Lecture 6: Model Selection and Regularization

STATS 202: Statistical Learning and Data Science

Linh Tran

tranlm@stanford.edu

Department of Statistics
Stanford University

July 9, 2025

Announcements

- HW1 is still being graded.
- HW2 due Monday (available on both Gradescope & syllabus page).
- Midterm is in 1 week.
  - Will be in person.
  - Let the teaching staff know if you need special accomodations.
  - Practice exam will be released tonight.
  - Solutions to practice midterm will be posted this weekend.

- Bootstrap
- Jackknife
- Subset selection
- Shrinkage methods
  - Ridge
  - LASSO
  - Elastic net

Standard errors

An example: Standard errors in linear regression

```
Residuals:
    Min      1Q  Median      3Q     Max
-15.594  -2.730  -0.518   1.777  26.199

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)  3.646e+01  5.103e+00   7.144 3.28e-12 ***
crim        -1.080e-01  3.286e-02  -3.287 0.001087 **
zn           4.642e-02  1.373e-02   3.382 0.000778 ***
indus        2.056e-02  6.150e-02   0.334 0.738288
chas         2.687e+00  8.616e-01   3.118 0.001925 **
nox         -1.777e+01  3.820e+00  -4.651 4.25e-06 ***
rm           3.810e+00  4.179e-01   9.116  < 2e-16 ***
age          6.922e-04  1.321e-02   0.052 0.958229
dis         -1.476e+00  1.995e-01  -7.398 6.01e-13 ***
rad          3.060e-01  6.635e-02   4.613 5.07e-06 ***
tax         -1.233e-02  3.761e-03  -3.280 0.001112 **
ptratio     -9.527e-01  1.308e-01  -7.283 1.31e-12 ***
black        9.312e-03  2.686e-03   3.467 0.000573 ***
lstat       -5.248e-01  5.072e-02 -10.347  < 2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 4.745 on 492 degrees of freedom
Multiple R-Squared: 0.7406,     Adjusted R-squared: 0.7338
F-statistic: 108.1 on 13 and 492 DF,  p-value: < 2.2e-16
```

Standard errors

More generally: Obtain estimator's *sampling distribution*

Standard errors

More generally: Obtain estimator's *sampling distribution*

Example: The variance of a sample $x_1, x_2, ..., x_n$

$$\hat{\sigma}_n^2 = \frac{1}{n-1} \sum_{i=1}^{n} (x_i - \bar{x})^2 \tag{1}$$

Standard errors

More generally: Obtain estimator's *sampling distribution*

Example: The variance of a sample $x_1, x_2, ..., x_n$

$$\hat{\sigma}_n^2 = \frac{1}{n-1} \sum_{i=1}^{n} (x_i - \bar{x})^2 \tag{1}$$

How to get the standard error of $\hat{\sigma}_n^2$

1. Assume $x_1, x_2, ..., x_n \overset{iid}{\sim} \mathcal{N}(\mu_0, \sigma_0^2)$
2. Assume that $\hat{\sigma}_n^2$ is close to σ_0^2 and $\bar{x}$ is close to μ_0
3. Then $\hat{\sigma}_n^2(n-1)$ has been shown to have a χ-squared distribution with n degrees of freedom
4. The SD of this sampling distribution is the standard error

Standard errors

What if:

- The sampling distribution is not easy to derive?
- Our distributional assumptions break down?

Standard errors

What if:

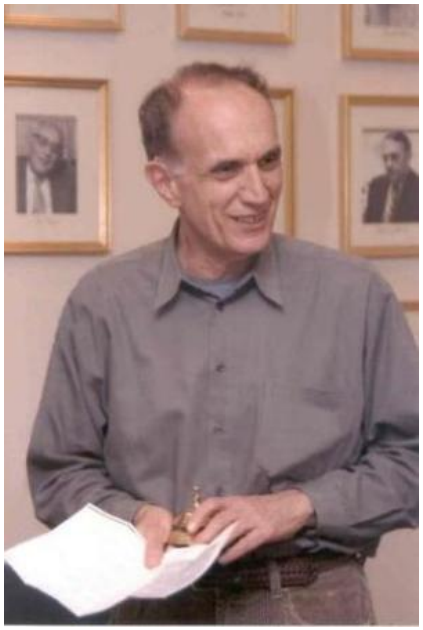
- The sampling distribution is not easy to derive?
- Our distributional assumptions break down?

Some possible options:

1. Bootstrap
2. Jackknife
3. Influence functions
    - Beyond scope of this course

The Bootstrap

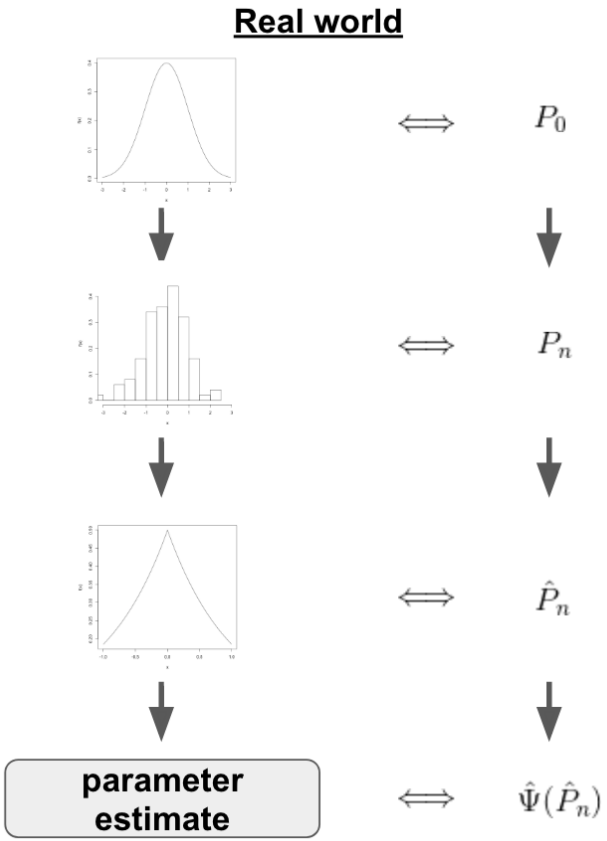
Method to simulate generating from the true distribution P_0



- Provides standard error of estimates
- Popularized by Brad Efron (Stanford)
  - Wrote "An Introduction to the Bootstrap" with Robert Tibshirani
- Very popular among practitioners
- Computer intensive (d/t the approach)

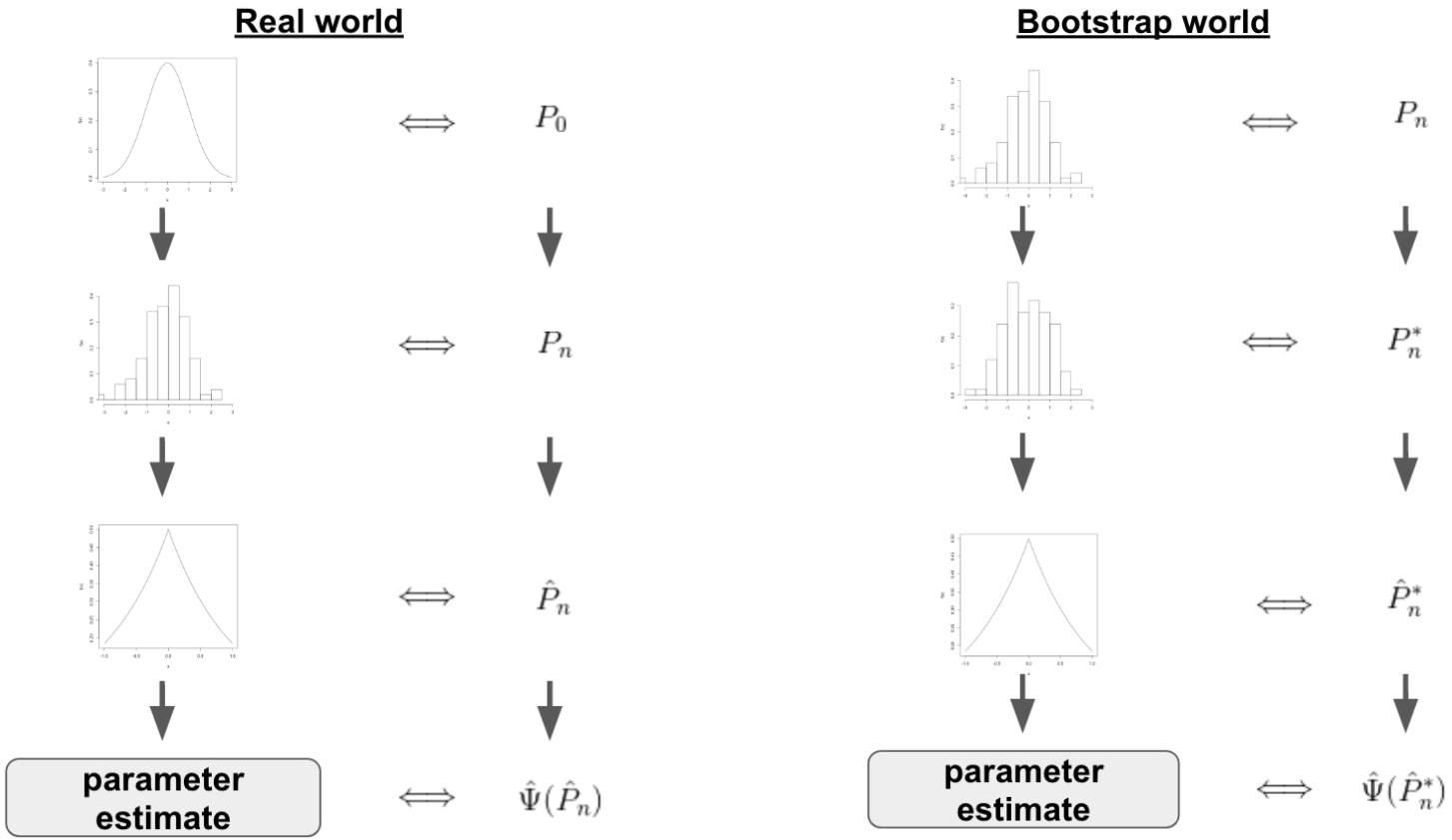
The bootstrap

Method to simulate generating from the true distribution P_0

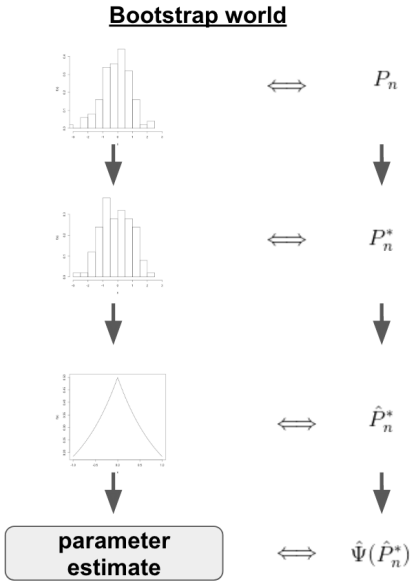


The bootstrap

Method to simulate generating from the true distribution P_0

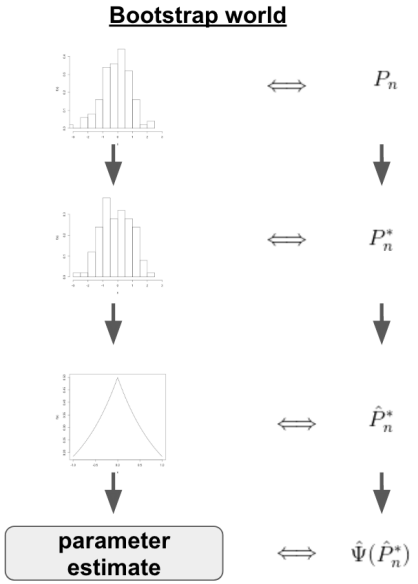


The bootstrap



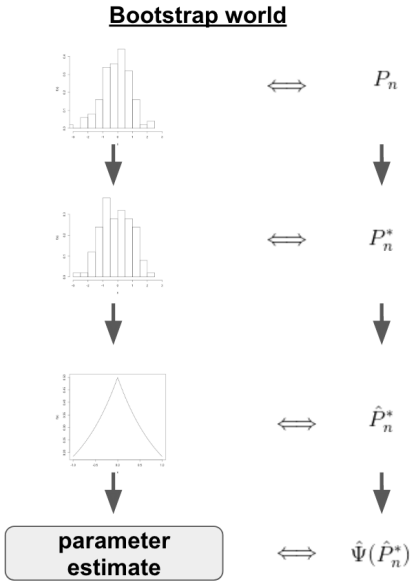
- This resampling method is repeated (say, B times) until we have "*enough*" iterations to get a stable distribution.
  - Results in a simulated sampling distribution

