05 - Regression and Regularization

ml4econ, HUJI 2020

Itamar Caspi

April 5, 2020 (updated: 2020-04-06)

Packages and setup

Use the pacman package that automatically loads and installs packages:

```
if (!require("pacman")) install.packages("pacman")

pacman::p_load(
  tidyverse,   # for data wrangling and visualization
  knitr,       # for displaying nice tables
  broom,       # for tidying estimation output
  here,        # for referencing folders and files
  glmnet,      # for estimating lasso and ridge
  gamlr        # for forward stepwise selection
)
```

Set a theme for `ggplot` (Relevant only for the presentation)

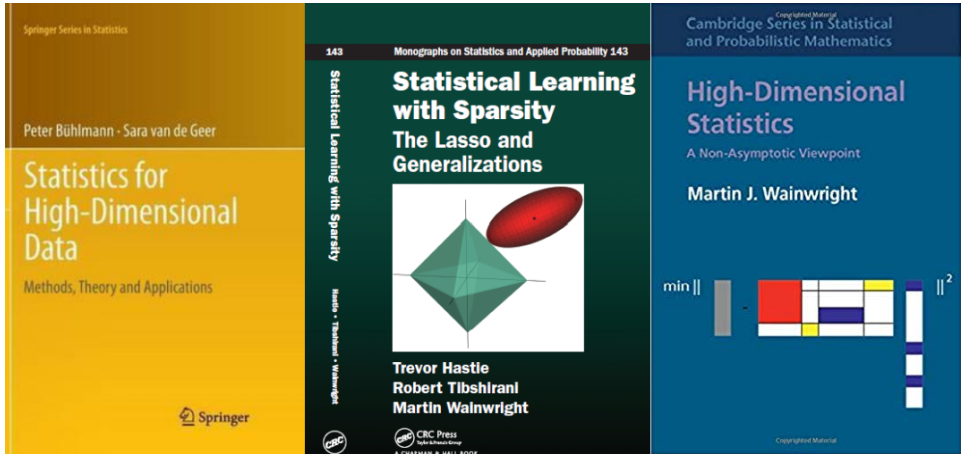
```
theme_set(theme_grey(20))
```

And set a seed

```
set.seed(1203)
```

Resources on hige-dimensional statsitics

- *Statistical Learning with Sparsity - The Lasso and Generalizations* (Hastie, Tibshirani, and Wainwright), (PDF available online)
- *Statistics for High-Dimensional Data - Methods, Theory and Applications* (Buhlmann and van de Geer)
- *High Dimensional Statistics - A Non-Asymptotic Viewpoint* (Wainwright)



Outline

- Linear regression
- Subset selection
- Shrinkage
- Dimension Reduction
- Hands-on

Linear Regression

Econometrics

In econometrics, we typically assume a "true" linear data generating process (DGP):

$$y_i = \beta_0 + \sum_{j=1}^{p} x_{ij}\beta_j + \varepsilon_i$$

where y_i is the outcome variable, $x_{ij}, \ldots, x_{ip}$ is a set of explanatory or control variables (+ interactions, polynomials, etc.), and ε_i is the regression error.

sample : $\{(x_1, \ldots, x_p, y_i)\}_{i=1}^{n}$

Estimation

Ordinary least squares minimizes

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

The emphasis here is on in-sample fit (minimize residual sum of squares).

Typical workflow:

- impose identifying assumptions (causal claims).
- estimate $\beta_0, \ldots, \beta_p$ using the entire sample.
- assume a random sample from a larger population.
- hypothesis testing.

Supervised Learning

Consider the following linear data generating process (DGP):

$$y_i = \beta_0 + \sum_{j=1}^{p} x_{ij}\beta_j + \varepsilon_i$$

where y_i is the predicted (response) variable, $x_{i1}, \ldots, x_{ip}$ is a set of "features", and ε_i is the irreducible error.

- **Training set** : $\{(x_{1i}, \ldots, x_{ip}, y_i)\}_{i=1}^{n}$
- **Test set** : $\{(x_{1i}, \ldots, x_{ip}, y_i)\}_{i=n+1}^{m}$

> Typical assumptions: (1) independent observations; (2) stable DGP across training *and* test sets.

Our object of interest is $\hat{y}_i$, predicting unseen data.

Dffierent objective, different approach

To illustrate how these two approaches (estimation vs. prediction) differ, consider the following data generating process[1]:

$$y_i = \beta_0 + \beta_1 x_i + \varepsilon_i, \qquad \varepsilon_i \sim N(0, \sigma^2)$$

where $\beta_0 = 0$ and $\beta_1 = 2$.

Next, suppose you get a sample of size N and use it to estimate the model and get (standard errors in parentheses):

$$\underset{\quad}{y_i} = \underset{(0.2)}{0.0} + \underset{(10.0)}{2.0} \times x_i + \hat{\varepsilon}_i$$

Given a new unseen x^0 and the output above, how would you predict y^0?

[1] Adapted from Susan Athey's lecture.

