CSE/STAT 416

Regularization – Ridge Regression

Hunter Schafer
University of Washington
July 1, 2019

Administrivia

Homework 1 Due tomorrow night!

Remember to do all 3 parts

- [A1: Concept]
- [A1: Programming]
- [A1: Upload]

No section on Thursday! 4th of July!

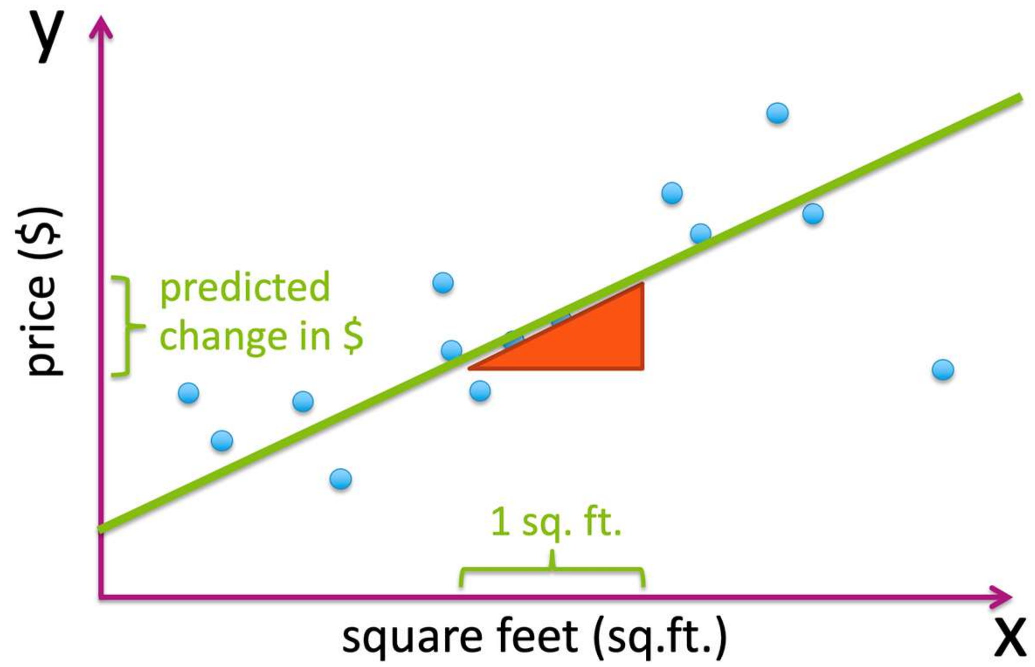
HW2 goes out on Wednesday and is due next Tuesday (like regular)

HW assignments are weighted equally!