The bootstrap



- This resampling method is repeated (say, B times) until we have "*enough*" iterations to get a stable distribution.
  - Results in a simulated sampling distribution
- The SD of this sampling distribution is our estimated standard error

The bootstrap

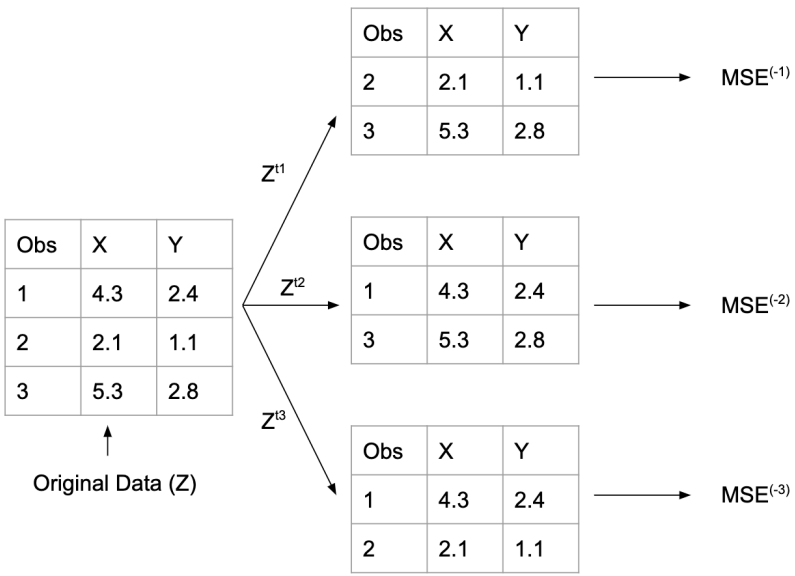


- This resampling method is repeated (say, B times) until we have "*enough*" iterations to get a stable distribution.
  - Results in a simulated sampling distribution
- The SD of this sampling distribution is our estimated standard error
- n.b. Two approximations are made:

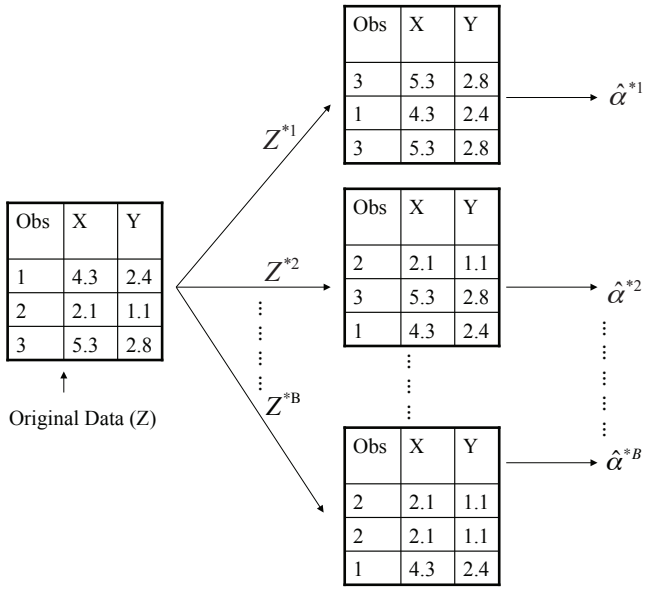
$$SE(\hat{\psi}_n)^2 \overbrace{\approx}^{\text{not so small}} \hat{SE}(\hat{\psi}_n)^2 \overbrace{\approx}^{\text{small}} \hat{SE}_B(\hat{\psi}_n)^2 \quad (2)$$

Bootstrap vs Cross-validation

Cross-validation: provides estimates of the (test) error.



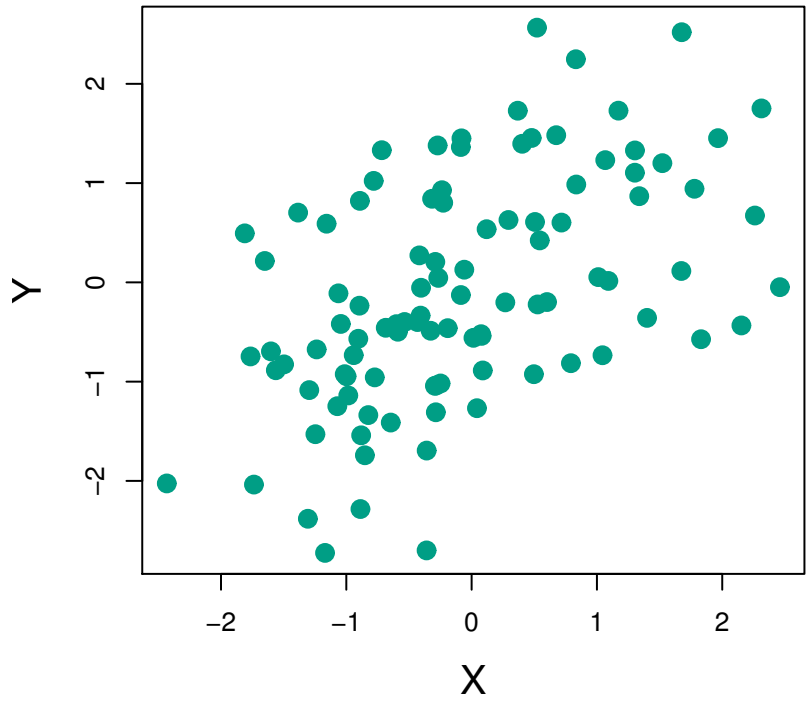
Bootstrap: provides the (standard) error of estimates.



Example. Investing in two assets

Suppose that X and Y are the returns of two assets.

The returns are observed every day, i.e. $(x_1, y_1), ..., (x_n, y_n)$.



Example. Investing in two assets

We only have a fixed amount of money to invest, so we'll invest

- α in X and $(1-\alpha)$ in Y, where α is between 0 and 1, i.e.

$$\alpha X + (1-\alpha)Y \tag{3}$$

Example. Investing in two assets

We only have a fixed amount of money to invest, so we'll invest

- α in X and $(1-\alpha)$ in Y, where α is between 0 and 1, i.e.

$$\alpha X + (1-\alpha)Y \tag{3}$$

Our goal: Minimize the variance of our return as a function of α

- One can show that the optimal α_0 is:

$$\alpha_0 = \frac{\sigma_Y^2 - \sigma_{XY}}{\sigma_X^2 + \sigma_Y^2 - 2\sigma_{XY}} \tag{4}$$

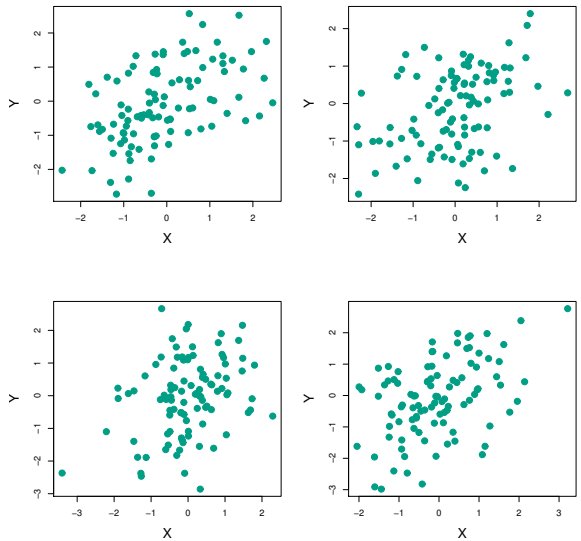
- which we can estimate using our data, i.e.

$$\hat{\alpha}_n = \frac{\hat{\sigma}_{Y,n}^2 - \hat{\sigma}_{XY,n}}{\hat{\sigma}_{X,n}^2 + \hat{\sigma}_{Y,n}^2 - 2\hat{\sigma}_{XY,n}} \tag{5}$$

Example. Investing in two assets

If: we knew P_0, we could just resample the n observations and re-calculate $\hat{\alpha}_n$.

- We could iterate on this until we have enough estimates to form a sampling distribution
- Would then estimate the SE via the SD of the distribution

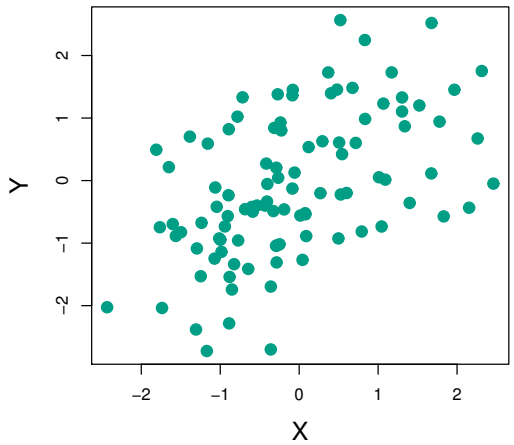


Four draws from P_0.

Example. Investing in two assets

Reality: We don't know P_0 and only have n observations.

