ResearchGate

See discussions, stats, and author profiles for this publication at: https://www.researchgate.net/publication/333163734

Lecture 8: Bias-Variance Trade-off

Presentation · May 2019

CITATIONS
0

READS
198

1 author:

Alaa Tharwat
Fachhochschule Bielefeld
120 PUBLICATIONS **6,195** CITATIONS

SEE PROFILE

All content following this page was uploaded by Alaa Tharwat on 17 May 2019.

The user has requested enhancement of the downloaded file.

Lecture 8: Bias-Variance Tradeoff

Alaa Tharwat

Lecture 8: Bias-Variance Tradeoff

- Review of Lecture 7
- Bias and Variance
- Learning curves

Lecture 8: Bias-Variance Tradeoff

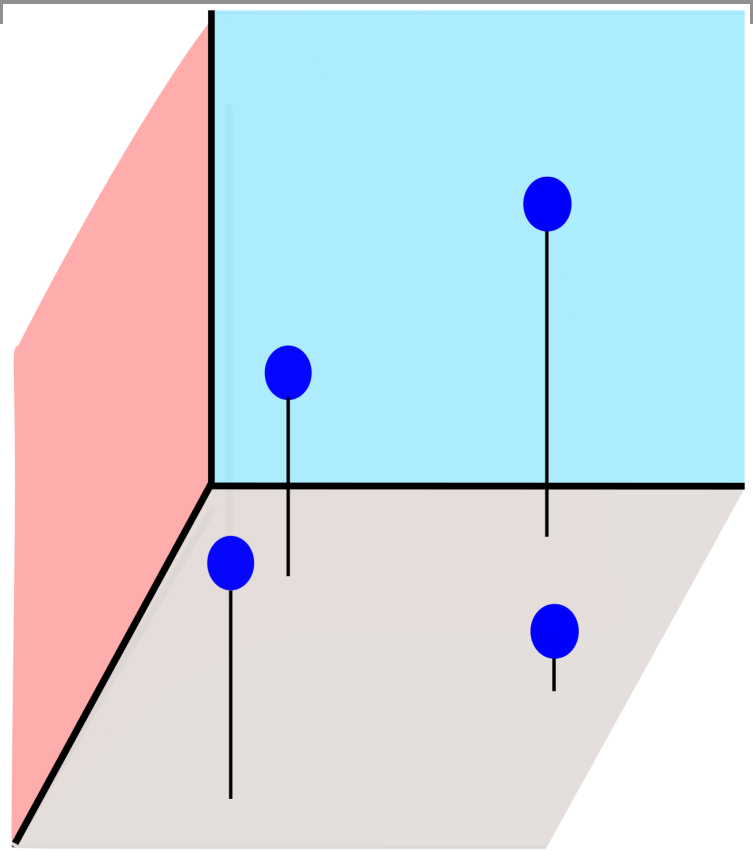
- Review of Lecture 7
- Bias and Variance
- Learning curves

- If $N \leq d_{VC}(H) \Rightarrow H$ can shatter N points
- If $k > d_{VC}(H) \Rightarrow k$ is a break point for H

