CSE/STAT 416

Regularization – LASSO Regression

Pre-Class Videos

Hunter Schafer
University of Washington
April 5, 2023

Pre-Class Video 1

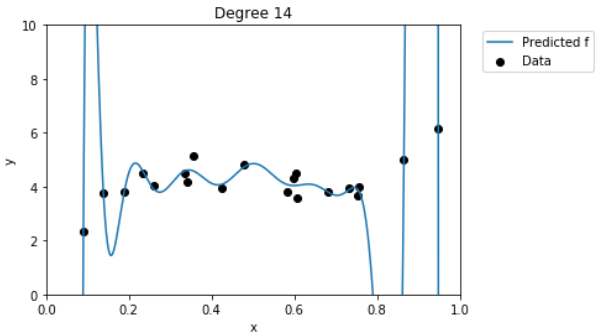
Ridge Regression

Recap: Number of Features

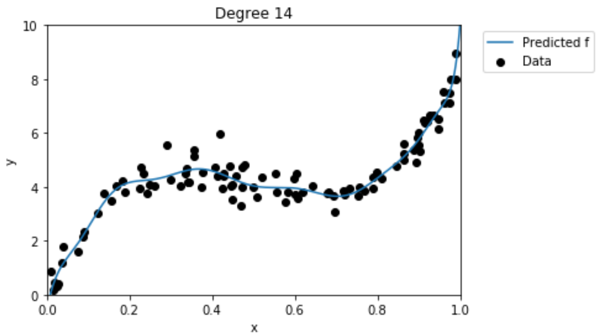
Overfitting is not limited to polynomial regression of large degree. It can also happen if you use a large number of features!

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.

$|w_j| >> 0$



$|w_j| \approx$ reasonable



Recap: Ridge Regression

L2 norm $||w||_2^2 = \sum_{j=1}^{D} w_j^2$

Change quality metric to minimize

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

λ is 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 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 $RSS(w) + \lambda ||w||_2^2 = RSS(w) < \infty$

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

λ in between?

$0 \leq ||\hat{w}||_2^2 \leq ||\hat{w}_{LS}||_2^2$

slido

Think

2 min

sli.do #cs416

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

Choosing λ

For any particular setting of λ, use Ridge Regression objective

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

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

How do we choose the right value of λ? 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 λ. They are in different "units" in some sense.

Hyperparameter tuning

Choosing λ

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

```
for λ in λs:
  Train a model using using Gradient Descent
```

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

```
  Compute validation error
```

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

```
  Track λ with smallest validation_error
Return λ* & estimated future error RSS_test(ŵ_ridge(λ*))
```

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.

Pre-Class Video 2

Feature Selection & All Subsets

Benefits

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

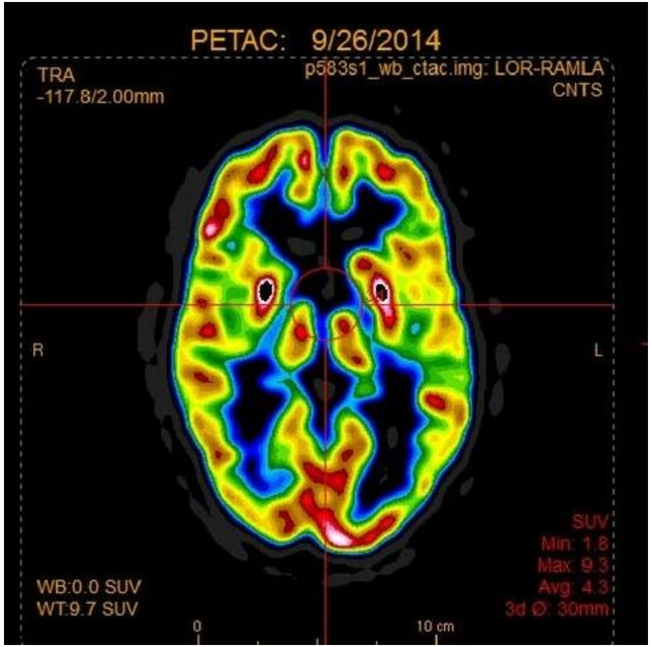
Dishwasher
Garbage disposal
Microwave
Range / Oven
Refrigerator
Washer
Dryer
Laundry location
Heating type
Jetted Tub
Deck
Fenced Yard
Lawn
Garden
Sprinkler System

…

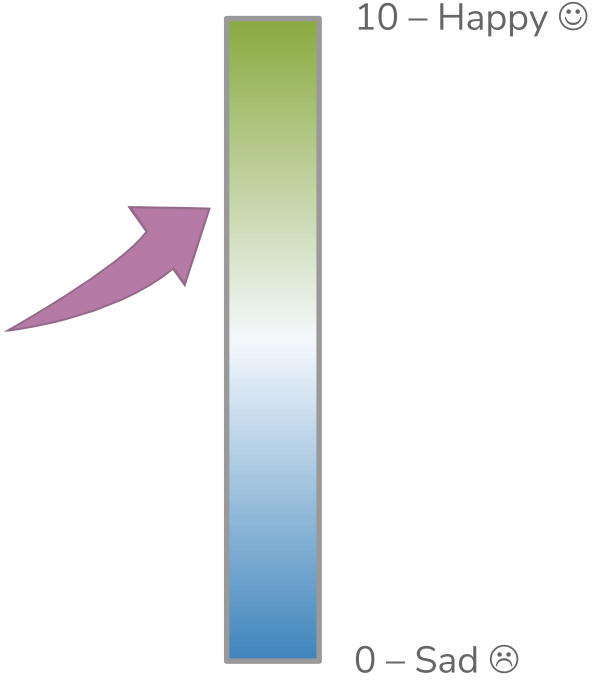
Sparsity: Reading Minds

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

x

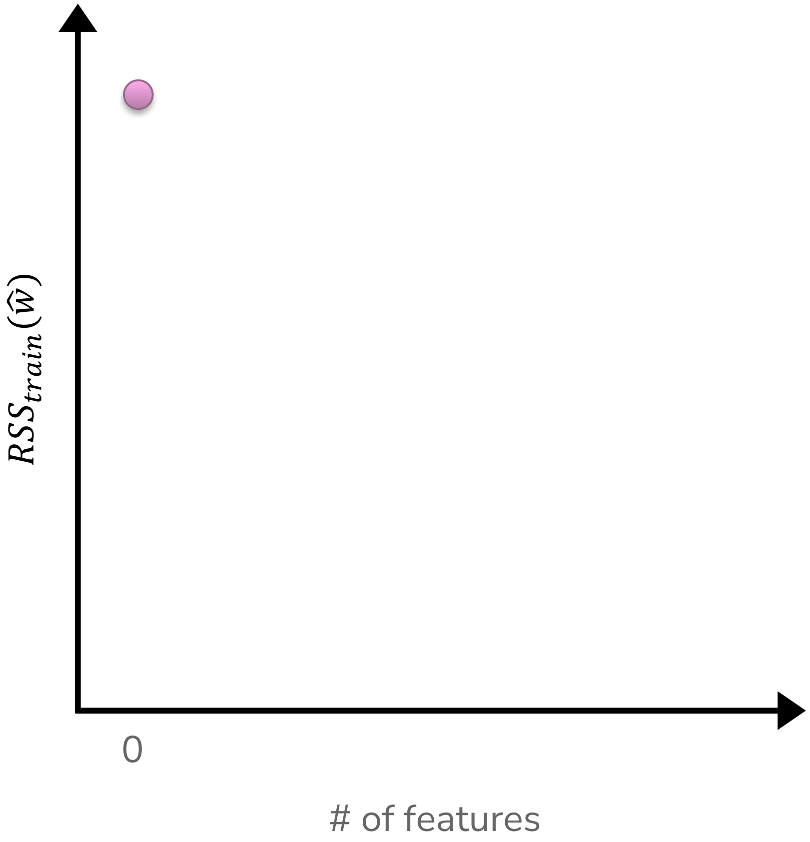


y



Best Model Size 0

All Subsets

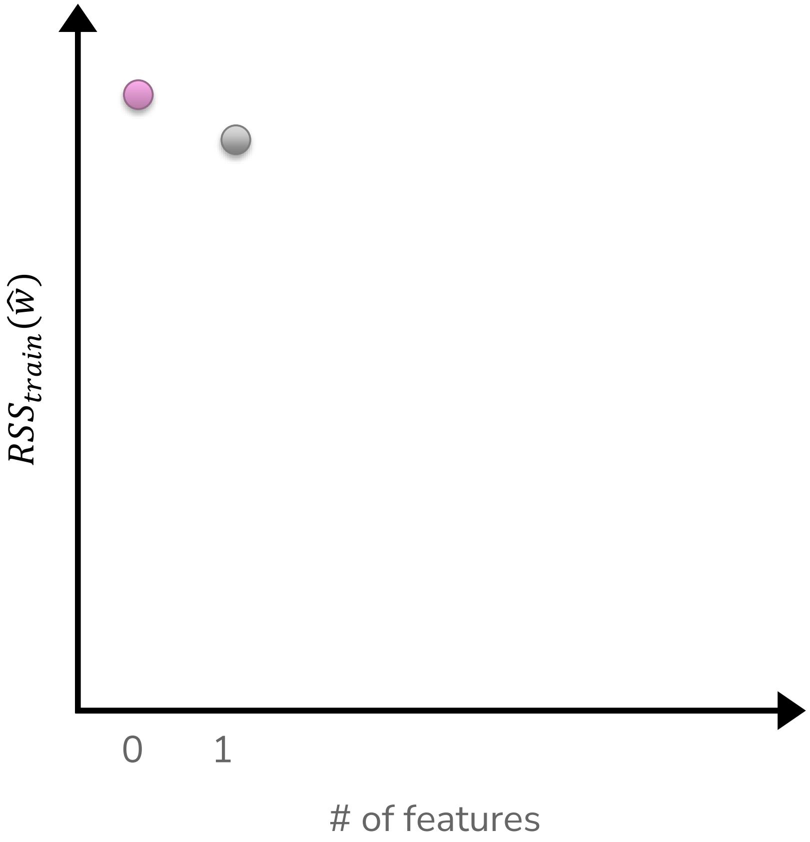


Noise only:

$y_i = \varepsilon_i$

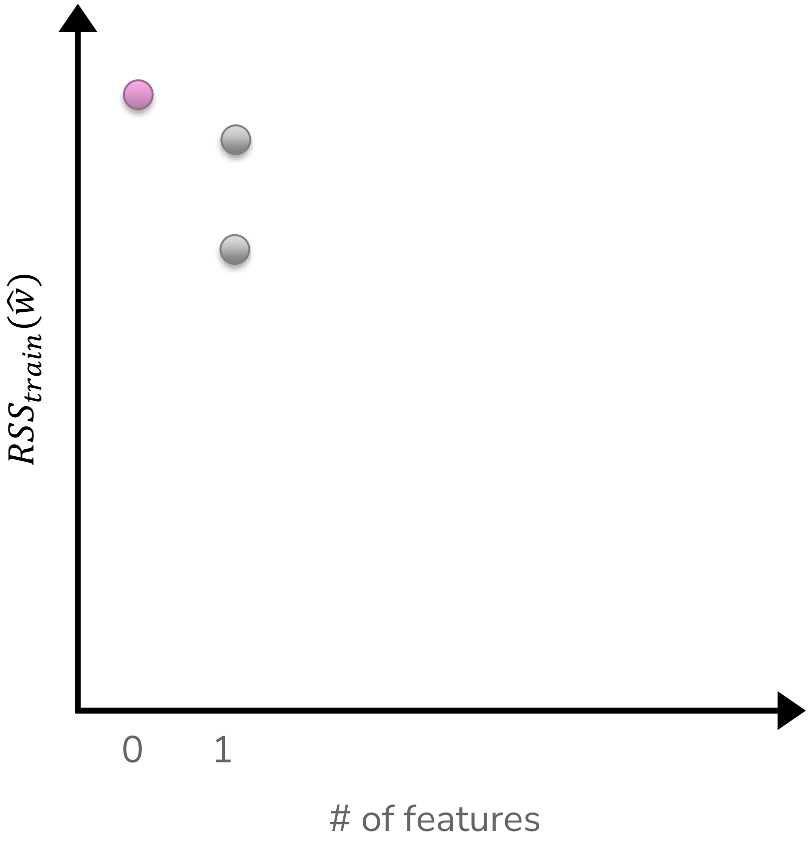
Features
bathrooms
bedrooms
sq.ft. living
sq.ft lot
floors
year built
year renovated
waterfront

