As you come in
Which tool did you use to work on Assignment 1?

- Google Colab (online)
- Anaconda + Jupyter Notebooks (local)
- Other?

If you said other, what are you using?
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.
Recap: Ridge Regression

\[ \sum_{j=1}^{D} w_j^2 = \sum_{j=1}^{D} w_j^2 \]

Change quality metric to minimize

\[ \hat{w} = \min_w RSS(w) + \lambda \|w\|_2^2 \]

\( \lambda \) is tuning parameter that changes how much the model cares about the regularization term.

What if \( \lambda = 0 \)?

\[ \hat{w} = \min_w RSS(w) \]

This is called the least squares 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 \( RSS(w) + \lambda \|w\|_2^2 = RSS(w) \leq \infty \)

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

\( \lambda \) in between?

\[ 0 \leq \|\hat{w}\|_2^2 \leq \|\hat{w}_L\|_2^2 \]
How should we choose the best value of $\lambda$?

- Pick the $\lambda$ that has the smallest $RSS(\hat{w})$ on the **training set**
- Pick the $\lambda$ that has the smallest $RSS(\hat{w})$ on the **test set**
- Pick the $\lambda$ that has the smallest $RSS(\hat{w}) + \lambda ||\hat{w}||_{2}^{2}$ on the **training set**
- Pick the $\lambda$ that has the smallest $RSS(\hat{w}) + \lambda ||\hat{w}||_{2}^{2}$ on the **test set**
- Pick the $\lambda$ that has the smallest $RSS(\hat{w}) + \lambda ||\hat{w}||_{2}^{2}$ on the **validation set**
- Pick the $\lambda$ that results in the smallest coefficients
- Pick the $\lambda$ that results in the largest coefficients
- None of the above
For any particular setting of $\lambda$, use Ridge Regression objective

$$\hat{w}_{ridge} = \min_w RSS(w) + \lambda \|w_{1:D}\|^2_2$$

If $\lambda$ is too small, will overfit to training set. Too large, $\hat{w}_{ridge} = 0$.

How do we choose the right value of $\lambda$? We want the one that will do best on future data. This means we want to minimize error on the validation set.

Don’t need to minimize $RSS(w) + \lambda \|w_{1:D}\|^2_2$ on validation because you can’t overfit to the validation data (you never train on it).

Another argument is that it doesn’t make sense to compare those values for different settings of $\lambda$. They are in different “units” in some sense.
Choosing $\lambda$

The process for selecting $\lambda$ is exactly the same as we saw with using a validation set or using cross validation.

for $\lambda$ in $\lambda$s:

Train a model using Gradient Descent

$$\hat{w}_{\text{ridge}(\lambda)} = \min_w RSS_{\text{train}}(w) + \lambda \|w_{1:D}\|_2^2$$

Compute validation error

$validation\_error = RSS_{\text{val}}(\hat{w}_{\text{ridge}(\lambda)})$

Track $\lambda$ with smallest $validation\_error$

Return $\lambda^*$ & estimated future error $RSS_{\text{test}}(\hat{w}_{\text{ridge}(\lambda^*)})$

There is no fear of overfitting to validation set since you never trained on it! You can just worry about error when you aren’t worried about overfitting to the data.
Feature Selection
Why do we care about selecting features? Why not use them all?

**Complexity**
Models with too many features are more complex. Might overfit!

**Interpretability**
Can help us identify which features carry more information.

**Efficiency**
Imagine if we had MANY features (e.g. DNA). \( \hat{w} \) could have \( 10^{11} \) coefficients. Evaluating \( \hat{y} = \hat{w}^T h(x) \) would be very slow!

If \( \hat{w} \) is **sparse**, only need to look at the non-zero coefficients

\[
\hat{y} = \sum_{\hat{w}_j \neq 0} \hat{w}_j h_j(x)
\]
Sparsity: Housing

Might have many features to potentially use. Which are useful?

- Lot size
- Single Family
- Year built
- Last sold price
- Last sale price/sqft
- Finished sqft
- Unfinished sqft
- Finished basement sqft
- # floors
- Flooring types
- Parking type
- Parking amount
- Cooling
- Heating
- Exterior materials
- Roof type
- Structure style

- Dishwasher
- Garbage disposal
- Microwave
- Range / Oven
- Refrigerator
- Washer
- Dryer
- Laundry location
- Heating type
- Jetted Tub
- Deck
- Fenced Yard
- Lawn
- Garden
- Sprinkler System

...
How happy are you? What part of the brain controls happiness?