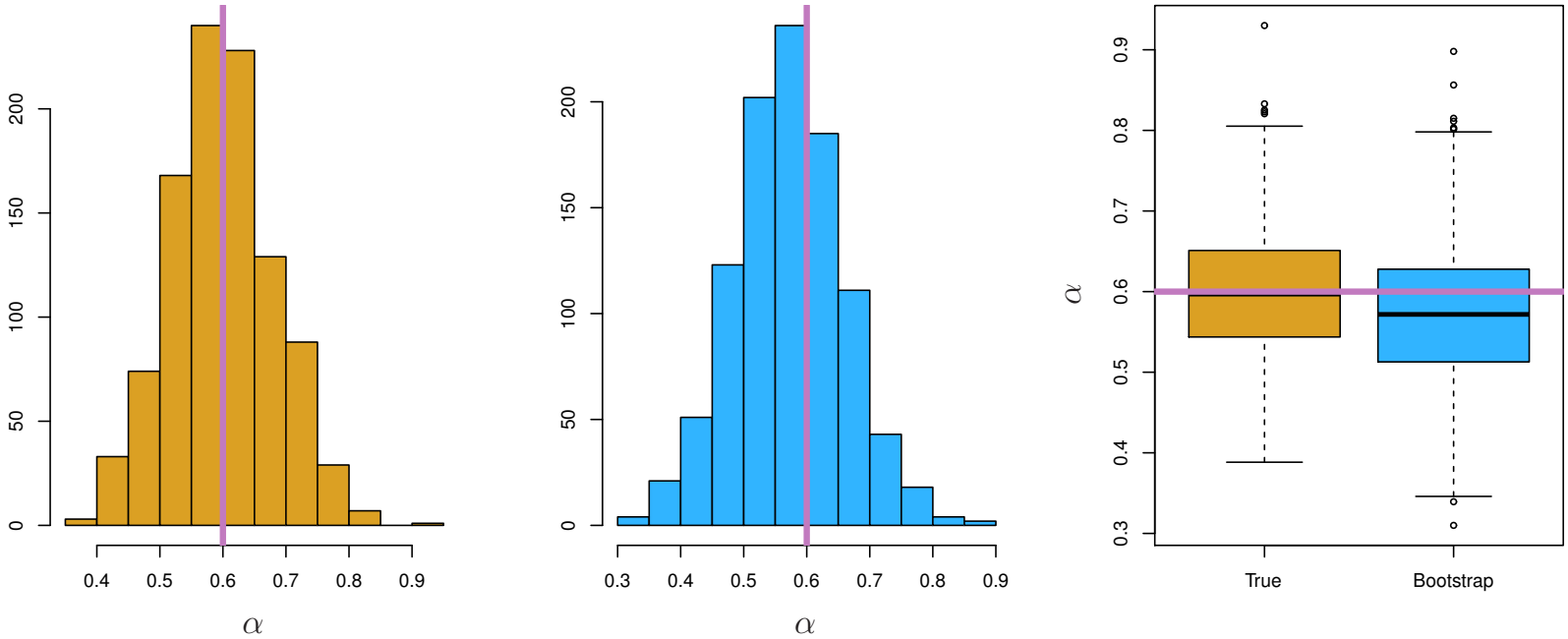
But: We can mimic as if we did know P_0.



- Assume that P_n is a good approximation of P_0
- Iteratively (say, B times):
  - Resample from P_n, i.e. sample from the n observations with replacement, n times (call this $P_n^{*,r}$)
  - Calculate $\hat{\alpha}_n$ from $P_n^{*,r}$ (call this $\hat{\alpha}_n^{*,r}$)
- Calculate the SD of the $\hat{\alpha}_n^{*,r}$ estimates, i.e.

$$\widehat{SE}_B(\hat{\alpha}_n) = \sqrt{\frac{1}{B-1}\sum_{r=1}^{B}\left(\hat{\alpha}_n^{*,r} - \frac{1}{B}\sum_{r'=1}^{B}\hat{\alpha}_n^{*,r'}\right)^2}$$

Bootstrap distribution vs true distribution



True (*left*) and bootstrap (*center*) sampling distributions

Bootstrap and error rates

Each bootstrap iteration will only have about 2/3 of the original data, i.e.

$$\mathbb{P}(x_j \notin P_n^b) = (1 - 1/n)^n \tag{7}$$

Bootstrap and error rates

Each bootstrap iteration will only have about 2/3 of the original data, i.e.

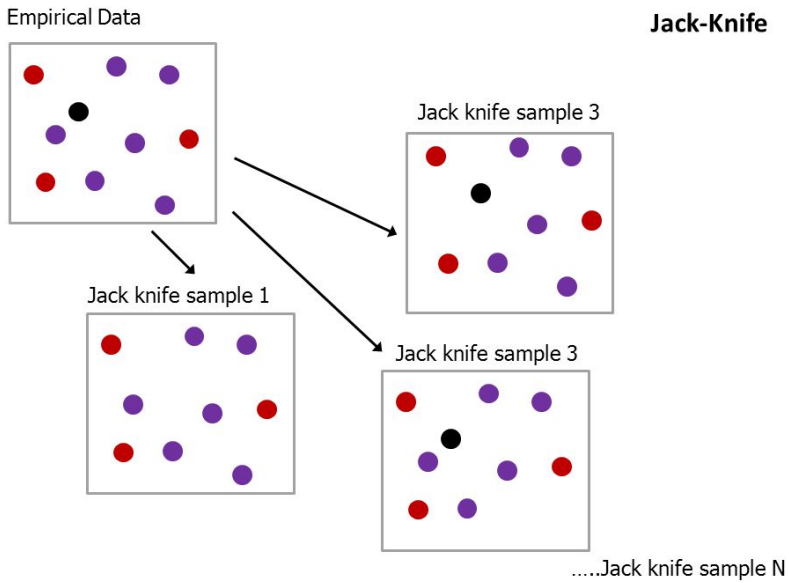
$$\mathbb{P}(x_j \notin P_n^b) = (1 - 1/n)^n \tag{7}$$

We could use the out of bag observations to calculate estimate our test set error, i.e.

$$\widehat{Err} = \frac{1}{n}\sum_{i=1}^{n}\frac{1}{|C^{-i}|}\sum_{b \in C^{-i}} L(y_i, \hat{f}^{*b}(x_i)) \tag{8}$$

- Doing this still encounters 'training-set' bias (i.e. you're using less observations to estimate f_0).

The jackknife



A resampling method (like the Bootstrap), but

- The Bootstrap resamples data from P_n and calculates $\hat{\Psi}(\hat{P}_n^*)$
- The Jackknife leaves out (random) partitions from P_n and calculates $\hat{\Psi}(\hat{P}_n^*)$

Both methods use simulated distributions to calculate SE

The jackknife

The general algorithm (applied to our investment example):

- Assume that P_n is a good approximation of P_0 and choose a number of observations d to delete
  - where $0 < d < n$
- Iteratively:
  - Exclude d observations from our data (resulting in $P_n^{*,d}$)
  - Calculate $\hat{\alpha}_n$ from $P_n^{*,d}$ (call this $\hat{\alpha}_n^{*,d}$)
- Calculate the SD of the $\hat{\alpha}_n^{*,d}$ estimates

If $d > 1$:

$$\widehat{SE}_B(\hat{\alpha}_n) = \sqrt{\frac{n-d}{d\binom{n}{d}} \sum_z \left(\hat{\alpha}_n^{*,z} - \frac{1}{\binom{n}{d}} \sum_{z'} \hat{\alpha}_n^{*,z'} \right)^2} \tag{9}$$

When $d = 1$, this simplifies to:

$$\widehat{SE}_B(\hat{\alpha}_n) = \sqrt{\frac{n-1}{n} \sum_{i=1}^{n} \left(\hat{\alpha}_n^{*,i} - \frac{1}{n} \sum_{i'=1}^{n} \hat{\alpha}_n^{*,i'} \right)^2} \tag{10}$$

Jackknife vs Bootstrap

Some similarities:

- The Jackknife and Bootstrap are asymptotically equivalent
- The theoretical arguments proving the validity of both methods rely on large samples

Jackknife vs Bootstrap

Some similarities:

- The Jackknife and Bootstrap are asymptotically equivalent
- The theoretical arguments proving the validity of both methods rely on large samples

Some differences:

- The jackknife is less computationally expensive
- The jackknife is a linear approximation to the bootstrap
- The jackknife doesn't work well for sample quantiles like the median
- The bootstrap procedure has lots of variations
  - e.g. You can bootstrap the bootstrapped samples to try and get second-order accuracy (aka bootstrap-t)

Review

What we know so far

- In linear regression, adding predictors always decreases the training error or RSS.

$$RSS = (\mathbf{Y} - \mathbf{X}\beta)^\top (\mathbf{Y} - \mathbf{X}\beta) \tag{11}$$

- We can estimate β by minimizing the RSS.

$$\hat{\beta} = (\mathbf{X}^\top \mathbf{X})^{-1} \mathbf{X}^\top y \tag{12}$$

- However, adding predictors does not necessarily improve the test error.
- Selecting significant predictors is hard when n is not much larger than p.

Singular matrices

- When our matrix is not of *full column rank* (e.g. $n < p$), we have that $(\mathbf{X}^\top \mathbf{X})^{-1}$ is not invertible.
- Consequently, there is no least squares solution:

$$\hat{\beta} = \underbrace{(\mathbf{X}^\top \mathbf{X})}_{\text{Singular}}^{-1} \mathbf{X}^\top y \tag{13}$$

  - So, we must find a way around this.

Accounting for singularity

$$\hat{\beta} = \underbrace{(\mathbf{X}^\top \mathbf{X})}_{\text{Singular}}^{-1} \mathbf{X}^\top y \tag{14}$$

Three common approaches for dealing with this:

1. Subset selection
   - Select a subset k of the p predictors ($k \leqslant p$).
   - Use criteria to help select which subset k we want.

Accounting for singularity

$$\hat{\beta} = \underbrace{(\mathbf{X}^\top \mathbf{X})}_{\text{Singular}}^{-1} \mathbf{X}^\top y \tag{14}$$

Three common approaches for dealing with this:

1. Subset selection
   - Select a subset k of the p predictors ($k \leqslant p$).
   - Use criteria to help select which subset k we want.
2. Shrinkage methods
   - Constrain the parameters we're estimating in some way

Accounting for singularity

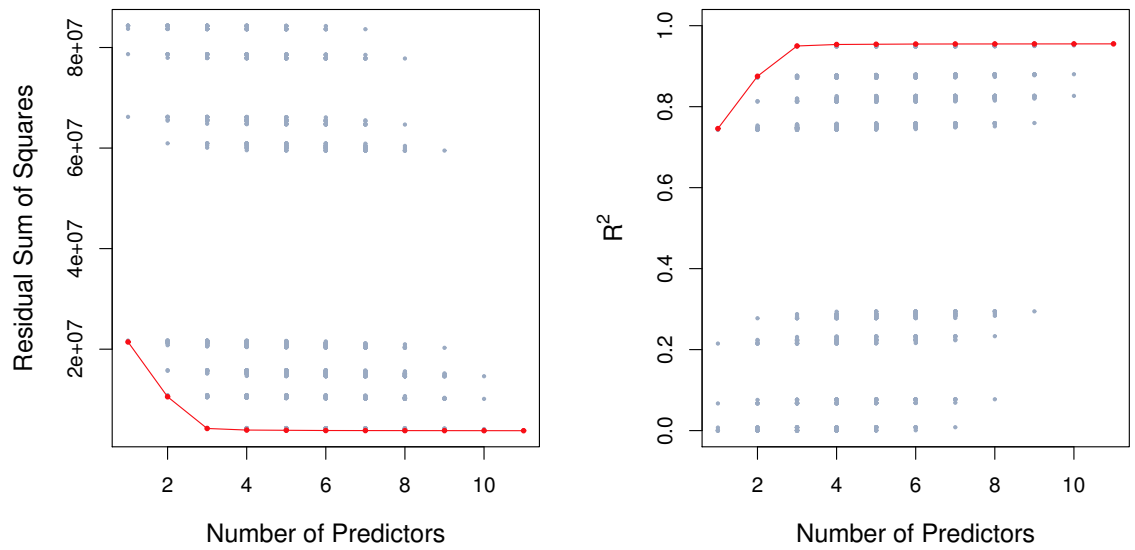
$$\hat{\beta} = \underbrace{(\mathbf{X}^\top \mathbf{X})}_{\text{Singular}}^{-1} \mathbf{X}^\top y \tag{14}$$

Three common approaches for dealing with this:

1. Subset selection
   - Select a subset k of the p predictors ($k \leqslant p$).
   - Use criteria to help select which subset k we want.
2. Shrinkage methods
   - Constrain the parameters we're estimating in some way
3. Dimension reduction
   - Project all our predictors to a smaller dimension space
   - Not covered in this class

Subset selection

- *Simple idea*: Compare all models with k predictors
- **Note**: There are $\binom{p}{k} = p!/(k!(p-k)!)$ possible models
- Choose the model with the smallest RSS
  - Doing this for every possible k:



Note: As expected, the RSS and R^2 improve with higher k.

