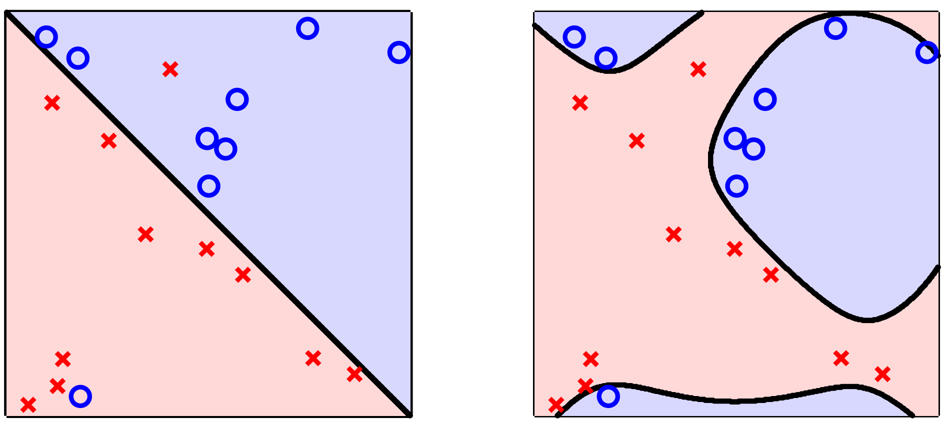
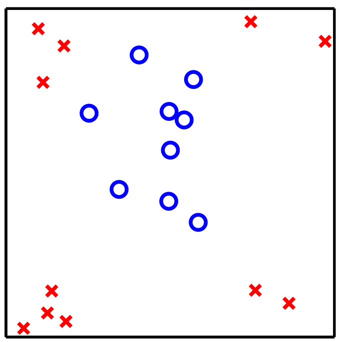
Learning From Data
Lecture 11
Overfitting

What is Overfitting
When does Overfitting Occur
Stochastic and Deterministic Noise



M. Magdon-Ismail
CSCI 4100/6100

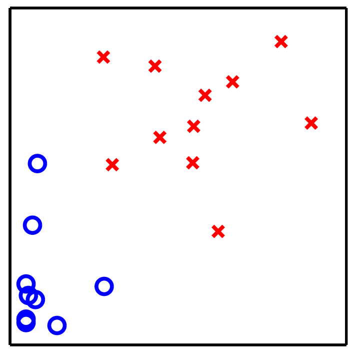
RECAP: Nonlinear Transforms



1. Original data

$\mathbf{x}_n \in \mathcal{X}$

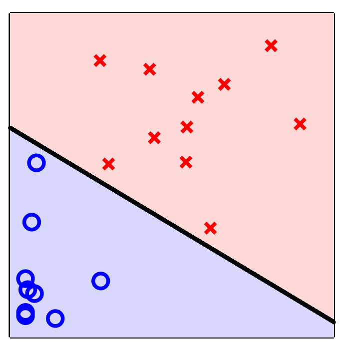
$\xrightarrow{\boldsymbol{\Phi}}$



2. Transform the data

$\mathbf{z}_n = \Phi(\mathbf{x}_n) \in \mathcal{Z}$

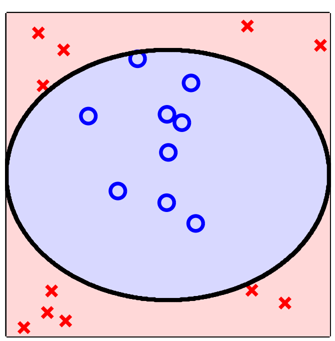
$\downarrow$



3. Separate data in $\mathcal{Z}$-space

$\tilde{g}(\mathbf{z}) = \text{sign}(\tilde{\mathbf{w}}^{\mathrm{T}}\mathbf{z})$

$\xleftarrow{\text{`}\boldsymbol{\Phi}^{-1}\text{'}}$

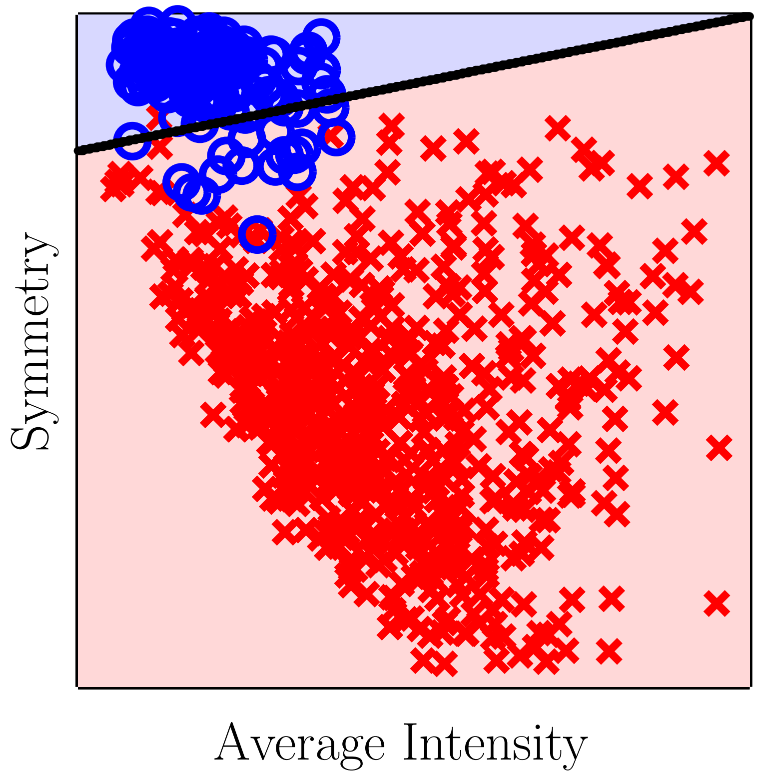


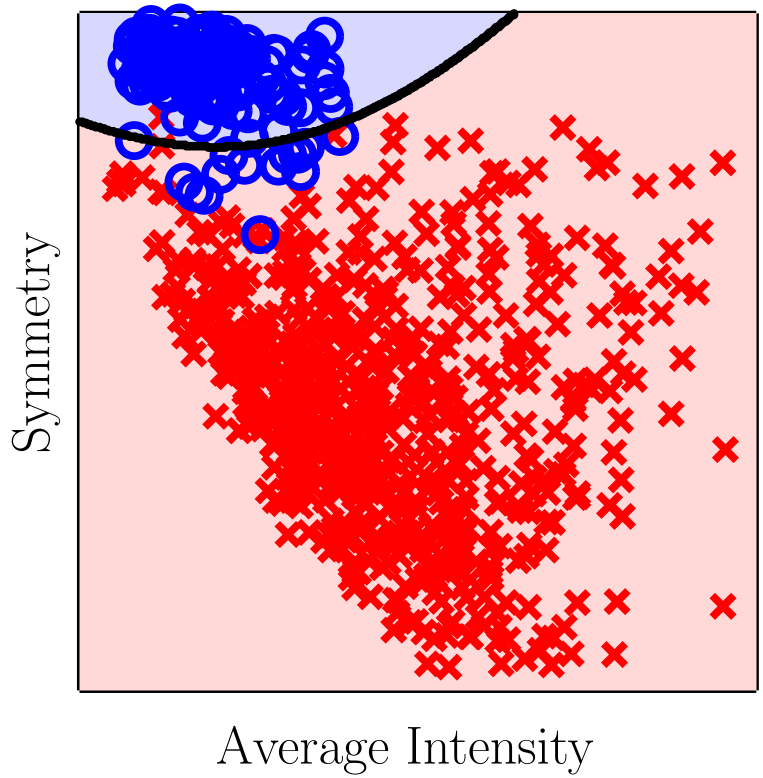
4. Classify in $\mathcal{X}$-space

$g(\mathbf{x}) = \tilde{g}(\Phi(\mathbf{x})) = \text{sign}(\tilde{\mathbf{w}}^{\mathrm{T}}\Phi(\mathbf{x}))$

