points).
- $N > d_{\text{VC}}$: N is a break point for $\mathcal{H}$; $\mathcal{H}$ cannot possibly shatter your data.

$$m_{\mathcal{H}}(N) \leq N^{d_{\text{VC}}} + 1 \sim N^{d_{\text{VC}}}$$

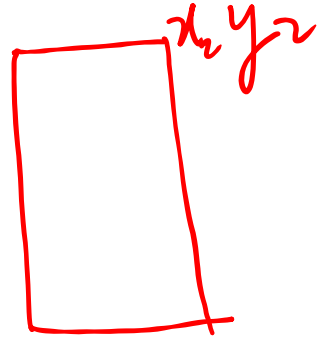
$$E_{\text{out}}(g) \leq E_{\text{in}}(g) + O\left(\sqrt{\frac{d_{\text{VC}} \log N}{N}}\right)$$

Examples

- N

N	1	2	3	4	5	6	dvc
2-d perception	2	4	8	14			3
1-d tve ray	2	3			$< 2^5$		1
2-d tve rectangle	2	4	8	16			4
2^N	2	4	8	16	32	64	

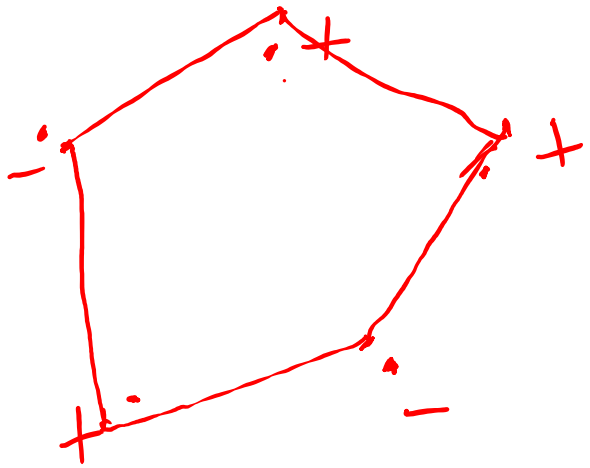
2-d perceptions	w_0, w_1, w_2 (w_0 threshold)	3
1-d tve ray	x_1, y_1 x_2, y_2	1
2-d tve rectangle		4

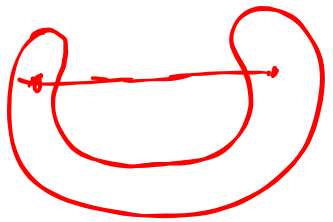


* Relationship we cannot always guarantee

Convex Sets

- the convex sets. ($\mathcal{H}$)





$d_{VC} = \infty$

Not useful for learning.

Summarize

	N							
	1	2	3	4	5	$\cdots$	#Param	d_{VC}
2-D perceptron	2	4	8	14	$\cdots$		3	3
1-D pos. ray	2	3	4	5	$\cdots$		1	1
2-D pos. rectangles	2	4	8	16	$< 2^5$	$\cdots$	4	4
pos. convex sets	2	4	8	16	32	$\cdots$	∞	∞

There are models with few parameters but infinite d_{VC}.

There are models with redundant parameters but small d_{VC}.

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

d - dimensional we have $w_0, w_1, \ldots w_d$

$= d+1$ parameters.

Theorem : The VC - dimension of the perceptron

$$\boxed{d_{VC}(\text{perceptron}) = d+1}$$

$$E_{out} \leq E_{in} + O\left(\sqrt{d_{VC}\frac{\ln N}{N}}\right)$$

Proof : (ex: 2.4 — HW problem)

i) Show that $d_{VC} \geq d+1$ → Any $(d+1)$ points d_{VC} = max. no. of data points you can shatter

ii) Show that $d_{VC} \leq d+1$

→ $d+2$ points. → linear algebra (hint)

