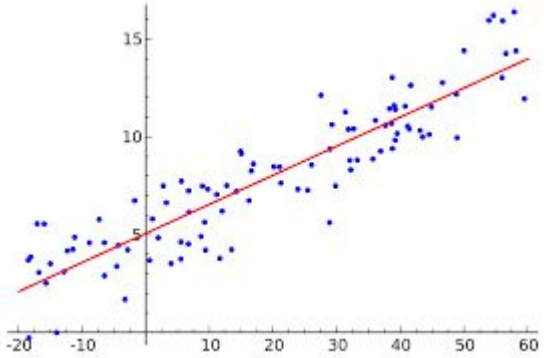
Ridge Regression

Tejasvi Kothapalli, Karna Mendonca,
Brian Zhu, Joe Zou

Ordinary Least Squares (Review)

Featurization

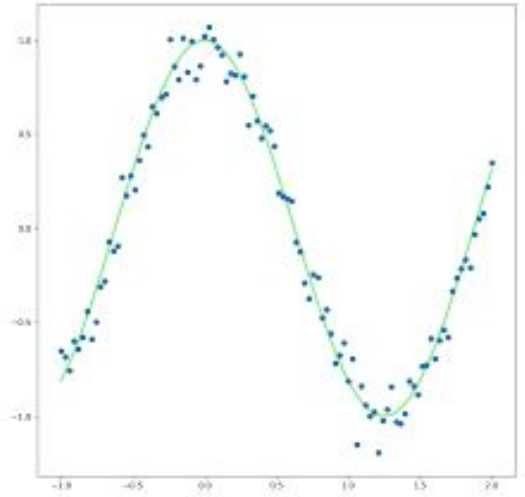
- Recall that Linear Regression essentially finds a “line of best fit”



- How can we model a more complex function, such as a polynomial?

$$y_i = w_d x^d + w_{d-1} x^{d-1} + \cdots + w_2 x^2 + w_1 x + w_0$$

where $w_0 \ldots w_d \in \mathbb{R}$ are constants



Featurization

- Solution:
  - Turn the data point into a vector, with terms raised to a certain degree
  - Turn the coefficients into a weight vector

$$\vec{x_i} = \begin{bmatrix} 1 \\ x_i \\ x_i^2 \\ \vdots \\ x_i^d \end{bmatrix} \qquad w = \begin{bmatrix} w_0 \\ w_1 \\ w_2 \\ \vdots \\ w_d \end{bmatrix} \qquad \vec{x_i}^\top w = w_d x^d + \cdots + w_2 x^2 + w_1 x + w_0 = y_i$$

Featurization

- This allows us to collect all of our data points into a matrix, **X**

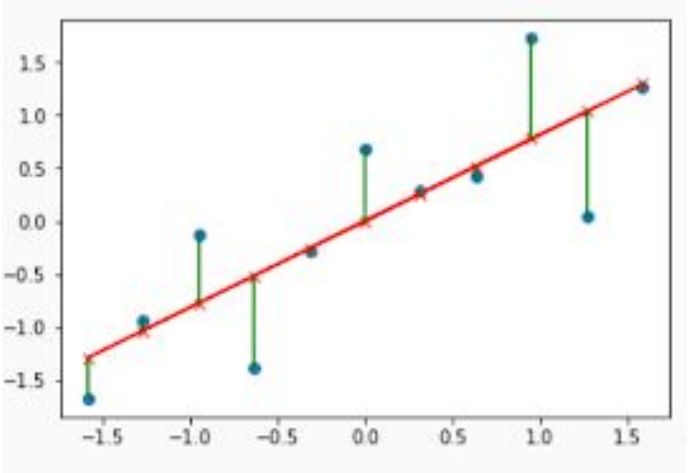
$$X = \begin{bmatrix} -\vec{x_1}^\top - \\ -\vec{x_2}^\top - \\ \vdots \\ -\vec{x_n}^\top - \end{bmatrix} = \begin{bmatrix} 1 & x_1 & x_1^2 & \dots & x_1^d \\ 1 & x_2 & x_2^2 & \dots & x_2^d \\ \vdots & & & & \\ 1 & x_n & x_n^2 & \dots & x_n^d \end{bmatrix}$$