$\underline{\mathcal{X}\text{-space is } \mathbb{R}^d}$	$\underline{\mathcal{Z}\text{-space is } \mathbb{R}^{\tilde{d}}}$
$\mathbf{x} = \begin{bmatrix} 1 \\ x_1 \\ \vdots \\ x_d \end{bmatrix}$	$\mathbf{z} = \boldsymbol{\Phi}(\mathbf{x}) = \begin{bmatrix} 1 \\ \Phi_1(\mathbf{x}) \\ \vdots \\ \Phi_{\tilde{d}}(\mathbf{x}) \end{bmatrix} = \begin{bmatrix} 1 \\ z_1 \\ \vdots \\ z_{\tilde{d}} \end{bmatrix}$
$\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N$	$\mathbf{z}_1, \mathbf{z}_2, \ldots, \mathbf{z}_N$
$y_1, y_2, \ldots, y_N$	$y_1, y_2, \ldots, y_N$
no weights	$\tilde{\mathbf{w}} = \begin{bmatrix} w_0 \\ w_1 \\ \vdots \\ w_{\tilde{d}} \end{bmatrix}$
$d_{\text{VC}} = d + 1$	$\boldsymbol{d_{\text{VC}} = d + 1}$
$g(\mathbf{x}) = \text{sign}(\tilde{\mathbf{w}}^{\mathrm{T}}\boldsymbol{\Phi}(\mathbf{x}))$	

RECAP: Digits Data "1" Versus "All"





Linear model
$E_{\text{in}} = 2.13\%$
$E_{\text{out}} = 2.38\%$

3rd order polynomial model
$E_{\text{in}} = 1.75\%$
$E_{\text{out}} = 1.87\%$

Superstitions – Myth or Reality?

- Paraskevedekatriaphobia – fear of Friday the 13th.
  - Are *future* Friday the 13ths really more dangerous?

- OCD [medical journal, citation lost, can you find it?]

  the subjects performs an action which leads to a good outcome and thereby generalizes it as cause and effect: the action will always give good results. Having *overfit* the data, the subject compulsively engages in that activity.

Humans are **overfitting machines**, very good at "finding coincidences".

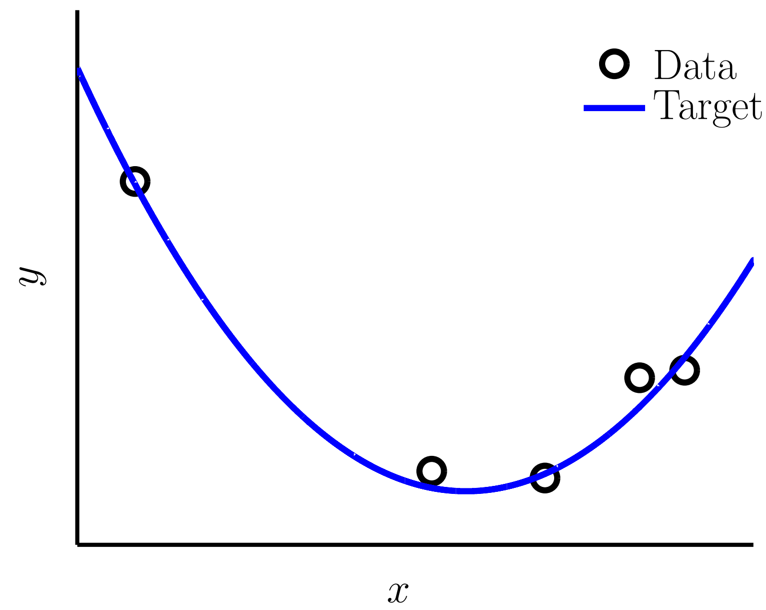
An Illustration of Overfitting on a Simple Example

Quadratic f

5 data points

A *little* noise (measurement error)

5 data points $\rightarrow$ 4th order polynomial fit



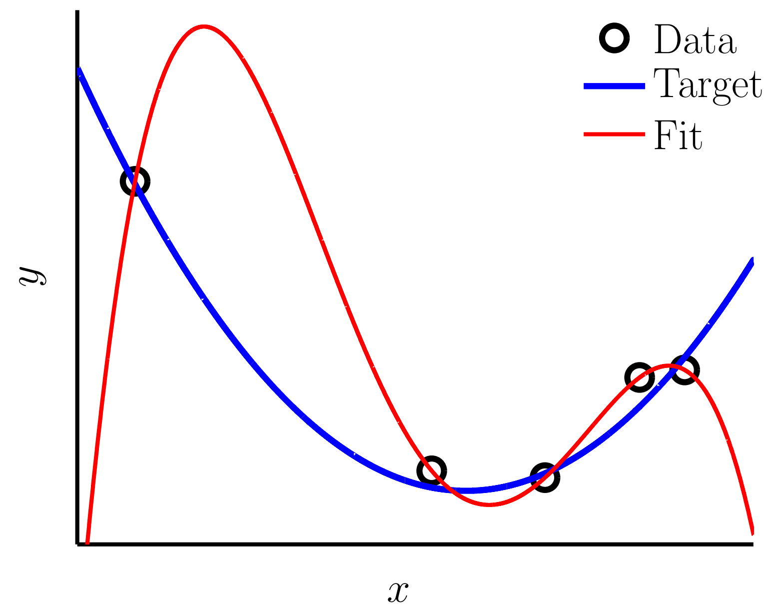
An Illustration of Overfitting on a Simple Example

Quadratic f

5 data points

A *little* noise (measurement error)

5 data points $\rightarrow$ 4th order polynomial fit



Classic overfitting: simple target with excessively complex $\mathcal{H}$.

