# Data Science Methods for Clean Energy Research

Week 8 L1: 1) k-fold cross validation + finish resampling, 2) model selection and regularization

Feb 22, 2017

UNIVERSITY of WASHINGTON

This is not my dog, but he is a really good dog…

# Outline

- Quick review from last time
- Finish resampling methods
  - Cross-validation
- Linear model selection / regularization
  - Subset selection
  - Ridge regression
  - LASSO regression
  - Python
- Wrap up

# Topics last time

- Error in regression models
- Multiple regression
  - Python example
- Bootstrap and cross validation

Big picture concepts:

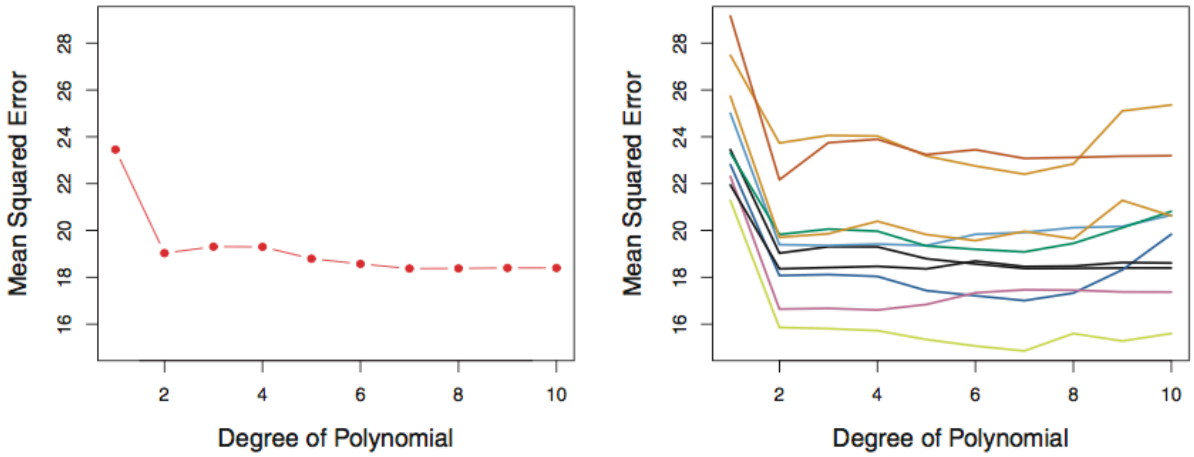
- The training error (e.g., RSS) will always be lower than the validation set or test set error
- Increasing the number of parameters (given P < N), always decreases the training error
- The bias/variance tradeoff emerges when we have to make decisions about how much data to withhold for validation

# Resampling methods (CH5, ISL)

- **Resampling concept**
- **Doing more with your data**
- **A big warning:** as introduced today, the resampling schemes are not to be used to generate independent predictions (**of Y**) for averaging later.
- This is a concept related to 'ensemble' methods, which we will discuss soon

# Cross Validation (k-fold)

- **Suppose you have one set of data and you have to decide how to break it into pieces for** training **and** validation
- **Simplest approach is the "validation set" , just break it into two pieces**
- **Example (Fig 5.2) looking at variations on** $Y = \beta_0 + \beta_0 + X^n$




Left: training MSE vs n for one data set
Right: training MSE vs n for 10 validation sets

**Riddle me this**, how many ways are there to choose two 500 data sets from 1000?

# Cross Validation (k-fold)

- **Since you only use a portion of your data in the training, the "validation set" approach will tend to** overestimate **your error!**
- **Leave One Out Cross Validation approach**

