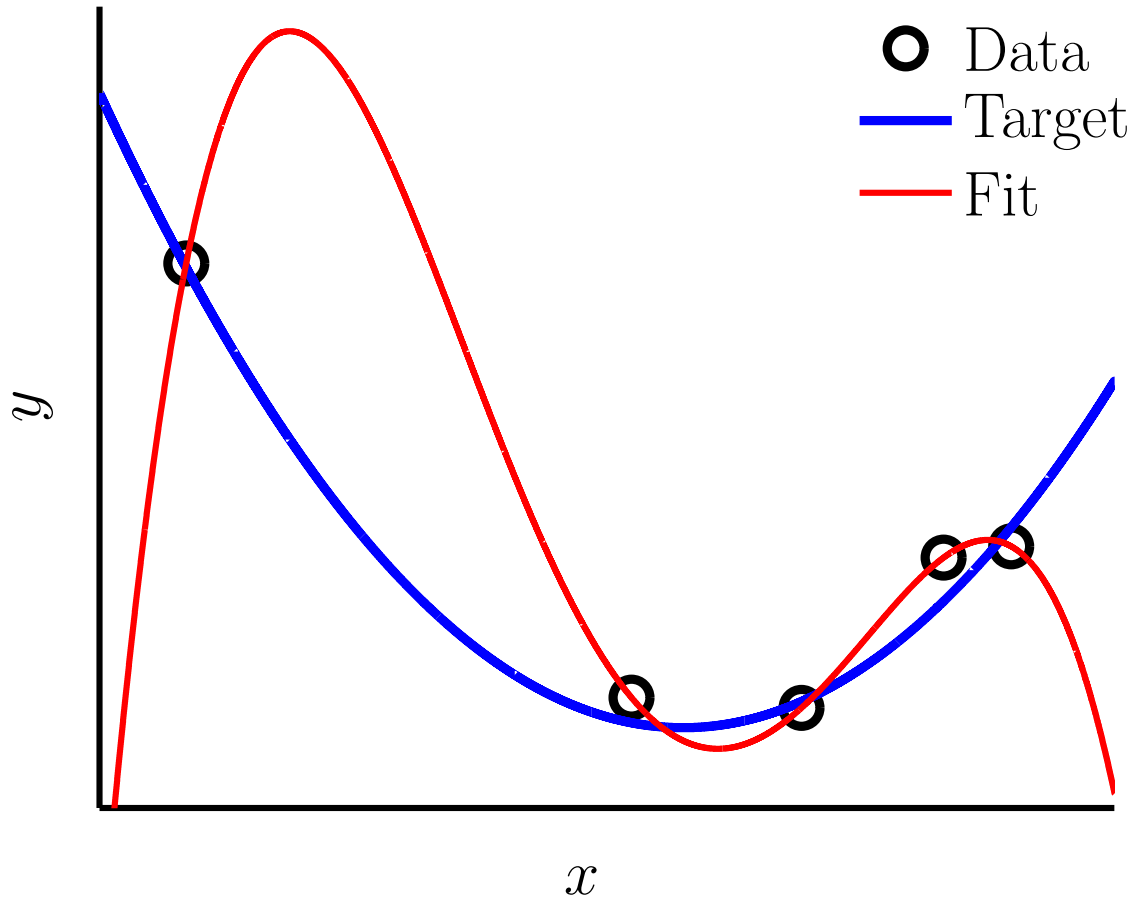
Learning From Data
Lecture 12
Regularization

Constraining the Model
Weight Decay
Augmented Error

M. Magdon-Ismail
CSCI 4100/6100

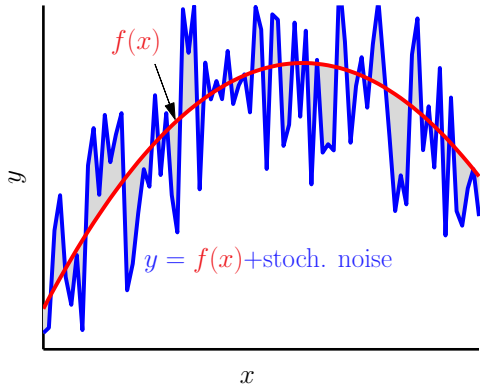
Overfitting

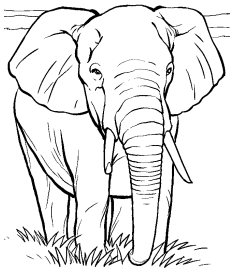
Fitting the data more than is warranted



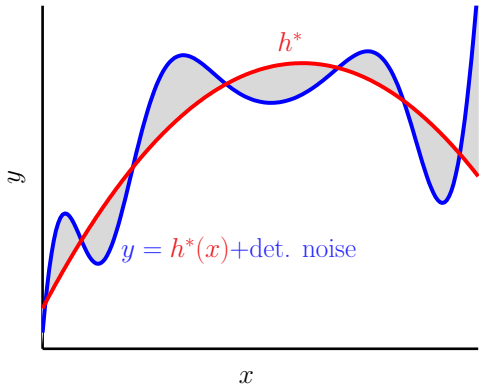
RECAP: Noise is Part of y We Cannot Model

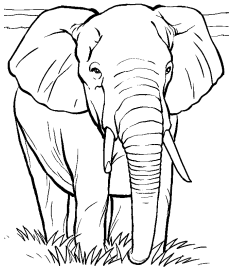
Stochastic Noise

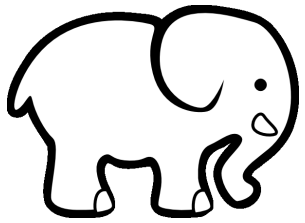




Deterministic Noise







Stochastic and Deterministic Noise Hurt Learning

Human: Good at extracting the simple pattern, ignoring the noise and complications.

Computer: Pays equal attention to all pixels. Needs help simplifying $\rightarrow$ (features ✓, regularization).

Regularization

What is regularization?

A cure for our tendency to fit (get distracted by) the noise, hence improving E_{out}.

How does it work?