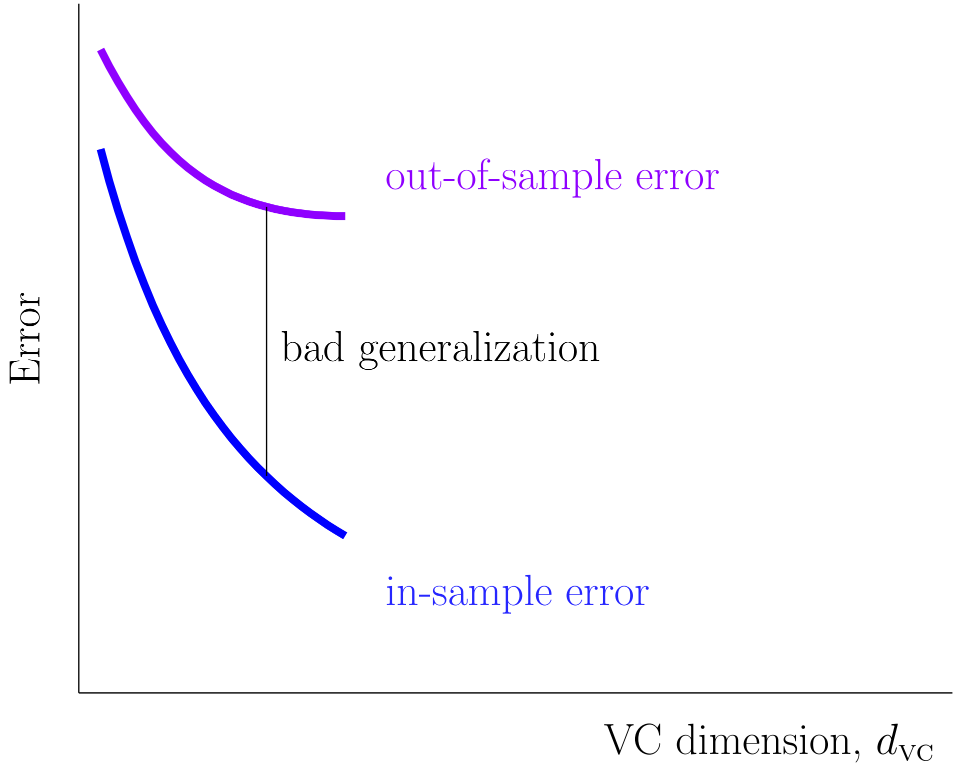
$$E_{\text{in}} \approx 0;\ E_{\text{out}} \gg 0$$

The noise did us in. (why?)

What is Overfitting?

Fitting the data more than is warranted

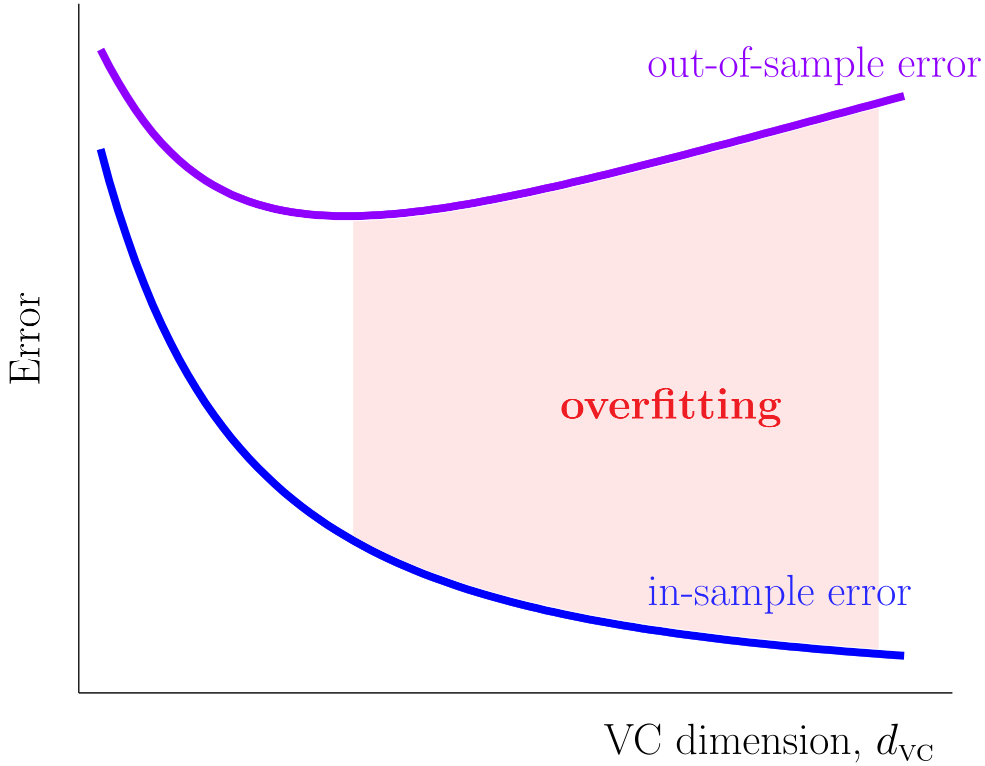
Overfitting is Not Just Bad Generalization



VC Analysis:

Covers bad generalization but with lots of slack – the VC bound is loose

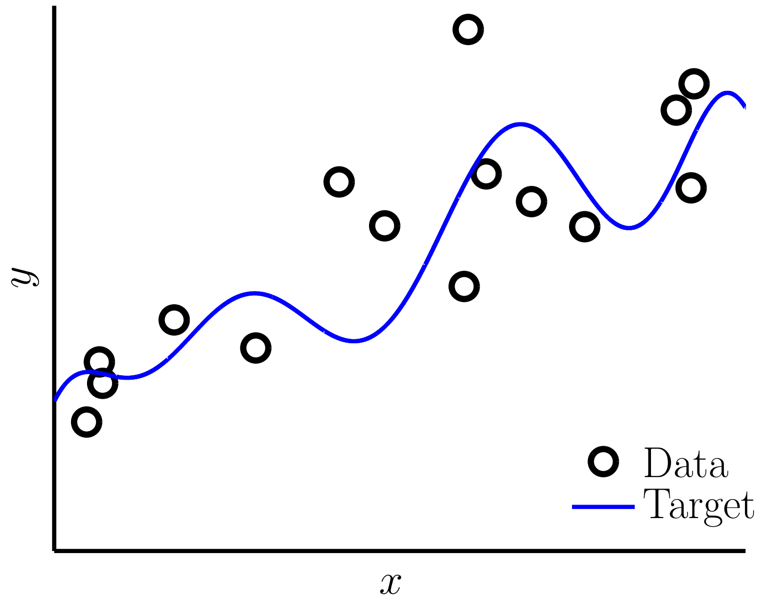
Overfitting is Not Just Bad Generalization



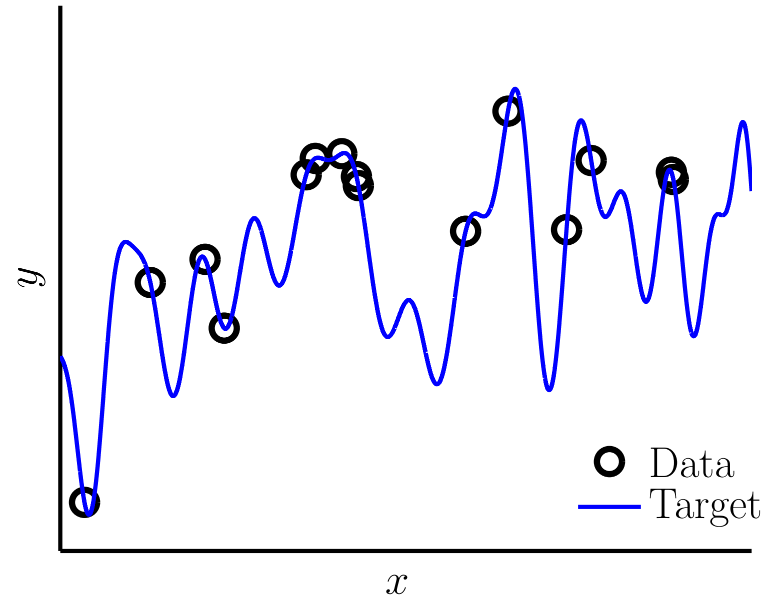
Overfitting:

Going for lower and lower E_{in} results in higher and higher E_{out}

Case Study: 2nd vs 10th Order Polynomial Fit

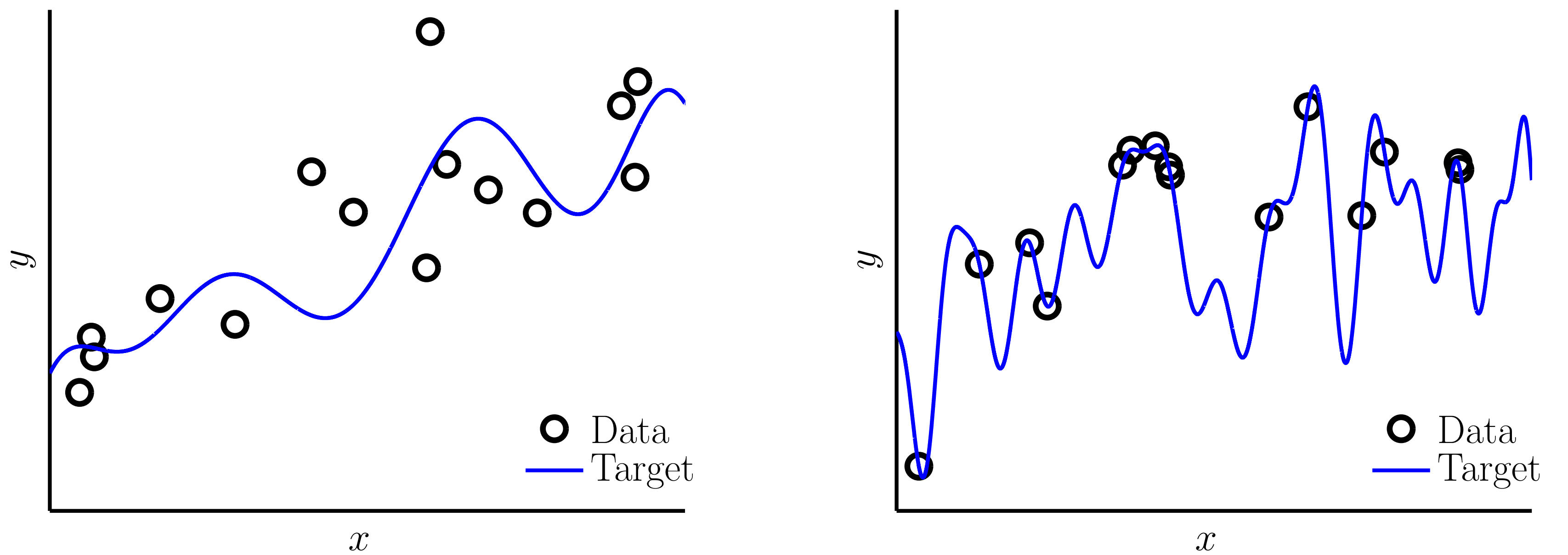


10th order f with noise.



50th order f with no noise.

Case Study: 2nd vs 10th Order Polynomial Fit



10th order f with noise.

50th order f with no noise.

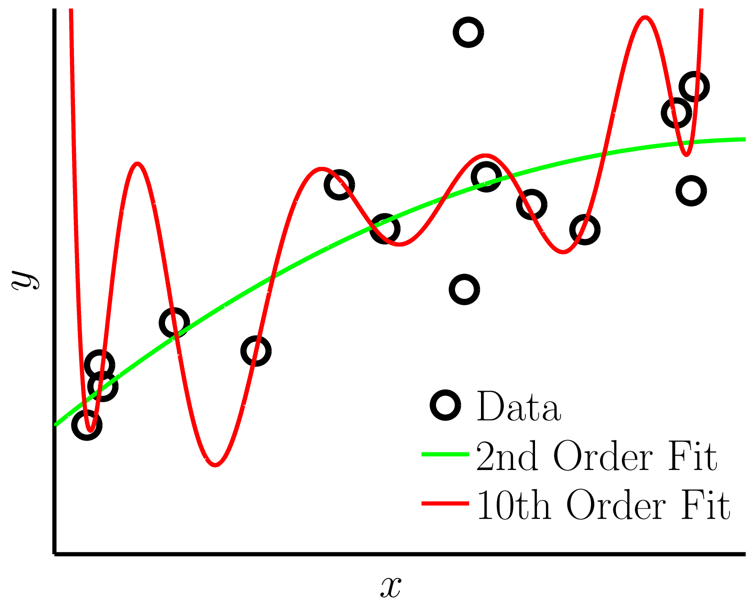
$\mathcal{H}_2$: 2nd order polynomial fit

$\mathcal{H}_{10}$: 10th order polynomial fit

$\longleftarrow$ special case of linear models with feature transform $x \mapsto (1, x, x^2, \cdots)$.

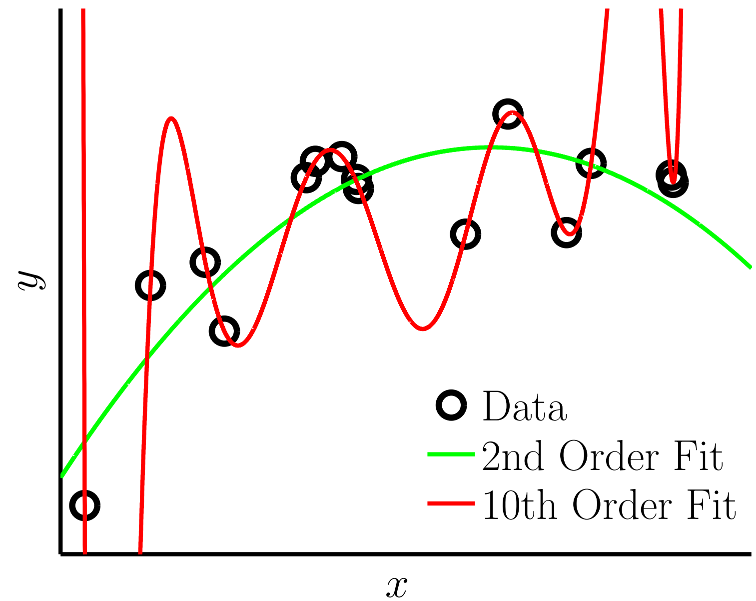
Which model do you pick for which problem and why?

