CSE 417T
Introduction to Machine Learning

Lecture 6

Instructor: Chien-Ju (CJ) Ho

Recap

Theory of Generalization

- Learning from finite hypothesis set: learn $g \in \{h_1, \dots, h_M\}$

$$\Pr[|E_{out}(g) - E_{in}(g)| > \epsilon] \leq 2Me^{-2\epsilon^2 N}$$

- What can we say about infinite hypothesis set with $M \to \infty$

- Counting "effective" # hypothesis
  - Dichotomies
    - Informally, consider a dichotomy as "data-dependent" hypothesis
    - Characterized by both H and N data points $(\vec{x}_1, \dots, \vec{x}_N)$

$$H(\vec{x}_1, \dots \vec{x}_N) = \{h(\vec{x}_1), \dots, h(\vec{x}_N) | h \in H\}$$

    - The set of possible prediction combinations $h \in H$ can induce on $\vec{x}_1, \dots, \vec{x}_N$

  - Growth function
    - Largest number of dichotomies H can induce across all possible data sets of size N

$$m_H(N) = \max_{(\vec{x}_1, \dots, \vec{x}_N)} |H(\vec{x}_1, \dots, \vec{x}_N)|$$

Why Growth Function?

- Growth function $m_H(N)$
  - Largest number of "effective" hypothesis H can induce on N data points
  - A more precise "complexity" measure for H
  - Goal: Replace M in finite-hypothesis analysis with $m_H(N)$
    - With prob at least $1-\delta, E_{out}(g) \leq E_{in}(g) + \sqrt{\frac{1}{2N} ln \frac{2M}{\delta}}$

- VC Generalization Bound (VC Inequality, 1971)

  With prob at least $1-\delta$

$$E_{out}(g) \leq E_{in}(g) + \sqrt{\frac{8}{N} ln \frac{4m_H(2N)}{\delta}}$$

Bounding Growth Functions

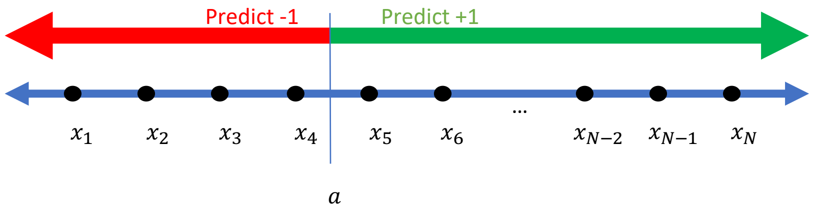
- More definitions….
  - Shatter
    - H **shatters** $(\vec{x}_1, \ldots, \vec{x}_N)$ if $|H(\vec{x}_1, \ldots, \vec{x}_N)| = 2^N$
    - H can induce all label combinations for $(\vec{x}_1, \ldots, \vec{x}_N)$
  - Break point
    - k is a **break point** for H if no data set of size k can be shattered by H
    - k is a break point for $H \leftrightarrow m_H(k) < 2^k$
  - VC Dimension: $d_{vc}(H)$ or d_{vc}
    - The VC dimension of H is the largest N such that $m_H(N) = 2^N$
    - Equivalently, if k^* is the smallest break point for H, $d_{vc}(H) = k^* - 1$

Examples

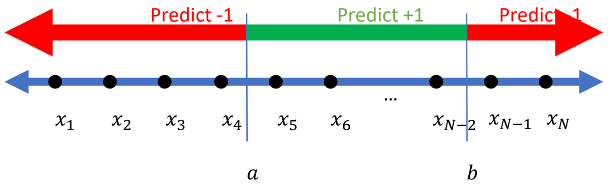
$\boldsymbol{m_H(N)}$

	N=1	N=2	N=3	N=4	N=5	Break Points	VC Dimension
Positive Rays	2	3	4	5	6	$k = 2,3,4,\ldots$	1
Positive Intervals	2	4	7	11	16	$k = 3,4,5,\ldots$	2
Convex Sets	2	4	8	16	32	None	∞
2D Perceptron	2	4	8	14	?	$k = 4,5,6,\ldots$	3

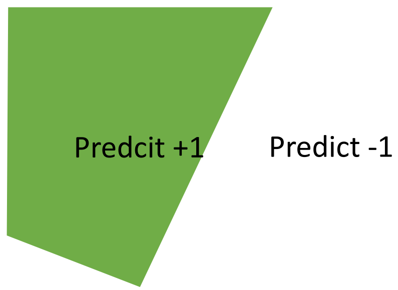
Positive Rays



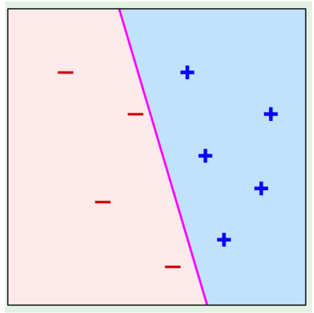
Positive Intervals



Convex Sets



2D Perceptron



Bounding Growth Functions

- Theorem statement:
  - If there is no break point for H, then $m_H(N) = 2^N$ for all N.
  - If k is a break point for H, i.e., if $m_H(k) < 2^k$ for some value k, then

$$m_H(N) \leq \sum_{i=0}^{k-1} \binom{N}{i}$$

- Rephrase the 2nd statement of the above theorem
  - If k is a break point for H, the following statements are true
    - $m_H(N) \leq N^{k-1} + 1$ [Can be proven using induction from above. See LFD Problem 2.5]
    - $m_H(N) = O(N^{k-1})$
    - $m_H(N)$ is polynomial in N
  - If d_{vc} is the VC dimension of H, then
    - $m_H(N) \leq \sum_{i=0}^{d_{vc}} \binom{N}{i}$
    - $m_H(N) \leq N^{d_{vc}} + 1$
    - $m_H(N) = O(N^{d_{vc}})$

If d_{vc} is the VC dimension of H, $d_{vc} + 1$ is a break point for H

Vapnik–Chervonenkis (VC) Bound

- VC Generalization Bound

  With prob at least $1 - \delta$

$$E_{out}(g) \leq E_{in}(g) + \sqrt{\frac{8}{N} ln \frac{4m_H(2N)}{\delta}}$$

- Let d_{vc} be the VC dimension of H, we have $\boldsymbol{m_H(N) \leq N^{d_{vc}} + 1}$. Therefore,

  With prob at least $1 - \delta$

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

- If we <u>treat δ as a constant</u>, then we can say, with high probability

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

Discussion on the VC Bound

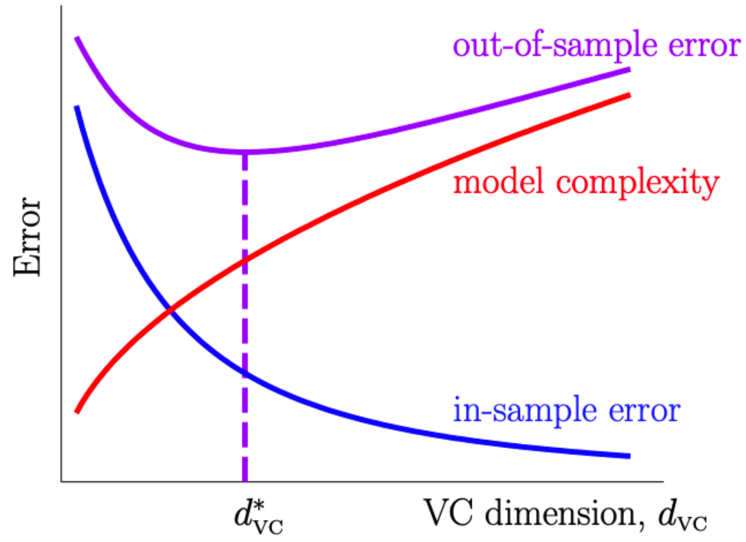
- Think about the high-level tradeoff of choosing d_{VC} and its dependency on N
- The approximation-generalization trade-off

What we want to minimize

How well g generalizes

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

How well g approximates f in training data



Today’s Lecture

The notes are not intended to be comprehensive. They should be accompanied by lectures and/or textbook.