Best Model Size 1



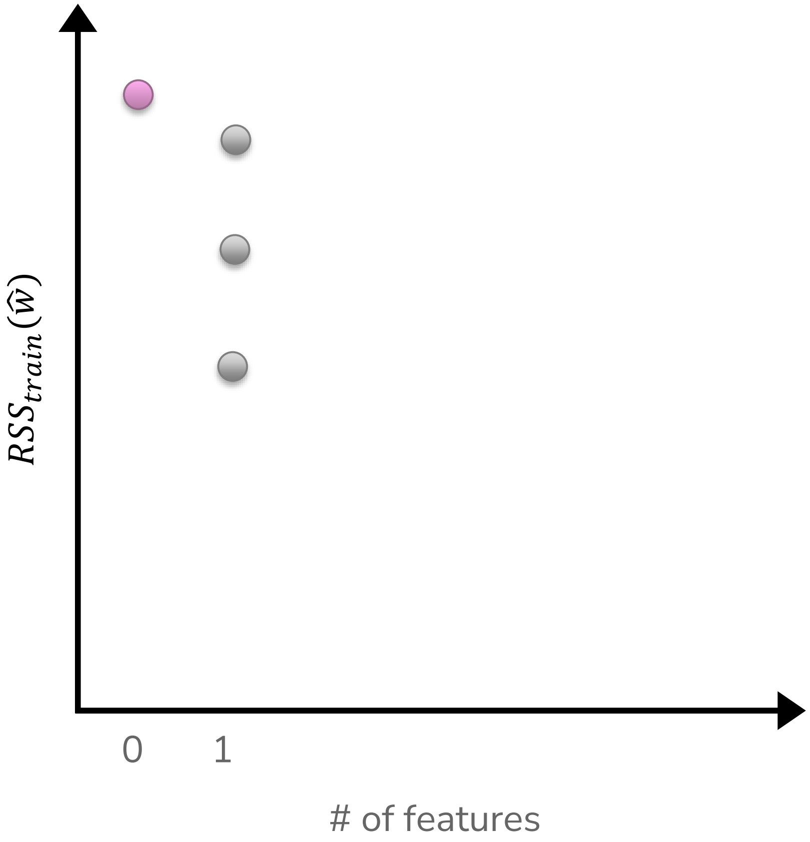
Features
bathrooms
bedrooms
sq.ft. living
sq.ft lot
floors
year built
year renovated
waterfront

Best Model Size 1



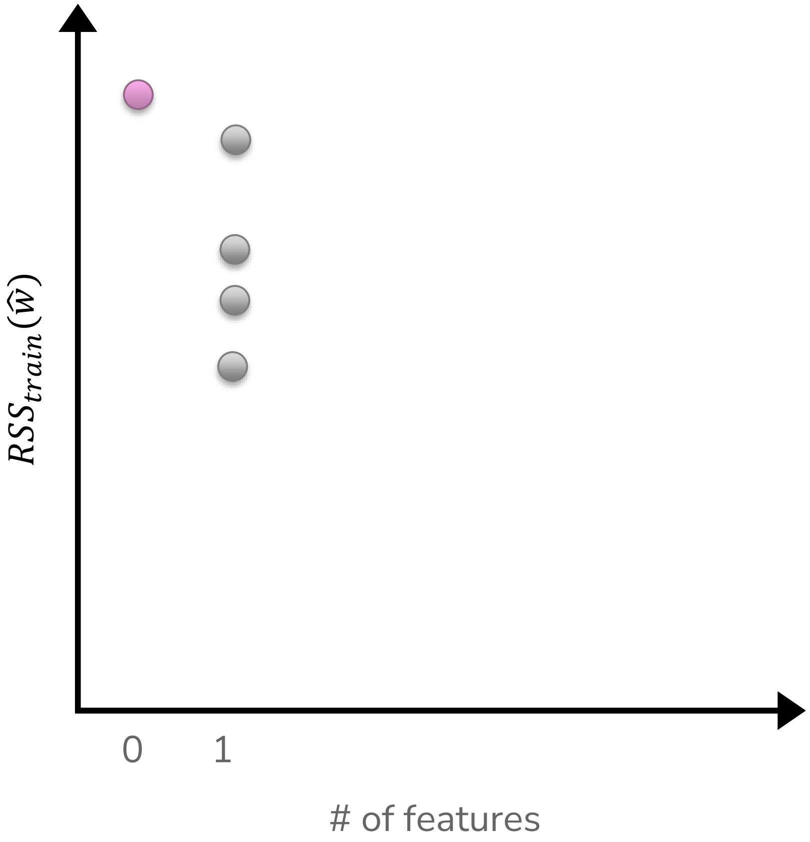
Features
bathrooms
bedrooms
sq.ft. living
sq.ft lot
floors
year built
year renovated
waterfront

Best Model Size 1



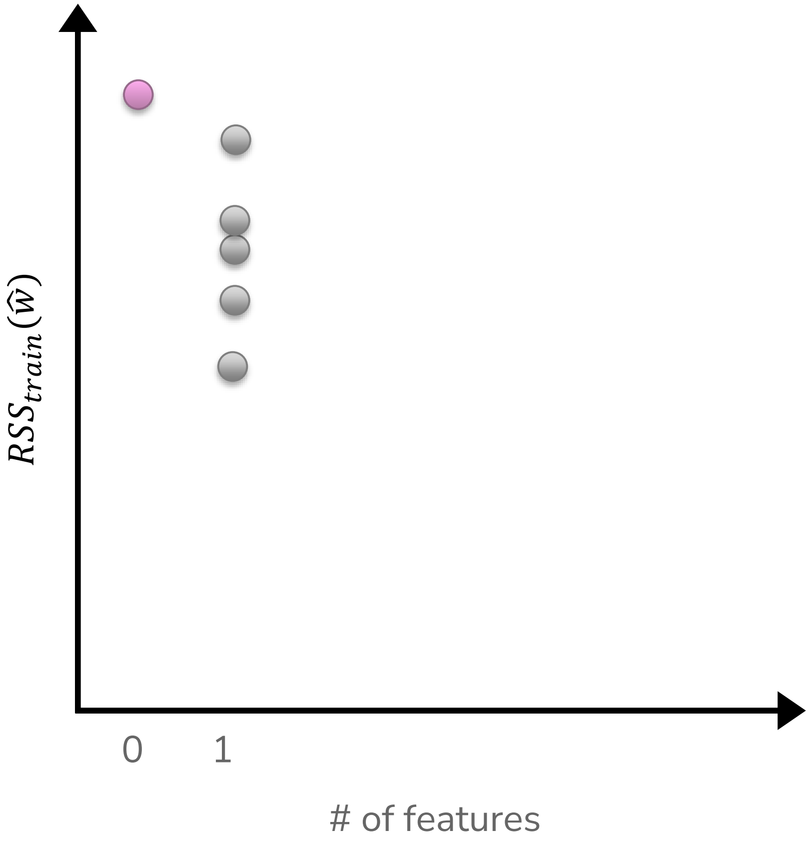
Features
bathrooms
bedrooms
sq.ft. living
sq.ft lot
floors
year built
year renovated
waterfront

Best Model
Size 1



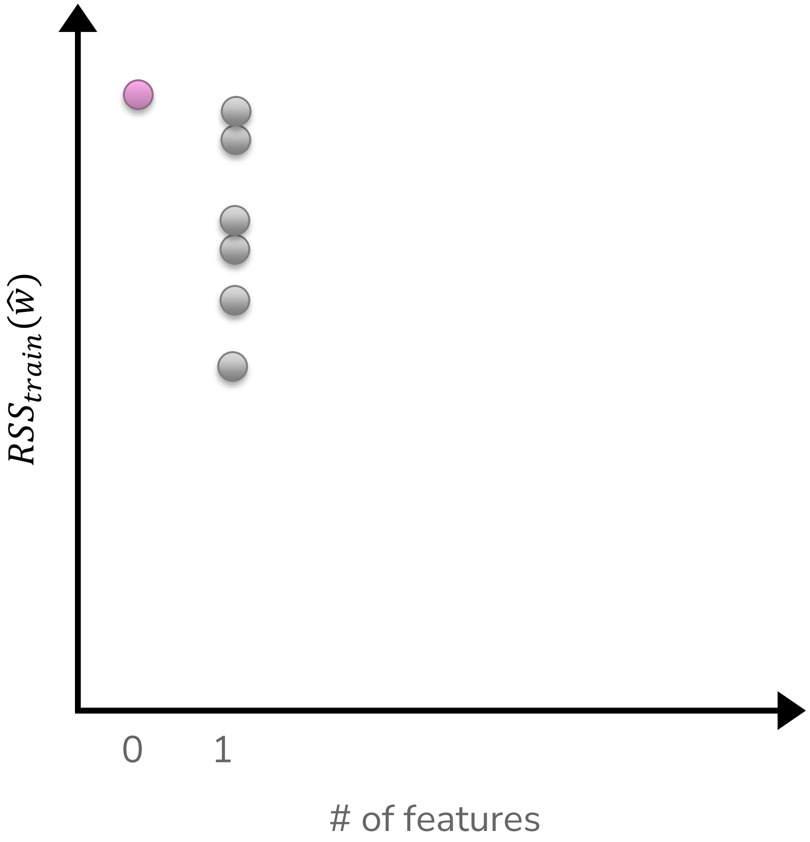
Features
bathrooms
bedrooms
sq.ft. living
sq.ft lot
floors
year built
year renovated
waterfront

Best Model Size 1



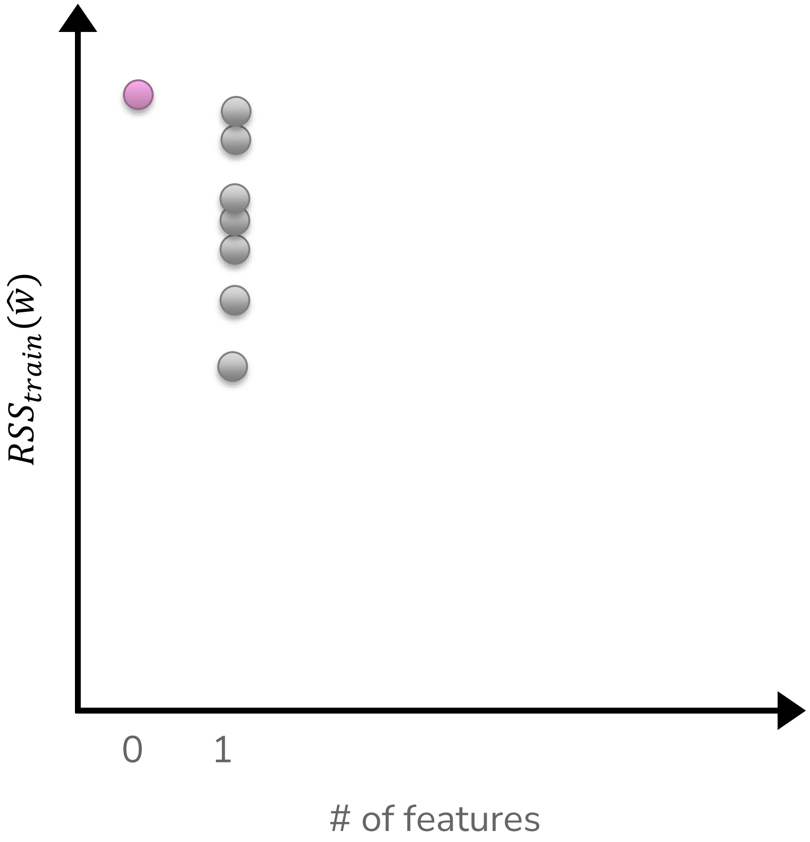
Features
bathrooms
bedrooms
sq.ft. living
sq.ft lot
floors
year built
year renovated
waterfront

