# Ch 6.1: Subset Selection

## Lecture 11 - CMSE 381

Michigan State University
::
Dept of Computational Mathematics, Science & Engineering

Feb 21, 2024

# Announcements

**Last time**

- Boostrapping

**Covered in this lecture**

- Subset selection
- Forward and Backward Selection
- Ridge regression

**Announcements:**

- HW #6 posted and due next Wednesday
- Spring break next week

Section 1

## Last time

# Goals of fitting a given model

Up to now, we've focused on standard linear model: $Y = \beta_0 + \beta_1 X_1 + \cdots + \beta_p X_p + \varepsilon$ and done least squares estimation. *Translation: Minimize RSS* $= \sum_i (y_i - \hat{y}_i)^2$

**Prediction accuracy**

- If the true relationship is approximately linear, least squares estimates have low bias
- If $n \gg p$, then least squares also has low variance
- If $n$ not much larger than $p$, then high variability, overfitting, poor predictions
- If $n < p$ then no unique solution so can't use at all
- Goal on next slide: shrink/constrain number of variables to improve accuracy

# Goals of fitting a given model

Up to now, we've focused on standard linear model: $Y = \beta_0 + \beta_1 X_1 + \cdots + \beta_p X_p + \varepsilon$ and done least squares estimation. *Translation: Minimize RSS* $= \sum_i (y_i - \hat{y}_i)^2$

**Model Interpretability**

- Often, many variables included aren't associated with response
- Including them leads to unnecessary complexity in the model
- Idea: Set these coefficients to 0 (or close to zero)
- Goal on next slide: Do some automatic feature selection / variable selection to get rid of unnecessary variables

# Goal of next chapter

*Classes of methods*

- Subset selection: identify subset of predictors, fit model using least squares on smaller set of variables
- Shrinkage:
  - Estimated coeff are shrunken towards zero relative to the least squares estimate
- Dimension reduction: Project the $p$-dimensional data into $M$-dimensional subspace, $M < p$.

pick $X_i$ to use before doing the regression

first do regression then pick $X_i$

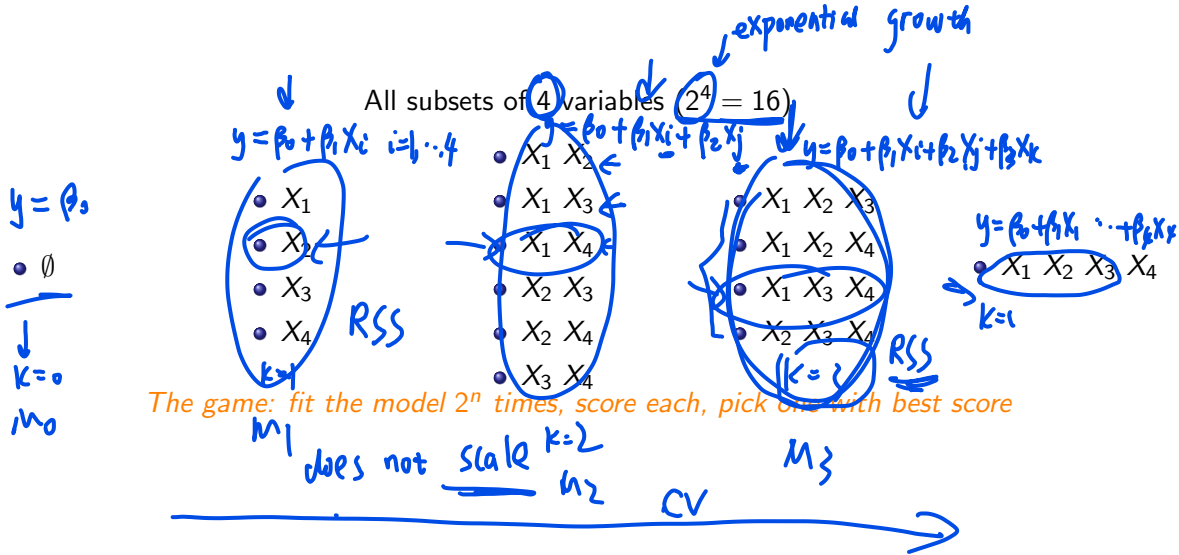
PCA (others) + regression
↓
pick $X_i$

Section 2

# Best Subset Selection

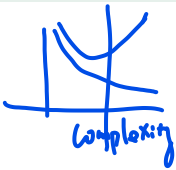
# Brute-force, but too slow...

