Ch.6 Model Selection (How to set up "flexibility")

web.stanford.edu/class/stats202

Sergio Bacallado, Jonathan Taylor

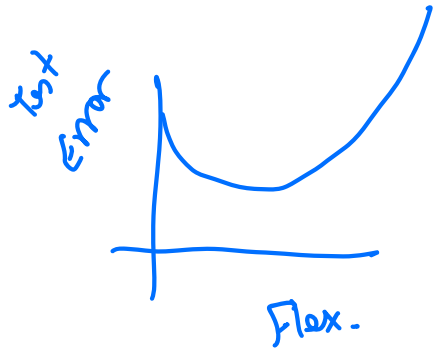
Autumn 2022

$Y \sim X_1 + X_2 + \cdots + X_p$

$E \subset \{1, \ldots, p\}$

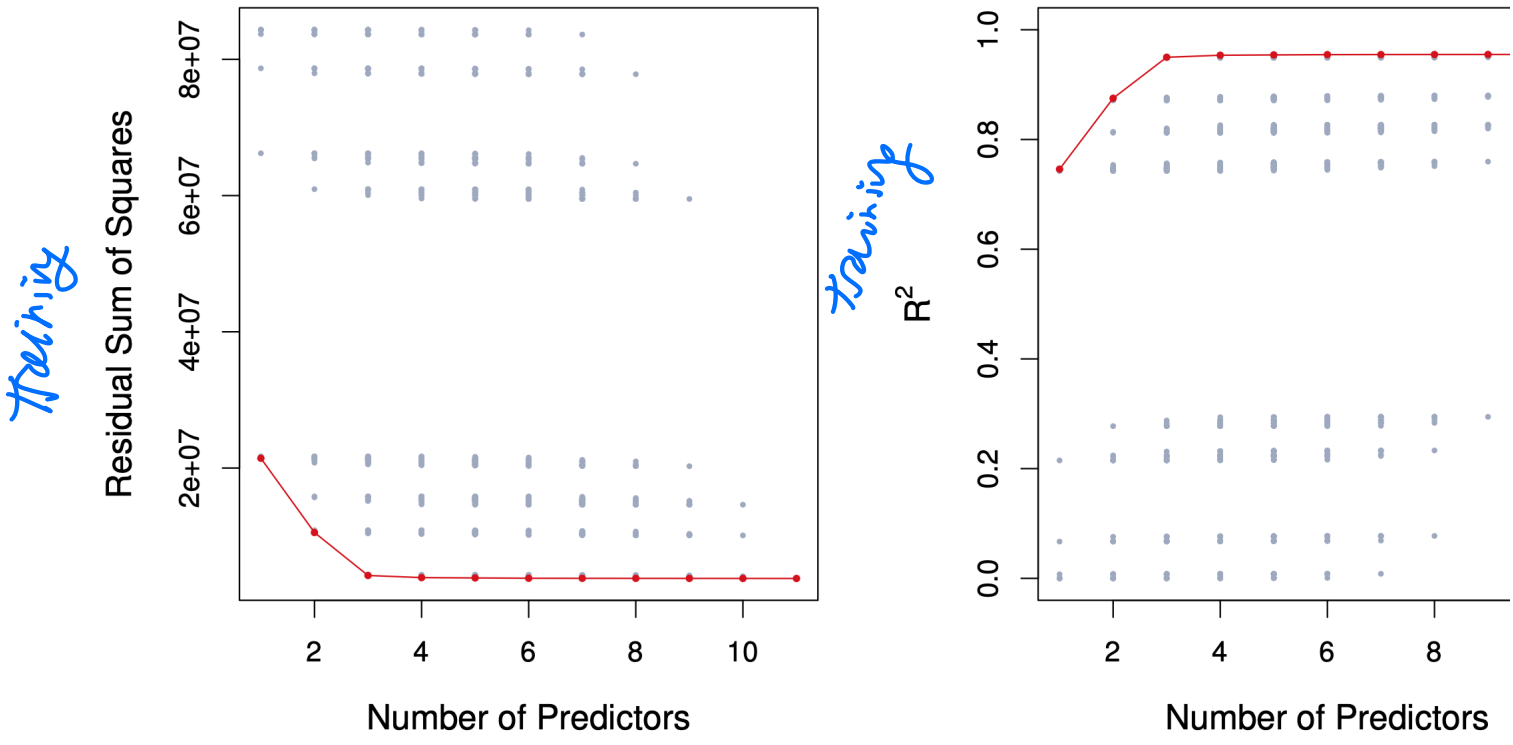
$Y \sim X[, E]$

$\# E = 2^p$



Best subset selection

- Simple idea: let's compare all models with k predictors.
- There are $\binom{p}{k} = p!/\left[k!(p-k)!\right]$ possible models.
- For every possible k, choose the model with the smallest RSS.



Best subset with `regsubsets`

```
library(ISLR2) # where Credit is stored
library(leaps) # where regsubsets is found
summary(regsubsets(Balance ~ ., data=Credit))
```

```
## Subset selection object
## Call: regsubsets.formula(Balance ~ ., data = Credit)
## 11 Variables  (and intercept)
##             Forced in Forced out
## Income          FALSE      FALSE
## Limit           FALSE      FALSE
## Rating          FALSE      FALSE
## Cards           FALSE      FALSE
## Age             FALSE      FALSE
## Education       FALSE      FALSE
## OwnYes          FALSE      FALSE
## StudentYes      FALSE      FALSE
## MarriedYes      FALSE      FALSE
## RegionSouth     FALSE      FALSE
## RegionWest      FALSE      FALSE
## 1 subsets of each size up to 8
## Selection Algorithm: exhaustive
##          Income Limit Rating Cards Age Education OwnYes StudentYes MarriedYes
## 1  ( 1 ) " "    " "   "*"    " "   " " " "       " "    " "        " "
## 2  ( 1 ) "*"    " "   "*"    " "   " " " "       " "    " "        " "
## 3  ( 1 ) "*"    " "   "*"    " "   " " " "       " "    "*"        " "
## 4  ( 1 ) "*"    "*"   " "    "*"   " " " "       " "    "*"        " "
## 5  ( 1 ) "*"    "*"   "*"    "*"   " " " "       " "    "*"        " "
## 6  ( 1 ) "*"    "*"   "*"    "*"   "*" " "       " "    "*"        " "
## 7  ( 1 ) "*"    "*"   "*"    "*"   "*" " "       "*"    "*"        " "
## 8  ( 1 ) "*"    "*"   "*"    "*"   "*" " "       "*"    "*"        " "
##          RegionSouth RegionWest
## 1  ( 1 ) " "         " "
## 2  ( 1 ) " "         " "
## 3  ( 1 ) " "         " "
## 4  ( 1 ) " "         " "
## 5  ( 1 ) " "         " "
## 6  ( 1 ) " "         " "
## 7  ( 1 ) " "         " "
## 8  ( 1 ) " "         "*"
```

$p \approx 1000$ MIXED INTEGER PROGRAMS...

$\Rightarrow 11$

- Best model with 4 variables includes: `Cards, Income, Student, Limit`.

Choosing k

Naturally, RSS and

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

improve as we increase k.

To optimize k, we want to minimize the test error, not the training error.

We could use **cross-validation**, or alternative estimates of test error:

- **Akaike Information Criterion (AIC)** (closely related to Mallow's C_p) given an estimate of the irreducible error $\hat{\sigma}^2$:

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

- **Bayesian Information Criterion (BIC):**

$$\frac{1}{n}(\text{RSS} + \log(n)\hat{\sigma}^2 k)$$

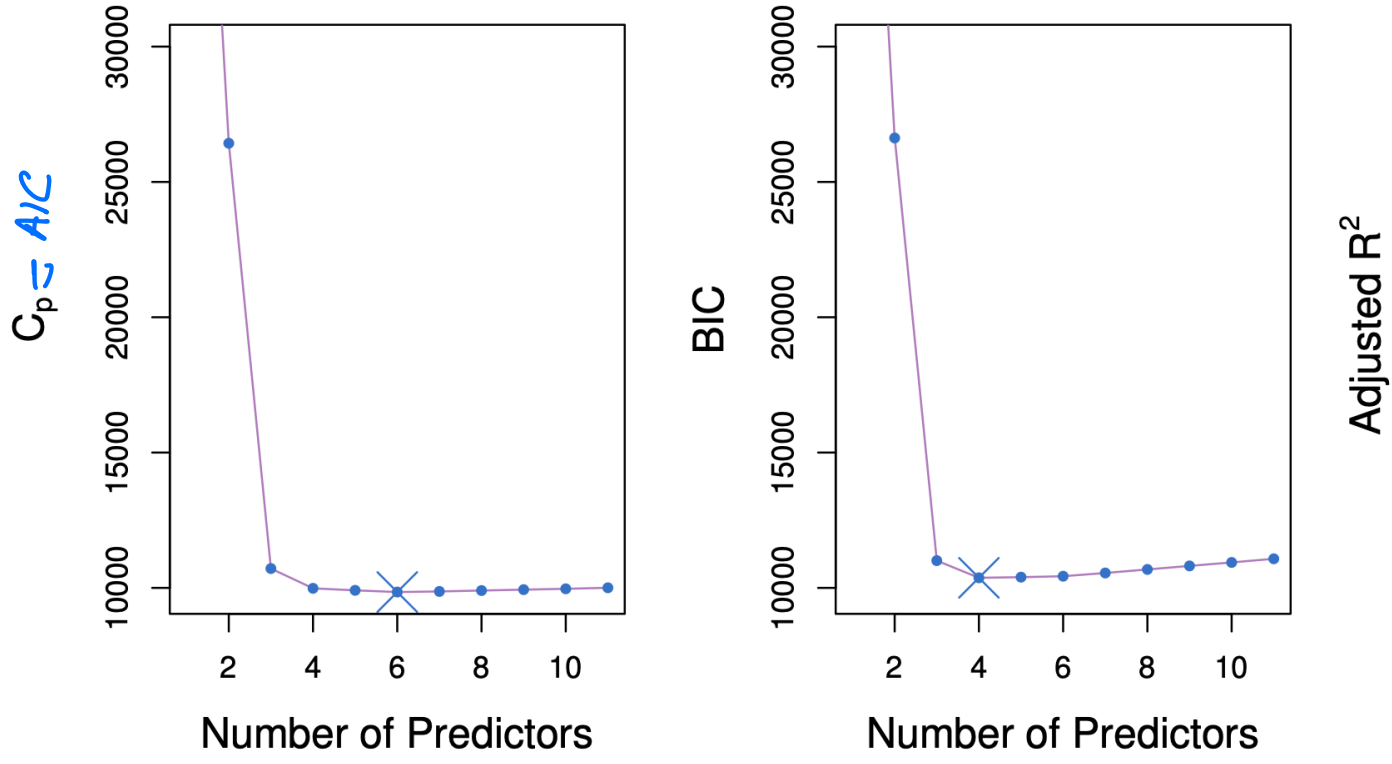
- **Adjusted R^2**:

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

How do these criteria above compare to cross validation?

- They are much less expensive to compute.
- They are motivated by asymptotic arguments and rely on model assumptions (eg. normality of the errors).
- Equivalent concepts for other models (e.g. logistic regression).