Best Model Size 1



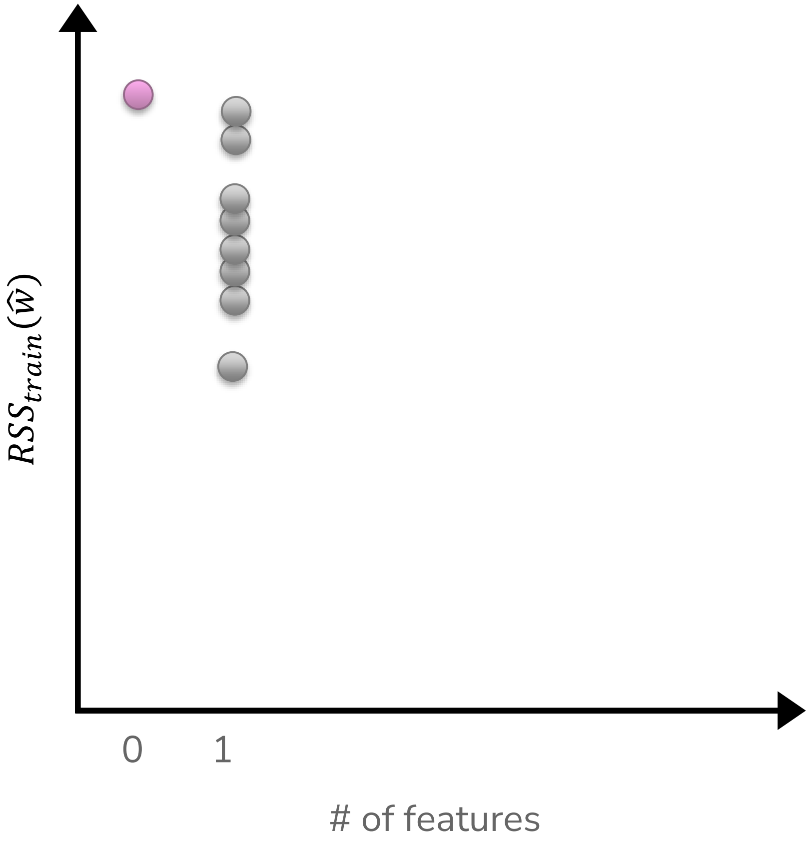
Features
bathrooms
bedrooms
sq.ft. living
sq.ft lot
floors
year built
year renovated
waterfront

Best Model Size 1



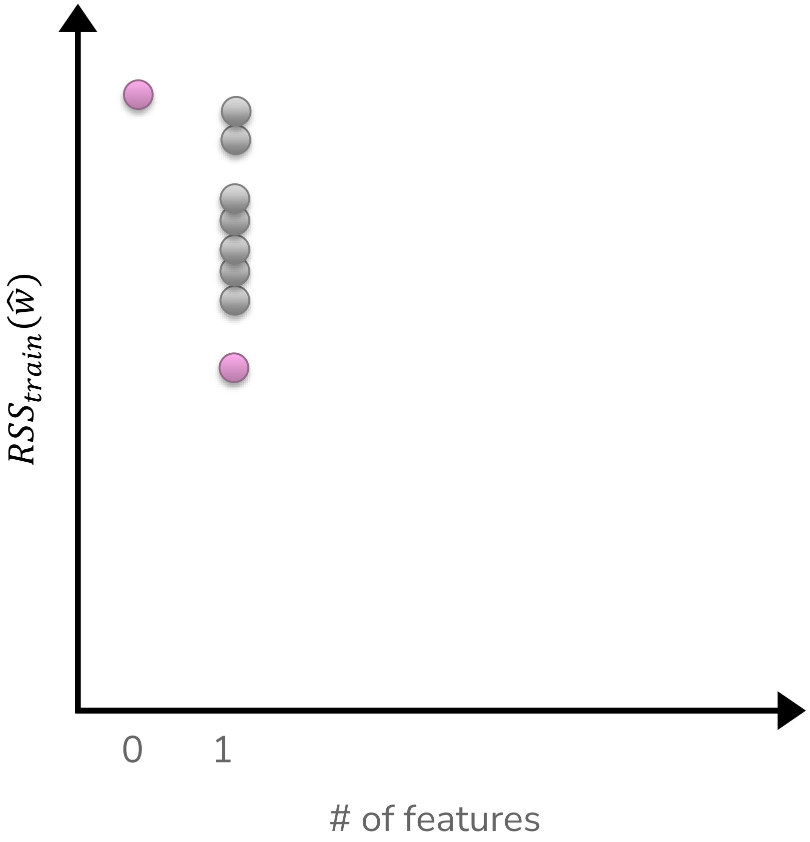
Features
bathrooms
bedrooms
sq.ft. living
sq.ft lot
floors
year built
year renovated
waterfront

Best Model Size 1



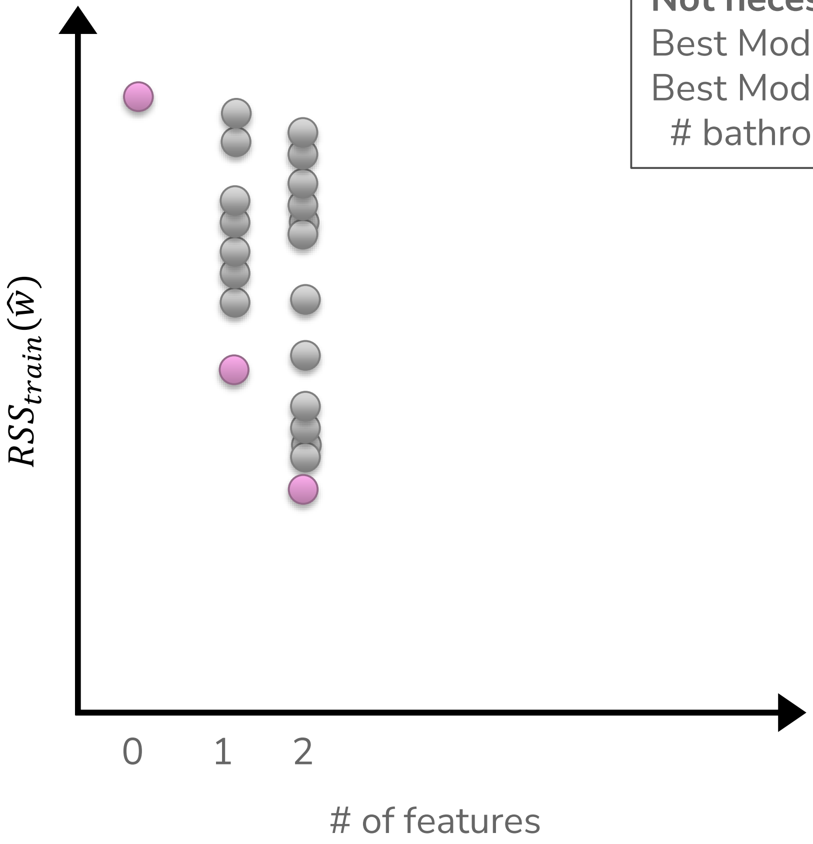
Features
bathrooms
bedrooms
sq.ft. living
sq.ft lot
floors
year built
year renovated
waterfront

Best Model Size 1



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

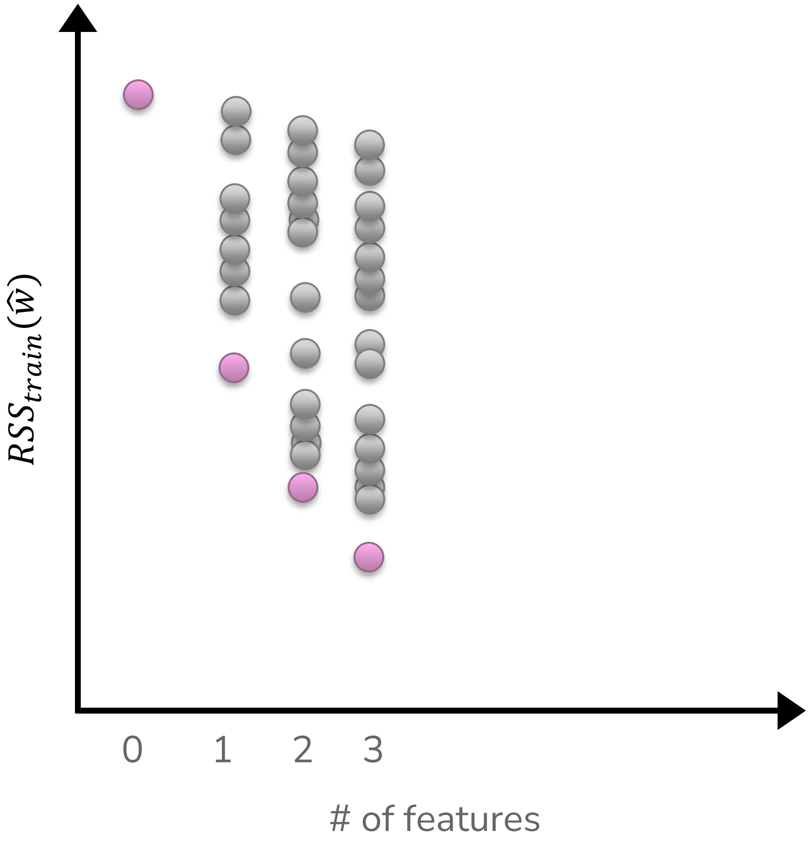
Best Model Size 2



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

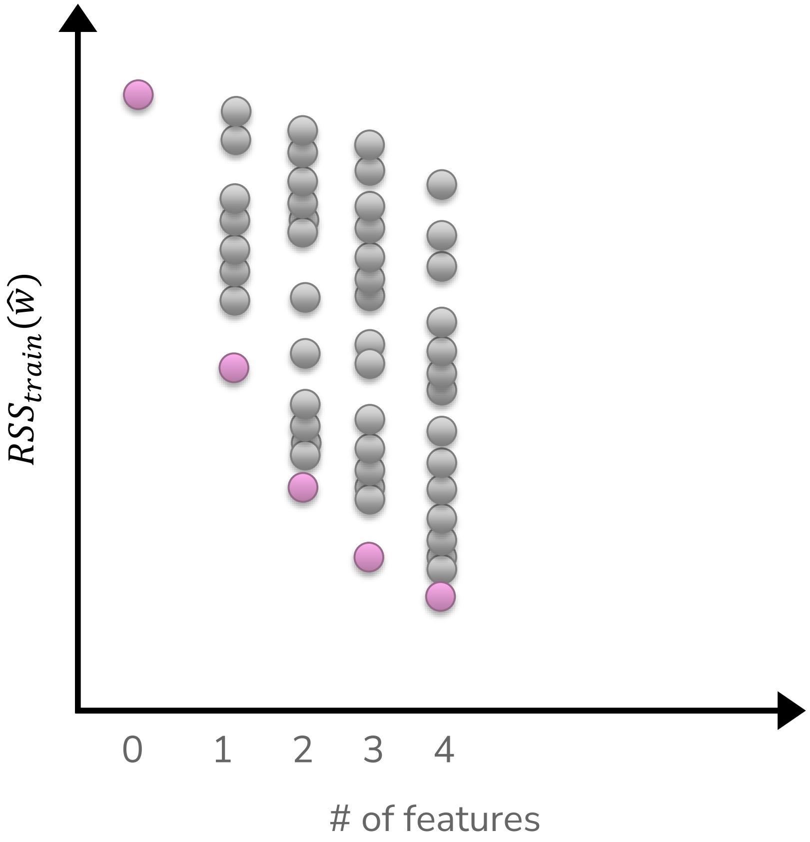
Features
bathrooms
bedrooms
sq.ft. living
sq.ft lot
floors
year built
year renovated
waterfront