Case Study: 2nd vs 10th Order Polynomial Fit



simple noisy target

	2nd Order	10th Order
E_{in}	0.050	0.034
E_{out}	0.127	**9.00**



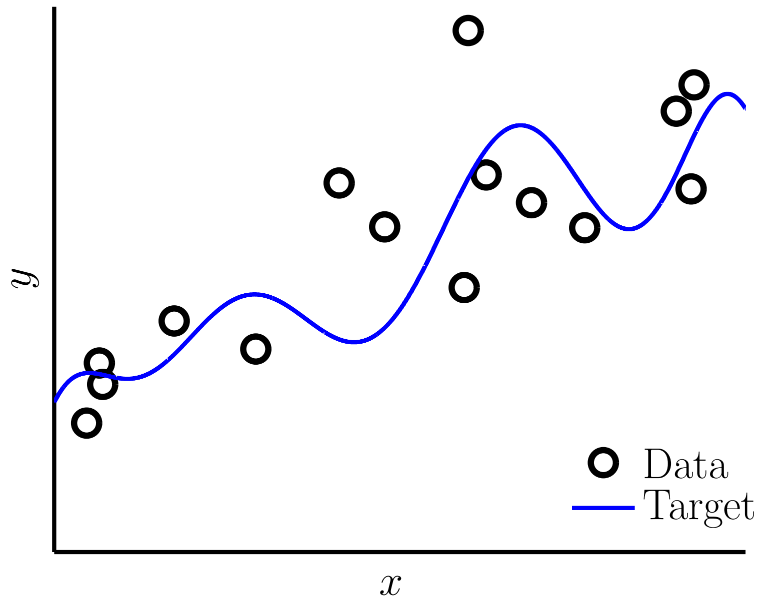
complex noiseless target

	2nd Order	10th Order
E_{in}	0.029	10^{-5}
E_{out}	0.120	**7680**

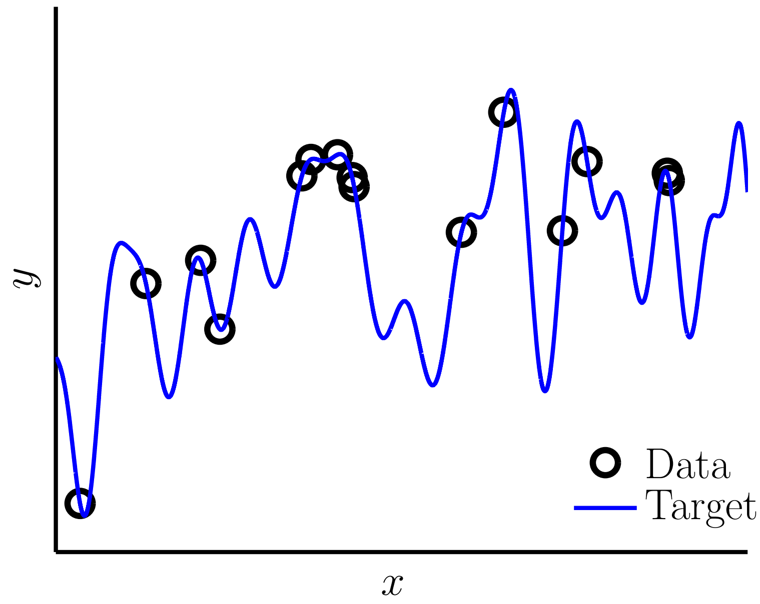
Go figure:

Simpler $\mathcal{H}$ is better even for the more complex target with no noise.

Is there Really "No Noise" with the Complex f?

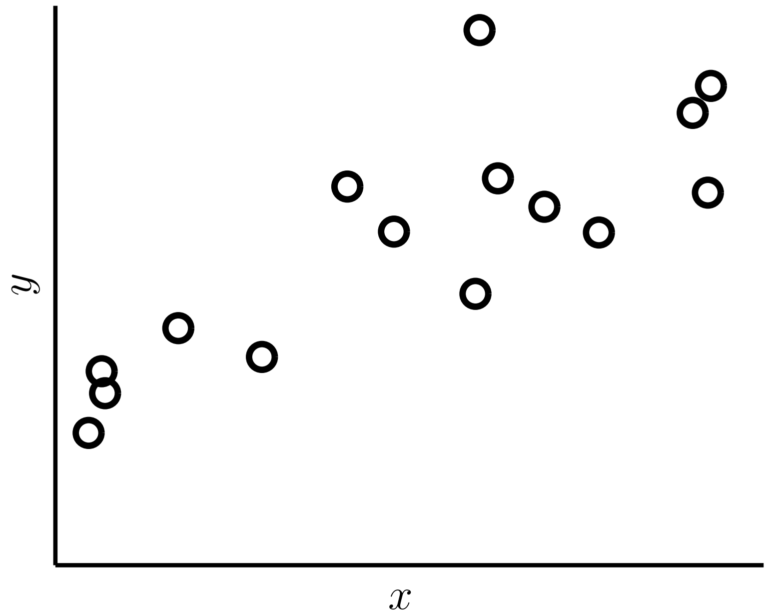


Simple f with noise.

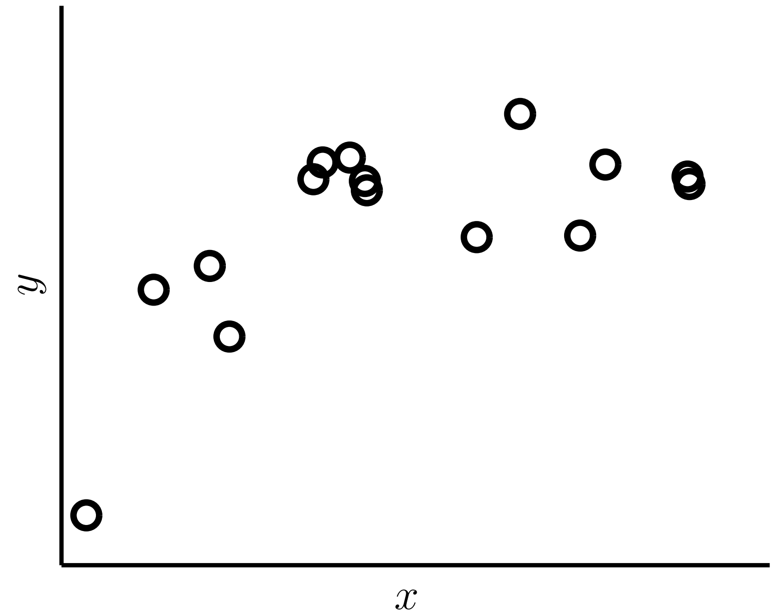


Complex f with no noise.

Is there Really "No Noise" with the Complex f?



Simple f with noise.

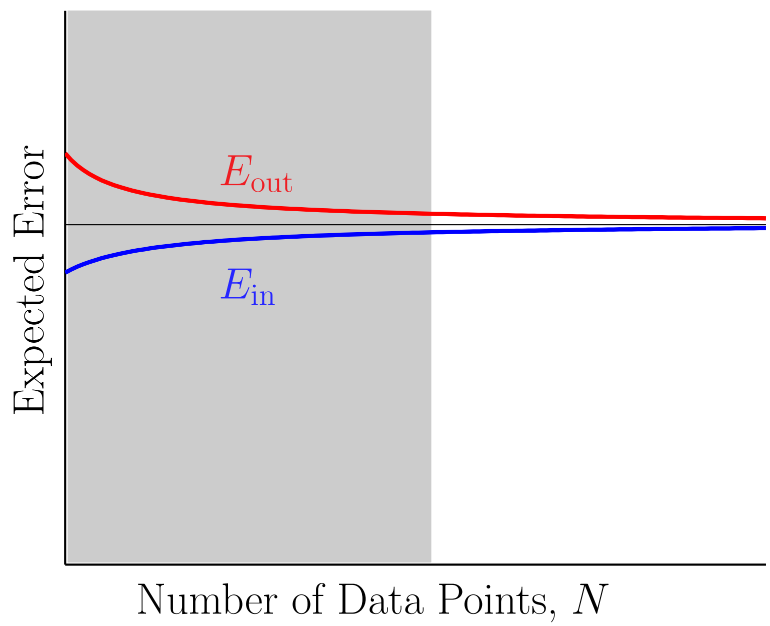


Complex f with no noise.