- Thus allowing us to make a prediction: **Xw = y**

$$\begin{bmatrix} -\vec{x_1}^\top - \\ -\vec{x_2}^\top - \\ \vdots \\ -\vec{x_n}^\top - \end{bmatrix} \begin{bmatrix} w_0 \\ w_1 \\ \vdots \\ w_d \end{bmatrix} = \begin{bmatrix} y_0 \\ y_1 \\ \vdots \\ y_d \end{bmatrix}$$

Least Squares

- How can we evaluate how good our models predictions are?



- Solution: Take the square of the residuals

Least Squares

- The sum of our squared residuals is:

$$\sum_{i=0}^{n}(y_i - x_i^\top w)^2 = \|y - Xw\|_2^2$$

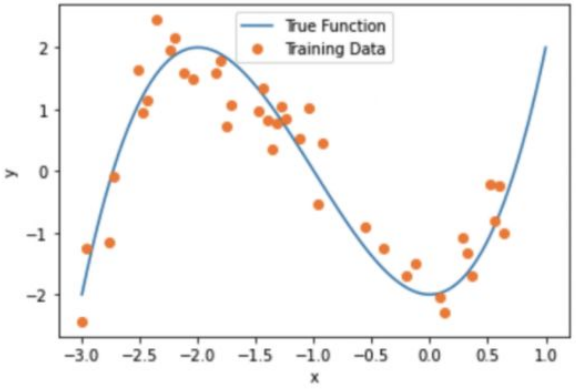
- Thus, finding the optimal **w** simplifies to the following optimization problem:

$$\hat{w} = \arg\min_w \|y - Xw\|_2^2$$

Noise and Overfitting in OLS, Motivation for Ridge

Noise and Overfitting with OLS

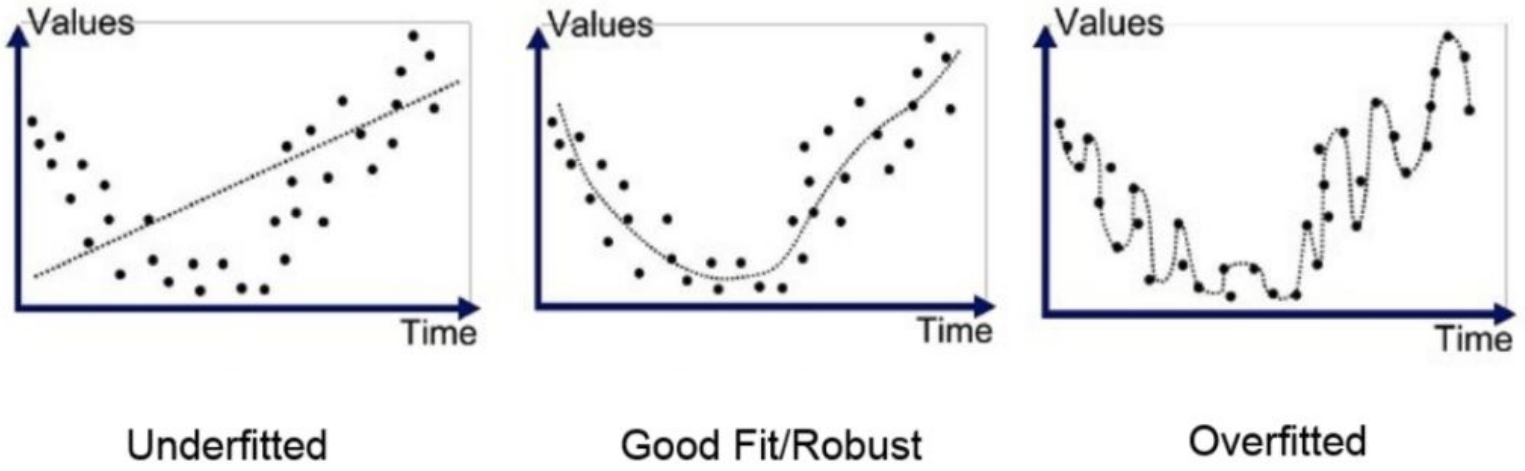
The real world isn't perfect, so we will often have noise in our data like such:



This can lead to a phenomenon called Overfitting, where the learned model fits the training data noise and cannot generalize to other data samples.