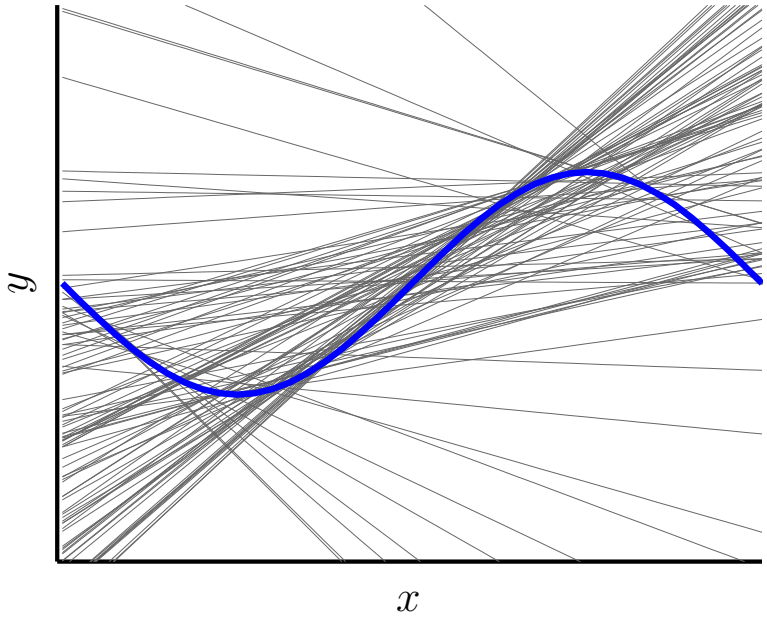
By constraining the model so that we cannot fit the noise.

↑
putting on the brakes

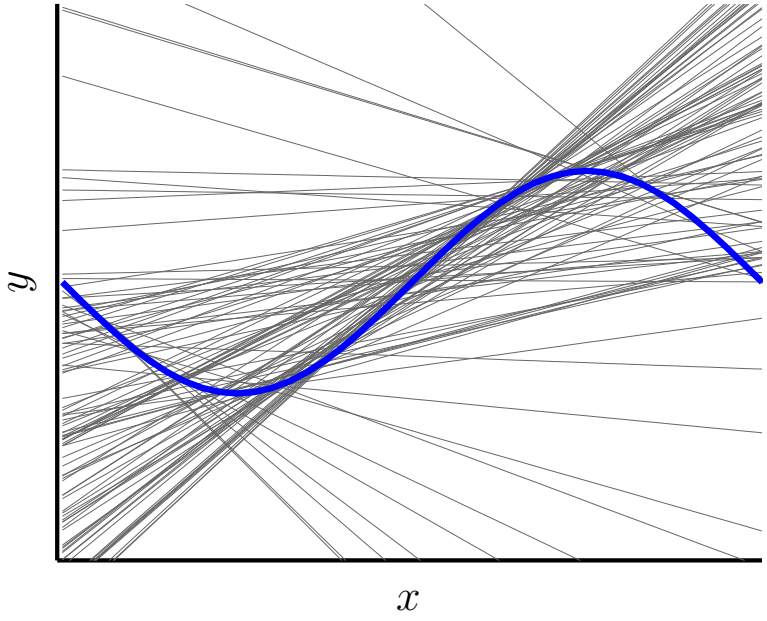
Side effects?

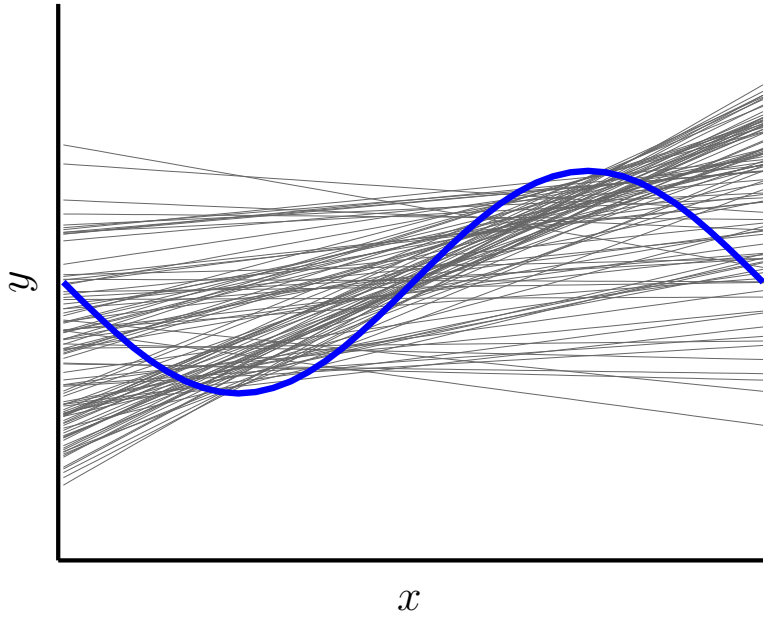
The medication will have side effects – if we cannot fit the noise, maybe we cannot fit f (the signal)?

Constraining the Model: Does it Help?



Constraining the Model: Does it Help?

