Lasso Regression: Regularization for feature selection

Symptom of overfitting

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

Very large coefficients (+/-)
Consider specific total cost

Want to balance:

i. How well function fits data
   
ii. Magnitude of coefficients

**Total cost** = measure of fit + measure of magnitude of coefficients

\[ \text{RSS}(\mathbf{w}) + \| \mathbf{w} \|_2^2 = \sum_{j=0}^{D} w_j^2 \]
Consider resulting objective

What if $\mathbf{w}$ selected to minimize

$$\text{RSS}(\mathbf{w}) + \lambda \| \mathbf{w} \|^2_2$$

Ridge regression
(a.k.a $L_2$ regularization)

Measure of magnitude of regression coefficient

What summary # is indicative of size of regression coefficients?

- Sum? $w_0 = 1,527,301$, $w_1 = -1,605,253$; $w_0 + w_1 = \text{small}$ #

- Sum of absolute value?

- Sum of squares ($L_2$ norm)
Feature selection task

Why might you want to perform feature selection?

Efficiency:
- If size($w$) = 100B, each prediction is expensive
- If $\hat{w}$ sparse, computation only depends on # of non-zeros

Interpretability:
- Which features are relevant for prediction?
Sparsity: Housing application

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 your mind

very sad
very happy

Activity in which brain regions can predict happiness?
Option 1: All subsets or greedy variants

Find best model of size: 0

RSS($\hat{w}$) vs. # of features

- # bedrooms
- # bathrooms
- sq.ft. living
- sq.ft. lot
- floors
- year built
- year renovated
- waterfront
Find best model of size: 1

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

# of features
Find best model of size: 1

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

# of features

RSS(\(\mathbf{w}\))

0 1
Find best model of size: 1

# of features

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

Find best model of size: 1

# of features
Find best model of size: 1

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

# of features

RSS(w)
Find best model of size: 1

RSS(\(w\))

0 1

# of features

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

Find best model of size: 2

RSS(\(w\))

0 1 2

# of features

- # bedrooms
- # bathrooms
- sq.ft. living
- sq.ft. lot
- floors
- year built
- year renovated
- waterfront
Note: Not necessarily nested!

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

Note: Not necessarily nested!

- # bedrooms
- # bathrooms
- sq.ft. living
- sq.ft. lot
- floors
- year built
- year renovated
- waterfront
Find best model of size: 3

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

Find best model of size: 4

- # bedrooms
- # bathrooms
- sq.ft. living
- sq.ft. lot
- floors
- year built
- year renovated
- waterfront
Find best model of size: 5

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

Find best model of size: 6

- # bedrooms
- # bathrooms
- sq.ft. living
- sq.ft. lot
- floors
- year built
- year renovated
- waterfront
Find best model of size: 7

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

Find best model of size: 8

- # bedrooms
- # bathrooms
- sq.ft. living
- sq.ft. lot
- floors
- year built
- year renovated
- waterfront
Choosing model complexity?

Option 1: Assess on validation set

Option 2: Cross validation

Option 3+: Other metrics for penalizing model complexity like BIC...

Complexity of “all subsets”

How many models were evaluated?
- each indexed by features included

\[
\begin{align*}
  y_1 &= \varepsilon_1 \\
  y_i &= w_0 h_0(x) + \varepsilon_i \\
  y_i &= w_1 h_1(x) + \varepsilon_i \\
  \vdots \\
  y_i &= w_0 h_0(x) + w_1 h_1(x) + \varepsilon_i \\
  \vdots \\
  y_i &= w_0 h_0(x) + w_1 h_1(x) + \ldots + w_D h_D(x) + \varepsilon_i
\end{align*}
\]

\[2^D = 2^{41} = 30,223,169,468,014,889,521,119,403,232,307,487,257,460,992,844,400,314,260,624,513,978,230,401, \ldots \]

Typically, computationally infeasible
Greedy algorithms

Forward stepwise:
Starting from simple model and iteratively add features most useful to fit

Backward stepwise:
Start with full model and iteratively remove features least useful to fit

Combining forward and backward steps:
In forward algorithm, insert steps to remove features no longer as important

Lots of other variants, too.

Option 2: Regularize
Ridge regression: $L_2$ regularized regression

Total cost =

- measure of fit + $\lambda$ measure of magnitude of coefficients

$$\text{RSS}(w) = w_0^2 + \cdots + w_D^2$$

Encourages small weights

but not exactly 0

Coefficient path – ridge
Using regularization for feature selection

Instead of searching over a discrete set of solutions, can we use regularization?
- Start with full model (all possible features)
- “Shrink” some coefficients exactly to 0
  - i.e., knock out certain features
- Non-zero coefficients indicate “selected” features

Thresholding ridge coefficients?

Why don’t we just set small ridge coefficients to 0?
Thresholding ridge coefficients?

Selected features for a given threshold value

Let’s look at two related features...

Nothing measuring bathrooms was included!
Thresholding ridge coefficients?

If only one of the features had been included...

Would have included bathrooms in selected model

Can regularization lead directly to sparsity?
Try this cost instead of ridge...

Total cost = \( \text{measure of fit} + \lambda \text{measure of magnitude of coefficients} \)

\[ \text{RSS}(w) + \lambda \|w\|_1 = |w_0| + \ldots + |w_D| \]

Leads to sparse solutions!