Overfitting Example

Our model's complexity affects the quality of our predictions. If the model is too complex, our predictions become too susceptible to noise.



Motivation for Ridge

At a high level, Overfitting often occurs because the learned function is too complex and thus is able to model the noise as well. In order to prevent this from happening, we want to find a way to limit the complexity of the model. We will see next how Ridge Regression can achieve this.

Ridge Regression

Ridge Regression

Ridge Regression is a variation of OLS that penalizes the model's complexity to help prevent overfitting.

Recall the OLS Optimization problem and closed form solution:

Optimization Problem: $\hat{w} = \arg\min_{w} \|y - Xw\|_2^2$

Closed-Form Solution: $\hat{\mathbf{w}} = (\mathbf{X}^\top \mathbf{X})^{-1} \mathbf{X}^\top \mathbf{y}$

Ridge Regression

Remember that the goal of Ridge Regression is to penalize the complexity of the model, thus the optimization problem for Ridge Regression has an extra penalty on the weights:

$$\hat{w} = \arg\min_{w} \|y - Xw\|_2^2 + \lambda \|w\|_2^2$$

This will net us a closed form solution of:

$$\hat{\mathbf{w}} = (\mathbf{X}^\top \mathbf{X} + \lambda \mathbf{I})^{-1} \mathbf{X}^\top \mathbf{y}$$

We can make predictions the same way: $\hat{\mathbf{y}} = \mathbf{X}\hat{\mathbf{w}}$

Ridge Regression Derivation

$$\hat{w} = \arg\min_{w} \|y - Xw\|_2^2 + \lambda\|w\|_2^2 = w^\top X^\top X w - 2w^\top X^\top y + y^\top y + \lambda w^\top w$$

Once again, our loss function is convex with respect to w. So we take the gradient and set it to 0:

$$\frac{\partial \mathcal{L}}{\partial w} = 2X^\top X w - 2X^\top y + 2\lambda w = 0$$

Now we simply isolate w:

$$(X^\top X + \lambda I)w = X^\top y$$

$$\hat{w} = (X^\top X + \lambda I)^{-1} X^\top y$$

Ridge Regression

Lambda is a hyperparameter that we can choose using cross-validation.