Interpreting Coefficients

Interpreting Coefficients – Simple Linear Regression

$$\hat{y} = \hat{w}_0 + \hat{w}_1 x$$

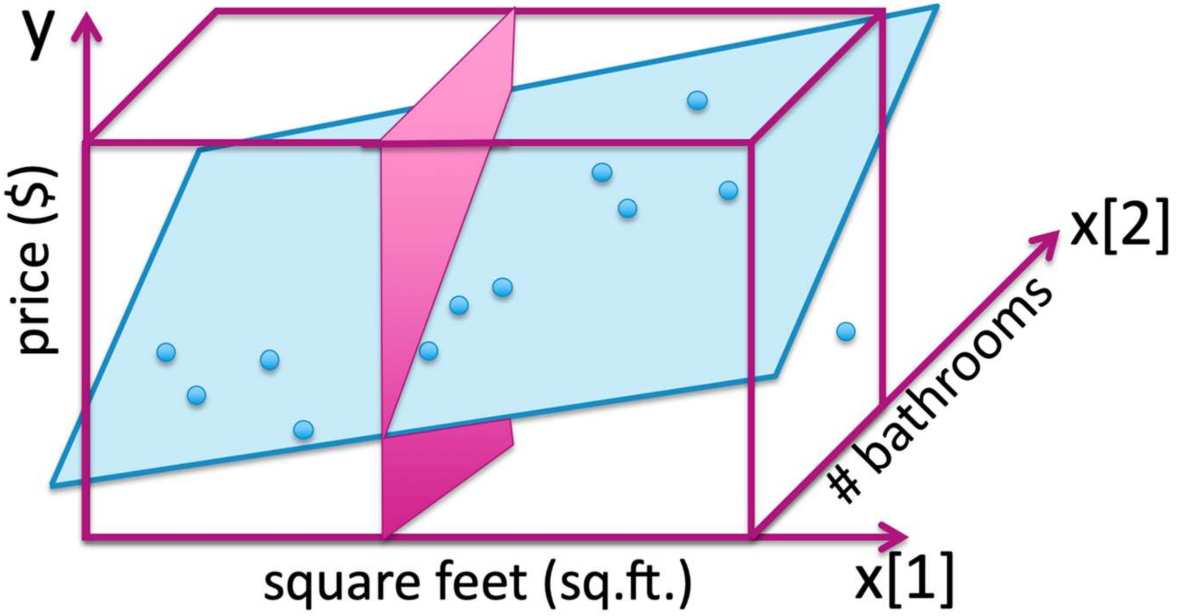


Interpreting Coefficients

Interpreting Coefficients – Multiple Linear Regression

$$\hat{y} = \hat{w}_0 + \hat{w}_1 x[1] + \hat{w}_2 x[2]$$

Fix



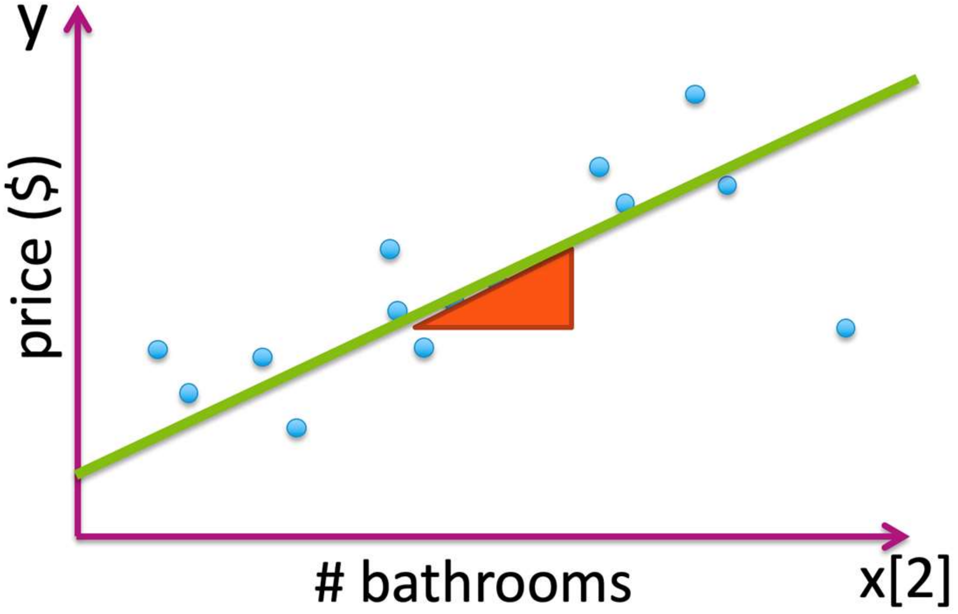
Interpreting Coefficients

Interpreting Coefficients – Multiple Linear Regression

$$\hat{y} = \hat{w}_0 + \hat{w}_1 x[1] + \hat{w}_2 x[2]$$

Fix

Holding x[1] fixed!



Interpreting Coefficients

This also extends for multiple regression with many features!