$$m_H(N) \leq \sum_{i=0}^{d_{VC}} \binom{N}{i}$$

- $d_{VC} = d + 1$
- d_{VC} measures the effective number of parameters

$$E_{out} \leq E_{in} + \Omega(N, H, \delta)$$

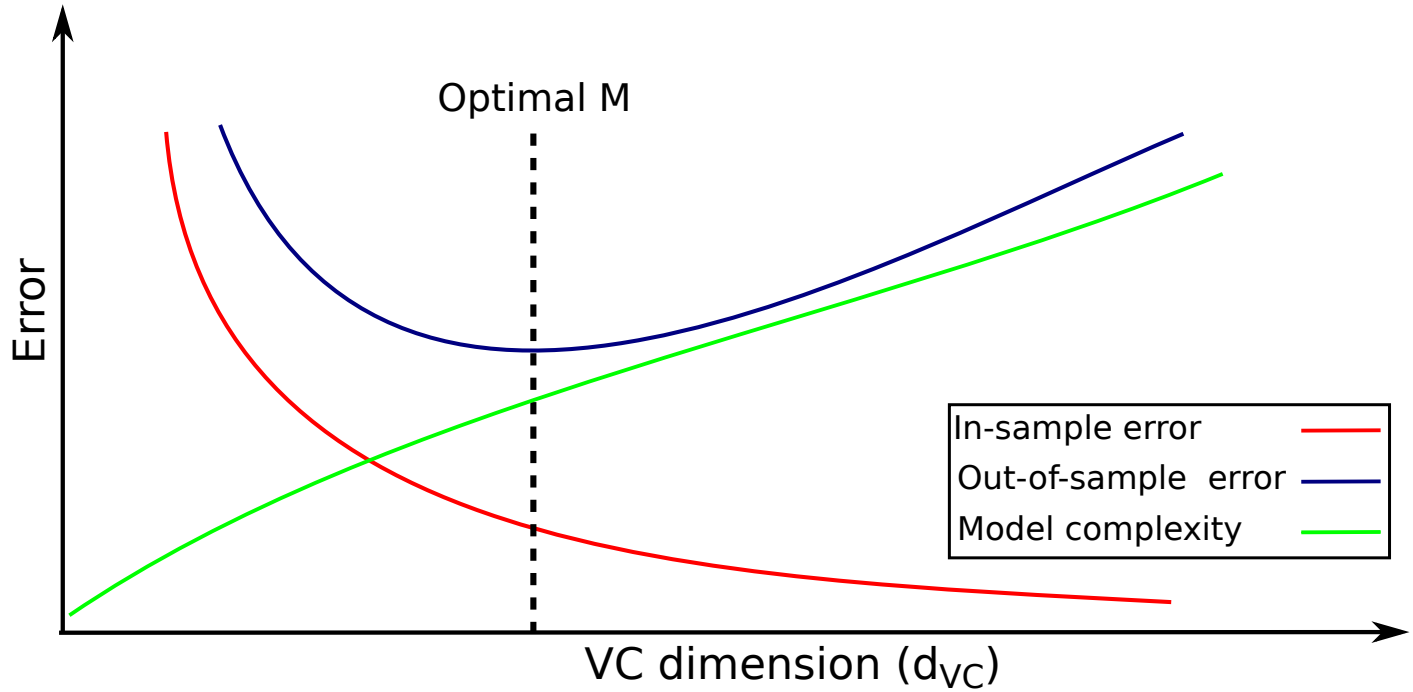


Lecture 8: Bias-Variance Tradeoff

- Review of Lecture 7
- Bias and Variance
- Learning curves

- Small E_{out} means good approximation for the target function f out of sample
- Model complexity $\uparrow \Rightarrow E_{in} \downarrow$ and hence better chance for approximating f
- Model complexity $\downarrow \Rightarrow E_{out}$ becomes closer to E_{in} and hence better chance of generalizing out of sample

$$E_{out} \leq E_{in} + \Omega(N, H, \delta)$$



- Assume we have noisy targets $y(x) = f(x) + \epsilon$, where ϵ is the noise with zero mean ($E[\epsilon] = 0$) and the variance of the noise is (σ)
- The out-of-sample error (E_{out}) is calculated using squared error

$$E_{out}(g^{(D)}) = E_x[(g^{(D)}(x) - y(x))^2]$$

$$\begin{aligned}
&E_D[E_{out}(g^{(D)})] = E_D[E_x(g^{(D)}(x) - y(x))^2] \\
&= E_D[E_x[(g^{(D)}(x))^2] - 2E_x[g^{(D)}(x)]E_x[y(x)] + E_x[y(x)^2]] \\
&= E_x[E_D[(g^{(D)}(x))^2]] - 2E_x[E_D[g^{(D)}(x)]E_D[y(x)]] + E_x[E_D[y(x)^2]] \\
&= \underbrace{E_x[E_D[(g^{(D)}(x))^2] - \bar{g}(x)^2}\, + \bar{g}(x)^2 - 2\bar{g}(x)E_D[y(x)] + E_D[y(x)^2]
\end{aligned}$$

where

$$\begin{aligned}
&E_D[(g^{(D)}(x))^2] - \bar{g}(x)^2 \\
&= E_D[(g^{(D)}(x))^2] - 2\bar{g}(x)^2 + \bar{g}(x)^2 \\
&= E_D[(g^{(D)}(x))^2 - 2g^{(D)}(x)\bar{g}(x) + \bar{g}(x)^2] \\
&= E_D[(g^{(D)}(x) - \bar{g}(x))^2] \qquad (1)
\end{aligned}$$

$$E_x[E_D[(g^{(D)}(x))^2] - \bar{g}(x)^2 + \underbrace{\bar{g}(x)^2 - 2\bar{g}(x)E_D[y(x)] + E_D[y(x)^2]}$$

$$\bar{g}(x)^2 - 2\bar{g}(x)E_D[y(x)] + E_D[y(x)^2] =$$
$$\bar{g}(x)^2 - 2\bar{g}(x)E_D[f(x) + \epsilon] + E_D[(f(x) + \epsilon)^2] =$$
$$\bar{g}(x)^2 - 2\bar{g}(x)f(x) - 2\bar{g}(x)\cancelto{0}{E_D[\epsilon]} + \cancelto{f(x)^2}{E_D[f(x)^2]}$$
$$+\cancelto{0}{E_D[f(x)\epsilon]} + E_D[\epsilon^2] =$$
$$\bar{g}(x)^2 - 2\bar{g}(x)f(x) + f(x)^2 + E_D[\epsilon^2] =$$
$$(\bar{g}(x) - f(x))^2 + E_D[\epsilon^2] \quad (2)$$

where

- $\bar{g}(x) = E_D[g^{(D)}(x)]$ represents the average function. This can be estimated by generating many datasets $D_1, D_2, \ldots, D_K$ and apply the learning algorithm on each dataset, and the generated final hypotheses are $g_1, g_2, \ldots, g_K$ and hence the average function can be estimated as follows, $\bar{g}(x) \approx \frac{1}{K} \sum_{k=1}^{K} g_k(x)$.
- $E_D[f(x)^2] = f(x)^2$ because f is deterministic.
- $E_D[\epsilon] = 0$ as mentioned before.
- $E_D[f(x)\epsilon] = E_D[f(x)]E_D[\epsilon] = 0$ because $E_D[\epsilon] = 0$.