All subsets of 4 variables ($2^4 = 16$)




*The game: fit the model $2^n$ times, score each, pick one with best score*

# One way of breaking this up




**Algorithm 6.1** *Best subset selection*

1. Let $\mathcal{M}_0$ denote the *null model*, which contains no predictors. This model simply predicts the sample mean for each observation.
2. For $k = 1, 2, \ldots p$:
   (a) Fit all $\binom{p}{k}$ models that contain exactly $k$ predictors.
   (b) Pick the best among these $\binom{p}{k}$ models, and call it $\mathcal{M}_k$. Here *best* is defined as having the smallest RSS, or equivalently largest $R^2$.
3. Select a single best model from among $\mathcal{M}_0, \ldots, \mathcal{M}_p$ using cross-validated prediction error, $C_p$ (AIC), BIC, or adjusted $R^2$

- Part 2b goes for lowest training score, Part 3 then goes for lowest testing score.
- Step 2 is computational infeasable for large $p$

# Group work: calculate by hand

We train a model using four variables, $X_1, X_2, X_3, X_4$. We're interested in getting a subset of the variables to use. The following table shows the mean squared error and the $R^2$ value computed for the model learned using each possible subset of variables.

| | Training MSE (x10^7) | k-fold CV Testing Error |
|---|---|---|
| Null model | 8.76 | 10.08 |
| X1 | 8.63 | 9.98 |
| X2 | 7.42 | 8.01 |
| X3 | 8.16 | 8.3 |
| X4 | 8.33 | 9.06 |
| X1,X2 | 4.33 | 7.47 |
| X1,X3 | 5.82 | 5.22 |
| X1,X4 | 3.17 | 4.23 |
| X2,X3 | 4.07 | 3.78 |
| X2,X4 | 3.31 | 4.01 |
| X3,X4 | 3.06 | 4.16 |
| X1,X2,X3 | 3.08 | 5.49 |
| X1,X2,X4 | 3.55 | 4.02 |
| X1,X3,X4 | 2.97 | 4.23 |
| X2,X3,X4 | 2.98 | 3.17 |
| X1,X2,X3,X4 | 2.16 | 4.39 |

1. What subset of variables is found for each of the sets $\mathcal{M}_0, \mathcal{M}_1, \mathcal{M}_2, \mathcal{M}_3, \mathcal{M}_4$ when using best subset selection?
2. What subset of variables is returned using best subset selection?

# Extra work space

| | Training MSE (x10^7) | k-fold CV Testing Error |
|---|---|---|
| Null model | 8.76 | 10.08 |
| X1 | 8.63 | 9.98 |
| X2 | 7.42 | 8.01 |
| X3 | 8.16 | 8.3 |
| X4 | 8.33 | 9.06 |
| X1,X2 | 4.33 | 7.47 |
| X1,X3 | 5.82 | 5.22 |
| X1,X4 | 3.17 | 4.23 |
| X2,X3 | 4.07 | 3.78 |
| X2,X4 | 3.31 | 4.01 |
| X3,X4 | 3.06 | 4.16 |
| X1,X2,X3 | 3.06 | 5.49 |
| X1,X2,X4 | 3.55 | 4.02 |
| X1,X3,X4 | 2.97 | 4.3 |
| X2,X3,X4 | 2.98 | 3.17 |
| X1,X2,X3,X4 | 2.16 | 5 |

- $\emptyset$
- $X_1$
- $X_2$
- $X_3$
- $X_4$
- $X_1\ X_2$
- $X_1\ X_3$
- $X_1\ X_4$
- $X_2\ X_3$
- $X_2\ X_4$
- $X_3\ X_4$
- $X_1\ X_2\ X_3$
- $X_1\ X_2\ X_4$
- $X_1\ X_3\ X_4$
- $X_2\ X_3\ X_4$
- $X_1\ X_2\ X_3\ X_4$