$$\mathrm{CV}_{(n)} = \frac{1}{n}\sum_{i=1}^{n}\mathrm{MSE}_i. \qquad (5.1)$$

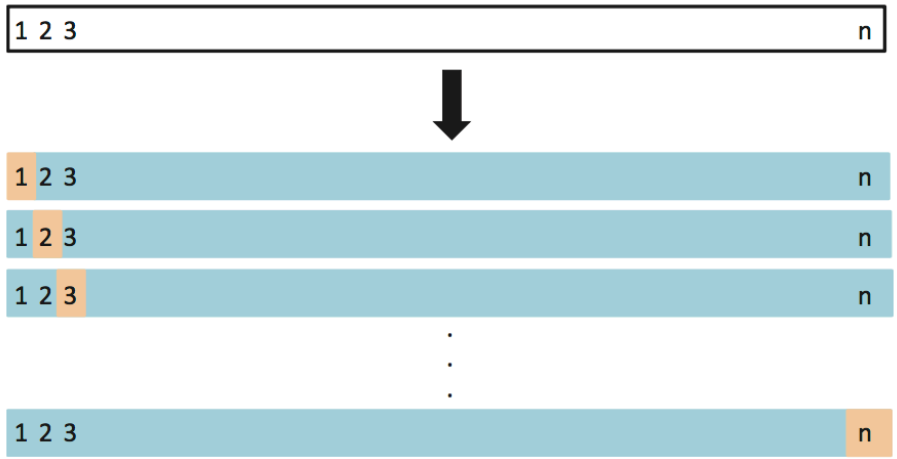
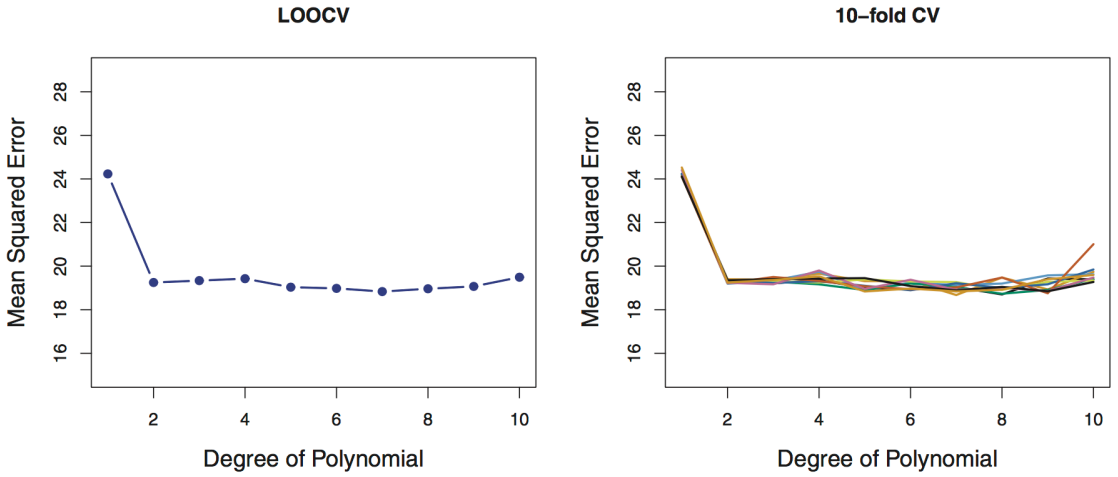



**FIGURE 5.3.** *A schematic display of LOOCV. A set of n data points is repeatedly split into a training set (shown in blue) containing all but one observation, and a validation set that contains only that observation (shown in beige). The test error is then estimated by averaging the n resulting MSE's. The first training set contains all but observation 1, the second training set contains all but observation 2, and so forth.*

# Cross Validation (k-fold)

- LOOCV is way more accurate (Fig 5.4), but more computationally expensive!
- An alternate is to break the data into larger pieces than n and n-1
- We break it into "k" folds of data, e.g. 5-fold. The 1st set is saved for validation, remaining k-1 sets are used for training

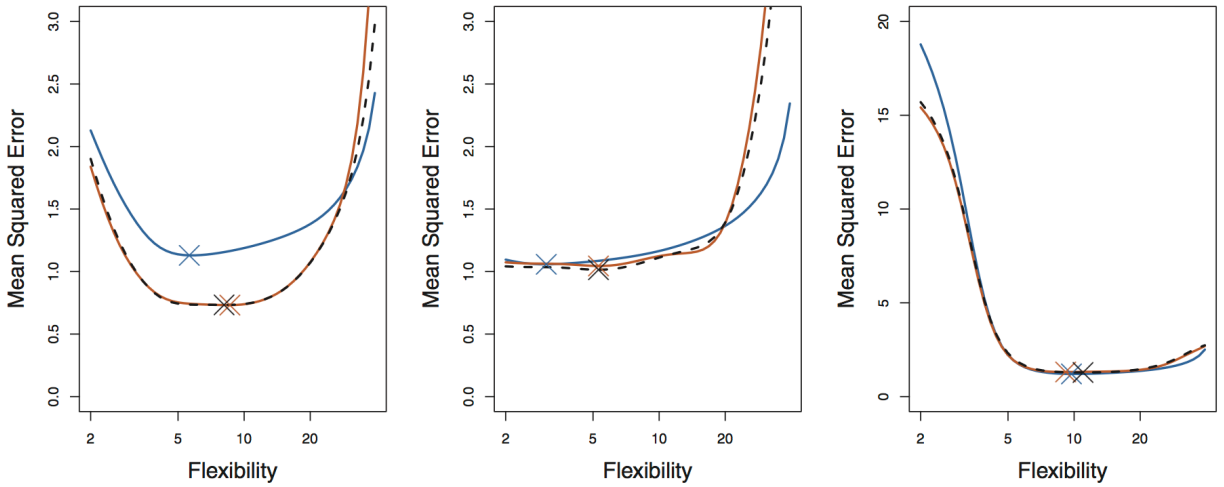
$$\mathrm{CV}_{(k)} = \frac{1}{k}\sum_{i=1}^{k}\mathrm{MSE}_i. \quad (5.3)$$



# Bias/variance tradeoff: use 5 or 10 folds

- Can anyone recall what we meant by the bias/variance tradeoff?
- Empirically people usually use 5 or 10 folds to avoid too much bias or variance in their resampling algorithm
- This is a great way to get a true estimate of your model's MSE