The optimal k

Two approaches:

1. Use a hold out set (e.g. validation or test set)
    - c.f. Cross-validation
2. Use *modified* metrics that account for the size of k, e.g.
    - Akaike Information Criterion (AIC)
    - Bayesian Information Criterion (BIC)
    - Adjusted R^2

The optimal k

Two approaches:

1. Use a hold out set (e.g. validation or test set)
    - c.f. Cross-validation
2. Use *modified* metrics that account for the size of k, e.g.
    - Akaike Information Criterion (AIC)
    - Bayesian Information Criterion (BIC)
    - Adjusted R^2

How the modified metrics compare to using hold out sets

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

Akaike Information Criterion (AIC)

Similar to Mallow's C_p:

$$C_p = \frac{1}{n}(RSS + 2k\hat{\sigma}^2) \tag{15}$$

- i.e. Adds the penalty $2k\hat{\sigma}^2$ to the RSS
- Can be shown to be unbiased estimate of test set error

But, also normalizes for $\hat{\sigma}^2$:

$$AIC = \frac{1}{n\hat{\sigma}^2}(RSS + 2k\hat{\sigma}^2) = \frac{C_p}{\hat{\sigma}^2} \tag{16}$$

Since the two are proportional, (for least squares models) both are optimized at the same k.

Bayesian Information Criterion (BIC)

Similar to Mallow's C_p, but derived from Bayesian POV:

$$BIC = \frac{1}{n}(RSS + \log(n)k\hat{\sigma}^2) \tag{17}$$

n.b. $\log(n) > 2$ for $n > 7$

- BIC will penalize more for large k (i.e. optimizes for smaller k)

Adjusted R^2

Recall:

$$R^2 = 1 - \frac{RSS}{TSS} \tag{18}$$

The adjusted R^2 penalizes for larger k:

$$R^2_{adj} = 1 - \frac{RSS/(n-d-1)}{TSS/(n-1)} \tag{19}$$

Adjusted R^2

Recall:

$$R^2 = 1 - \frac{RSS}{TSS} \tag{18}$$

The adjusted R^2 penalizes for larger k:

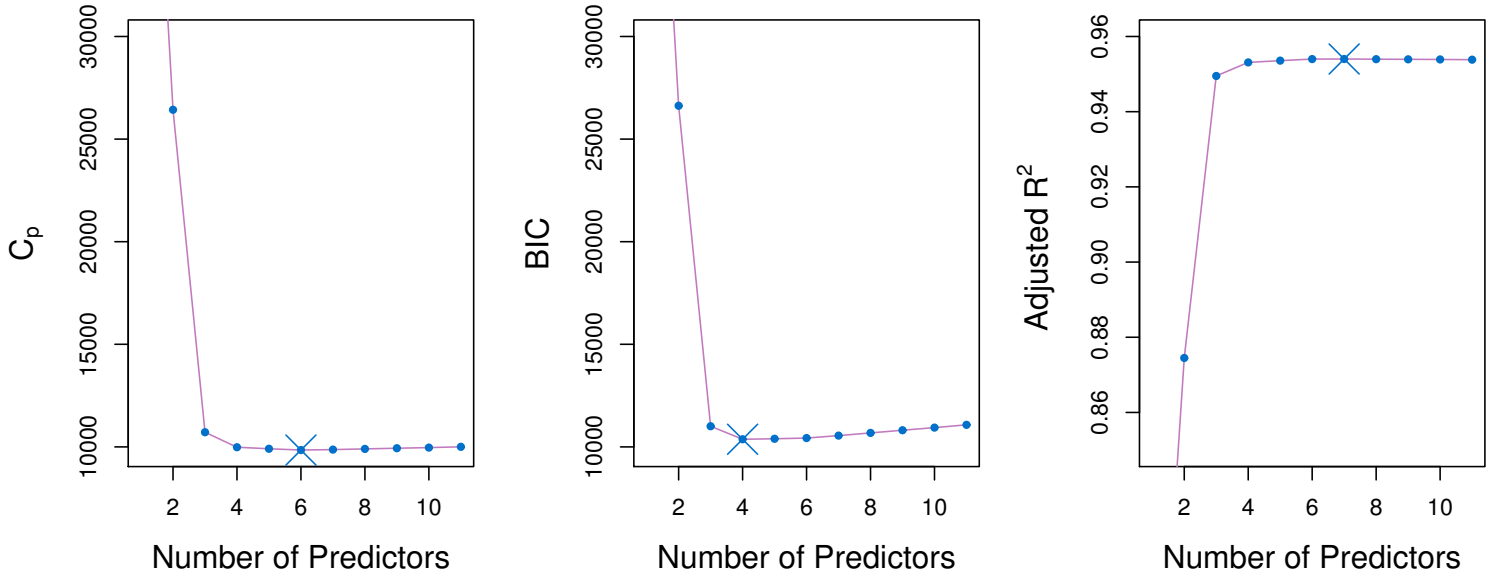
$$R^2_{adj} = 1 - \frac{RSS/(n-d-1)}{TSS/(n-1)} \tag{19}$$

Maximizing R^2_{adj} is equivalent to minimizing $1 - R^2_{adj}$, i.e.:

$$\frac{RSS}{n-d-1} \tag{20}$$

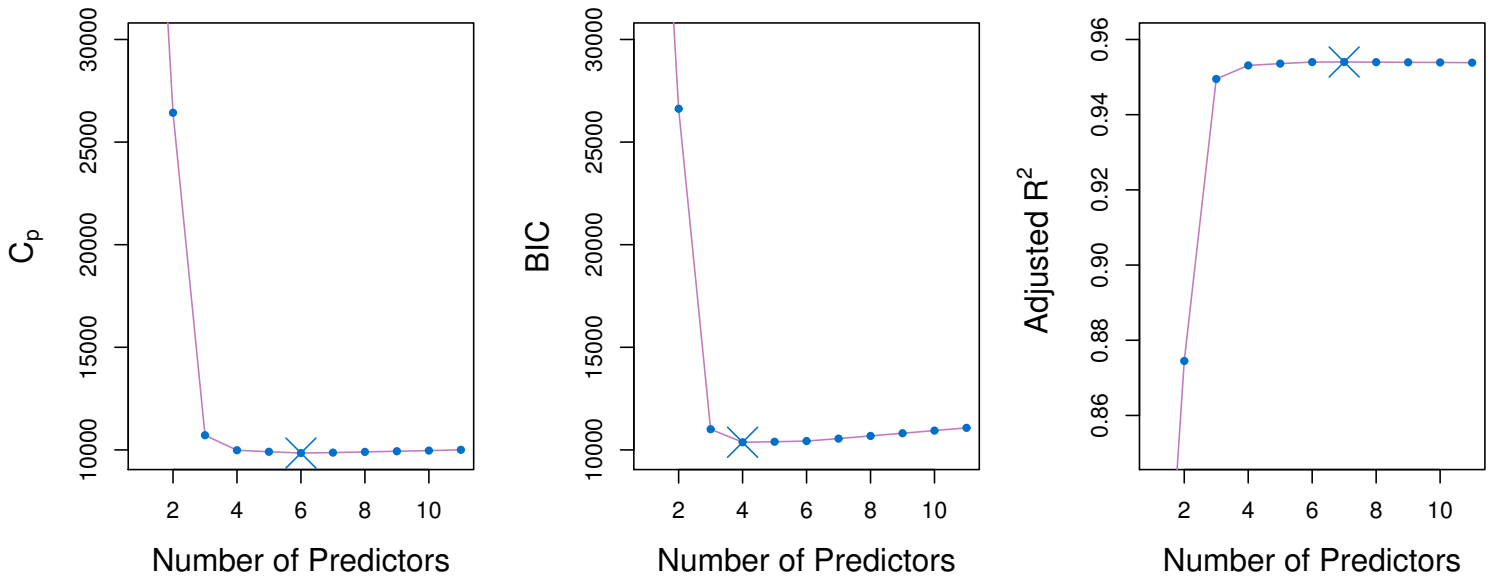
Applied example

Best subset selection for the Credit data set



Applied example

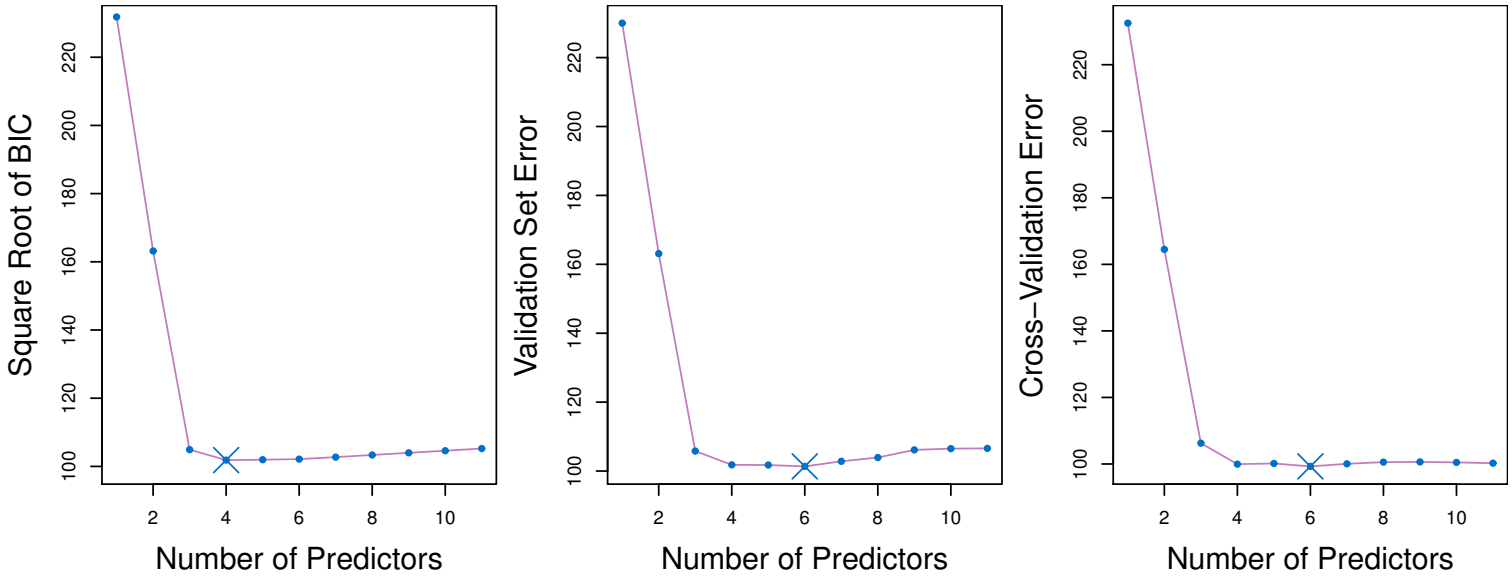
Best subset selection for the Credit data set