Best Model Size 3



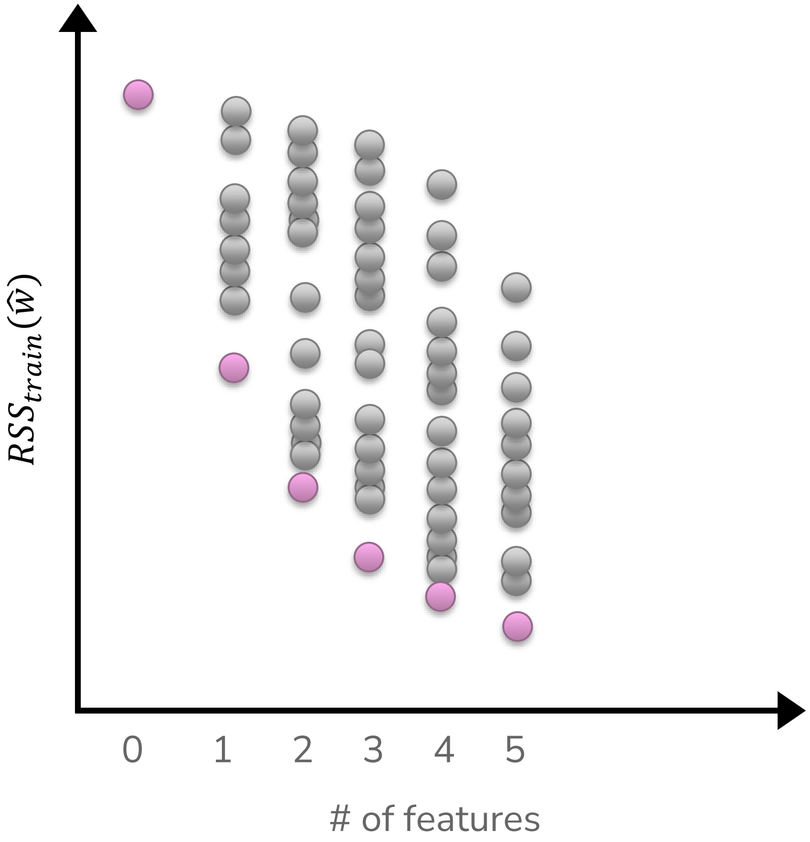
Features
bathrooms
bedrooms
sq.ft. living
sq.ft lot
floors
year built
year renovated
waterfront

Best Model Size 4



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

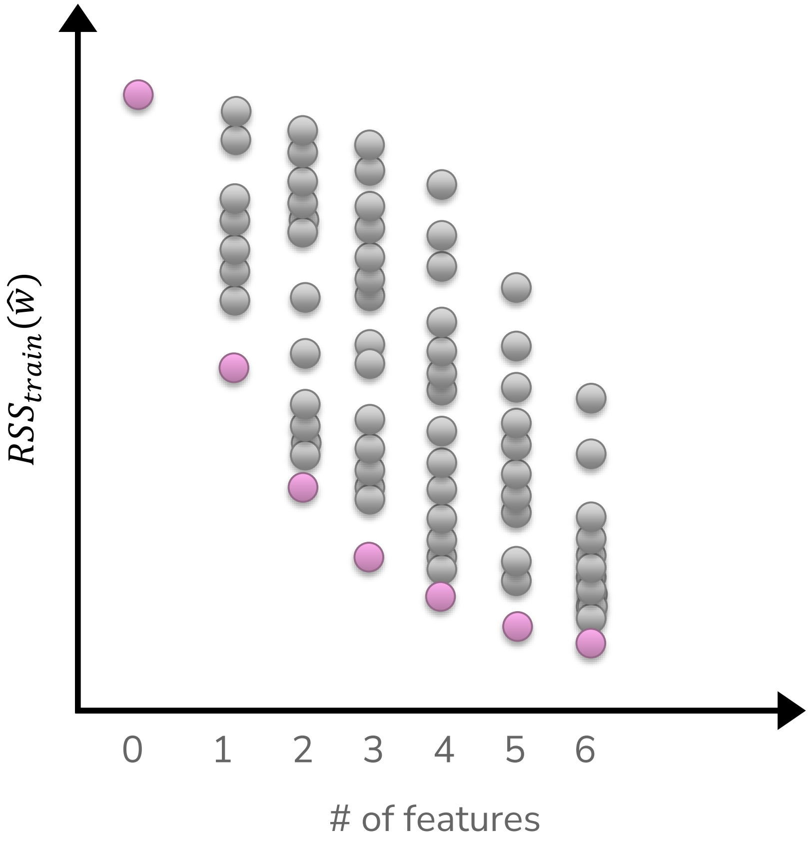
Best Model Size 5



Features

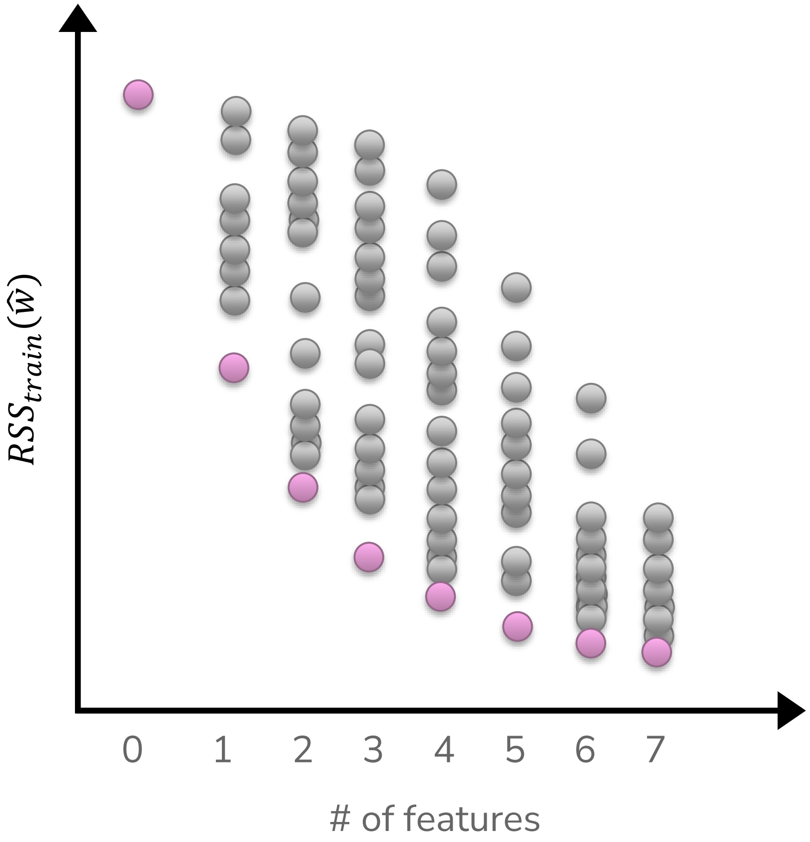
bathrooms
bedrooms
sq.ft. living
sq.ft lot
floors
year built
year renovated
waterfront

Best Model Size 6



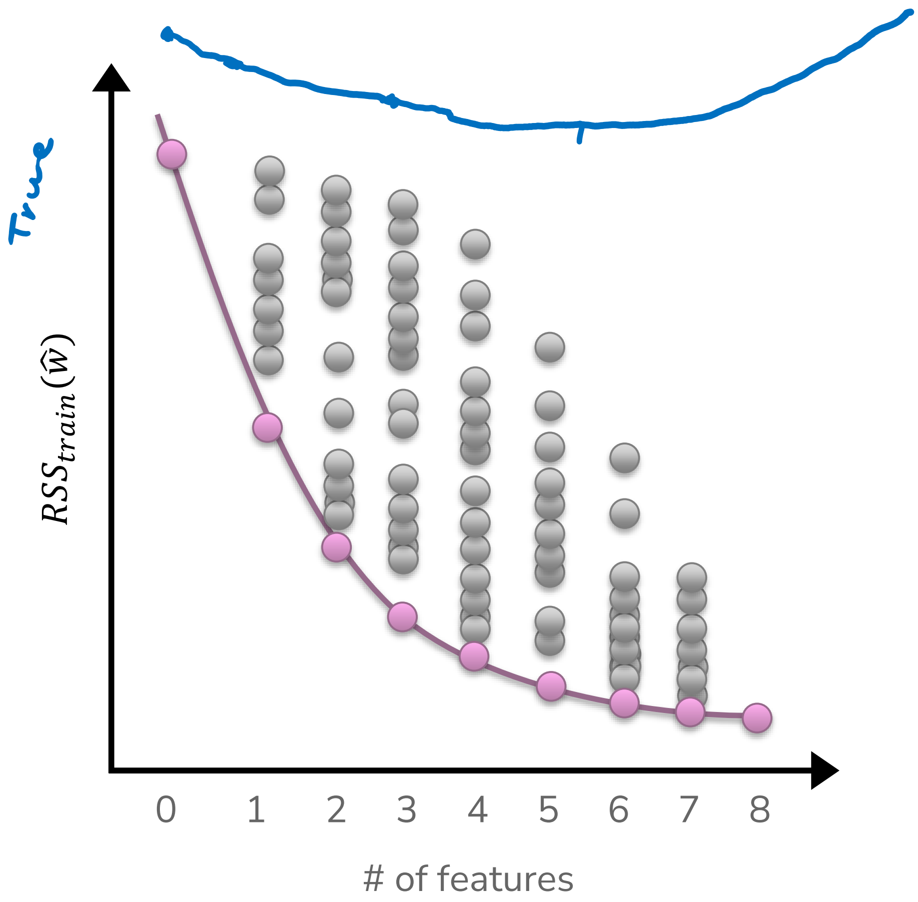
Features
bathrooms
bedrooms
sq.ft. living
sq.ft lot
floors
year built
year renovated
waterfront

Best Model Size 7



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

Best Model Size 8



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

Choose Num Features?

Option 1

Assess on a validation set

Option 2

Cross validation

Option 3+

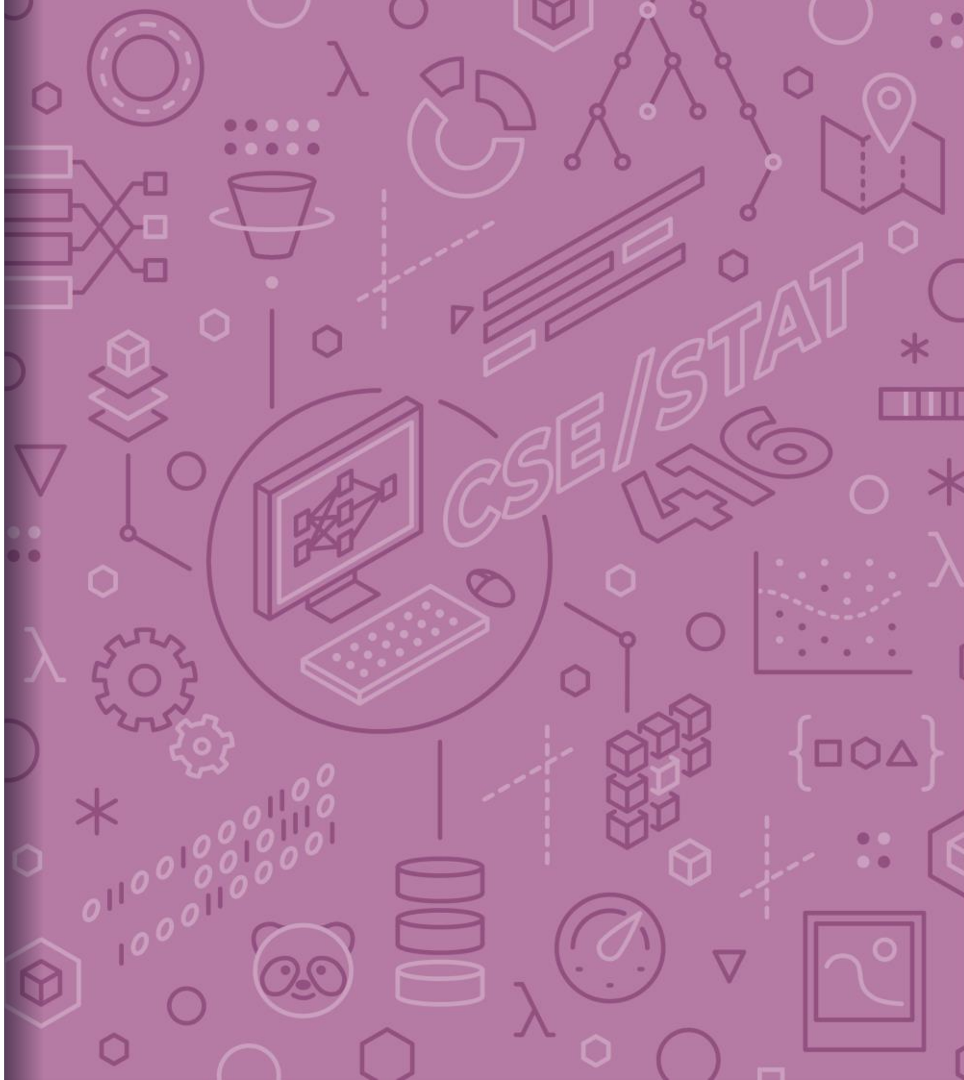
Other metrics for penalizing model complexity like Bayesian Information Criterion (BIC)

CSE/STAT 416

Regularization – LASSO Regression

Hunter Schafer
University of Washington
April 5, 2023

Questions? Raise hand or **sli.do #cs416**
Before Class: Favorite food near campus?
Listening to: [Sammy Rae & The Friends]



Administrivia

- Last lecture in the "Regression" case study!
  - Next 2 weeks: Classification
  - Following 1 week: Deep Learning
- Section Tomorrow:
  - Coding up RIDGE and Lasso (helpful for HW1!)
- Upcoming Due Dates:
  - HW0 Late due date Thurs 4/6 11:59PM (if using 2 late days)
  - HW1 out right after class, due Tues 4/11 11:59PM
  - Learning Reflection 1 due Fri 11:59PM
- OH is a great place to ask your learning reflection questions!
- Reminder of resources

Recap: Ridge Regression

L2 norm $||w||_2^2 = \sum_{j=1}^{D} w_j^2$

Change quality metric to minimize

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

λ is 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 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 $RSS(w) + \lambda ||w||_2^2 = RSS(w)$ $<\infty$

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

λ in between?