Standard errors and prediction accuracy

- OLS is lexicographic in the sense that it first ensure unbiasedness, then efficiency.
- OLS is unbiased, hence you would be right *on average*, but. Is that what we want?
- If your estimated coefficient is noisy (high std. error), so will be your prediction (high variability).
- Bias-variance trade off again.
- Prediction is about balancing bias and variance.

NOTE: in multivariate regression, things are more complicated due to the correlation structure of the x's.

Illustration: Browse data

In this lecture, we will use Matt Taddy's **browser dataset** (available in our repo) which contains web browsing logs for 10,000 people. Taddy has extracted a year's worth of browser logs for the 1000 most heavily trafficked websites. Each browser in the sample spent at least $1 online in the same year.

The goal of estimation: predict spending from browser history:

$$\log(spend_i) = \beta_0 + \beta' visits_i + \varepsilon_i, \qquad \text{for } i = 1, \ldots, n$$

where $visits_i$ is a vector site-visit percentage. This model can be used to segment expected user budget as a function of browser history.

Load data

In this lecture we will only use a sample from the browser dataset: 250 websites and 1000 users.

```
load(here("05-regression-regularization/data","Xweb.Rdata")) # log spend_i
load(here("05-regression-regularization/data","Yweb.Rdata")) # visits_i
```

Log spending by the first user and the fraction of times she spent on the first 4 websites:

```
Yweb[1,]
```

```
## log_spend
##   6.57647
```

```
Xweb[1, 1:4]
```

```
## americansingles.com             opm.gov    netzerovoice.com
##                   0                   0                   0
##  mobilesidewalk.com
##                   0
```

Results

Estimate the model using `lm()`

```
fit_lm <- lm(Yweb ~ Xweb)
```

Show estimation output, sorted by p-values

```
fit_lm %>%
  tidy() %>%
  arrange(p.value) %>%
  mutate(term = map_chr(term, ~ str_remove(., "Xweb"))) %>%
  head(5) %>%
  kable(format = "html", digits = 2)
```

term	estimate	std.error	statistic	p.value
(Intercept)	6.50	0.28	23.56	0.00
qvc.com	0.39	0.13	3.05	0.00
yahoo.net	3.13	1.04	3.02	0.00
gator.com	-0.21	0.07	-2.84	0.00
connextra.com	-0.53	0.19	-2.79	0.01

Model perfoemance

What is the training-set MSE[1]?

```
fit_lm %>%
  glance() %>%
  mutate(MSE = deviance / 1000) %>%
  select(MSE) %>%
  kable(format = "html", digits = 3)
```

MSE
1.989

This is clearly an *underestimate* of the test set MSE.

[1] In the case of a linear regression model, deviance is defined as the sum of squares.

Penalized linear regression

Estimation

Penalized (or *regularized*) sum of squares minimizes solves

$$\min_{\beta_0,\beta} \sum_{i=1}^{N} \left(y_i - \beta_0 - \sum_{j=1}^{p} x_{ij}\beta_j \right)^2 \text{ subject to } R(\beta) \leq t$$

where $R(\cdot)$ is a penalty function that measures **expressiveness** of the model.

As the number of features grow, linear models become *more* expressive.

Notation: Norms

Suppose $\boldsymbol{\beta}$ is a $k \times 1$ vector with typical element β_i.

- The ℓ_0-norm is defined as $||\boldsymbol{\beta}||_0 = \sum_{j=1}^{k} \mathbf{1}_{\{\beta_j \neq 0\}}$, i.e., the number of non-zero elements in $\boldsymbol{\beta}$.
- The ℓ_1-norm is defined as $||\boldsymbol{\beta}||_1 = \sum_{j=1}^{k} |\beta_j|$.
- The ℓ_2-norm is defined as $||\boldsymbol{\beta}||_2 = \left(\sum_{j=1}^{k} |\beta_j|^2 \right)^{\frac{1}{2}}$, i.e., Euclidean norm.

Commonly used penaltiy functions

It is often convenient to rewrite the regularization problem in the Lagrangian form:

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

NOTE: There is one-to-one correspondence between λ and t.

Method	$R(\boldsymbol{\beta})$
OLS	0
Subset selection	$\|\boldsymbol{\beta}\|_0$
Lasso	$\|\boldsymbol{\beta}\|_1$
Ridge	$\|\boldsymbol{\beta}\|_2^2$
Elastic Net[1]	$\alpha\|\boldsymbol{\beta}\|_1 + (1-\alpha)\|\boldsymbol{\beta}\|_2^2$

[1] Will not be covered in this lecture. Essentially, a fancy name for combining ridge and lasso.

Best subset selection

Our goal

$$\min_{\beta_0,\beta} \sum_{i=1}^{N} \left(y_i - \beta_0 - \sum_{j=1}^{p} x_{ij}\beta_j \right)^2 \text{ subject to } \|\boldsymbol{\beta}\|_0 \leq t$$

In words: select the best model according to some statistical criteria, among all possible combination of t feature or less.