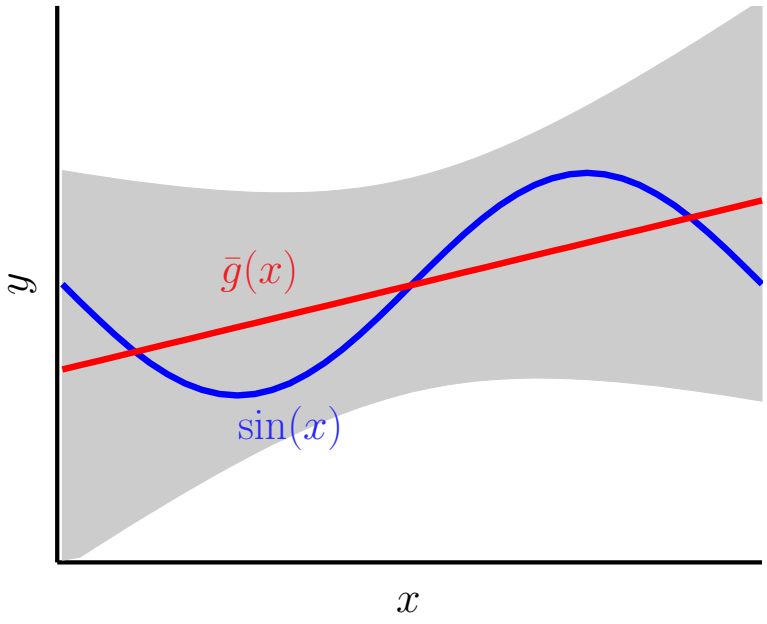




constrain weights to be smaller

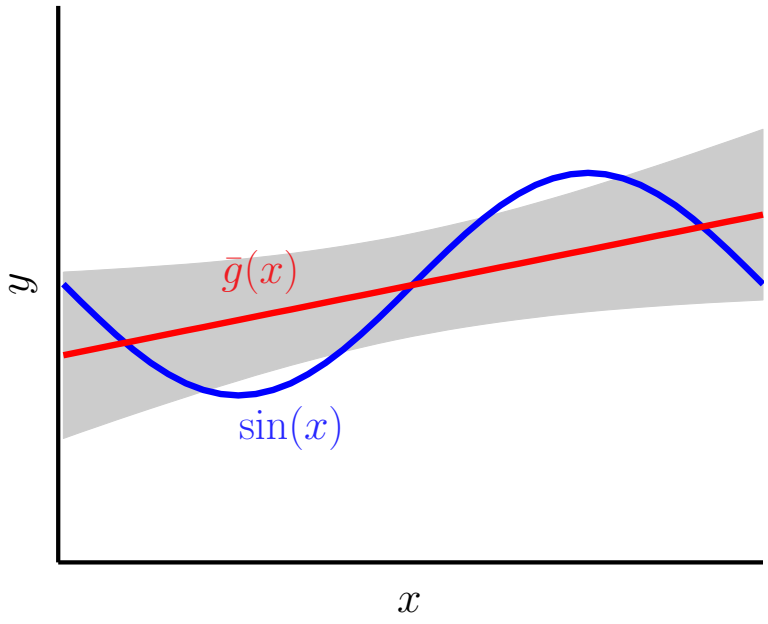
. . . and the winner is:

Bias Goes Up A Little



no regularization

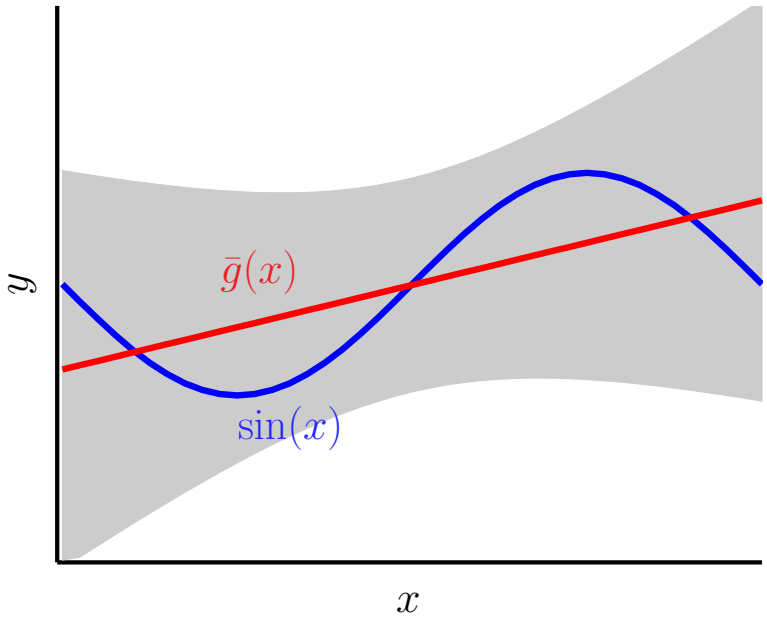
bias $= 0.21$

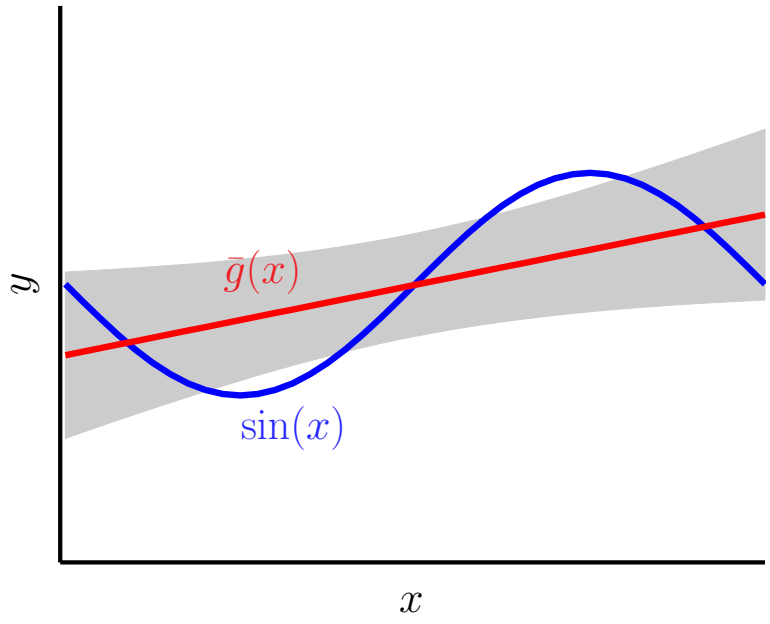


regularization

bias $= 0.23$ ← side effect

Variance Drop is Dramatic!





no regularization

$\mathsf{bias} = 0.21$

$\mathsf{var} = 1.69$

regularization

$\mathsf{bias} = 0.23$ ← side effect

$\mathsf{var} = 0.33$ ← treatment

(Constant model had bias=0.5 and var=0.25.)

Regularization in a Nutshell

VC analysis:

$$E_{\text{out}}(g) \leq E_{\text{in}}(g) + \Omega(\mathcal{H})$$

If you use a simpler $\mathcal{H}$ and get a good fit, then your E_{out} is better.