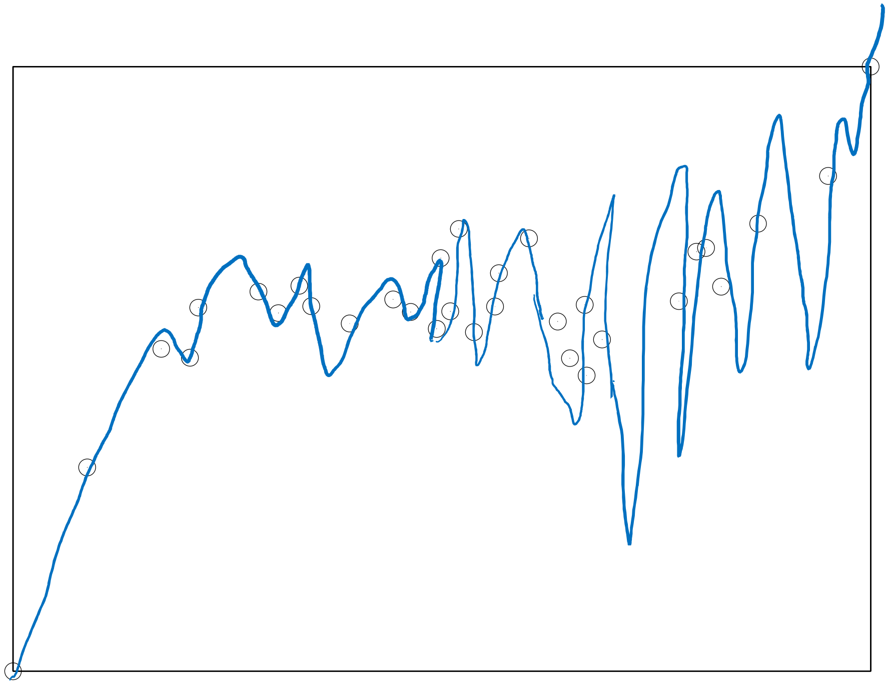
$$\hat{y} = \hat{w}_0 + \sum_{j=1}^{D} \hat{w}_j h_j(x)$$

Interpret $\hat{w}_j$ as the change in y per unit change in $h_j(x)$ if all other features are held constant.

This is generally not possible for polynomial regression or if other features use same data input!

- Can’t “fix” other features if they are derived from same input.

Overfitting



$\hat{w} = [\hat{w}_0, \hat{w}_1, \ldots, \hat{w}_D]$

Often, overfitting is associated with very large estimated parameters $\hat{w}$!

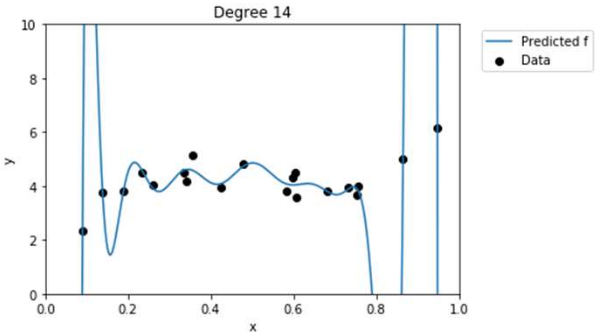
$|\hat{w}_j| >> 0$

Number of Features

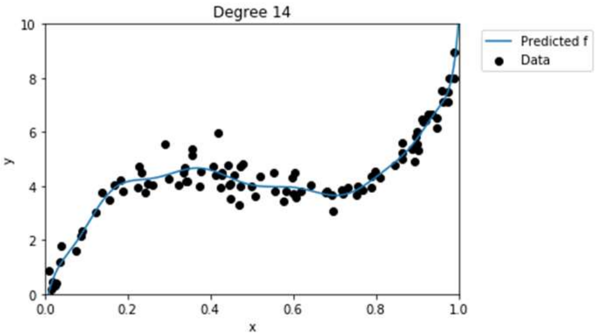
Overfitting is not limited to polynomial regression of large degree. It can also happen if you use a large number of features!