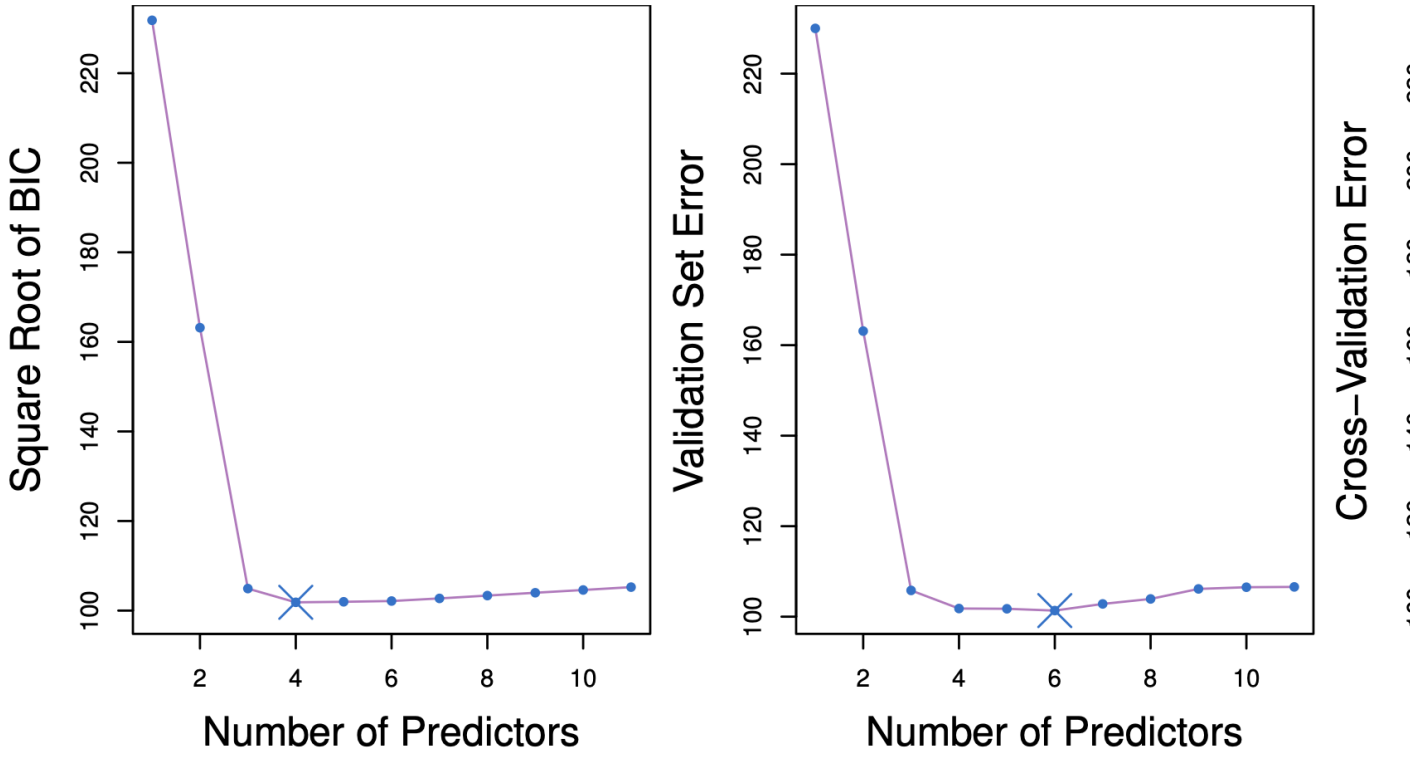
Example: best subset selection for the Credit dataset



Theory: - AIC (asymptotically) no false negatives

- BIC (") " " + no false positives

Best subset selection for the Credit dataset



Recall: In K-fold cross validation, we can estimate a standard error or accuracy for our test error estimate. Then, we can apply the *1 SE rule*.

Stepwise selection methods

Best subset selection has 2 problems:

1. It is often very expensive computationally. We have to fit 2^p models!
2. If for a fixed k, there are too many possibilities, we increase our chances of overfitting. The model selected has *high variance*.

In order to mitigate these problems, we can restrict our search space for the best model. This reduces the variance of the selected model at the expense of an increase in bias.

Forward selection

Algorithm (6.2 of ISLR)

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

Forward selection in using `step`

```
M.full = lm(Balance ~ ., data=Credit)
M.null = lm(Balance ~ 1, data=Credit)
step(M.null, scope=list(upper=M.full), direction='forward', trace=0)
```


```
##
## Call:
## lm(formula = Balance ~ Rating + Income + Student + Limit + Cards +
##     Age, data = Credit)
##
## Coefficients:
## (Intercept)       Rating       Income   StudentYes        Limit        Cards
##   -493.7342       1.0912      -7.7951     425.6099       0.1937      18.2119
##         Age
##     -0.6241
```

- First four variables are `Rating, Income, Student, Limit`.
- **Different from best subsets of size 4.**

Backward selection

Algorithm (6.3 of ISLR)

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

Backward selection

```
step(M.full, scope=list(lower=M.null), direction='backward', trace=0)
```

```
##
## Call:
## lm(formula = Balance ~ Income + Limit + Rating + Cards + Age +
##     Student, data = Credit)
##
## Coefficients:
## (Intercept)       Income        Limit       Rating        Cards          Age
##   -493.7342      -7.7951       0.1937       1.0912      18.2119      -0.6241
##  StudentYes
##    425.6099
```

In this case "forward" = "backward"

Forward selection with BIC

k default is 2

```
step(M.null, scope=list(upper=M.full), direction='forward', trace=0, k=log(nrow(Credit))) # k defaults to 2, i.e AIC
```

```
##
## Call:
## lm(formula = Balance ~ Rating + Income + Student + Limit + Cards,
##     data = Credit)
##
## Coefficients:
## (Intercept)       Rating       Income   StudentYes        Limit        Cards
##   -526.1555       1.0879      -7.8749     426.8501       0.1944      17.8517
```

Forward vs. backward selection

- You cannot apply backward selection when $p > n$.
- Although we might like them to, they need not produce the same sequence of models.
- **Example:** $X_1, X_2 \sim \mathcal{N}(0, \sigma)$ independent and set

$$\begin{aligned} X_3 &= X_1 + 3X_2 \\ Y &= X_1 + 2X_2 + \epsilon \end{aligned}$$

- Regress Y onto X_1, X_2, X_3.
  - *Forward:* $\{X_3\} \to \{\mathbf{X_3, X_2}\} \mathbf{or} \{\mathbf{X_3, X_1}\} \to \{X_3, X_2, X_1\}$
  - *Backward:* $\{X_1, X_2, X_3\} \to \{\mathbf{X_1, X_2}\} \to \{X_2\}$

Forward vs. backward selection

```
n = 5000
X1 = rnorm(n)
X2 = rnorm(n)
X3 = X1 + 3 * X2
Y = X1 + 2 * X2 + rnorm(n)
D = data.frame(X1, X2, X3, Y)
```

Forward

```
step(lm(Y ~ 1, data=D), list(upper=~ X1 + X2 + X3), direction='forward')
```

```
## Start:  AIC=9000.85
## Y ~ 1
##
##        Df Sum of Sq     RSS    AIC
## + X3    1   24872.6  5368.7  359.7
## + X2    1   20513.5  9727.8 3331.7
## + X1    1    4871.7 25369.6 8124.6
## <none>              30241.3 9000.9
##
## Step:  AIC=359.7
## Y ~ X3
##
##        Df Sum of Sq    RSS    AIC
## + X1    1    400.51 4968.1 -25.95
## + X2    1    400.51 4968.1 -25.95
## <none>              5368.7 359.70
##
## Step:  AIC=-25.95
## Y ~ X3 + X1
##
##        Df Sum of Sq    RSS     AIC
## <none>              4968.1 -25.952
```

```
##
## Call:
## lm(formula = Y ~ X3 + X1, data = D)
##
## Coefficients:
## (Intercept)           X3           X1
##    -0.01336      0.67367      0.29766
```

Backward

```
step(lm(Y ~ X1 + X2 + X3, data=D), list(lower=~ 1), direction='backward')
```

```
## Start:  AIC=-25.95
## Y ~ X1 + X2 + X3
##
##
## Step:  AIC=-25.95
## Y ~ X1 + X2
##
##        Df Sum of Sq     RSS    AIC
## <none>               4968.1  -26.0
## - X1    1    4759.6  9727.8 3331.7
## - X2    1   20401.4 25369.6 8124.6
```

```
##
## Call:
## lm(formula = Y ~ X1 + X2, data = D)
##
## Coefficients:
## (Intercept)           X1           X2
##    -0.01336      0.97134      2.02102
```

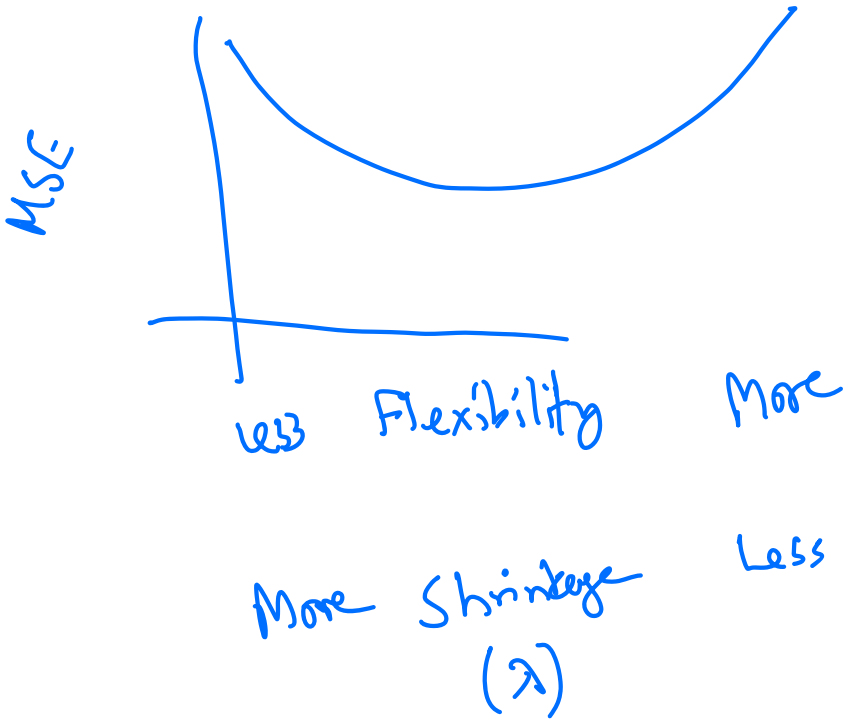
Other stepwise selection strategies

- **Mixed stepwise selection (`direction='both'`):** Do forward selection, but at every step, remove any variables that are no longer *necessary*.
- **Forward stagewise selection:** Roughly speaking, don't add in the variable *fully* at each step…
- …

Shrinkage methods

A mainstay of modern statistics!

- The idea is to perform a linear regression, while *regularizing* or *shrinking* the coefficients $\hat{\beta}$ toward 0.



Why would shrunk coefficients be better?

- This introduces *bias*, but may significantly decrease the *variance* of the estimates. If the latter effect is larger, this would decrease the test error.
- Extreme example: set $\hat{\beta}$ to 0 – variance is 0!
- There are Bayesian motivations to do this: priors concentrated near 0 tend to shrink the parameters' posterior distribution.

Example: $\hat{\beta}_{shrink}(\alpha) = \alpha (X^T X)^{-1} X^T Y$

$\alpha = 1$: Least squares estimate (No bias)

$\alpha = 0$: $\hat{\beta}_{shrink}(0) = 0$ (No variance)

Ridge regression

Ridge regression solves the following optimization:

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

- The RSS of the model at β.
- The squared ℓ_2 norm of β, or $\|\beta\|_2^2$.
- The parameter λ is a tuning parameter. It modulates the importance of fit vs. shrinkage.
- We find an estimate $\hat{\beta}_\lambda^R$ for many values of λ and then choose it by cross-validation. Fortunately, this is no more expensive than running a least-squares regression.