Regularization takes this a step further:

If you use a 'simpler' h and get a good fit, then is your E_{out} better?

Polynomials of Order Q - A Useful Testbed

$\mathcal{H}_{\mathrm{Q}}$: polynomials of order Q.

Standard Polynomial

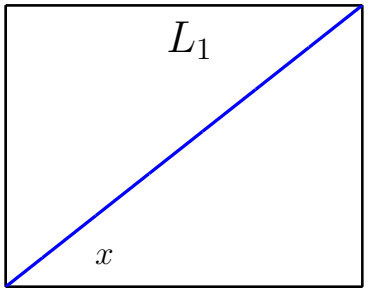
$$\mathbf{z} = \begin{bmatrix} 1 \\ x \\ x^2 \\ \vdots \\ x^{\mathrm{Q}} \end{bmatrix} \qquad \begin{aligned} h(x) &= \mathbf{w}^{\mathrm{T}}\mathbf{z}(x) \\ &= w_0 + w_1 x + \cdots + w_{\mathrm{Q}} x^{\mathrm{Q}} \end{aligned}$$

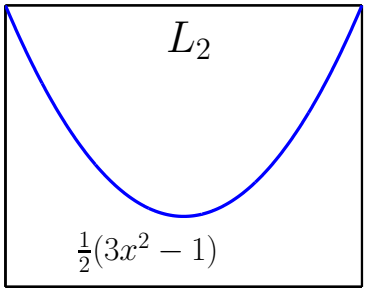
Legendre Polynomial

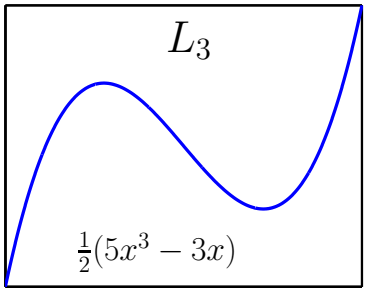
we're using linear regression

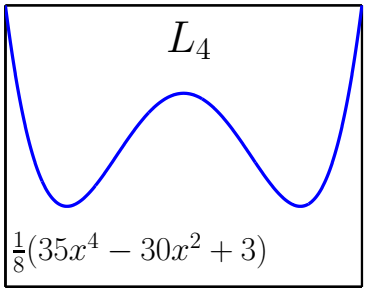
$$\mathbf{z} = \begin{bmatrix} 1 \\ L_1(x) \\ L_2(x) \\ \vdots \\ L_{\mathrm{Q}}(x) \end{bmatrix} \qquad \begin{aligned} h(x) &= \mathbf{w}^{\mathrm{T}}\mathbf{z}(x) \\ &= w_0 + w_1 L_1(x) + \cdots + w_{\mathrm{Q}} L_{\mathrm{Q}}(x) \end{aligned}$$

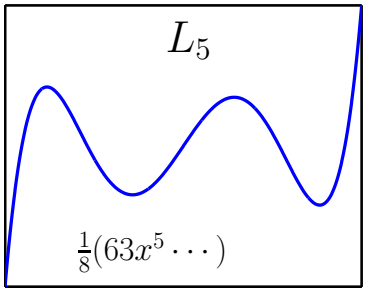
allows us to treat the weights 'independently'











Linear Regression

$$\underbrace{(x_1, y_1), \ldots, (x_N, y_N)}_{\mathrm{X}\ \mathbf{y}} \longrightarrow \underbrace{(\mathbf{z}_1, y_1), \ldots, (\mathbf{z}_N, y_N)}_{\mathrm{Z}\ \mathbf{y}}$$

$$\min: \ E_{\text{in}}(\mathbf{w}) = \frac{1}{N}\sum_{n=1}^{N}(\mathbf{w}^{\mathrm{T}}\mathbf{z}_n - y_n)^2$$

$$= \frac{1}{N}(\mathrm{Z}\mathbf{w} - \mathbf{y})^{\mathrm{T}}(\mathrm{Z}\mathbf{w} - \mathbf{y})$$

$$\mathbf{w}_{\text{lin}} = (\mathrm{Z}^{\mathrm{T}}\mathrm{Z})^{-1}\mathrm{Z}^{\mathrm{T}}\mathbf{y}$$

linear regression fit

Constraining The Model: $\mathcal{H}_{10}$ vs. $\mathcal{H}_2$

$$\mathcal{H}_{10} = \left\{ h(x) = w_0 + w_1\Phi_1(x) + w_2\Phi_2(x) + w_3\Phi_3(x) + \cdots + w_{10}\Phi_{10}(x) \right\}$$

$$\mathcal{H}_2 = \left\{ \begin{array}{c} h(x) = w_0 + w_1\Phi_1(x) + w_2\Phi_2(x) + w_3\Phi_3(x) + \cdots + w_{10}\Phi_{10}(x) \\ \text{such that: } w_3 = w_4 = \cdots = w_{10} = 0 \end{array} \right\}$$

↗

a 'hard' order constraint that sets some weights to zero

$$\mathcal{H}_2 \subset \mathcal{H}_{10}$$

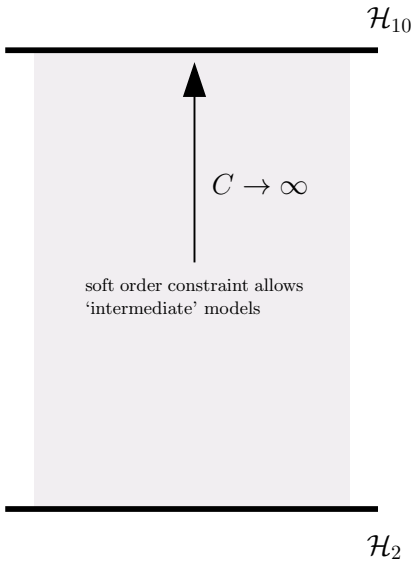
Soft Order Constraint

Don't set weights explicitly to zero (e.g. $w_3 = 0$).

