FSAN/ELEG815: Statistical Learning

Gonzalo R. Arce

Department of Electrical and Computer Engineering
University of Delaware

6: Overfitting and Regularization

Outline of the Course

1. Review of Probability
2. Eigen Analysis, Singular Value Decomposition (SVD) and Principal Component Analysis (PCA)
3. The Learning Problem and the VC Dimension
4. Training vs Testing
5. Nonlinear Transformation and Logistic Regression
6. Overfitting and Regularization (Ridge Regression)
7. Lasso Regression
8. Neural Networks
9. Convolutional Neural Networks

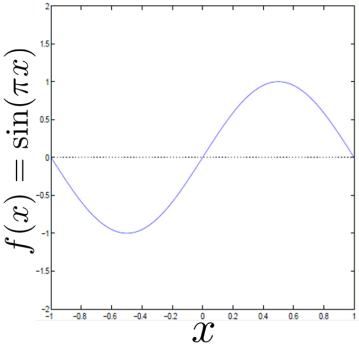
Example: Sine Target

$f : [-1, 1] \to \mathbb{R} \quad f(x) = \sin(\pi x)$ **unknown**

We sample x uniformly in $[-1, 1]$ to generate two training samples ($N = 2$)

Two models used for learning:

$$\mathcal{H}_0 : \quad h(x) = b$$
$$\mathcal{H}_1 : \quad h(x) = ax + b$$

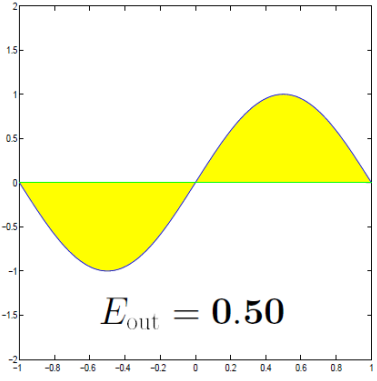


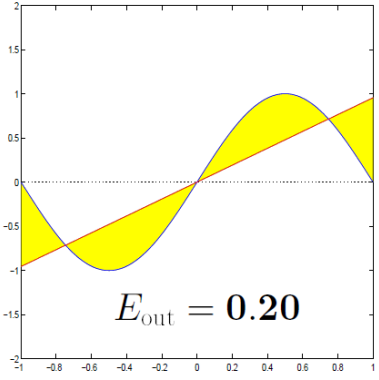
Which is better, $\mathcal{H}_0$ or $\mathcal{H}_1$?

Approximation - $\mathcal{H}_0$ versus $\mathcal{H}_1$

Based on the two models and assuming we know f, try to find the two functions that minimize the squared error:

$$\mathcal{H}_0 : h(x) = b \qquad \mathcal{H}_1 : h(x) = ax + b$$



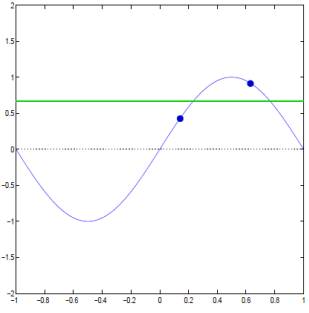


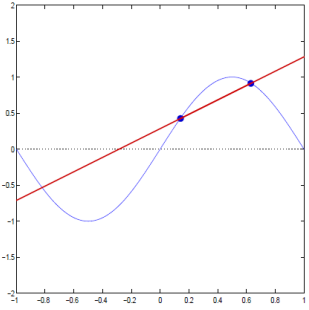
Learning - $\mathcal{H}_0$ versus $\mathcal{H}_1$

In learning, we do not know f. We use the two examples (x_1, y_1),(x_2, y_2) to learn the two functions that best fits the data.

$\mathcal{H}_0$: midpoint $\left(b = \frac{y_1+y2}{2}\right)$

$\mathcal{H}_1$: line passes through the two points

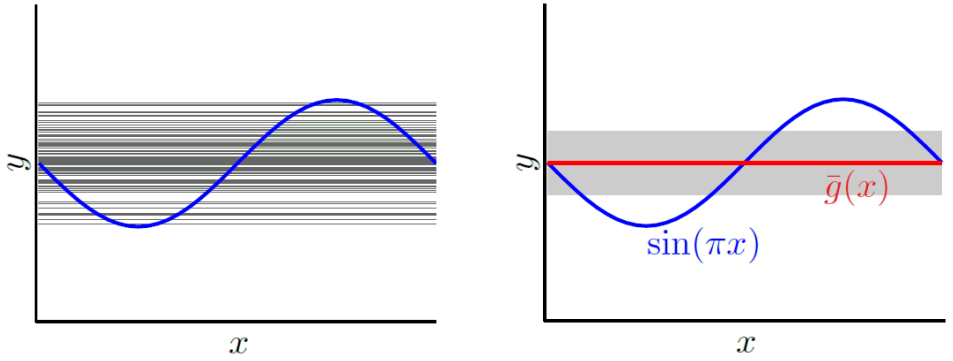




The result varies depending on the data points. We need bias-variance analysis to evaluate our result (considering other possible data sets).

Bias and Variance - $\mathcal{H}_0$

Repeating the process with many data sets, we can estimate the bias and the variance.



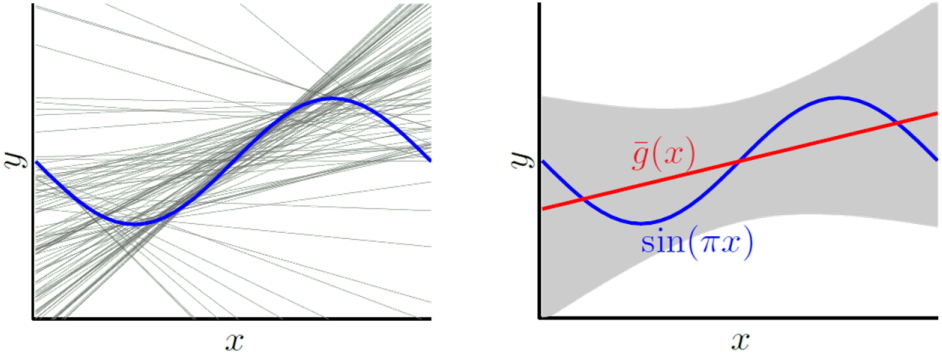
Average hypothesis $\bar{g}(x)$. In this case $\bar{g}(x) \approx 0$ that is close to the best approximation computed using f.

bias: difference between red function $\bar{g}(x)$ and blue function f.

var(x) is indicated by the gray shaded region that is $\bar{g}(x) \pm \sqrt{\textbf{var}(x)}$

Bias and Variance - $\mathcal{H}_1$

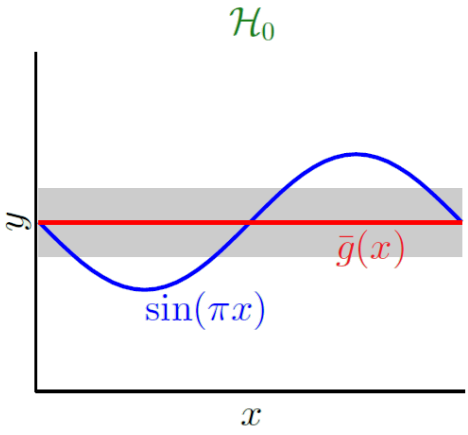
Using the same data sets as before, for the second model we get

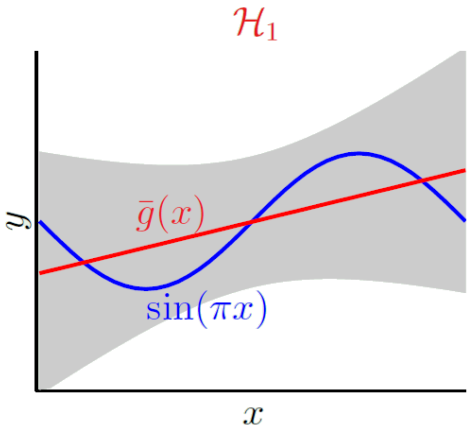


bias: difference between red function $\bar{g}(x)$ and blue function f.

var(x) is indicated by the gray shaded region that is $\bar{g}(x) \pm \sqrt{\textbf{var}(x)}$

The Winner is ...





bias = 0.50 var=0.25 bias=0.21 var=1.69

The simpler model wins by significantly decreasing the **var** at the expense of a smaller increase in **bias**

Lesson Learned

However, the **var** term decreases as N increases, so if we get a bigger data set, the **bias** term will be dominant in E_{out}, and $\mathcal{H}_1$ will win.

Match the '**model complexity**'

to the **data resources**, not to the **target complexity**

Approximation- Generalization Tradeoff

Balance between approximating f in the training data and generalizing on new data.

Goal: small $E_{out} \rightarrow$ good approximation of f out of sample.

More complex $\mathcal{H} \implies$ better chance of **approximating** f

Less complex $\mathcal{H} \implies$ better chance of **generalizing** out of sample

A more complex $\mathcal{H}$ better approximates f, however, it might be more difficult for the algorithm to zoom in on the right hypothesis.

The ideal $\mathcal{H}$ is a singleton hypothesis set containing only the target function.

$$\mathcal{H} = \{f\} \equiv \text{Wining the lottery!}$$

Approximation-Generalization Tradeoff

Two different approaches:

- VC analysis (binary error): $E_{out} \leq E_{in} + \Omega$.
  - $E_{in} \rightarrow$ Approximation
  - $\Omega \rightarrow$ Generalization

  The optimal model is a compromise that minimizes a combination of the two terms.