Why? Overfitting depends on how much data you have and if there is enough to get a representative sample for the complexity of the model.

Large w_j's



Reasonable w_j's



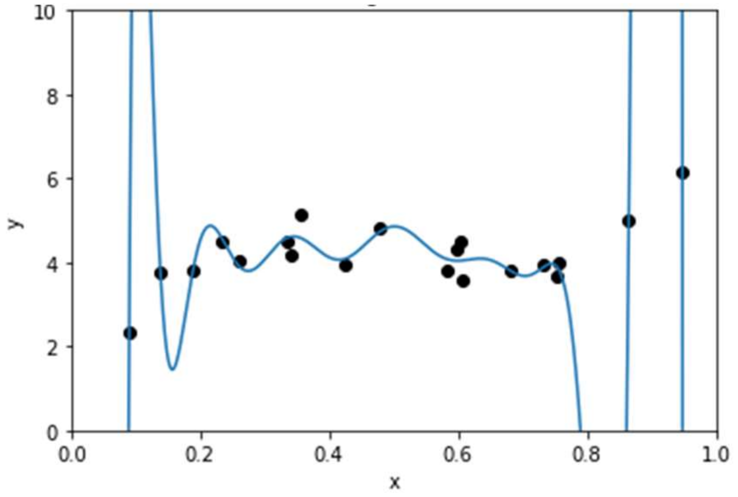
Number of Features

How do the number of features affect overfitting?

1 feature

Data must include representative example of all (x, y) pairs to avoid overfitting

HARD



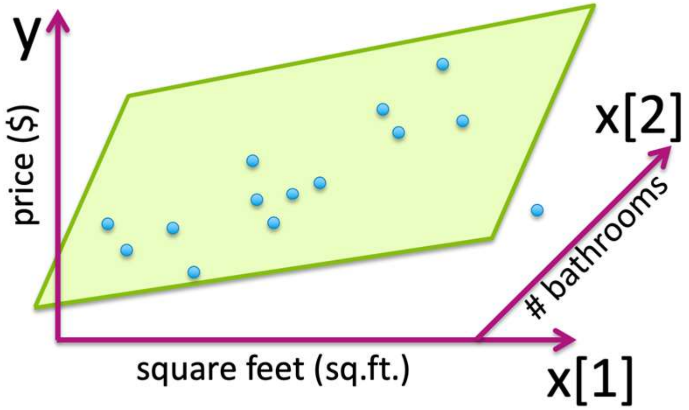
Number of Features

How do the number of features affect overfitting?

D features

Data must include representative example of all $((x[1], x[2], \ldots, x[D]), y)$ combos to avoid overfitting!

MUCH HARDER!!



Introduction to the **Curse of Dimensionality**.
We will come back to this later in the quarter!