n.b. The curve is pretty flat for $k \geq 4$

Applied example

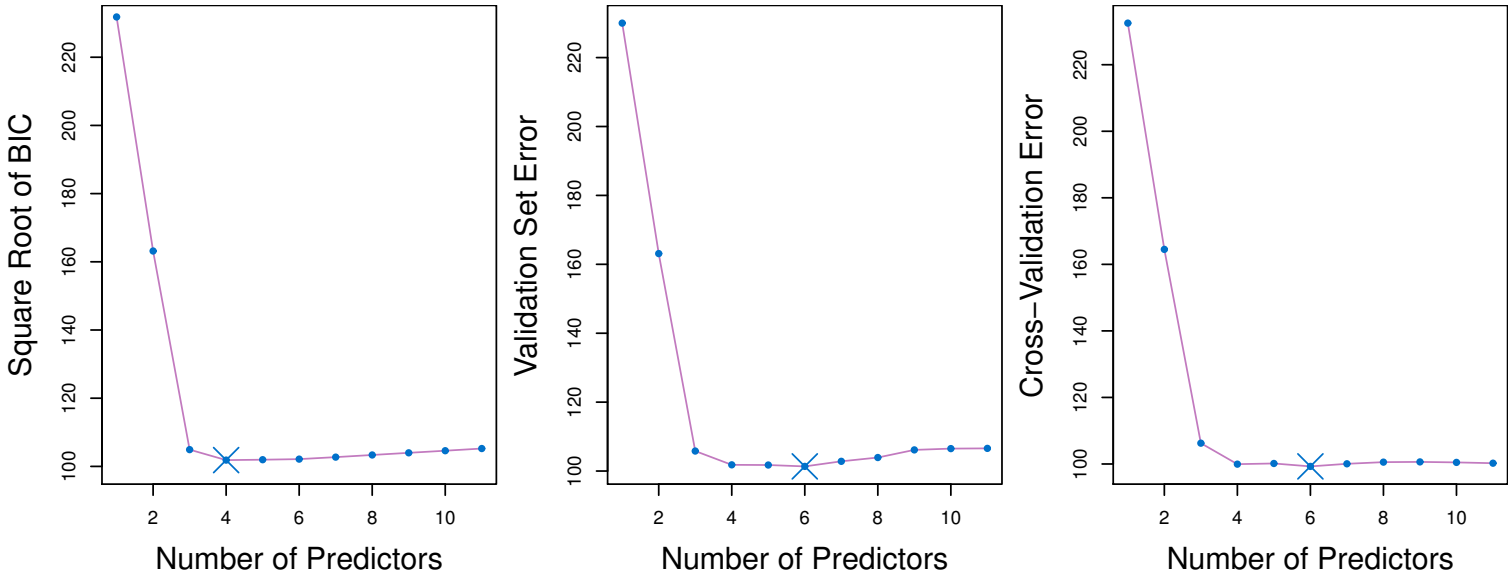
BIC vs validation sets



n.b. The curves are also pretty flat for $k \geq 4$.

Applied example

BIC vs validation sets



n.b. The curves are also pretty flat for $k \geq 4$.

Can use the *one-standard-error rule*

- Choose the parsimonious model (i.e. lowest k) such that the test error is within 1-SE of the lowest point

Stepwise selection methods

Best subset selection has 2 problems:

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

Stepwise selection methods

Best subset selection has 2 problems:

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

One solution: 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 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.

Forward selection vs best subset

# Variables	Best subset	Forward stepwise
One	`rating`	`rating`
Two	`rating`, `income`	`rating`, `income`
Three	`rating`, `income`, `student`	`rating`, `income`, `student`
Four	`cards`, `income` `student`, `limit`	`rating`, `income`, `student`, `limit`

TABLE 6.1. *The first four selected models for best subset selection and forward stepwise selection on the* `Credit` *data set. The first three models are identical but the fourth models differ.*

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.

Forward vs backward selection

- You cannot apply backward selection when $p > n$
  - Though should still have a "reasonable" number of observations
- **Important**: they may not produce the same sequence of models.
  Example: $X_1, X_2 \overset{iid}{\sim} \mathcal{N}(0, \sigma^2)$

$$X_3 = X_1 + 3X_2 \tag{21}$$
$$Y = X_1 + 2X_2 + \epsilon \tag{22}$$

Regressing Y onto X_1, X_2, X_3:

- Forward: $\{X_3\} \to \{X_3, X_2\} \to \{X_3, X_2, X_1\}$
- Backward: $\{X_1, X_2, X_3\} \to \{X_1, X_2\} \to \{X_2\}$

Other stepwise selection methods

- *Mixed stepwise selection*: Do forward selection, but at every step, remove any variables that are no longer "necessary"
  - e.g. using p-values
- *Forward stagewise selection*: Do forward selection, but after every step, modify the remaining predictors such that they are uncorrelated to the selected predictors.
- etc.

Issues with stepwise methods

Important things to keep in mind:

- The selected model is not guaranteed to be optimal
  - There are often several equally good models
- The procedure does not take into account a researcher's knowledge about the predictors
- Outliers can have a large impact on the procedure
- Some predictors should be considered together as a group (e.g. dummy indicators for seasons of the year)
- The coefficients, R^2, p-values, CI's, etc are all biased/invalid
- Should not over-interpret the order that the predictors are included
- Cannot conclude that all variables included are important, or all excluded variables are unimportant

Shrinkage methods

Allows us to use all p predictors, but will regularize (i.e. shrink) their coefficients in some way.

- Common to shrink them towards 0

Shrinkage methods

Allows us to use all p predictors, but will regularize (i.e. shrink) their coefficients in some way.

- Common to shrink them towards 0

Question: Why would shrunk coefficients be better?

- Will introduce bias, but can significantly reduce the variance
  - If the variance is noticeably larger, this decreases the test error
- There are Bayesian motivations to do this: the prior tends to shrink the parameters.

Shrinkage methods

Allows us to use all p predictors, but will regularize (i.e. shrink) their coefficients in some way.

- Common to shrink them towards 0

Question: Why would shrunk coefficients be better?

- Will introduce bias, but can significantly reduce the variance
  - If the variance is noticeably larger, this decreases the test error
- There are Bayesian motivations to do this: the prior tends to shrink the parameters.

Three common shrinkage methods:

1. Ridge regression
2. Lasso regression
3. Elastic net

Ridge regression

Ridge regression solves the following optimization:

$$\min_{\beta} \quad \sum_{i=1}^{n} \left(y_i - \beta_0 - \sum_{j=1}^{p} \beta_j x_{i,j} \right)^2 + \lambda \sum_{j=1}^{p} \beta_j^2 \tag{23}$$

In blue: the model RSS

In red: the squared ℓ_2 norm of β, or $\|\beta\|_2^2$

The parameter $\lambda > 0$ is a tuning parameter. It modulates the importance of fit vs. shrinkage.

- Typically determined via e.g. cross-validation

Ridge regression

Writing our loss function in matrix form

$$(\mathbf{Y} - \mathbf{X}\beta)^\top(\mathbf{Y} - \mathbf{X}\beta) + \lambda\beta^\top\beta \tag{24}$$

it can be shown that

$$\hat{\beta}_{\boldsymbol{n}}^{\boldsymbol{ridge}} = (\mathbf{X}^\top\mathbf{X} + \lambda\mathbf{I_n})^{-\mathbf{1}}\mathbf{X}^\top\mathbf{Y} \tag{25}$$

- So ridge regression simply adds a positive constant to $\mathbf{X}^\top\mathbf{X}$, making it non-singular.

Ridge regression

Under the linear model, the mean and covariance of $\hat{\beta}_n^{ridge}$ are:

$$\begin{aligned}\mathbb{E}[\hat{\beta}_n^{ridge}|\mathbf{X}] &= (\mathbf{X}^\top\mathbf{X} + \lambda\mathbf{I_n})^{-1}\mathbf{X}^\top\mathbb{E}[\mathbf{Y}|\mathbf{X}] \\ &= (\mathbf{X}^\top\mathbf{X} + \lambda\mathbf{I_n})^{-1}\mathbf{X}^\top\mathbf{X}\beta\end{aligned} \tag{26}$$

$$\begin{aligned}Cov[\hat{\beta}_n^{ridge}|\mathbf{X}] &= (\mathbf{X}^\top\mathbf{X} + \lambda\mathbf{I_n})^{-1}\mathbf{X}^\top\mathbf{Cov}[\mathbf{Y}|\mathbf{X}] \\ &\quad X^\top(\mathbf{X}^\top\mathbf{X} + \lambda\mathbf{I_n})^{-1} \\ &= \sigma^2(\mathbf{X}^\top\mathbf{X} + \lambda\mathbf{I_n})^{-1}\mathbf{X}^\top\mathbf{X}(\mathbf{X}^\top\mathbf{X} + \lambda\mathbf{I_n})^{-1}\end{aligned} \tag{27}$$