- Bias-variance analysis (squared error): decomposing E_{out} into
  1. How well $\mathcal{H}$ can approximate f
  2. How well we can zoom in on a good $h \in \mathcal{H}$

  We apply this analysis to **real-valued targets** and use **squared error** (linear regression).

Start with E_{out}

$$E_{out}(g^{(\mathcal{D})}) = \mathbb{E}_{\mathbf{x}}[(g^{(\mathcal{D})}(\mathbf{x}) - f(\mathbf{x}))^2]$$

where $\mathbb{E}_{\mathbf{x}}$ denotes the expected value with respect to $\mathbf{x}$ (based on P on $\mathcal{X}$).

Rid of the dependence on a particular data set by taking the expectation with respect to all data sets:

$$\begin{aligned}\mathbb{E}_{\mathcal{D}}\left[E_{out}(g^{(\mathcal{D})})\right] &= \mathbb{E}_{\mathcal{D}}\left[\mathbb{E}_{\mathbf{x}}[(g^{(\mathcal{D})}(\mathbf{x}) - f(\mathbf{x}))^2]\right] \\ &= \mathbb{E}_{\mathbf{x}}\left[\mathbb{E}_{\mathcal{D}}[(g^{(\mathcal{D})}(\mathbf{x}) - f(\mathbf{x}))^2]\right]\end{aligned}$$

Now, let us focus on:

$$\mathbb{E}_{\mathcal{D}}[(g^{(\mathcal{D})}(\mathbf{x}) - f(\mathbf{x}))^2]$$

The Average Hypothesis

To evaluate $\mathbb{E}_{\mathcal{D}}[(g^{(\mathcal{D})}(\mathbf{x}) - f(\mathbf{x}))^2]$:

We define the 'average' hypothesis $\bar{g}(\mathbf{x})$:

$$\bar{g}(\mathbf{x}) = \mathbb{E}_{\mathcal{D}}[g^{(\mathcal{D})}(\mathbf{x})]$$

Imagine we generate **many** data sets $\mathcal{D}_1, \mathcal{D}_1, \cdots, \mathcal{D}_K$. We can estimate an average function for any $\mathbf{x}$ by

$$\bar{g}(\mathbf{x}) \approx \frac{1}{K} \sum_{k=1}^{K} g^{(\mathcal{D}_k)}(\mathbf{x})$$

$g(\mathbf{x})$ is seen as a RV, with the randomness coming from the randomness in the data set.

For a particular $\mathbf{x}$, $\bar{g}(\mathbf{x})$ is the expectation of this RV.

Using $\bar{g}(\mathbf{x})$

$$\begin{aligned}\mathbb{E}_{\mathcal{D}}[(g^{(\mathcal{D})}(\mathbf{x}) - f(\mathbf{x}))^2] &= \mathbb{E}_{\mathcal{D}}[(g^{(\mathcal{D})}(\mathbf{x}) - \bar{g}(\mathbf{x}) + \bar{g}(\mathbf{x}) - f(\mathbf{x}))^2] \\ &= \mathbb{E}_{\mathcal{D}}[(g^{(\mathcal{D})}(\mathbf{x}) - \bar{g}(\mathbf{x}))^2 + (\bar{g}(\mathbf{x}) - f(\mathbf{x}))^2 \\ &\quad +2(g^{(\mathcal{D})}(\mathbf{x}) - \bar{g}(\mathbf{x}))(\bar{g}(\mathbf{x}) - f(\mathbf{x}))]\end{aligned}$$

Since $\mathbb{E}_{\mathcal{D}}[g^{(\mathcal{D})}(\mathbf{x})] = \bar{g}(\mathbf{x})$, cross term cancels.

$$= \mathbb{E}_{\mathcal{D}}[(g^{(\mathcal{D})}(\mathbf{x}) - \bar{g}(\mathbf{x}))^2] + (\bar{g}(\mathbf{x}) - f(\mathbf{x}))^2$$

Bias and Variance

$$\mathbb{E}_{\mathcal{D}}[(g^{(\mathcal{D})}(\mathbf{x}) - f(\mathbf{x}))^2] = \underbrace{\mathbb{E}_{\mathcal{D}}[(g^{(\mathcal{D})}(\mathbf{x}) - \bar{g}(\mathbf{x}))^2]}_{\text{var}(\mathbf{x})} + \underbrace{(\bar{g}(\mathbf{x}) - f(\mathbf{x}))^2}_{\text{bias}(\mathbf{x})}$$

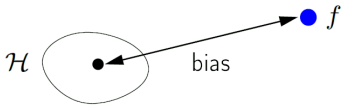
var($\mathbf{x}$) is the variance of the RV $g^{(\mathcal{D})}(\mathbf{x})$ and measures the variation in the final hypothesis depending on the data set.

bias($\mathbf{x}$) measures how much the average function that we would learn using different data sets $\mathcal{D}$ deviates from the target function.

Therefore,

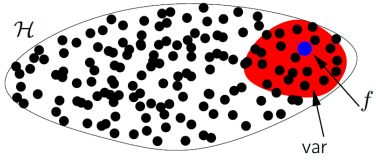
$$\begin{aligned}
\mathbb{E}_{\mathcal{D}}\left[E_{out}(g^{(\mathcal{D})})\right] &= \mathbb{E}_{\mathbf{x}}\left[\mathbb{E}_{\mathcal{D}}[(g^{(\mathcal{D})}(\mathbf{x}) - f(\mathbf{x}))^2]\right] \\
&= \mathbb{E}_{\mathbf{x}}\left[\text{bias}(\mathbf{x}) + \text{var}(\mathbf{x})\right]] \\
&= \textbf{bias} + \textbf{var}
\end{aligned}$$

$$\textbf{bias} = \mathbb{E}_{\mathbf{x}}\left[(\bar{g}(\mathbf{x}) - f(\mathbf{x}))^2\right]$$



Very small model (one hypothesis). The final hypothesis $g^{(\mathcal{D})}$ will be the same as $\bar{g}$, for any data set $\rightarrow$ **var** $= 0$. The **bias** will depend solely on how well this single hypothesis approximates the target f, and unless we are extremely lucky, we expect a large **bias**.

$$\textbf{var} = \mathbb{E}_{\mathbf{x}}\left[\mathbb{E}_{\mathcal{D}}[(g^{(\mathcal{D})}(\mathbf{x}) - \bar{g}(\mathbf{x}))^2]\right]$$



Very large model (all hypothesis). $f \in \mathcal{H}$. Different data sets will lead to different hypotheses that agree with f on the data set, and are spread around f in the red region. Thus, **bias**≈ 0 because $\bar{g}$ is likely to be close to f. The **var** is large (represented by the size of the red region in the figure).

The Tradeoff:

$\mathcal{H}\uparrow$ **bias**$\downarrow$ **var**$\uparrow$

Expected E_{out} and E_{in}

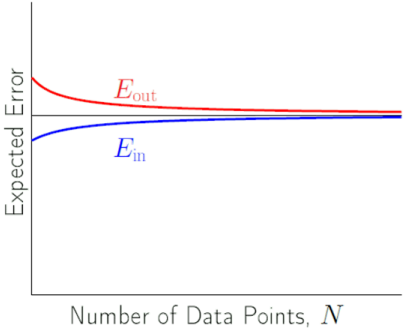
Consider learning with a data set $\mathcal{D}$ of size N,

the final hypothesis has a expected out-of-sample error $\mathbb{E}_{\mathcal{D}}\left[E_{out}(g^{(\mathcal{D})})\right]$ and

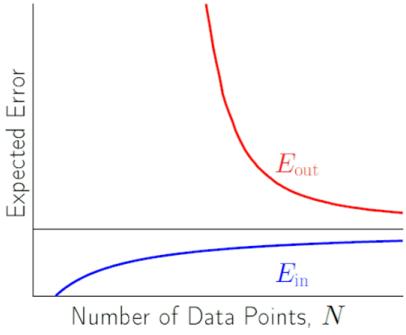
expected in-sample error $\mathbb{E}_{\mathcal{D}}\left[E_{in}(g^{(\mathcal{D})})\right]$

How do they vary with N?

The Curves



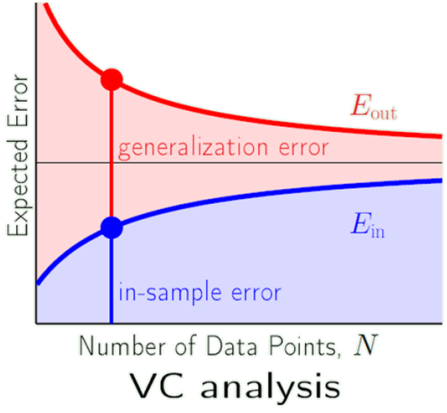
Simple Model

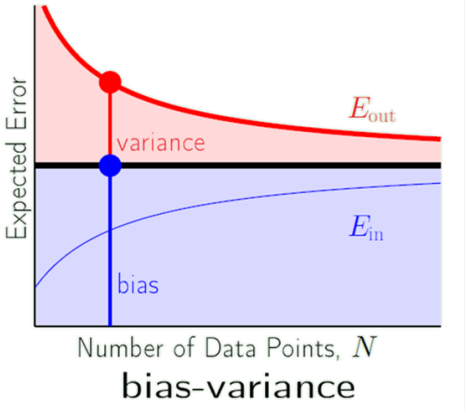


Complex Model

Note: the simple model converges more quickly but to a higher error. In both models, E_{out} decreases while E_{in} increases toward the smallest error the learning model can achieve in approximating f.