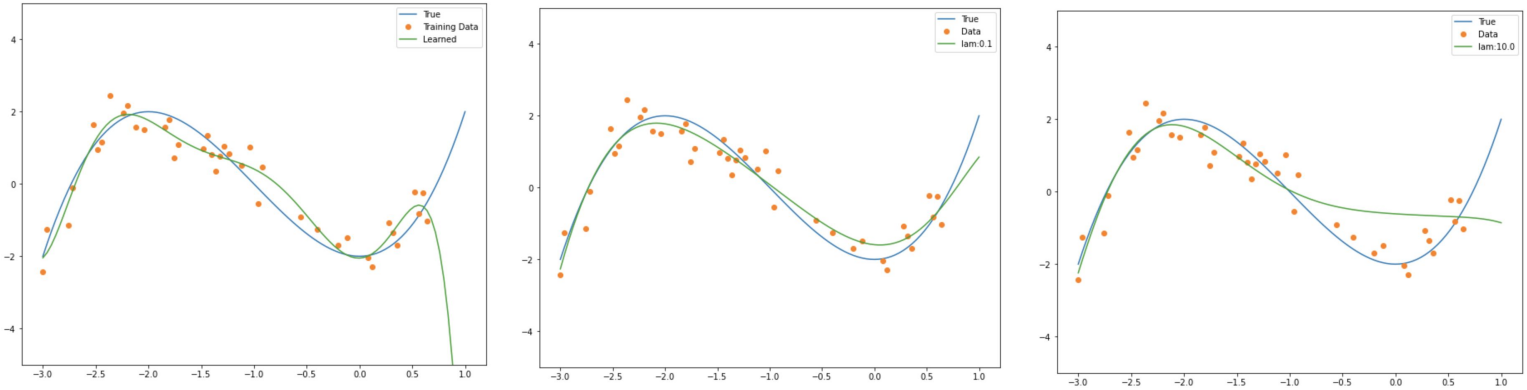


Figure 2: Ridge Regression: $\lambda = \{0\ (\text{OLS}), 0.1, 10\}$

Ridge Regression with Scikit-learn

Scikit-learn is an open source machine learning package for Python that comes with a sklearn.linear_model.Ridge class for Ridge Regression.

Here is an example of the code implementation given an X matrix and y vector

```
#Specify a lambda to use
lambd = 0.1
ridge = Ridge(alpha=lambd)
ridge.fit(X, y)
y_pred = ridge.predict(X)
```

Documentation at: https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.Ridge.html

Bias-Variance Trade-Off

Bias-Variance Analysis for Ridge Regression

Goal: $\min_{\lambda} \text{bias}^2 + \text{variance}$

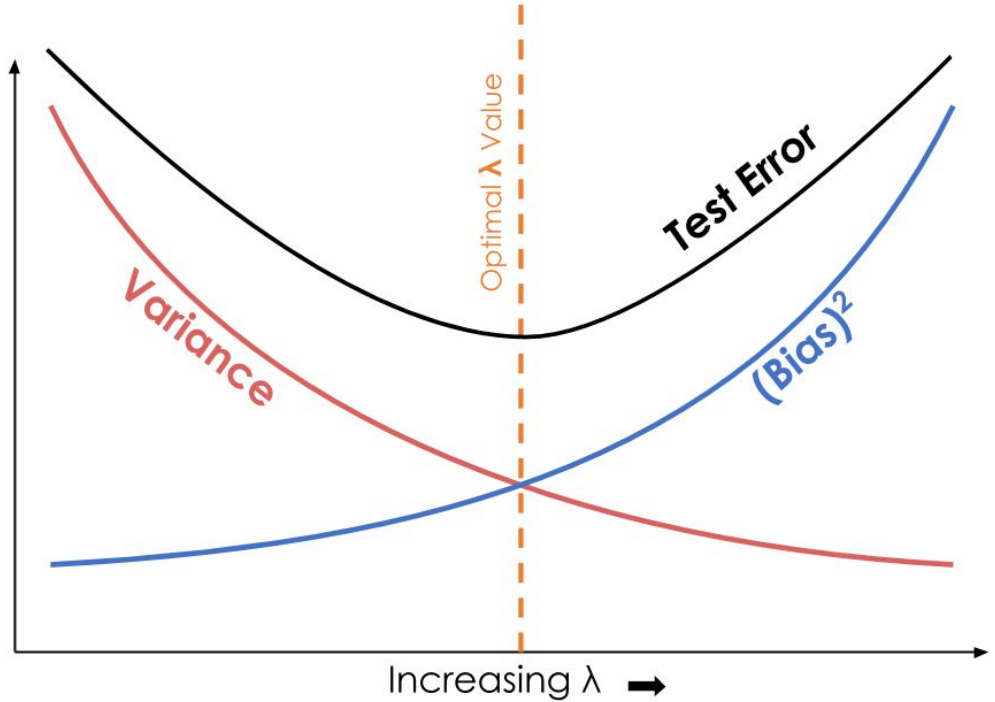
Bias:

- high bias means lack of ability to capture patterns in data
- high bias occurs when λ is large (forces $|\hat{w}|_2 \to 0$)

Variance:

- high variance means model is susceptible to small changes in training data
- high variance occurs when λ is small ($\hat{w}$ will try to fit the noise in data)

Bias-Variance Plot



- optimal λ occurs where $(bias)^2$ + variance is minimized
- minimum test error also occurs where $(bias)^2$ + variance is minimized

Eigenvalue Perspective

Instability in OLS

- Let’s try to understand the mathematical perspective of overfitting

- Recall our solution to OLS: $\hat{\mathbf{w}} = (\mathbf{X}^\top \mathbf{X})^{-1} \mathbf{X}^\top \mathbf{y}$

    - The term $(\mathbf{X}^\top \mathbf{X})^{-1}$ can cause instability in our solution. Why might this be?