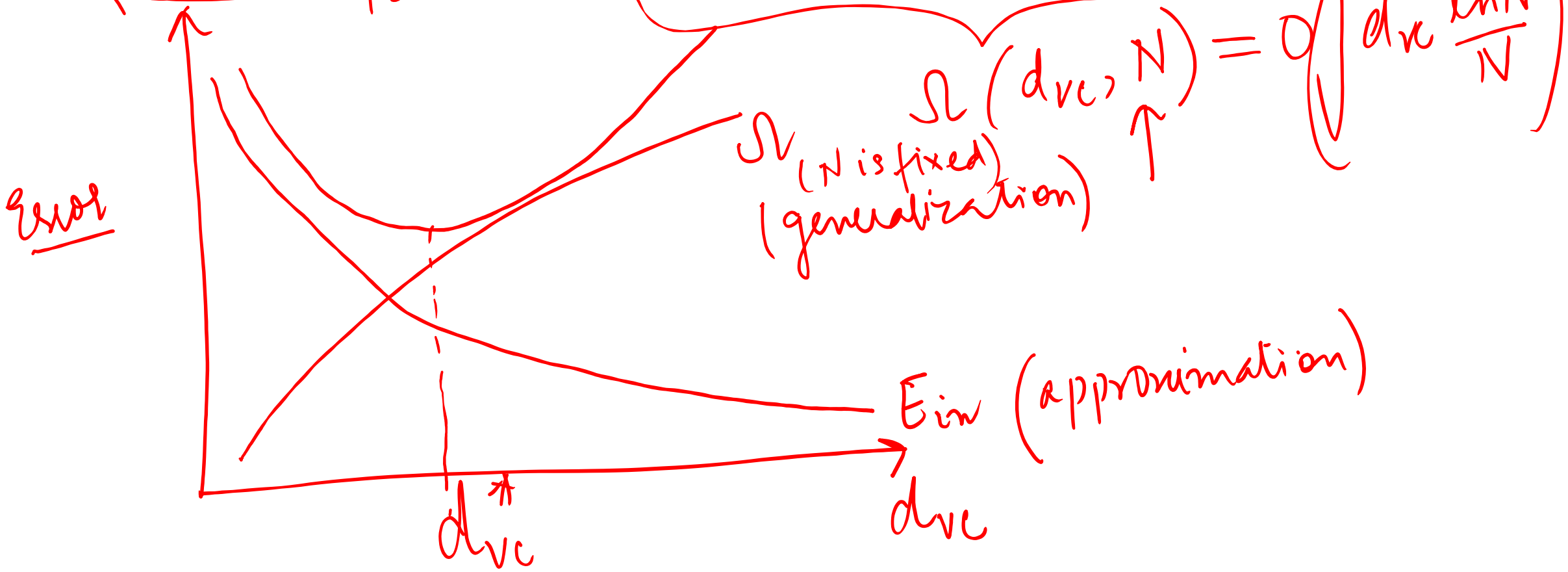
Approximation vs Generalization
(Bias - Variance trade-off)

$$E_{out}(g) \leq E_{in}(g) + \sqrt{\frac{8}{N}\ln\left(\frac{4\left((2N)^{d_{vc}}+1\right)}{\delta}\right)}$$

E_{out} ↑ ; E_{in} ↓

$\Omega(d_{vc}, N) = O\left(\sqrt{d_{vc}\frac{\ln N}{N}}\right)$ ↑ complexity

Ω (N is fixed) (generalization)



$\mathcal{H}$ (neural network) $\longrightarrow$ d_{VC} $\longrightarrow$ picking some N

$E_{in} \simeq E_{out}$

SAMPLE COMPLEXITY

$\varepsilon \longrightarrow$ generalization error

$\delta \longrightarrow$ confidence parameter.

$$\varepsilon = \sqrt{\frac{8}{N} \frac{\ln 4(2N)^{d_{VC}}}{\delta}}$$

Sample Complexity?

Set the error bar at ϵ.

$$\epsilon = \sqrt{\frac{8}{N} \ln \frac{4((2N)^{d_{\text{vc}}} + 1)}{\delta}}$$

Solve for N:

$$N = \frac{8}{\epsilon^2} \ln \frac{4((2N)^{d_{\text{vc}}} + 1)}{\delta} = O\left(d_{\text{vc}} \ln N\right)$$

$N = 10 \times d_{vc}$ (sufficient)

$N = 100 \times d_{vc}$

$N = 1000 \times d_{vc}$ great

2-d perception

1-8

Example. $d_{\text{vc}} = 3$; error bar $\epsilon = 0.1$; confidence 90% ($\delta = 0.1$). A simple iterative method works well. Trying $N = 1000$ we get

$$N \approx \frac{1}{0.1^2} \log\left(\frac{4(2000)^3 + 4}{0.1}\right) \approx 21192.$$

We continue iteratively, and converge to $N \approx 30000$.
If $d_{\text{vc}} = 4$, $N \approx 40000$; for $d_{\text{vc}} = 5$, $N \approx 50000$.