Let me know if you spot errors.

Recall: Theorem in Bounding Growth Function

- Theorem statement:
  - If there is no break point for H, then $m_H(N) = 2^N$ for all N.
  - If k is a break point for H, i.e., if $m_H(k) < 2^k$ for some value k, then

$$m_H(N) \leq \sum_{i=0}^{k-1} \binom{N}{i}$$

- You are asked to take this as a fact last lecture
- Will provide proof sketch now

Proof Sketch

[See LFD Section 2.1.2 for the formal proof]

[Safe to Skip] (This proof won’t appear in exams/homework)

[Safe to Skip]

Key Intuitions

- When there exist a break point k
  - No datasets of size k can be shattered
  - It also imposes strong constraints on dataset of size $k' > k$
    - No subset of data with size k can be shattered
  - This leads to the bound $m_H(N) = O(N^{k-1})$

[Safe to Skip]

Proof Intuitions

- Max # dichotomies can you list on **2 points** when **no 2 points can be shattered**

$\vec{x}_1$	$\vec{x}_2$
+1	+1
+1	-1
-1	+1

[Safe to Skip]

Proof Intuitions

- Max # dichotomies can you list on **4 points** when **no 2 points can be shattered**

$\vec{x}_1$	$\vec{x}_2$	$\vec{x}_3$	$\vec{x}_4$
+1	+1	+1	+1
+1	+1	+1	-1
+1	+1	-1	+1
+1	-1	+1	+1
-1	+1	+1	+1

Can you add an additional dichotomy?

[Safe to Skip]

Proof Intuitions

- How **"no 2 points can be shattered"** impacts the scenario with **4 points**?

$\vec{x}_1$	$\vec{x}_2$	$\vec{x}_3$	$\vec{x}_4$
+1	+1	+1	+1
+1	+1	+1	-1
+1	+1	-1	+1
+1	-1	+1	+1
-1	+1	+1	+1

$(\vec{x}_1, \vec{x}_2, \vec{x}_3)$ appear twice, with different $\vec{x}_4$

No 1 points can be shattered

$(\vec{x}_1, \vec{x}_2, \vec{x}_3)$ appear once (including one in each of the pair above)

No 2 points can be shattered

[Safe to Skip]

Proof Intuitions

- Max # dichotomies can you list on **4 points** when **no 2 points can be shattered**

No 1 point can be shattered

$\vec{x}_1$	$\vec{x}_2$	$\vec{x}_3$	$\vec{x}_4$
+1	+1	+1	+1
+1	+1	+1	-1
+1	+1	-1	+1
+1	-1	+1	+1
-1	+1	+1	+1

No 2 points can be shattered

$\boldsymbol{B(N,k)}$: max # dichotomies on N points when no k points are shattered

A recursive definition:

$$\boldsymbol{B(N,k)} \le \boldsymbol{B(N-1,k)} + \boldsymbol{B(N-1,k-1)}$$

Sauer's Lemma: $\boldsymbol{B(N,k)} \le \sum_{i=0}^{k-1}\binom{N}{i}$

Can be proved by induction

$\boldsymbol{B(N,k)} \le \sum_{i=0}^{k-1}\binom{N}{i}$ is the bound of $m_H(N)$ for H with break point k

Bounding Growth Function using Break Points

- Theorem statement:
  - If there is no break point for H, then $m_H(N) = 2^N$ for all N.
  - If k is a break point for H, i.e., if $m_H(k) < 2^k$ for some value k, then

$$m_H(N) \leq \sum_{i=0}^{k-1} \binom{N}{i}$$

- Rephrase the above theorem
  - If k is a break point for H, the following statements are true
    - $m_H(N) \leq N^{k-1} + 1$ [Can be proven using induction. See LFD Problem 2.5]
    - $m_H(N) = O(N^{k-1})$
    - $m_H(N)$ is polynomial in N
  - If d_{vc} is the VC dimension of H, then
    - $m_H(N) \leq \sum_{i=0}^{d_{vc}} \binom{N}{i}$
    - $m_H(N) \leq N^{d_{vc}} + 1$
    - $m_H(N) = O(N^{d_{vc}})$

If d_{vc} is the VC dimension of H, $d_{vc} + 1$ is a break point for H

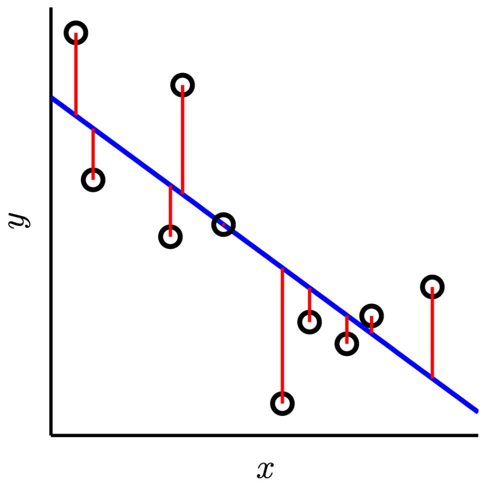
Bias-Variance Decomposition

Another theory of generalization

Real-Value Target and Squared Error

- So far, we focus on binary target function and binary error
  - Binary target function $f(\vec{x}) \in \{-1,1\}$
  - Binary error $e\big(h(\vec{x}), f(\vec{x})\big) = \mathbb{I}[h(\vec{x}) \neq f(\vec{x})]$

- Real-value functions ["**regression**"] and squared error?
  - Real-value target function $f(\vec{x}) \in \mathbb{R}$
  - Square error $e\big(h(\vec{x}), f(\vec{x})\big) = \big(h(\vec{x}) - f(\vec{x})\big)^2$



Real-Value Target and Square Error

- Real-value functions [called "**regression**"] and squared error?
  - Real-value target function $f(\vec{x}) \in \mathbb{R}$
  - Square error $e\big(h(\vec{x}), f(\vec{x})\big) = \big(h(\vec{x}) - f(\vec{x})\big)^2$

- Errors:
  - In-sample error: $E_{in}(g) = \frac{1}{N}\sum_{n=1}^{N} e(h(\vec{x}_n), f(\vec{x}_n)) = \frac{1}{N}\sum_{n=1}^{N}\big(h(\vec{x}_n) - f(\vec{x}_n)\big)^2$
  - Out-of-sample error: $E_{out}(g) = \mathbb{E}_{\vec{x}}[e\big(h(\vec{x}), f(\vec{x})\big)] = \mathbb{E}_{\vec{x}}[\big(g(\vec{x}) - f(\vec{x})\big)^2]$

- Theory of generalization: What can we say about $E_{out}(g)$?