$\lambda = 0$: $\hat{\beta}_0 = \hat{\beta}_{LS}$

$\lambda = \infty$: $Y \sim 1$ (just an intercept)

Ridge regression

- In least-squares linear regression, scaling the variables has no effect on the fit of the model:

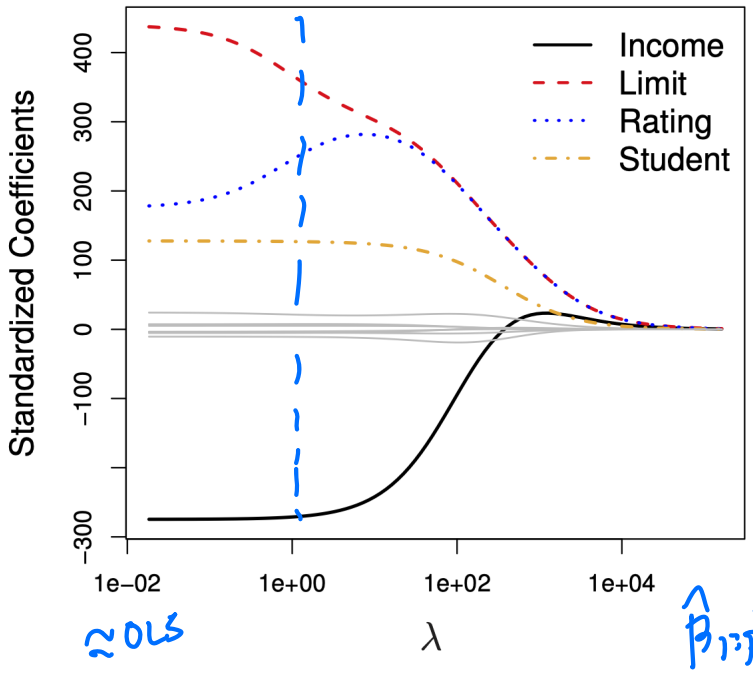
$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_p X_p + \epsilon.$$

$\tilde{X}_p = 2 X_p$

OLS: $\tilde{\beta}_p = \frac{1}{2} \hat{\beta}_p$

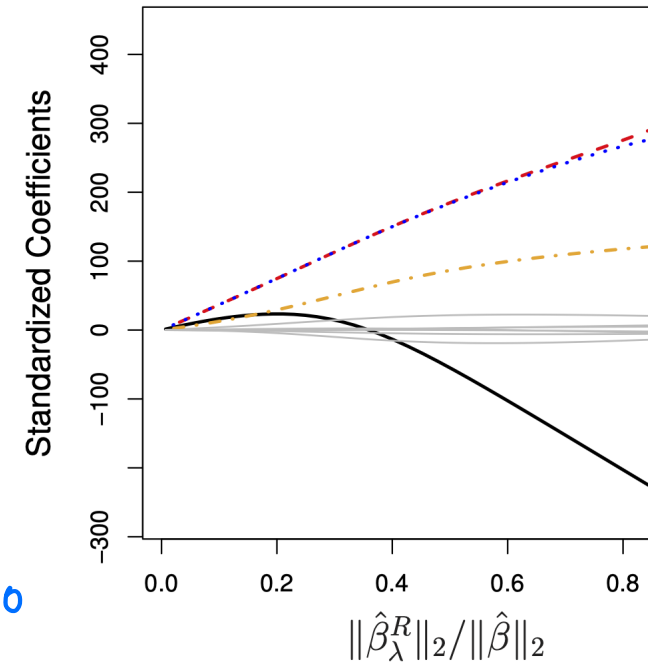
- Multiplying X_1 by c can be compensated by dividing $\hat{\beta}_1$ by c: after scaling we have the same RSS.
- In ridge regression, this is not true.
- In practice, what do we do?
  - *Scale each variable such that it has sample variance 1 before running the regression.*
  - *This prevents penalizing some coefficients more than others.*

Ridge regression of `balance` in the `Credit` dataset



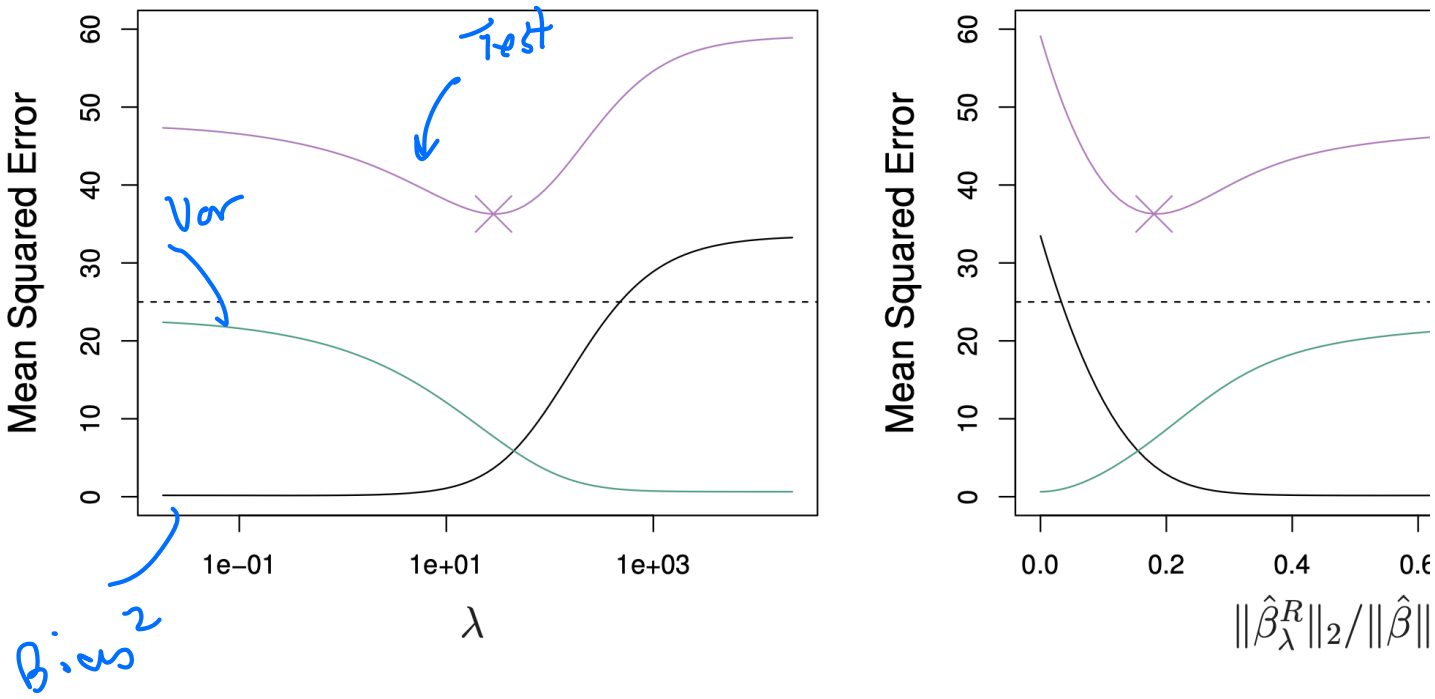
$\approx$ OLS

$\hat{\beta}_{\lambda=\infty} \approx 0$



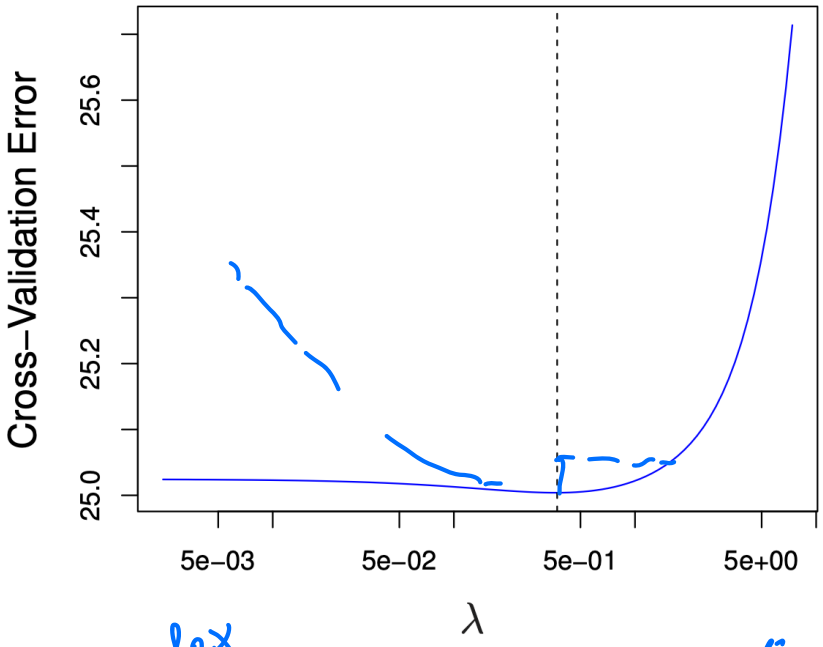
$\hat{\beta}^R_\lambda \neq 0$ for all $\lambda > 0$

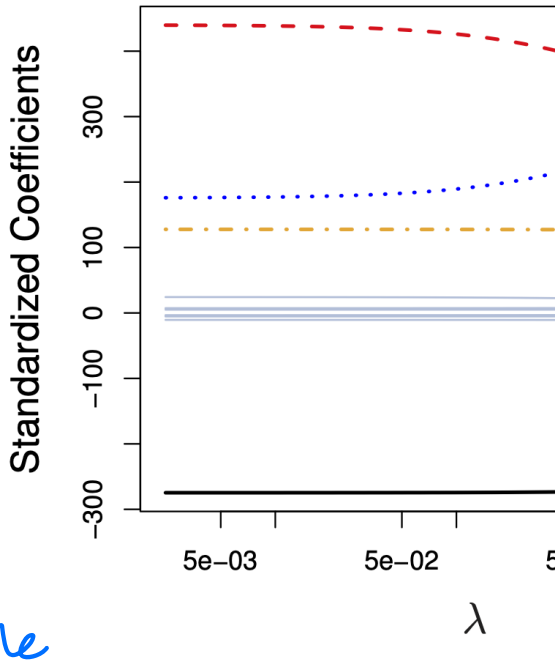
Ridge regression



- In a simulation study, we compute squared bias, variance, and test error as a function of λ.
- In practice, cross validation would yield an estimate of the test error but bias and variance are unobservable.

Selecting λ by cross-validation for ridge regression





complex

simple

Lasso regression

Lasso regression solves the following optimization:

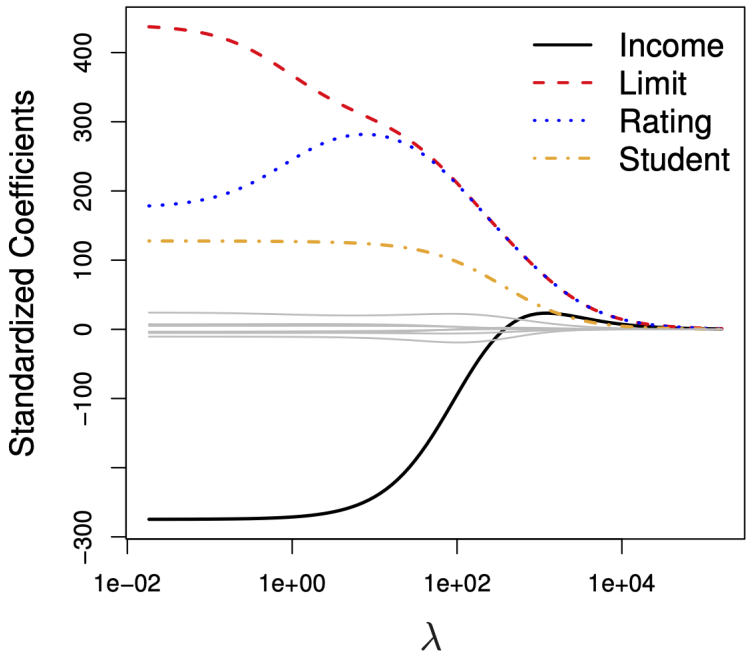
$$\min_{\beta} \quad \sum_{i=1}^{n} \left(y_i - \beta_0 - \sum_{j=1}^{p} \beta_j x_{i,j} \right)^2 + \lambda \sum_{j=1}^{p} |\beta_j|$$

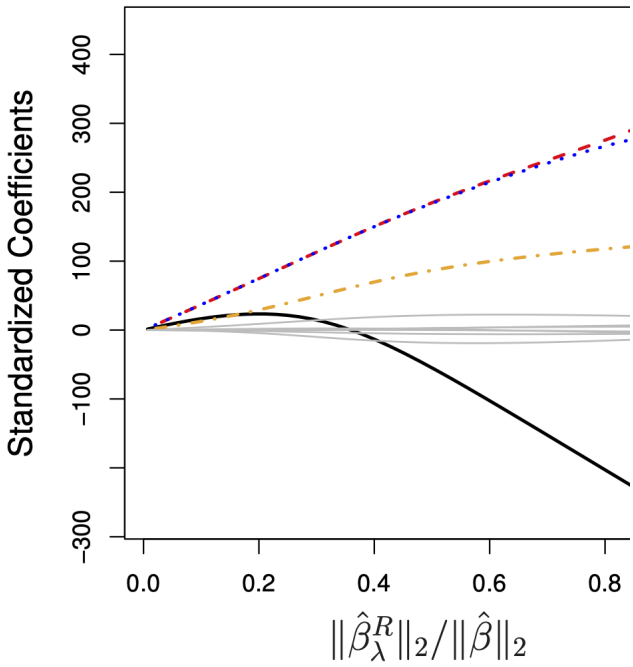
- RSS of the model at β.
- ℓ_1 norm of β, or $\|\beta\|_1$.
- The parameter λ is a tuning parameter. It modulates the importance of fit vs. shrinkage.