{X3, X4}

Section 3

# Forward Selection

# What's the problem?

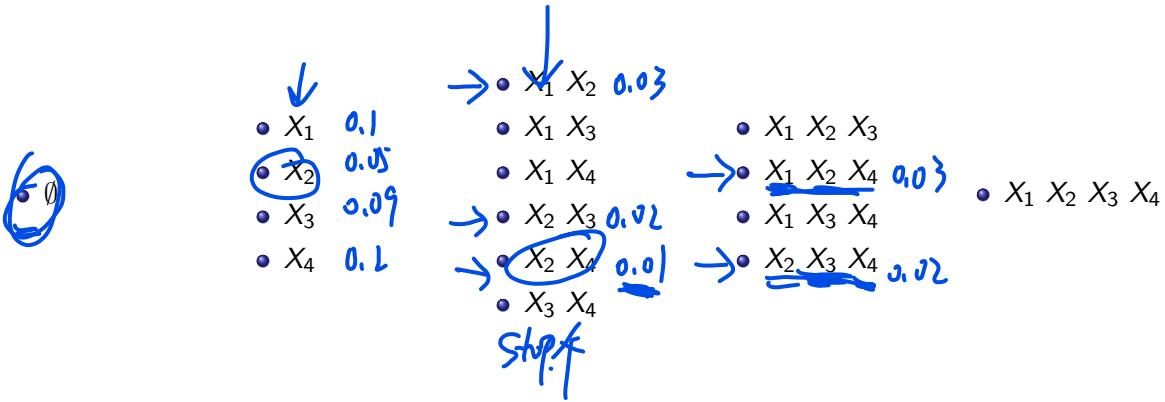
- Checking $2^p$ models is not reasonable for large $p$, $p > 40$
- The next bits are finding alternatives to Step 2

# Forward Stepwise Selection

**Algorithm 6.2** *Forward stepwise selection*

1. Let $\mathcal{M}_0$ denote the *null* model, which contains no predictors.
2. For $k = 0, \ldots, p-1$:
   (a) Consider all $p - k$ models that augment the predictors in $\mathcal{M}_k$ with one additional predictor.
   (b) Choose the *best* among these $p - k$ models, and call it $\mathcal{M}_{k+1}$. Here *best* is defined as having smallest RSS or highest $R^2$.
3. Select a single best model from among $\mathcal{M}_0, \ldots, \mathcal{M}_p$ using cross-validated prediction error, $C_p$ (AIC), BIC, or adjusted $R^2$.

# An example for Forward Stepwise Selection



# Group work: by hand same example with forward example

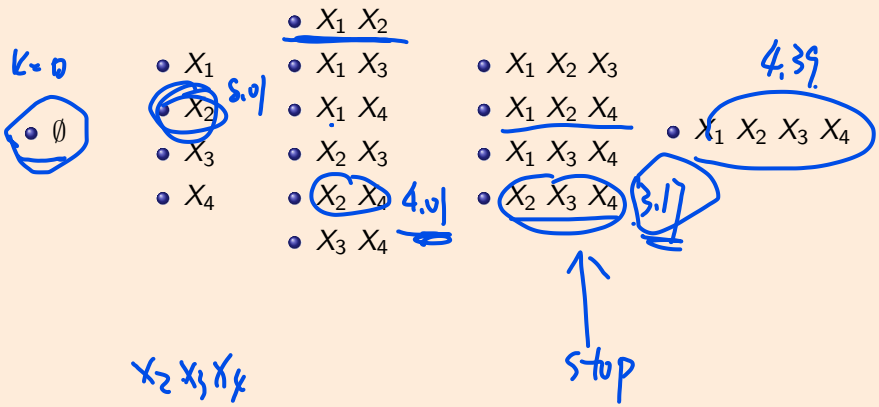
We train a model using four variables, $X_1, X_2, X_3, X_4$. We're interested in getting a subset of the variables to use. The following table shows the mean squared error and the $R^2$ value computed for the model learned using each possible subset of variables.

| | Training MSE (x10^7) | k-fold CV Testing Error |
|---|---|---|
| Null model | 8.76 | 10.08 |
| X1 | 8.63 | 9.98 |
| X2 | 7.42 | 8.01 |
| X3 | 8.16 | 8.3 |
| X4 | 8.33 | 9.06 |
| X1,X2 | 4.33 | 7.47 |
| X1,X3 | 5.82 | 5.22 |
| X1,X4 | 3.17 | 4.23 |
| X2,X3 | 4.07 | 3.78 |
| X2,X4 | 3.31 | 4.01 |
| X3,X4 | 3.06 | 4.16 |
| X1,X2,X3 | 3.08 | 5.49 |
| X1,X2,X4 | 3.55 | 4.02 |
| X1,X3,X4 | 2.97 | 4.23 |
| X2,X3,X4 | 2.98 | 3.17 |
| X1,X2,X3,X4 | 2.16 | 4.39 |

1. What subset of variables is found for each of the sets $\mathcal{M}_0, \mathcal{M}_1, \mathcal{M}_2, \mathcal{M}_3, \mathcal{M}_4$ when using forward selection?
2. What subset of variables is returned using forward subset selection?