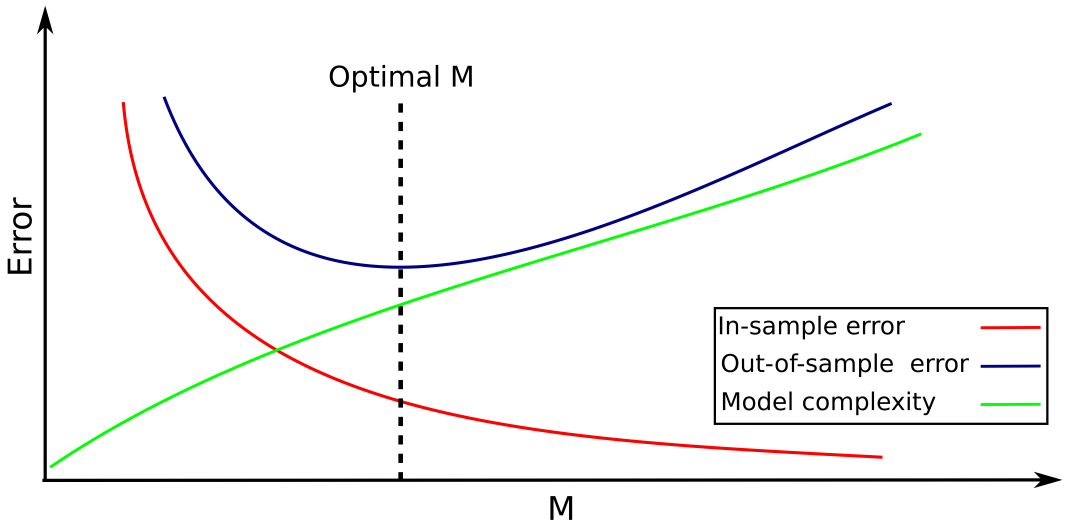
From equations (1 and 2)

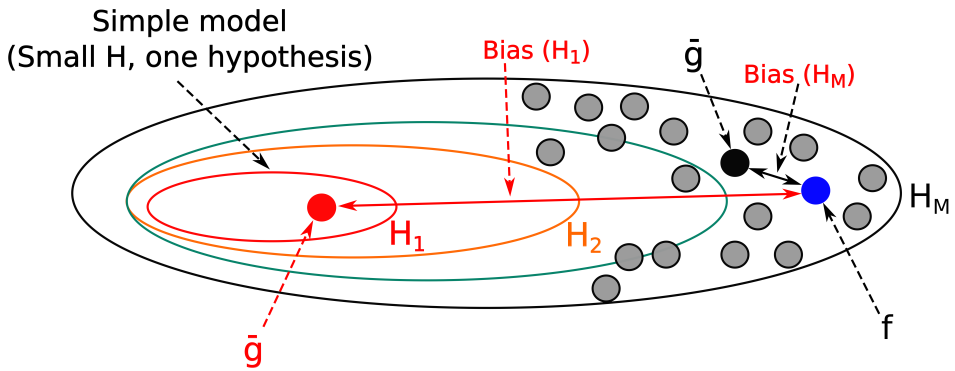
$$\begin{aligned} E_D[E_{out}(g^{(D)})] &= E_D[(g^{(D)}(x) - \bar{g}(x))^2] + (\bar{g}(x) - f(x))^2 + E_D[\epsilon^2] \\ &= \text{Variance} + \text{Bias} + \sigma^2 \end{aligned}$$

$$E_D[E_{out}(g^{(D)})] = E_D[(g^{(D)}(x) - \bar{g}(x))^2] + (\bar{g}(x) - f(x))^2 + E_D[\epsilon^2]$$
$$= \text{Variance} + \text{Bias} + \sigma^2$$

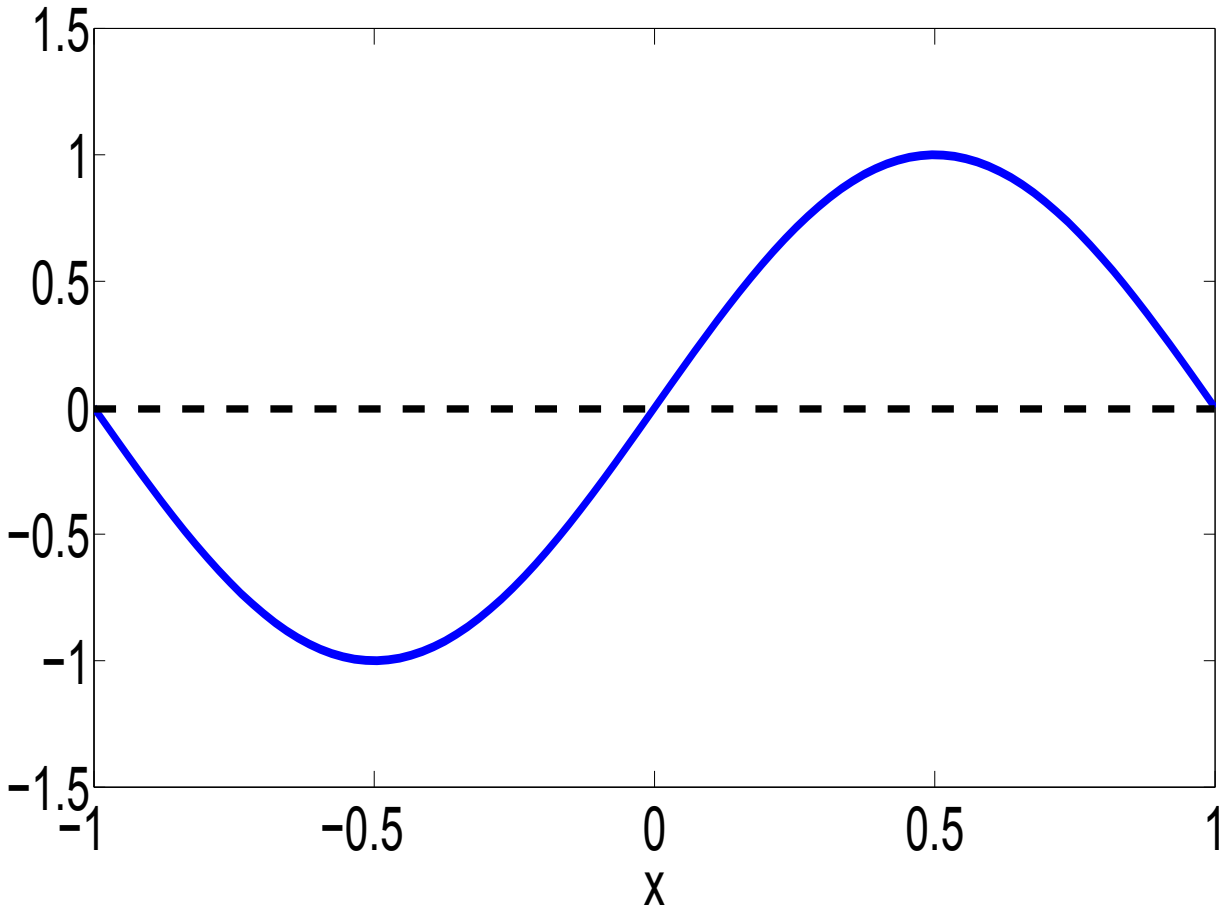
- The first term $(g^{(D)}(x) - \bar{g}(x))^2$ indicates how far your hypotheses set from the best possible you can get (**Variance**)
- The second term $(\bar{g}(x) - f(x))^2$ is the **Bias** (i.e. biased from the target function) and it represents how far the average/best model from the target function. Your Bias hypotheses set is biased away from the target function
- The term σ^2 represents the variance of noise ($\text{Var}[\epsilon] = \sigma^2$). Thus, this term cannot be minimized and hence it is called *irreducible error*, and it is independent of the classification or regression model