**Lasso regression**
(a.k.a. \( L_1 \) regularized regression)

Lasso regression: \( L_1 \) regularized regression

Just like ridge regression, solution is governed by a continuous parameter \( \lambda \)

\[ \text{RSS}(w) + \lambda \|w\|_1 \]

- If \( \lambda = 0 \): \( \hat{w}^{\text{lasso}} = \hat{w}^{\text{LS}} \) (unreg. soln)
- If \( \lambda = \infty \): \( \hat{w}^{\text{lasso}} = 0 \)
- If \( \lambda \) in between: \( 0 \leq \|\hat{w}^{\text{lasso}}\|_1 \leq \|\hat{w}^{\text{LS}}\|_1 \)
Coefficient path – ridge

- Coefficients $\hat{w}_j$ as a function of $\lambda$
- Features: bedrooms, bathrooms, sqft_living, sqft_lot, floors, yr_built, yr_renovated, waterfront
- All coefficients decrease as $\lambda$ increases.
- For large $\lambda$, $\hat{w}_j = 0$ for most features.

Coefficient path – lasso

- Coefficients $\hat{w}_j$ as a function of $\lambda$
- Features: bedrooms, bathrooms, sqft_living, sqft_lot, floors, yr_built, yr_renovated, waterfront
- Only features with non-zero $\hat{w}_j$ are kept in the model.
- For large $\lambda$, all coefficients except those with large weight on sqft_living are set to zero.
Revisit polynomial fit demo

What happens if we refit our high-order polynomial, but now using lasso regression?

Will consider a few settings of $\lambda$ ...

How to choose $\lambda$: Cross validation
If sufficient amount of data...

- **Training set**
- **Validation set**
- **Test set**

Fit \( \hat{w}_\lambda \)

Test performance of \( \hat{w}_\lambda \) to select \( \lambda^* \)

Assess generalization error of \( \hat{w}_{\lambda^*} \)

---

Start with smallish dataset

All data
Still form test set and hold out

Rest of data  Test set

How do we use the other data?

Rest of data

use for both training and validation, but not so naively
Recall naïve approach

Is validation set enough to compare performance of $\hat{w}_\lambda$ across $\lambda$ values?

No

Choosing the validation set

Didn’t have to use the last data points tabulated to form validation set

Can use any data subset
Choosing the validation set

Valid. set

Which subset should I use?

ALL! average performance over all choices

K-fold cross validation

Rest of data

Preprocessing: Randomly assign data to K groups

(use same split of data for all other steps)
K-fold cross validation

For k=1,...,K
1. Estimate $\hat{w}_\lambda^{(k)}$ on the training blocks
2. Compute error on validation block: $\text{error}_k(\lambda)$
**K-fold cross validation**

For $k=1,\ldots,K$

1. Estimate $\hat{\mathbf{w}}_{\lambda}^{(k)}$ on the training blocks
2. Compute error on validation block: $\text{error}_k(\lambda)$
For $k=1,...,K$
1. Estimate $\hat{w}_\lambda^{(k)}$ on the training blocks
2. Compute error on validation block: $\text{error}_k(\lambda)$

Compute average error: $\text{CV}(\lambda) = \frac{1}{K} \sum_{k=1}^{K} \text{error}_k(\lambda)$

Repeat procedure for each choice of $\lambda$
Choose $\lambda^*$ to minimize $\text{CV}(\lambda)$

Choose $\lambda^*$ to minimize $\text{CV}(\lambda)$

$\lambda^* = \lambda_3$
What value of K?

Formally, the best approximation occurs for validation sets of size 1 ($K=N$)

leave-one-out cross validation

Computationally intensive
- requires computing $N$ fits of model per $\lambda$

Typically, $K=5$ or 10

5-fold CV

10-fold CV

Choosing $\lambda$ via cross validation for lasso

Cross validation is choosing the $\lambda$ that provides best predictive accuracy

Tends to favor less sparse solutions, and thus smaller $\lambda$, than optimal choice for feature selection

c.f., "Machine Learning: A Probabilistic Perspective", Murphy, 2012 for further discussion
Practical concerns with lasso

Debiasing lasso

Lasso shrinks coefficients relative to LS solution → more bias, less variance

Can reduce bias as follows:
1. Run lasso to select features
2. Run least squares regression with only selected features

“Relevant” features no longer shrunk relative to LS fit of same reduced model

Figure used with permission of Mario Figueiredo (captions modified to fit course)
Issues with standard lasso objective

1. With group of highly correlated features, lasso tends to select amongst them arbitrarily
   - Often prefer to select all together

2. Often, empirically ridge has better predictive performance than lasso, but lasso leads to sparser solution

Elastic net aims to address these issues
   - hybrid between lasso and ridge regression
   - uses $L_1$ and $L_2$ penalties

See Zou & Hastie ’05 for further discussion

Summary for feature selection and lasso regression
Impact of feature selection and lasso

Lasso has changed machine learning, statistics, & electrical engineering

But, for feature selection in general, be careful about interpreting selected features
- selection only considers features included
- sensitive to correlations between features
- result depends on algorithm used
- there are theoretical guarantees for lasso under certain conditions

What you can do now...

• Describe "all subsets" and greedy variants for feature selection
• Analyze computational costs of these algorithms
• Formulate lasso objective
• Describe what happens to estimated lasso coefficients as tuning parameter $\lambda$ is varied
• Interpret lasso coefficient path plot
• Contrast ridge and lasso regression
• Implement K-fold cross validation to select lasso tuning parameter $\lambda$