# Extra work space if it helps

| | Training MSE (x10^7) | k-fold CV Testing Error |
|---|---|---|
| Null model | 8.76 | 10.08 |
| X1 | 8.63 | 9.98 |
| X2 | 7.42 | 8.01 |
| X3 | 8.16 | 8.3 |
| X4 | 8.33 | 9.06 |
| X1,X2 | 4.33 | 7.47 |
| X1,X3 | 5.82 | 5.22 |
| X1,X4 | 3.17 | 4.23 |
| X2,X3 | 4.07 | 3.78 |
| X2,X4 | 3.31 | 4.01 |
| X3,X4 | 3.06 | 4.16 |
| X1,X2,X3 | 3.08 | 5.49 |
| X1,X2,X4 | 3.55 | 4.02 |
| X1,X3,X4 | 2.97 | 4.23 |
| X2,X3,X4 | 2.98 | 3.17 |
| X1,X2,X3,X4 | 2.16 | 4.39 |



# Pros and Cons of Forward Stepwise

**Pros:**

- Computationally cheaper
- Number of models fit is

$$1 + \sum_{k=0}^{p-1}(p-k) = 1 + \frac{p(p+1)}{2}$$

which is way better than $2^p$

**Cons:**

- Not guaranteed to find the best model
- As example: if best 1-variable model is $X_1$, but best 2-variable model is $X_2X_3$, then forward selection won't find it.
- Is this a con? Maybe just a limitation If $n < p$, then can only construct models $\mathcal{M}_0 \cdots \mathcal{M}_{n-1}$

Section 4

# Backward Selection

# Backward stepwise selection

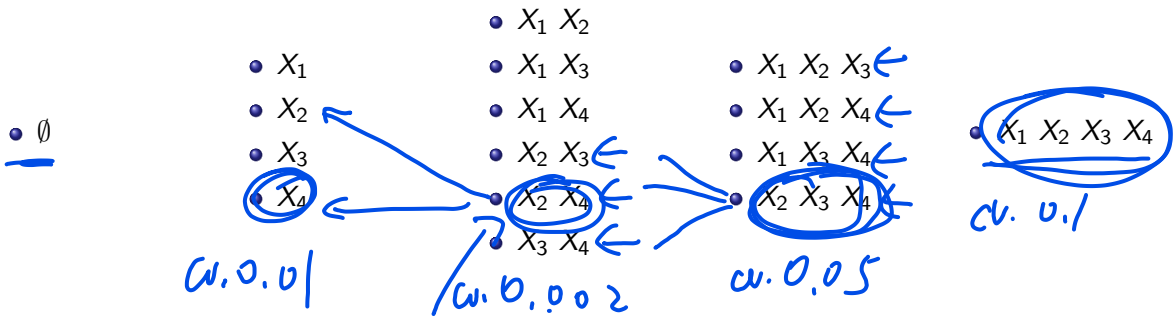
**Algorithm 6.3** *Backward stepwise selection*

1. Let $\mathcal{M}_p$ denote the *full* model, which contains all $p$ predictors.
2. For $k = p, p-1, \ldots, 1$:
   (a) Consider all $k$ models that contain all but one of the predictors in $\mathcal{M}_k$, for a total of $k-1$ predictors.
   (b) Choose the *best* among these $k$ models, and call it $\mathcal{M}_{k-1}$. Here *best* is defined as having smallest RSS or highest $R^2$.
3. Select a single best model from among $\mathcal{M}_0, \ldots, \mathcal{M}_p$ using cross-validated prediction error, $C_p$ (AIC), BIC, or adjusted $R^2$.