- With a simple model, the hypotheses set is small (maybe one hypothesis) and hence no variance but the bias is high
- With a complex model, the hypotheses set is large and hence the variance between the best hypothesis ($\bar{g}$) and all the other hypotheses is large, but the bias is small



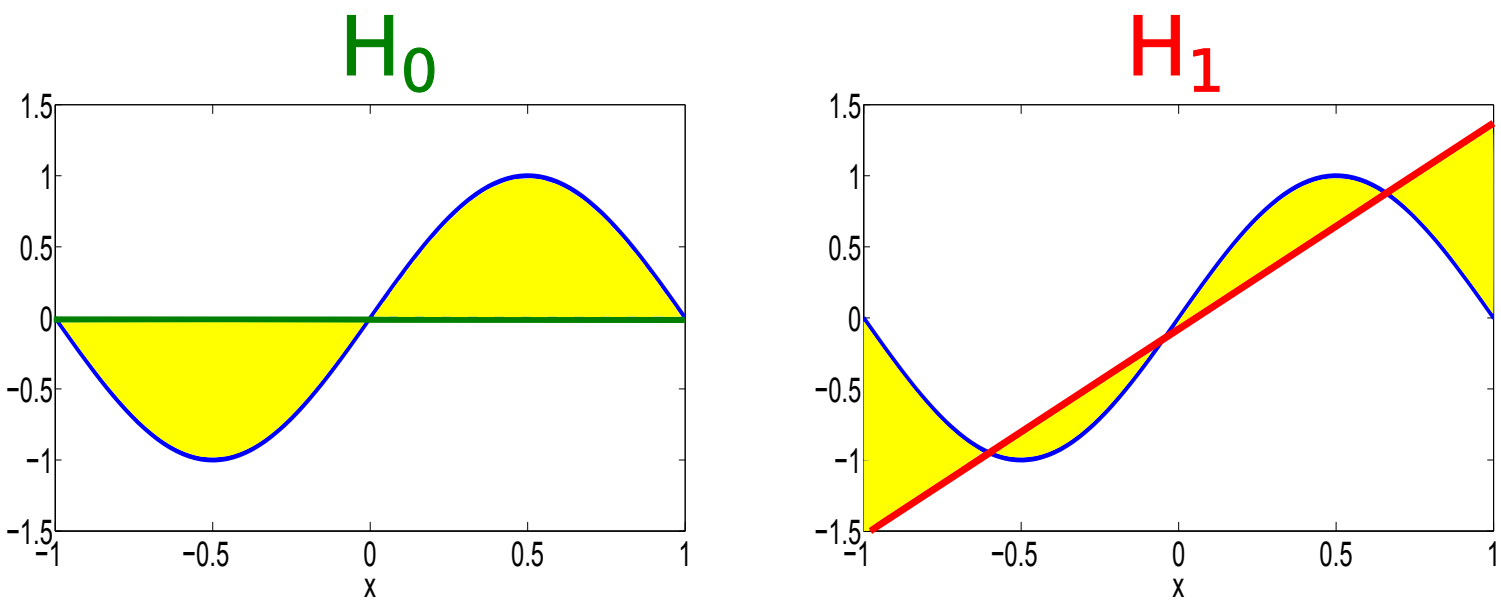


- Given a target function $f(x) = sin(\pi x)$
- Given two training samples $(N = 2)$
- Assume we have two simple models:
  - Constant model $H_0 : h(x) = b$
  - Linear model $H_1 : h(x) = ax + b$