VC versus Bias-Variance





In the VC analysis, $E_{out} \leq E_{in} + \Omega$. In the **bias-variance**, it is assumed that, for every N, $\bar{g}$ has the same performance as the best approximation to f in the learning model.

Both capture the tradeoff: Approximation-Generalization

Example - Linear Regression Case

Noisy target $y = f(\mathbf{x}) + \epsilon = \mathbf{w}^T\mathbf{x} + \epsilon$
where ϵ represents noise with zero mean and variance σ^2.

Data set $\mathcal{D} = \{(\mathbf{x}_1, y_1), \cdots, (\mathbf{x}_N, y_N)\}$

Linear regression solution: $\mathbf{w} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y}$

In sample error vector $= \mathbf{X}\mathbf{w} - \mathbf{y}$

Out-of-sample error vector $= \mathbf{X}\mathbf{w} - \mathbf{y}$'

where $\mathbf{y}$' correspond to the output of the target function to the same inputs $\mathbf{x}$ but with a different realization of the noise. $y' = f(\mathbf{x}) + \epsilon'$

Learning Curves for Linear Regression

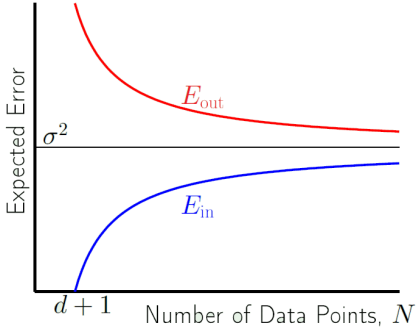
Best approximation error $= \sigma^2$

Expected in-sample error $= \sigma^2\left(1 - \frac{d+1}{N}\right)$

Expected out-of-sample error $= \sigma^2\left(1 + \frac{d+1}{N}\right)$

Expected generalization error $= 2\sigma^2\left(\frac{d+1}{N}\right)$

$d+1 \rightarrow$ VC dimension in perceptron
$d+1 \rightarrow$ 'degrees of freedom' in regression.



Conclusion: the generalization error is a compromise between the 'degrees of freedom' (complexity of the model) and the size of the dataset.

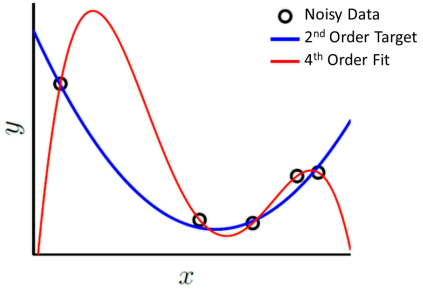
Outline

- What is overfitting?
- The role of noise
- Deterministic noise
- Dealing with overfitting

Illustration of Overfitting

- Simple target function $\rightarrow$ 2nd order polynomial.
- Generate 5 data points (noisy).
- Solve regression problem $\rightarrow$ 5 points fit by a 4th order polynomial.

$$E_{in} = 0$$



However, result does not match the target.

The complex model uses additional degrees of freedom to learn noise.

Overfitting: Process of picking a hypothesis with lower E_{in} and higher E_{out}.

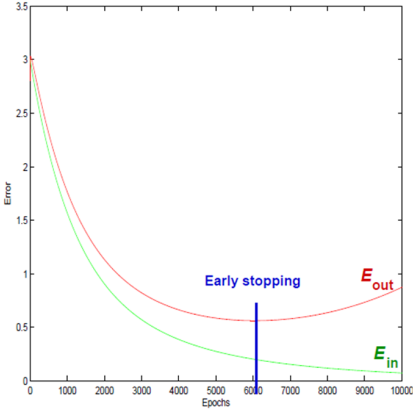
Overfitting vs Bad Generalization

Neural network fitting noisy data:

- Green curve: Running gradient descent and evaluate E_{in} for each epoch.
- Red curve: Use test set to evaluate E_{out} for each epoch.
- Generalization error (difference between the two curves) is increasing.

Overfitting: $E_{in} \downarrow$ $E_{out} \uparrow$

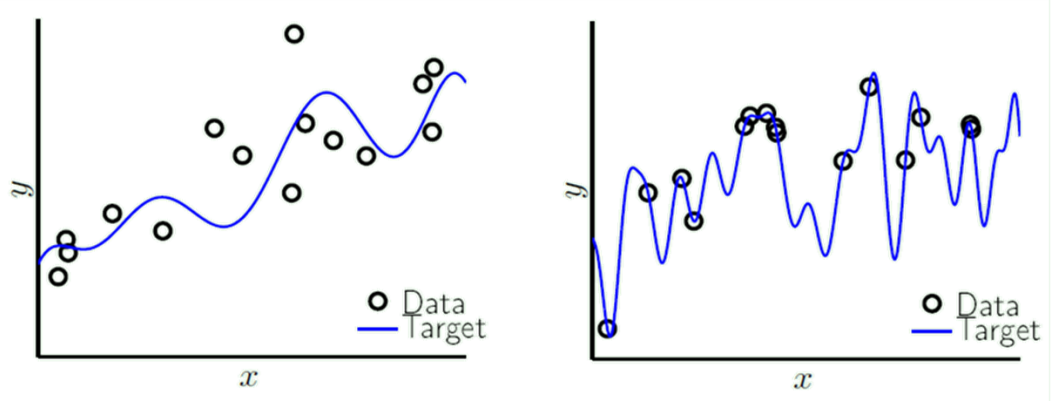
Possible solution: Early stopping



Case Study

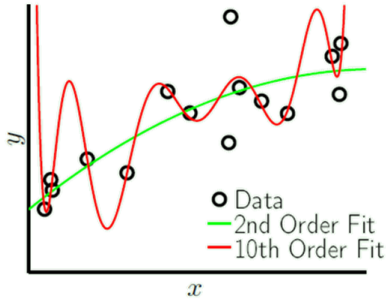
Polynomial regression: $x \to (1, x, x^2, \cdots)$.

- 10th order target function +noise
- 50th order target function (noiseless)



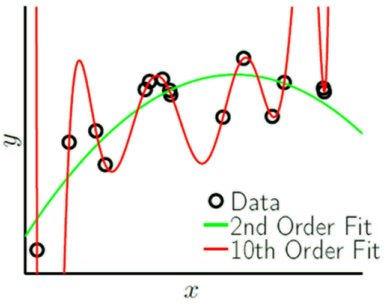
Data set $\mathcal{D}$ contains 15 data points.

Two Fits for Each Target



Noisy low-order target (10th)

	2nd Order	10th Order
E_{in}	0.05	0.034
E_{out}	0.127	9.00



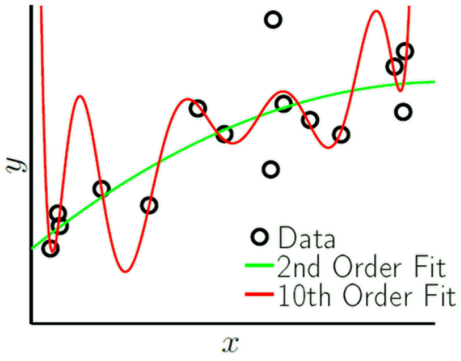
Noiseless high-order target (50th)

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

The 10th order polynomial heavily overfits the data.

An Irony of Two Learners

- Two learners O and R
- They know the target is 10th order.
- O chooses $\mathcal{H}_{10}$
- R chooses $\mathcal{H}_2$.
  - Give up implementing the true target function.
  - Best you can do considering # data points ($N \geq 10 d_{\text{VC}}$)

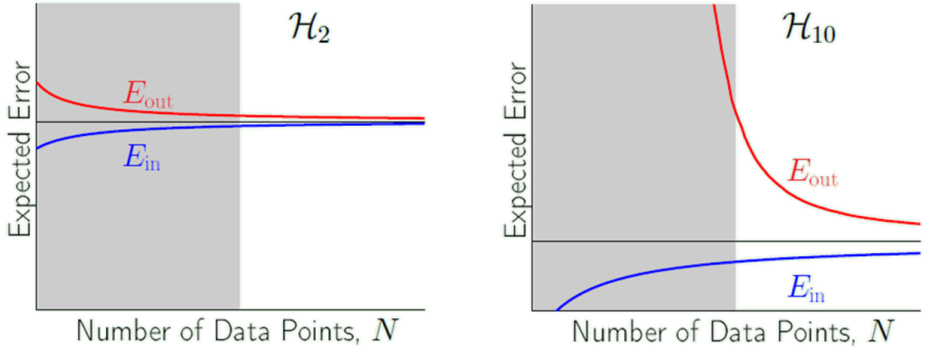


Learning a 10th-order target (noisy)

Match the resources, rather than the target complexity.

Irony: The belief that the best results are obtained by incorporating as much information about the target function as it is available.

Learning Curves



Overfitting is occurring in the shaded region by choosing $\mathcal{H}_{10}$ which has better E_{in} but worse E_{out}.

What matters is how the model complexity matches quantity and quality of the data, instead of only matching the target function.

A Detailed Experiment

Goal: Study impact of **noise level** σ^2, **target complexity** Q_f and **number of data points** N.

$$y = f(x) + \underbrace{\epsilon(x)}_{\sigma^2} = \sum_{q=0}^{Q_f} a_q L_q(x) + \epsilon(x) (*)$$

where $\epsilon(x)$ are iid standard Normal random variables.