$x$ - Happy

$y$ - Sad

10 – Happy 😊

0 – Sad 😞
Option 1
All Subsets
Best Model
Size 0

\[ \text{Features} \]
- # bathrooms
- # bedrooms
- sq.ft. living
- sq.ft lot
- floors
- year built
- year renovated
- waterfront

\[ y_i = \varepsilon_i \]
Best Model
Size 1

Features
# bathrooms
# bedrooms
sq.ft. living
sq.ft lot
floors
year built
year renovated
waterfront

$R_{SS \text{train}}(\bar{y})$

# of features
Best Model Size 1

Features
# bathrooms
# bedrooms
sq.ft. living
sq.ft lot
floors
year built
year renovated
waterfront

RSS_{train}(\hat{\omega})

# of features
Best Model
Size 1

Features
- # bathrooms
- # bedrooms
- sq. ft. living
- sq. ft. lot
- floors
- year built
- year renovated
- waterfront

\[ \text{Features} \]

\[ R_{\text{RSS}} \]

# of features

0
1
Best Model
Size 1

Features
# bathrooms
# bedrooms
sq.ft. living
sq.ft. lot
floors
year built
year renovated
waterfront
Best Model
Size 1

Features
- # bathrooms
- # bedrooms
- sq.ft. living
- sq.ft lot
- floors
- year built
- year renovated
- waterfront

$RSS_{train}(\hat{\theta})$
Best Model
Size 1

Features
# bathrooms
# bedrooms
sq.ft. living
sq.ft. lot
floors
year built
year renovated
waterfront

$RSS_{\text{train}}(\hat{\omega})$ vs. # of features
Best Model
Size 1

Features
- # bathrooms
- # bedrooms
- sq.ft. living
- sq.ft lot
- floors
- year built
- year renovated
- waterfront

$\mathbf{RSS}_{\text{train}}(\hat{\mathbf{w}})$

# of features
Best Model
Size 1

Features
# bathrooms
# bedrooms
sq.ft. living
sq.ft lot
floors
year built
year renovated
waterfront

$RSS_{\text{train}}(\hat{\theta})$

# of features

0 1
Best Model
Size 1

<table>
<thead>
<tr>
<th>Features</th>
<th>RSS_{train}(\bar{w})</th>
</tr>
</thead>
<tbody>
<tr>
<td># bathrooms</td>
<td></td>
</tr>
<tr>
<td># bedrooms</td>
<td></td>
</tr>
<tr>
<td>sq.ft. living</td>
<td></td>
</tr>
<tr>
<td>sq.ft lot</td>
<td></td>
</tr>
<tr>
<td>floors</td>
<td></td>
</tr>
<tr>
<td>year built</td>
<td></td>
</tr>
<tr>
<td>year renovated</td>
<td></td>
</tr>
<tr>
<td>waterfront</td>
<td></td>
</tr>
</tbody>
</table>

Feature sizes: 0, 1
Not necessarily nested!
Best Model – Size 1: sq.ft living
Best Model – Size 2:
# bathrooms & # bedrooms

Features
# bathrooms
# bedrooms
sq.ft living
sq.ft lot
floors
year built
year renovated
waterfront
Best Model
Size 3

Features
# bathrooms
# bedrooms
sq.ft. living
sq.ft lot
floors
year built
year renovated
waterfront

$RSS_{train}(\hat{\theta})$

# of features
Best Model
Size 4

Features
# bathrooms
# bedrooms
sq.ft. living
sq.ft lot
floors
year built
year renovated
waterfront

$$RSS_{train}(\hat{\theta})$$

# of features
Best Model
Size 5

Features
# bathrooms
# bedrooms
sq.ft. living
sq.ft lot
floors
year built
year renovated
waterfront
**Features**
- # bathrooms
- # bedrooms
- sq.ft. living
- sq.ft lot
- floors
- year built
- year renovated
- waterfront

**Best Model**
**Size 6**

![Graph showing RSStrain(\(\hat{w}\)) vs. # of features](image)
Best Model
Size 7

Features
# bathrooms
# bedrooms
sq.ft. living
sq.ft lot
floors
year built
year renovated
waterfront

$RSS_{\text{train}}(\hat{\beta})$

# of features

0 1 2 3 4 5 6 7
Best Model
Size 8

Features
# bathrooms
# bedrooms
sq.ft. living
sq.ft lot
floors
year built
year renovated
waterfront

$R_{\text{SS, train}}(\bar{w})$ vs # of features
The past example was talking about training data. **(True or False)**

We would expect to see a graph with (relatively) the same shape if we were measuring test RSS instead. Why or why not?

- True
- False

Not on Poll Everywhere: Why or why not? Use your reasoning!
The past example was talking about training data. (True or False) We would expect to see a graph with (relatively) the same shape if we were measuring test RSS instead. Why or why not?

- True
- False

Not on PollEverywhere: Why or why not? Use your reasoning!
Choose Num Features?

Option 1
Assess on a validation set

Option 2
Cross validation

Option 3+
Other metrics for penalizing model complexity like BIC
How many models did we evaluate?
\[ \hat{y}_i = \epsilon_i \]
\[ \hat{y}_i = w_0 h_0(x) + \epsilon_i \]
\[ \hat{y}_i = w_1 h_1(x) + \epsilon_i \]

... \[ \hat{y}_i = w_0 h_0(x) + w_1 h_1(x) + \epsilon_i \]

... \[ \hat{y}_i = w_0 h_0(x) + w_1 h_1(x) + \ldots + w_D h_D(x) \epsilon_i \]

If evaluating all subsets of 8 features only took 5 seconds, then
- 16 features would take 21 minutes
- 32 features would take almost 3 years
- 100 features would take almost \(7.5 \times 10^{20}\) years
  - \(50,000,000,000\times\) longer than the age of the universe!
Knowing it’s impossible to find exact solution, approximate it!