($N \propto d_{\text{vc}}$, but gross overestimates)

Practical Rule of Thumb: $N = 10 \times d_{\text{vc}}$

Theory Vs Practice

- Practice bulls us

↓ VC

general
$\mathcal{H}$-sets, LA,
Input (X), P^d,
f

VC, $N \approx 10,000$

$N \downarrow$

i) Hoeffding Inequality ($\mathcal{H}$-sets)
Accomodate for general

ii) $m_{\mathcal{H}}(N)$ ↔ worst case
Any dataset

iii) $m_{\mathcal{H}}(N) \leq B(N,k) \leq N^{d_{vc}+1}$

$d_{vc}+1$

technical slack

VC Bound Quantifies Approximation Vs. Generalization

$d_{\text{VC}} \uparrow \implies$ better chance of **approximating** f $(E_{\text{in}} \approx 0)$.

$d_{\text{VC}} \downarrow \implies$ better chance of **generalizing** to out of sample $(E_{\text{in}} \approx E_{\text{out}})$.

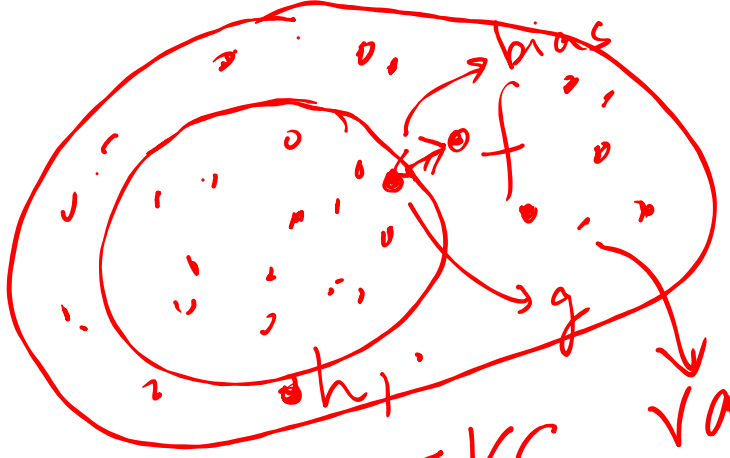
i) N-fixed $\xrightarrow{d_{VC}}$ $\mathcal{H}$-set.

ii) $\mathcal{H}$-set $\xrightarrow[\text{fixed}]{d_{VC}}$ N-choose

$$E_{\text{out}} \leq E_{\text{in}} + \Omega(d_{\text{VC}}).$$

model complexity.

Bias Variance Trade-off



Bias

1. How well *can* the learning approximate f.

   ... as opposed to how well *did* the learning approximate f in-sample (E_{in}).

2. How close can you get to that approximation with a finite data set.

   ... as opposed to how close is E_{in} to E_{out}.

