Regularized Linear Regression

Matt Drury

Based on Prior Work of RH and JFO

- Relate regularization to feature selection in linear regression
- Compare and contrast Lasso and Ridge regression.
- Build test error curve to determine optimal level or regularization

“HELP, my model is overfitting!”

You have a couple of options...

1. Get more data… (not usually possible/practical)
2. **Regularization:** restrict your model’s parameter space

Regularization Smooths out Overfitting

Play with the app at

http://madrury.github.io/smoothers/

In high dimensions, data is (usually) sparse

Linear regression can have high variance (i.e. tends to overfit) on high dimensional data (i.e. when it has access to many predictors

We'd like to restrict ("normalize", or "regularize") the model so that it has less variance.

Linear Regression (another review)

We model the world as:

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + ... + \beta_p X_p + \epsilon$$

We estimate the model parameters by minimizing:

$$\sum_{i=1}^{N} (y_i - \hat{\beta}_0 - \sum_{j=1}^{p} x_{ij} \hat{\beta}_j)^2$$

Ridge Regression

(Linear Regression w/ Tikhonov (L2) Regularization)

We model the world as:

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \ldots + \beta_p X_p + \epsilon$$

(same as before)

We estimate the model parameters by minimizing:

$$\sum_{i=1}^{N}(y_i - \hat{\beta}_0 - \sum_{j=1}^{p} x_{ij}\hat{\beta}_j)^2 + \overset{\text{(the “regularization” parameter)}}{\boxed{\lambda}} \underbrace{\underline{\sum_{i=1}^{p} \hat{\beta}_i^2}}_{\text{(new term!)}}$$

Ridge Regression

$$\sum_{i=1}^{N}(y_i - \hat{\beta}_0 - \sum_{j=1}^{p} x_{ij}\hat{\beta}_j)^2 + \lambda \sum_{i=1}^{p} \hat{\beta}_i^2$$

What if we set the lambda equal to zero?

What does the new term accomplish?

What happens to a features whose corresponding coefficient value (beta) is zero?

Ridge Regression

$$\sum_{i=1}^{N}(y_i - \hat{\beta}_0 - \sum_{j=1}^{p} x_{ij}\hat{\beta}_j)^2 + \lambda \sum_{i=1}^{p} \hat{\beta}_i^2$$

Notice, we do not penalize B_0.

Changing lambda changes the amount that large coefficients are penalized.

Increasing lambda increases the model's bias and decreases its variance. ← this is cool!

Ridge Regression

$$\sum_{i=1}^{N}(y_i - \hat{\beta}_0 - \sum_{j=1}^{p} x_{ij}\hat{\beta}_j)^2 + \lambda \sum_{i=1}^{p} \hat{\beta}_i^2$$

We have basically added a “slider” to our model.

Changing lambda changes the amount of bias and variance of our model.
The goal is to find a sweet spot.