$0 \leq ||\hat{w}||_2^2 \leq ||\hat{w}_{LS}||_2^2$

Benefits

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.

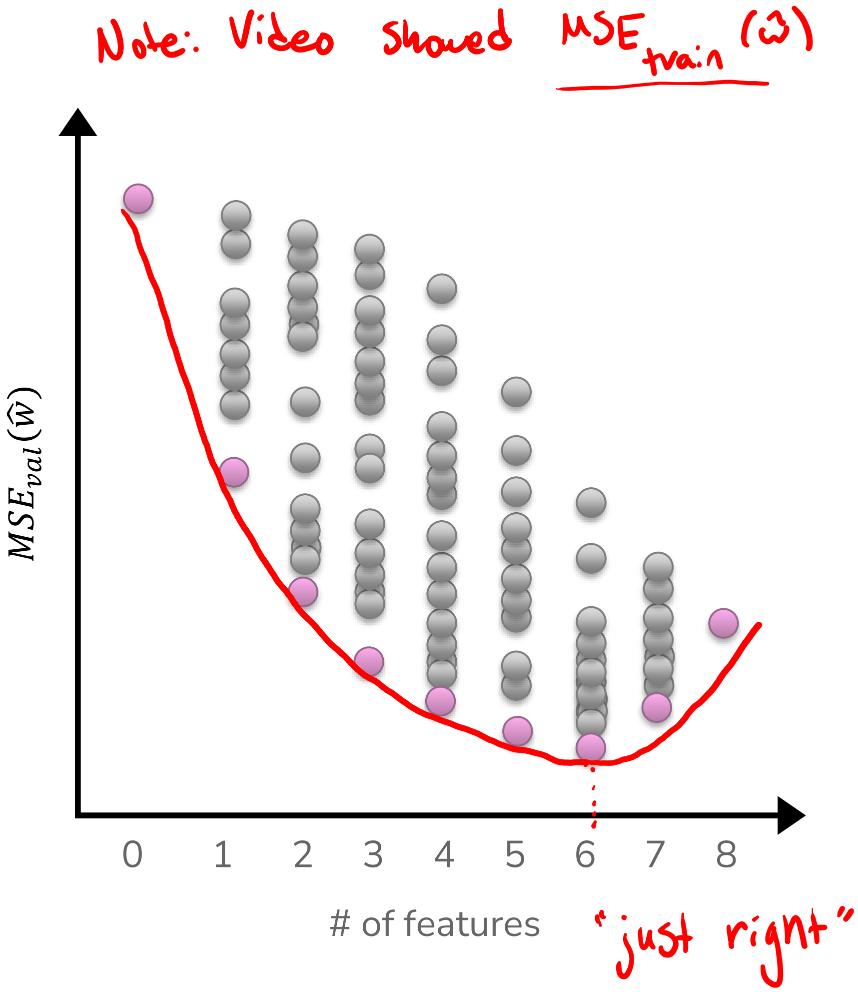
Efficiency

Imagine if we had MANY features (e.g. DNA). $\widehat{w}$ 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

$$\hat{y} = \sum_{\hat{w}_j \neq 0} \hat{w}_j h_j(x)$$

Best Model
Size 8



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

Efficiency of All Subsets

whether or not $h_1(x)$ is in model

... $h_D(x)$

How many models did we evaluate?

$\hat{y}_i = w_0$ → [0 0 0 … 0 0 0]

$\hat{y}_i = w_0 + w_1 h_1(x)$ → [1 0 0 … 0 0 0]

$\hat{y}_i = w_0 + w_2 h_2(x)$ → [0 1 0 … 0 0 0]

… …

$\hat{y}_i = w_0 + w_1 h_1(x) + w_2 h_2(x)$ [1 1 0 … 0 0 0]

… …

$\hat{y}_i = w_0 + w_1 h_1(x) + \ldots + w_D h_D(x)$ [1 1 1 … 1 1 1]

2^D possible combos

$2 \cdot 2 \cdot 2 \cdot \ldots \cdot 2 \cdot 2 \cdot 2 = 2^D$

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 7.5×10^{20} years
  - 50,000,000,000x longer than the age of the universe!

Choose Num Features?

Clearly all subsets is unreasonable. How can we choose how many and which features to include?

Option 1

Greedy Algorithm

Option 2

LASSO Regression (L1 Regularization)

L2 => Ridge

L1 => LASSO

Greedy Algorithms

Greedy Algorithms

Knowing it's impossible to find exact solution, approximate it!

Greedy algorithms take locally optimal steps

Forward stepwise

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

Locally optimal $\neq$ Globally optimal

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: Forward Stepwise

Start by selecting number of desired features k

min_val = ∞

$S_0 \leftarrow \emptyset$

S_i = selected features

for $i \leftarrow 1..k$:

try every feature, one at a time

    Find feature f_i not in S_{i-1}, that when combined with S_{i-1}, minimizes the validation loss the most.

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

    if val_loss(S_i) > min_val:

stop once val. err. begins to worsen

        break # No need to look at more features

Called greedy because it makes choices that look best at the time.

- Greedily optimal !=

slido

Think

1 min

sli.do #cs416

- Say you want to find the optimal two-feature model, using the forward stepwise algorithm. What model would the forward stepwise algorithm choose?

Subsets of Size 1

Features	**Val Loss**
# bath	201
# bed	300
sq ft	157
year built	224

Subsets of Size 2

Features *(unordered)*	**Val Loss**
(# bath, # bed)	120
(# bath, sq ft)	131
(# bath, year built)	190
(# bed, sq ft)	137
(# bed, year built)	209
(sq ft, year built)	145

slido

Group

1 min

sli.do #cs416

- Say you want to find the optimal two-feature model, using the forward stepwise algorithm. What model would the forward stepwise algorithm choose?

Subsets of Size 1

Features	Val Loss
# bath	201
# bed	300
sq ft	157
year built	224

Subsets of Size 2

Features *(unordered)*	Val Loss
(# bath, # bed)	120
(# bath, sq ft)	131
(# bath, year built)	190
(# bed, sq ft)	137
(# bed, year built)	209
(sq ft, year built)	145

Globally optimal (all subsets)

forward Stepwise

Would select for S_1

Brain Break

Option 2
Regularization

Recap: Regularization

Before, we used the quality metric that minimize loss

$$\hat{w} = \underset{w}{\operatorname{argmin}}\, L(w)$$

Change quality metric to balance loss with measure of overfitting

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

$$\hat{w} = \underset{w}{\operatorname{argmin}}\, L(w) + \lambda R(w)$$

How do we actually measure the magnitude of coefficients?

Recap: Magnitude

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

$$\hat{w} = \underset{w}{\operatorname{argmin}}\, MSE(w) + \lambda R(w)$$

Sum? X

$$R(w) = w_0 + w_1 + \cdots + w_d$$

Doesn't work because the weights can cancel out (e.g. $w_0 = 1000, w_1 = -1000$) which so $R(w)$ doesn't reflect the magnitudes of the weights

Sum of absolute values? => L1 Regularization

$$R(w) = |w_0| + |w_1| + \cdots + |w_d| = \|w\|_1$$

It works! We're using L1-norm, for L1-regularization (LASSO)

Sum of squares? => L2 Regularization

$$R(w) = |w_0|^2 + |w_1|^2 + \ldots + |w_d|^2 = w_0^2 + w_1^2 + \ldots + w_d^2 = \|w\|_2^2$$

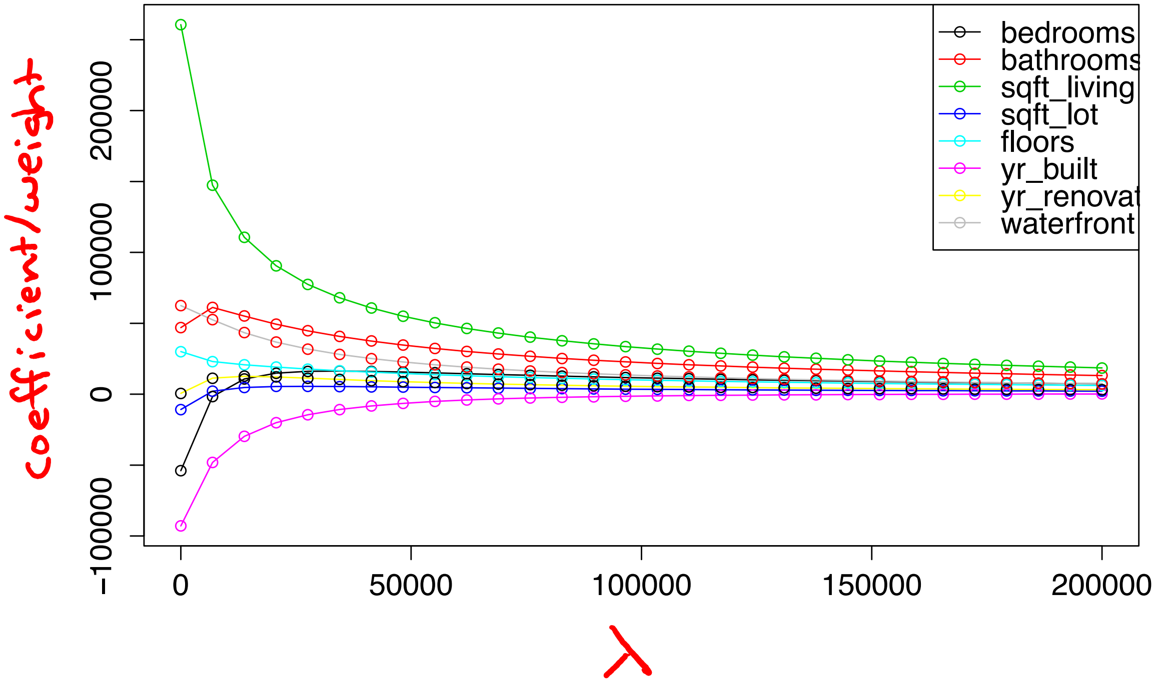
It works! We're using L2-norm, for L2-regularization (Ridge Regression)

Note: Definition of p-Norm: $\|w\|_p^p = |w_0|^p + |w_1|^p + \ldots + |w_d|^p$

Ridge for Feature Selection

MUST NORMALIZE

We saw that Ridge Regression shrinks coefficients, but they don't become 0. What if we remove weights that are sufficiently small?