VC

VC

Variance

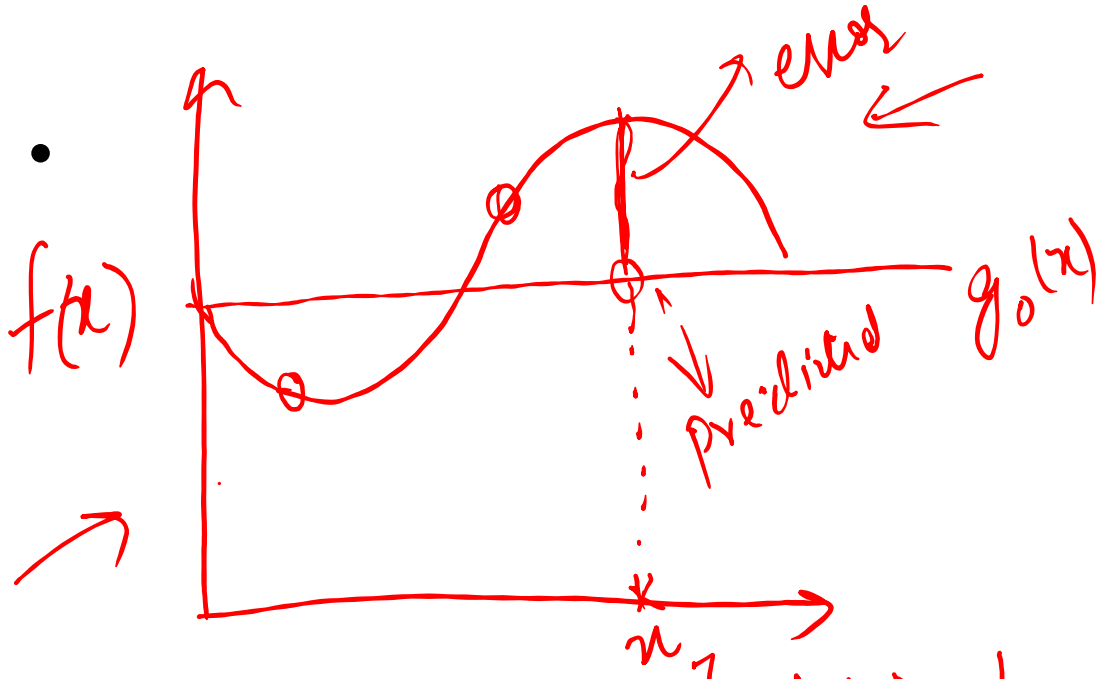
regression problems

Bias-variance analysis applies to squared errors (classification and regression)

Bias-variance analysis can take into account the *learning algorithm*

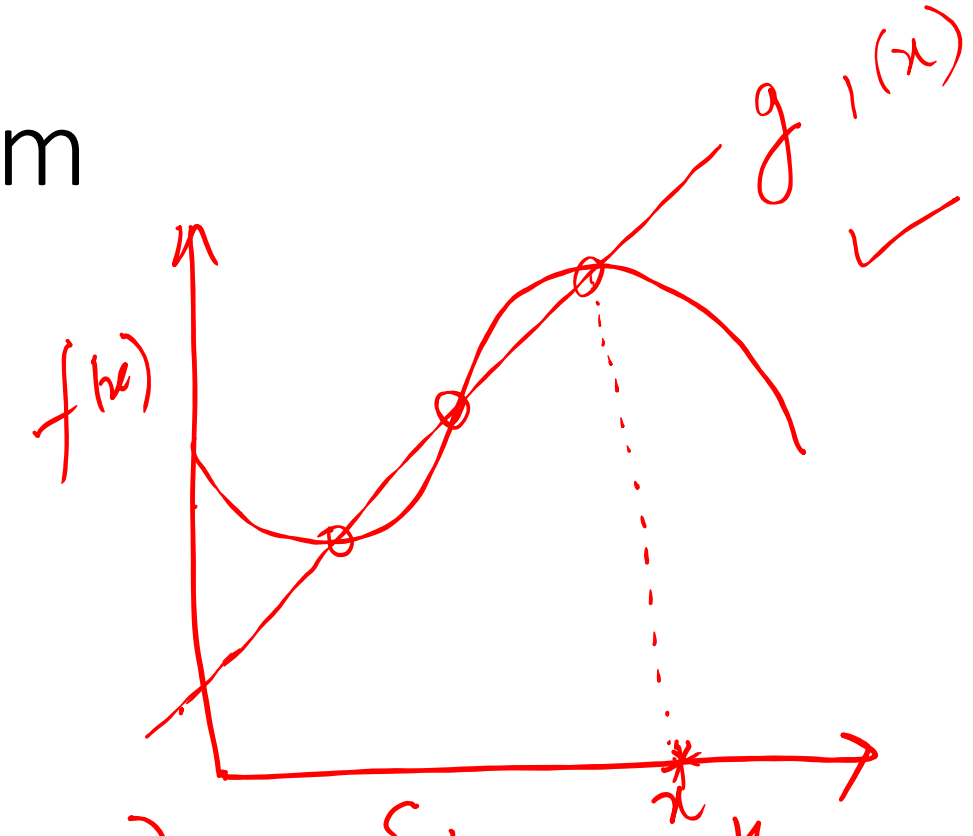
Different learning algorithms can have different E_{out} when applied to the same $\mathcal{H}$!