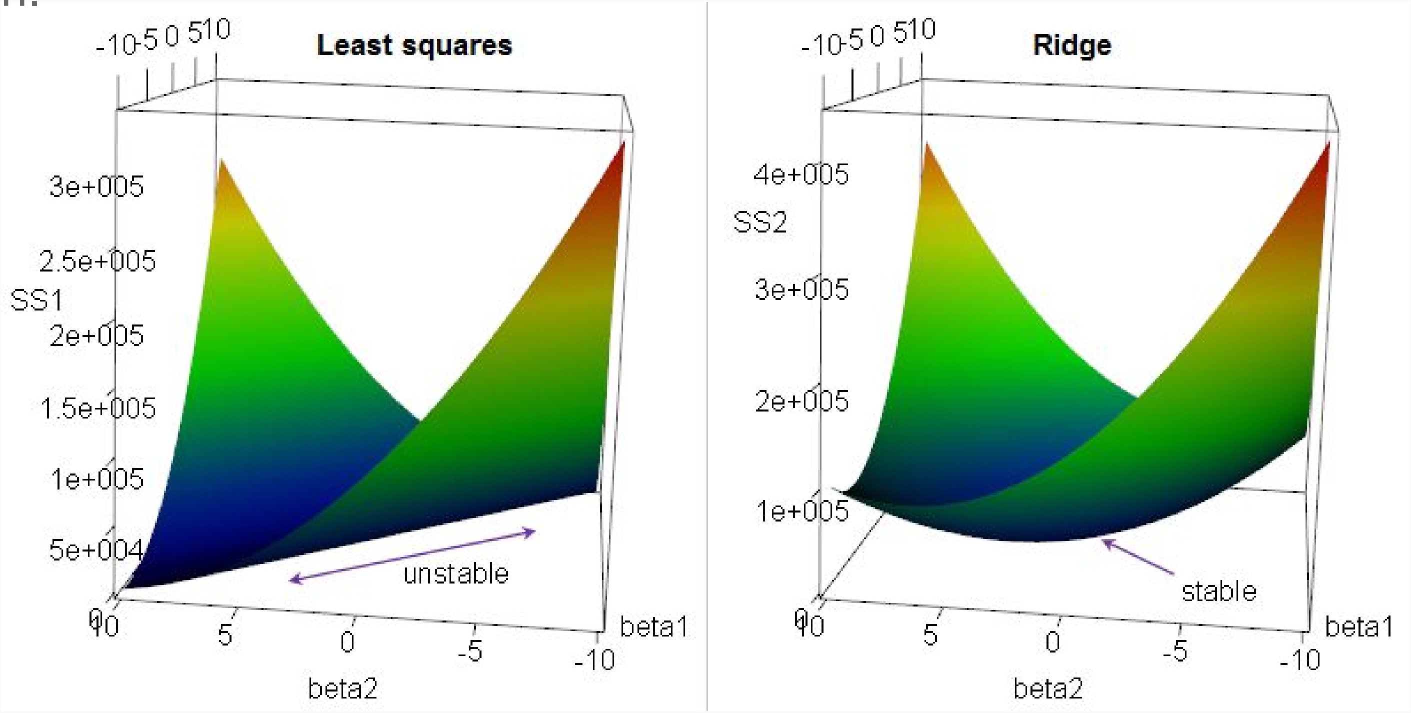
Regularization Smooths out Overfitting

Let’s look at this again with more context...