Interesting targets $\rightarrow L_i(x)$: increasing complexity polynomials (Legendre polynomials $^{(*)}$). a_q's selected independently from a standard Normal.

$$y = \underbrace{\sum_{q=0}^{Q_f} \alpha_q x^q}_{\text{normalized}} + \epsilon(x) \qquad \alpha_q : \text{ sum of coefficients paired with } x^q$$

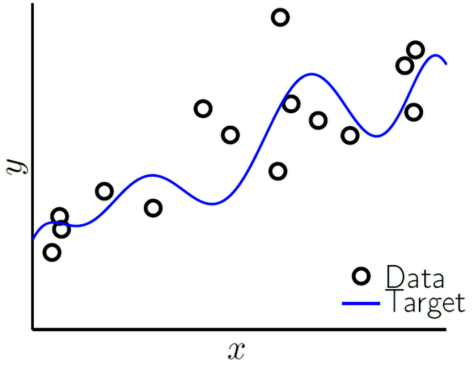
Rescale α_i's so that $\mathbb{E}_{\alpha,x}[f^2] = 1$

$^{(*)}$A Legendre polynomial $L_i(x)$ has specific coefficients such that they are orthogonal.

A Detailed Experiment

Goal: Study impact of **noise level** σ^2, **target complexity** Q_f and **number of data points** N.

Example



$$y = f(x) + \underbrace{\epsilon(x)}_{\sigma^2} = \sum_{q=0}^{Q_f} \alpha_q x^p + \epsilon(x)$$

- Noise level: σ^2
- Target complexity: $Q_f = 10$
- Data size: $N = 15$

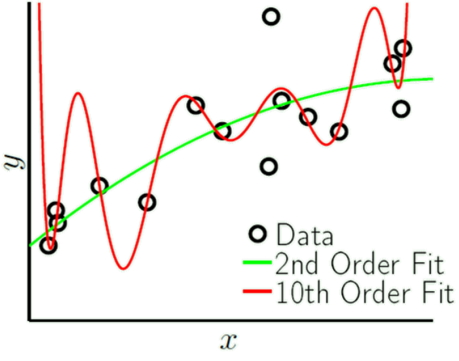
The Results

Fit the data set $(x_1, y_1), \cdots, (x_N, y_N)$ using our two models:

$\mathcal{H}_2$: 2nd-order polynomials

$\mathcal{H}_{10}$: 10th-order polynomials

Target: 10th order polynomial (noisy)



Compare out-of-sample errors of

- $g_2 \in \mathcal{H}_2$
- $g_{10} \in \mathcal{H}_{10}$

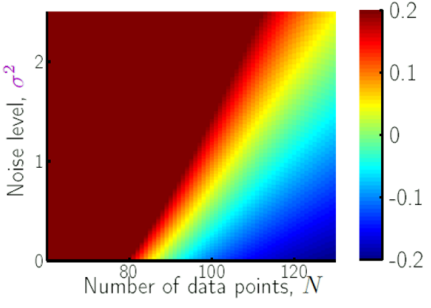
Overfit Measure:

$E_{out}(g_{10}) - E_{out}(g_2)$

More positive $\rightarrow$ More overfitting

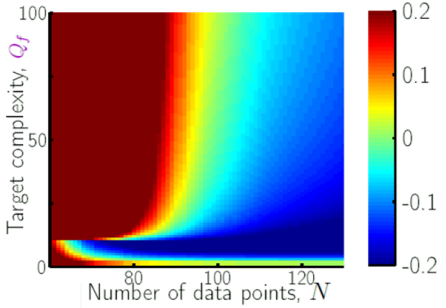
The Results

The colors map to overfit measure: $E_{out}(g_{10}) - E_{out}(g_2)$



Impact of σ^2

Less overfitting when σ^2 drops or N increases ($Q_f = 20$).



Impact of Q_f

Less overfitting when Q_f drops or N increases ($\sigma^2 = 0.1$).

Number of Data Points	$\uparrow$	Overfitting	$\downarrow$
Noise	$\uparrow$	Overfitting	$\uparrow$
Target Complexity	$\uparrow$	Overfitting	$\uparrow$

Definition of Deterministic Noise (DN)

Part of f that $\mathcal{H}$ cannot capture: $f(\mathbf{x}) - h^*(\mathbf{x})$

Why called "noise"?

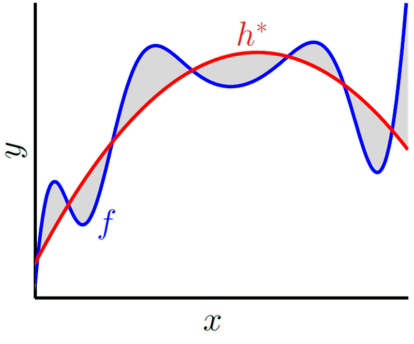
Similarities with stochastic noise:

- It cannot be modeled.
- Trying to learn model it results in overfitting and a spurious final hypothesis.

Differences with stochastic noise:

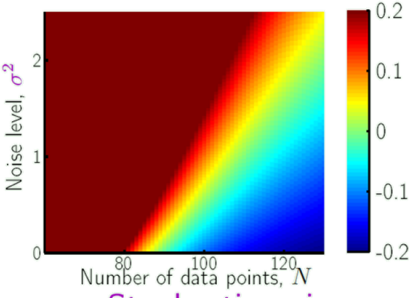
- DN depends on $\mathcal{H}$ ($\uparrow$ Complexity $\downarrow$ DN)
- DN is fixed for a given $\mathbf{x}$.

For a given learning model, there is a best approximation h^* to the target function f.

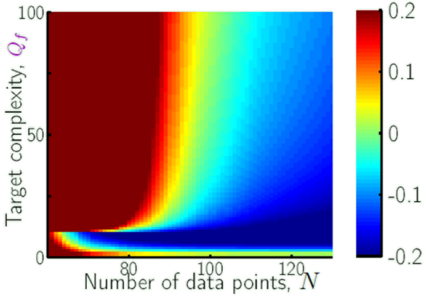


Shading area: Deterministic noise.

Impact of "Noise"



Stochastic noise



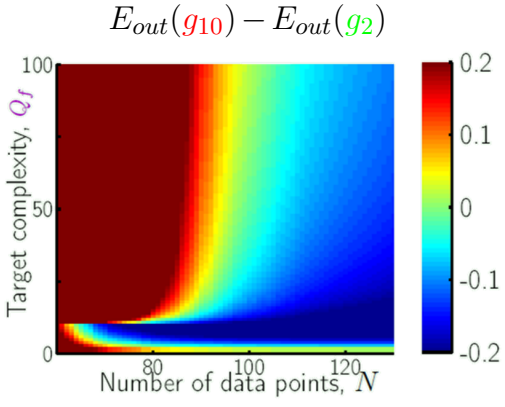
Deterministic noise

Number of Data Points	↑	Overfitting	↓
Stochastic Noise	↑	Overfitting	↑
Deterministic Noise	↑	Overfitting	↑

Deterministic Noise- Impact on Overfitting

Deterministic noise and target complexity Q_f

- As Q_f increases, deterministic noise increases.
- Why overfit starts at $Q_f = 10$? $\mathcal{H}_{10}$ cannot completely capture targets of order greater than 10 (Deterministic Noise).
- For a finite N: $\mathcal{H}$ tries to fit stochastic and deterministic noise.



How much overfit

Noise and Bias-Variance

For f a noiseless target:

$$\mathbb{E}_{\mathcal{D}}[(g^{(\mathcal{D})}(\mathbf{x}) - f(\mathbf{x}))^2] = \underbrace{\mathbb{E}_{\mathcal{D}}[(g^{(\mathcal{D})}(\mathbf{x}) - \bar{g}(\mathbf{x}))^2]}_{\text{var}(\mathbf{x})} + \underbrace{(\bar{g}(\mathbf{x}) - f(\mathbf{x}))^2}_{\text{bias}(\mathbf{x})}$$

- The best approximation h^* to the target function f is approximately the 'average' hypothesis $\bar{g}$.
- What if f is a noisy target?

$$y = f(\mathbf{x}) + \epsilon(\mathbf{x}) \qquad \mathbb{E}[\epsilon(\mathbf{x})] = 0$$

A Noise Term

$$\begin{aligned}\mathbb{E}_{\mathcal{D},\epsilon}[(g^{(\mathcal{D})}(\mathbf{x})-y)^2] &= \mathbb{E}_{\mathcal{D},\epsilon}[(g^{(\mathcal{D})}(\mathbf{x})-f(\mathbf{x})-\epsilon(\mathbf{x}))^2] \\ &= \mathbb{E}_{\mathcal{D},\epsilon}[(g^{(\mathcal{D})}(\mathbf{x})-\bar{g}(\mathbf{x})+\bar{g}(\mathbf{x})-f(\mathbf{x})-\epsilon(\mathbf{x}))^2] \\ &= \mathbb{E}_{\mathcal{D},\epsilon}[(g^{(\mathcal{D})}(\mathbf{x})-\bar{g}(\mathbf{x}))^2+(\bar{g}(\mathbf{x})-f(\mathbf{x}))^2+(\epsilon(\mathbf{x}))^2 \\ &\quad +\text{cross terms}]\end{aligned}$$