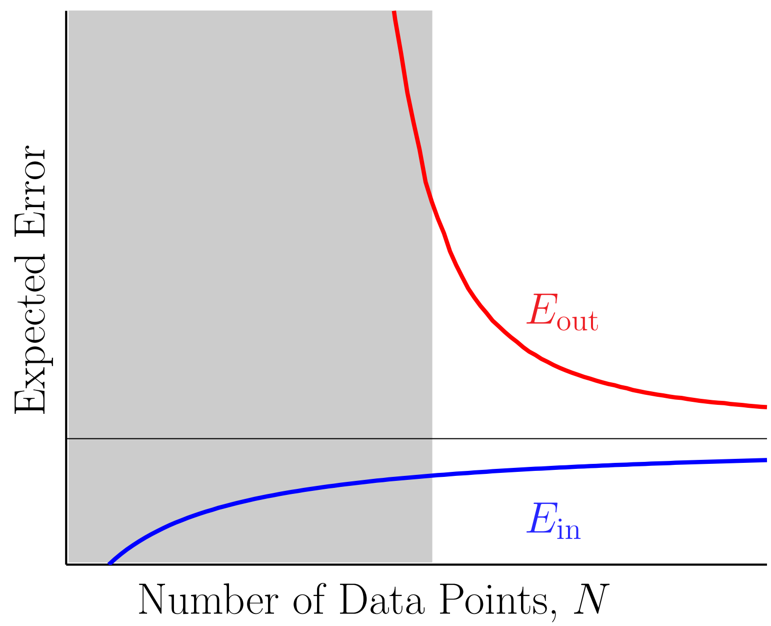
$\mathcal{H}$ should match *quantity and quality of data*, not f

When is $\mathcal{H}_2$ Better than $\mathcal{H}_{10}$?

Learning curves for $\mathcal{H}_2$



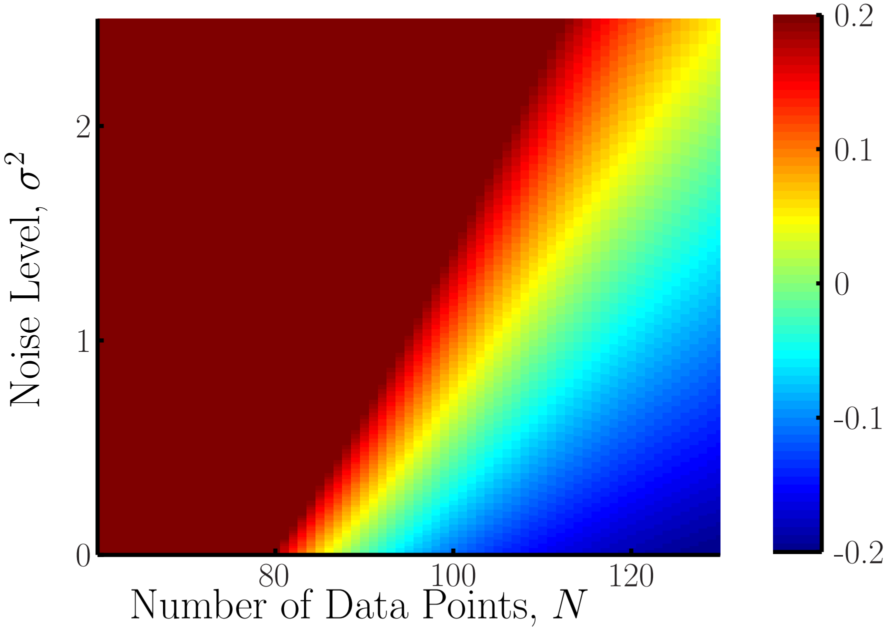
Learning curves for $\mathcal{H}_{10}$



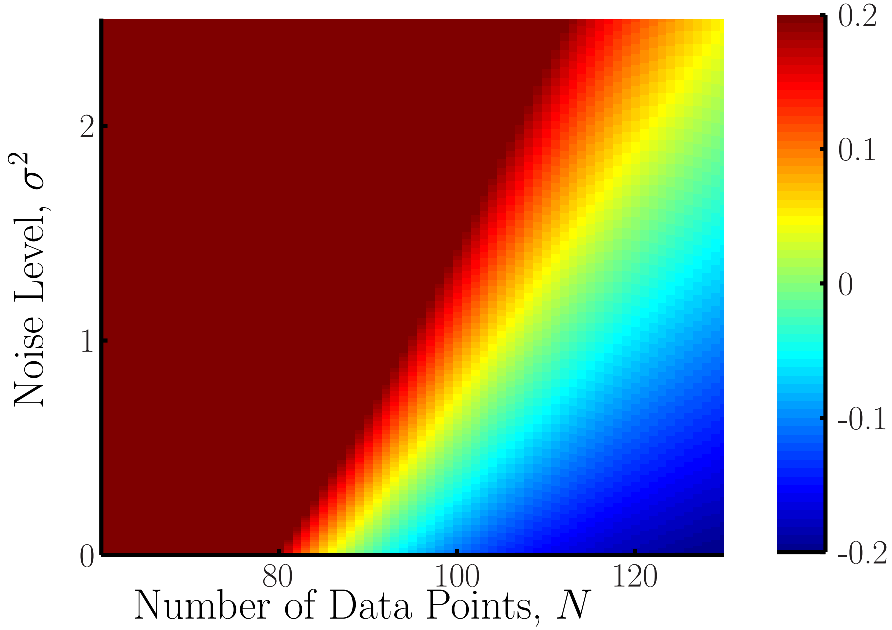
Overfitting:

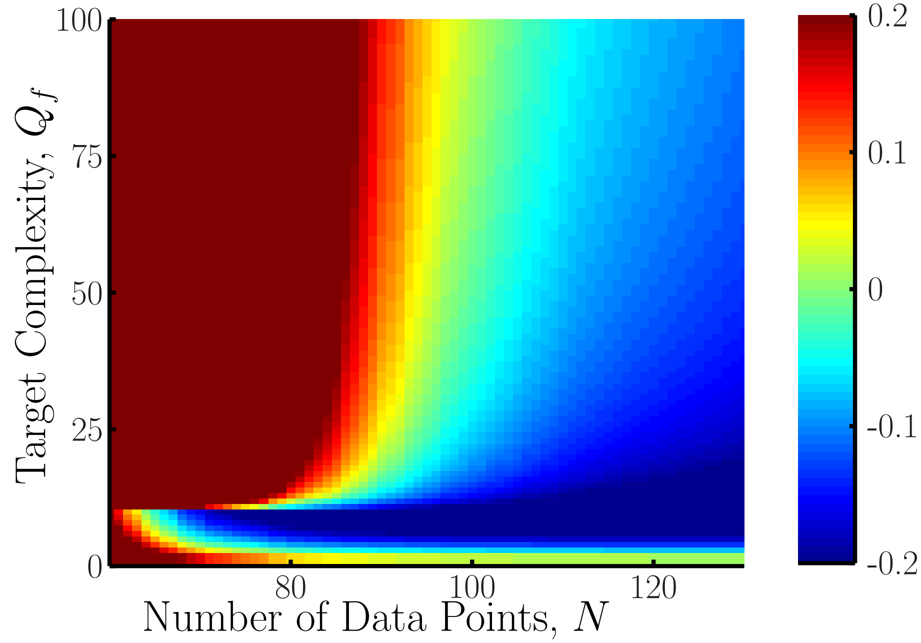
$$E_{\text{out}}(\mathcal{H}_{10}) > E_{\text{out}}(\mathcal{H}_2)$$

Overfit Measure: $E_{\text{out}}(\mathcal{H}_{10}) - E_{\text{out}}(\mathcal{H}_2)$



Overfit Measure: $E_{\text{out}}(\mathcal{H}_{10}) - E_{\text{out}}(\mathcal{H}_2)$





Number of data points $\uparrow$	Overfitting $\downarrow$
Noise $\uparrow$	Overfitting $\uparrow$
Target complexity $\uparrow$	Overfitting $\uparrow$

Noise

That part of y we *cannot* model

it has two sources . . .