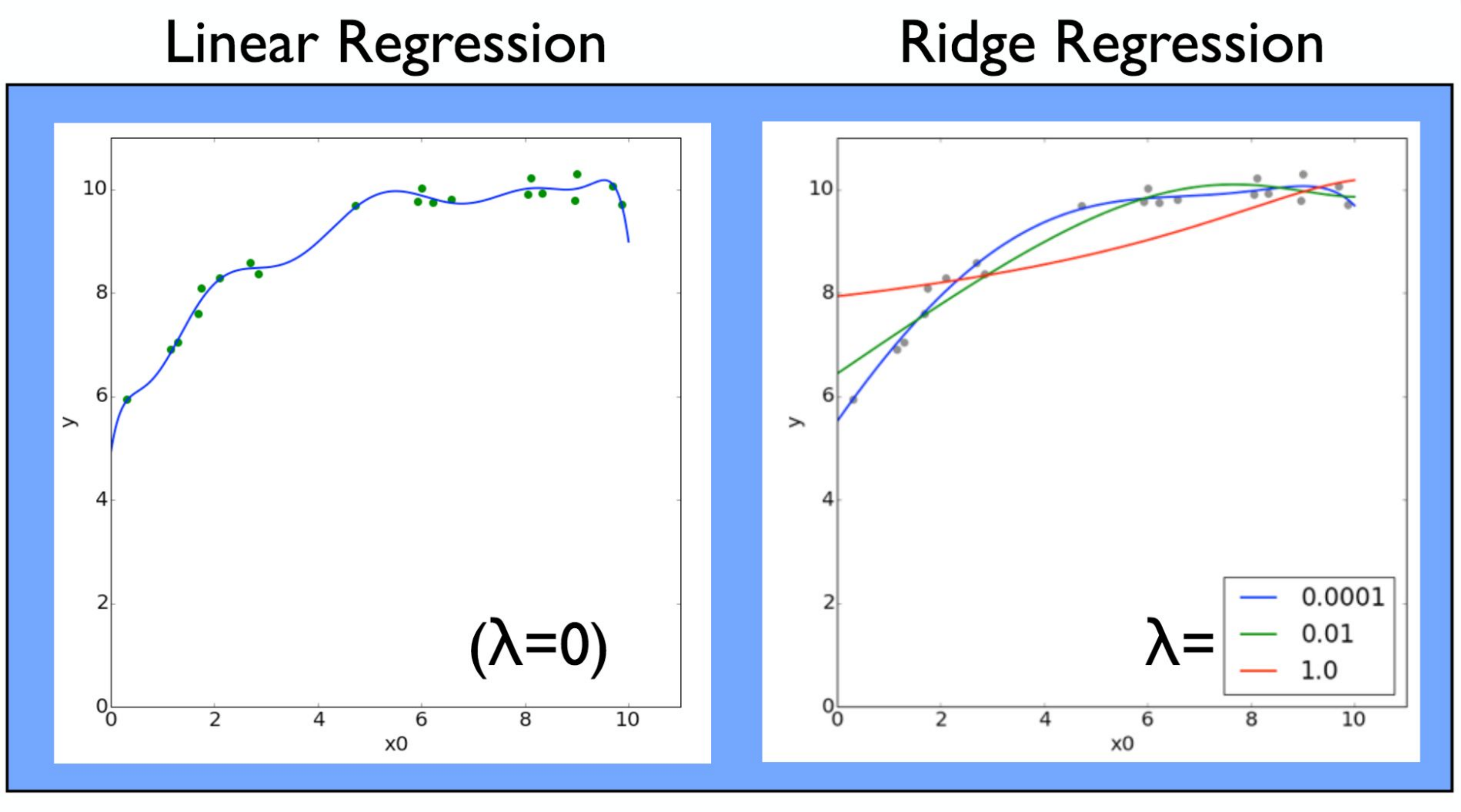
http://madrury.github.io/smoothers/

Why Is it Called That?

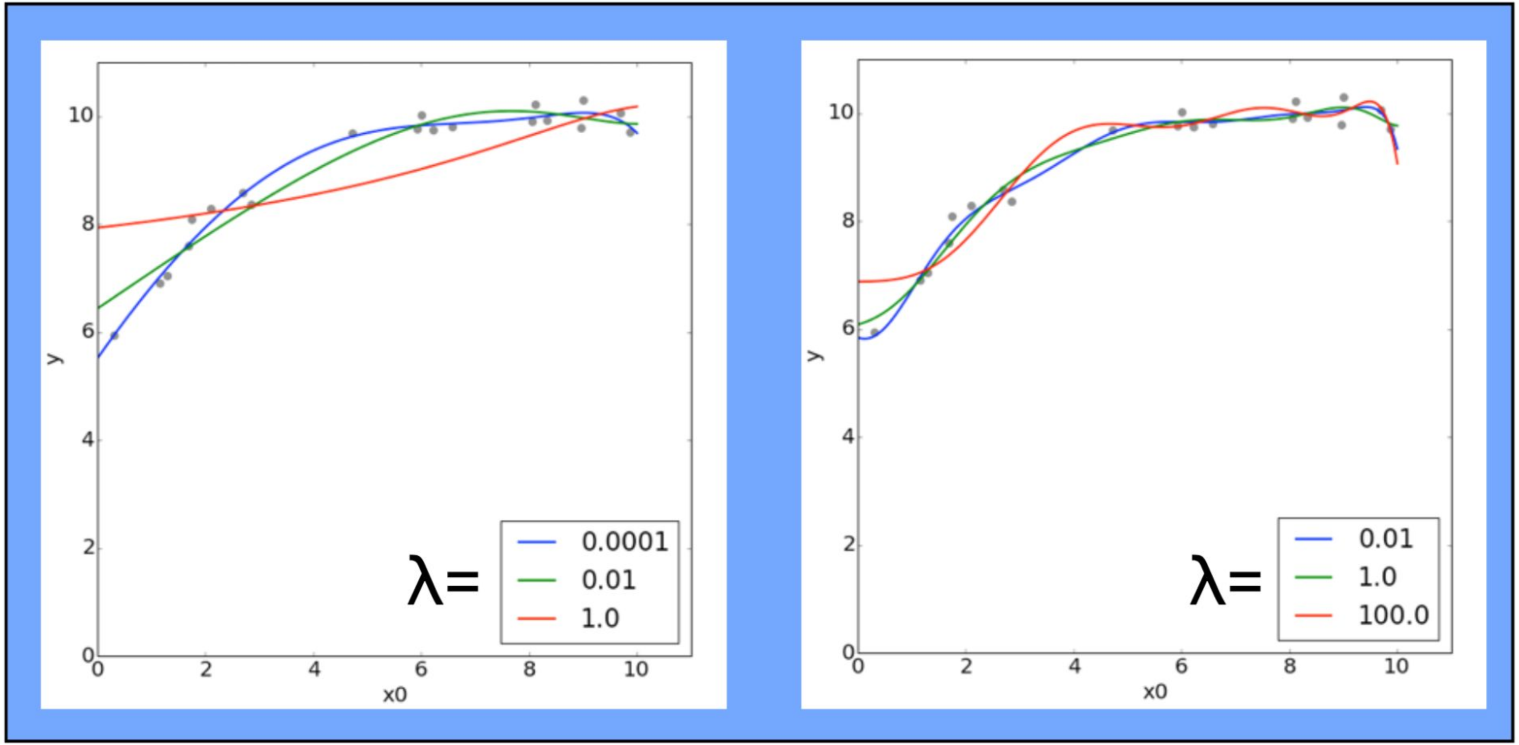
Ridge Regression was originally introduced to fix perfect collinearity in linear regression.