A simple learning problem



i) $\mathcal{H}_0 = \{\text{flat line}\}$ $h(x) = b$

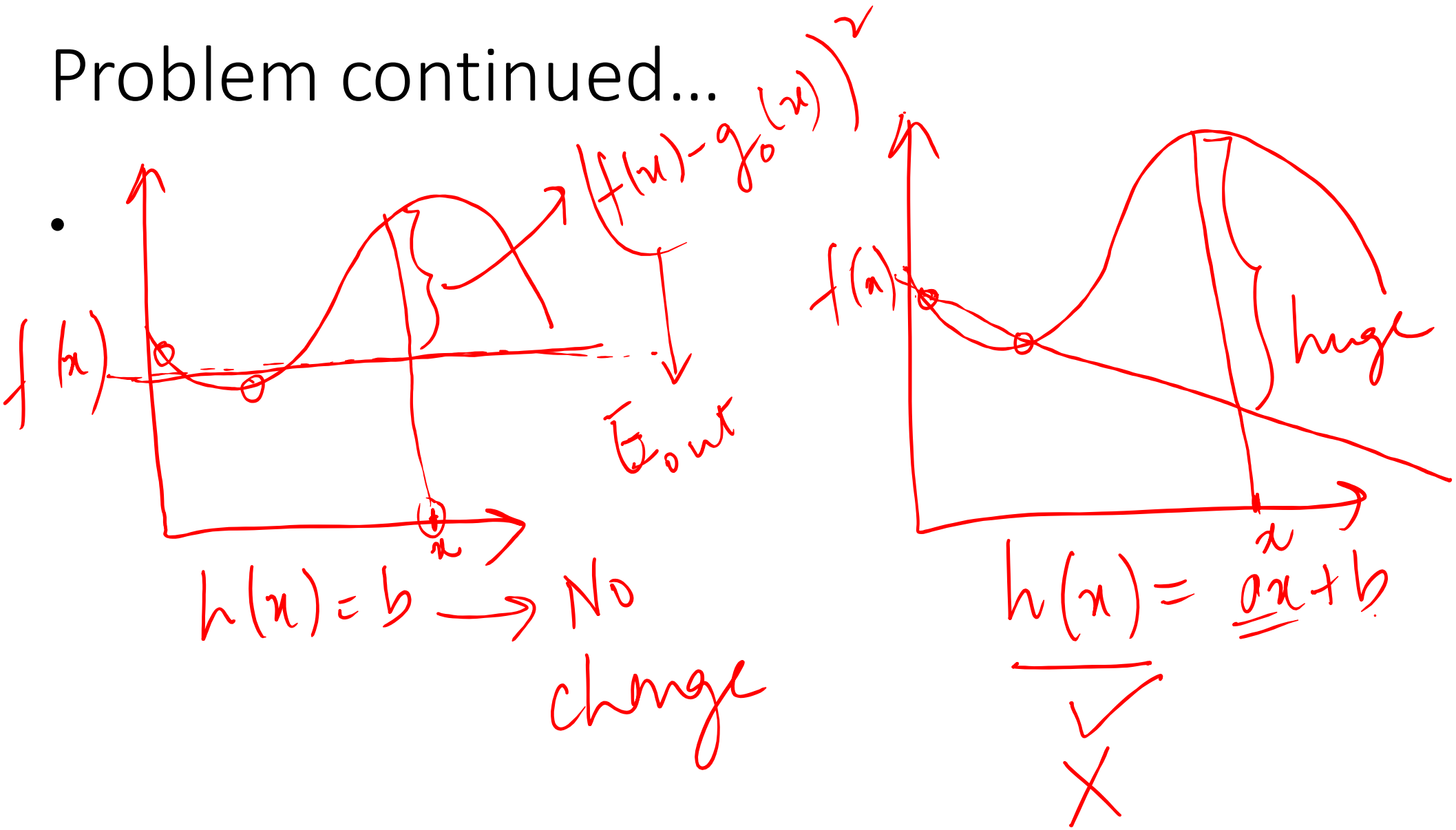
$(f(x) - g_0(x))^2$



ii) $\mathcal{H}_1 = \{\text{line with a slope}\}$

$(f(x) - g_1(x))^2$ $ax + b$

Problem continued…

- 

Expected Behavior with Datasets

- $D_1 \rightarrow g^{D_1} \rightarrow g^{D_1}(x)$
  $D_2 \rightarrow g^{D_2} \rightarrow g^{D_2}(x)$
  $D_3 \rightarrow g^{D_3}$
  $\vdots$
  $D_M \rightarrow g^{D_M} \rightarrow g^{D_M}(x)$

$$\bar{g}(x) = \frac{1}{M} \sum_{i=1}^{M} g^{D_i}(x)$$

$$\rightarrow E_D\left[g^D(x)\right] \rightarrow \text{expected prediction}$$

$$Var(x) = \frac{1}{M} \sum_{i=1}^{M} \left(g^{D_i}(x) - \bar{g}(x)\right)^2$$

$$E_D\left[\left(g^D(x) - \bar{g}(x)\right)^2\right]$$

$\downarrow$

Variance of prediction

$\left(f(x) - g^{Di}(x)\right)^2$ → Avg. error = $\frac{1}{M}\sum_{i=1}^{M}\left(f(x) - g^{Di}(x)\right)^2$