- Note that g is learned by some algorithm on the dataset D
  - We'll make the dependency on D explicit and write it as $\boldsymbol{g}^{(\boldsymbol{D})}$ here.
  - [In VC theory, we consider the worst-case D through the definition of growth function $m_H(N)$]

Define "expected" hypothesis
$\bar{g}(\vec{x}) = \mathbb{E}_D\left[g^{(D)}(\vec{x})\right]$

- $E_{out}\left(g^{(D)}\right) = \mathbb{E}_{\vec{x}}[\left(g^{(D)}(\vec{x}) - f(\vec{x})\right)^2]$
- $\mathbb{E}_D\left[E_{out}\left(g^{(D)}\right)\right]$

$$= \mathbb{E}_D\left[\mathbb{E}_{\vec{x}}\left[\left(g^{(D)}(\vec{x}) - f(\vec{x})\right)^2\right]\right]$$

$$= \mathbb{E}_{\vec{x}}\left[\mathbb{E}_D\left[\left(g^{(D)}(\vec{x}) - \bar{g}(\vec{x}) + \bar{g}(\vec{x}) - f(\vec{x})\right)^2\right]\right]$$

$$= \mathbb{E}_{\vec{x}}\left[\mathbb{E}_D\left[\left(g^{(D)}(\vec{x}) - \bar{g}(\vec{x}) + \bar{g}(\vec{x}) - f(\vec{x})\right)^2\right]\right]$$

$$= \mathbb{E}_{\vec{x}}\left[\mathbb{E}_D\left[\left(g^{(D)}(\vec{x}) - \bar{g}(\vec{x})\right)^2 + \left(\bar{g}(\vec{x}) - f(\vec{x})\right)^2 + 2\left(g^{(D)}(\vec{x}) - \bar{g}(\vec{x})\right)\left(\bar{g}(\vec{x}) - f(\vec{x})\right)\right]\right]$$

- Note that $\mathbb{E}_D\left[\left(g^{(D)}(\vec{x}) - \bar{g}(\vec{x})\right)\left(\bar{g}(\vec{x}) - f(\vec{x})\right)\right] = \left(\bar{g}(\vec{x}) - f(\vec{x})\right)\mathbb{E}_D\left[\left(g^{(D)}(\vec{x}) - \bar{g}(\vec{x})\right)\right] = 0$

Finishing Up

- $\mathbb{E}_D\left[E_{out}\left(g^{(D)}\right)\right]$

$$= \mathbb{E}_{\vec{x}}\left[\mathbb{E}_D\left[\left(g^{(D)}(\vec{x}) - \bar{g}(\vec{x})\right)^2 + \left(\bar{g}(\vec{x}) - f(\vec{x})\right)^2\right]\right]$$

$$= \mathbb{E}_{\vec{x}}\left[\mathbb{E}_D\left[\left(g^{(D)}(\vec{x}) - \bar{g}(\vec{x})\right)^2\right]\right] + \mathbb{E}_{\vec{x}}\left[\left(\bar{g}(\vec{x}) - f(\vec{x})\right)^2\right]$$

$$= \mathbb{E}_{\vec{x}}\left[\text{Variance of } g^{(D)}(\vec{x}) + \text{Bias of } \bar{g}(\vec{x})\right]$$

$$= \text{Variance} + \text{Bias}$$

- Bias-Variance Decomposition

$\bar{g}(\vec{x}) = \mathbb{E}_D\left[g^{(D)}(\vec{x})\right]$

X: a random variable
μ: the mean of X

Variance of X:
$Var(X) = \mathbb{E}[(X-\mu)^2]$

Discussion

- $\mathbb{E}_D\left[E_{out}\left(g^{(D)}\right)\right] = \mathbb{E}_{\vec{x}}\left[\underbrace{\left(\bar{g}(\vec{x}) - f(\vec{x})\right)^2}_{\text{Bias}(\vec{x})}\right] + \mathbb{E}_{\vec{x}}\left[\underbrace{\mathbb{E}_D\left[\left(g^{(D)}(\vec{x}) - \bar{g}(\vec{x})\right)^2\right]}_{\text{Var}(\vec{x})}\right]$

- This is a **conceptual** decomposition
  - Both $\bar{g}$ and f are unknown
  - We can't really calculate bias and variance in practice

- However, it provides a conceptual guideline in decreasing E_{out}

Example of Bias-Variance Decomposition

- Fitting a sine function
  - $f(x) = \sin(\pi x)$
  - x is drawn uniformly at random from $[0,2]$

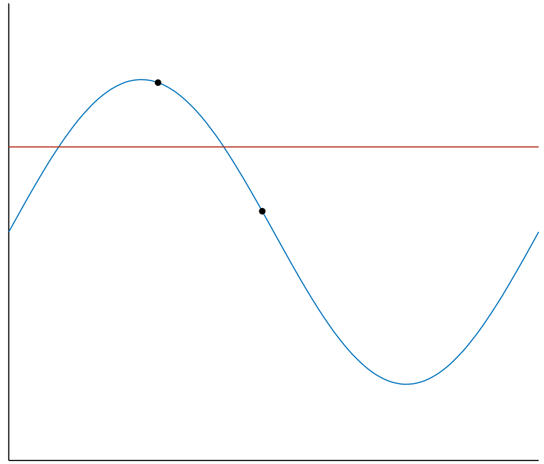
- Two hypothesis set
  - H_0: $h(x) = b$
  - H_1: $h(x) = ax + b$

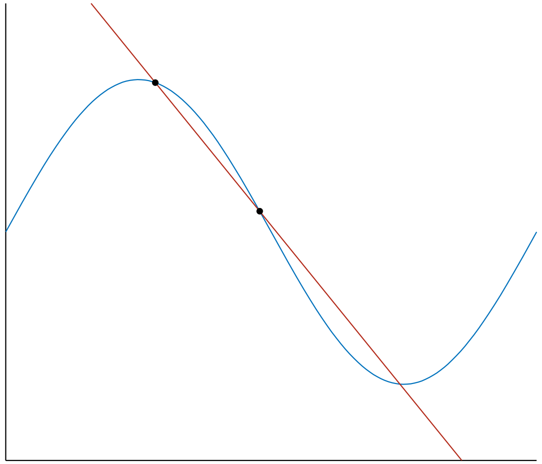
- Assume our algorithm finds g with minimum in-sample error

Example of Bias-Variance Decomposition

$H_0: h(x) = b$ $\qquad$ $H_1: h(x) = ax + b$

N=2





$$\mathbb{E}_D\left[E_{out}\left(g^{(D)}\right)\right] = \mathbb{E}_{\vec{x}}\left[\overbrace{\left(\bar{g}(\vec{x}) - f(\vec{x})\right)^2}^{\text{Bias}(\vec{x})}\right] + \mathbb{E}_{\vec{x}}\left[\overbrace{\mathbb{E}_D\left[\left(g^{(D)}(\vec{x}) - \bar{g}(\vec{x})\right)^2\right]}^{\text{Var}(\vec{x})}\right]$$

Discussion:

If $N = 2$, would you choose H_0 or H_1? Why?

If $N = 5$, would you choose H_0 or H_1? Why?