Scaling predictors

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

$$Y = X_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_p X_p \tag{28}$$

e.g. Multiplying X_1 by c can be compensated by dividing $\hat{\beta}_1$ by c

- i.e. Doing this results in the same RSS

Scaling predictors

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

$$Y = X_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_p X_p \tag{28}$$

e.g. Multiplying X_1 by c can be compensated by dividing $\hat{\beta}_1$ by c

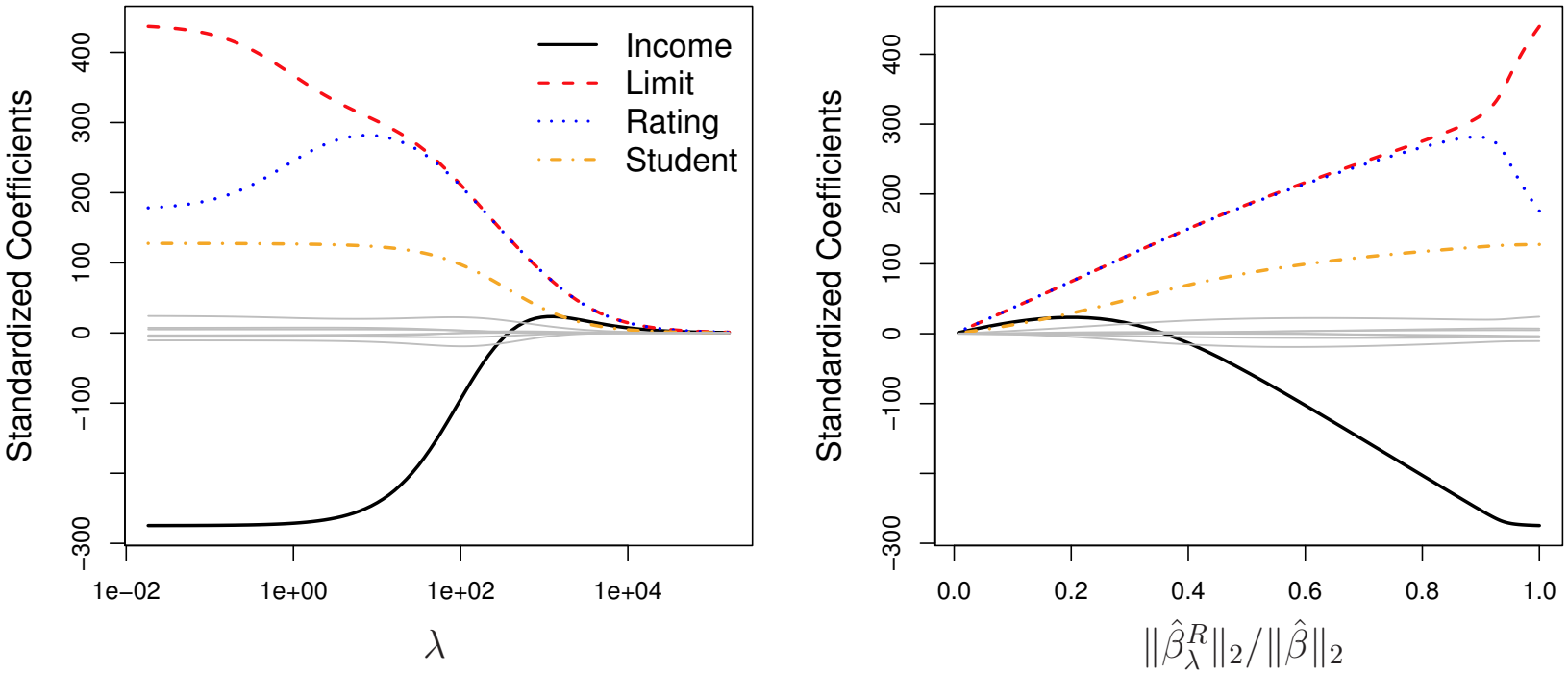
- i.e. Doing this results in the same RSS

This is not true for ridge regression!

- Due to $\|\beta\|_2^2$
- **In practice**: standardize all predictors (i.e. center and scale such that it has sample variance 1)
  - e.g. *glmnet* (by Hastie, Tibshirani, and Friedman)

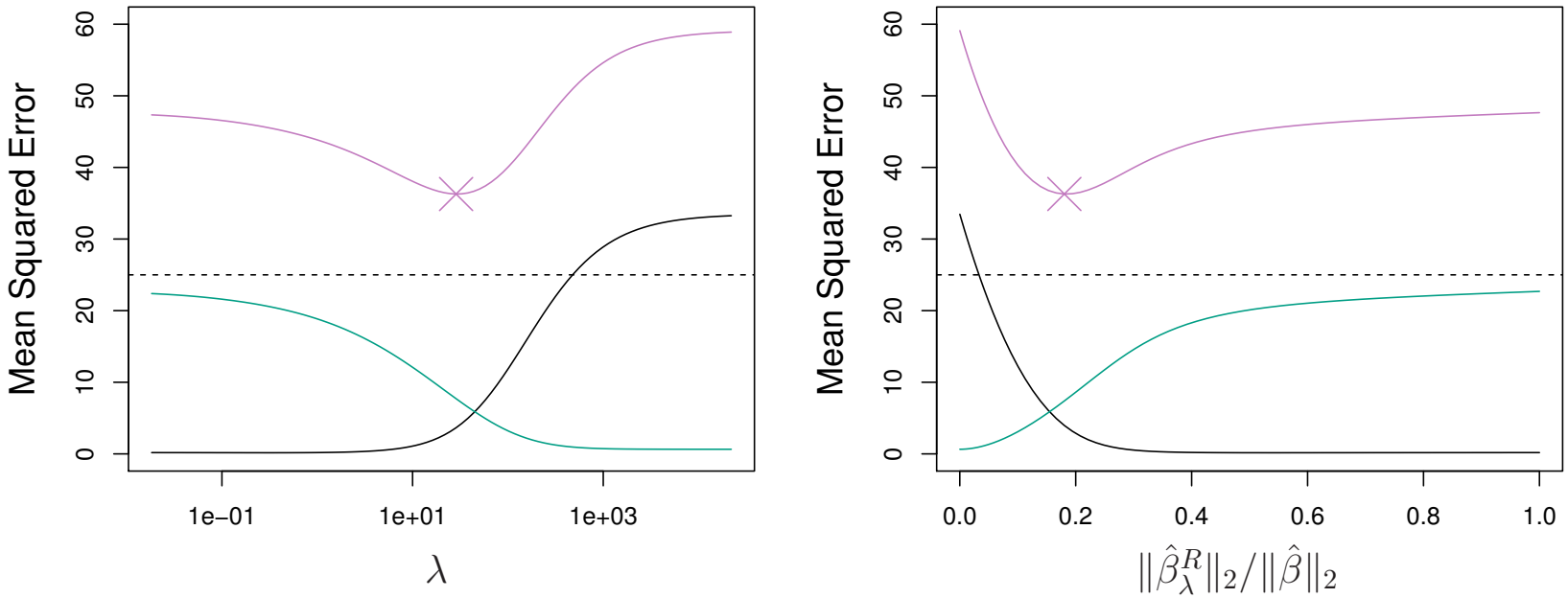
Example: Ridge regression

Ridge regression of `default` in the Credit dataset.



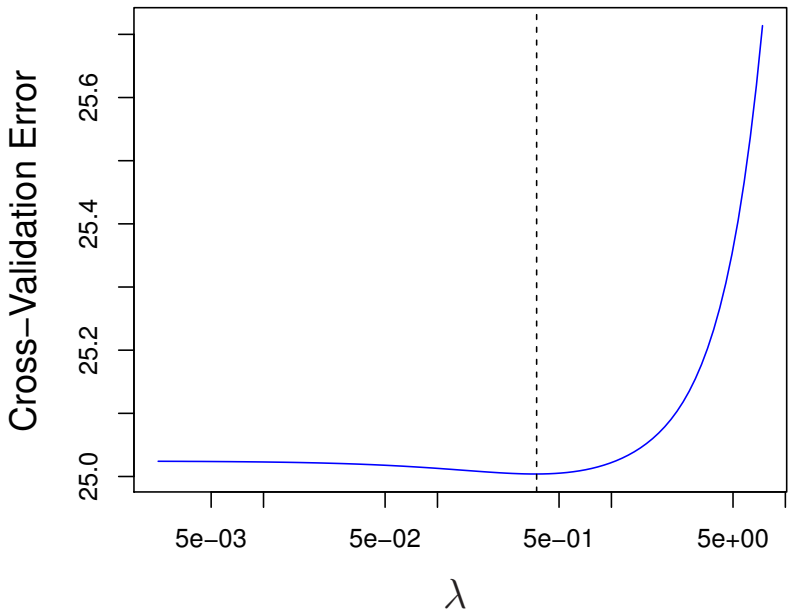
Bias variance trade-off

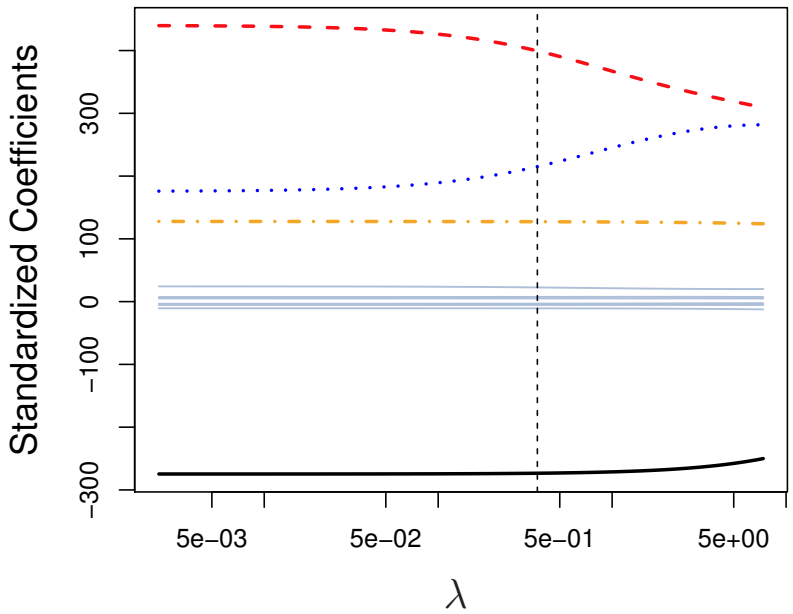
Computing the bias, variance, and test error as a function of λ (in simulation).



Cross validation would yield an estimate of the test error.

Selecting λ by cross-validation





The Lasso

The **L**east **A**bsolute **S**hrinkage and **S**election **O**perator regression solves the following optimization:

$$\min_{\beta} \quad \sum_{i=1}^{n} \left(y_i - \beta_0 - \sum_{j=1}^{p} \beta_j x_{i,j} \right)^2 + \lambda \sum_{j=1}^{p} |\beta_j| \tag{29}$$

In blue: the model RSS

In red: the ℓ_1 norm of β, or $\|\beta\|_1$

The Lasso

The **L**east **A**bsolute **S**hrinkage and **S**election **O**perator regression solves the following optimization:

$$\min_{\beta} \quad \sum_{i=1}^{n} \left(y_i - \beta_0 - \sum_{j=1}^{p} \beta_j x_{i,j} \right)^2 + \lambda \sum_{j=1}^{p} |\beta_j| \tag{29}$$

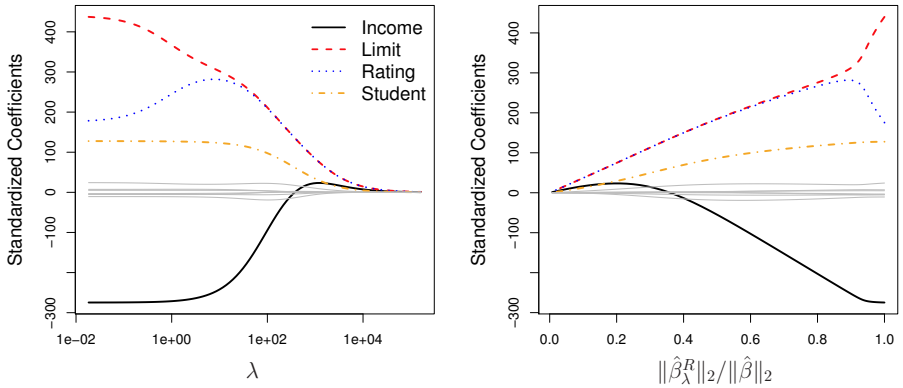
In blue: the model RSS

In red: the ℓ_1 norm of β, or $\|\beta\|_1$ **Note**: Unlike ridge regression, LASSO does not have a closed form solution.

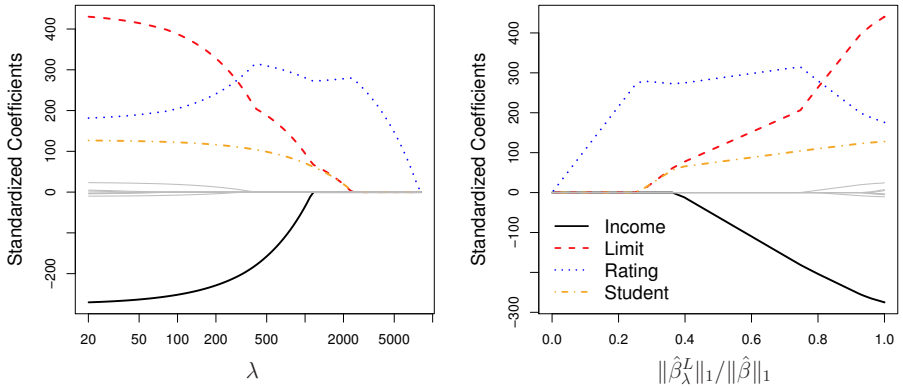
Why would we use the Lasso instead of Ridge regression?