Why would we use the Lasso instead of Ridge regression?

- Ridge regression shrinks all the coefficients to a non-zero value.
- The Lasso shrinks some of the coefficients all the way to zero.
- A **convex** alternative (relaxation) to best subset selection (and its approximation stepwise selection)!

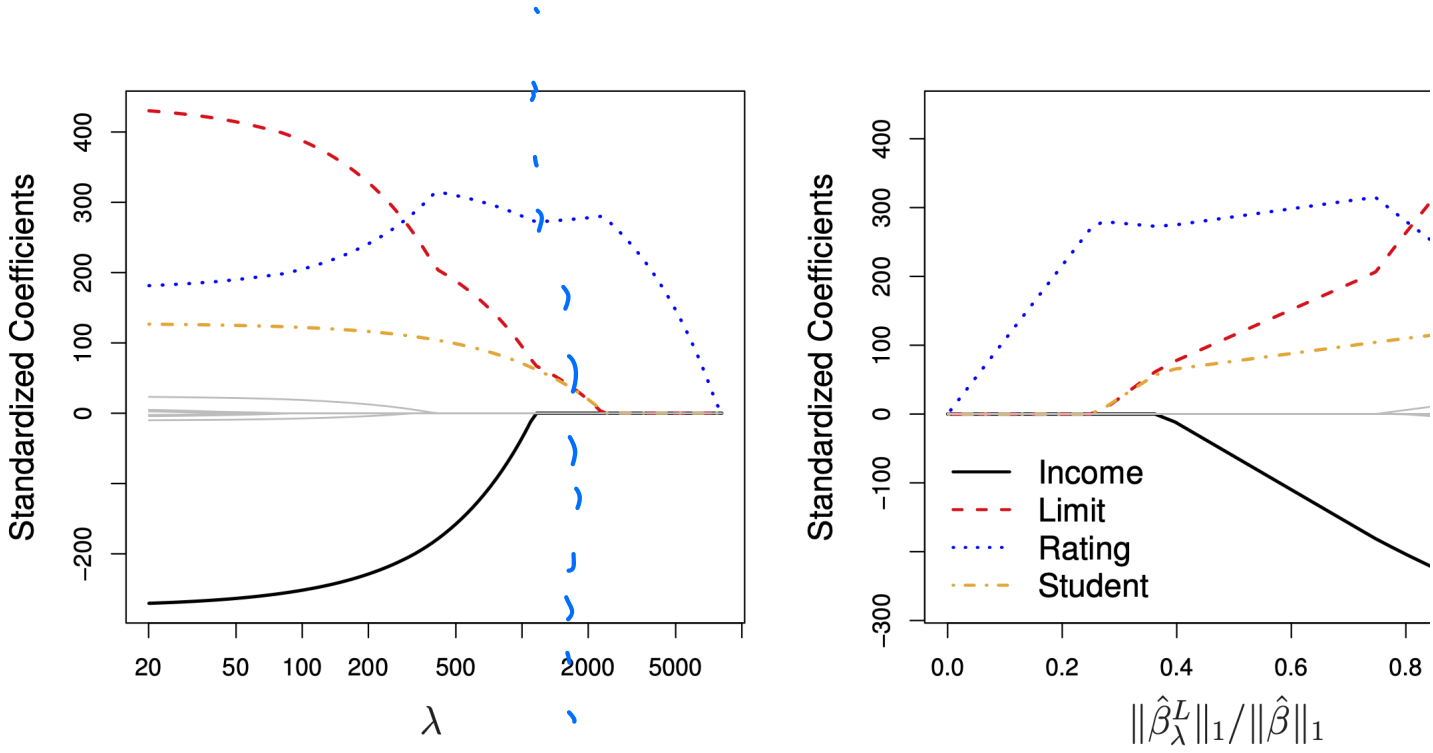
Ridge regression of `balance` in the `Credit` dataset





A lot of pesky small coefficients throughout the regularization path

Lasso regression of `balance` in the `Credit` dataset



- Those coefficients are shrunk to zero

Ridge: Just shrinkage

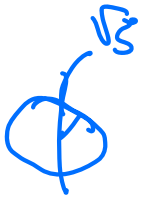
Lasso: Shrinkage & selection

Above 104 all coefs are zero!

An alternative formulation for regularization

- **Ridge:** for every λ, there is an s such that $\hat{\beta}^R_\lambda$ solves:

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



- **Lasso:** for every λ, there is an s such that $\hat{\beta}^L_\lambda$ solves:

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

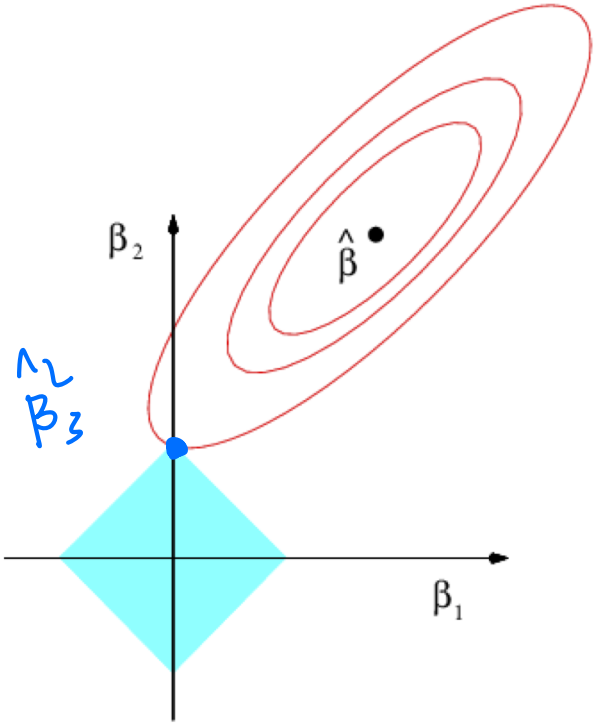
- **Best subset:**

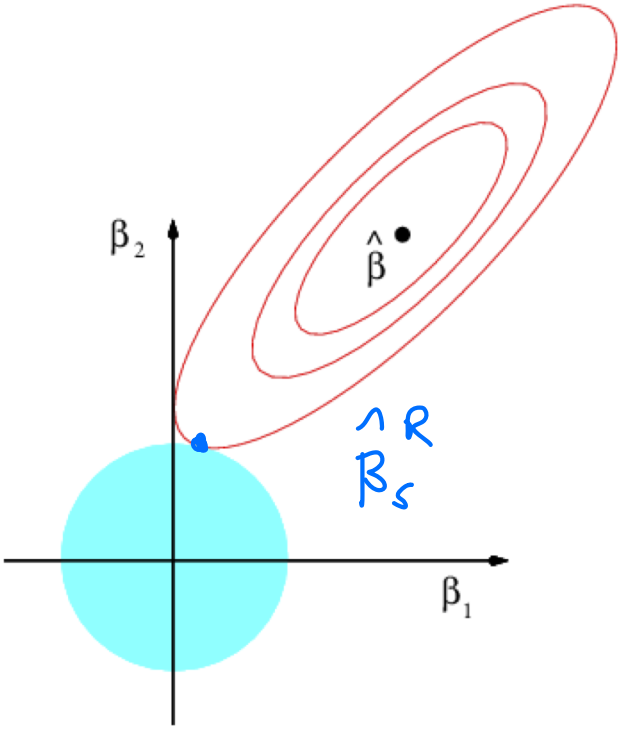
$$\underset{\beta}{\text{minimize}} \left\{ \sum_{i=1}^n \left(y_i - \beta_0 - \sum_{j=1}^p \beta_j x_{i,j} \right)^2 \right\} \quad \text{s.t.} \quad \sum_{j=1}^p \mathbf{1}(\beta_j \neq 0) < s.$$

Lagrange multiplier

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

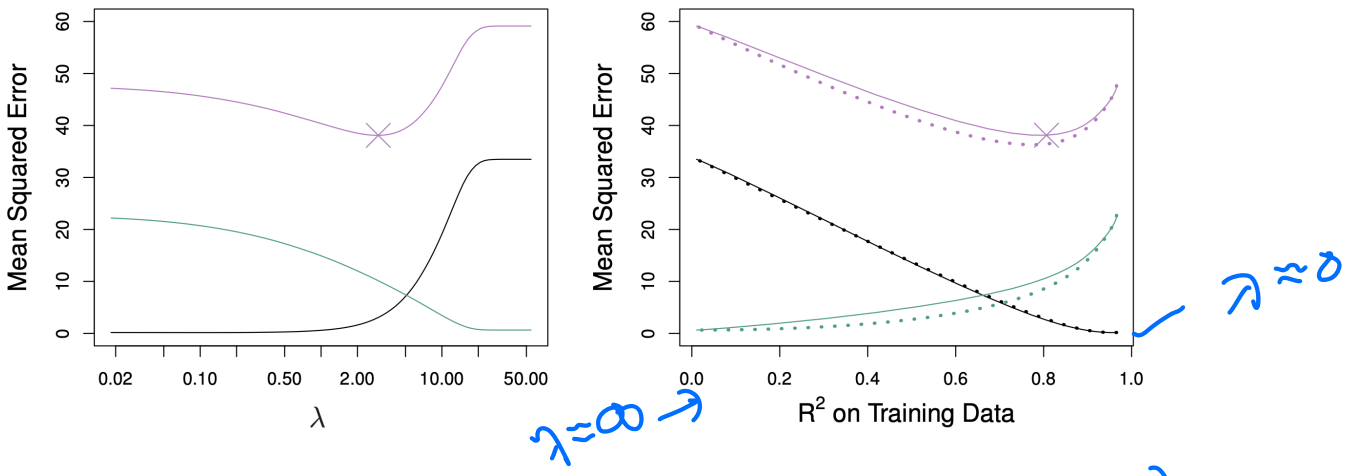
Visualizing Ridge and the Lasso with 2 predictors





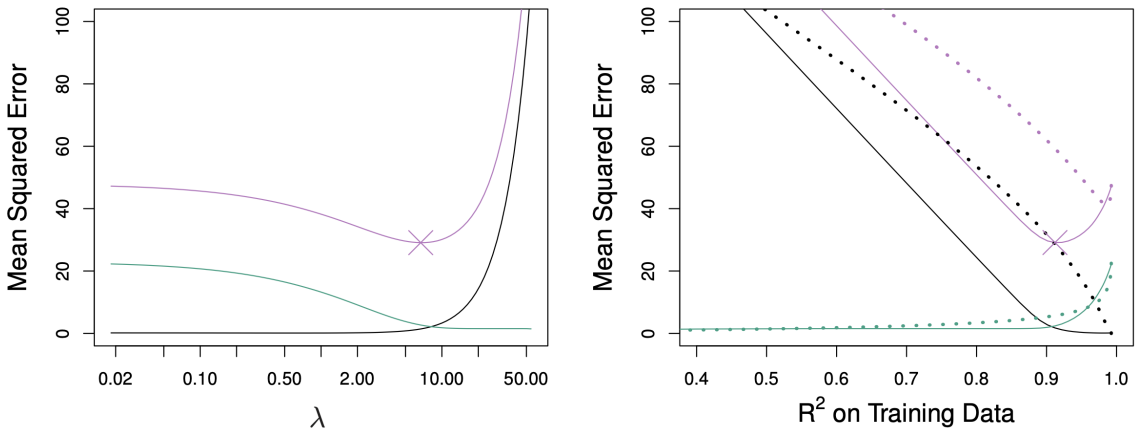
$Diamond: \sum_{j=1}^p |\beta_j| < s$ $Circle: \sum_{j=1}^p \beta_j^2 < s$, Best subset with $s = 1$ is union of the axes…

When is the Lasso better than Ridge?