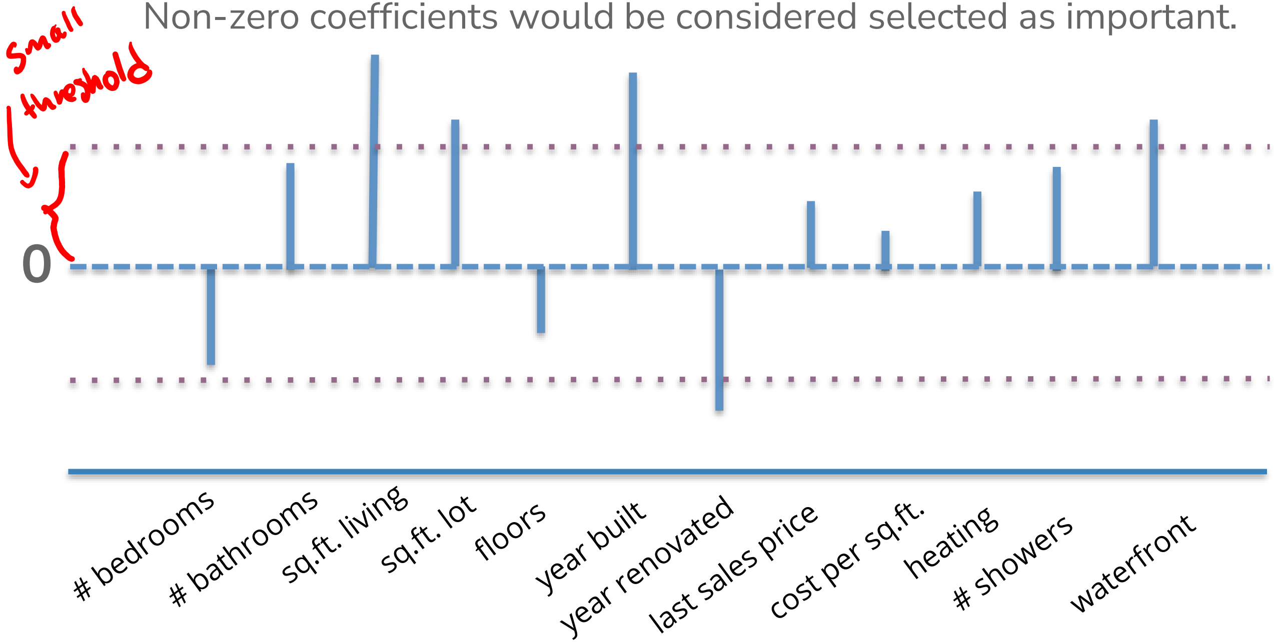


Ridge for Feature Selection

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.

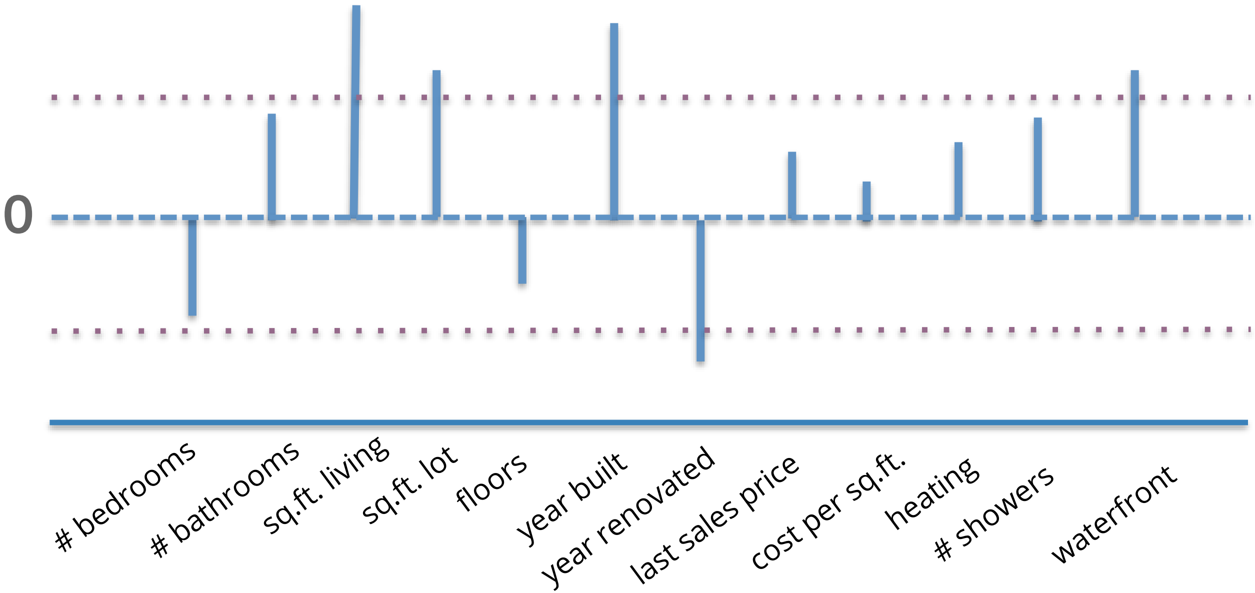
Non-zero coefficients would be considered selected as important.



Ridge for Feature Selection

Look at two related features #bathrooms and # showers.

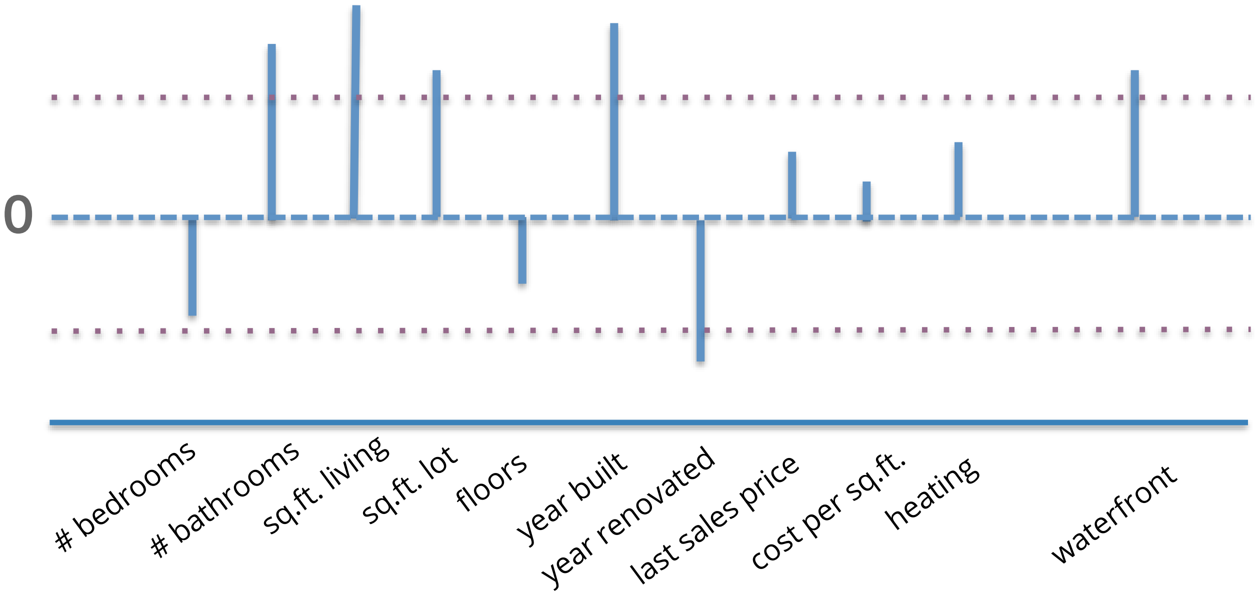
Our model ended up not choosing any features about bathrooms!



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

L1 norm: $||w||_1 = \sum_{j=1}^{D} |w_j|$

Change quality metric to minimize

$$\hat{w} = \underset{w}{\operatorname{argmin}}\, MSE(w) + \lambda ||w||_1$$

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

What if $\lambda = 0$?

$$\hat{w} = \underset{w}{\operatorname{argmin}}\, MSE(w) = \hat{w}_{OLS}$$

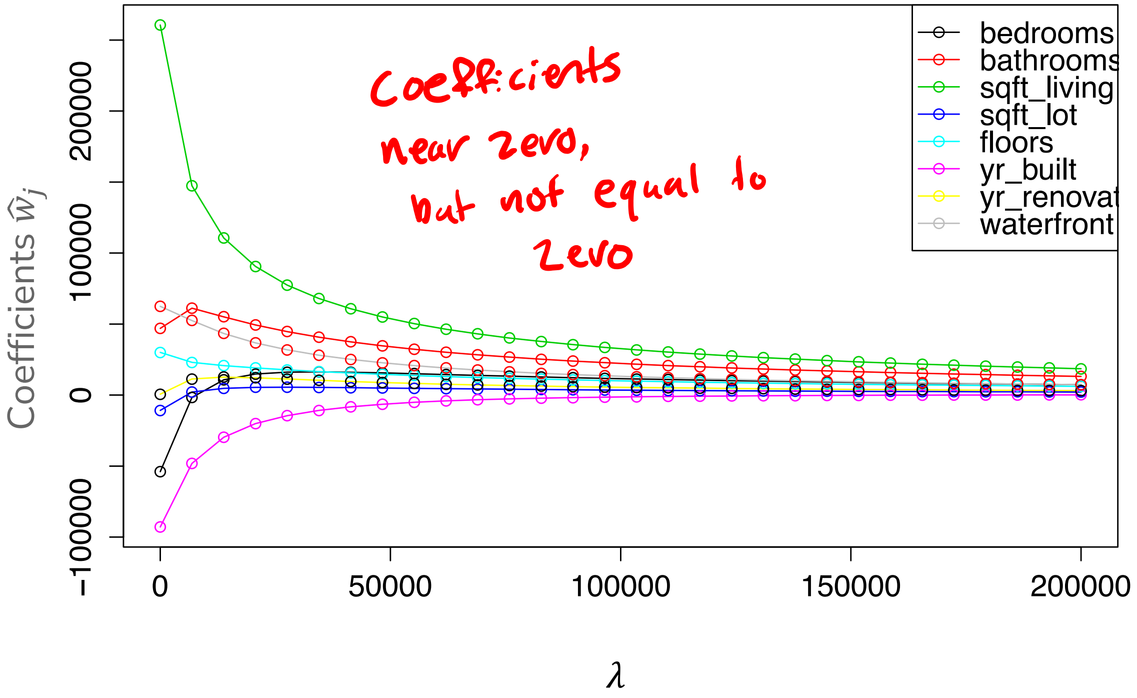
What if $\lambda = \infty$?

$$\hat{w} = \underset{w}{\operatorname{argmin}}\, \lambda ||w||_1 = \vec{0} = [0, 0, \dots, 0]$$