computationally heavy

historical

# An example for Backward Stepwise Selection



# Group work: by hand same example with backward

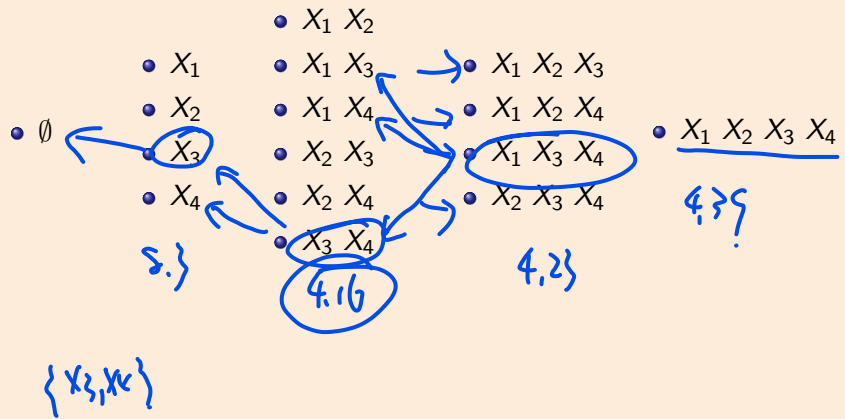
We train a model using four variables, $X_1, X_2, X_3, X_4$. We're interested in getting a subset of the variables to use. The following table shows the mean squared error and the $R^2$ value computed for the model learned using each possible subset of variables.

| | Training MSE (x10^7) | k-fold CV Testing Error |
|---|---|---|
| Null model | 8.76 | 10.08 |
| X1 | 8.63 | 9.98 |
| X2 | 7.42 | 8.01 |
| X3 | 8.16 | 8.3 |
| X4 | 8.33 | 9.06 |
| X1,X2 | 4.33 | 7.47 |
| X1,X3 | 5.82 | 5.22 |
| X1,X4 | 3.17 | 4.23 |
| X2,X3 | 4.07 | 3.78 |
| X2,X4 | 3.31 | 4.01 |
| X3,X4 | 3.06 | 4.16 |
| X1,X2,X3 | 3.08 | 5.49 |
| X1,X2,X4 | 3.55 | 4.02 |
| X1,X3,X4 | 2.97 | 4.23 |
| X2,X3,X4 | 2.98 | 3.17 |
| X1,X2,X3,X4 | 2.16 | 4.39 |

1. What subset of variables is found for each of the sets $\mathcal{M}_0, \mathcal{M}_1, \mathcal{M}_2, \mathcal{M}_3, \mathcal{M}_4$ when using forward selection?
2. What subset of variables is returned using forward subset selection?

# Extra work space

| | Training MSE (x10^7) | k-fold CV Testing Error |
|---|---|---|
| Null model | 8.76 | 10.08 |
| X1 | 8.63 | 9.98 |
| X2 | 7.42 | 8.01 |
| X3 | 8.16 | 8.3 |
| X4 | 8.33 | 9.06 |
| X1,X2 | 4.33 | 7.47 |
| X1,X3 | 6.82 | 5.22 |
| X1,X4 | 8.17 | 4.23 |
| X2,X3 | 4.07 | 3.78 |
| X2,X4 | 3.31 | 4.01 |
| X3,X4 | 3.06 | 4.16 |
| X1,X2,X3 | 3.08 | 5.49 |
| X1,X2,X4 | 3.55 | 4.02 |
| X1,X3,X4 | 2.97 | 4.23 |
| X2,X3,X4 | 2.98 | 3.17 |
| X1,X2,X3,X4 | 2.16 | 4.39 |



# Pros and Cons of Backward Stepwise

**Pros:**

- Computationally cheaper
- Number of models fit is still

$$1 + \sum_{k=1}^{p} k = 1 + \frac{p(p+1)}{2}$$

which is way better than $2^p$

**Cons:**

- Not guaranteed to find the best model
- Unlike forward selection, this can't be used at all if $n < p$

Section 5

# Alternatives for Approximating Test Error

# Remembering what we're doing

**Algorithm 6.1** *Best subset selection*

1. Let $\mathcal{M}_0$ denote the *null model*, which contains no predictors. This model simply predicts the sample mean for each observation.
2. For $k = 1, 2, \ldots p$:
   (a) Fit all $\binom{p}{k}$ models that contain exactly $k$ predictors.
   (b) Pick the best among these $\binom{p}{k}$ models, and call it $\mathcal{M}_k$. Here *best* is defined as having the smallest RSS, or equivalently largest $R^2$.
3. Select a single best model from among $\mathcal{M}_0, \ldots, \mathcal{M}_p$ using cross-validated prediction error, $C_p$ (AIC), BIC, or adjusted $R^2$.