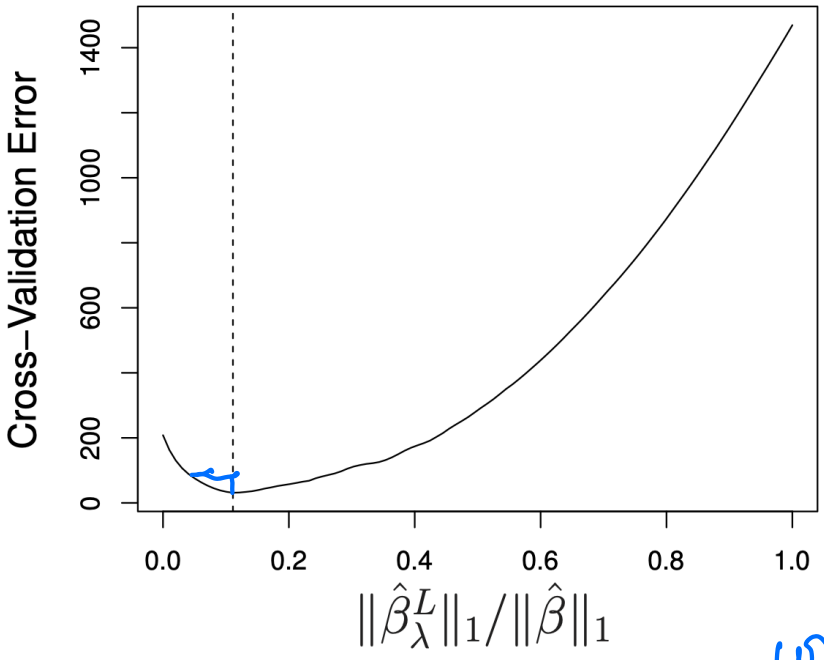
- **Example 1:** Most of the coefficients are non-zero. (Dense β)
- Bias, variance, MSE
- The Lasso (—), Ridge (⋯). Ridge wins
- The bias is about the same for both methods.
- The variance of Ridge regression is smaller, so is the MSE.

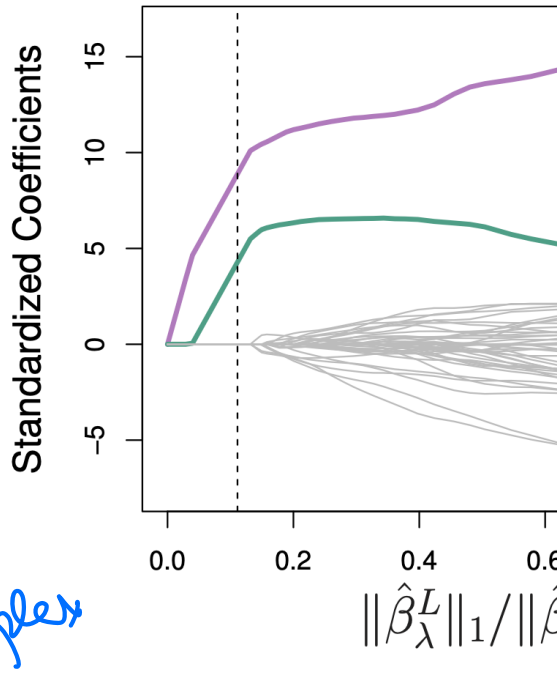
When is the Lasso better than Ridge?



- **Example 2:** Only 2 coefficients are non-zero. (Sparse B)
- Bias, variance, MSE
- The Lasso (—), Ridge (⋯). LASSO wins
- The bias, variance, and MSE are lower for the Lasso.

Selecting λ by cross-validation for lasso regression





simple

complex

A very special case: ridge

- Suppose $n = p$ and our matrix of predictors is $\mathbf{X} = I$
- Then, the objective function in Ridge regression can be simplified:

$$\sum_{j=1}^{p}(y_j - \beta_j)^2 + \lambda \sum_{j=1}^{p} \beta_j^2$$

- We can minimize the terms that involve each β_j separately:

$$\min_{\beta} \; (y_j - \beta_j)^2 + \lambda \beta_j^2.$$

- It is easy to show that

$$\hat{\beta}_j^R = \frac{y_j}{1+\lambda}.$$

$\alpha = \frac{1}{1+\lambda}$

$\hat{\beta}_{\text{shrink}}(\alpha)$

$\frac{\partial}{\partial \beta_j}$: $\quad 2(\hat{\beta}_j - y_j) + 2\lambda \hat{\beta}_j = 0$

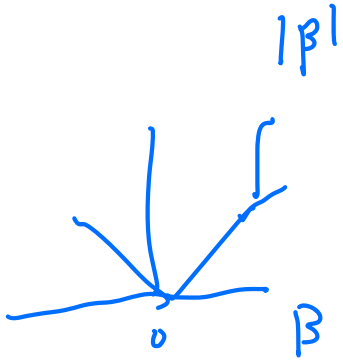
$2(1+\lambda)\hat{\beta}_j = 2y_j$

$\hat{\beta}_j = \frac{1}{1+\lambda} y_j$

A very special case: LASSO

- Similar story for the Lasso; the objective function is:

$$\sum_{j=1}^{p}(y_j - \beta_j)^2 + \lambda \sum_{j=1}^{p} |\beta_j|$$



- We can minimize the terms that involve each β_j separately:

$$(y_j - \beta_j)^2 + \lambda|\beta_j|.$$

- It is easy to show that

$$\hat{\beta}_j^L = \begin{cases} y_j - \lambda/2 & \text{if } y_j > \lambda/2; \\ y_j + \lambda/2 & \text{if } y_j < -\lambda/2; \\ 0 & \text{if } |y_j| < \lambda/2. \end{cases}$$

$\beta_j > 0$: $\quad 2(\hat{\beta}_j - y_j) + \lambda = 0$

$$\hat{\beta}_j = y_j - \frac{\lambda}{2}$$

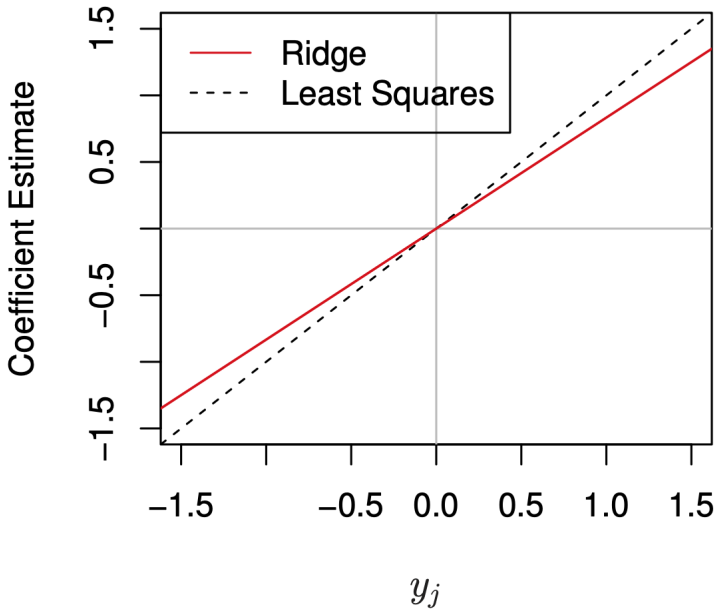
$\beta_j < 0$: $\quad 2(\hat{\beta}_j - y_j) - \lambda = 0$

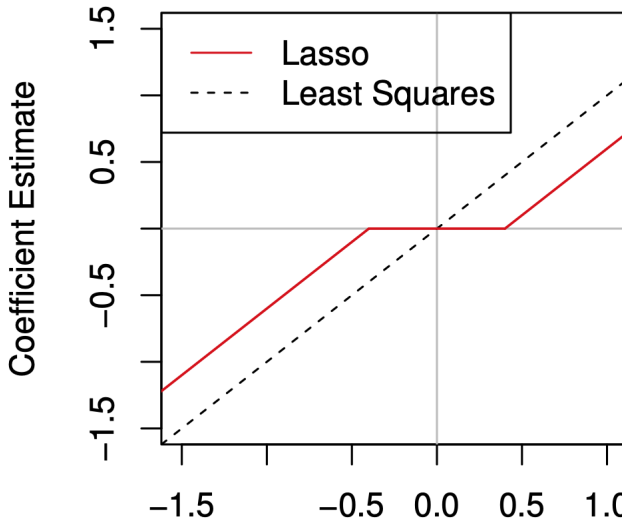
$$\hat{\beta}_j = y_j + \frac{\lambda}{2}$$

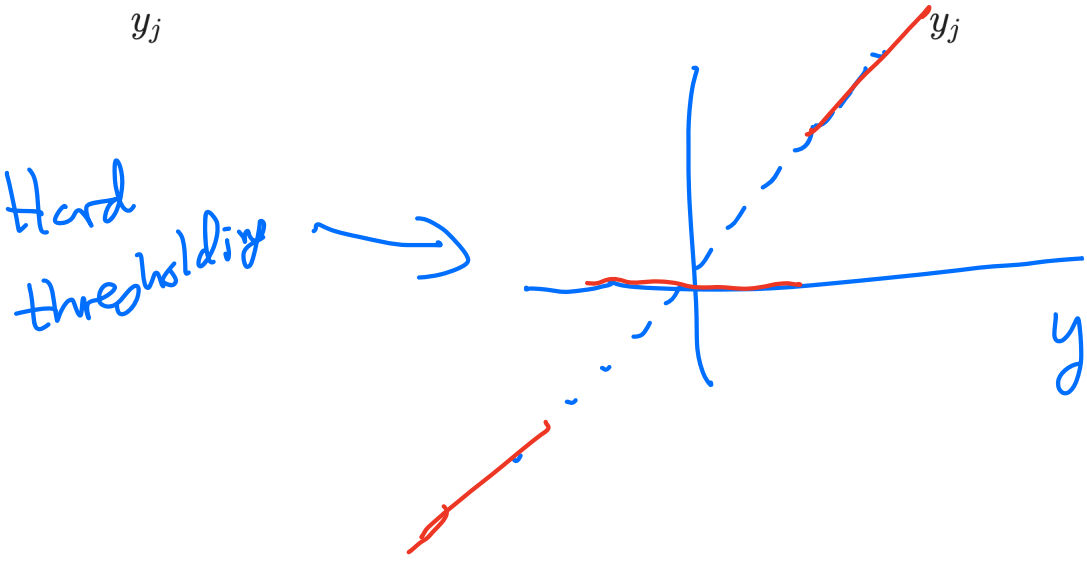
$$\hat{\beta}_j^L = \text{sign}(y_j)\ \max\left(|y_j| - \frac{\lambda}{2}, 0\right)$$

"soft thresholding"