Fig 5.6 revisits Fig 2.9 in the context of k-fold cross validation



# Bootstrap

- **The bootstrap is one of the most versatile tools you will use in statistical analysis of data sets**
- **It involves resampling** with replacement **from your data set**
- **Algorithm:**
  - **Randomly draw, with replacement, some subset from your training data**
  - **Train your model and make an estimate of your coefficient and MSE**
  - **Rinse and repeat until the errors converge**

# The power of the bootstrap in one figure

## > Fig 5.10, estimates of some parameter, α

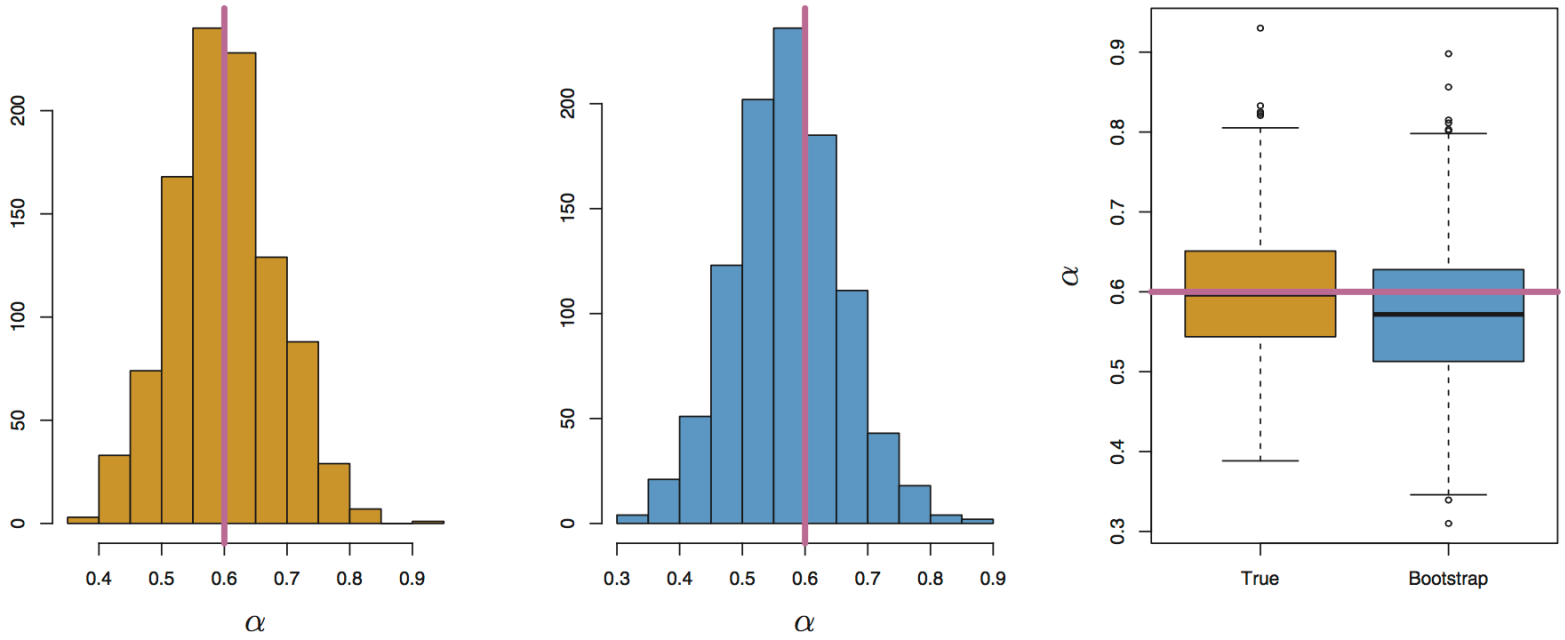



**FIGURE 5.10.** Left: *A histogram of the estimates of α obtained by generating 1,000 simulated data sets from the true population.* Center: *A histogram of the estimates of α obtained from 1,000 bootstrap samples from a single data set.* Right: *The estimates of α displayed in the left and center panels are shown as boxplots. In each panel, the pink line indicates the true value of α.*

# Take care

- **Depending on how large your bootstrap sample data set is, I recommend you avoid using the standard error formula (Eq 5.8) and instead you should use simply the standard deviation of the bootstrap estimates.**
  - Can anyone explain why?
- **In this context $\alpha$, could be any quantity from your training procedure (MSE, $\beta$, etc..)**

$$\mathrm{SE}_B(\hat{\alpha}) = \sqrt{\frac{1}{B-1}\sum_{r=1}^{B}\left(\hat{\alpha}^{*r} - \frac{1}{B}\sum_{r'=1}^{B}\hat{\alpha}^{*r'}\right)^2}. \qquad (5.8)$$