Prevent Overfitting

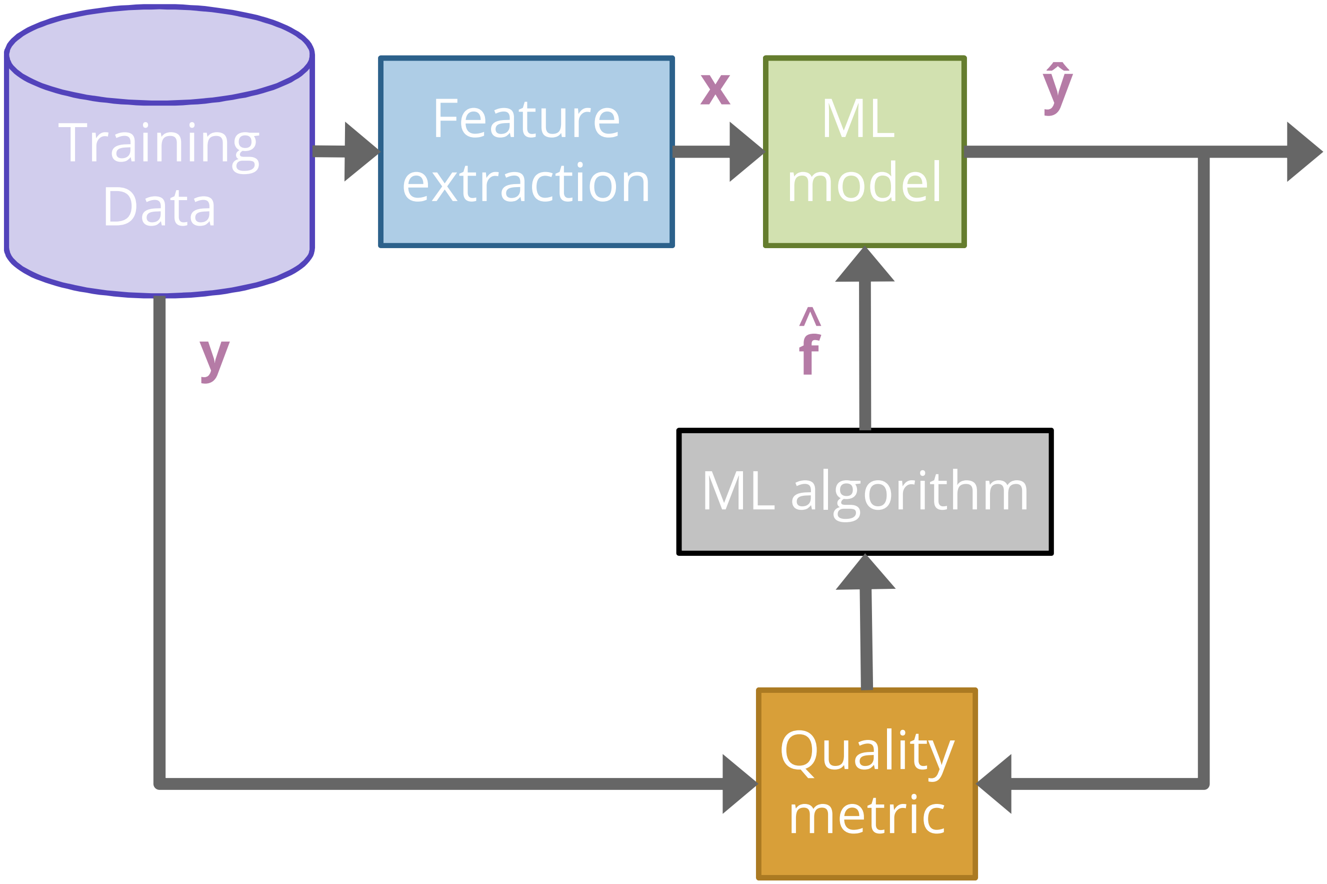
Last time, we saw we could use cross validation / validation set to pick which model complexity to use

- In the case of polynomial regression, we just chose degree p
- For deciding which or how many features to use, there are a lot of choices!
    - For d inputs, there are 2^d subsets of those features!

What if we use a model that wasn't prone to overfitting?

- Big Idea: Have the model self-regulate to prevent overfitting by making sure its coefficients don't get ”too large”

This idea is called **regularization**.


Training Data
Feature extraction
x
ML model
ŷ
y
f̂
ML algorithm
Quality metric

Regularization

Before, we used the quality metric that minimized loss

$$\hat{w} = \min_{w} L(w)$$

we've used RSS(w)

Change quality metric to balance loss with measure of overfitting

- $L(w)$ is the measure of fit
- $R(w)$ measures the magnitude of coefficients

$$\hat{w} = \min_{w} L(w) + \lambda R(w)$$

tuning parameter

How do we actually measure the magnitude of coefficients?