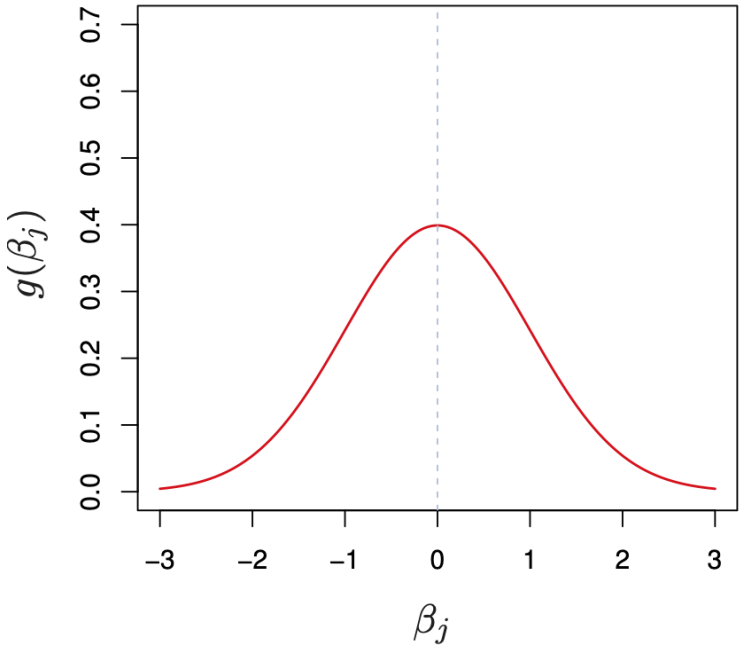
Lasso and Ridge coefficients as a function of y

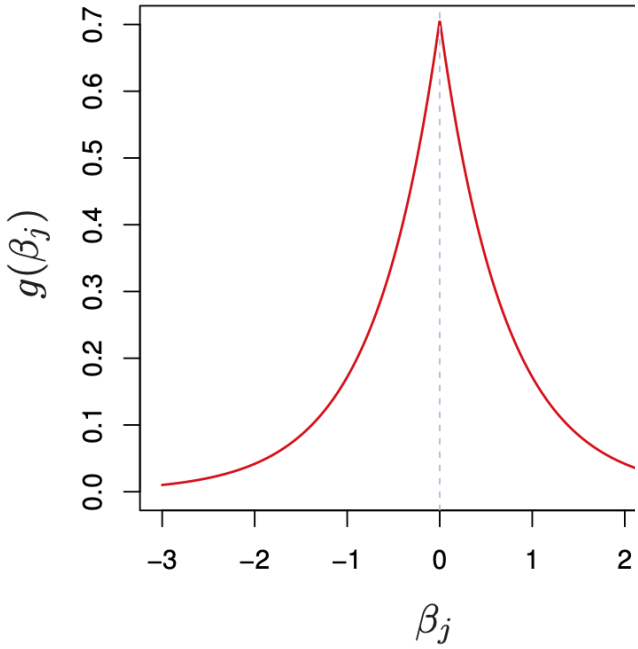






Bayesian interpretation





- **Ridge:** $\hat{\beta}^R$ is the posterior mean, with a Normal prior on β.
- **Lasso:** $\hat{\beta}^L$ is the posterior mode, with a Laplace prior on β.

Dimensionality reduction

Regularization methods

- Variable selection:
  - *Best subset selection*
  - *Forward and backward stepwise selection*
- Shrinkage
  - *Ridge regression*
  - *The Lasso (a form of variable selection)*
- Dimensionality reduction:
  - ***Idea:*** *Define a small set of M predictors which summarize the information in all p predictors.*

Principal Component Regression

- The loadings $\phi_{11}, \ldots, \phi_{p1}$ for the first principal component define the directions of greatest variance in the space of variables.

```
princomp(USArrests, cor=TRUE)$loadings[,1] # cor=TRUE makes sure to scale variables
```

```
##    Murder   Assault  UrbanPop      Rape
## 0.5358995 0.5831836 0.2781909 0.5434321
```

Interpretation: The first principal component measures the overall rate of crime.

Principal Component Regression

- The scores $z_{11}, \ldots, z_{n1}$ for the first principal component define the deviation of the samples along this direction.

```
princomp(USArrests, cor=TRUE)$scores[,1] # cor=TRUE makes sure to scale variables
```

```
##        Alabama         Alaska        Arizona       Arkansas     California
##     0.98556588     1.95013775     1.76316354    -0.14142029     2.52398013
##       Colorado    Connecticut       Delaware        Florida        Georgia
##     1.51456286    -1.35864746     0.04770931     3.01304227     1.63928304
##         Hawaii          Idaho       Illinois        Indiana           Iowa
##    -0.91265715    -1.63979985     1.37891072    -0.50546136    -2.25364607
##         Kansas       Kentucky      Louisiana          Maine       Maryland
##    -0.79688112    -0.75085907     1.56481798    -2.39682949     1.76336939
##  Massachusetts       Michigan      Minnesota    Mississippi       Missouri
##    -0.48616629     2.10844115    -1.69268181     0.99649446     0.69678733
##        Montana       Nebraska         Nevada New Hampshire     New Jersey
##    -1.18545191    -1.26563654     2.87439454    -2.38391541     0.18156611
##     New Mexico       New York North Carolina   North Dakota           Ohio
##     1.98002375     1.68257738     1.12337861    -2.99222562    -0.22596542
##       Oklahoma         Oregon   Pennsylvania   Rhode Island South Carolina
##    -0.31178286     0.05912208    -0.88841582    -0.86377206     1.32072380
##   South Dakota      Tennessee          Texas          Utah        Vermont
##    -1.98777484     0.99974168     1.35513821    -0.55056526    -2.80141174
##       Virginia     Washington  West Virginia      Wisconsin        Wyoming
##    -0.09633491    -0.21690338    -2.10858541    -2.07971417    -0.62942666
```

Interpretation: The scores for the first principal component measure the overall rate of crime in each state.

Principal Component Regression

- **Idea:**
  - *Replace the original predictors, $X_1, X_2, \ldots, X_p$, with the first M score vectors $Z_1, Z_2, \ldots, Z_M$.*
  - *Perform least squares regression, to obtain coefficients $\theta_0, \theta_1, \ldots, \theta_M$.*
- The model with these derived features is:

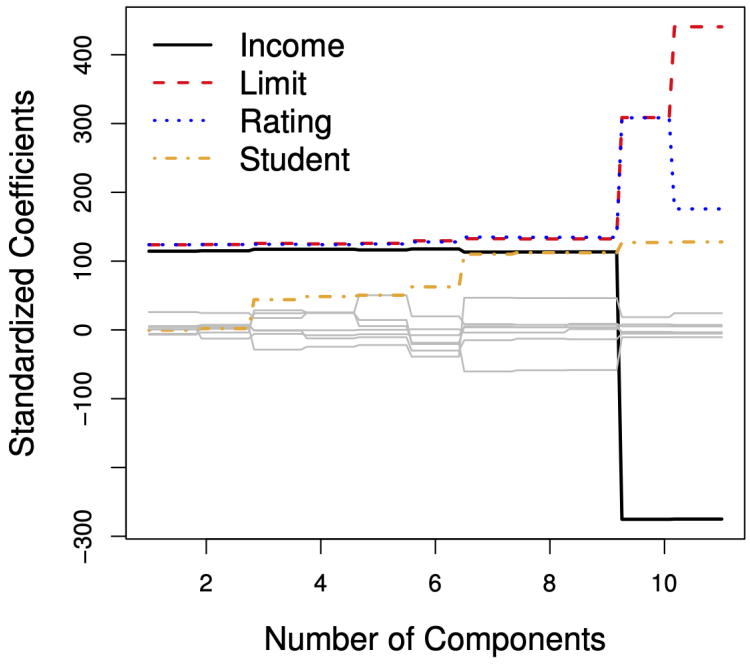
$$\begin{aligned} y_i &= \theta_0 + \theta_1 z_{i1} + \theta_2 z_{i2} + \cdots + \theta_M z_{iM} + \varepsilon_i \\ &= \theta_0 + \theta_1 \sum_{j=1}^{p} \phi_{j1} x_{ij} + \theta_2 \sum_{j=1}^{p} \phi_{j2} x_{ij} + \cdots + \theta_M \sum_{j=1}^{p} \phi_{jM} x_{ij} + \varepsilon_i \\ &= \theta_0 + \left[\sum_{m=1}^{M} \theta_m \phi_{1m} \right] x_{i1} + \cdots + \left[\sum_{m=1}^{M} \theta_m \phi_{pm} \right] x_{ip} + \varepsilon_i \end{aligned}$$

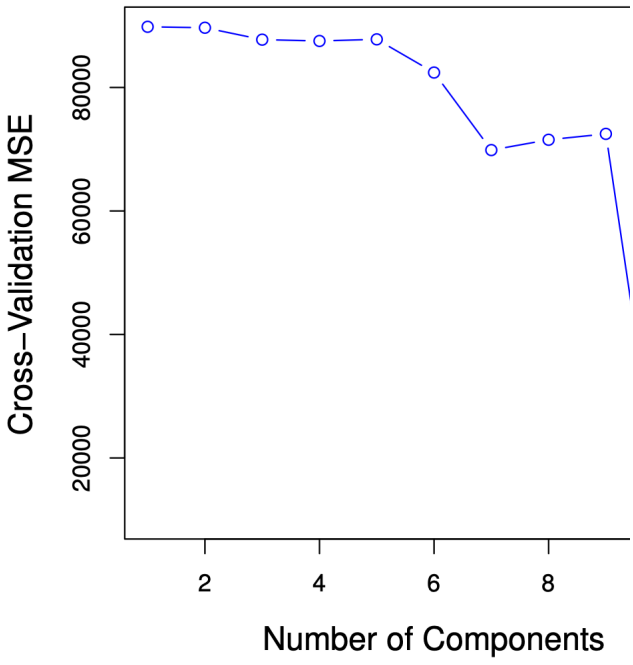
- Equivalent to a linear regression onto $X_1, \ldots, X_p$, with coefficients:

$$\beta_j = \sum_{m=1}^{M} \theta_m \phi_{jm}$$

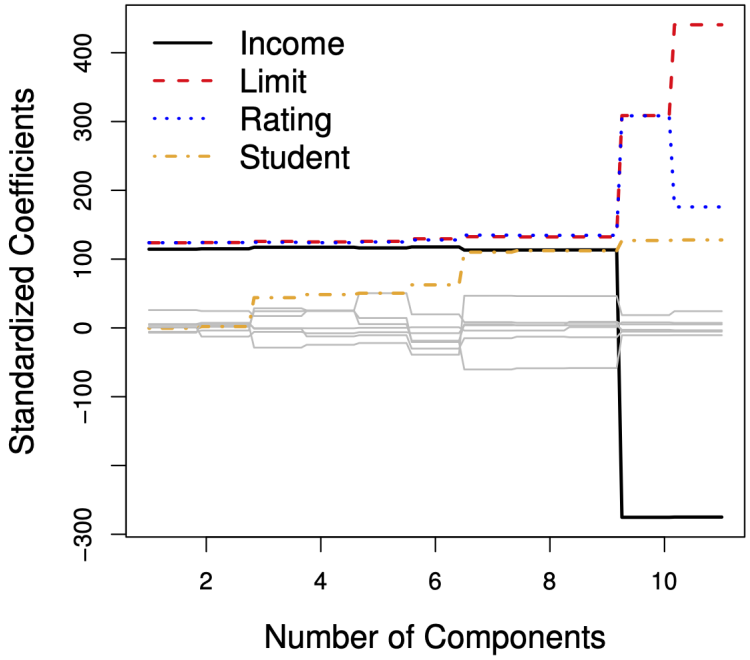
- This constraint in the form of β_j introduces *bias*, but it can lower the *variance* of the model.