Ridge Regression



Ridge Regression



Normalized Data

Non-Normalized Data

Single value for λ assumes features are on the same scale!!

LASSO Regression

(Linear Regression w/ LASSO (L1) Regularization)

We model the world as:

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \ldots + \beta_p X_p + \epsilon$$

(same as before)

We estimate the model parameters to minimizing:

(the “regularization” parameter)

$$\sum_{i=1}^{N}(y_i - \hat{\beta}_0 - \sum_{j=1}^{p} x_{ij}\hat{\beta}_j)^2 + \lambda \sum_{i=1}^{p} |\hat{\beta}_i|$$

(absolute value instead of squared)

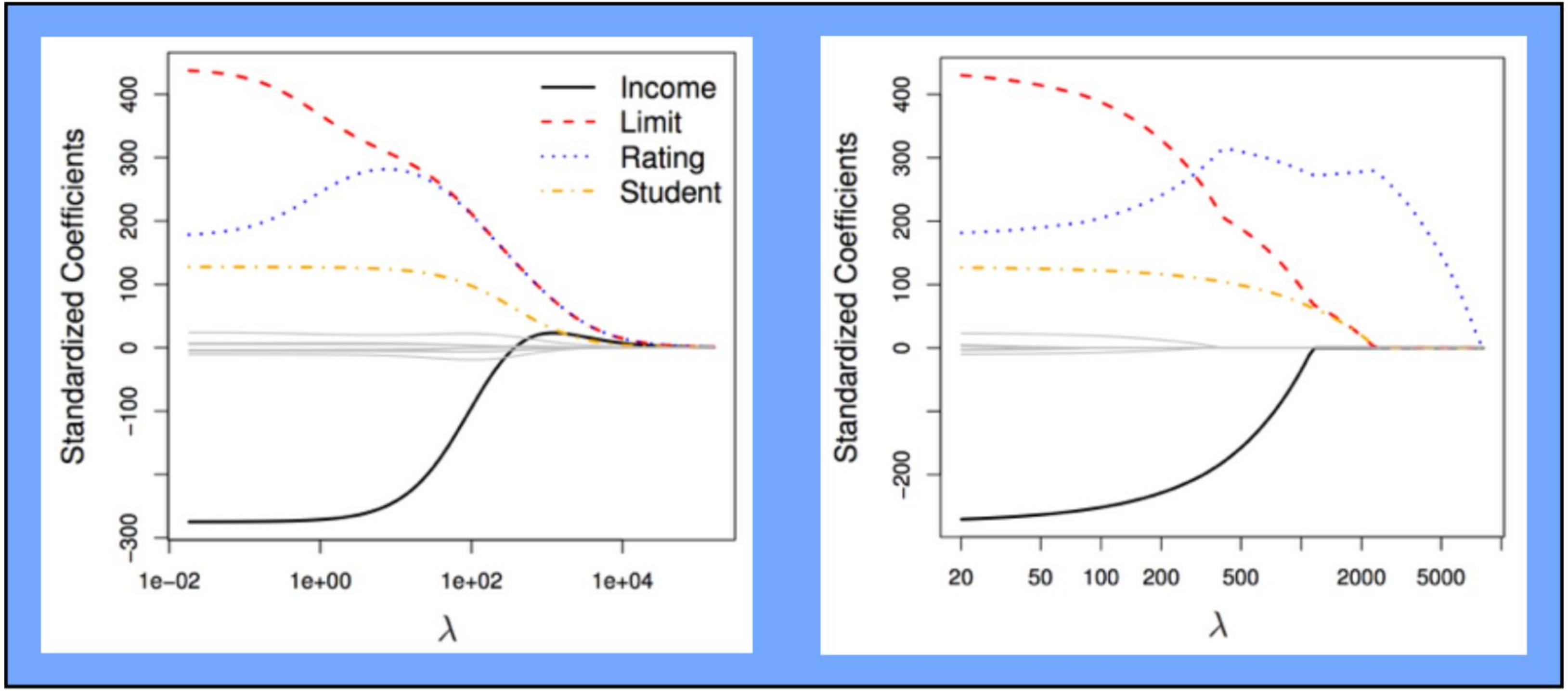
LASSO Regression
(Linear Regression w/ LASSO (L1) Regularization)

