Lasso Regression:
Regularization for feature selection
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

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

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

Activity in which brain regions can predict happiness?
Explaining Predictions

P(\text{guitar}) = 0.32

P(\text{guitar}) = 0.24

P(\text{dog}) = 0.21

“Why should I trust you?”: Explaining the Predictions of Any Classifier. Ribeiro, Singh & G. KDD 16
Option 1: All subsets or greedy variants
Find best model of for each size

- # bedrooms
- # bathrooms
- sq.ft. living
- sq.ft. lot
- floors
- year built
- year renovated
- waterfront
Complexity of “all subsets”

\[
y_i = \varepsilon_i \\
y_i = w_0 h_0(x_i) + \varepsilon_i \\
y_i = w_1 h_1(x_i) + \varepsilon_i \\
\vdots \\
y_i = w_0 h_0(x_i) + w_1 h_1(x_i) + \varepsilon_i \\
\vdots \\
y_i = w_0 h_0(x_i) + w_1 h_1(x_i) + \ldots + w_D h_D(x_i) + \varepsilon_i
\]

\[
\begin{bmatrix}
0 & 0 & \ldots & 0 & 0 & 0 \\
1 & 0 & \ldots & 0 & 0 & 0 \\
0 & 1 & \ldots & 0 & 0 & 0 \\
1 & 1 & \ldots & 0 & 0 & 0 \\
\vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\
1 & 1 & \ldots & 1 & 1 & 1
\end{bmatrix}
\]

\[2^8 = 256\]
\[2^{30} = 1,073,741,824\]
\[2^{1000} = 1.071509 \times 10^{301}\]
\[2^{1000} \text{B} = \text{HUGE!}!!!!!!\]

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 = $\text{measure of fit} + \lambda \text{measure of magnitude of coefficients}$

$\text{RSS}(w)$

$||w||_2^2 = w_0^2 + \ldots + w_D^2$
Coefficient path – ridge

\[ \hat{w}_j \]

\[ \lambda \]

- bedrooms
- bathrooms
- sqft_living
- sqft_lot
- floors
- yr_built
- yr_renovation
- waterfront
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 \textit{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
Thresholding ridge coefficients?

Let’s look at two related features...

Nothing measuring bathrooms was included!
Thresholding ridge coefficients?

If only one of the features had been included...
Thresholding ridge coefficients?

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 \sum |w_i|

\Rightarrow \text{Lasso regression (a.k.a. } L_1 \text{ regularized regression)}

Leads to sparse solutions!
Lasso regression: $L_1$ regularized regression

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

$$\text{RSS}(w) + \lambda \left\| w \right\|_1$$

**tuning parameter = balance of fit and sparsity**

If $\lambda = 0$:

If $\lambda = \infty$:

If $\lambda$ in between:
Coefficient path – ridge
Coefficient path – lasso
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 $\lambda$