- Ridge regression shrinks all the coefficients to a non-zero value
- The Lasso shrinks some of the coefficients all the way to zero.
  - Similar to subset selection: will select variables for you

Ridge regression of `default` in the Credit dataset.



Lasso regression of `default` in the Credit dataset.



An alternative formulation for regularization

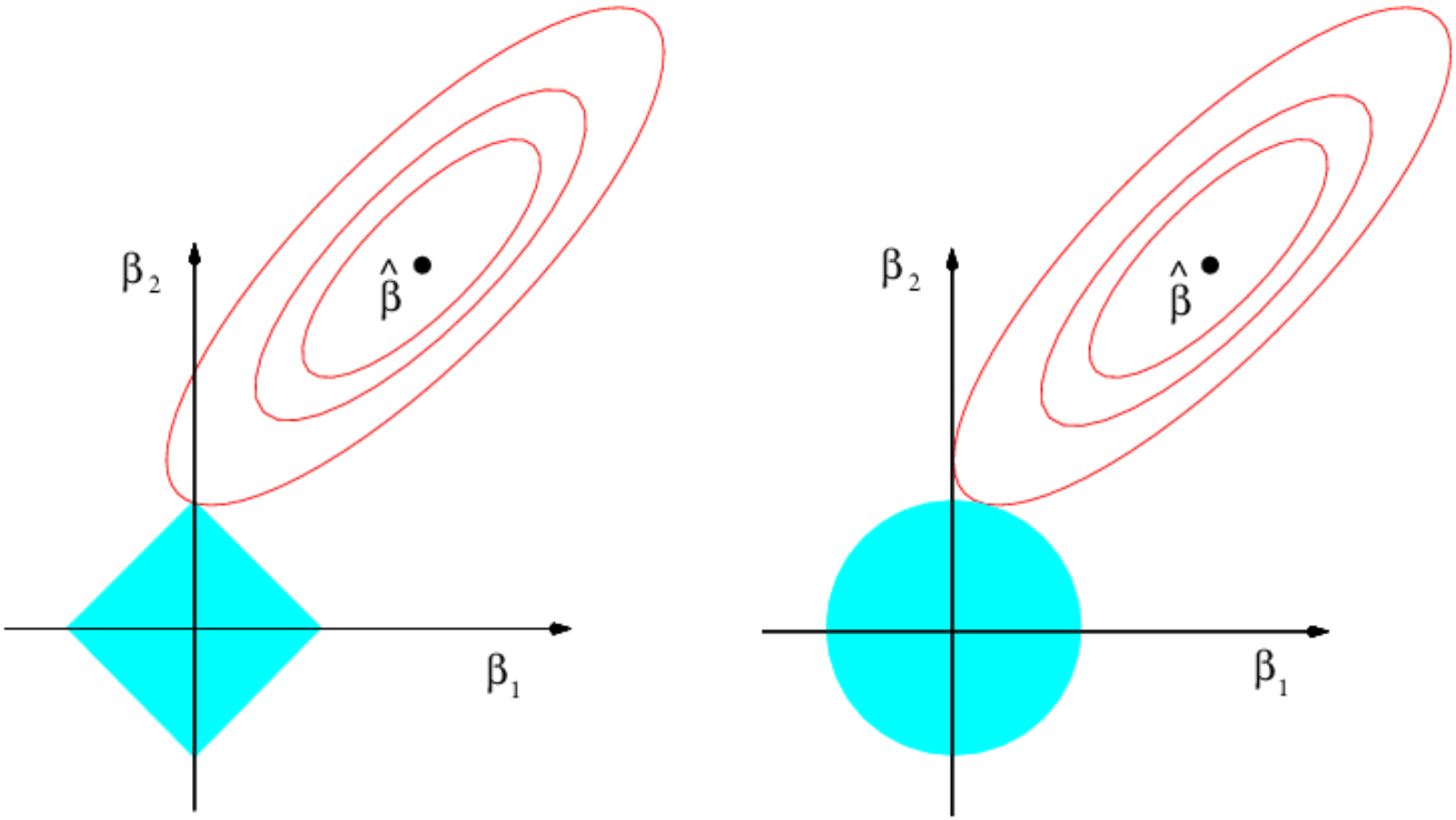
- **Ridge:** for every λ, there is an s such that $\hat{\beta}^R_\lambda$ solves:

$$\min_\beta \left\{ \sum_{i=1}^{n} \left(y_i - \beta_0 - \sum_{j=1}^{p} \beta_j x_{i,j} \right)^2 \right\} \text{ subject to } \sum_{j=1}^{p} \beta_j^2 < s \quad (30)$$

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

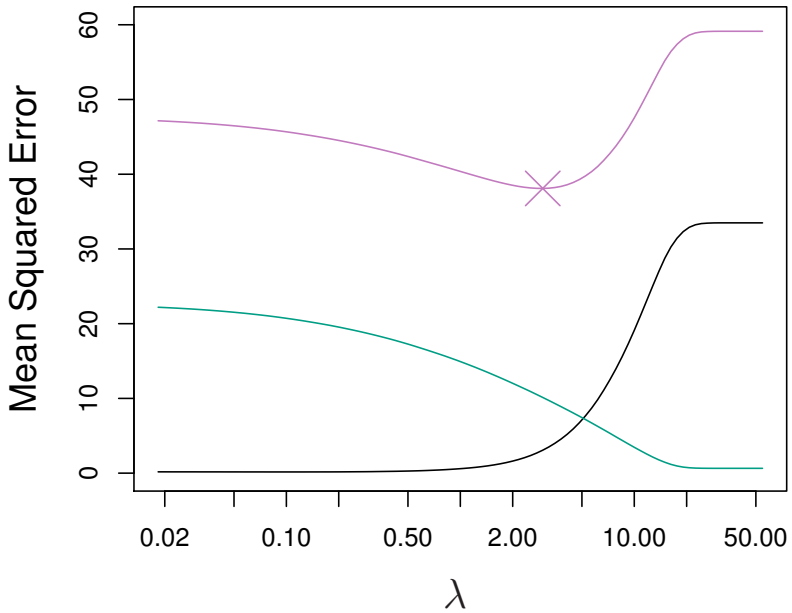
$$\min_\beta \left\{ \sum_{i=1}^{n} \left(y_i - \beta_0 - \sum_{j=1}^{p} \beta_j x_{i,j} \right)^2 \right\} \text{ subject to } \sum_{j=1}^{p} |\beta_j| < s \quad (31)$$

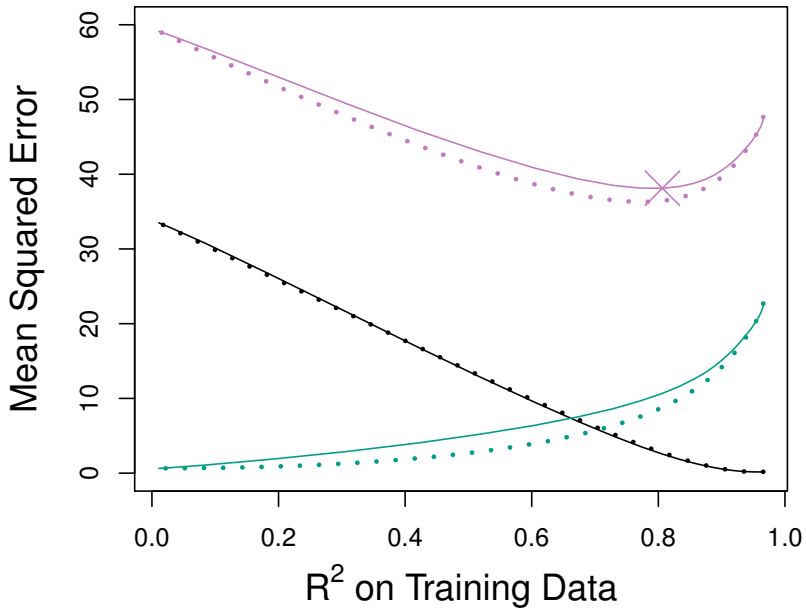
An alternative formulation for regularization



When is the Lasso better than Ridge?

Example 1. Most of the coefficients are non-zero.

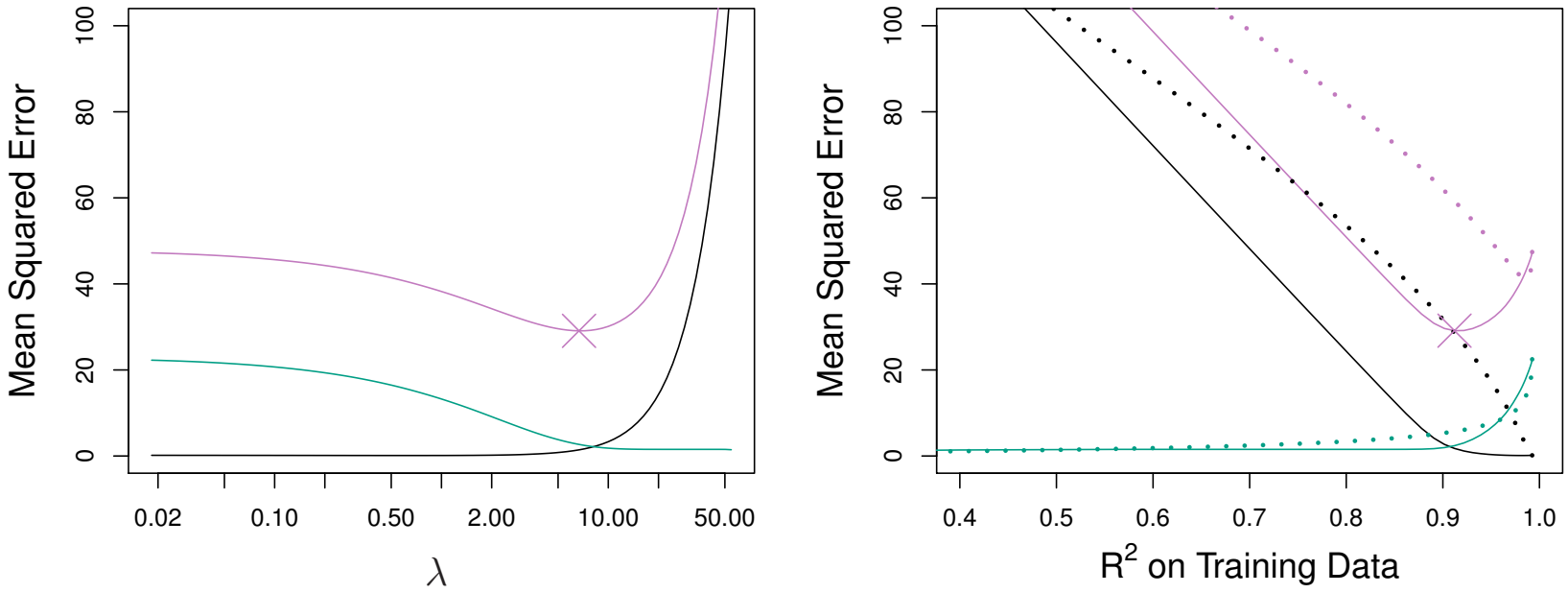




- Bias, Variance, MSE. The Lasso (—), Ridge ($\cdots$).
- The bias is about the same for both methods.
- The variance of Ridge regression is smaller, so is the MSE.