Best subset selection algorithm

1. For $k = 0, 1, \ldots, p$

   1.1 Fit all models that contain exactly k predictors. If $k = 0$, the forecast is the unconditional mean.

   1.2 Pick the best (e.g, highest R^2) among these models, and denote it by $\mathcal{M}_k$.

2. Optimize over $\{\mathcal{M}_0, \ldots, \mathcal{M}_p\}$ using cross-validation (or other criteria)

Issues:

1. The algorithm is very slow: at each step we deal with $\binom{p}{k}$ models ("N-P complete".)
2. The prediction is highly unstable: the subsets of variables in $\mathcal{M}_{10}$ and $\mathcal{M}_{11}$ can be very different from each other, leading to high variance (the best subset of $\mathcal{M}_3$ need not include any of the variables in best subset of $\mathcal{M}_2$.)

Faster subset selection algorithms

Instead of estimating all possible combinations, follow a particular path of models:

- Forward stepwise selection: start simple and expand (feasible even if $p > n$)
- Backward stepwise selection: start with the full model and drop features (not recommended)

Forward stepwise algorithm

1. Let $\mathcal{M}_0$ denote the null model, which contains just an intercept.

2. For $k = 0, \ldots, p-1$:

   2.1 Consider all $p - k$ models that augment the predictors in $\mathcal{M}_k$ with one additional predictor.

   2.2 Choose the best among these $p - k$ models, and call it $\mathcal{M}_{k+1}$. Here best is defined as having highest R^2

3. Select a single best model from among $\mathcal{M}_0, \ldots, \mathcal{M}_p$ using cross-validation.

> This is our first example of a **greedy algorithm**: a making the locally optimal selection at each stage with the intent of finding a global optimum.

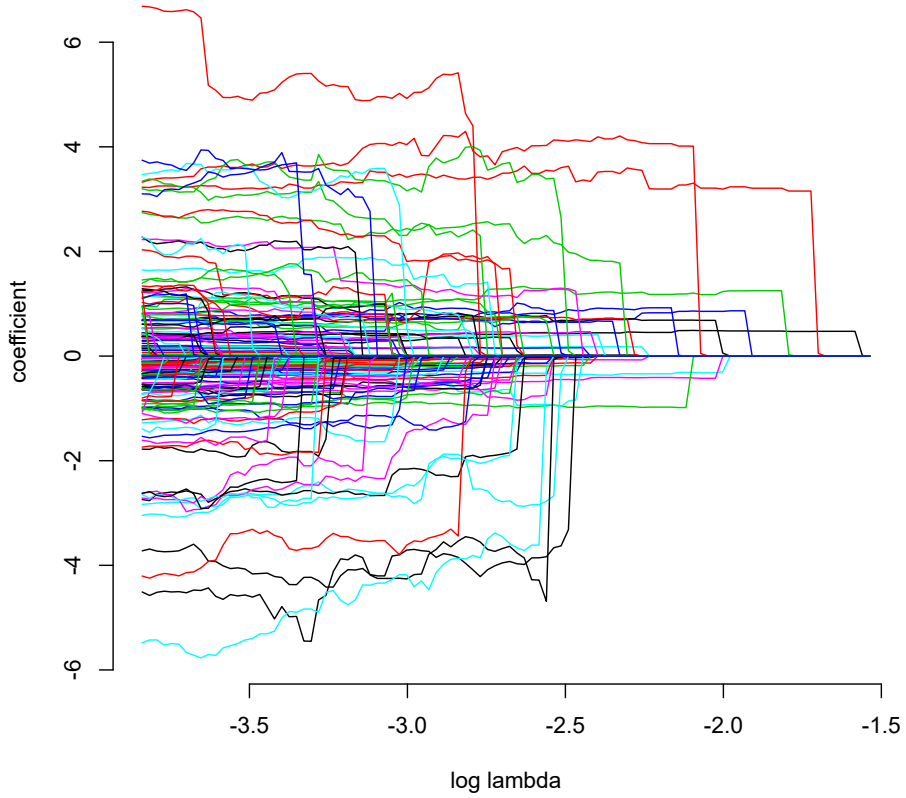
Stepwise using `gamlr`

`gamlr` is an R package that enables you, among other things, to estimate a forward stepwise regression model.

```
fit_step <- gamlr(Xweb, Yweb, gamma=Inf, lmr=.1)
plot(fit_step, df=FALSE, select=FALSE)
```

The figure on the right shows the value of the coefficients along the forward stepwise selection path.

Notice how the jagged are the solution paths. This discontinuity is the cause for instability in subset selection algorithms.



Shrinkage

Prerequisite: centering and scaling

In what follows, we assume that each feature is centered and scaled to have mean zero and unit variance, as follows:

$$\frac{x_{ij} - \widehat{\mu}_i}{\widehat{\sigma}_i}, \qquad \text{for } j = 1, 2, \ldots, p$$

where $\widehat{\mu}_i$ and $\widehat{\sigma}_i$ are the estimated mean and standard deviation of x_i estimated over the *training* set.