Again, what is with that name?

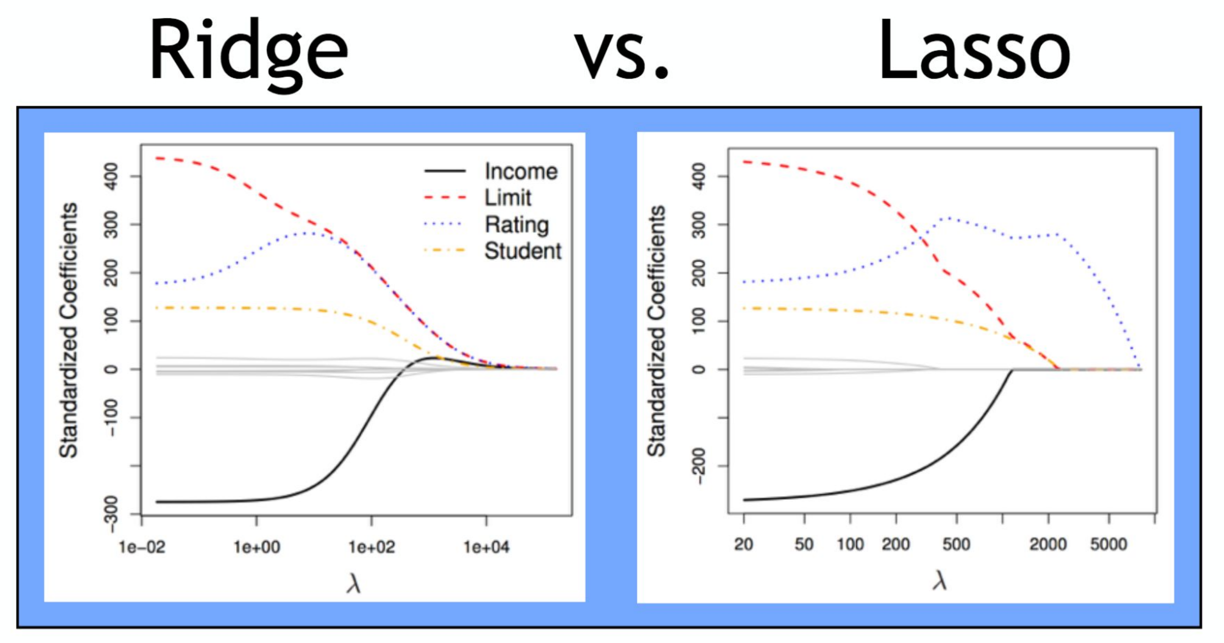
Least **A**bsolute **S**hrinkage and **S**election **O**perator.