Since $\mathbb{E}_{\mathcal{D}}[g^{(\mathcal{D})}(\mathbf{x})]=\bar{g}(\mathbf{x})$, $\mathbb{E}[\epsilon(\mathbf{x})]=0$ and ϵ is independent of others, cross term cancels.
Taking $\mathbb{E}_{\mathbf{x}}[\cdot]$, i.e. average over all input space:

$$\mathbb{E}_{\mathcal{D},\mathbf{x},\epsilon}[(g^{(\mathcal{D})}(\mathbf{x})-y)^2]=\underbrace{\mathbb{E}_{\mathcal{D},\mathbf{x}}[(g^{(\mathcal{D})}(\mathbf{x})-\bar{g}(\mathbf{x}))^2]}_{\text{var}(\mathbf{x})}+\underbrace{\mathbb{E}_{\mathbf{x}}[(\bar{g}(\mathbf{x})-f(\mathbf{x}))^2]}_{\text{bias}(\mathbf{x})}+\underbrace{\mathbb{E}_{\mathbf{x},\epsilon}(\epsilon(\mathbf{x}))^2}_{\sigma^2}$$

Actually, Two Noise Terms

$$\mathbb{E}_{\mathcal{D},\mathbf{x},\epsilon}[(g^{(\mathcal{D})}(\mathbf{x})-y)^2] = \underbrace{\mathbb{E}_{\mathcal{D},\mathbf{x}}[(g^{(\mathcal{D})}(\mathbf{x})-\bar{g}(\mathbf{x}))^2]}_{\text{var}(\mathbf{x})} + \underbrace{\mathbb{E}_{\mathbf{x}}[(\bar{g}(\mathbf{x})-f(\mathbf{x}))^2]}_{\text{bias}(\mathbf{x})} + \underbrace{(\epsilon(\mathbf{x}))^2}_{\sigma^2}$$

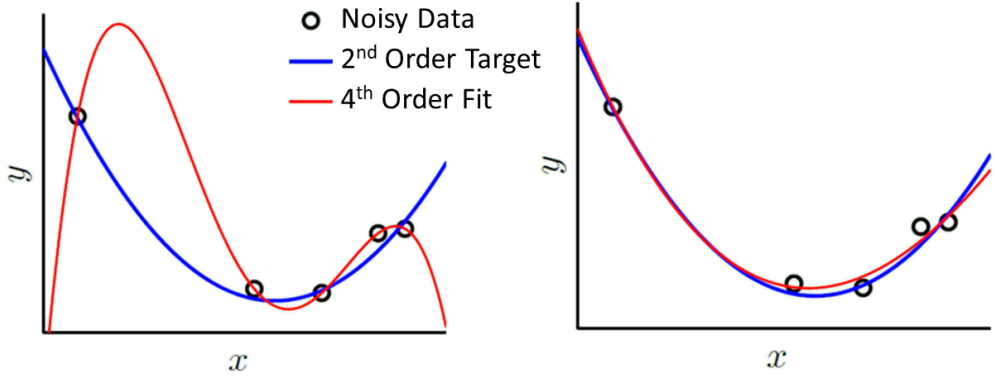
- $\sigma^2 \rightarrow$ Stochastic Noise
- **bias** $\rightarrow$ Deterministic Noise
  Captures model's inability to approximate f.
- **var** $\rightarrow$ Variance of the model
  Captures model's susceptibility to being led in the wrong direction by the two types of noise.

Size of set $N \uparrow$ **var** $\downarrow$.

Given a hypothesis set $\mathcal{H}$, **bias** and σ^2 are fix (irreducible error).

Dealing with Overfitting

- **Regularization:** Putting the brakes.
- **Validation:** Checking the bottom line.



Two Approaches to Regularization

- **Mathematical**:
  - Ill-posed problems in function approximation (solved by smoothness constrains).
  - Bayesian Approach (prior knowledge). Assumptions might not be realistic
- **Heuristic:**
  - Constraining on the minimization of E_{in}

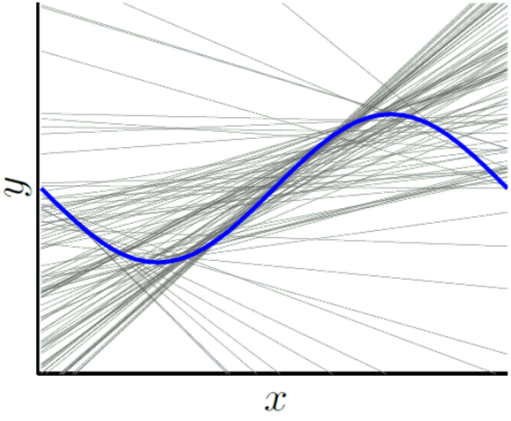
A Familiar Example

$f : [-1, 1] \to \mathbb{R} \quad f(x) = \sin(\pi x)$ **unknown**

We sample x uniformly in $[-1, 1]$ to generate two training samples ($N = 2$)

Two models used for learning:

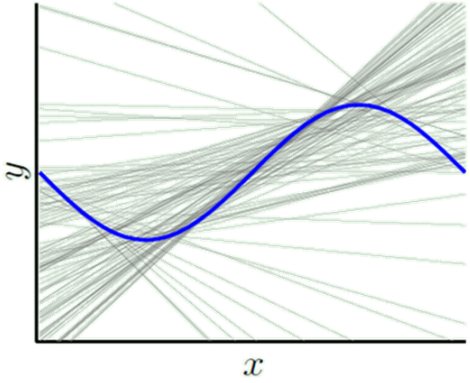
$$\mathcal{H}_0 : \quad h(x) = b$$
$$\mathcal{H}_1 : \quad h(x) = ax + b$$



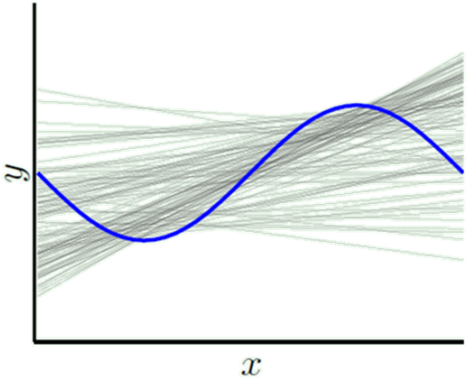
Which was better, $\mathcal{H}_0$ or $\mathcal{H}_1$?

$\mathcal{H}_0$ beats $\mathcal{H}_1$

A Familiar Example



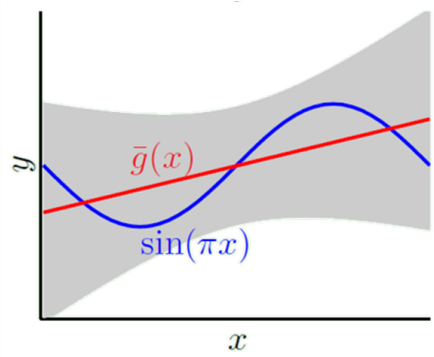
Without Regularization:
Learned function varies extensively depending on the data set.



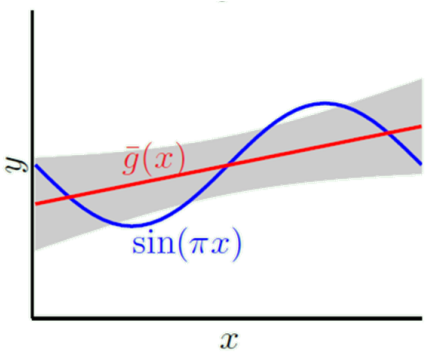
With regularization:
The same data sets are less volatile.

Bias-Variance Decomposition

var(x) gray shaded region $(\bar{g}(x) \pm \sqrt{\textbf{var}(x)})$.



Without Regularization:
bias $= 0.21$ **var** $= 1.69$



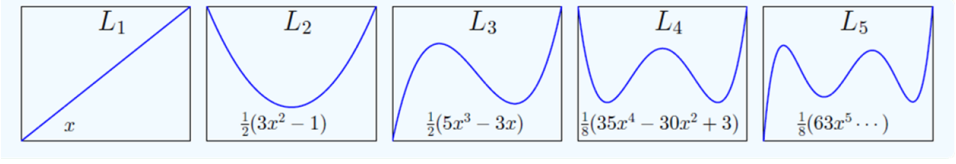
With regularization:
bias $= 0.23$ **var** $= 0.33$.

Regularized $\mathcal{H}_1$ also beats the constant model $\mathcal{H}_0$ (**bias**=0.50, **var**=0.25)

Legendre Polynomials

Standard set of polynomials in one variable $x \in [-1, 1]$ with nice analytic properties:

- Curves get more complex when order increases.
- Orthogonal to each other within $x \in [-1, 1]$.
- Any regular polynomial can be written as a linear combination of Legendre Polynomials.



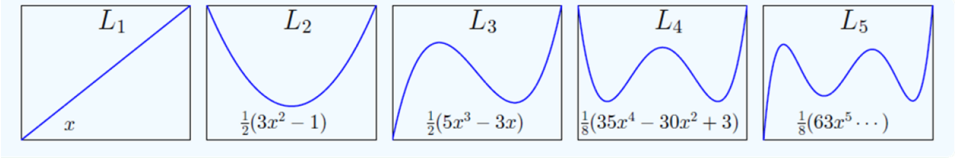
The Polynomial Model

$\mathcal{H}_Q$: polynomials of order Q

$$\mathcal{H}_Q = \left\{ h \middle| h(x) = \mathbf{w}^T \mathbf{z} = \sum_{q=0}^{Q} w_q L_q(x) \right\}_{\mathbf{w} \in \mathbb{R}^{Q+1}}$$

where $\mathbf{z} = [1, L_1(x), \dots L_Q(x)]^T$ (L_q: Legendre Polynomials).