Stochastic Noise — Data Error

We would like to learn from $\textcolor{blue}{\bigcirc}$:

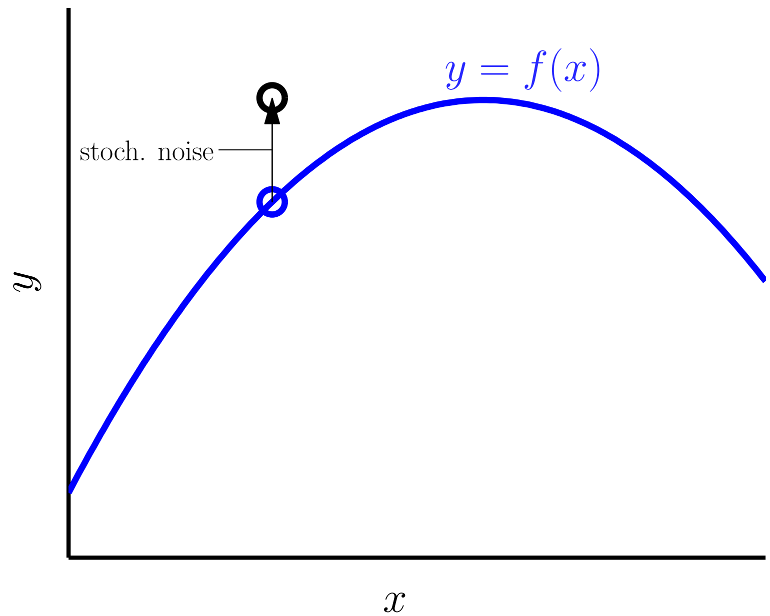
$$y_n = f(x_n)$$

Unfortunately, we only observe $\bigcirc$:

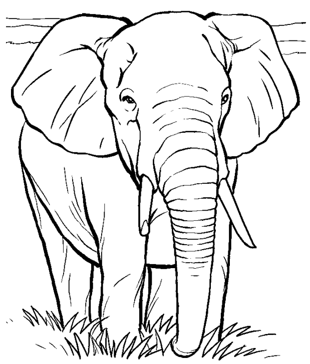
$$y_n \;=\; f(x_n) + \text{'stochastic noise'}$$

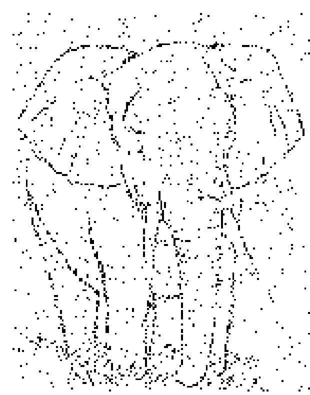
$\uparrow$

no one can model this



Stochastic Noise: fluctuations/measurement errors we cannot model.





Deterministic Noise — Model Error

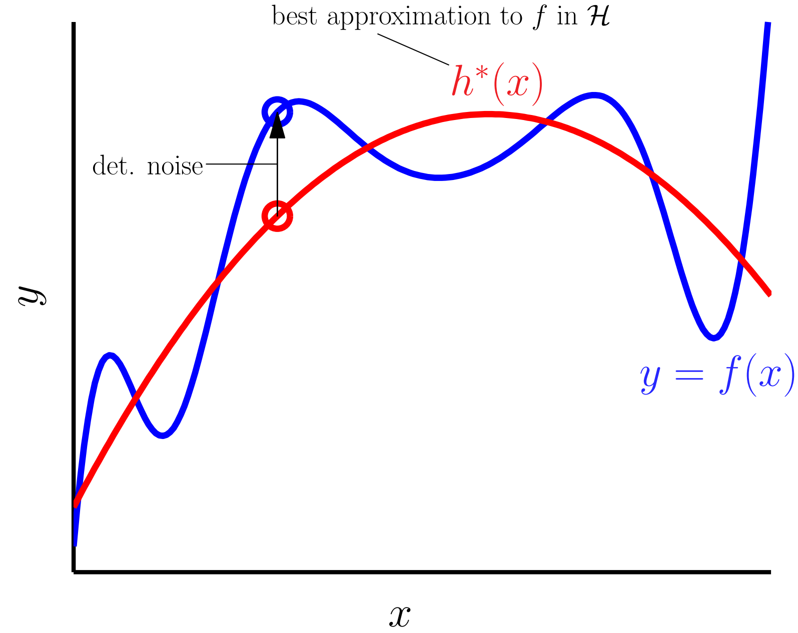
We would like to learn from ○:

$$y_n = h^*(x_n)$$

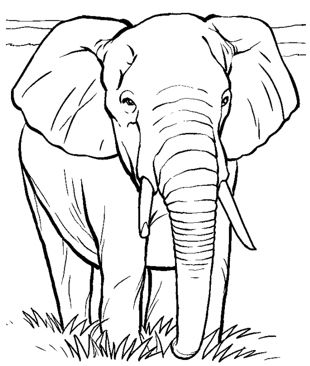
Unfortunately, we only observe ○:

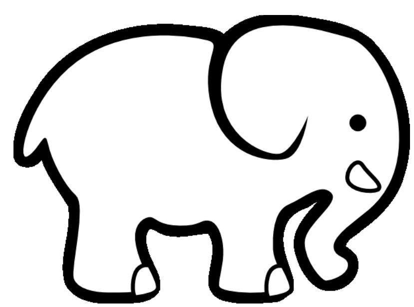
$$\begin{aligned} y_n &= f(x_n) \\ &= h^*(x_n) + \text{`deterministic noise'} \end{aligned}$$

$\mathcal{H}$ cannot model this



Deterministic Noise: the part of f we cannot model.