Give a budget and let the learning choose.

$$\sum_{q=0}^{\mathrm{Q}} w_q^2 \leq C$$

budget for weights



Soft Order Constrained Model $\mathcal{H}_C$

$$\mathcal{H}_{10} = \left\{ h(x) = w_0 + w_1\Phi_1(x) + w_2\Phi_2(x) + w_3\Phi_3(x) + \cdots + w_{10}\Phi_{10}(x) \right\}$$

$$\mathcal{H}_2 = \left\{ \begin{array}{c} h(x) = w_0 + w_1\Phi_1(x) + w_2\Phi_2(x) + w_3\Phi_3(x) + \cdots + w_{10}\Phi_{10}(x) \\ \text{such that: } w_3 = w_4 = \cdots = w_{10} = 0 \end{array} \right\}$$

$$\mathcal{H}_C = \left\{ \begin{array}{c} h(x) = w_0 + w_1\Phi_1(x) + w_2\Phi_2(x) + w_3\Phi_3(x) + \cdots + w_{10}\Phi_{10}(x) \\ \text{such that: } \sum_{q=0}^{10} w_q^2 \leq C \end{array} \right\}$$

a 'soft' budget constraint
on the sum of weights

VC-perspective: $\mathcal{H}_C$ is smaller than $\mathcal{H}_{10}$ $\implies$ better generalization.

Fitting the Data

The optimal weights

$$\mathbf{w}_{\text{reg}} \in \mathcal{H}_C$$

regularized

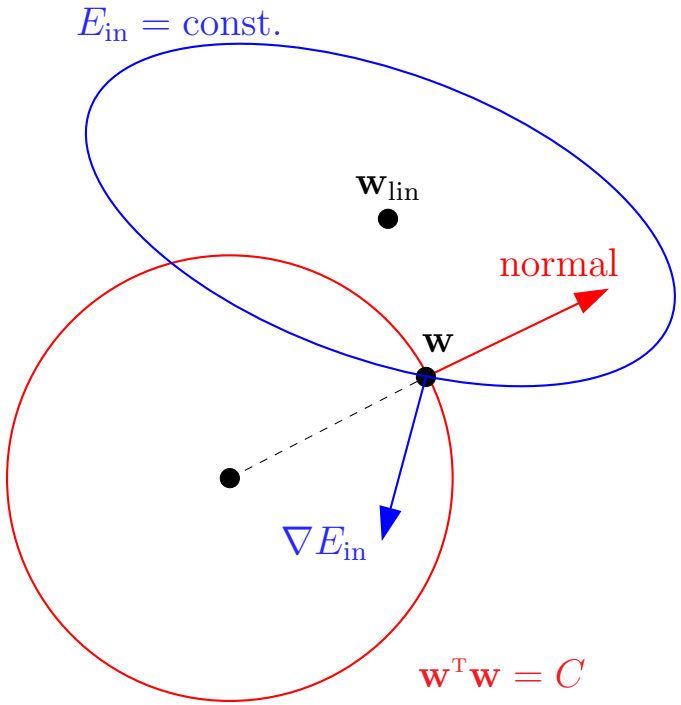
should minimize the in-sample error, but be within the budget.

$\mathbf{w}_{\text{reg}}$ is a solution to

$$\begin{aligned} \min: \quad & E_{\text{in}}(\mathbf{w}) = \frac{1}{N}(\mathrm{Z}\mathbf{w} - \mathbf{y})^{\mathrm{T}}(\mathrm{Z}\mathbf{w} - \mathbf{y}) \\ \text{subject to:} \quad & \mathbf{w}^{\mathrm{T}}\mathbf{w} \leq C \end{aligned}$$

Solving For $\mathbf{w}_{\text{reg}}$

$$\begin{aligned} \min: \quad & E_{\text{in}}(\mathbf{w}) = \frac{1}{N}(\mathrm{Z}\mathbf{w} - \mathbf{y})^{\mathrm{T}}(\mathrm{Z}\mathbf{w} - \mathbf{y}) \\ \text{subject to:} \quad & \mathbf{w}^{\mathrm{T}}\mathbf{w} \leq C \end{aligned}$$



Observations:

1. Optimal $\mathbf{w}$ tries to get as 'close' to $\mathbf{w}_{\text{lin}}$ as possible.
   Optimal $\mathbf{w}$ will use full budget and be on the surface $\mathbf{w}^{\mathrm{T}}\mathbf{w} = C$.
2. Surface $\mathbf{w}^{\mathrm{T}}\mathbf{w} = C$, at optimal $\mathbf{w}$, should be perpindicular to ∇E_{in}.
   Otherwise can move along the surface and decrease E_{in}.
3. Normal to surface $\mathbf{w}^{\mathrm{T}}\mathbf{w} = C$ is the vector $\mathbf{w}$.
4. Surface is $\perp \nabla E_{\text{in}}$; surface is $\perp$ normal.
   ∇E_{in} is parallel to normal (but in opposite direction).

$$\boxed{\nabla E_{\text{in}}(\mathbf{w}_{\text{reg}}) = -2\lambda_C \mathbf{w}_{\text{reg}}}$$

λ_C, the lagrange multiplier, is positive.
The 2 is for mathematical convenience.

Solving For $\mathbf{w}_{\text{reg}}$

$E_{\text{in}}(\mathbf{w})$ is minimized, subject to: $\mathbf{w}^{\text{T}}\mathbf{w} \leq C$

$$\Leftrightarrow \quad \nabla E_{\text{in}}(\mathbf{w}_{\text{reg}}) + 2\lambda_C \mathbf{w}_{\text{reg}} = \mathbf{0}$$

$$\Leftrightarrow \quad \nabla \left(E_{\text{in}}(\mathbf{w}) + \lambda_C \mathbf{w}^{\text{T}}\mathbf{w} \right)\Big|_{\mathbf{w}=\mathbf{w}_{\text{reg}}} = \mathbf{0}$$

$\Leftrightarrow$ $E_{\text{in}}(\mathbf{w}) + \lambda_C \mathbf{w}^{\text{T}}\mathbf{w}$ is minimized, unconditionally

There is a correspondence: $C \uparrow \quad \lambda_C \downarrow$

The Augmented Error

Pick a C and minimize

$$E_{\text{in}}(\mathbf{w}) \qquad \text{subject to: } \mathbf{w}^{\mathrm{T}}\mathbf{w} \leq C$$

$$\updownarrow$$

Pick a λ_C and minimize

$$E_{\text{aug}}(\mathbf{w}) = E_{\text{in}}(\mathbf{w}) + \lambda_C \mathbf{w}^{\mathrm{T}}\mathbf{w} \qquad \text{unconditionally}$$

A penalty for the 'complexity' of h, measured by the size of the weights.