λ in between? $0 \le ||\hat{w}_{LASSO}||_1 \le ||\hat{w}_{OLS}||_1$

Ridge (L2) Coefficient Paths

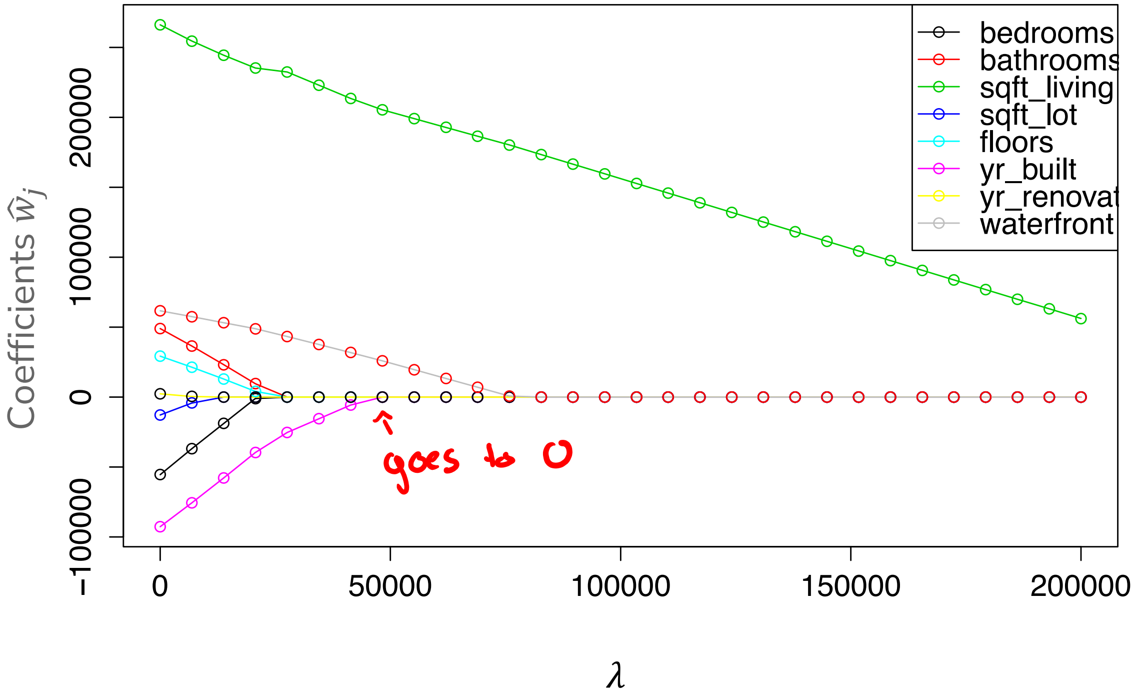
$R(w) = \|w\|_2^2$



LASSO (L1) Coefficient Paths

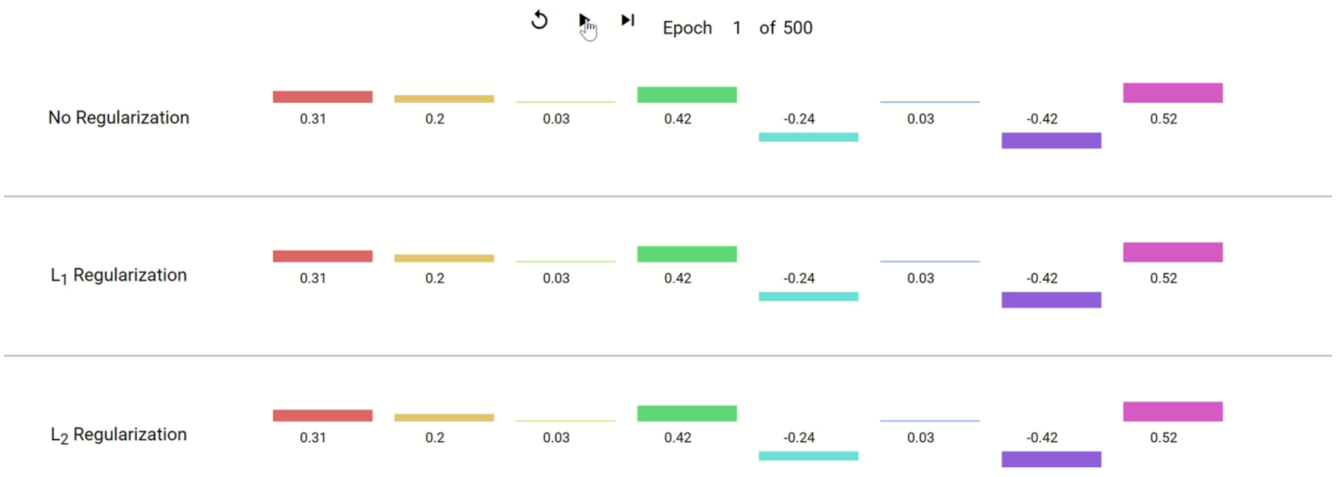
$R(w) = \|w\|_1$

sparse → more zero coefficients
→ fewer important features



Coefficient Paths – Another View

Example from Google's [Machine Learning Crash Course]



Demo

Similar demo to last time's with Ridge but using the LASSO penalty

slido

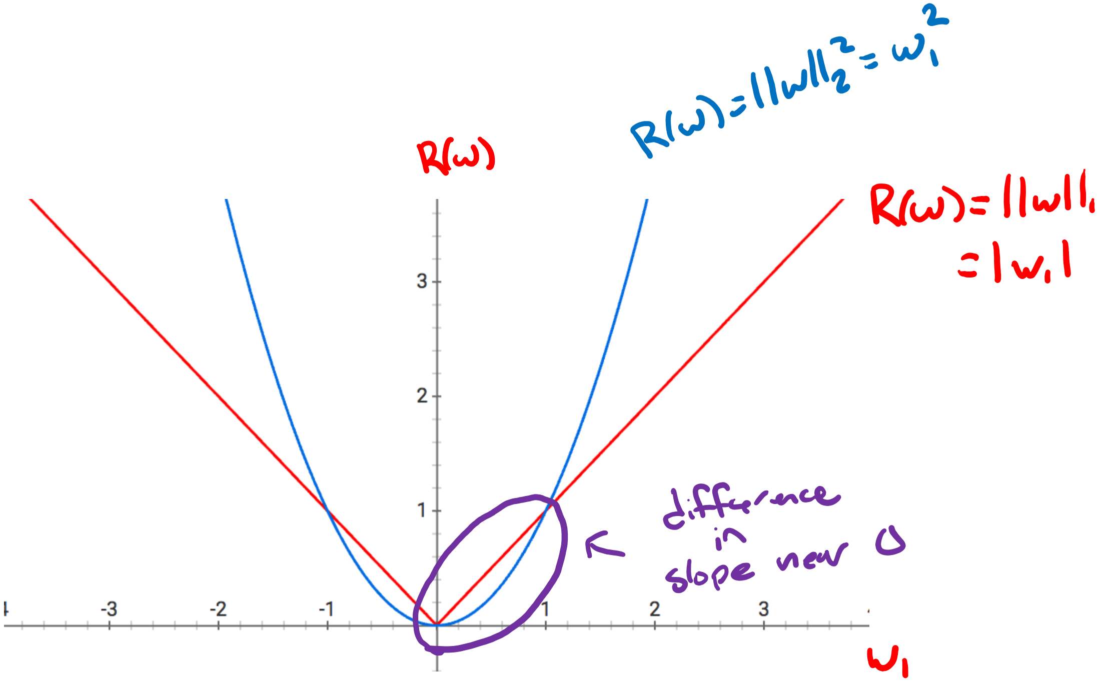
Group

2 minutes

No poll

$w = [w_0, w_1]$

Why might the shape of the L1 penalty cause more sparsity than the L2 penalty?



$\hat{w} = \underset{w}{\text{argmin}}\ L(w) + \lambda R(w)$ where $w = [w_0, w_1]$

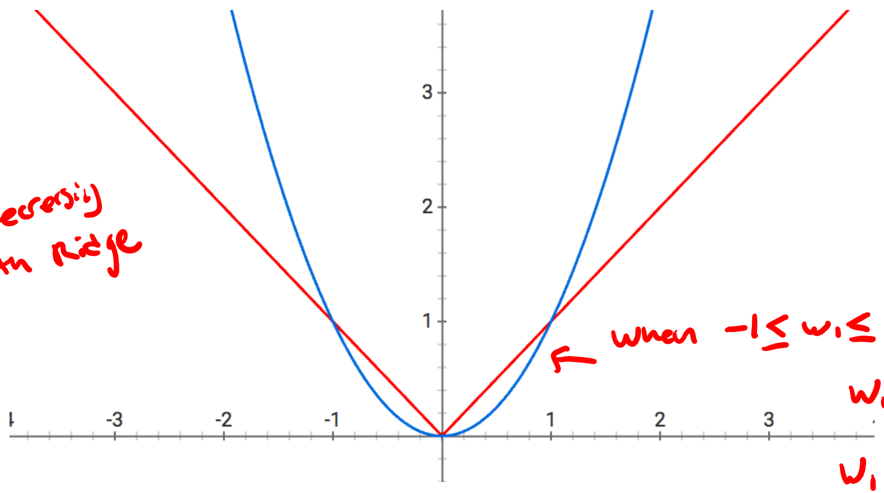
Sparsity

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

$R(w) = ||w||_2^2 = w_1^2$

if $w_1 = 10^{-2}$, $w_1^2 = 10^{-4}$



Diminishing returns decreasing $w_1 \le 1$ with Ridge

when $-1 \le w_1 \le 1$, then $w_1^2 \le |w_1|$

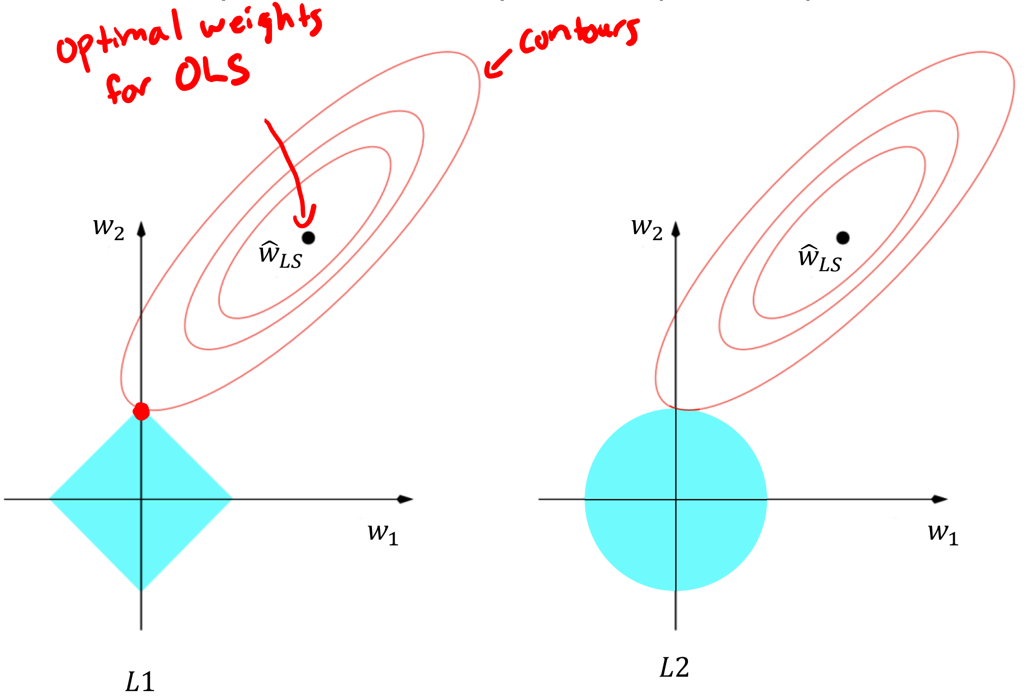
When w_j is small, w_j^2 is VERY small! Diminishing returns on decreasing w_j with Ridge penalty

Sparsity Geometry

Can also frame regularization

$\hat{w} = \underset{w}{\operatorname{argmin}} \; MSE(w)$ such that $R(w) \leq \beta$

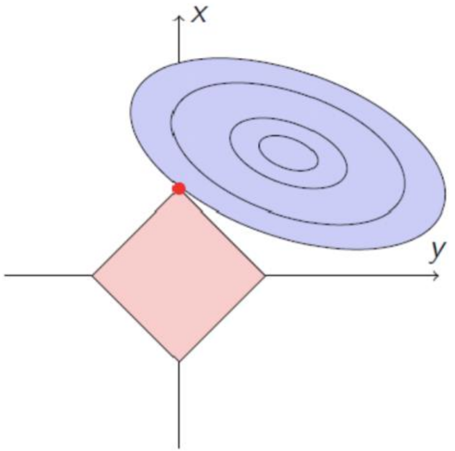
Another way to visualize why LASSO prefers sparse solutions



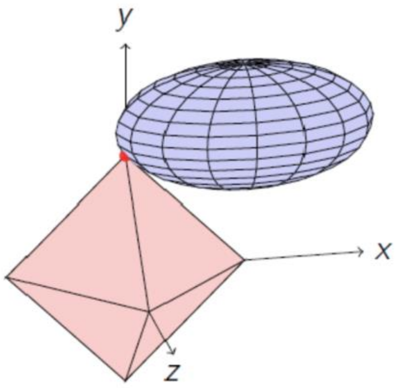
The L1 ball has spikes (places where some coefficients are 0)

More likely to hit a min at a spike

Sparsity Geometry



$L1$ (2 features)



$L1$ (3 features)

Brain Break

slido

Think

1 min

sli.do #cs416

How should we choose the best value of λ for LASSO?