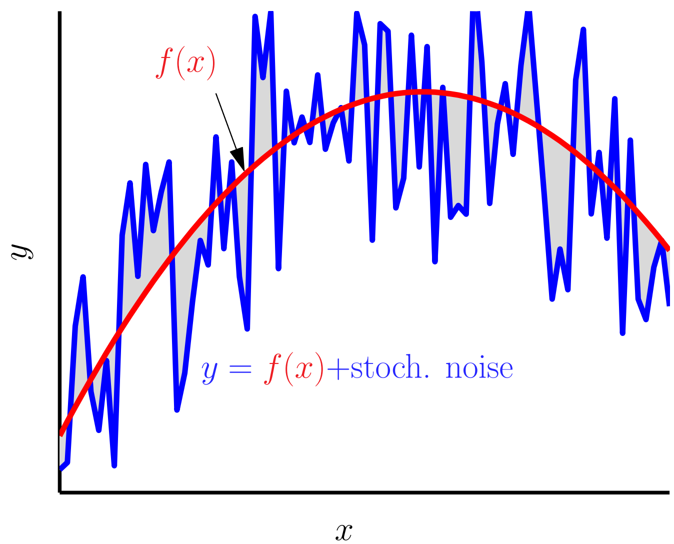




© Creator: Malik Magdon-Ismail

Stochastic & Deterministic Noise Hurt Learning

Stochastic Noise



source: random measurement errors

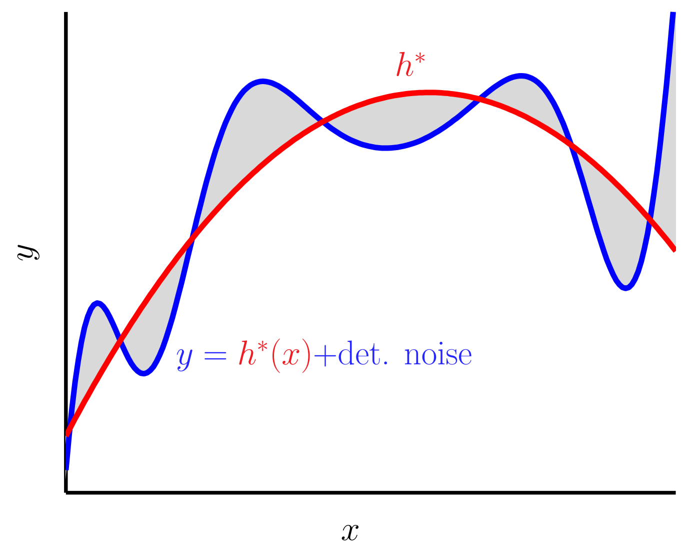
re-measure y_n

stochastic noise changes.

change $\mathcal{H}$

stochastic noise the same.

Deterministic Noise



source: learner's $\mathcal{H}$ cannot model f

re-measure y_n

deterministic noise the same.

change $\mathcal{H}$

deterministic noise changes.

We have single $\mathcal{D}$ and fixed $\mathcal{H}$ so we cannot distinguish

Noise and the Bias-Variance Decomposition

$$y = f(\mathbf{x}) + \epsilon$$

$$\uparrow$$

measurement error

$$\begin{aligned}
\mathbb{E}[E_{\text{out}}(\mathbf{x})] &= \mathbb{E}_{\mathcal{D},\epsilon}[(g(\mathbf{x}) - f(\mathbf{x}) - \epsilon)^2] \\
&= \mathbb{E}_{\mathcal{D},\epsilon}[\underbrace{(g(\mathbf{x}) - f(\mathbf{x}))^2}_{\text{bias} + \text{var}} + \underbrace{2(g(\mathbf{x}) - f(\mathbf{x}))\epsilon}_{0} + \underbrace{\epsilon^2}_{\sigma^2}]
\end{aligned}$$

Noise and the Bias-Variance Decomposition

$$y = f(\mathbf{x}) + \epsilon$$

$$\uparrow$$

measurement error

$$\mathbb{E}[E_{\text{out}}(\mathbf{x})] = \underbrace{\sigma^2}_{\text{stochastic noise}} + \underbrace{\textsf{bias}}_{\text{deterministic noise}} + \underbrace{\textsf{var}}_{\text{indirect impact of noise}}$$

stochastic noise

deterministic noise

indirect **impact** of noise

Noise is the Culprit

Overfitting is the disease

Noise is the cause

Learning is led astray by fitting the noise more than the signal

Cures

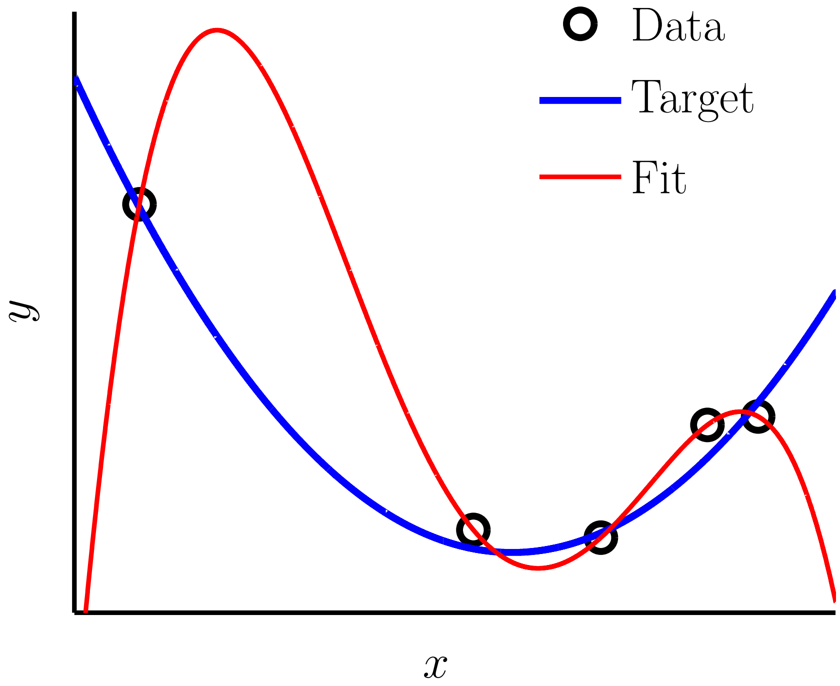
Regularization: Putting on the brakes.

Validation: A reality check from peeking at E_{out} (the bottom line).

Regularization

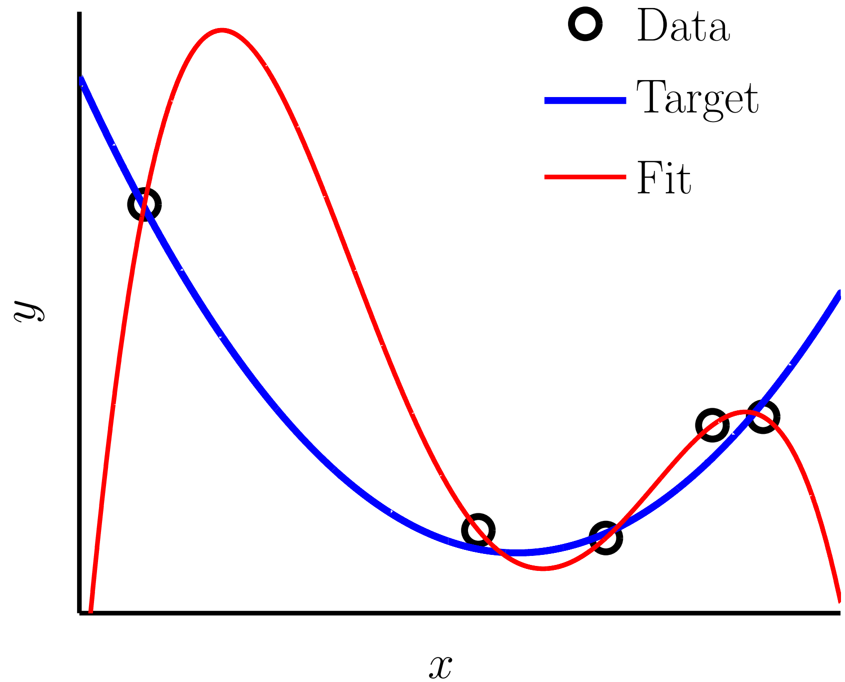
no regularization

regularization!



Regularization

no regularization



regularization!