**Forward stepwise**
Start from model with no features, iteratively add features as performance improves.

**Backward stepwise**
Start with a full model and iteratively remove features that are the least useful.

**Combining forward and backwards steps**
Do a forward greedy algorithm that eventually prunes features that are no longer as relevant

*And many many more!*
Example
Greedy Algorithm

Start by selecting number of features $k$

$$S_0 \leftarrow \{\}$$
for $i \leftarrow 1..k$:
  Find feature $f_i$ not in $S_{i-1}$, that when combined with $S_{i-1}$, minimizes the loss the most.

$$S_i \leftarrow S_{i-1} \cup \{f_i\}$$
Return $S_k$

Called greedy because it makes choices that look best at the time.
Brain Break
Option 2
Regularization
Recap: Regularization

Before, we used the quality metric that minimized loss

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

Change quality metric to balance loss with measure of overfitting

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

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

How do we actually measure the magnitude of coefficients?
Recap: Magnitude

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

**Sum?**

- Doesn't work! What is $w_1 = 10,002$ and $w_i = -10,001$? Then $R(w) = 1$ which indicates not overfit.

**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?**

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

This is called the L2 norm.
We saw that Ridge Regression shrinks coefficients, but they don't become 0. What if we remove weights that are sufficiently small?
Ridge for Feature Selection

Instead of searching over a **discrete** set of solutions, use regularization to reduce coefficient of unhelpful features.

Start with a full model, and then “shrink” ridge coefficients near 0. Non-zero coefficients would be considered selected as important.
Look at two related features #bathrooms and # showers. Our model ended up not choosing any features about bathrooms!
Ridge for Feature Selection

What if we had originally removed the # showers feature?

- The coefficient for # bathrooms would be larger since it wasn’t “split up” amongst two correlated features
- Instead, it would be nice if there were a regularizer that favors sparse solutions in the first place to account for this...
LASSO Regression

Change quality metric to minimize

$$\hat{\omega} = \min_w \text{RSS}(w) + \lambda ||w||_1$$

$\lambda$ is a tuning parameter that changes how much the model cares about the regularization term.

What if $\lambda = 0$?

$$\hat{\omega} = \min_w \text{RSS}(w) \Rightarrow \hat{\omega}_{LS}$$

What if $\lambda = \infty$?

$$\hat{\omega} = \hat{0}$$

$\lambda$ in between?

$$0 \leq ||\hat{\omega}||_1 \leq ||\hat{\omega}_{LS}||_1$$

Notice smaller than $\hat{\omega}_{LS}$ when measured with L1 Norm.
Ridge Coefficient Paths

![Graph showing Ridge Coefficient Paths](image)

- **Coefficients** $\hat{w}_j$
- **$\lambda$**

Legend:
- bedrooms
- bathrooms
- sqft living
- sqft lot
- floors
- yr built
- yr renovated
- waterfront
LASSO Coefficient Paths

Only feature when \( \lambda = 100/600 \)
When using the L1 Norm ($\|w\|_1$) as a regularizer, it favors solutions that are **sparse**. Sparsity for regression means many of the learned coefficients are 0.

This has to do with the shape of the norm

When $w_j$ is small, $w_j^2$ is VERY small!
Another way to visualize why LASSO prefers sparse solutions.

The L1 ball has corners (places where some coefficients are 0).
Sparsity Geometry
Brain Break
Choosing $\lambda$

Exactly the same as Ridge Regression :)

This will be true for almost every hyper-parameter we talk about

A **hyper-parameter** is a parameter you specify for the model that influences which parameters (e.g. coefficients) are learned by the ML algorithm.
De-biasing LASSO

LASSO adds bias to the Least Squares solution (this was intended to avoid the variance that leads to overfitting)

- Recall Bias-Variance Tradeoff

It’s possible to try to remove the bias from the LASSO solution using the following steps

1. Run LASSO to select which features should be used (those with non-zero coefficients)
2. Run regular Ordinary Least Squares on the dataset with only those features

Coefficients are no longer shrunk from their true values
Issues with LASSO

1. Within a group of highly correlated features (e.g. # bathroom and # showers), LASSO tends to select amongst them arbitrarily.
   - Maybe it would be better to select them all together?

2. Often, empirically Ridge tends to have better predictive performance

Elastic Net aims to address these issues

$$\hat{w}_{ElasticNet} = \min_w RSS(w) + \lambda_1||w||_1 + \lambda_2||w||_2$$

Combines both to achieve best of both worlds!
Be careful when interpreting results of feature selection or feature importances in Machine Learning!

- Selection only considers features included
- Sensitive to correlations between features
- Results depend on the algorithm used!
Recap

**Theme**: Use regularization to do feature selection

**Ideas:**
- Describe “all subsets” approach to feature selection and why it’s impractical to implement.
- Formulate LASSO objective
- Describe how LASSO coefficients change as hyper-parameter $\lambda$ is varied
- Interpret LASSO coefficient path plot
- Compare and contrast LASSO and ridge