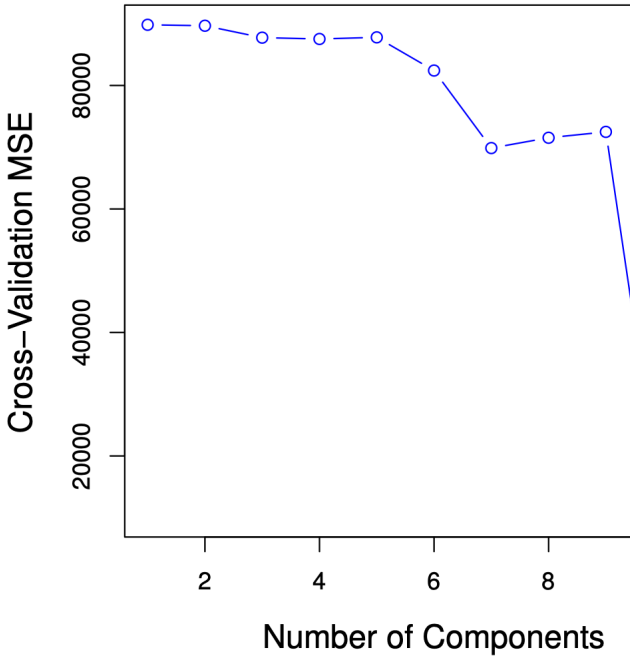
Application to the `Credit` dataset





- A model with 11 components is equivalent to least-squares regression
- Best CV error is achieved with 10 components (almost no dimensionality reduction)





- The left panel shows the coefficients β_j estimated for each M.
- The coefficients shrink as we decrease M!

Relationship between PCR and Ridge regression

- **Least squares regression:** want to minimize

$$RSS = (y - \mathbf{X}\beta)^T (y - \mathbf{X}\beta)$$

- **Score equation:**

$$\frac{\partial RSS}{\partial \beta} = -2\mathbf{X}^T (y - \mathbf{X}\beta) = 0$$

- **Solution:**

$$\implies \hat{\beta} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T y$$

- Compute singular value decomposition: $\mathbf{X} = UD^{1/2}V^T$, where $D^{1/2} = \text{diag}(\sqrt{d_1}, \ldots, \sqrt{d_p})$.

- Then

$$(\mathbf{X}^T\mathbf{X})^{-1} = VD^{-1}V^T$$

where $D^{-1} = \text{diag}(1/d_1, 1/d_2, \ldots, 1/d_p)$.

Relationship between PCR and Ridge regression

- **Ridge regression:** want to minimize

$$RSS + \lambda \|\beta\|_2^2 = (y - \mathbf{X}\beta)^T (y - \mathbf{X}\beta) + \lambda \beta^T \beta$$

$$RSS = (y - \mathbf{X}\beta)^T (y - \mathbf{X}\beta)$$

- **Score equation:**

$$\frac{\partial (RSS + \lambda \|\beta\|_2^2)}{\partial \beta} = -2\mathbf{X}^T (y - \mathbf{X}\beta) + 2\lambda\beta = 0$$

- **Solution:**

$$\Longrightarrow \hat{\beta}_\lambda^R = (\mathbf{X}^T\mathbf{X} + \lambda I)^{-1}\mathbf{X}^T y$$

- Note

$$(\mathbf{X}^T\mathbf{X} + \lambda I)^{-1} = V D_\lambda^{-1} V^T$$

where $D_\lambda^{-1} = \text{diag}(1/(d_1 + \lambda), 1/(d_2 + \lambda), \ldots, 1/(d_p + \lambda))$.

Relationship between PCR and Ridge regression

- **Predictions of least squares regression:**

$$\hat{y} = \mathbf{X}\hat{\beta} = \sum_{j=1}^{p} u_j u_j^T y, \qquad u_j \text{ is the } j\text{th column of } U$$

- **Predictions of Ridge regression:**

$$\hat{y} = \mathbf{X}\hat{\beta}_\lambda^R = \sum_{j=1}^{p} u_j \frac{d_j}{d_j + \lambda} u_j^T y$$

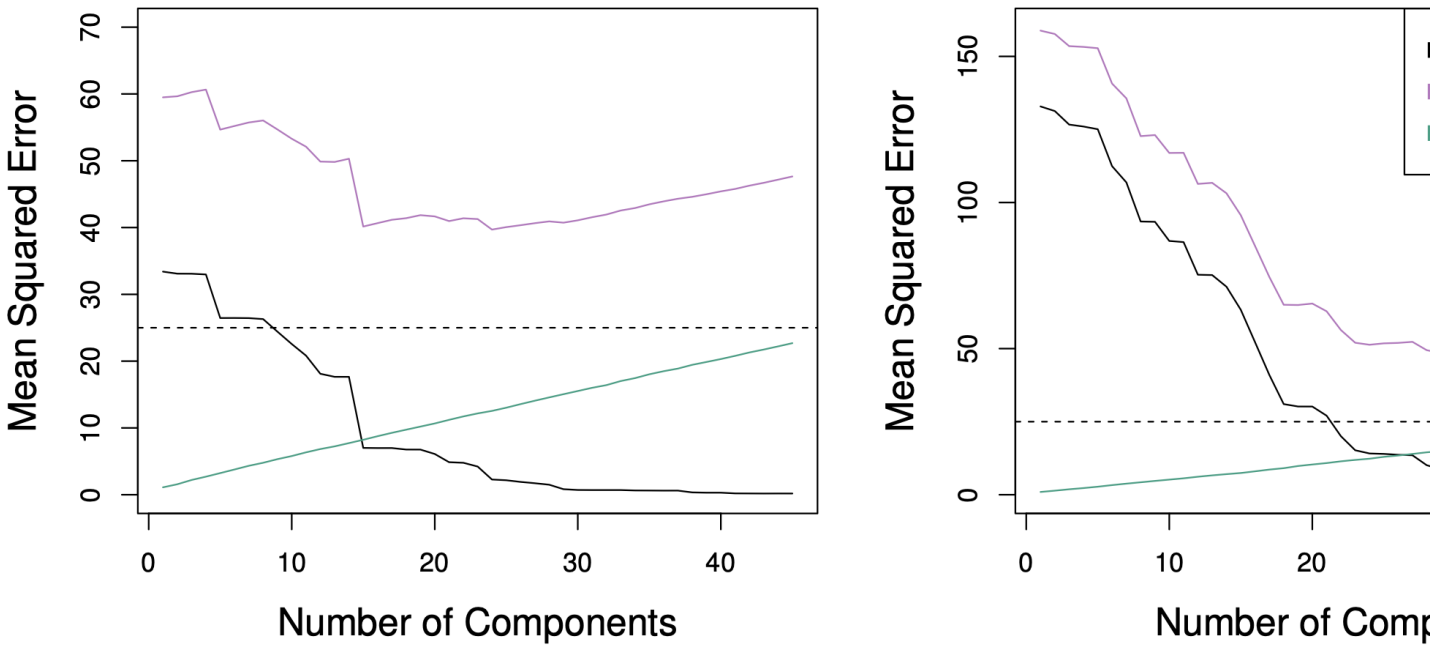
- The projection of y onto a principal component is shrunk toward zero. The smaller the principal component, the larger the shrinkage.

- **Predictions of PCR:**

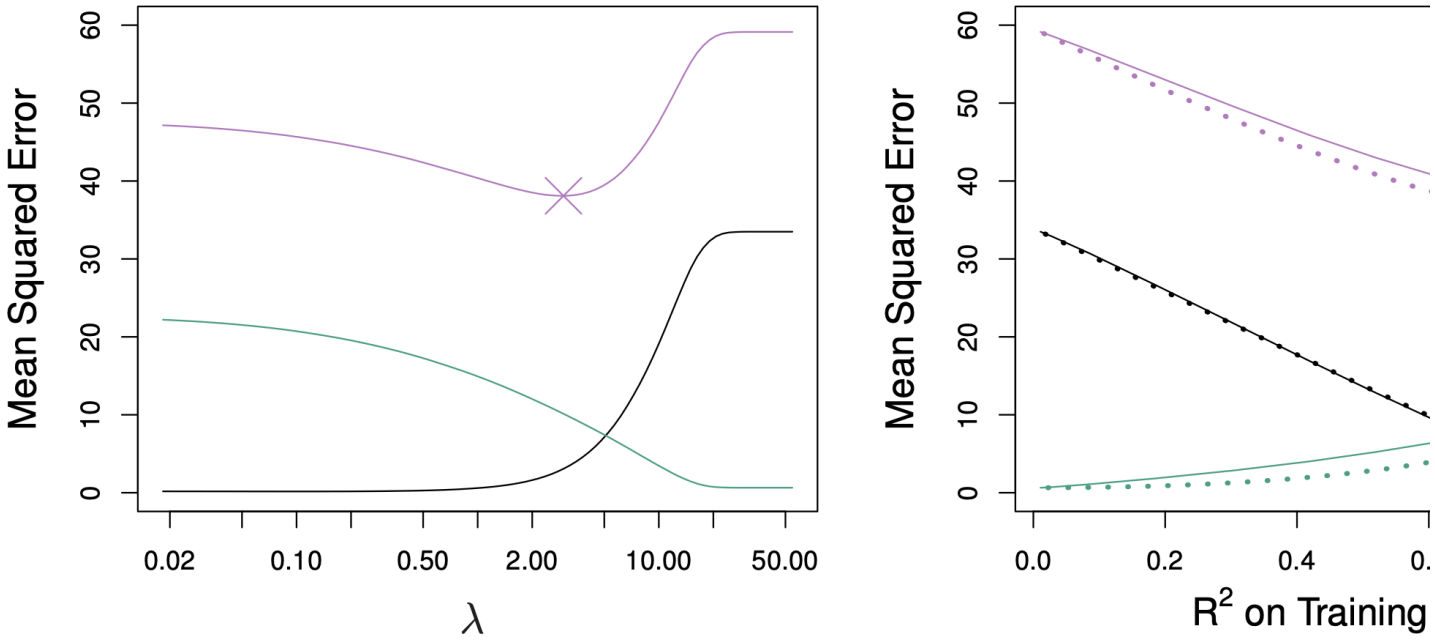
$$\hat{y} = \mathbf{X}\hat{\beta}^{\text{PC}} = \sum_{j=1}^{p} u_j \mathbf{1}(j \leq M) u_j^T y$$

- The projections onto small principal components are shrunk to zero.

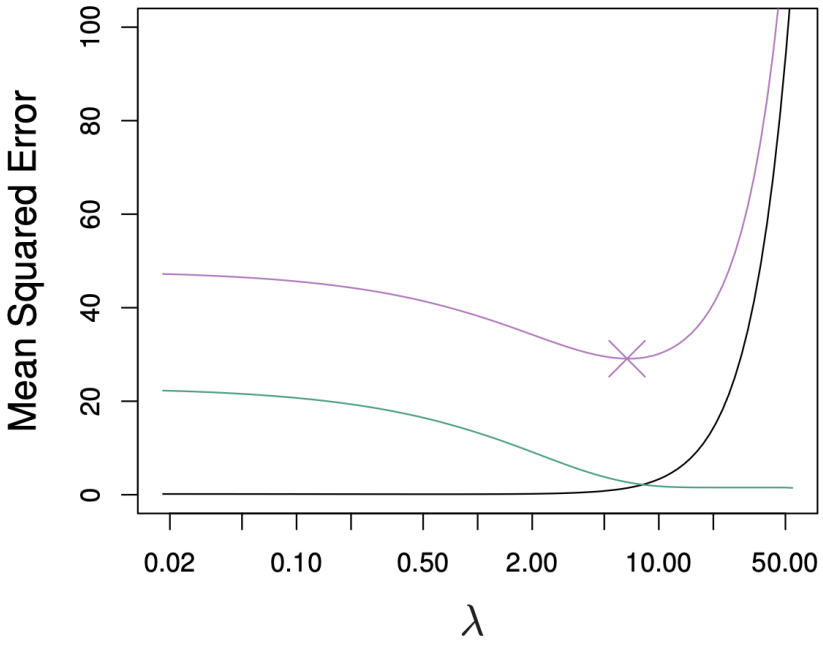
Principal Component Regression

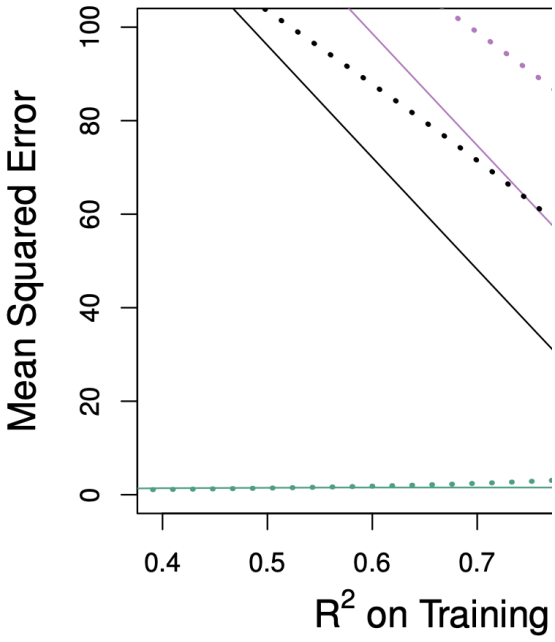


- In each case $n = 50$, $p = 45$.
- Left: response is a function of all the predictors (*dense*).
- Right: response is a function of 2 predictors (*sparse* - good for Lasso).