Result from probability

constant

steps (Textbook)

$\left(f(x) - \bar{g}(x)\right)^2 + Var(x)$

bias

BIAS VARIANCE DECOMPOSITION

Theorem: $E_D\left[\left(f(x) - g^D(x)\right)^2\right] = \underbrace{\left(f(x) - \bar{g}(x)\right)^2}_{\text{bias}(x)} + \underbrace{\text{Var}(x)}_{\text{Var}(x)}$

expected Error

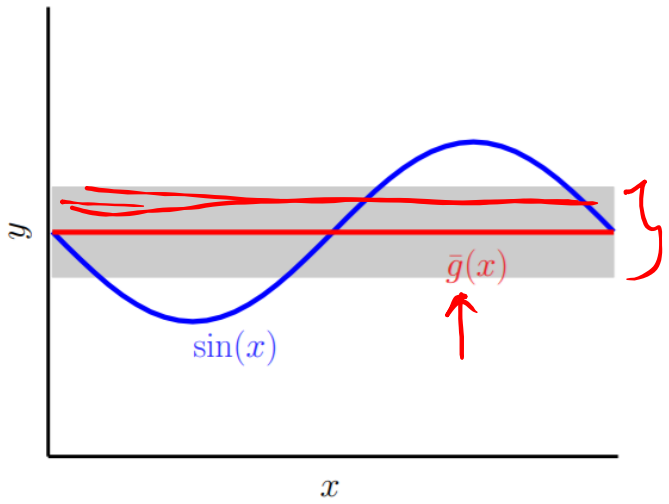
$$E_x\left[E_D\left[f(x) - g^D(x)\right]^2\right] = E_x(\text{bias}(x)) + E_x(\text{Var}(x))$$

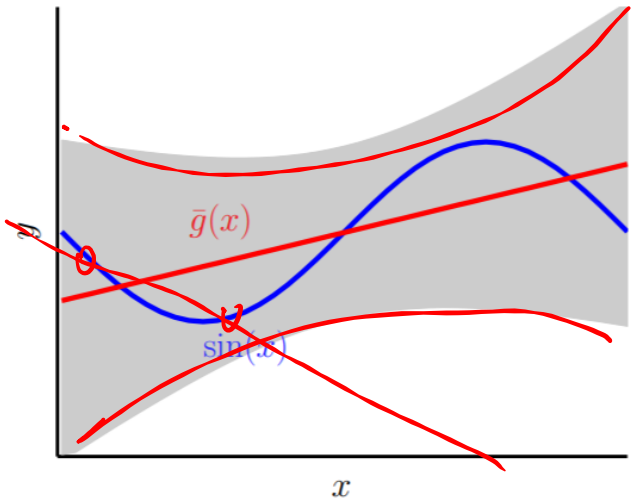
$h_0(x) = b$ } Bias ↑ Variance ↓

$h_1(x) = ax + b$ } Bias ↓ Variance ↑ → generalization

↓ approx

Which one is better?