Magnitude

$w = [w_0, w_1, \dots, w_D]$

$R(w) =$ measure of overfitting

Come up with some number that summarizes the magnitude of the coefficients in w.

Sum? Doesn't work! What if $w_1 = 10{,}002$ and $w_1 = -10{,}001$? Then $R(w) = 1$ which indicates not overfit

$$R(w) = \sum_{j=0}^{D} w_j$$

Sum of absolute values?

$$R(w) = \sum_{j=0}^{D} |w_j| \triangleq \|w\|_1$$

This is called the L1 Norm

(We'll discuss it Wed.)

Sum of squares?

$$R(w) = \sum_{j=0}^{D} w_j^2 \triangleq \|w\|_2^2$$

This is called the L2 norm.

Ridge Regression

Change quality metric to minimize

$$\hat{w} = \min_{w} RSS(W) + \lambda \|w\|_2^2$$

λ is a tuning parameter that changes how much the model cares about the regularization term.

What if $\lambda = 0$?

$\hat{w} = \min_{w} RSS(w)$ exactly old problem!

$\rightarrow \hat{w}_{LS}$ This is called the <u>least squares</u> solution

What if $\lambda = \infty$?

If any $w_j \neq 0$, then $RSS(w) + \lambda \|w\|_2^2 = \infty$

If $w = \vec{0}$ (all $w_j = 0$), then $RSS(w) + \lambda \|w\|_2^2 = RSS(w) < \infty$

Therefore, $\hat{w} = \vec{0}$ if $\lambda = \infty$

λ in between? $0 \leq \|\hat{w}\|_2^2 \leq \|\hat{w}_{LS}\|_2^2$

Poll Everywhere

Think

1.5 Minutes

pollev.com/cse416

How does λ affect the bias and variance of the model? For each underlined section, select "Low" or "High" appropriately.

When $\lambda = 0$

The model has **(Low / High)** Bias and **(Low / High)** Variance.

When $\lambda = \infty$

The model has **(Low / High)** Bias and **(Low / High)** Variance.

Poll Everywhere

Pair

3 Minutes

pollev.com/cse416

How does λ affect the bias and variance of the model? For each underlined section, select "Low" or "High" appropriately.

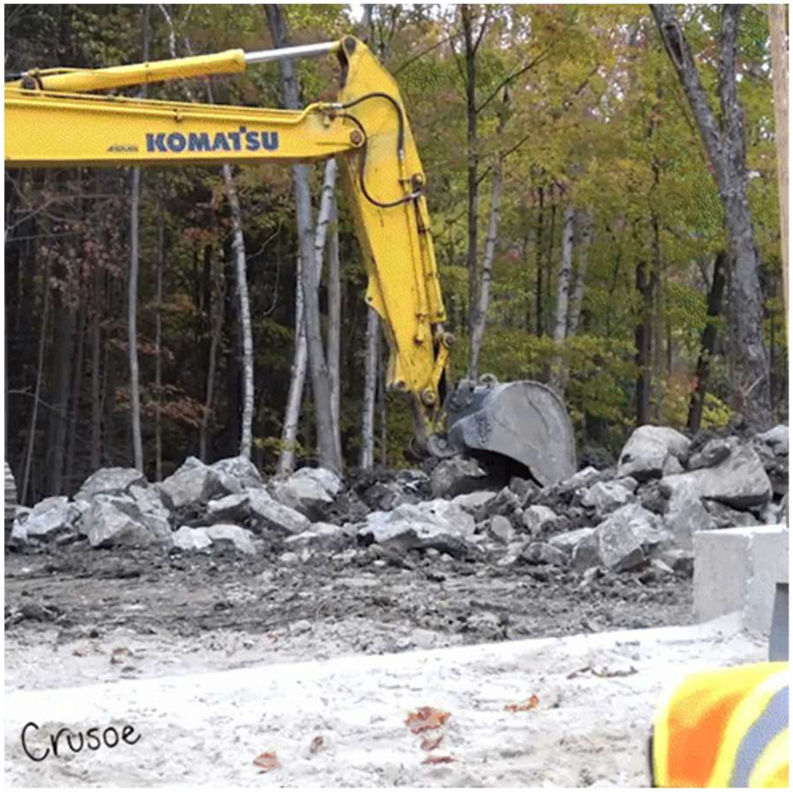
When $\lambda = 0$ => Complex model

The model has **([✓ Low] / High)** Bias and **(Low / [✓ High])** Variance.