*Now we're focusing on step 3*

**Algorithm 6.2** *Forward stepwise selection*

1. Let $\mathcal{M}_0$ denote the *null* model, which contains no predictors.
2. For $k = 0, \ldots, p-1$:
   (a) Consider all $p-k$ models that augment the predictors in $\mathcal{M}_k$ with one additional predictor.
   (b) Choose the *best* among these $p-k$ models, and call it $\mathcal{M}_{k+1}$. Here *best* is defined as having smallest RSS or highest $R^2$.
3. Select a single best model from among $\mathcal{M}_0, \ldots, \mathcal{M}_p$ using cross-validated prediction error, $C_p$ (AIC), BIC, or adjusted $R^2$.

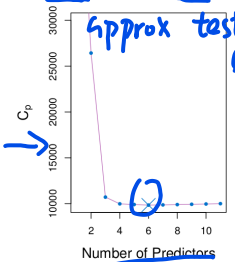
**Algorithm 6.3** *Backward stepwise selection*

1. Let $\mathcal{M}_p$ denote the *full* model, which contains all $p$ predictors.
2. For $k = p, p-1, \ldots, 1$:
   (a) Consider all $k$ models that contain all but one of the predictors in $\mathcal{M}_k$, for a total of $k-1$ predictors.
   (b) Choose the *best* among these $k$ models, and call it $\mathcal{M}_{k-1}$. Here *best* is defined as having smallest RSS or highest $R^2$.
3. Select a single best model from among $\mathcal{M}_0, \ldots, \mathcal{M}_p$ using cross-validated prediction error, $C_p$ (AIC), BIC, or adjusted $R^2$.

*The goal is to come up with ways to adjust the training scores to get something that better approximates testing scores*

# The $C_p$ estimate

$$C_p = \frac{1}{n}(\text{RSS} + 2d\hat{\sigma}^2)$$



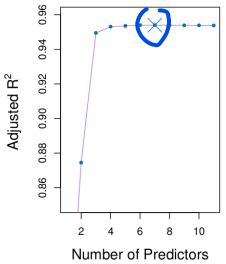

Example using `Credit`

- The $p$ does nothing. It's not our usual $p$. WTF
- $d$ is the number of predictors you're using to fit
- $\hat{\sigma}^2$ is an estimate of $\text{Var}(\varepsilon)$

- Penalty increases with more $d$, so aims for fewer predictors
- This acts to adjust for the overfitting decrease in RSS from higher $d$
- One can show that if $\hat{\sigma}^2$ is an unbiased estimate of $\sigma^2$, then $C_p$ is an unbiased estimate of test MSE.
- So.... aim for model with lowest $C_p$
- This example takes a 6 variable model

# Adjusted $R^2$

$$R^2 = 1 - \frac{\text{RSS}}{\text{TSS}}$$




- $TSS = \sum(y_i - \bar{y})^2$ total sum of squares
- A large value means small test error, so we want the $R^2$ to go up
- $RSS$ always decreases as number of variables increases, but $RSS/(n-d-1)$ could go up or down.

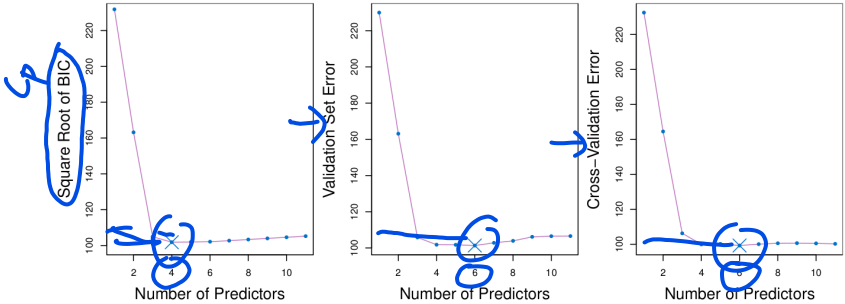
$$\text{adjusted } R^2 = 1 - \frac{\text{RSS}/(n-d-1)}{\text{TSS}/(n-1)}$$

- Idea is that including additional variables to the model that are noise leads to small decrease in RSS.
- In that case, $d$ always up by 1, so $\frac{RSS}{n-d-1}$ will increase, causing $R^2$ to decrease
- View this as penalizing adding unnecessary variables

# Comparisons

- $C_p$ has rigorous justifications that we're gonna skip
- $R^2$ is intuitive, but doesn't really come with theoretical justification
- These equations presented are in the case of least squares fit.