NOTE: This is not important when using OLS (why?)

The ridge regression

Ridge regression was introduced into the statistics literature by Hoerl and Kennard (1970).The optimization problem:

$$\min_{\beta_0,\beta} \sum_{i=1}^{N} \left(y_i - \beta_0 - \sum_{j=1}^{p} x_{ij}\beta_j \right)^2 + \lambda \|\boldsymbol{\beta}\|_2^2$$

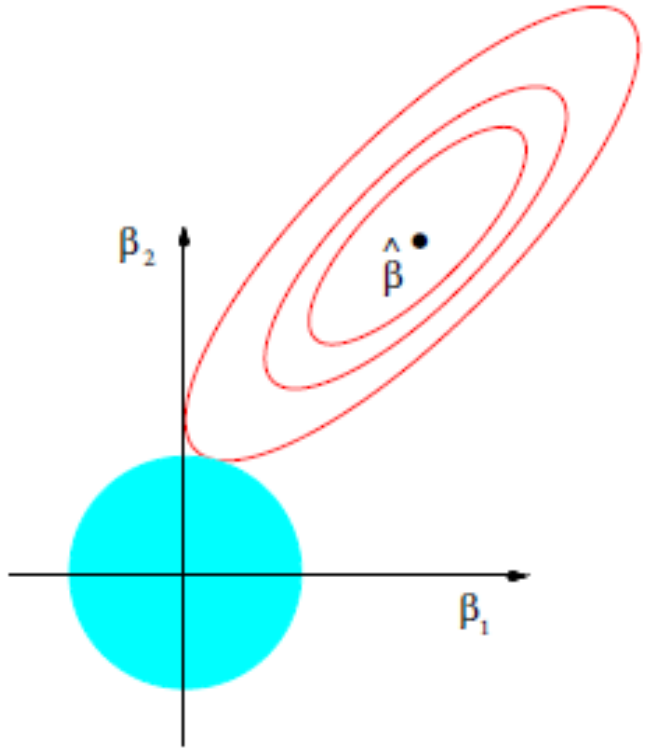
or in a budget constraint form:

$$\min_{\beta_0,\beta} \sum_{i=1}^{N} \left(y_i - \beta_0 - \sum_{j=1}^{p} x_{ij}\beta_j \right)^2 \textbf{ subject to } \|\boldsymbol{\beta}\|_2^2 \leq t$$

Ridge puts a "budget constraint" on the sum of squared betas. This constraint incorporate the cost of being "too far away" from the null hypothesis of $\beta_j = 0$ (what if this assumption is wrong null?)

Illistration of ridge in 2-D

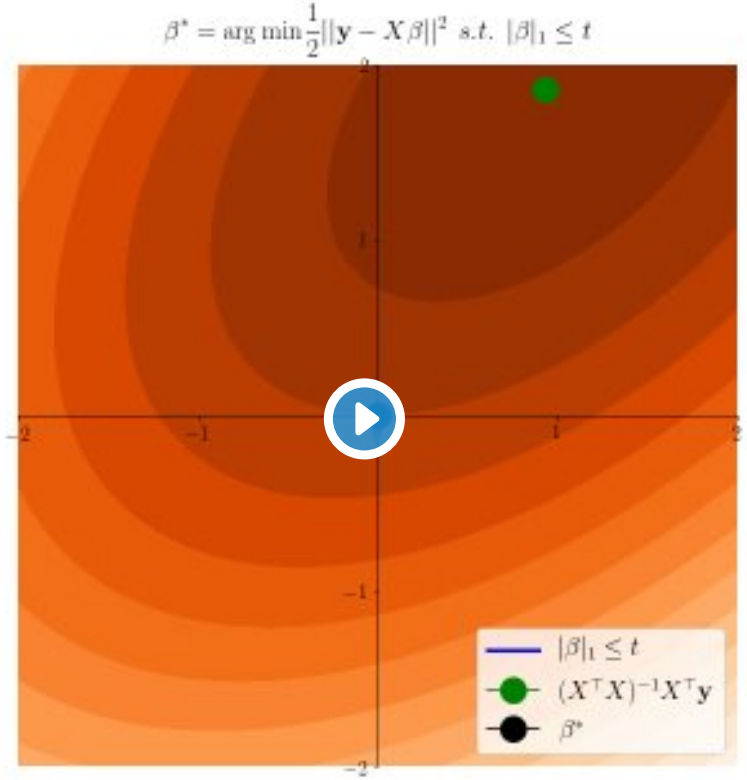
Contours of the error and constraint functions for ridge regression. The solid blue area is the constraint region, $\beta_1^2 + \beta_2^2 \leq t$, while the red ellipses are the contours of the RSS and $\widehat{\beta}$ is the OLS estimator.



Source: James et al. (2017)

Pierre Ablin
@PierreAblin

Illustration of the Lasso and its path in 2D: for t small enough, the solution is sparse!