Approximation: H_0 vs. H_1

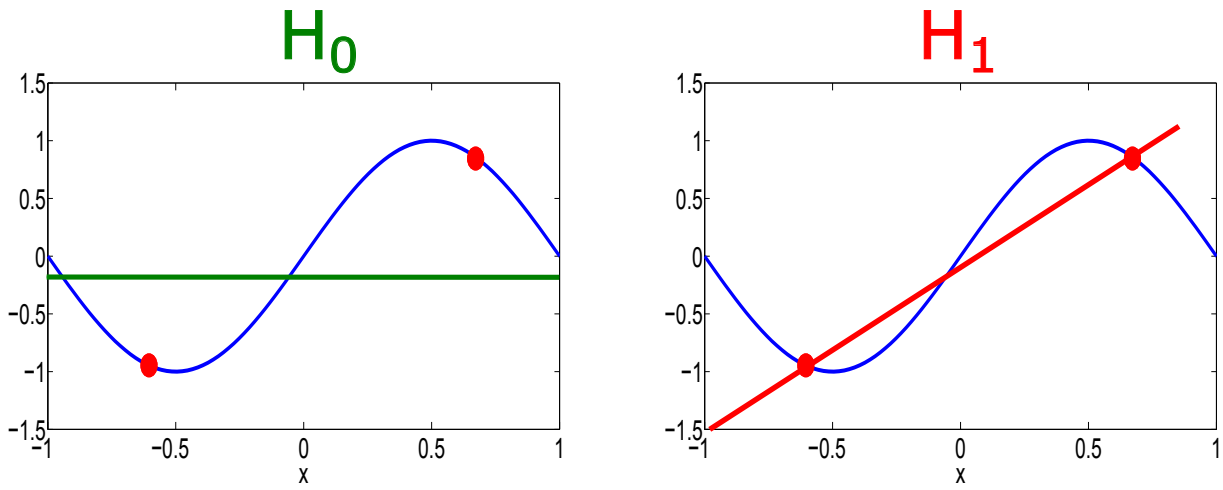
- The line in the figure below is the best model that fits f



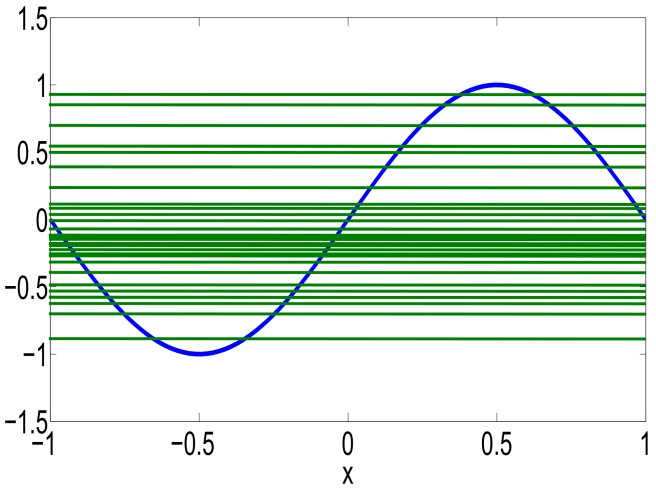
- $E_{out}(H_0) = 0.5$
- $E_{out}(H_1) = 0.2$

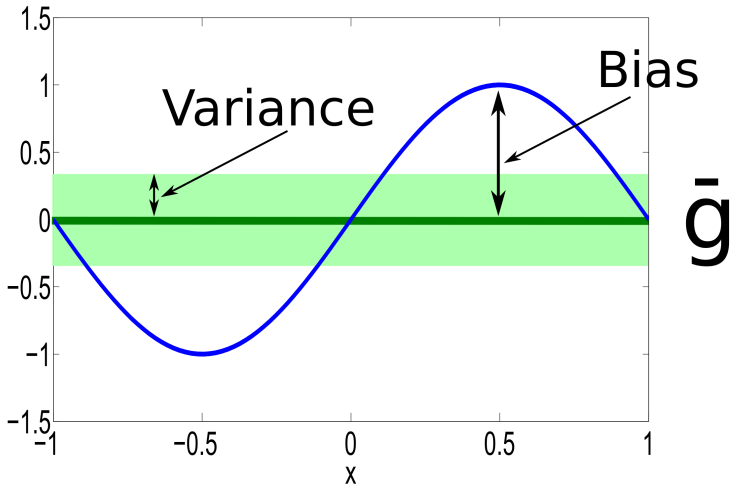
Learning: H_0 vs. H_1

- The line in the figure below is the best model that fits f given only two points/samples

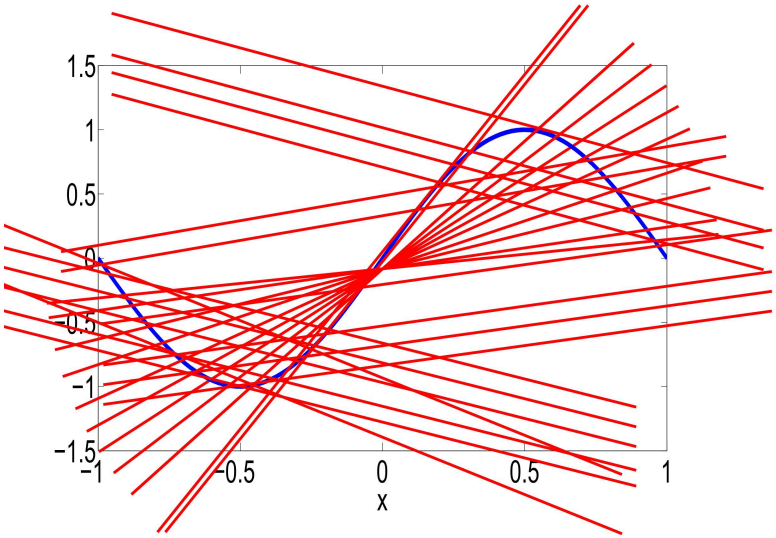


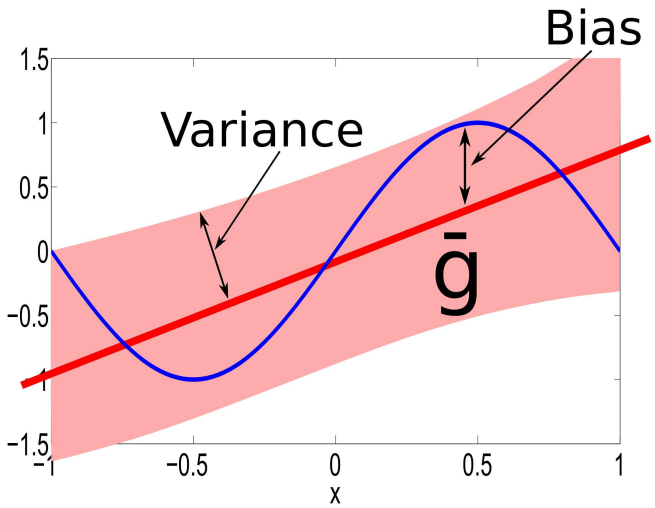
Using H_0 model:

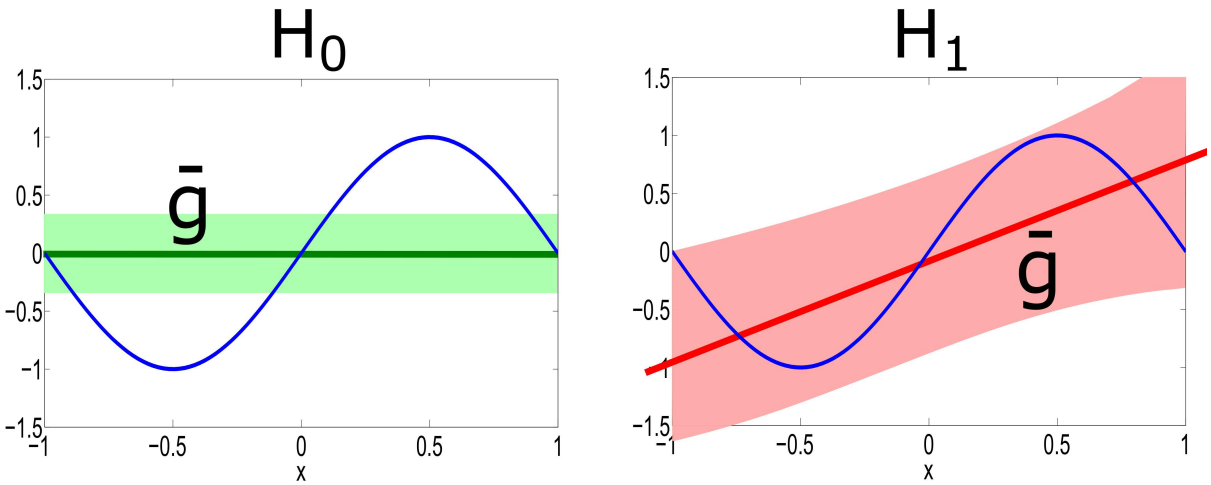




Using H_1 model:





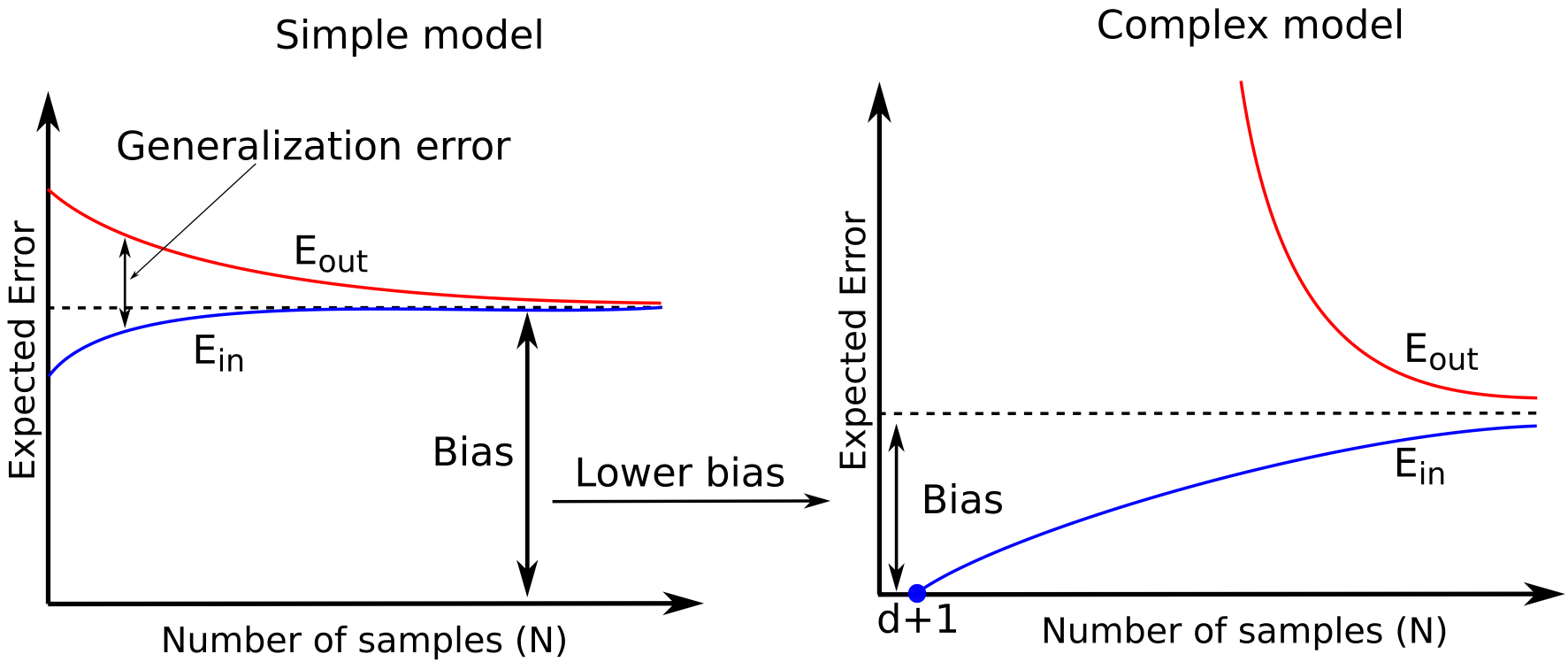


- For H_0: bias=0.5, variance=0.25, and $E_{out} = 0.75$
- For H_1: bias=0.21, variance=1.69, and $E_{out} = 1.9$
- Hence, we need to match the model complexity to the data resources (i.e. the dataset) we have, not to the target complexity (because we don't know the target function or its complexity)
- What do you think when we increase the number of samples?
  - Answer: this decreases the variance of both simple and complex models. Hence, $E_{out}(H_1) < E_{out}(H_0)$

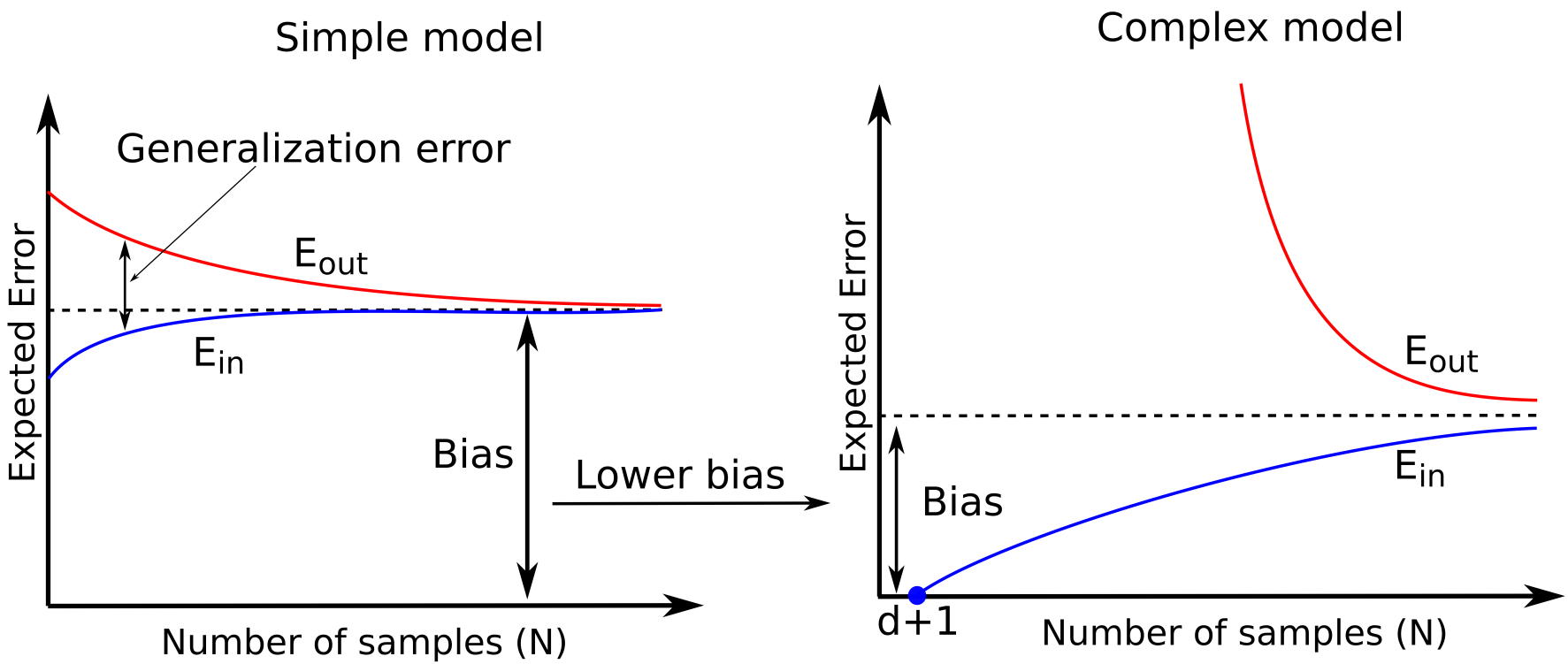
Lecture 8: Bias-Variance Tradeoff

- Review of Lecture 7
- Bias and Variance
- Learning curves

- What is the influence of N on $E_{in} = E_D[E_{in}(g^{(D)})]$ and $E_{out} = E_D[E_{out}(g^{(D)})]$, with simple and complex models