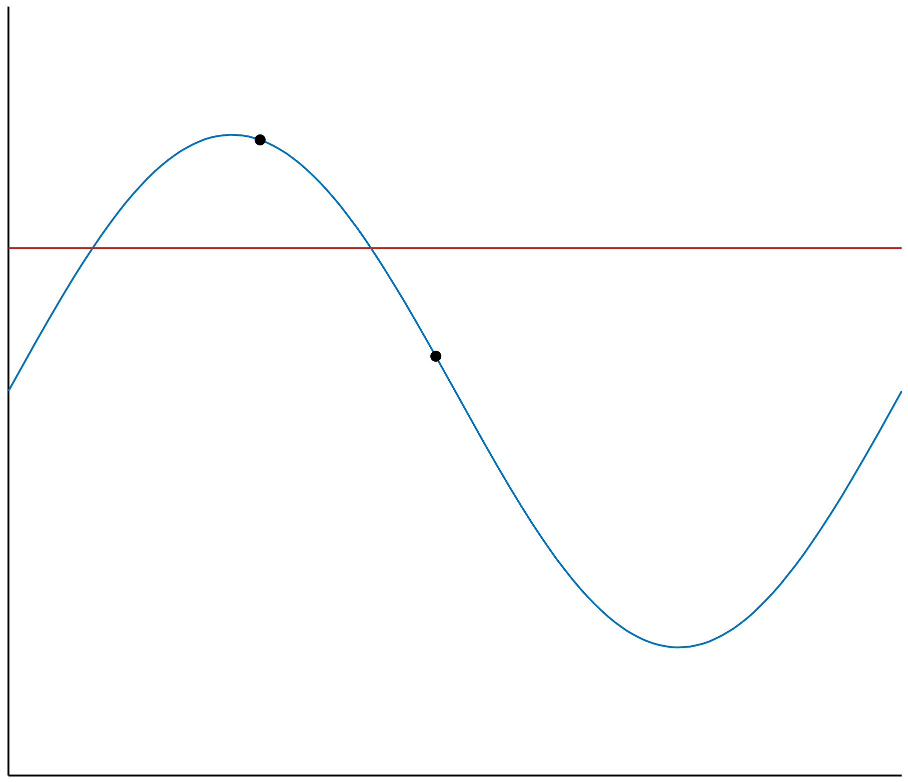
What's the change of biases/variances for H_0/ H_1 from $N = 2$ to $N = 5$.