When $\lambda = \infty$ => Simple model

The model has **(Low / [✓ High])** Bias and **([✓ Low] / High)** Variance.

Brain Break



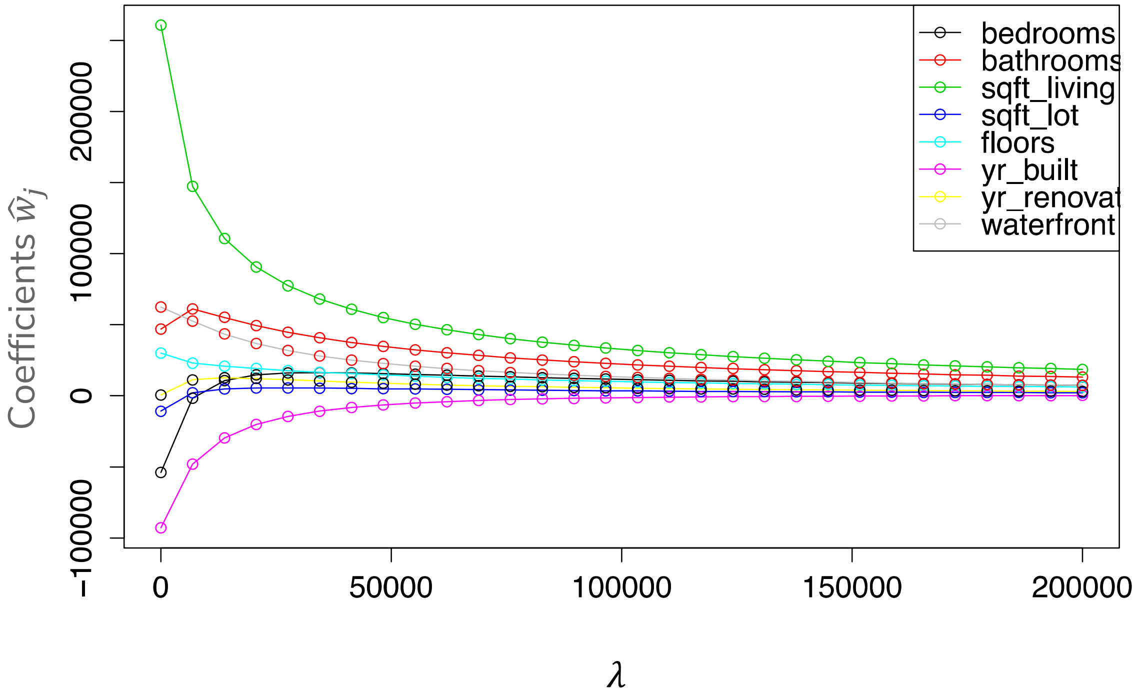
Demo: Ridge Regression

See Jupyter Notebook for interactive visualization.

Shows relationship between

- Regression line
- Residual Sum of Squares Quality Metric
  - Also called Ordinary Least Squares
- Ridge Regression Quality Metric
- Coefficient Paths

Coefficient Paths



Poll Everywhere

Think

1 min

pollev.com/cse416

How should we choose the best value of λ?