# Linear model selection and regularization (CH6 ISL)

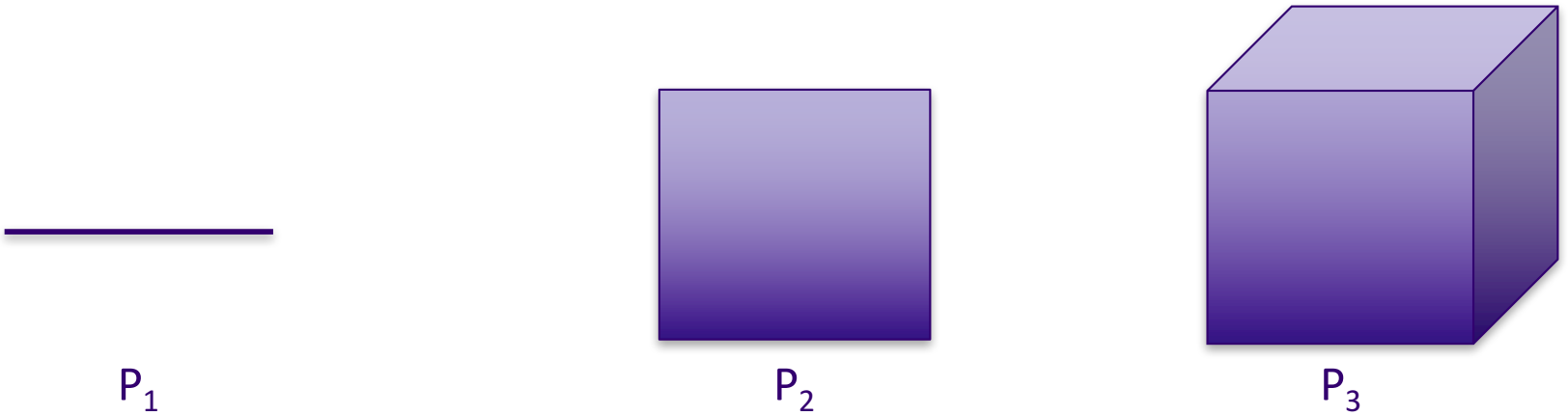
- The curse of high dimensionality and taking care with your training data
- Linear model selection / regularization
  - Subset selection
  - Ridge regression
  - LASSO regression
  - Python
- Dimensionality reduction approaches

# When good models go bad

- **Increasing** P **can be seductive, especially if you "feel" like you have a large data set:** In high dimension your training data may severely under sample the space of P
- **Supervised learning models are powerful:** If you have a lot of data you can often get a model that is predictive
  - **Can be difficult (or impossible) to make inference about important effects when** P **(the number of parameters) gets large..**

# The curse of dimensionality

- Consider a situation in which you need n=10 in order to cover response range (in Y) for each $X_i$.
  - For P=1, n=1… P=2, n=100 P=3, n=1000…




  - Our ability to capture significant fractions of the predictor space collapses after a few dimensions

# The curse of dimensionality

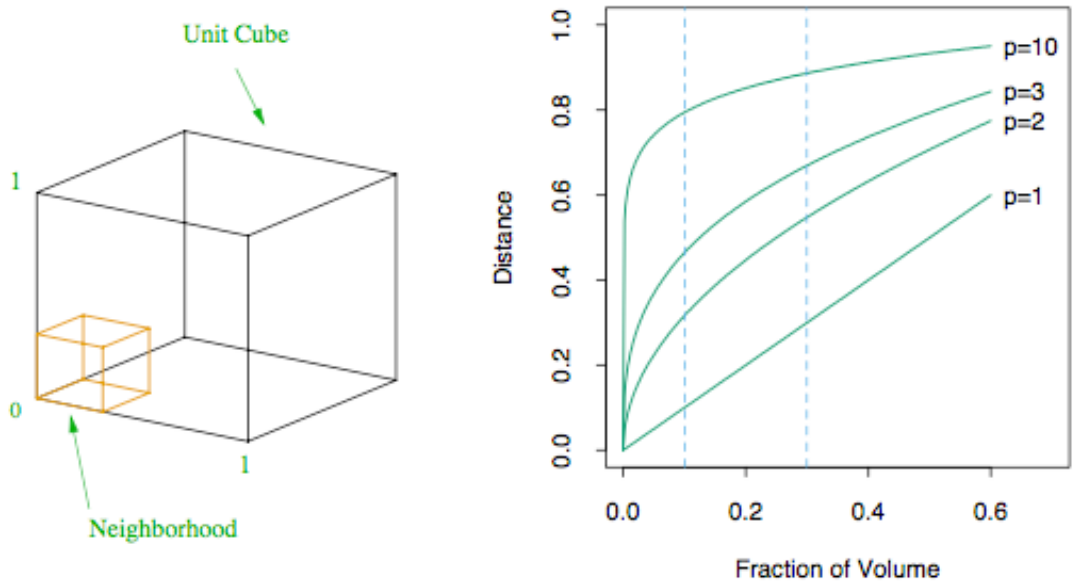



**FIGURE 2.6.** *The curse of dimensionality is well illustrated by a subcubical neighborhood for uniform data in a unit cube. The figure on the right shows the side-length of the subcube needed to capture a fraction r of the volume of the data, for different dimensions p. In ten dimensions we need to cover* 80% *of the range of each coordinate to capture* 10% *of the data.*