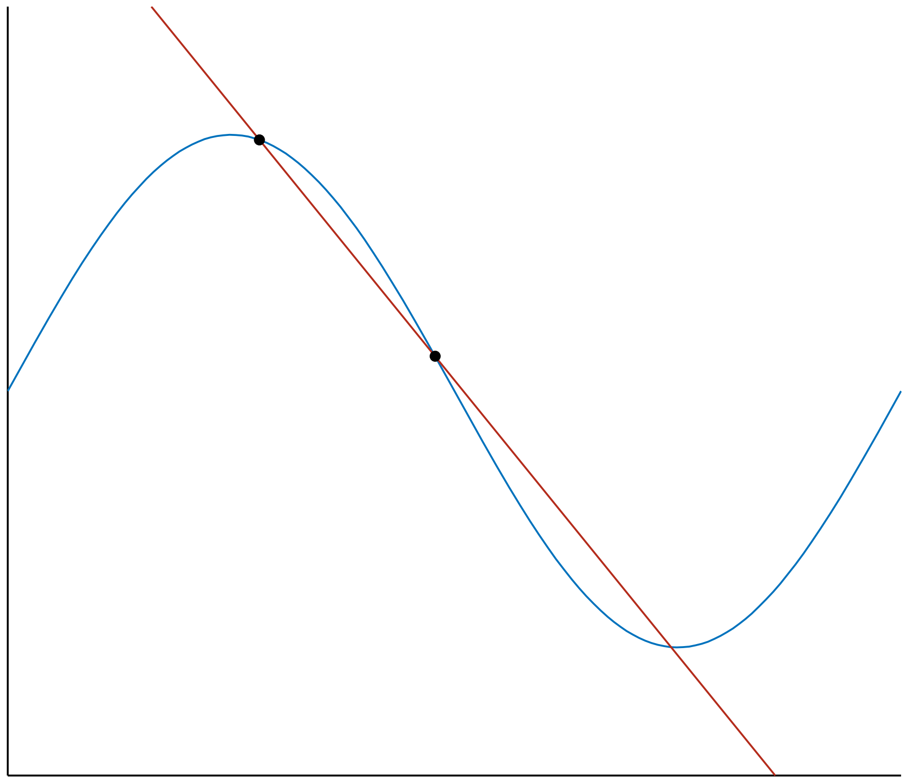
Example of Bias-Variance Decomposition

$H_0: h(x) = b$

$H_1: h(x) = ax + b$

N=2



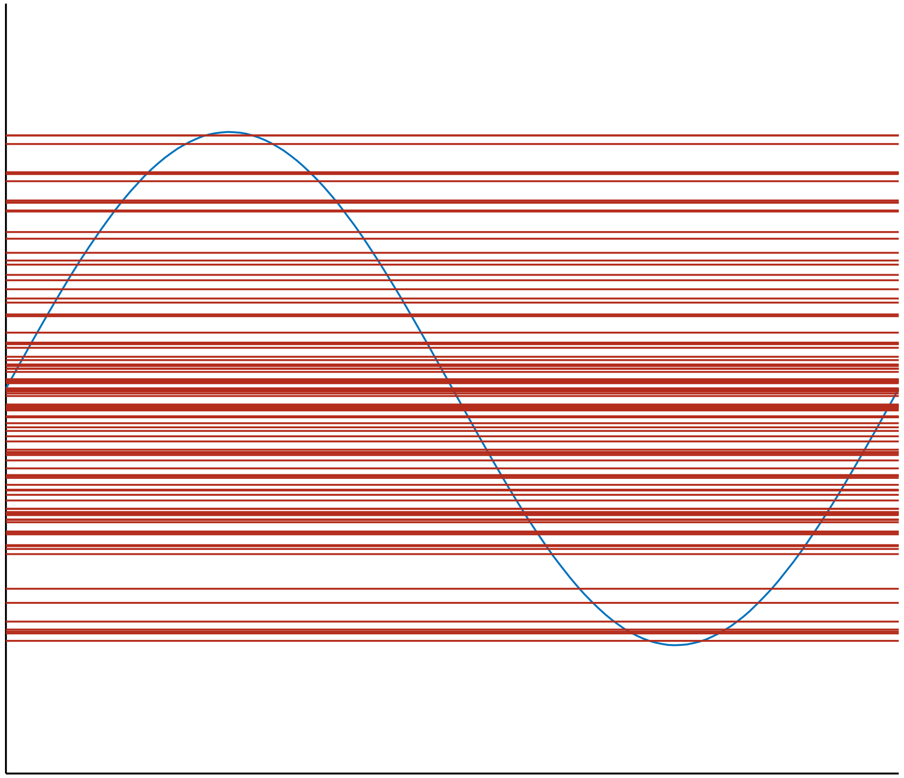


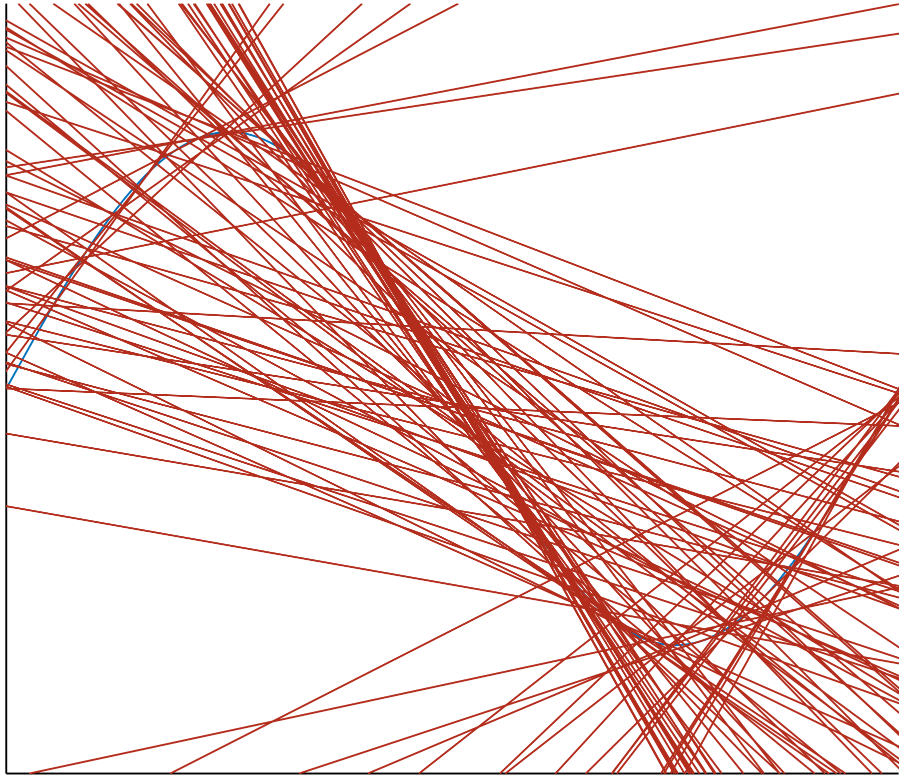
Example of Bias-Variance Decomposition

$H_0: h(x) = b$

$H_1: h(x) = ax + b$

N=2



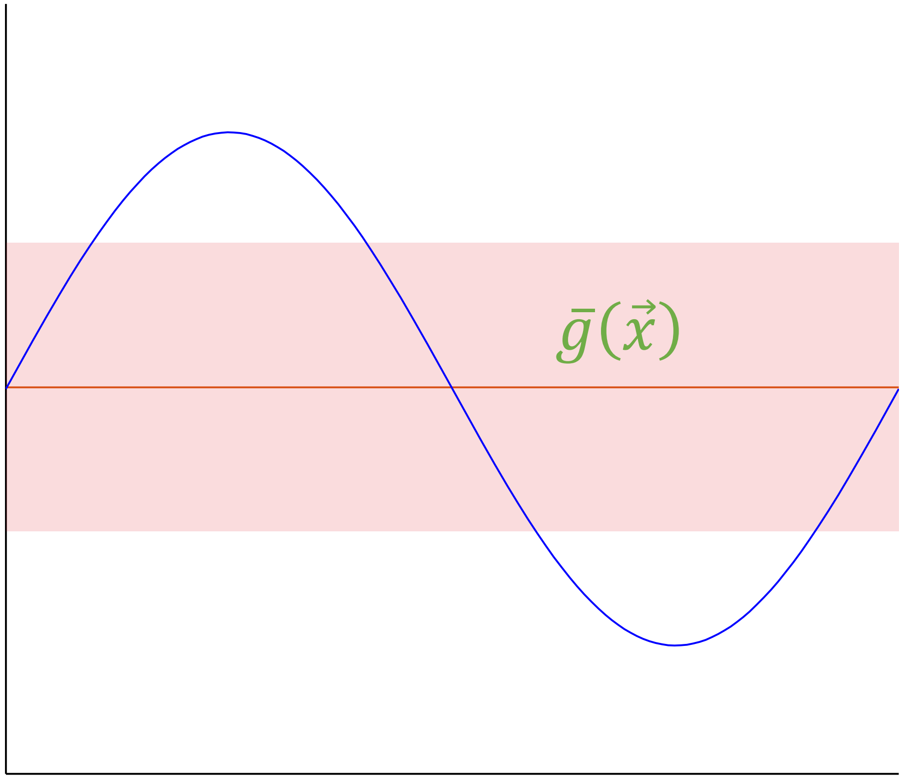


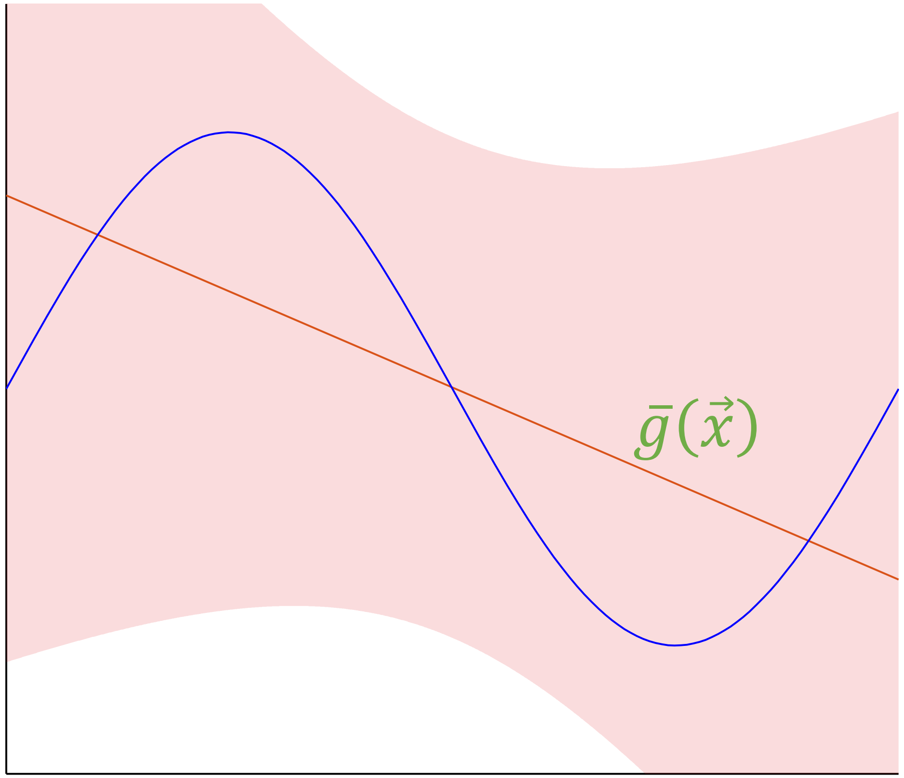
Example of Bias-Variance Decomposition

$H_0: h(x) = b$

$H_1: h(x) = ax + b$

N=2





Bias of $\bar{g}(\vec{x}) \approx 0.50$

Variance of $g_{\mathcal{D}}(\vec{x}) \approx 0.25$

$\mathbb{E}_{\mathcal{D}}[E_{out}(g_{\mathcal{D}})] \approx 0.75$

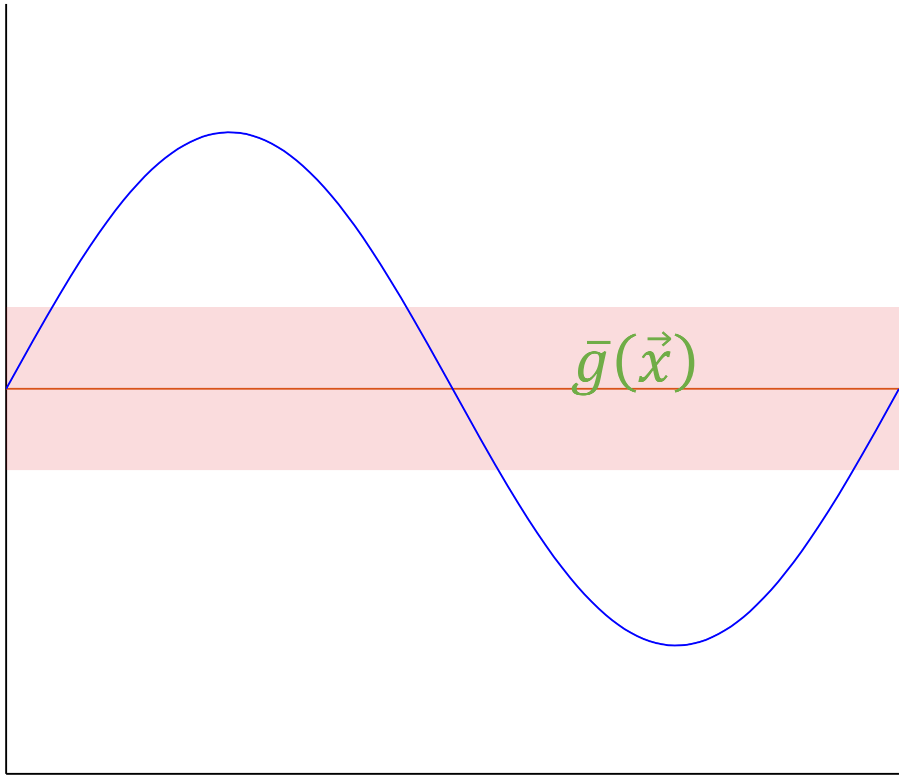
Bias of $\bar{g}(\vec{x}) \approx 0.21$

Variance of $g_{\mathcal{D}}(\vec{x}) \approx 1.74$

$\mathbb{E}_{\mathcal{D}}[E_{out}(g_{\mathcal{D}})] \approx 1.95$

What if we increase N to 5?

H_0: $h(x) = b$

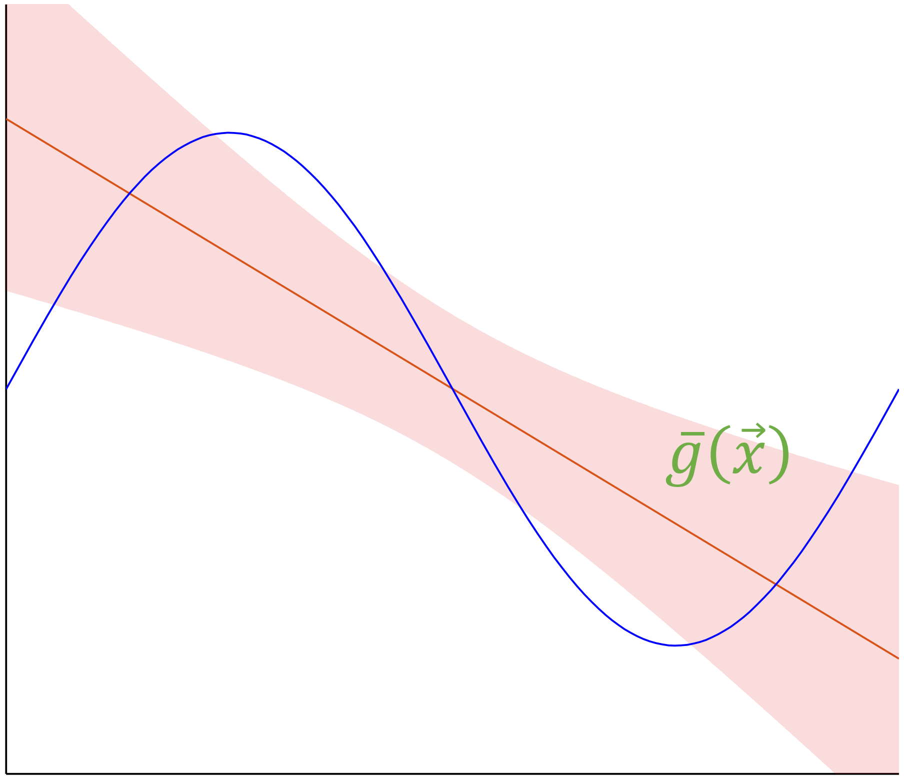


Bias of $\bar{g}(\vec{x}) \approx 0.50$

Variance of $g_{\mathcal{D}}(\vec{x}) \approx 0.10$

$\mathbb{E}_{\mathcal{D}}[E_{out}(g_{\mathcal{D}})] \approx 0.60$

H_1: $h(x) = ax + b$



Bias of $\bar{g}(\vec{x}) \approx 0.21$

Variance of $g_{\mathcal{D}}(\vec{x}) \approx 0.21$

$\mathbb{E}_{\mathcal{D}}[E_{out}(g_{\mathcal{D}})] \approx 0.42$

Discussion

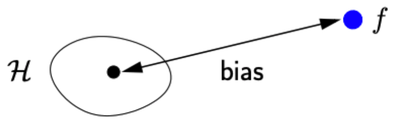