We can pick any budget C. Translation: we are free to pick any multiplier λ_C

What's the right C? $\quad\leftrightarrow\quad$ What's the right λ_C?

Linear Regression With Soft Order Constraint

$$E_{\text{aug}}(\mathbf{w}) = \frac{1}{N}(\mathrm{Z}\mathbf{w} - \mathbf{y})^{\mathrm{T}}(\mathrm{Z}\mathbf{w} - \mathbf{y}) + \lambda_C \mathbf{w}^{\mathrm{T}}\mathbf{w}$$

Convenient to set $\lambda_C = \frac{\lambda}{N}$

$$E_{\text{aug}}(\mathbf{w}) = \frac{(\mathrm{Z}\mathbf{w} - \mathbf{y})^{\mathrm{T}}(\mathrm{Z}\mathbf{w} - \mathbf{y}) + \lambda \mathbf{w}^{\mathrm{T}}\mathbf{w}}{N}$$

called 'weight decay' as the penalty encourages smaller weights

Unconditionally minimize $E_{\text{aug}}(\mathbf{w})$.

The Solution for $\mathbf{w}_{\mathbf{reg}}$

$$\begin{aligned}\nabla E_{\text{aug}}(\mathbf{w}) &= 2\mathrm{Z}^{\mathrm{T}}(\mathrm{Z}\mathbf{w}-\mathbf{y})+2\lambda\mathbf{w}\\ &= 2(\mathrm{Z}^{\mathrm{T}}\mathrm{Z}+\lambda\mathrm{I})\mathbf{w}-2\mathrm{Z}^{\mathrm{T}}\mathbf{y}\end{aligned}$$

Set $\nabla E_{\text{aug}}(\mathbf{w})=\mathbf{0}$

$$\mathbf{w}_{\text{reg}}=(\mathrm{Z}^{\mathrm{T}}\mathrm{Z}+\lambda\mathrm{I})^{-1}\mathrm{Z}^{\mathrm{T}}\mathbf{y}$$

λ determines the amount of regularization

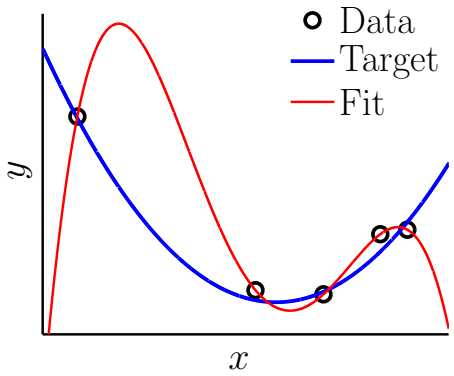
Recall the unconstrained solution ($\lambda=0$):

$$\mathbf{w}_{\text{lin}}=(\mathrm{Z}^{\mathrm{T}}\mathrm{Z})^{-1}\mathrm{Z}^{\mathrm{T}}\mathbf{y}$$

A Little Regularization . . .

Minimizing $E_{\text{in}}(\mathbf{w}) + \dfrac{\lambda}{N}\mathbf{w}^{\mathrm{T}}\mathbf{w}$ with different λ's

$\underline{\lambda = 0}$ $\underline{\lambda = 0.0001}$

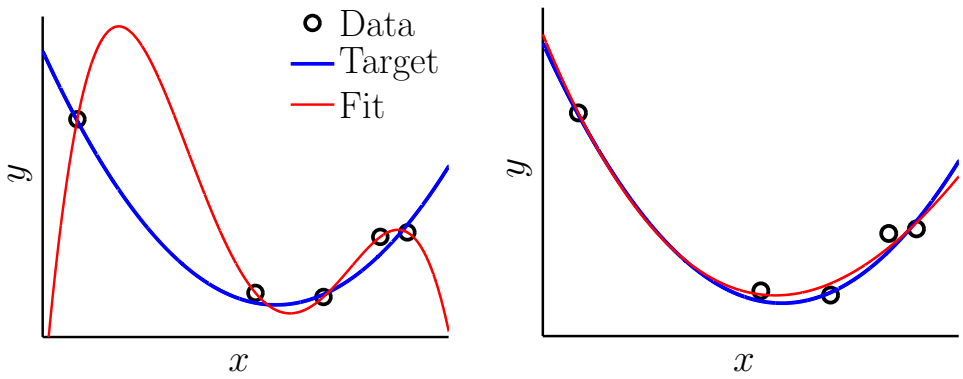


Overfitting

. . . Goes A Long Way

Minimizing $E_{\text{in}}(\mathbf{w}) + \frac{\lambda}{N}\mathbf{w}^{\text{T}}\mathbf{w}$ with different λ's