When is the Lasso better than Ridge?

Example 2. Only 2 coefficients are non-zero.



- Bias, Variance, MSE. The Lasso (—), Ridge ($\cdots$).
- The bias, variance, and MSE are lower for the Lasso.

Elastic Net

Combines $\|\beta\|_2^2$ (ridge) and $\|\beta\|_1$ (lasso) penalties.

Elastic net solves the following optimization:

$$\min_{\beta} \quad \sum_{i=1}^{n} \left(y_i - \beta_0 - \sum_{j=1}^{p} \beta_j x_{i,j} \right)^2 + \lambda_2 \sum_{j=1}^{p} \beta_j^2 + \lambda_1 \sum_{j=1}^{p} |\beta_j| \quad (32)$$

In blue: the model RSS
In red: both $\|\beta\|_2^2$ and $\|\beta\|_1$

This provides a nice trade off between sparsity and grouping.

Elastic Net

Combines $\|\beta\|_2^2$ (ridge) and $\|\beta\|_1$ (lasso) penalties.

Elastic net solves the following optimization:

$$\min_{\beta} \quad \sum_{i=1}^{n} \left(y_i - \beta_0 - \sum_{j=1}^{p} \beta_j x_{i,j} \right)^2 + \lambda_2 \sum_{j=1}^{p} \beta_j^2 + \lambda_1 \sum_{j=1}^{p} |\beta_j| \quad (32)$$

In blue: the model RSS
In red: both $\|\beta\|_2^2$ and $\|\beta\|_1$

This provides a nice trade off between sparsity and grouping.

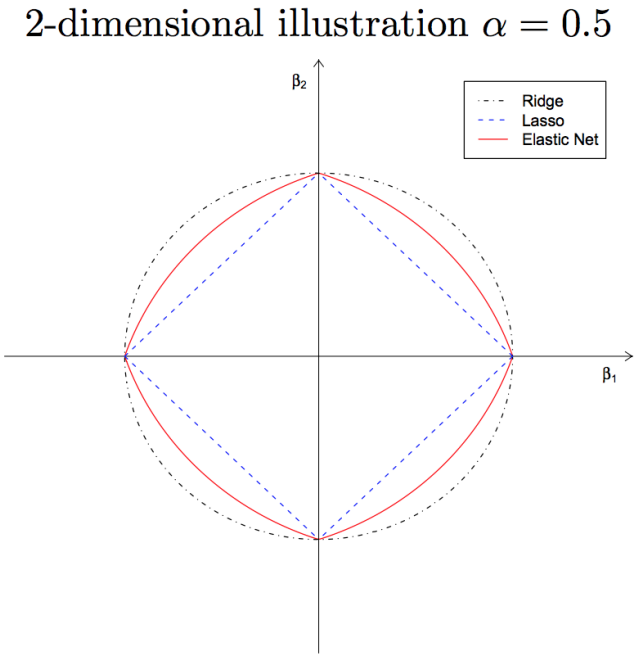
Typically, we define $\alpha = \frac{\lambda_2}{\lambda_2 + \lambda_1}$ and instead optimize:

$$\min_{\beta} \quad \sum_{i=1}^{n} \left(y_i - \beta_0 - \sum_{j=1}^{p} \beta_j x_{i,j} \right)^2 + \alpha \sum_{j=1}^{p} \beta_j^2 + (1-\alpha) \sum_{j=1}^{p} |\beta_j| \quad (33)$$

An alternative formulation for regularization

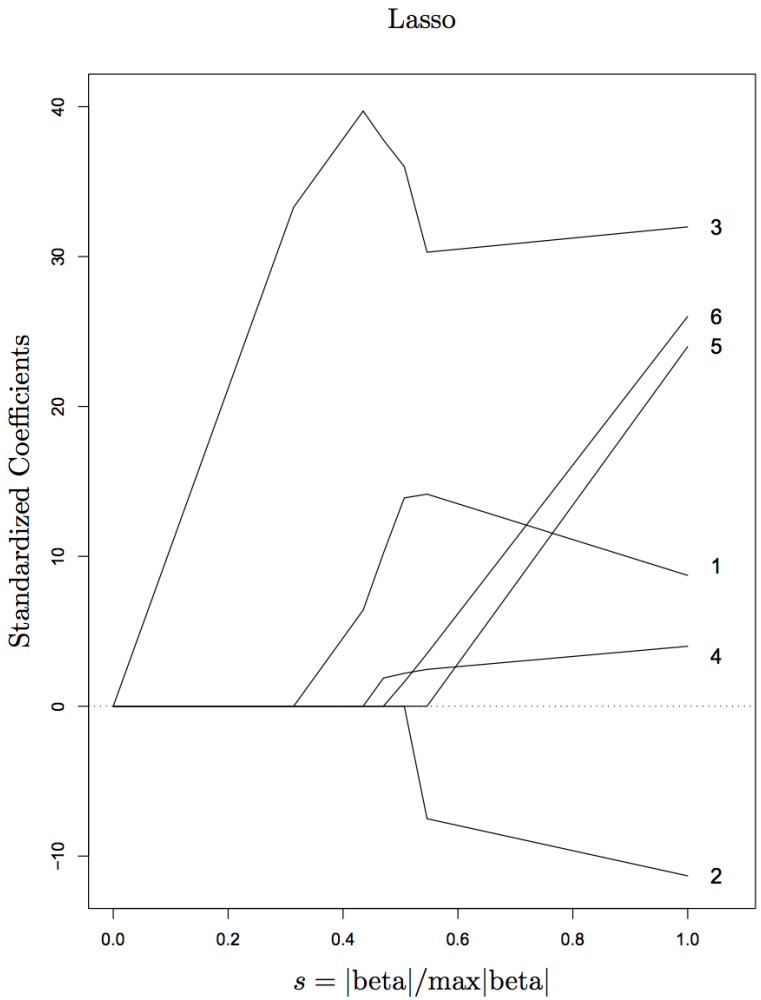
Elastic net:

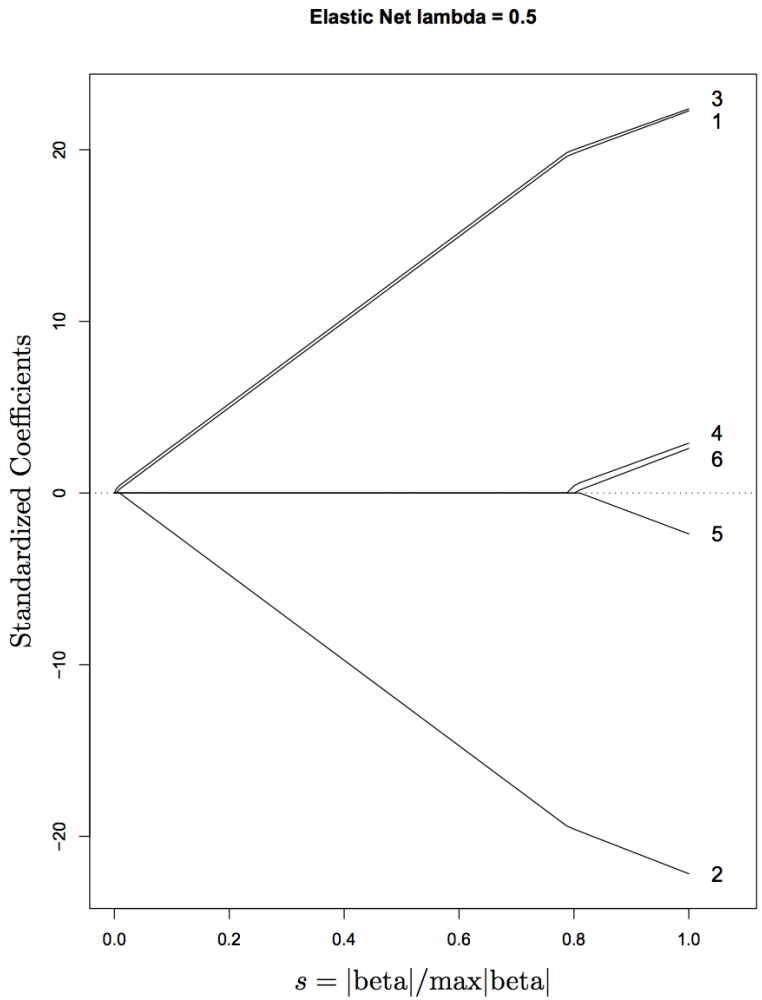
$$\min_{\beta} \left\{ \sum_{i=1}^{n} \left(y_i - \beta_0 - \sum_{j=1}^{p} \beta_j x_{i,j} \right)^2 \right\} \text{ s.t. } \alpha\|\beta\|_2^2 + (1-\alpha)\|\beta\|_1 < s \quad (34)$$



- Singularities at the vertexes (to encourage sparsity)
- Strict convex edges (to encourage grouping)
  - The strength of convexity varies with α

Example: Elastic net





Shrinkage summaries

$$\min_{\beta} \quad \sum_{i=1}^{n} \left(y_i - \beta_0 - \sum_{j=1}^{p} \beta_j x_{i,j} \right)^2 + \lambda_2 \sum_{j=1}^{p} \beta_j^2 + \lambda_1 \sum_{j=1}^{p} |\beta_j| \quad (35)$$

Method	**Shrinkage parameters**
OLS	$\lambda_1 = \lambda_2 = 0$
Ridge	$\lambda_1 = 0, \lambda_2 > 0$
LASSO	$\lambda_1 > 0, \lambda_2 = 0$
Elastic net	$\lambda_1 > 0, \lambda_2 > 0$
$\hat{\beta}_n = 0$	$\lambda_1 = \infty$ or $\lambda_2 = \infty$

Things to consider

- If desired, we could instead consider L_q penalties for values other than 0, 1, and 2 (e.g. $q \in (1, 2)$ or $q > 2$).
- Regularization methods such as the elastic net have been extended to generalized linear models (GLM) as well.
- L_1 and L_2 penalties are also used in contexts other than linear models (e.g. neural networks).
- As usual, we are faced with the bias-variance tradeoff when choosing our shrinkage parameters, λ_1 and λ_2.
- Other regularized methods are also available, e.g.
  - Non-negative Garotte Regression
  - Least Angle Regression
  - Best subset

Degrees of freedom

Degrees of freedom give us a measure of our model's complexity, i.e. the number of free parameters to fit on our data.

- For OLS, the degrees of freedom is equal to $p + 1$.
- In regularized regression, our parameters are estimated in a restricted manner, controlled by λ_1 and λ_2.
  - Effectively reduced the degrees of freedom in our model
- We can still compare across models using an *effective degrees of freedom*:

$$df(y, \hat{y}) = \frac{1}{\sigma^2} \sum_{i=1}^{n} Cov[y_i, \hat{y}_i | x_i] \tag{36}$$

- In the case of OLS, this can be shown to reduce to the ''standard'' degrees of freedom, i.e. $p + 1$.

References

[1] ISL. Chapters 6.

[2] ESL. Chapter 18.