- $\mathbb{E}_D\left[E_{out}\left(g^{(D)}\right)\right] = \mathbb{E}_{\vec{x}}\left[\underbrace{\left(\bar{g}(\vec{x}) - f(\vec{x})\right)^2}_{\mathrm{Bias}(\vec{x})}\right] + \mathbb{E}_{\vec{x}}\left[\underbrace{\mathbb{E}_D\left[\left(g^{(D)}(\vec{x}) - \bar{g}(\vec{x})\right)^2\right]}_{\mathrm{Var}(\vec{x})}\right]$

- Increasing the number of data points N
  - Biases roughly stay the same
  - Variances decrease
  - Expected E_{out} decreases

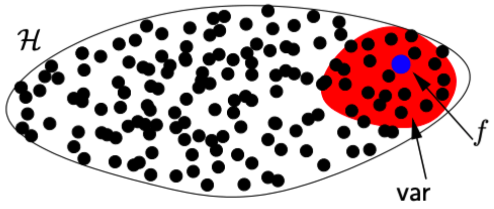
Discussion

- $\mathbb{E}_D[E_{out}(g^{(D)})] = \mathbb{E}_{\vec{x}}\Big[\underbrace{\big(\bar{g}(\vec{x}) - f(\vec{x})\big)^2}_{\text{Bias}(\vec{x})}\Big] + \mathbb{E}_{\vec{x}}\Big[\underbrace{\mathbb{E}_D\Big[\big(g^{(D)}(\vec{x}) - \bar{g}(\vec{x})\big)^2\Big]}_{\text{Var}(\vec{x})}\Big]$

- Increasing the complexity of H
  - Bias goes down (more likely to approximate f)
  - Variance goes up (The stability of $g^{(D)}$ is worse)



Very small model



Very large model

Discussion

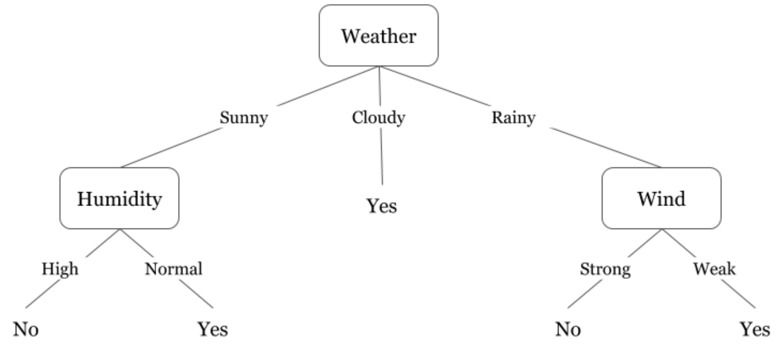
- $$\mathbb{E}_D\left[E_{out}\left(g^{(D)}\right)\right] = \mathbb{E}_{\vec{x}}\left[\underbrace{\left(\bar{g}(\vec{x}) - f(\vec{x})\right)^2}_{\text{Bias}(\vec{x})}\right] + \mathbb{E}_{\vec{x}}\left[\underbrace{\mathbb{E}_D\left[\left(g^{(D)}(\vec{x}) - \bar{g}(\vec{x})\right)^2\right]}_{\text{Var}(\vec{x})}\right]$$