- Pick the λ that has the smallest $RSS(\hat{w})$ on the **training set**
- Pick the λ that has the smallest $RSS(\hat{w})$ on the **test set**
- Pick the λ that has the smallest $RSS(\hat{w})$ on the **validation set**
- Pick the λ that has the smallest $RSS(\hat{w}) + \lambda||\hat{\boldsymbol{w}}||_2^2$ on the **training set**
- Pick the λ that has the smallest $RSS(\hat{w}) + \lambda||\hat{w}||_2^2$ on the **test set**
- Pick the λ that has the smallest $RSS(\hat{w}) + \lambda||\hat{w}||_2^2$ on the **validation set**
- Pick the λ that results in the smallest coefficients
- Pick the λ that results in the largest coefficients
- None of the above

Poll Everywhere

Pair

2 min

pollev.com/cse416

How should we choose the best value of λ?

- Pick the λ that has the smallest $RSS(\hat{w})$ on the **training set**
- Pick the λ that has the smallest $RSS(\hat{w})$ on the **test set**
- [✓ Pick the λ that has the smallest $RSS(\hat{w})$ on the **validation set**]
- Pick the λ that has the smallest $RSS(\hat{w}) + \lambda||\hat{\mathbf{w}}||_2^2$ on the **training set**
- Pick the λ that has the smallest $RSS(\hat{w}) + \lambda||\hat{w}||_2^2$ on the **test set**
- Pick the λ that has the smallest $RSS(\hat{w}) + \lambda||\hat{w}||_2^2$ on the **validation set**
- Pick the λ that results in the smallest coefficients
- Pick the λ that results in the largest coefficients
- None of the above

Regularization

At this point, I've hopefully convinced you that regularizing coefficient magnitudes is a good thing to avoid overfitting!

You:

We might have gotten a bit carried away, it doesn't ALWAYS make sense...

The Intercept

$w_{rest} = [w_1, \ldots, w_D]$

For most of the features, looking for large coefficients makes sense to spot overfitting. The one it does not make sense for is the **intercept**.

We shouldn't penalize the model for having a higher intercept since that just means the y value units might be really high!

- My demo before does this wrong and penalizes w_0 as well!

Two ways of dealing with this

- Change the measure of overfitting to not include the intercept

$$\min_{w_0, w_{rest}} RSS(w_0, w_{rest}) + \lambda \|w_{rest}\|_2^2$$

- Center the y values so they have mean 0
  - This means forcing w_0 to be small isn't a problem

Scaling Features

The other problem we looked over is the “scale” of the coefficients.

Remember, the coefficient for a feature increase per unit change in that feature (holding all others fixed in multiple regression)

Consider our housing example with $(sq.ft., price)$ of houses

- Say we learned a coefficient $\hat{w}_1$ for that feature
- What happens if we change the unit of x to square **miles?** Would $\hat{w}_1$ need to change?
  - It would need to get bigger since the prices are the same but its inputs are smaller

This means we accidentally penalize features for having large coefficients due to having small value inputs!

Scaling Features

Fix this by **normalizing** the features so all are on the same scale!

$$\tilde{h}_j(x_i) = \frac{h_j(x_i) - \mu_j(x_1, \dots, x_N)}{\sigma_j(x_1, \dots, x_N)}$$

Where

The mean of feature j:

$$\mu_j(x_1, \dots, x_N) = \frac{1}{N}\sum_{i=1}^{N} h_j(x_i)$$

The standard devation of feature j:

$$\sigma_j(x_1, \dots, x_N) = \sqrt{\frac{1}{N}\sum_{i=1}^{N}\left(h_j(x_i) - \mu_j(x_1, \dots, x_N)\right)^2}$$

Important: Must scale the test data and all future data using the means and standard deviations **of the training set!**

- Otherwise the units of the model and the units of the data are not comparable!

Recap

Theme: Use regularization to prevent overfitting

Ideas:

- How to interpret coefficients
- How overfitting is affected by number of data points
- Overfitting affecting coefficients
- Use regularization to prevent overfitting
- How L2 penalty affects learned coefficients
- Visualizing what regression is doing
- Practicalities: Dealing with intercepts and feature scaling