Instability in OLS

- Let's look at the eigendecomposition of $\mathbf{X}^\top\mathbf{X}$:

$$X^\top X = V\Lambda V^{-1}$$

$$\Lambda = \begin{bmatrix} \tilde{\lambda}_1 & 0 & \cdots & 0 \\ 0 & \tilde{\lambda}_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \tilde{\lambda}_n \end{bmatrix}$$

Instability in OLS

- Now let's look at $(\mathbf{X}^\top \mathbf{X})^{-1}$:

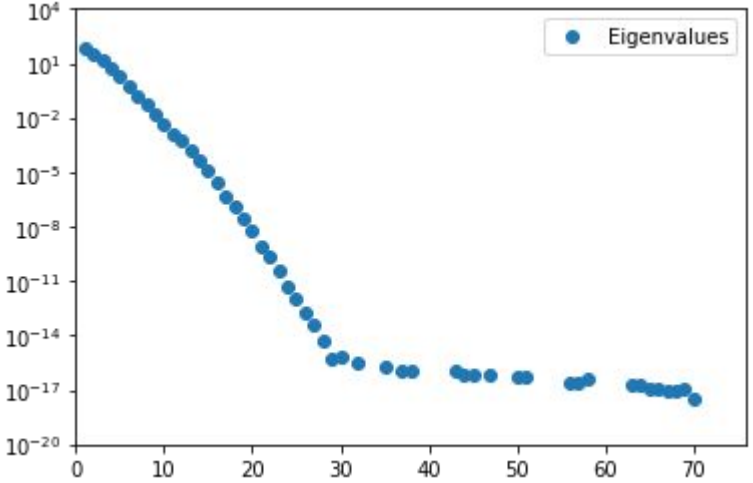
$$(X^\top X)^{-1} = V^{-1}\Lambda^{-1}V$$

$$\Lambda^{-1} = \begin{bmatrix} \frac{1}{\tilde{\lambda}_1} & 0 & \cdots & 0 \\ 0 & \frac{1}{\tilde{\lambda}_2} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \frac{1}{\tilde{\lambda}_n} \end{bmatrix}$$

- Do you notice something that could go wrong?

Instability in OLS

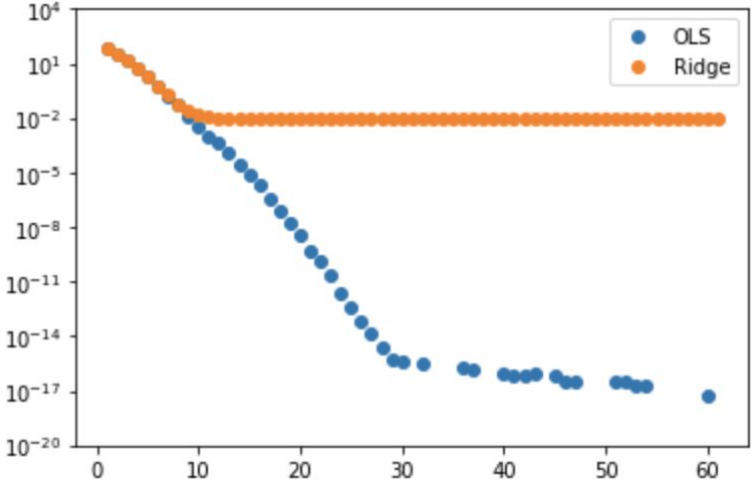
- Let's graph the eigenvalues of $\mathbf{X}^\top \mathbf{X}$:



- Will taking the inverse cause problems?