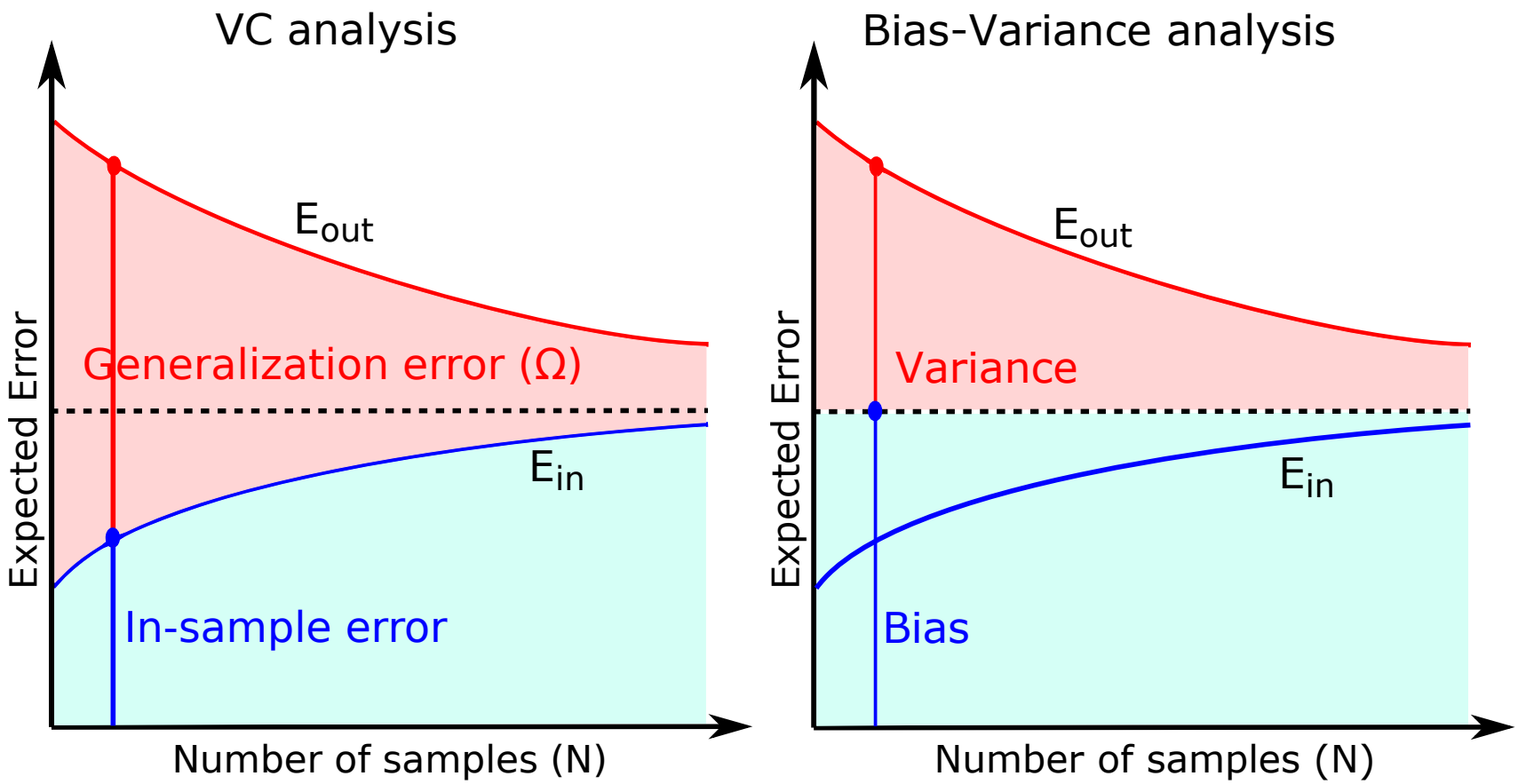
- In a simple model, the bias is high. Due to the noise, increasing N increases E_{in} because the model is simple and cannot fit all samples, but E_{out} decreases, and the gap between E_{in} and E_{out} becomes smaller



- On the contrary, with a complex model the bias is small.
- E_{in} is small (approximately zero) when the VC-dimension (degrees of freedom) of the model can shatter the points perfectly, and with a large N may be the degrees of freedom is not sufficient for shattering these points and hence increase E_{in} and reduce E_{out}

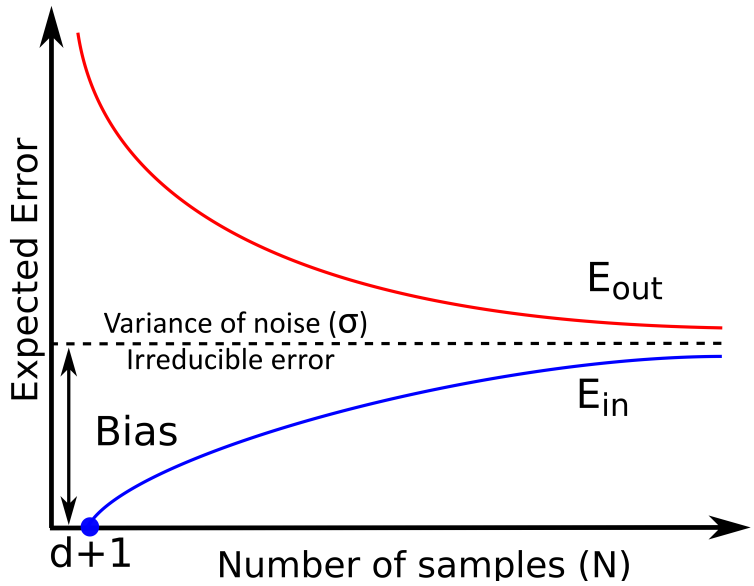
- Do you remember 2D perceptron?
- The VC dimension of the 2D perceptron model is 3, and hence the model can shatter up to 3 samples perfectly
- By increasing the number of samples more than three, the model cannot shatter the data and hence we get many errors (E_{in})

- What is the influence of N on E_{in} and E_{out}, with simple and complex models



Assume we have noisy targets $y = \mathbf{w}^{*T}\mathbf{x} +$ noise, give datasets D. Remember, in linear regression, $\mathbf{w} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T y$. $E_{in} = \mathbf{wX} - y$ and $E_{out} = \mathbf{wX} - y'$, where y' represents the outputs for the same inputs ($\mathbf{x}$) but with different noises

- Best approximation error $= \sigma^2$
- Expected in-sample error $= \sigma^2(1 - \frac{d+1}{N})$
- Expected out-of-sample error $= \sigma^2(1 + \frac{d+1}{N})$
- Expected generalization error $= 2\sigma^2(\frac{d+1}{N})$



View publication stats