$\mathcal{H}_0$

bias $= 0.50$

var $= 0.25$

$E_{\text{out}} = 0.75$ ✓

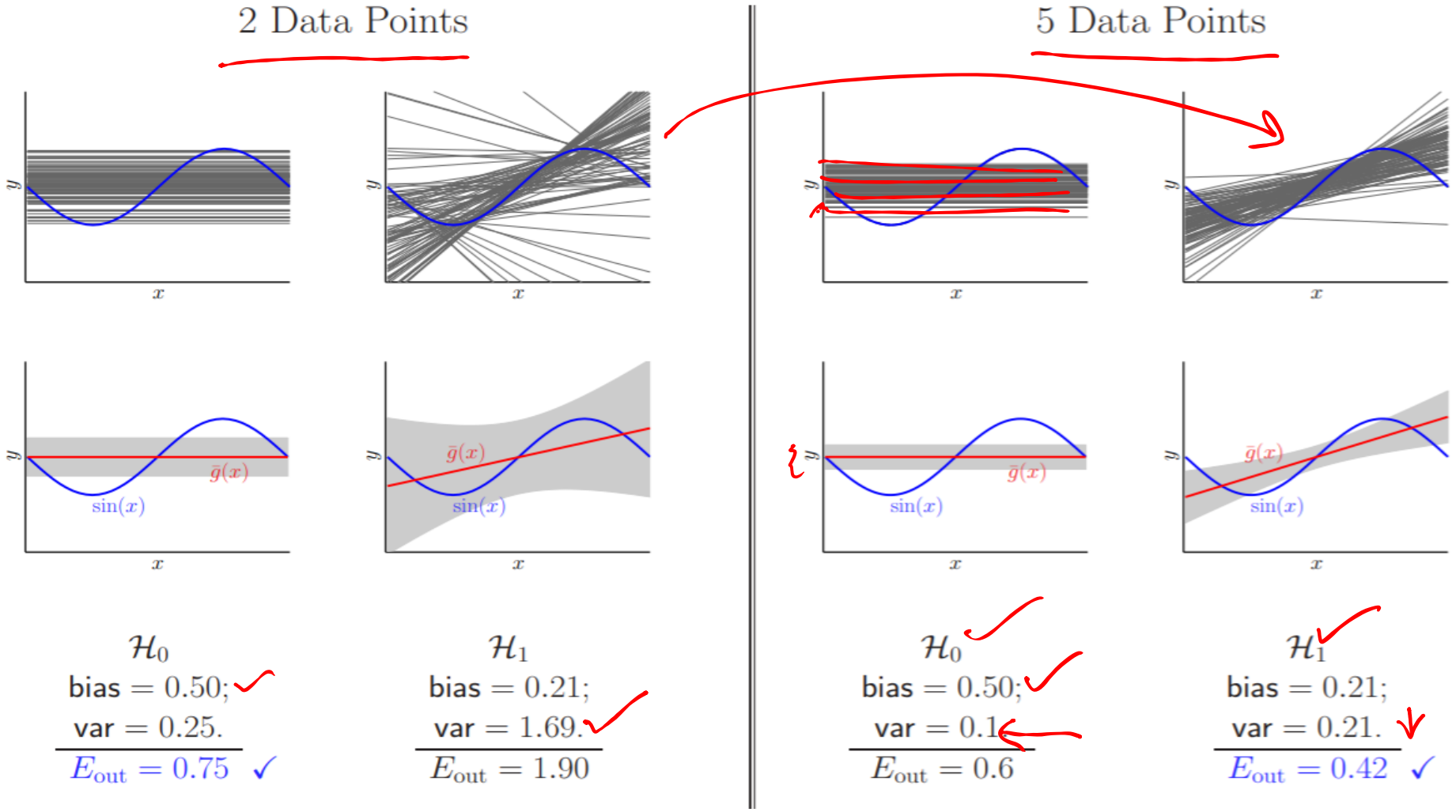
$\mathcal{H}_1$ ✓

bias $= 0.21$

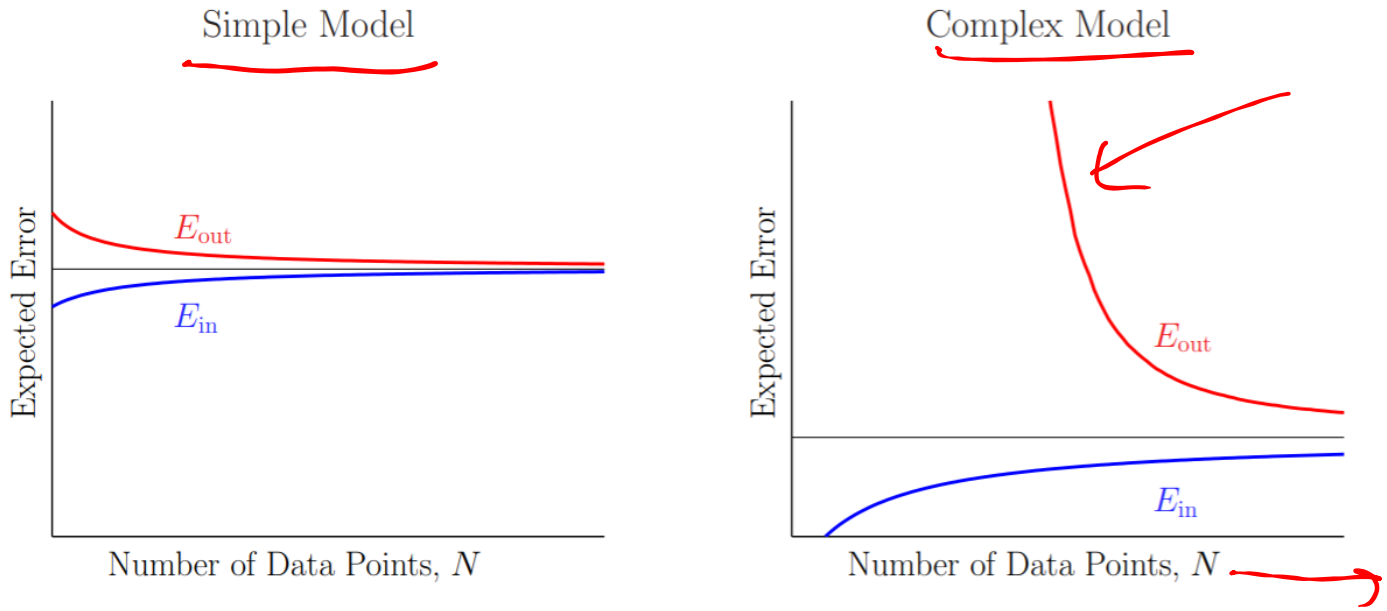
var $= 1.69$

$E_{\text{out}} = 1.90$

Data


2 Data Points
5 Data Points
y
x
$\bar{g}(x)$
$\sin(x)$
$\mathcal{H}_0$
bias $= 0.50$;
var $= 0.25$.
$E_{\text{out}} = 0.75$
$\mathcal{H}_1$
bias $= 0.21$;
var $= 1.69$.
$E_{\text{out}} = 1.90$
$\mathcal{H}_0$
bias $= 0.50$;
var $= 0.1$
$E_{\text{out}} = 0.6$
$\mathcal{H}_1$