- This is a **conceptual** decomposition
  - Both $\bar{g}$ and f are unknown
  - We can’t really calculate bias and variance for practical problems

- However, it provides a conceptual guidelines in decreasing E_{out}

Example

- Will talk about this in details in the 2nd half of the semester
- Decision tree
  - A low bias but high variance hypothesis set
  - Practical performance is not ideal



- Random forest
  - Trying to reduce the variance while not sacrificing bias
  - Idea: Generate many trees randomly and average them

Two Theories of Generalization

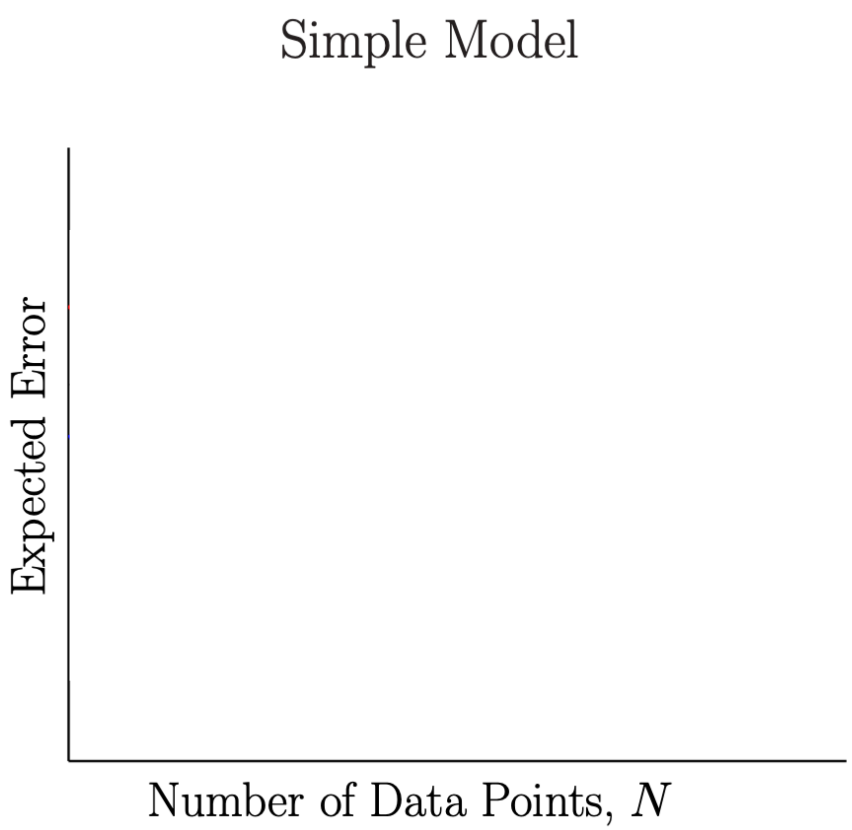
- VC Generalization Bound

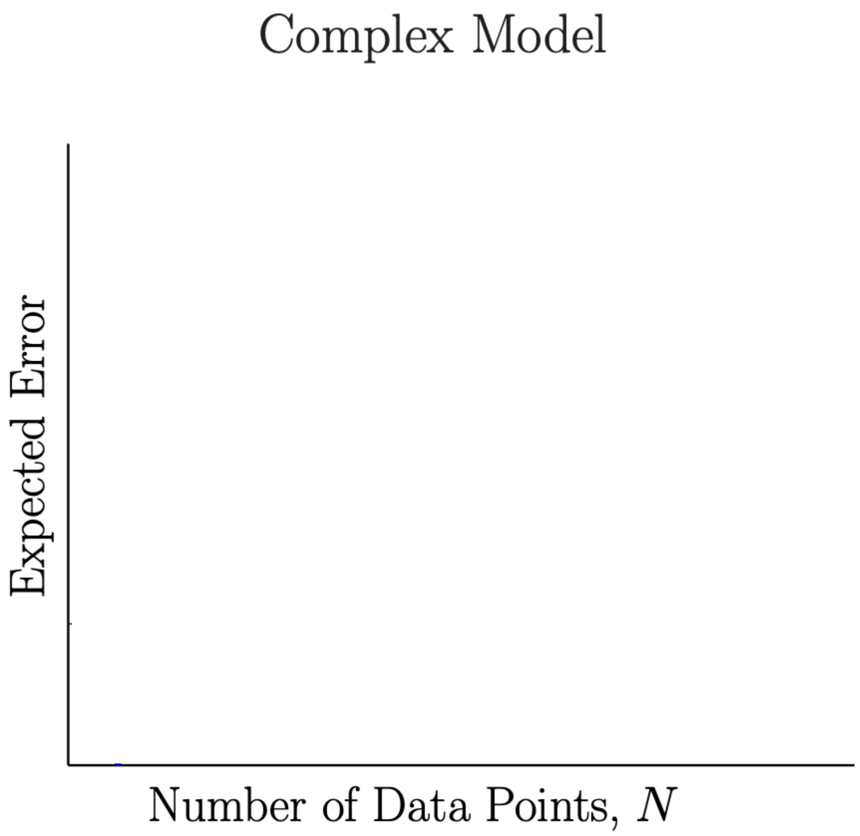
$$E_{out}(g) \leq E_{in}(g) + O\left(\sqrt{d_{vc}\frac{\ln N}{N}}\right)$$

- Bias-Variance Tradeoff

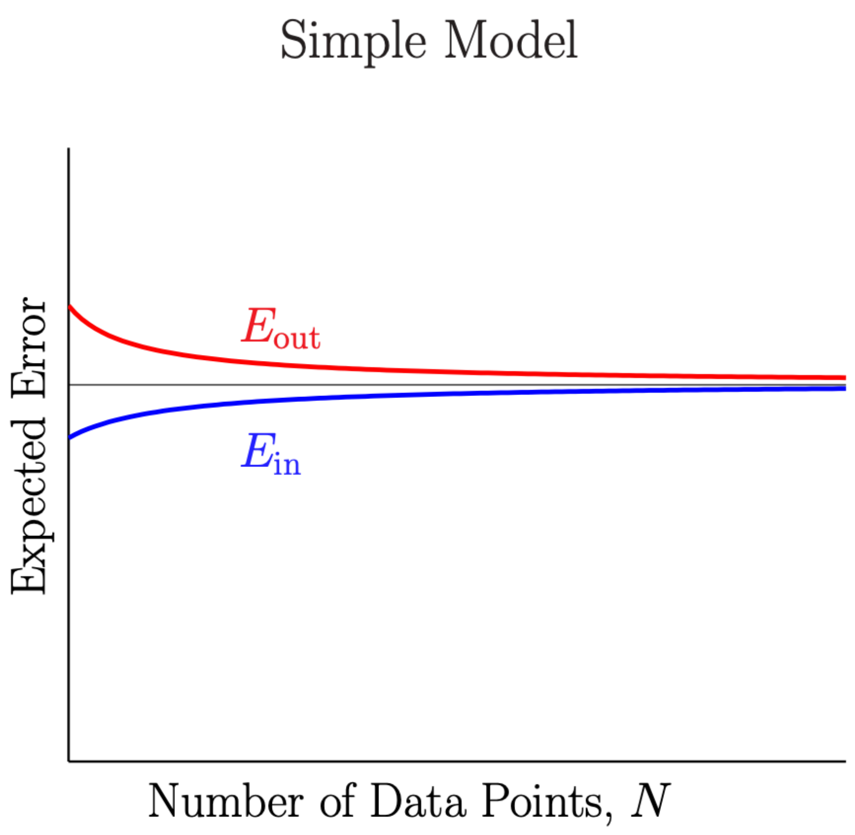
$$\mathbb{E}_D\left[E_{out}\left(g^{(D)}\right)\right] = \mathbb{E}_{\vec{x}}\left[\left(\bar{g}(\vec{x}) - f(\vec{x})\right)^2\right] + \mathbb{E}_{\vec{x}}\left[\mathbb{E}_D\left[\left(g^{(D)}(\vec{x}) - \bar{g}(\vec{x})\right)^2\right]\right]$$