✓a) Pick the λ that has the smallest $MSE(\hat{w})$ on the **validation set**

b) Pick the λ that has the smallest $MSE(\hat{w}) + \lambda||\hat{w}||_2^2$ on the **validation set**

c) Pick the λ that results in the most zero coefficients

d) Pick the λ that results in the fewest zero coefficients

e) None of the above

Same process as Ridge

Choosing λ

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 aglorithm

For almost every hyperparameter:

Pick hyperparameter that has the lowest $err_{val} = MSE_{val}(\hat{w})$

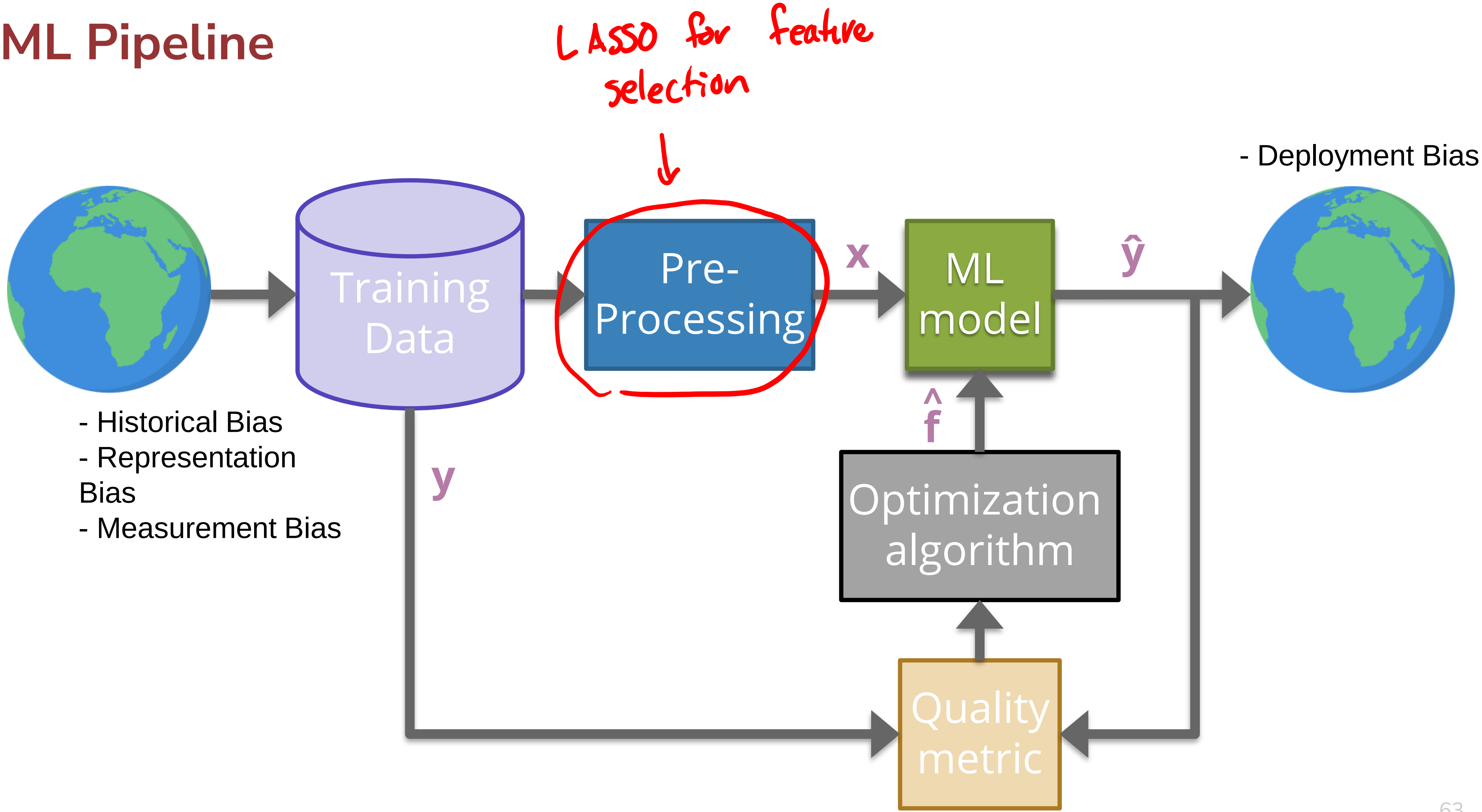
LASSO in Practice

A very common usage of LASSO is in feature selection. If you have a model with potentially many features you want to explore, you can use LASSO on a model with all the features and choose the appropriate λ to get the right complexity.

Then once you find the non-zero coefficients, you can identify which features are the most important to the task at hand*

* e.g., using domain-specific expertise

ML Pipeline


LASSO for feature selection
- Deployment Bias
Training Data
Pre-Processing
x
ML model
ŷ
f̂
- Historical Bias
- Representation Bias
- Measurement Bias
y
Optimization algorithm
Quality metric

De-biasing LASSO

As λ increases, the resulting model have higher bias and less variance.

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

- Recall Bias-Variance Tradeoff

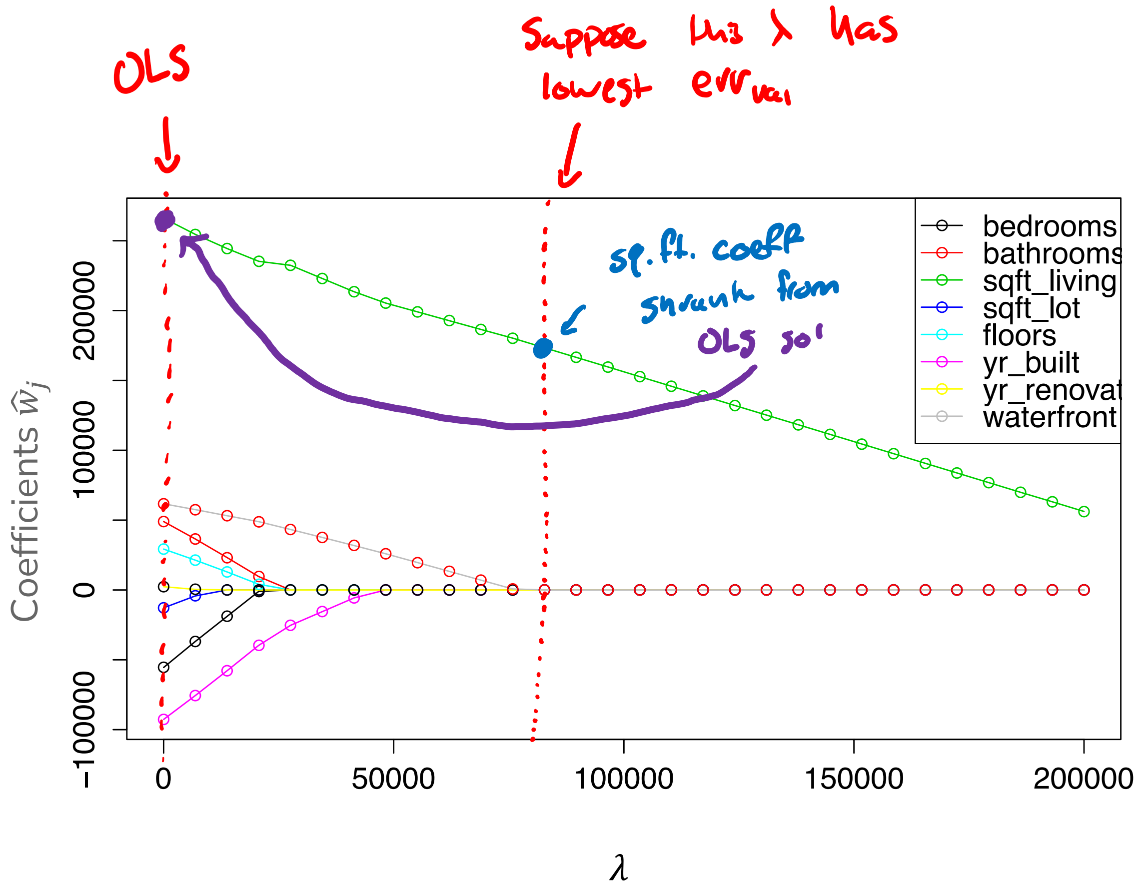
It's possible to try to remove the bias from the LASSO solution using the following steps

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

   Regular Linear Regression (no regularization)

Coefficients are no longer shrunk from their true values

LASSO (L1) Coefficient Paths



(De-biased) LASSO In Practice

preprocessing

1. Split the dataset into train, val, and test sets
2. Normalize features. Fit the normalization on the train set, apply that normalization on the train, val, and test sets.

   compute μ, σ for each feature
3. Use validation or cross-validation to find the value of λ that that results in a LASSO model with the lowest validation error.
4. Select the features of that model that have non-zero weights.
5. Train a Linear Regression model with only those features.
6. Evaluate on the test set.

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

$$\hat{w}_{ElasticNet} = \underset{w}{\operatorname{argmin}}\, MSE(w) + \lambda_1 ||w||_1 + \lambda_2 ||w||_2^2$$

Combines both to achieve best of both worlds!

A Big Grain of Salt

Be careful when interpreting the results of feature selection or feature importance in Machine Learning!

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

At the end of the day, the best models combine statistical insights with domain-specific expertise!

Results always in context of your experimental setup.

↳ i.e., change setup, possibly find completely different results for "most important"

Differences between L1 and L2 regularizations

L1 (LASSO):

- Introduces more sparsity to the model
- Less sensitive to outliers ← squaring usually sensitive to outliers
- Helpful for feature selection, making the model more interpretable
- More computational efficient as a model (due to the sparse solutions, so you have to compute less dot products)

L2 (Ridge):

- Makes the weights small (but not 0)
- More sensitive to outliers (due to the squared terms)
- Usually works better in practice in terms of accuracy

↑ not always! "No free lunch"

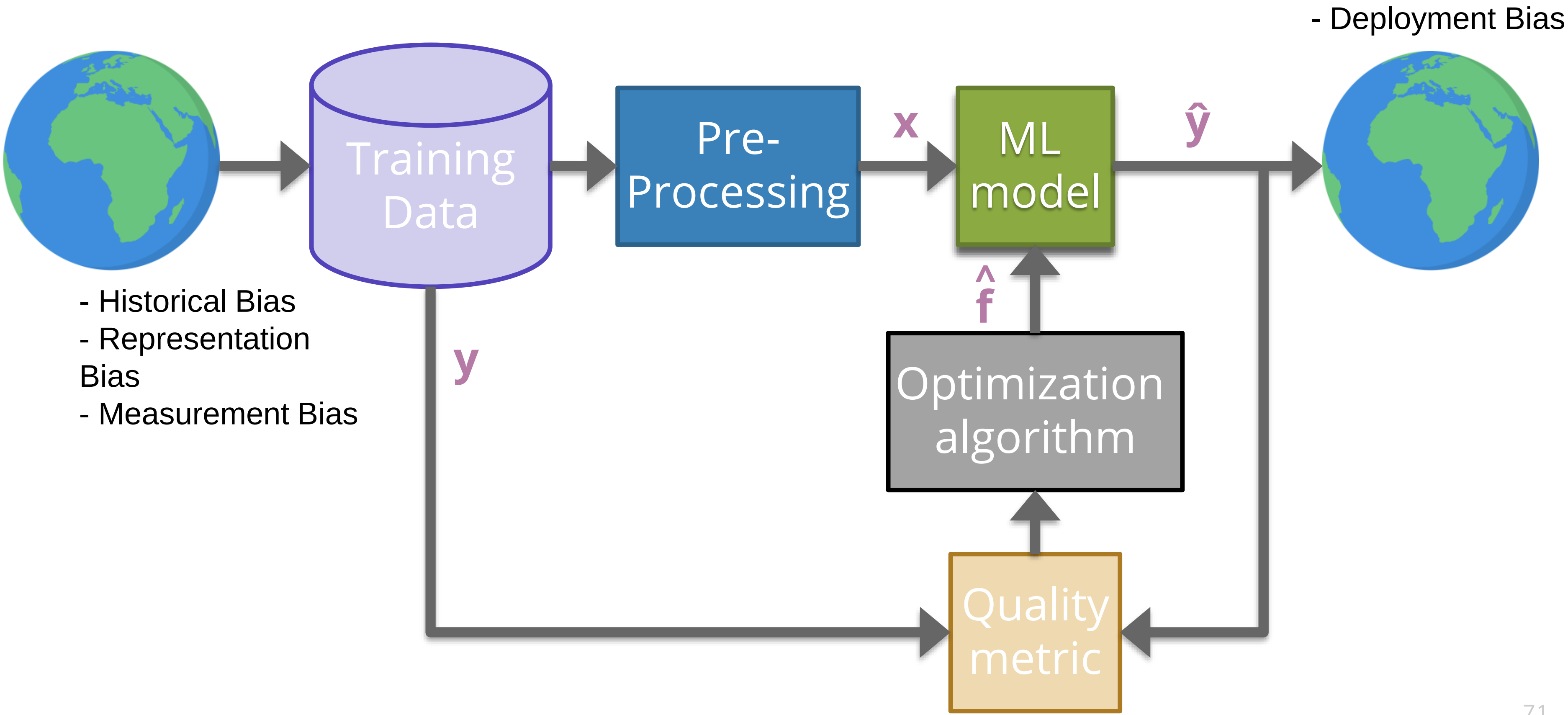
Recap

Theme: Using 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 λ is varied
- Interpret LASSO coefficient path plot
- Compare and contrast LASSO (L1) and Ridge (L2)

ML Pipeline


- Deployment Bias
Training Data
Pre-Processing
x
ML model
ŷ
f̂
y
Optimization algorithm
Quality metric
- Historical Bias
- Representation Bias
- Measurement Bias