Using Legendre Polynomials, coefficients w_q can be treated as independent (dealing with orthogonal coordinates).



Note: h is linear in $\mathbf{w}$ $\rightarrow$ Apply Linear Regression in $\mathcal{Z}$ space.

Unconstrained Solution

Given $(x_1, y_1), \cdots, (x_N, y_N) \quad \xrightarrow{\Phi} \quad (\mathbf{z}_1, y_1), \cdots, (\mathbf{z}_1, y_1), \cdots, (\mathbf{z}_N, y_N)$

where $\Phi : \mathcal{X} \to \mathcal{Z}$ is a nonlinear transformation.

$$\begin{aligned} E_{in}(\mathbf{w}) &= \frac{1}{N} \sum_{n=1}^{N} (\mathbf{w}^T \mathbf{z}_n - y_n)^2 \\ &= \frac{1}{N} ||\mathbf{Z}\mathbf{w} - \mathbf{y}||_2^2 \\ &= \frac{1}{N} (\mathbf{Z}\mathbf{w} - \mathbf{y})^T (\mathbf{Z}\mathbf{w} - \mathbf{y}) \end{aligned}$$

$$\begin{aligned} \mathbf{w}_{\text{lin}} &= \arg \min_{\mathbf{w} \in \mathbb{R}^{Q+1}} E_{in} \\ &= (\mathbf{Z}^T \mathbf{Z})^{-1} \mathbf{Z}^T y \end{aligned}$$

Constraining the Weights

- Hard constraint: $\mathcal{H}_2$ is constrained version of $\mathcal{H}_{10}$ with $w_q = 0$ for $q > 2$
- Softer version: $\sum_{q=0}^{Q} w_q^2 \leq C$ "soft-order" constraint

  It encourages each weight to be small.

  C determines the amount of regularization.

  Larger C, weaker constraint $\rightarrow$ less regularization.

The optimization problems becomes:

$$\mathbf{w}_{\text{reg}} = \arg\min_{\mathbf{w}} \frac{1}{N}(\mathbf{Z}\mathbf{w} - \mathbf{y})^T(\mathbf{Z}\mathbf{w} - \mathbf{y}) \qquad \text{subject to:} \qquad \mathbf{w}^T\mathbf{w} \leq C$$

$$\mathbf{w}_{\text{reg}} = \arg\min_{\mathbf{w}} \frac{1}{N}(\mathbf{Z}\mathbf{w}-\mathbf{y})^T(\mathbf{Z}\mathbf{w}-\mathbf{y}) \qquad \text{subject to:} \qquad \mathbf{w}^T\mathbf{w} \leq C$$

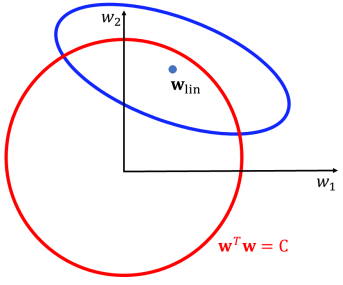
Define the soft-order-constrained hypothesis set $\mathcal{H}(C)$ by:

$$\mathcal{H}(C) = \{h | h(x) = \mathbf{w}^T\mathbf{z}, \mathbf{w}^T\mathbf{w} \leq C\}$$

Goal: Minimize E_{in} over $\mathcal{H}(C)$.

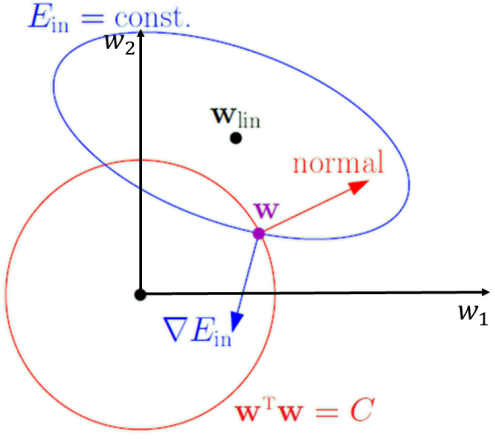
If $\mathbf{w}_{\text{lin}}^T\mathbf{w}_{\text{lin}} \leq C$ then $\mathbf{w}_{\text{reg}} = \mathbf{w}_{\text{lin}}$,

$\mathbf{w}_{\text{lin}} \in \mathcal{H}(C)$
(no regularization).



If $\mathbf{w}_{\text{lin}} \notin \mathcal{H}(C)$, then we minimize E_{in} subject to the equality constraint $\mathbf{w}^T\mathbf{w} = C$

- $\mathbf{w}$ lies on the surface of the sphere $\mathbf{w}^T\mathbf{w} = C$ with normal vector $\mathbf{w}$.
- ∇E_{in} is the normal vector to the quadratic surface of constant E_{in}.



Note that E_{in} is minimum when:

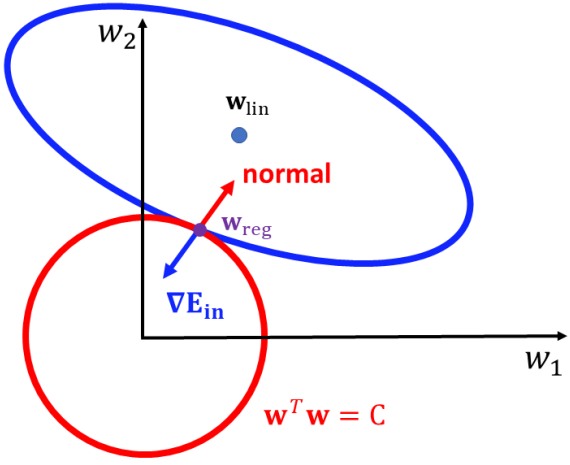
$$\begin{aligned}\nabla E_{in}(\mathbf{w}_{\text{reg}}) &\propto -\mathbf{w}_{\text{reg}} \\ &= -2\frac{\lambda}{N}\mathbf{w}_{\text{reg}}\end{aligned}$$

∇E_{in} must be parallel to $\mathbf{w}_{\text{reg}}$ but in the opposite direction.

$$\nabla E_{in}(\mathbf{w}_{\text{reg}}) + 2\frac{\lambda}{N}\mathbf{w}_{\text{reg}} = 0$$

$$\nabla E_{in}(\mathbf{w}_{\text{reg}} + \frac{\lambda}{N}\mathbf{w}_{\text{reg}}^T\mathbf{w}_{\text{reg}}) = 0$$

since $\nabla \mathbf{w}_{\text{reg}}^T\mathbf{w}_{\text{reg}} = 2\mathbf{w}_{\text{reg}}$



Augmented Error

$$E_{\text{aug}}(\mathbf{w}) = E_{in}(\mathbf{w}) + \frac{\lambda}{N}\mathbf{w}^T\mathbf{w}$$

$\mathbf{w}_{\text{reg}} = \arg\min_{\mathbf{w}} \quad E_{in}(\mathbf{w}) + \frac{\lambda}{N}\mathbf{w}^T\mathbf{w}$ unconditionally (Ridge Regression)

Solves:

$$\mathbf{w}_{\text{reg}} = \arg\min_{\mathbf{w}} \frac{1}{N}(\mathbf{Z}\mathbf{w}-\mathbf{y})^T(\mathbf{Z}\mathbf{w}-\mathbf{y}) \qquad \text{subject to:} \qquad \mathbf{w}^T\mathbf{w} \leq C$$

$$C \uparrow \qquad \lambda \downarrow$$

- $\lambda = 0 \implies C \to \infty$ Least Squares Solution
- $\lambda = \infty \implies C = 0$ $\mathbf{w}_{\text{reg}} = 0$

Ridge Regression

Given the data set $(\mathbf{x}_1, y_1), \cdots, (\mathbf{x}_N, y_N)$, Ridge regression shrinkage fit minimizes a penalized residual sum of squares,

$$\begin{aligned}\hat{\mathbf{w}}^{ridge} &= \arg\min_{\mathbf{w}\in\mathbb{R}^d} \left[\sum_{i=1}^{N}(y_i - w_0 - \sum_{j=1}^{d} x_{ij}w_j)^2 + \lambda\sum_{j=1}^{d} w_j^2\right] \\ &= \arg\min_{\mathbf{w}\in\mathbb{R}^d} \left[\underbrace{||\mathbf{y} - w_0 - \mathbf{X}\mathbf{w}||_2^2}_{\text{Loss}} + \underbrace{\lambda||\mathbf{w}||_2^2}_{\text{Penalty}}\right],\end{aligned}$$

where $||\mathbf{w}||_2$ is the ℓ_2 norm $||\mathbf{w}||_2 = \sqrt{\sum_{j=1}^{d} w_j^2}$.
Here $\lambda \geq 0$ is a tuning parameter, which controls the strength of the penalty term.

- When $\lambda = 0$, we get the linear regression estimate.
- When $\lambda = \infty$, we get $\mathbf{w}^{ridge} = 0$.