Figure 2.6, **Elements** of Statistical Learning

# Three approaches to building a smaller model (ISL, p 204)

- **Subset selection:** Systematically search different models created with subsets of all possible P values for an adequate model
- **Regularization (shrinkage):** Fit a model w/all P values, but use a different error penalty to force coefficients to be smaller
- **Dimensional reduction:** Fit a model with a reduced number of coarse parameters that represent subsets of groups of P's

# Subset selection: algorithms and challenges

- **Last week we discussed forward/backward selection algorithms as a method to determine which of the variables ($X_i$) are** important **in terms of describing the** variance in Y
  - **Subset selection algorithms are discussed in detail in section 6.1**
    - **Big picture takeaway: for $p$ features, there are $\binom{p}{k}$ different models of size $k$. This can be** computationally intractable
    - **Algorithms 6.2 and 6.3 describe the forward/backward selection algorithms**
- **How do you compare different algorithms trained with different numbers of parameters?**

# Comparing alternative models

- **Key concept:** The training error (**RSS**) will always decrease with increased number of parameters, how to evaluate different models with different number of parameters
- **We should therefore choose the best model considering the** testing error**. Two choices for this:**
  - **Use theoretical estimates of the test error. ISL offers four suggestions: Cp, AIC, BIC, adjusted $R^2$**
  - **Use bootstrapping or cross-validation to evaluate the test error directly**

# Theoretical models to estimate test error

- The models use the training RSS, number of training points ($n$), number of fit parameters ($d$) and estimated variance of response Y ($\widehat{\sigma^2}$)
- Cp: Baseline, adds penalty for more fit params

$$C_p = \frac{1}{n}\left(\mathrm{RSS} + 2d\hat{\sigma}^2\right), \qquad (6.2)$$

- AIC: Akaike information criterion; used for max likelihood methods: $\mathrm{AIC} = \frac{1}{n\hat{\sigma}^2}\left(\mathrm{RSS} + 2d\hat{\sigma}^2\right)$
- BIC: Bayesian information criterion $\mathrm{BIC} = \frac{1}{n}\left(\mathrm{RSS} + \log(n)d\hat{\sigma}^2\right)$
- Adjusted $R^2$: $R^2 = 1 - \frac{\mathrm{RSS}/(n-d-1)}{\mathrm{TSS}/(n-1)}$

# Comparison of test error estimates

- **This is a specific example from one of the ISL data sets!**
- **In your own model subset exercise it is likely that you would have** multiple models with same number of predictors!
- **Best practice is to monitor behavior of multiple test error estimates**

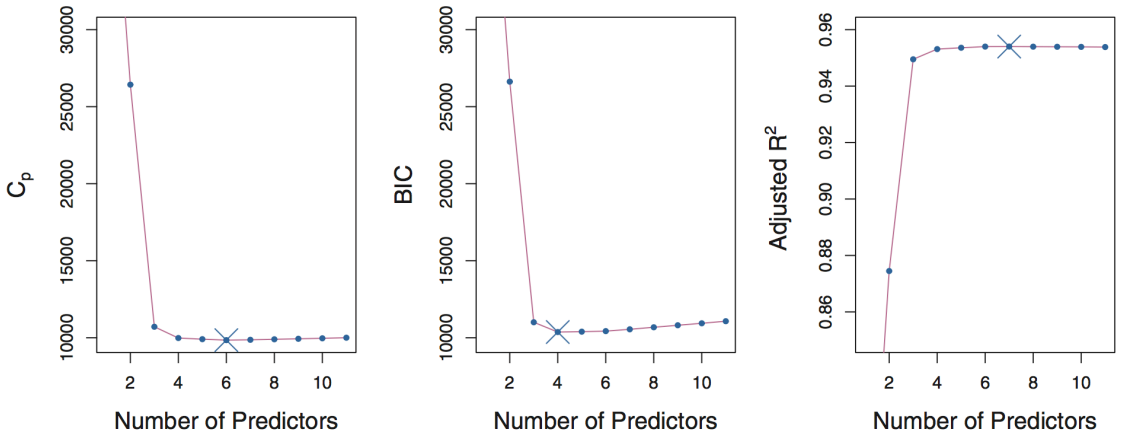



**FIGURE 6.2.** *$C_p$, BIC, and adjusted $R^2$ are shown for the best models of each size for the* `Credit` *data set (the lower frontier in Figure 6.1). $C_p$ and BIC are estimates of test MSE. In the middle plot we see that the BIC estimate of test error shows an increase after four variables are selected. The other two plots are rather flat after four variables are included.*

# Resampling and clarifying a bit of terminology…

- **The use of one large set of data combined with resampling yields the use of "training and validation" error estimates (in most statistical learning approaches)**
- **Problem: In cross-validation or bootstrap, the learning algorithm "sees" all the data eventually**
- **If possible, we often withhold a 3rd data set "the test set" that the model is totally blind to:** testing error data is withheld until the final model is selected
  - **Training data vs…**
  - **Validation data vs…**
  - **Testing data**