Ridge vs LASSO

Ridge vs. Lasso

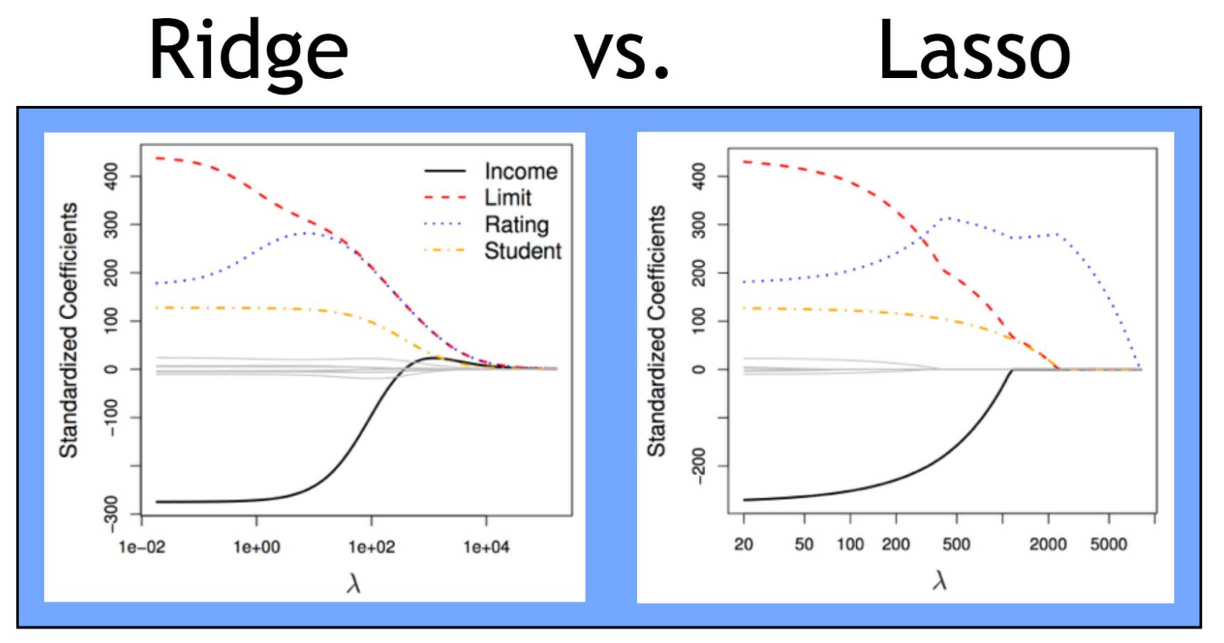


Ridge vs LASSO



Ridge makes the estimated parameters smaller as the regularization strength increases, but **they never become exactly zero.**

Ridge vs LASSO



Lasso will eventually make all the estimated parameters **exactly zero.**

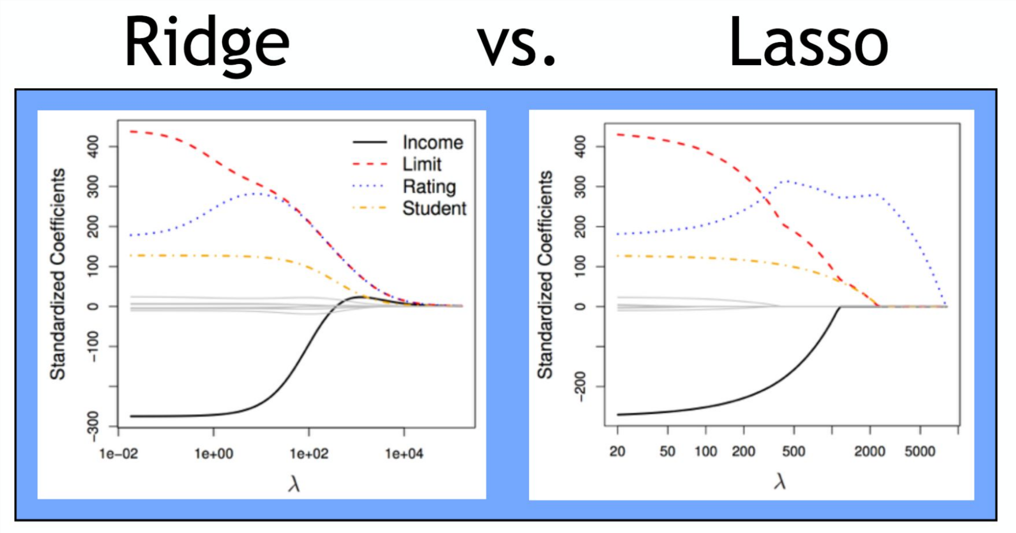
We say that **LASSO** creates **sparse models.**