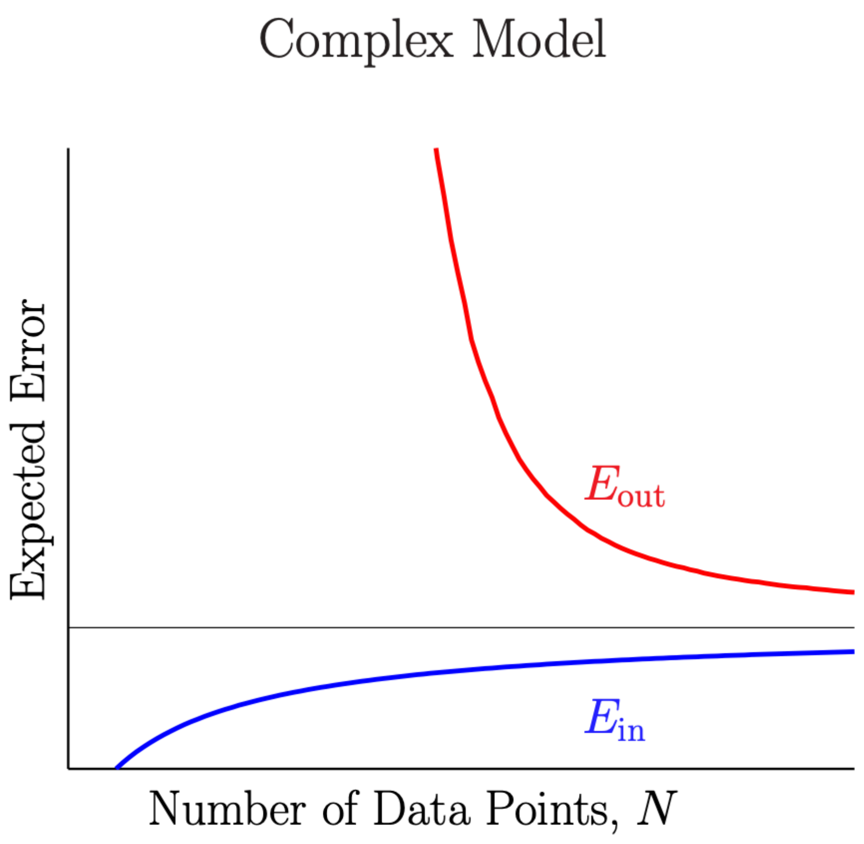
Learning Curves





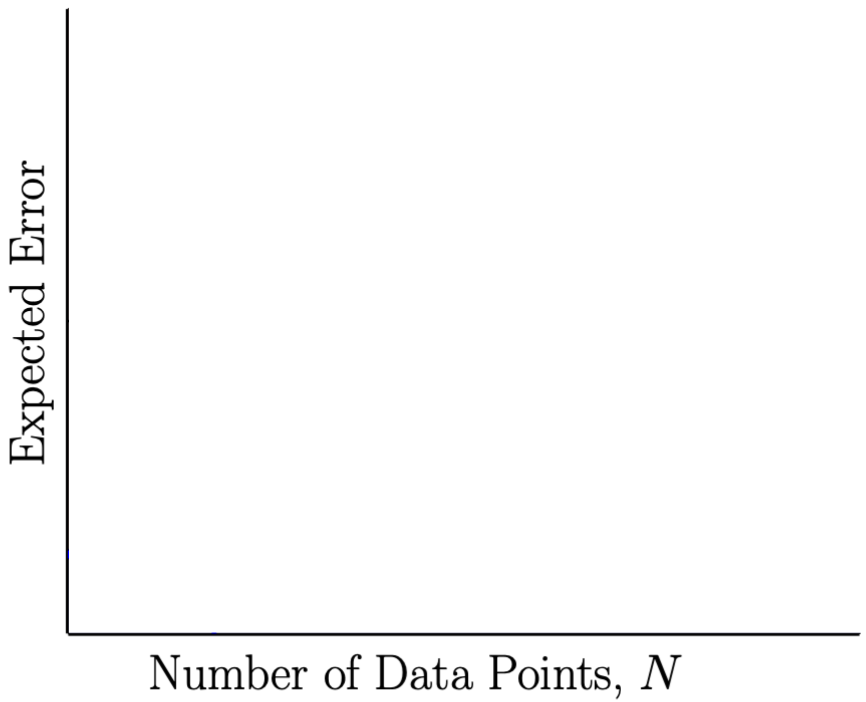
Learning Curves



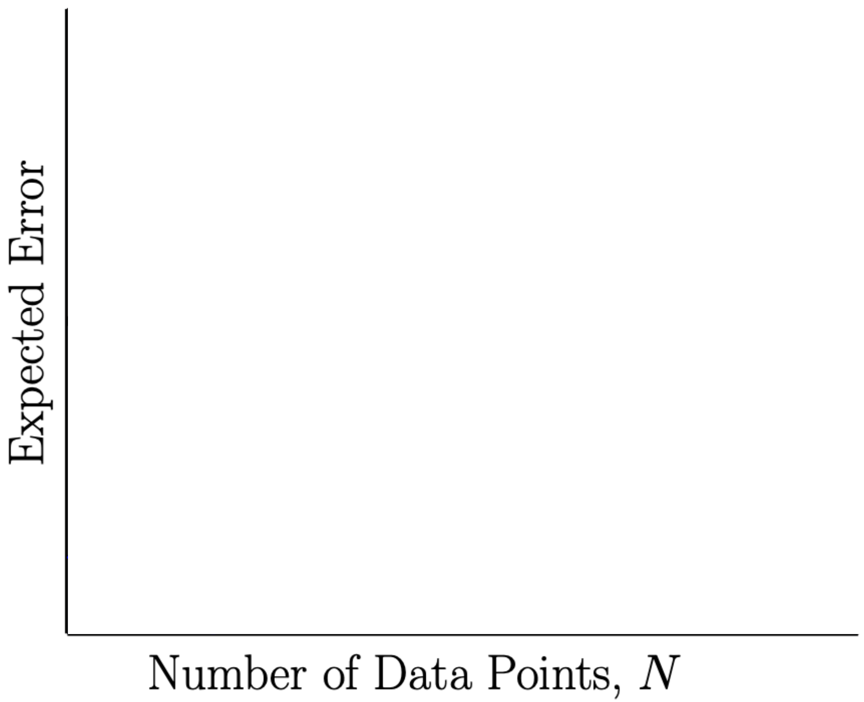


Learning Curves

VC Analysis



Bias-Variance Analysis



Learning Curves