# Resampling and testing error estimates

- If the number of candidate models you have to run is "small" (*compared to your total science workflow time*), then you should use bootstrap or cross-validation
- Especially true in the absence of high quality estimated variance of response Y ($\widehat{\sigma^2}$)
- Modern computers (even your laptop) can train huge numbers of models on a short amount of time

# Regularization: big picture concepts

- **Instead of subset selection, another option is to fit a model with all possible (P) parameters and add a penalty to the RSS term to** shrink **them**

- **Why do we want to do this? [THIS IS IMPORTANT]**
  - **Recall that models with more parameters will better estimate the training set, reducing the training error**
  - **However, adding more parameters increases the testing error (**the variance of response Y is increased via the bias-variance tradeoff**)**
  - **Reducing the magnitude of the coefficients, therefore is a plausible route to reducing test set error**

# Ridge regression

$$\text{RSS} = \sum_{i=1}^{n} \left( y_i - \beta_0 - \sum_{j=1}^{p} \beta_j x_{ij} \right)^2$$

- **Ridge replaces the RSS term: with a new minimizer that includes a so-called** shrinkage penalty**:**

$$\sum_{i=1}^{n} \left( y_i - \beta_0 - \sum_{j=1}^{p} \beta_j x_{ij} \right)^2 + \lambda \sum_{j=1}^{p} \beta_j^2 = \text{RSS} + \lambda \sum_{j=1}^{p} \beta_j^2, \qquad (6.5)$$

- **The adjustable parameter λ, trades the baseline RSS with a penalty for nonzero coefficients. As λ increases to infinity the minimized error drives all of the coefficients to zero**

# Ridge in practice

1. **Unlike RSS minimizers, which are** scale equivariant**, the response data must be normalized in regularization methods:**

$$\tilde{x}_{ij} = \frac{x_{ij}}{\sqrt{\frac{1}{n}\sum_{i=1}^{n}(x_{ij} - \overline{x}_j)^2}}$$

2. **Fit a series of ridge regression models across a wide range of $\lambda$ and track the coefficient values and test set error (or estimate) as $\lambda$ is changed**
3. **Determine the model that produces the smallest test error set**
   - **Alternatively determine the model w/smallest** validation **error and then estimate the true test error w/virgin data that was not used in the training**