bias $= 0.21$;
var $= 0.21$.
$E_{\text{out}} = 0.42$

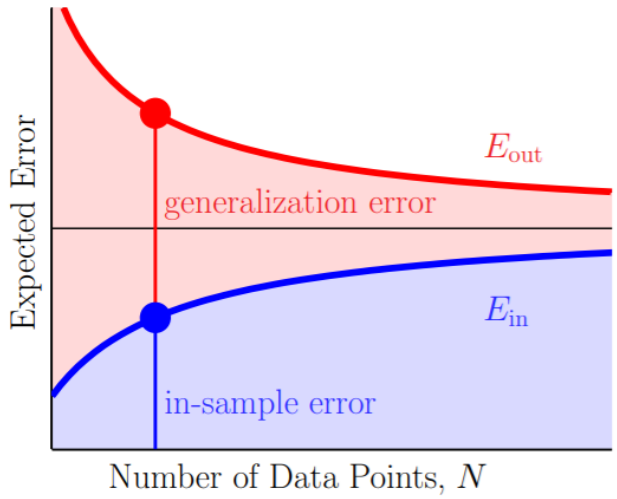
Learning Curve



$$E_{\text{out}} = \mathbb{E}_{\mathbf{x}}\left[E_{\text{out}}(\mathbf{x})\right]$$

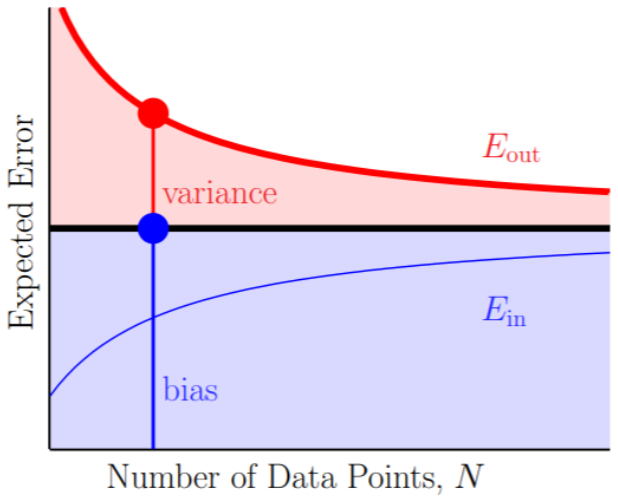
Comparison

VC Analysis



Pick $\mathcal{H}$ that can generalize and has a good chance to fit the data

Bias-Variance Analysis



Pick $(\mathcal{H}, \mathcal{A})$ to approximate f and not behave wildly after seeing the data

$\bar{g}$

$g^{(\mathcal{D})}$

Thanks!