389 ♡ 18 - 15:50 במרץ 2019

The solution

The problem in matrix notation:

$$\min_{\boldsymbol{\beta}} \ (\mathbf{y} - \mathbf{X}\boldsymbol{\beta})'(\mathbf{y} - \mathbf{X}\boldsymbol{\beta}) + \lambda \boldsymbol{\beta}'\boldsymbol{\beta}$$

The ridge estimator is given by

$$\widehat{\boldsymbol{\beta}}^{\mathrm{R}} = \left(\mathbf{X}'\mathbf{X} + \lambda \mathbf{I}\right)^{-1}\mathbf{X}'\mathbf{y}$$

NOTE: We can have a solution even if $\mathbf{X}$ is not of full rank (e.g., due to multicollinearity) since $\mathbf{X}'\mathbf{X} + \lambda\mathbf{I}$ is non-singular.

Bayesian interpretaion of ridge

- Consider the regression

$$y_i \sim N\left(\mathbf{x}_i'\boldsymbol{\beta}, \sigma^2\right)$$

  where we assume that σ is known.

- Suppose we put an independent prior on each β_j

$$\beta_j \sim \mathcal{N}\left(0, \tau^2\right)$$

- Then, the posterior mean for $\boldsymbol{\beta}$ is

$$\widehat{\boldsymbol{\beta}}_{\text{posterior}} = \left(\mathbf{X}'\mathbf{X} + \frac{\sigma^2}{\tau^2}\mathbf{1}\right)^{-1}\mathbf{X}'\mathbf{y}$$

- Hence, $\lambda = \frac{\sigma^2}{\tau^2}$.

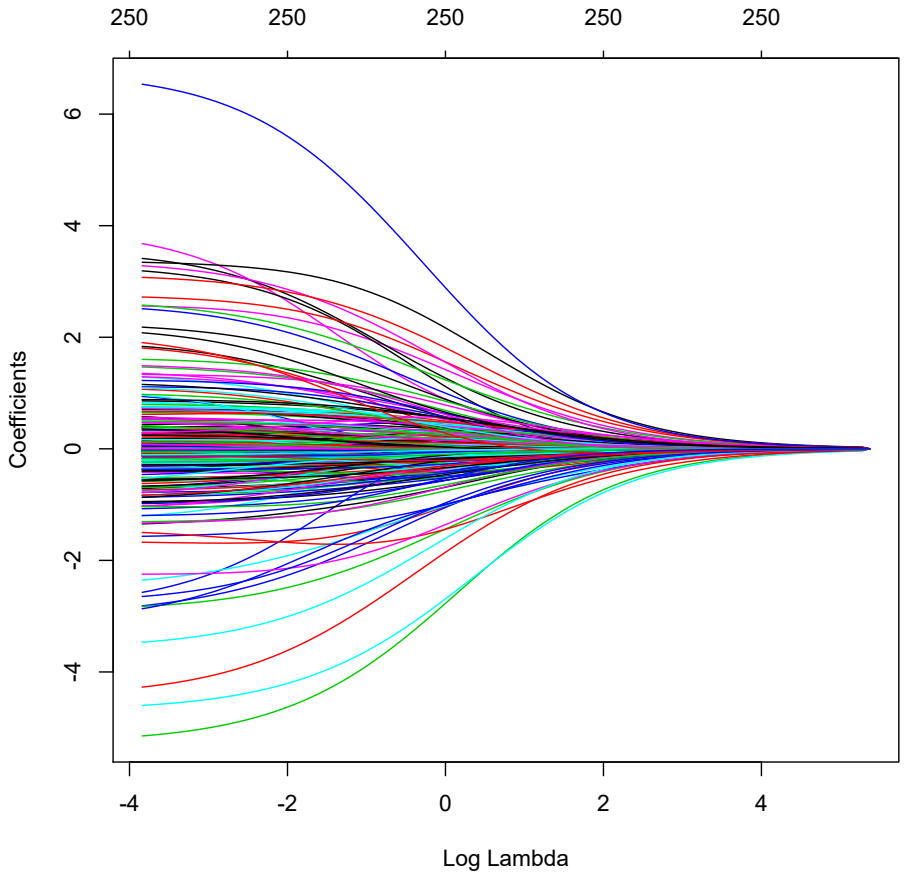
Estimating ridge using `glmnet`

The R package **`glmnet`** enables you to estimate ridge regression, along with its path for λ.

To do so, we need to maker sure to set `alpha = 0`.

```
fit_ridge <- glmnet(x = Xweb, y = Yweb, alpha = 0)
plot(fit_ridge, xvar = "lambda")
```

The figure on the right shows the ridge **regularization path**, i.e., the values of the (standardized) coefficients for different values of (log) λ.



How to choose λ?

K-fold cross validation algorithm:

1. Given a training set of n observations, randomly split the data into K roughly evenly sized folds (subsamples).

2. For $k = 1, \ldots, K$ 2.1 Fit the coefficients $\bar{\beta}$ using all but the k th fold of data. 2.2 Record MSE on the left-out k^{th} fold.