Ridge Regression

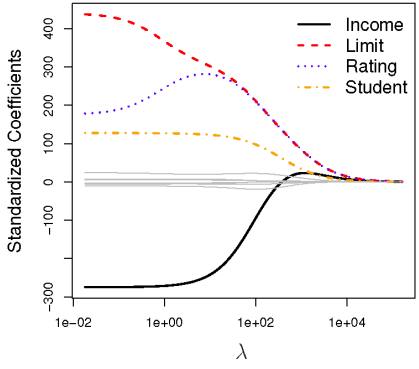
- For λ in between, we balance two ideas: a linear model of $\mathbf{y}$ on $\mathbf{X}$, and shrinking the coefficients.
  Given

$$\mathbf{y} = w_0 + w_1\mathbf{x}_1 + w_2\mathbf{x}_2 + w_3\mathbf{x}_3 + ... + w_{d-1}\mathbf{x}_{d-1} + w_d\mathbf{x}_d + \epsilon.$$

- If the columns of $\mathbf{X}$ are centered, then the intercept estimate is $\hat{w_0} = \bar{y}$, so we usually assume that $\mathbf{y}$, $\mathbf{X}$ have been centered (zero mean) and don't include an intercept.
- The penalty term $||\mathbf{w}||_2^2$ is unfair if the predictor variables are not on the same scale. Variables are not measured in the same units, we typically scale the columns of $\mathbf{X}$ (to have sample variance 1), and then perform ridge regression.

Ridge Regression

Credit data set: **balance** (average credit card debt for a number of individuals), **age**, **cards** (number of credit cards), **education** (years of education), **income** (in thousand of dollars), **limit** (credit limit), and **rating** (credit rating).
Each curve corresponds to estimate for one of the seven variables.



- As $\lambda \uparrow$, the ridge estimates $\hat{w}_k \to 0$.

Ridge Regression

The penalized residual sum of squares (PRSS):

$$\begin{aligned} PRSS &= (\mathbf{y}-\mathbf{X}\mathbf{w})^T(\mathbf{y}-\mathbf{X}\mathbf{w})+\lambda||\mathbf{w}||_2^2 \\ PRSS &= \mathbf{y}^T\mathbf{y}-\mathbf{w}^T\mathbf{X}^T\mathbf{y}+\mathbf{y}\mathbf{X}\mathbf{w}-\mathbf{w}^T\mathbf{X}^T\mathbf{X}\mathbf{w}+\lambda\mathbf{w}^T\mathbf{w} \end{aligned}$$

Differentiating with respect to $\mathbf{w}$, we obtain,

$$\begin{aligned} \frac{\partial PRSS}{\partial \mathbf{w}} &= -2\mathbf{X}^T(\mathbf{y}-\mathbf{X}\mathbf{w})+2\lambda\mathbf{w} \\ \frac{\partial PRSS}{\partial \mathbf{w}} &= -2\mathbf{X}^T\mathbf{y}+2\mathbf{X}^T\mathbf{X}\mathbf{w}+2\lambda\mathbf{w} \end{aligned}$$

PRSS($\mathbf{w}$) is convex. Set the first derivative to zero,

$$\begin{aligned} -2\mathbf{X}^T\mathbf{y}+2\mathbf{X}^T\mathbf{X}\mathbf{w}+2\lambda\mathbf{w} &= 0 \\ \mathbf{X}^T\mathbf{X}\mathbf{w}+\lambda\mathbf{w} &= \mathbf{X}^T\mathbf{y} \\ (\mathbf{X}^T\mathbf{X}+\lambda\mathbf{I})\mathbf{w} &= \mathbf{X}^T\mathbf{y} \\ \hat{\mathbf{w}}^{ridge} &= (\mathbf{X}^T\mathbf{X}+\lambda\mathbf{I})^{-1}\mathbf{X}^T\mathbf{y}. \end{aligned}$$

Ridge Regression

The ridge regression solution is

$$\hat{\mathbf{w}}^{ridge} = (\mathbf{X}^T\mathbf{X} + \lambda\mathbf{I})^{-1}\mathbf{X}^T\mathbf{y}. \quad (1)$$

- Inclusion of λ makes problem non-singular even if $\mathbf{X}^T\mathbf{X}$ is not invertible.
- Solution indexed by the parameter λ
- For each shrinkage λ value, we have a solution.(path of solutions).
- λ controls the size of the coefficients and the amount of regularization.
- As $\lambda \to 0$, we obtain the LS solutions.
- As $\lambda \to \infty$, we have $\hat{\mathbf{w}}^{\text{ridge}}_{\lambda=\infty} = 0$.

Ridge Regression

Setting $\mathbf{R} = \mathbf{X}^T\mathbf{X}$,

$$\begin{aligned}
\hat{\mathbf{w}}_\lambda^{\text{ridge}} &= (\mathbf{X}^T\mathbf{X} + \lambda\mathbf{I}_d)^{-1}\mathbf{X}^T\mathbf{y} \\
&= (\mathbf{R} + \lambda\mathbf{I}_d)^{-1}\mathbf{R}(\mathbf{R}^{-1}\mathbf{X}^T\mathbf{y}) \\
&= (\mathbf{R}(\mathbf{I}_d + \lambda\mathbf{R}^{-1}))^{-1}\mathbf{R}\underbrace{((\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y})}_{\mathbf{w}^{ls}} \\
&= (\mathbf{I}_d + \lambda\mathbf{R}^{-1})^{-1}\mathbf{R}^{-1}\mathbf{R}\hat{\mathbf{w}}^{ls} \\
&= (\mathbf{I}_d + \lambda\mathbf{R}^{-1})^{-1}\hat{\mathbf{w}}^{ls}
\end{aligned}$$

- If $\mathbf{X}$ is orthonormal and $\mathbf{X}^T\mathbf{X} = \mathbf{I}_d$, then:

$$\begin{aligned}
\hat{\mathbf{w}}_\lambda^{ridge} &= (\mathbf{X}^T\mathbf{X} + \lambda\mathbf{I}_d)^{-1}\mathbf{X}^T\mathbf{y} \\
\hat{\mathbf{w}}_\lambda^{ridge} &= (1+\lambda)^{-1}\mathbf{I}_d^{-1}\mathbf{X}\mathbf{y} \\
\hat{\mathbf{w}}_\lambda^{ridge} &= \frac{1}{1+\lambda}\hat{\mathbf{w}}^{ls}.
\end{aligned}$$

VC Formulation - Soft Order Constraint Error Minimization

For a given C, the soft-order constraint corresponds to selecting a hypothesis from the smaller set:

$$\mathcal{H}(C) = \{h | h(x) = \mathbf{w}^T\mathbf{z} \, , \, \mathbf{w}^T\mathbf{w} \leq C\}$$

If $C_1 < C_2$ then $\mathcal{H}(C_1) \subset \mathcal{H}(C_2)$ and so $d_{\text{VC}}(\mathcal{H}(C_1)) \leq d_{\text{VC}}(\mathcal{H}(C_2))$.

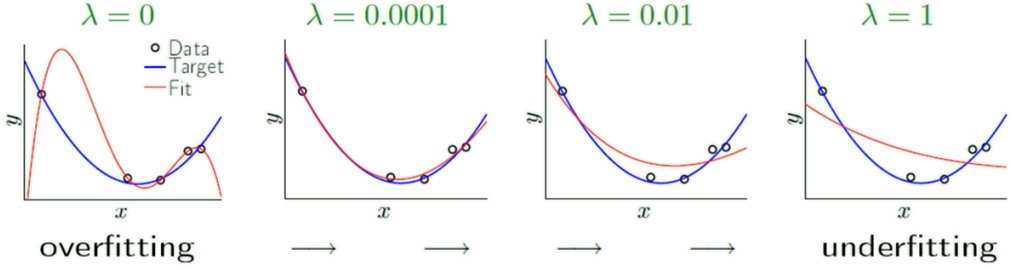
Thus, we expect better generalization with $\mathcal{H}(C_1)$

Conclusion: Better generalization when C decreases (λ increases).

Varying λ

$$\mathbf{w}_{\text{reg}} = \arg\min_{\mathbf{w}} \quad E_{in}(\mathbf{w}) + \frac{\lambda}{N}\mathbf{w}^T\mathbf{w} \qquad \text{for different } \lambda\text{'s}$$

The red fit gets flatter as we increase λ



The optimal regularization parameter typically depends on the data.

Varying the Regularizer

Emphasis of certain weights:

$$\Omega(\mathbf{w}) = \sum_{q=0}^{Q} \gamma_q w_q^2$$

Examples:

- Low-order regularizer

$$\gamma_q = 2^q \qquad \text{Encourages a lower-order fit}$$

- High-order regularizer

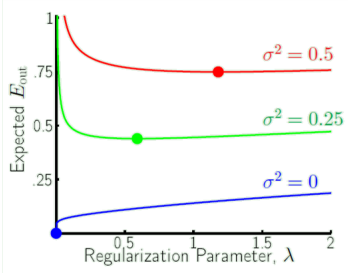
$$\gamma_q = 2^{-q} \qquad \text{Encourages a high-order fit}$$

Tikhonov Regularizer:

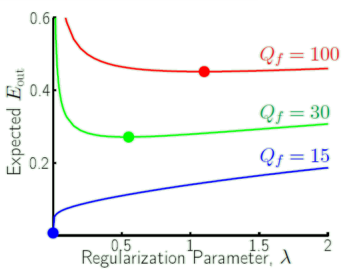
$$\mathbf{w}^T \Gamma^T \Gamma \mathbf{w}$$

The Optimal λ

Performance of the uniform regularizer at different levels of noise.
The optimal λ is highlighted for each curve.
Model: 15th order polynomials ($\mathcal{H}_{15}$).



Stochastic noise



Deterministic noise

Target: 15th order polynomial ($Q_f = 15$). For $\sigma^2 = 0$, regularization is not needed.

Zero stochastic noise ($\sigma = 0$). For $Q_f = 15$, regularization is not needed.

Note: The more noise there is, the more regularization is needed.