Instability in Ridge?

- Now let's graph the eigenvalues of $\mathbf{X}^\top \mathbf{X} + \lambda \mathbf{I}$:



- Will taking the inverse cause any problems?

Instability in Ridge?

- We see that there is a lower bound on the eigenvalues in Ridge Regression
- This will help stabilize the inverse in

$$\hat{w} = (X^\top X + \lambda I)^{-1} X^\top y$$

- Small changes in y due to noise will not cause $\hat{w}$ to change drastically now, making ridge regression more robust than OLS.

Aside

- We can argue that ridge regression acts much like a high-pass filter
- You can learn more about it in the notes

Alternate Solution to Ridge Regression, Fake Data/Features Perspectives

Alternative Solution to Ridge Regression

Recall how the closed-form solution for Ridge Regression parallels the closed-form solution of OLS:

Ridge: $\hat{w} = (X^\top X + \lambda I)^{-1} X^\top y.$

OLS: $\hat{w} = (X^\top X)^{-1} X^\top y.$

Similarly, there is an alternative solution for Ridge Regression which parallels the minimum-norm solution.

Recall the minimum-norm solution: $\hat{w} = X^\top (X X^\top)^{-1} y$

Alternative Solution to Ridge Regression

The alternative solution to Ridge Regression is:

$$\hat{w} = X^\top (XX^\top + \lambda I)^{-1} y$$

- This solution will net you the same result as standard ridge regression, but it will be useful in the future when you learn about Kernel's.

Fake Data and Fake Features Perspectives

Next, we will see how the closed-form solution to Ridge Regression is like adding Fake Data points and the alternative closed-form solution to Ridge Regression is like adding Fake Features.

Fake Data

Let's modify our X and y data points with fake data such that:

$$\hat{X} = \begin{bmatrix} X \\ \sqrt{\lambda} I \end{bmatrix}$$

$$\hat{y} = \begin{bmatrix} \vec{y} \\ 0 \end{bmatrix}$$

Now, notice that when we run OLS, we get the same result as the closed form solution for Ridge Regression:

$$\hat{w} = (\hat{X}^\top \hat{X}) \hat{X}^\top \hat{y}$$

$$\hat{w} = ([X^\top \sqrt{\lambda} I] \begin{bmatrix} X \\ \sqrt{\lambda} I \end{bmatrix})^{-1} [X \sqrt{\lambda} I] \begin{bmatrix} \vec{y} \\ 0 \end{bmatrix}$$

$$\hat{w} = (X^\top X + \lambda I)^{-1} y$$

Fake Features

Let's modify our X data points with fake features such that: $\hat{X} = [X \sqrt{\lambda} I]$

Notice that because there are N additional features now, we will have N additional entries in our weight vector: $\begin{bmatrix} \hat{w} \\ \hat{\epsilon} \end{bmatrix}$

Since the X matrix is now a wide matrix, we must use the minimum-norm solution to find our weight vector.

Fake Features

Using the minimum-norm solution, we can see that we will obtain the same ŵ as the alternative ridge regression solution.

$$\begin{bmatrix} \hat{w} \\ \hat{\epsilon} \end{bmatrix} = \hat{X}^\top (\hat{X}\hat{X}^\top)^{-1}\vec{y}$$

$$\begin{bmatrix} \hat{w} \\ \hat{\epsilon} \end{bmatrix} = \begin{bmatrix} X^\top \\ \sqrt{\lambda}I \end{bmatrix} ([X\sqrt{\lambda}I] \begin{bmatrix} X^\top \\ \sqrt{\lambda}I \end{bmatrix})^{-1}\vec{y}$$

$$\begin{bmatrix} \hat{w} \\ \hat{\epsilon} \end{bmatrix} = \begin{bmatrix} X^\top \\ \sqrt{\lambda}I \end{bmatrix} (XX^\top + \lambda I)^{-1}\vec{y}$$

$$\hat{w} = X^\top (XX^\top + \lambda I)^{-1}\vec{y}$$