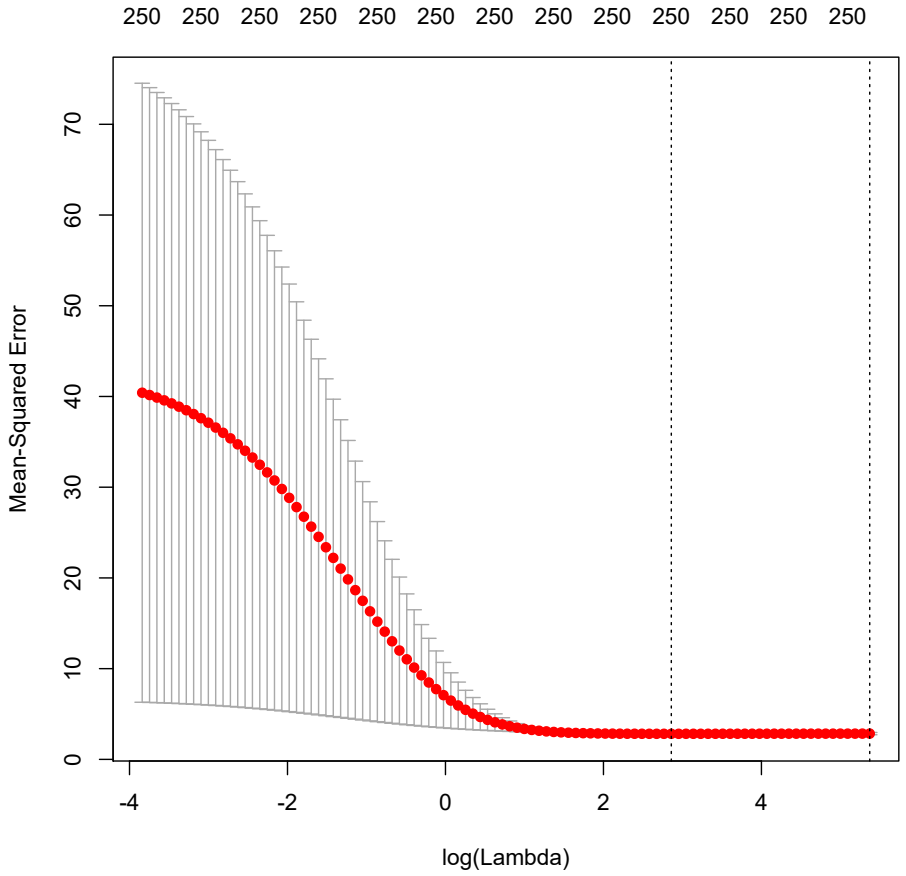
Tuning λ

The function `cv.glmnet()` automates cross validation for the ridge regression.

Note that you can change the default number of folds (10) by setting `nfolds` argument.

```
cv_ridge <- cv.glmnet(x = Xweb, y = Yweb, alpha = 0)
plot(cv_ridge)
```

- *Left* dotted vertical line: λ with min MSE
- *Right* dotted vertical line: the biggest λ with MSE no more than one SE away from the minimum



Pros and cons of ridge

PROS:

- when p is large, ridge significantly reduces variance (by introducing bias.)
- works even when $p > n$.

CONS:

- Though ridge shrinks all of the coefficients toward zero, it does not set any of them to exactly zero. Hence, when p is large, interpretation remains a challenge.

The lasso

Lasso (least absolute shrinkage and selection operator) was introduced by Tibshirani (1996). The optimization problem:

$$\min_{\beta_0,\beta} \sum_{i=1}^{N} \left(y_i - \beta_0 - \sum_{j=1}^{p} x_{ij}\beta_j \right)^2 + \lambda \|\boldsymbol{\beta}\|_1$$

Lasso puts a "budget constraint" on the sum of *absolute* β's.

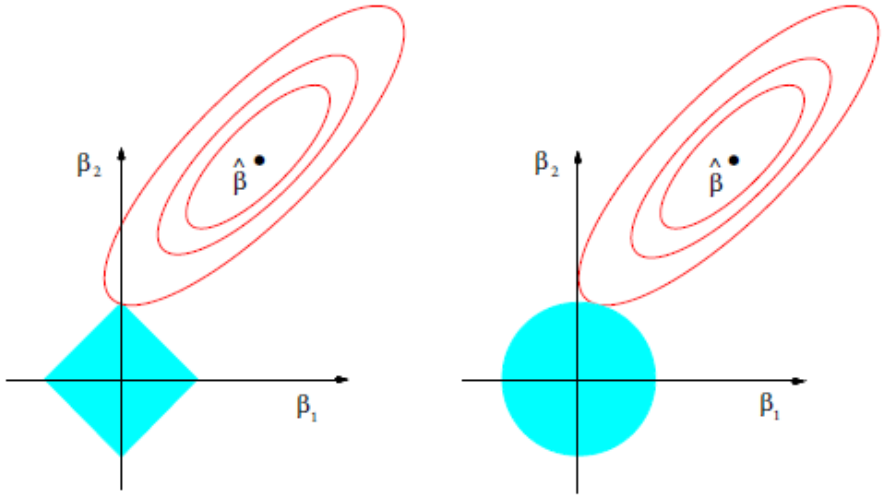
Unlike ridge, the lasso penalty is linear (moving from 1 to 2 is the same as moving from 101 to 102.)