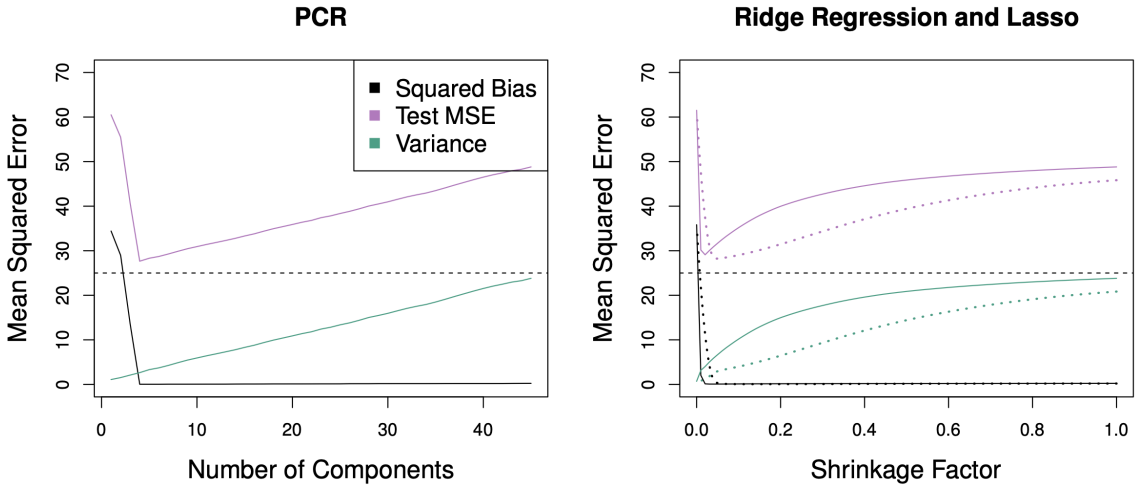


- Ridge and Lasso on *dense* dataset
- The Lasso (—), Ridge (⋯).
- Optimal PCR seems at least comparable to optimal LASSO and ridge.





- Ridge and Lasso on *sparse* dataset
- The Lasso (—), Ridge (⋯).
- Lasso clearly better than PCR here.



- Again, $n = 50$, $p = 45$ (as in ridge, LASSO examples)
- The response is a function of the first 5 principal components.

Partial least squares

- Principal components regression derives $Z_1, \ldots, Z_M$ using *only* the predictors $X_1, \ldots, X_p$.
- In partial least squares, we use the response Y as well. (To be fair, best subsets and stepwise do as well.)

Algorithm

1. Define $Z_1 = \sum_{j=1}^{p} \phi_{j1} X_j$, where ϕ_{j1} is the coefficient of a simple linear regression of Y onto X_j.
2. Let $X_j^{(2)}$ be the residual of regressing X_j onto Z_1.
3. Define $Z_2 = \sum_{j=1}^{p} \phi_{j2} X_j^{(2)}$, where ϕ_{j2} is the coefficient of a simple linear regression of Y onto $X_j^{(2)}$.
4. Let $X_j^{(3)}$ be the residual of regressing $X_j^{(2)}$ onto Z_2.

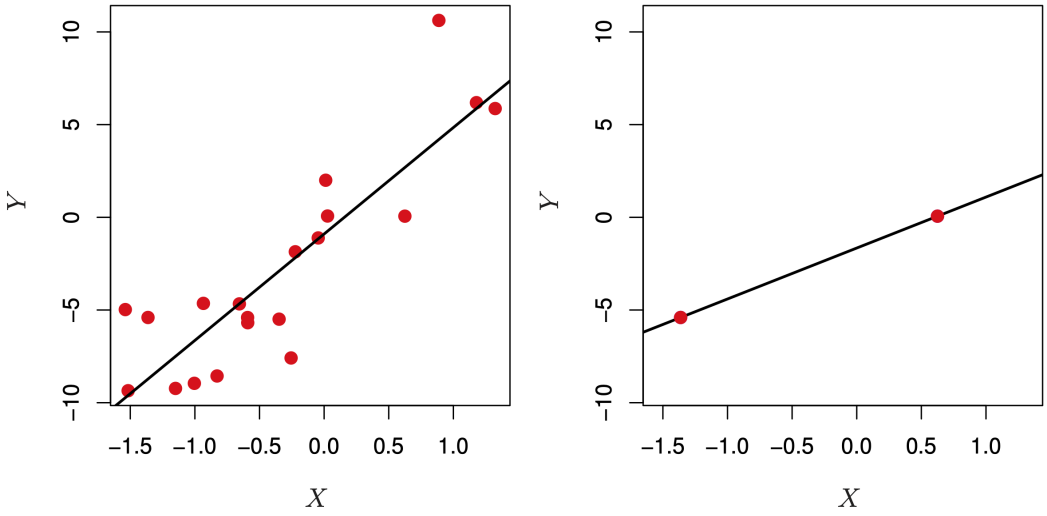
Partial least squares

- At each step, we try to find the linear combination of predictors that is highly correlated to the response (the highest correlation is the least squares fit).
- After each step, we transform the predictors such that they are from the linear combination chosen.
- Compared to PCR, partial least squares has less bias and more variance (a stronger tendency to overfit).

High-dimensional regression

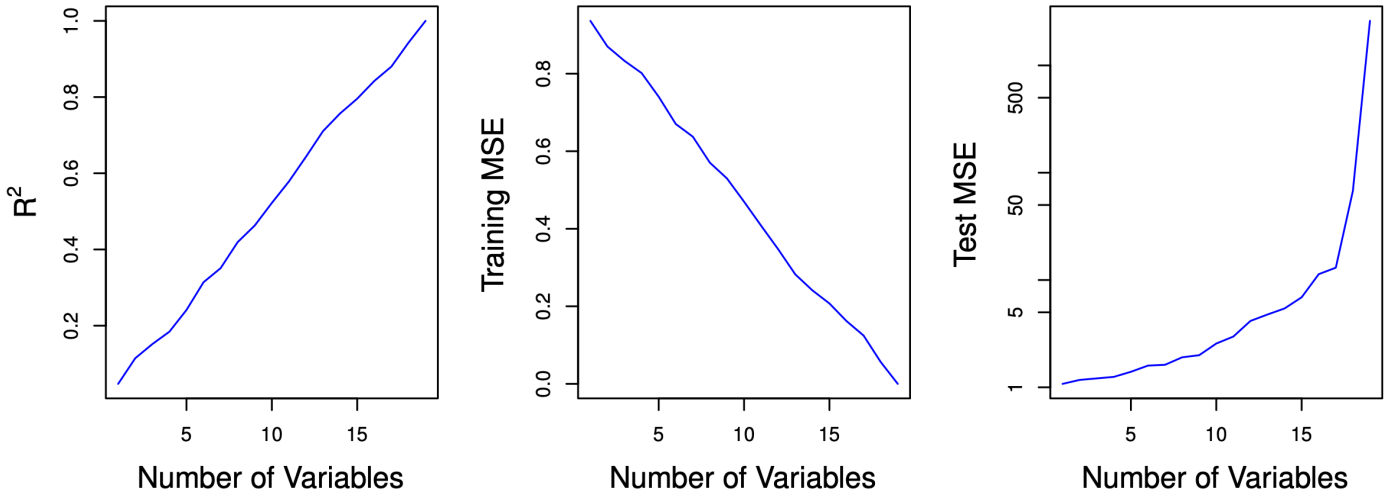
- Most of the methods we've discussed work best when n is much larger than p.
- However, the case $p \gg n$ is now common, due to experimental advances and cheaper computers:
  - ***Medicine:*** *Instead of regressing heart disease onto just a fewclinical observations (blood pressure, salt consumption, age), we use in addition 500,000 single nucleotide polymorphisms.*
  - ***Marketing:*** *Using search terms to understand online shopping patterns. A bag of words model defines one feature for every possible search term, which counts the number of times the term appears in a person's search. There can be as many features as words in the dictionary.*

Some problems



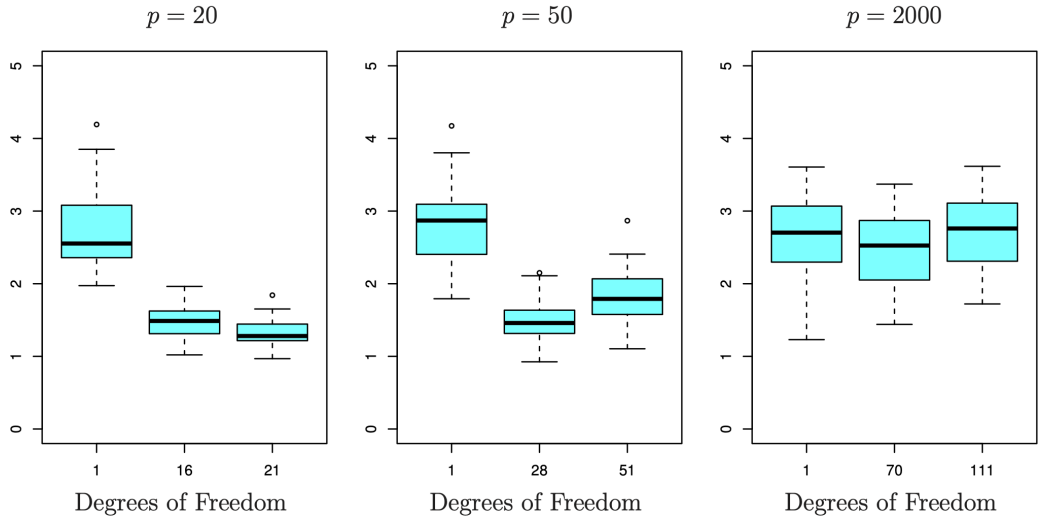
- When $n = p$, we can find a fit that goes through every point.
- We could use regularization methods, such as variable selection, ridge regression and the lasso.

Some problems



- Measures of training error are really bad.
- Furthermore, it becomes hard to estimate the noise $\hat{\sigma}^2$.
- Measures of model fit C_p, AIC, and BIC fail.

Some problems



- In each case, only 20 predictors are associated to the response.
- Plots show the test error of the Lasso.
- **Message:** Adding predictors that are uncorrelated with the response hurts the performance of the regression!

Interpreting coefficients when $p > n$

- When $p > n$, every predictor is a linear combination of other predictors, i.e. there is an extreme level of multicollinearity.
- The Lasso and Ridge regression will choose one set of coefficients.
- The coefficients selected $\{i \; ; \; |\hat{\beta}_i| > \delta\}$ are not guaranteed to be identical to $\{i \; ; \; |\beta_i| > \delta\}$. There can be many sets of predictors (possibly non-overlapping) which yield apparently good models.
- **Message:** Don't overstate the importance of the predictors selected.

Interpreting inference when $p > n$

- When $p > n$, LASSO might select a sparse model.
- Running `lm` on selected variables on *training data* is **bad.**
- Running `lm` on selected variables on independent *validation data* is **OK-ish** – is this `lm` a good model?
- **Message:** Don't use inferential methods developed for least squares regression for things like LASSO, forward stepwise, etc.
- Can we do better? Yes, but it's complicated and a little above our level here.