# Expected response in ridge regression

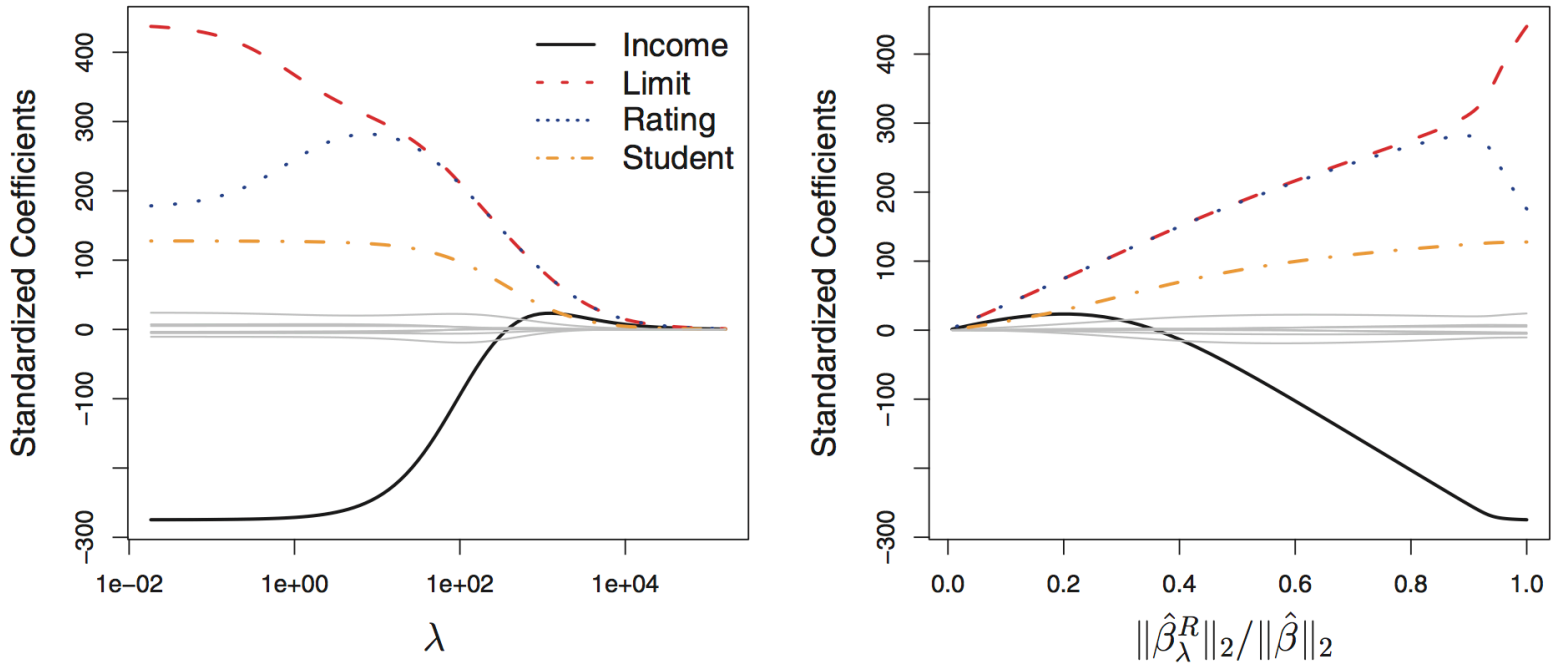



**FIGURE 6.4.** *The standardized ridge regression coefficients are displayed for the* `Credit` *data set, as a function of* $\lambda$ *and* $\|\hat{\beta}^R_\lambda\|_2/\|\hat{\beta}\|_2$.

# Expected response in ridge regression

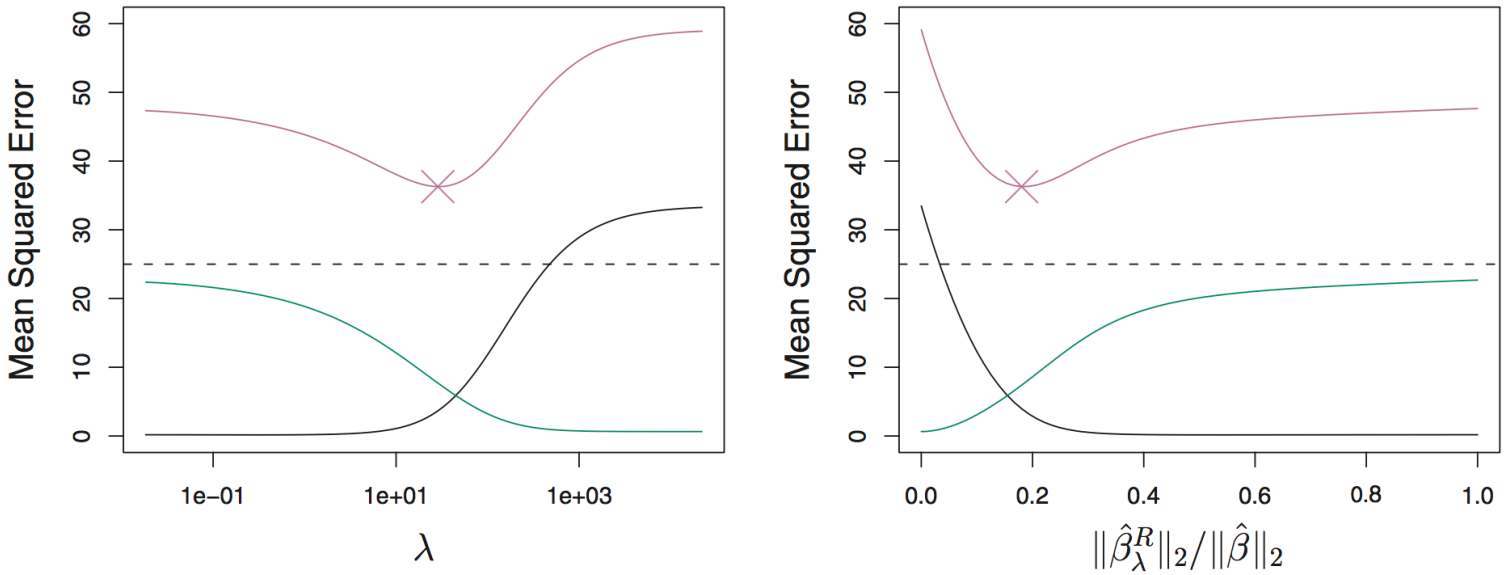



**FIGURE 6.5.** *Squared bias (black), variance (green), and test mean squared error (purple) for the ridge regression predictions on a simulated data set, as a function of* $\lambda$ *and* $\|\hat{\beta}^R_\lambda\|_2/\|\hat{\beta}\|_2$. *The horizontal dashed lines indicate the minimum possible MSE. The purple crosses indicate the ridge regression models for which the MSE is smallest.*

# LASSO regression

- Ridge regression does not set any of the coefficients exactly to zero but can shrink all of them
- The LASSO regression was developed, inspired by ridge, to provide the possibility that some of the coefficients can take a value of zero
- Like ridge, the LASSO operator is minimized as a function of an adjustable $\lambda$ parameter

$$\sum_{i=1}^{n}\left(y_i-\beta_0-\sum_{j=1}^{p}\beta_j x_{ij}\right)^2+\lambda\sum_{j=1}^{p}|\beta_j| \;=\mathrm{RSS}+\lambda\sum_{j=1}^{p}|\beta_j|. \qquad (6.7)$$