# All this vs. Validation and Cross Validation




- The goal was to approximate test error, so why not do validation or CV instead?
- Historically, CV was computationally prohibitive. These tools were an alternative to deal with the same question but with much cheaper computation

Section 6

# Ridge Regression

# Goal

- Fit model using all $p$ predictors
- Aim to constrain (regularize) coefficient estimates
- Shrink the coefficient estimates towards 0

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \beta_3 X_3 + \beta_4 X_4$$

- Ridge regression
- Lasso

Test error = train error + risk of fitting

complexity

# Ridge regression

**Before:**

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

**After:**

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

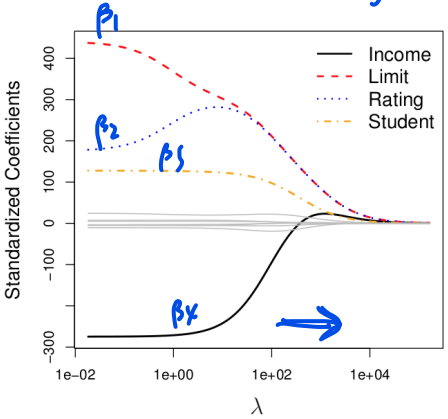
- $\lambda \geq 0$ is a tuning paramter to figure out separately.
- Second term is called "shrinkage penalty"
- $\lambda = 0$ means least squares estimate
- As $\lambda \to \infty$ impact of shrinkage penalty grows, ridge regression coeff will go towards 0
- Not applied to the intercept
- call coefficients found by ridge regression $\hat{\beta}_\lambda^R = (\beta_{1,\lambda}^R, \cdots, \beta_{p,\lambda}^R \cdots)$

$\lambda$ large $\Rightarrow \beta_j \approx 0$

small $\Rightarrow$ RSS $\approx 0$

# Example from the `Credit` data

$$RSS + \lambda \sum_{j=1}^{p} \beta_j^2$$

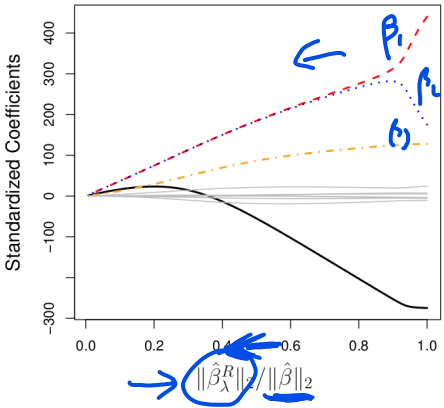



- Left x-axis, $\lambda \approx 0$
- Each line is the predicted coefficient value for a single variable
- as $\lambda$ increases everything goes to 0, basically giving the null model at the right end

# Same Setting, Different Plot

$$\hat{\beta}_\lambda = \min_{\beta_j} RSS + \lambda \sum_{j=1}^{p} \beta_j^2 \qquad \|\beta\|_2 = \sqrt{\sum_{j=1}^{p} \beta_j^2}$$

$\lambda \uparrow \Rightarrow \beta_j \downarrow \Rightarrow \|\beta\|_2 \downarrow$




- $\hat{\beta}$ is the vector of least squares coeff estimates
- $\|\cdot\|$ measures distance from 0
- $\|\hat{\beta}_\lambda^R\|/\|\hat{\beta}\|$ ranges from 1 (when $\lambda = 0$) down to 0
- Amount the coefficient estimates have been shrunk

# Scale equivavariance (or lack thereof)

**Scale equivariant:** Multiplying a variable by $c$ ($cX_i$) just returns a coefficient multiplied by $1/c$ ($1/c\beta_i$)

- Ex: `income variable`.
- Least squares is scale equivariant
- Ridge regression very much is not
- $X_j\hat{\beta}^R_{j,\lambda}$ depends not only on $\lambda$ but also on values of other predictors

in linear reg

$y = \beta_0 + \beta_1 X_1 + \beta_2 X_2$

$y = \beta_0 + \frac{\beta_1}{c} \cdot (cX_1) + \beta_2 X_2$

$\hat{\beta}_1$ $\quad$ $\hat{X}_1$

$X_1$: time in min

time in sec

contribution of $\beta_1 X_1$ stays the same.

in ridge $\sum (y - \beta_0 - \beta_1 X_1 - \beta_2 X_2)^2 + \lambda\left(\frac{\beta_1^2}{c^2} + \beta_2^2\right)$