Ridge vs LASSO: Summary

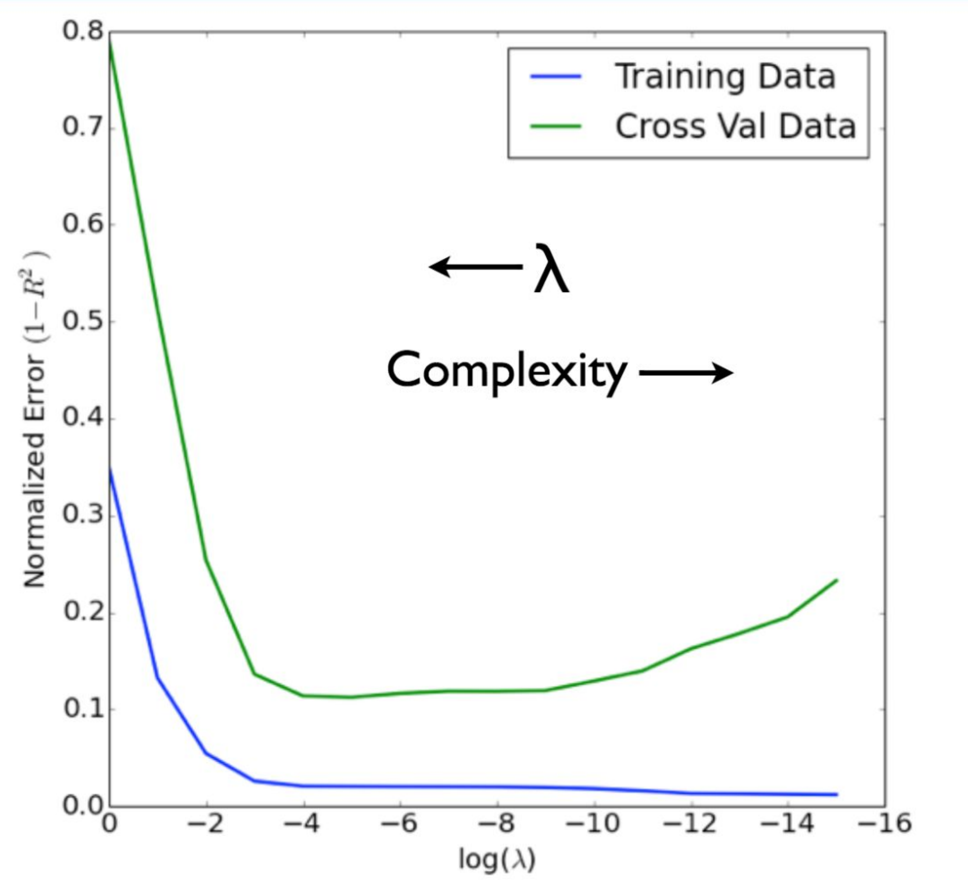
- Ridge forces parameters to be small + Ridge is computationally easier because it is differentiable
- Lasso tends to set coefficients exactly equal to zero
  - This is useful as a sort-of “diciplined feature selection” mechanism,
  - leads to “sparse” models, and

Which is better depends on your dataset and your goals!

True sparse models will benefit from lasso; true dense models will benefit from ridge.



Chose lambda via Cross-Validation



scikit-learn

Classes:

- sklearn.linear_model.**LinearRegression**(...)
- sklearn.linear_model.**Ridge**(alpha=my_alpha, …)
- sklearn.linear_model.**Lasso**(alpha=my_alpha, …)

All have these methods:

- fit(X, y)
- predict(X)
- score(X, y)

Other Options: glmnet

This is unfortunately a place where R dominates python in terms of options.

For serious work, I recommend using the **glmnet** library in R, which supports:

- Ridge and LASSO.
- Ridge and LASSO for logistic regression, and other regression loss functions.
- Combinations of **both** ridge and LASSO, which have the best features of both.

https://web.stanford.edu/~hastie/glmnet/glmnet_alpha.html