Ridge Regression- Prostate cancer example

Correlation between the level of prostate-specific antigen and clinical measures in men who were about to receive a radical prostatectomy: log cancer volume (lcavol), log prostate weight (lweight), age, log of the amount of benign prostatic hyperplasia (lbph), seminal vesicle invasion (svi), log of capsular penetration (lcp), Gleason score (gleason), and percent of Gleason scores 4 or 5 (pgg45). The correlation matrix of the predictors is:

	lcavol	lweight	age	lbph	svi	lcp	gleason
lweight	0.300						
age	0.286	0.317					
lbph	0.063	0.437	0.287				
svi	0.593	0.181	0.129	−0.139			
lcp	0.692	0.157	0.173	−0.089	0.671		
gleason	0.426	0.024	0.366	0.033	0.307	0.476	
pgg45	0.483	0.074	0.276	−0.030	0.481	0.663	0.757

Ridge Regression- Prostate cancer example

Estimated coefficients and test error results, for different subset and shrinkage methods applied to the prostate data. The blank entries correspond to variables omitted.

Term	LS	Best Subset	Ridge	Lasso
Intercept	2.465	2.477	2.452	2.468
lcavol	0.680	0.740	0.420	0.533
lweight	0.263	0.316	0.238	0.169
age	−0.141		−0.046	
lbph	0.210		0.162	0.002
svi	0.305		0.227	0.094
lcp	−0.288		0.000	
gleason	−0.021		0.040	
pgg45	0.267		0.133	
Test Error	0.521	0.492	0.492	0.479
Std Error	0.179	0.143	0.165	0.164

Prediction Error And The Bias-Variance Tradeoff

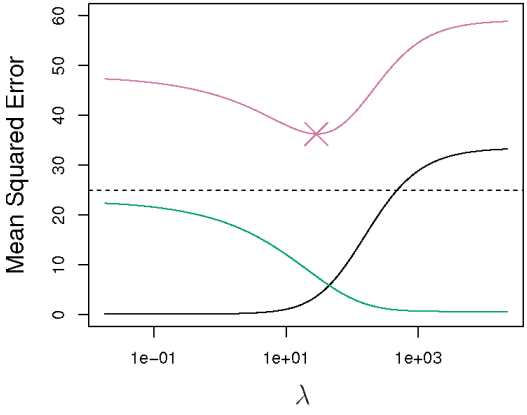
- Good estimators should have small prediction errors.
- Consider the PE at a particular point $\mathbf{x}_0$:

$$\mathrm{PE}(\mathbf{x}_0) = \sigma_\epsilon^2 + \mathrm{Bias}^2(\mathrm{f}(\mathbf{x}_0)) + \mathrm{Var}(\mathrm{f}(\mathbf{x}_0)). \tag{2}$$

- Bias-variance tradeoff.
  - As model becomes more complex, local structure/curvature can be picked up.
  - But coefficient estimates suffer from high variance as more terms are included in the model.
- Introducing a little bias in estimate for β might lead to decrease in variance, and to decrease PE.

Ridge Regression

Bias-variance trade-off.



Squared bias (black), variance (green), and test mean squared error (purple).

- $\lambda = 0$, the variance is high but there is no bias.
- As λ increases, the variance decreases, at the expense of bias.

50-th Year Anniversary of Ridge Regression

Ridge Regression: A Historical Context

Roger W. Hoerl

Department of Mathematics, Union College, Schenectady, NY

ABSTRACT

Two classical articles on Ridge Regression by Arthur Hoerl and Robert Kennard were published in *Technometrics* in 1970, making 2020 their 50th anniversary. The theory and practice of Ridge Regression, and of related biased shrinkage estimators, have been extensively developed over the years. Further, newer shrinkage estimators, such as the Lasso and the Elastic Net, have become popular more recently. These newer developments have led to renewed interest in the original 1970 articles. What has perhaps been lost since 1970 is the context of these classic articles. That is, who were Art Hoerl and Bob Kennard, and what led two statisticians working in the private sector to develop Ridge Regression in the first place? What are the origins of Ridge Regression? Where did the name come from? The purpose of this article is to provide this historical context by discussing the men involved, their work at DuPont, and their approach to methodological development. As Art Hoerl was my father, this is admittedly a personal viewpoint.

ARTICLE HISTORY
Received January 2020
Accepted March 2020

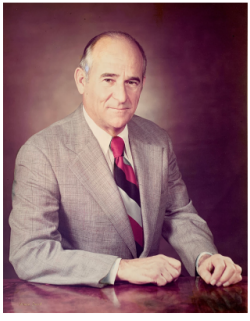
KEYWORDS
Art Hoerl; Bob Kennard; Retrospective; Ridge Analysis

1. Introduction

In 1970, *Technometrics* published two articles by Art Hoerl and Bob Kennard (Hoerl and Kennard 1970a, 1970b) on the topic of Ridge Regression, essentially introducing this methodology to the statistics community. It is fair to say that no one, including the authors, suspected how impactful these articles would ultimately turn out to be. While their proposed estimation approach for collinear data met its share of criticism and resistance (e.g., Draper and Smith 1981), the method not only became common in practice, but also led to further developments in shrinkage estimation, such as Lasso (Tibshirani 1996) and Elastic Net (Zou and Hastie 2005). Research and utilization of these more modern but related methods has renewed interest in the classic Hoerl–Kennard articles. Art Hoerl (hereafter referred to as AH) was my father; I studied under him in graduate school, and subsequently published two articles on Ridge Regression with him (Hoerl, Kennard, and Hoerl 1985; Hoerl, Schuenemeyer, and Hoerl 1986). As might be imagined, we discussed the origins of Ridge Regression quite a bit. I also spent two summers as an intern in the Applied Statistics Group (ASG) at DuPont while in graduate school. By that time, Robert (Bob) Kennard (hereafter referred to as RK) had been promoted outside of the ASG, but was still overseeing it. I met with him several times while working there, in addition to meeting him socially growing up. Ironically, I went to high school with his son Eric. So, I would like to think that I have a unique view into these men's journey to the development of Ridge Regression, which I share below. First, however, let me share some information on these men as individuals.

2. About the Authors

2.1. Who Was Art Hoerl (AH)?

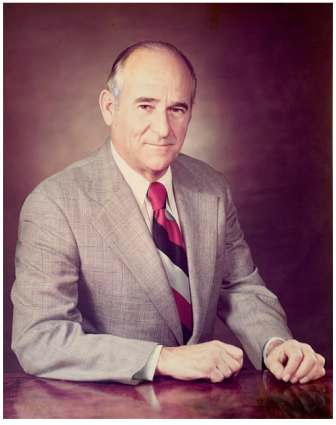


50-th Year Anniversary of Ridge Regression

- In 1970, Technometrics published two articles by Art Hoerl and Bob Kennard on the topic of Ridge Regression introducing this methodology to the statistics community.
- No one, including the authors, suspected how impactful these articles would ultimately turn out to be.
- It led to further developments in shrinkage estimation, such as Lasso (Tibshirani 1996) and Elastic Net (Zou and Hastie 2005).

Who Was Art Hoerl (AH)?

- Received his B.S. in Mechanical Engineering from the University of Southern California (USC) in 1944.
- Upon graduation he was drafted in the army
- Because of his engineering background and high scores on the Army math aptitude test, he was reassigned to the Manhattan Project at Los Alamos, New Mexico, working on bombing tables.
- He reentered USC, receiving an M.S. in mathematics in 1950.
- Upon graduation, he became the first statistician hired by the DuPont Company.
- In 1967, he left DuPont to join the University of Delaware faculty.

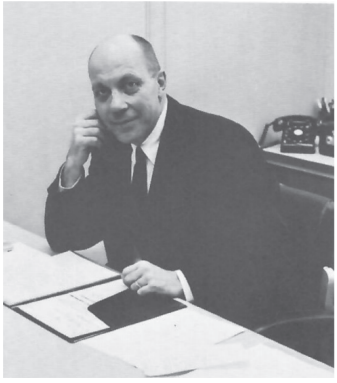


- He retired in 1986, and passed away in 1994.

AH's background in engineering significantly impacted the way he approached problems

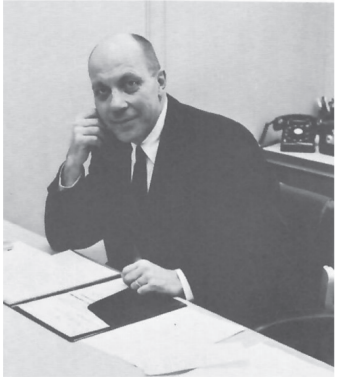
Who Was Bob Kennard (RK)?

- Graduated from Newark High School in 1940. President and Valedictorian at Newark High.
- RK eloped with Helen Elizabeth Staats (Betty) in nearby Elkton, Maryland (known for liberal marriage laws).
- Bob served in the army in World War II. He identified and intercepted Japanese radio signals transmitted in high speed Morse Code.
- He graduated in 1949 with a B.S. in physics, and M.S. in statistics in 1952 at University of Delaware and received his Ph.D. in mathematical statistics at Carnegie Technological University (now Carnegie-Mellon).



Who Was Bob Kennard (RK)?

- RK began his career with DuPont in 1955. Eventually becoming the manager for the Systems Engineering Division, within which the Applied Statistics Group resided.
- He retired in 1982. He taught math and statistics at Lake Sumter Community College for 10 years, and passed away in 2011.



The Hoerl-Kennard team was grounded in engineering problem solving, natural science, and also mathematical statistics. All three viewpoints required in the development of Ridge Regression.

Why is it call Ridge Regression