$\lambda(c^2\hat{\beta}_1^2 + \hat{\beta}_2^2)$

# Solution: Standardize predictors

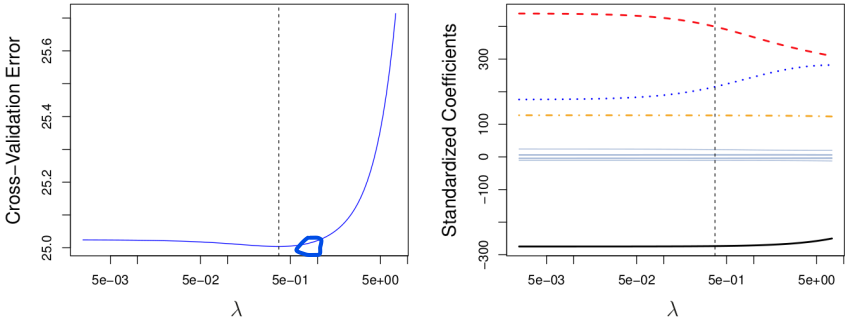
$$\widetilde{x}_{ij} = \frac{x_{ij}}{\sqrt{\frac{1}{n}\sum_{i=1}^{n}(x_{ij} - \overline{x}_j)^2}}$$

- Denominator is estimated standard deviation of the jth predictor
- Standardized predictors will all have a standard deviation of one
- Previous figures showe standardized ridge regression coeffs on the $y$-axis

# Using Cross-Validation to find $\lambda$

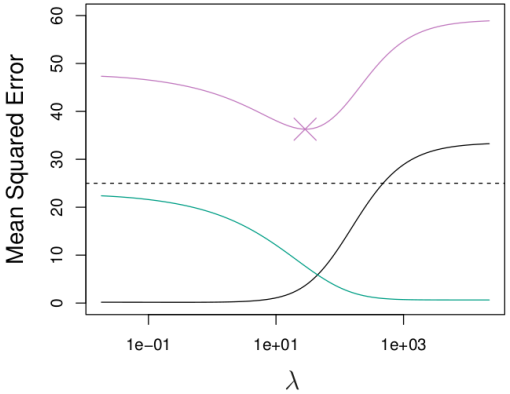
- Choose a grid of $\lambda$ values
- Compute the ($k$-fold) cross-validation error for each value of $\lambda$
- Select the tuning parameter value $\lambda$ for which the CV error is smallest.
- The model is re-fit using all of the available observations and the selected value of the tuning parameter.

# LOOCV choice of $\lambda$ for ridge regression and `Credit` data




- Dashed vertical lines indicate the selected value of $\lambda$.
- In this case, small $\lambda$ so optimal fit involves small amount of shrinkage relative to least squares
- The dip is not very pronounced, so there is rather a wide range of values that would give a very similar error.
- In a case like this we might simply use the least squares solution.

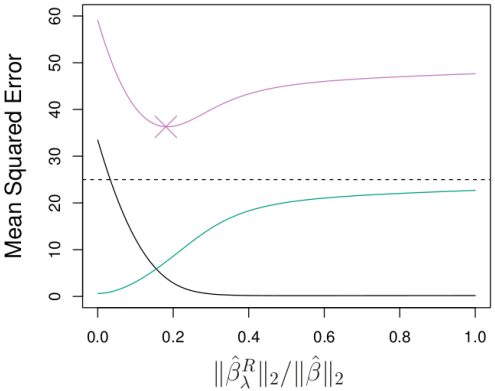
# Bias-Variance tradeoff




Squared bias (black), variance (green), and test mean squared error (purple) for simulated data. *Horizontal dashed line is minimum possible test MSE*

- Bias-variance tradeoff
- Minimum MSE achieved around $\lambda = 30$
- High variance means MSE for $\lambda = 0$ (least squares) and $\lambda = \infty$ (null model) are basically the same

# More Bias-Variance Tradeoff




Squared bias (black), variance (green), and test mean squared error (purple) for simulated data.

# Advantages of Ridge

**Ridge vs. Least Squares:**

- Previous slide and ability to lower variance
- Ridge regression works best in situations where the least squares estimates have high variance
- Can trade off small increase in bias for a large decrease in variance

**Ridge vs. Subset Selection:**

- So much better computationally!
- Only fits a single model rather than $2^p$
- Tricks available to find the optimal $\lambda$ without CV