A great advantage of the lasso is that it zeros out most of the β's in the model (the solution is *sparse*.)

Any penalty that involves the ℓ_1 norm will do this.

Lasso vs. ridge

Contours of the error and constraint functions for lasso (left) and ridge (right). The solid blue areas are the constraint regions, $\beta_1^2 + \beta_2^2 \leq t$, and $\beta_1^2 + \beta_2^2 \leq t$, while the red ellipses are the contours of the RSS and $\widehat{\beta}$ is the OLS estimator.

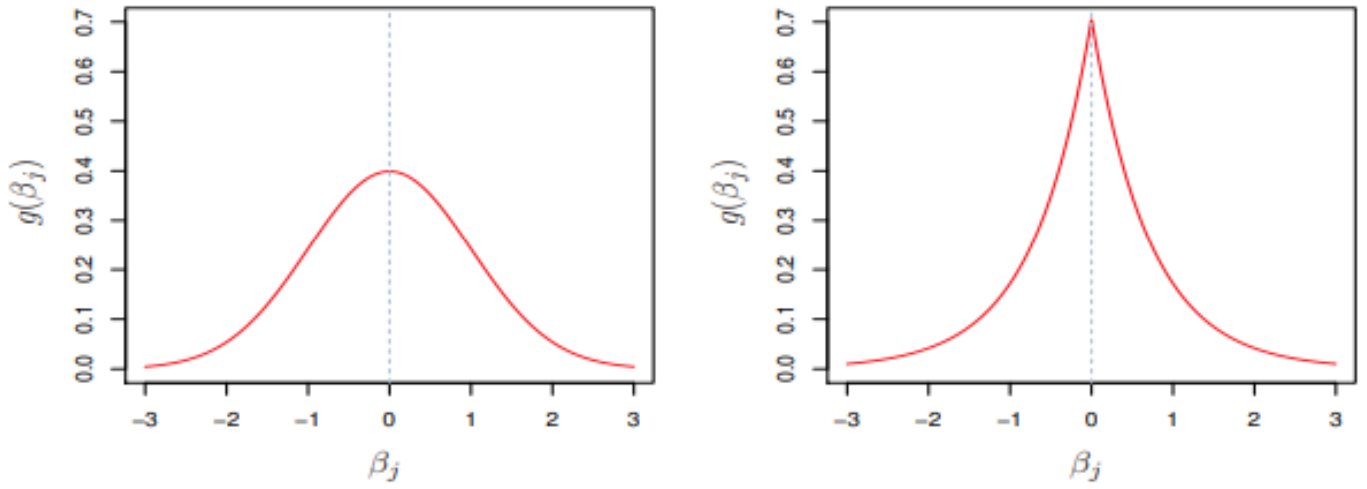


Source: James et al. (2017)

Bayesian interpretation of lasso

The prior distribution of β under lasso is the double exponential (Laplace) with density $1/2\tau \exp(-|\beta|/\tau)$, where $\tau = 1/\lambda$. The posterior mode is equal to the lasso solution

On the left, normal prior (ridge). On the right, Laplace prior (lasso):