# An extra bonus in LASSO: subset selection

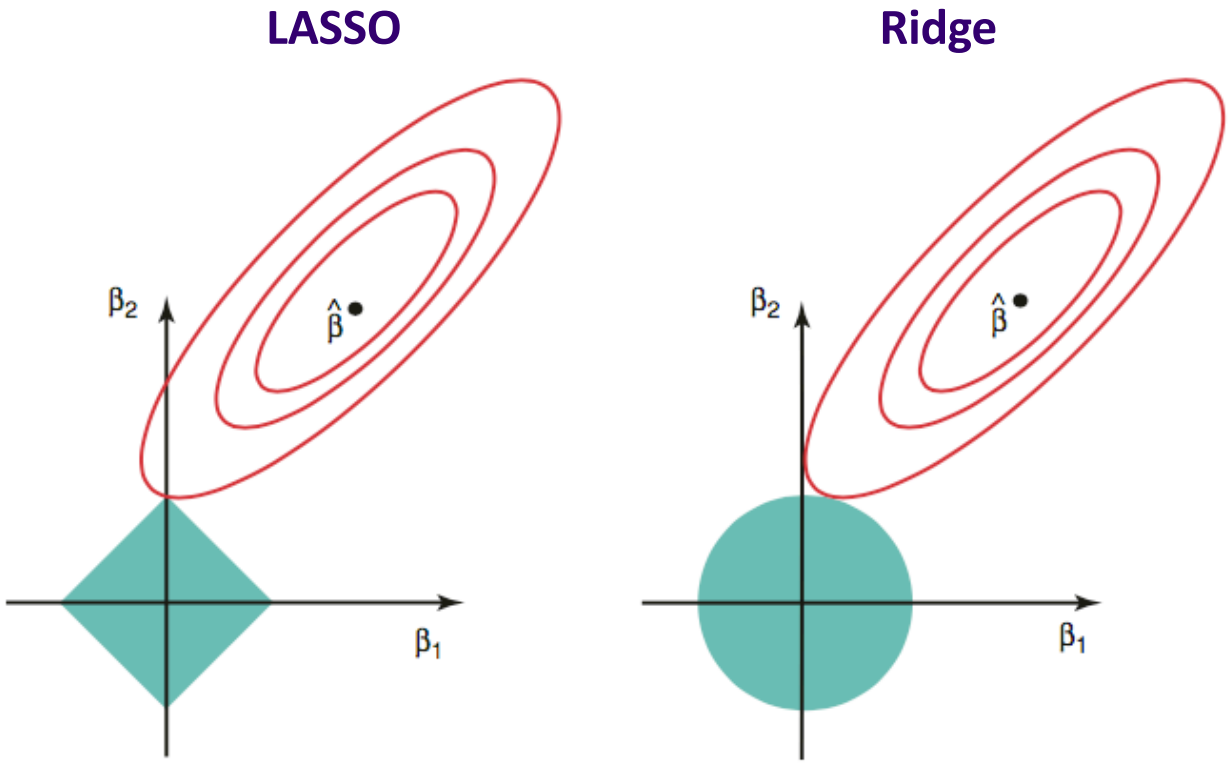
- **The key difference is the penalty due to nonzero coefficients. Ridge it is squared, in LASSO it is not. You can also formulate both as** constrained **minimization problems.**
- **A mathematical result of LASSO (6.8), is the possibility that some of the β values will be zero at the minimum error**

$$\underset{\beta}{\text{minimize}} \left\{ \sum_{i=1}^{n} \left( y_i - \beta_0 - \sum_{j=1}^{p} \beta_j x_{ij} \right)^2 \right\} \quad \text{subject to} \quad \sum_{j=1}^{p} |\beta_j| \leq s \tag{6.8}$$

and

$$\underset{\beta}{\text{minimize}} \left\{ \sum_{i=1}^{n} \left( y_i - \beta_0 - \sum_{j=1}^{p} \beta_j x_{ij} \right)^2 \right\} \quad \text{subject to} \quad \sum_{j=1}^{p} \beta_j^2 \leq s, \tag{6.9}$$

# Why can LASSO coefficients become zero?




**Red** = RSS

**Blue** = constraint zone

**FIGURE 6.7.** *Contours of the error and constraint functions for the lasso* (left) *and ridge regression* (right). *The solid blue areas are the constraint regions,* $|\beta_1| + |\beta_2| \leq s$ *and* $\beta_1^2 + \beta_2^2 \leq s$, *while the red ellipses are the contours of the RSS.*

# Python practice

- We will use a data set from UCI Irvine ML database related to insulation and energy efficiency
- There are 8 predictors (X1-8) and 2 responses (Y1-Y2)
- I have a notebook setup for ML regression, ridge and lasso.  Some missing pieces and suggestions.
- Find a friend and dig in!

# Wrap up

- **Bootstrapping and cross-validation are near-universal in their applicability…**
- **Must understand different methods for dealing with large P models!**
  - **Subset selection and regularization (shrinkage)**
  - **Also can use** dimensionality reduction **(see end of CH6, ISL)**
  - **These same concepts apply to many types of nonlinear models**
- **What we didn't cover in this module of the class:**
  - **Dimensionality reduction methods (CH6)**
  - **Nonlinear regression w/polynomials and splines (CH7)**
- **Monday: CH8 , tree methods**