$\underline{\lambda = 0}$ $\underline{\lambda = 0.0001}$

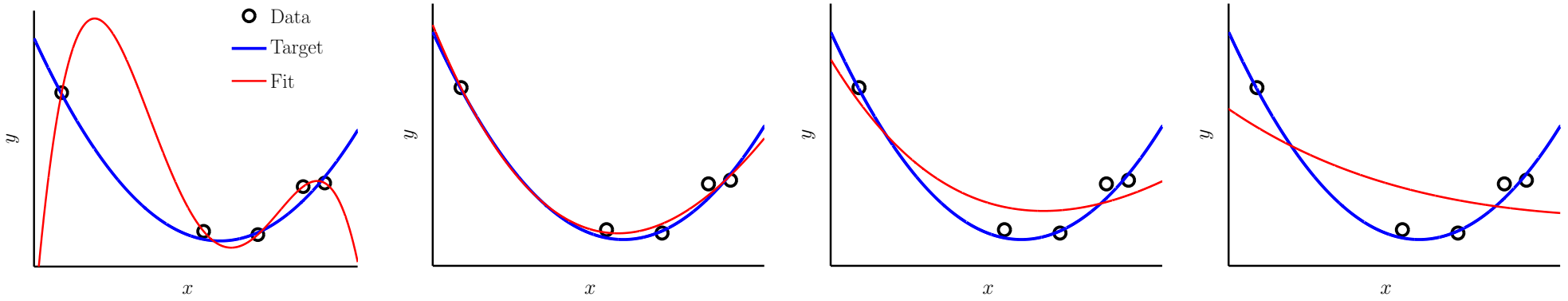


Overfitting **Wow!**

Don't Overdose

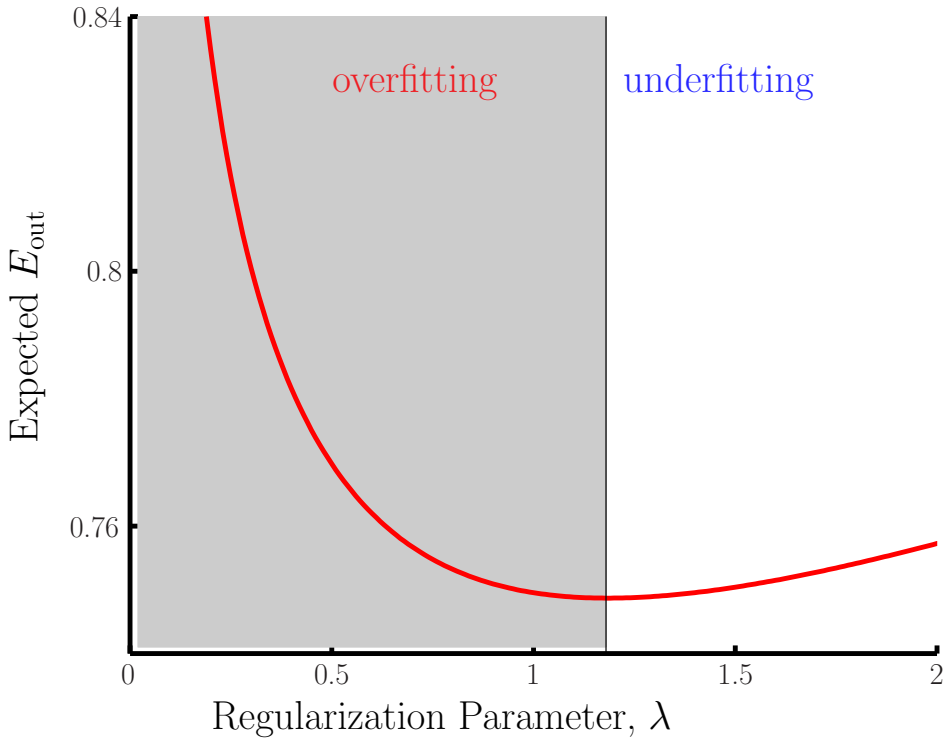
Minimizing $E_{\text{in}}(\mathbf{w}) + \frac{\lambda}{N}\mathbf{w}^{\mathrm{T}}\mathbf{w}$ with different λ's

$\underline{\lambda = 0}$ $\underline{\lambda = 0.0001}$ $\underline{\lambda = 0.01}$ $\underline{\lambda = 1}$

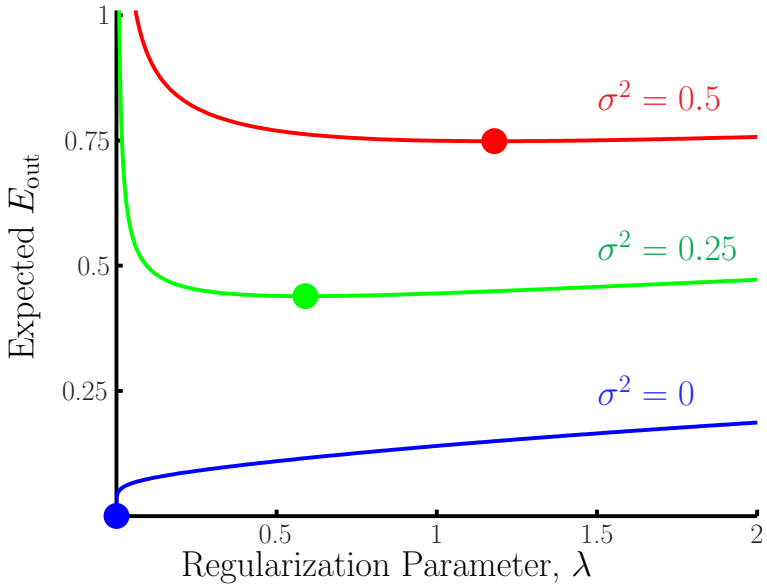


Overfitting $\longrightarrow$ $\longrightarrow$ **Underfitting**

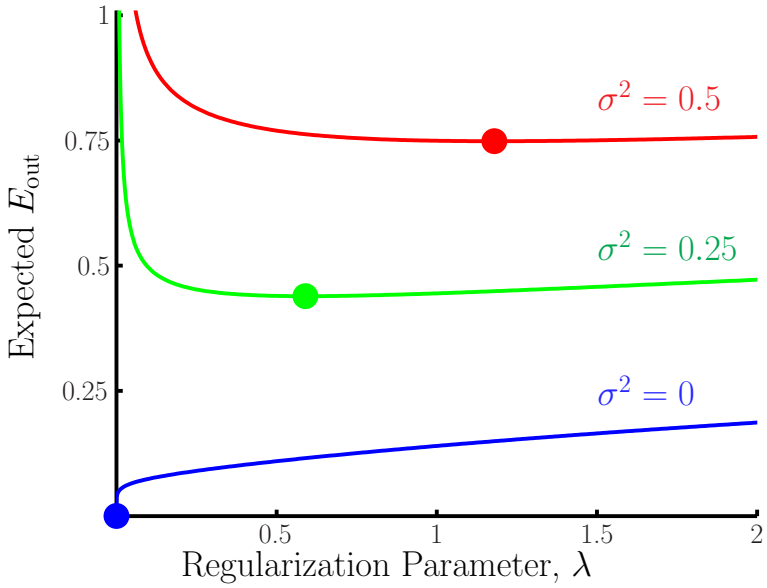
Overfitting and Underfitting

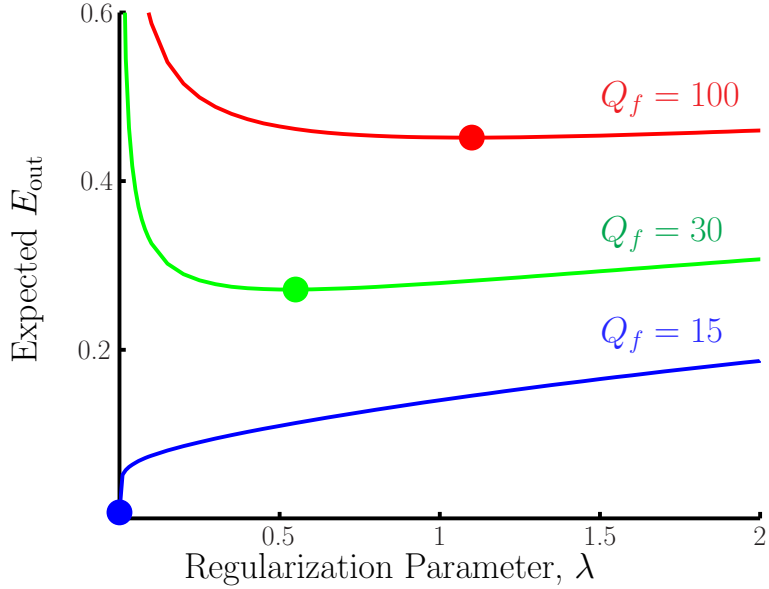


More Noise Needs More Medicine



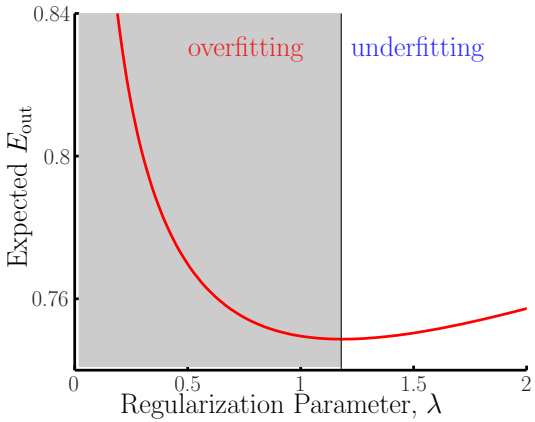
. . . Even For Deterministic Noise





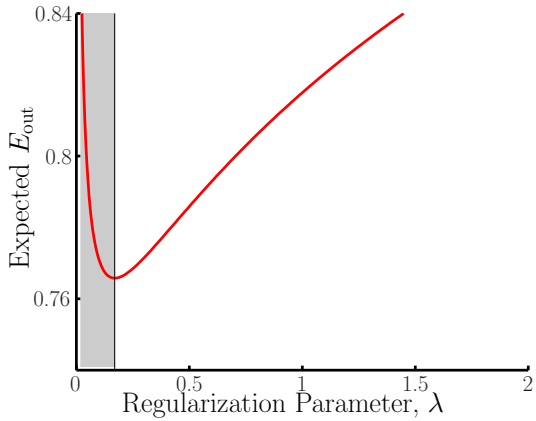
Variations on Weight Decay

Uniform Weight Decay



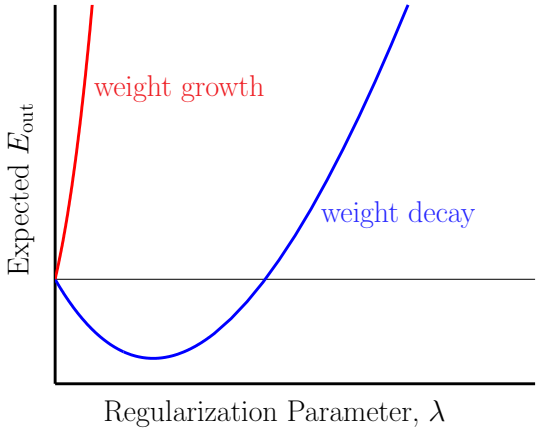
$$\sum_{q=0}^{Q} w_q^2$$

Low Order Fit



$$\sum_{q=0}^{Q} q w_q^2$$

Weight Growth!



$$\sum_{q=0}^{Q} \frac{1}{w_q^2}$$

Choosing a Regularizer – A Practitioner's Guide

The perfect regularizer:

constrain in the 'direction' of the target function.

target function is unknown (going around in circles ☺).

The guiding principle:

constrain in the 'direction' of **smoother** (usually simpler) hypotheses

hurts your ability to fit the 'high frequency' noise

smoother and simpler $\overset{\text{usually means}}{\longrightarrow}$ weight decay not weight growth.

What if you choose the wrong regularizer?

You still have λ to play with — **validation**.

How Does Regularization Work?

Stochastic noise $\longrightarrow$ nothing you can do about that.

Good features $\longrightarrow$ helps to reduce deterministic noise.

Regularization:

- Helps to combat what noise remains, especially when N is small.
- Typical modus operandi: sacrifice a little **bias** for a huge improvement in **var**.
- VC angle: you are using a smaller $\mathcal{H}$ without sacrificing too much E_{in}

Augmented Error as a Proxy for E_{out}

this was $\mathbf{w}^{\mathrm{T}}\mathbf{w}$

$$E_{\text{aug}}(h) \;=\; E_{\text{in}}(h) + \frac{\lambda}{N}\Omega(h)$$

$$\updownarrow$$

$$E_{\text{out}}(h) \;\leq\; E_{\text{in}}(h) + \Omega(\mathcal{H})$$

this was $O\left(\sqrt{\frac{d_{\text{vc}}}{N}\ln N}\right)$

E_{aug} can beat E_{in} as a proxy for E_{out}.

depends on choice of λ