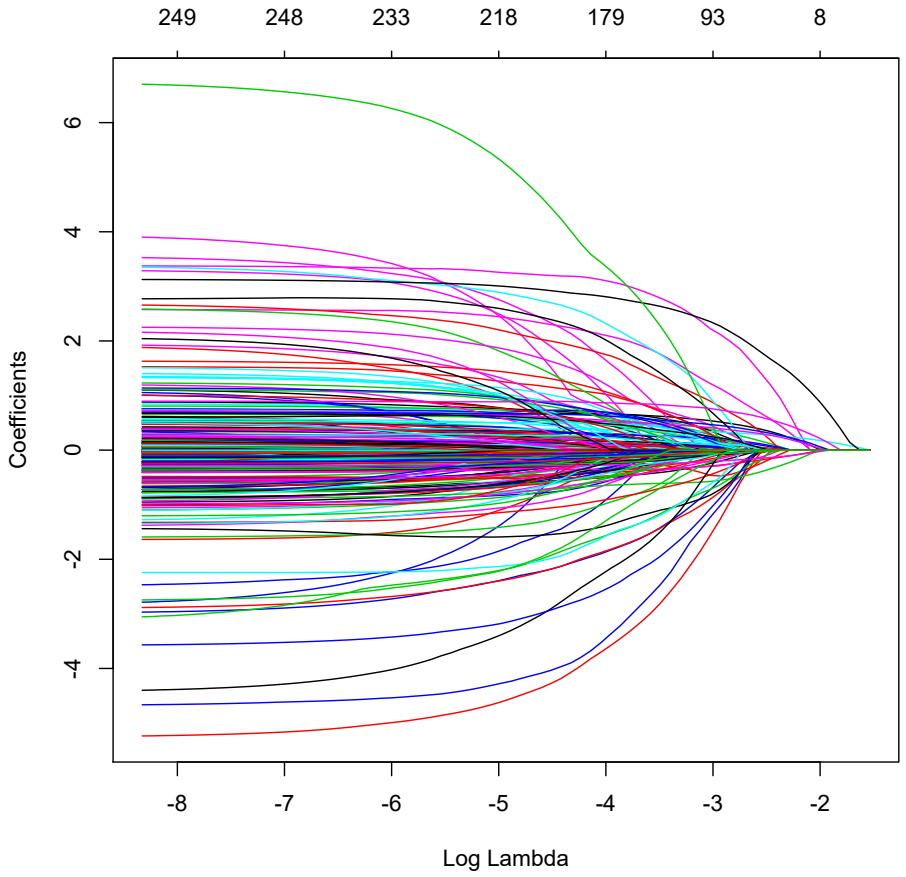


Source: James et al. (2017)

Estimating lasso using `glmnet`

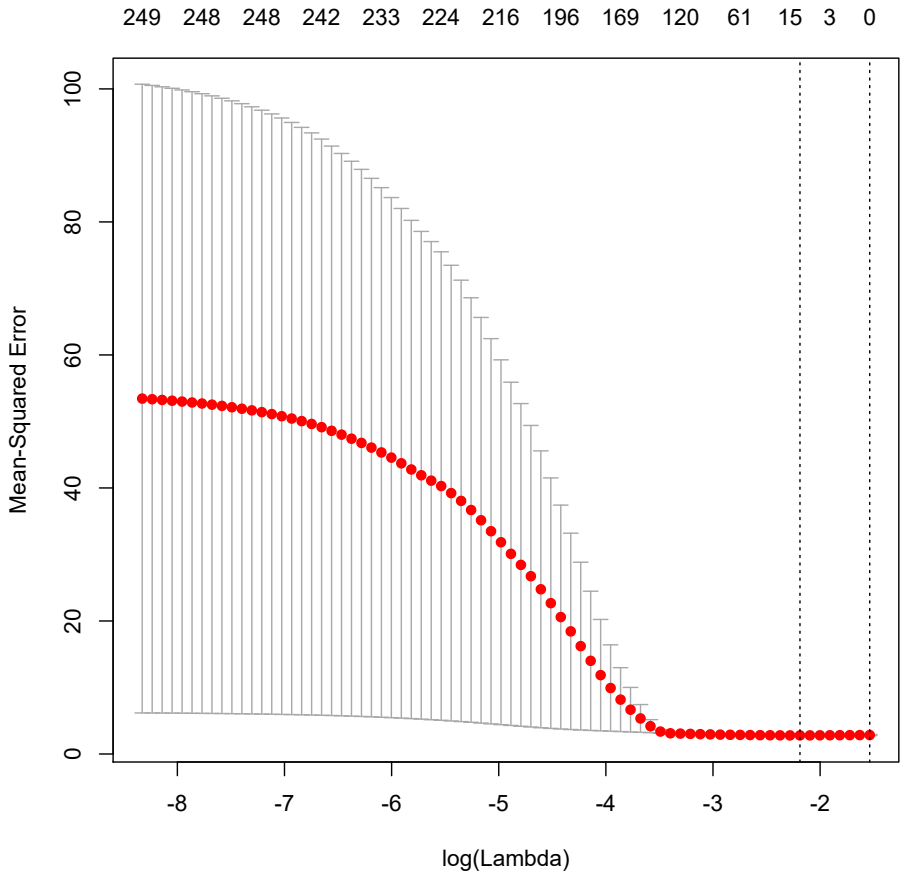
The `glmnet()` function with `alpha = 1` (the default) estimates the entire lasso regularization path.

```
fit_lasso <- glmnet(x = Xweb, y = Yweb, alpha = 1)
plot(fit_lasso, xvar = "lambda")
```



Tuning λ

```
cv_lasso <- cv.glmnet(x = Xweb, y = Yweb, alpha = 1)
plot(cv_lasso, xvar = "lambda")
```



Which features were selected?

Using `s = lambda.min`:

```
colnames(Xweb)[coef(cv_lasso, s = "lambda.min")@i]
```

```
##  [1] "oldnavy.com"       "connextra.com"     "smartbargains.com"
##  [4] "webratsmusic.com"  "checkm8.com"       "qvc.com"
##  [7] "travelocity.com"   "yahoo.net"         "windows.com"
## [10] "staples-deals.com" "msn.com"           "evite.com"
```

Using `s = lamda.1se`:

```
colnames(Xweb)[coef(cv_lasso, s = "lambda.1se")@i]
```

```
## character(0)
```

A note about shrinkage

Assume that $n = p$ and $\mathbf{X}$ is an $n \times p$ diagonal matrix with 1's on its diagonal.

OLS finds $\beta_1, \ldots, \beta_p$ that minimize

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

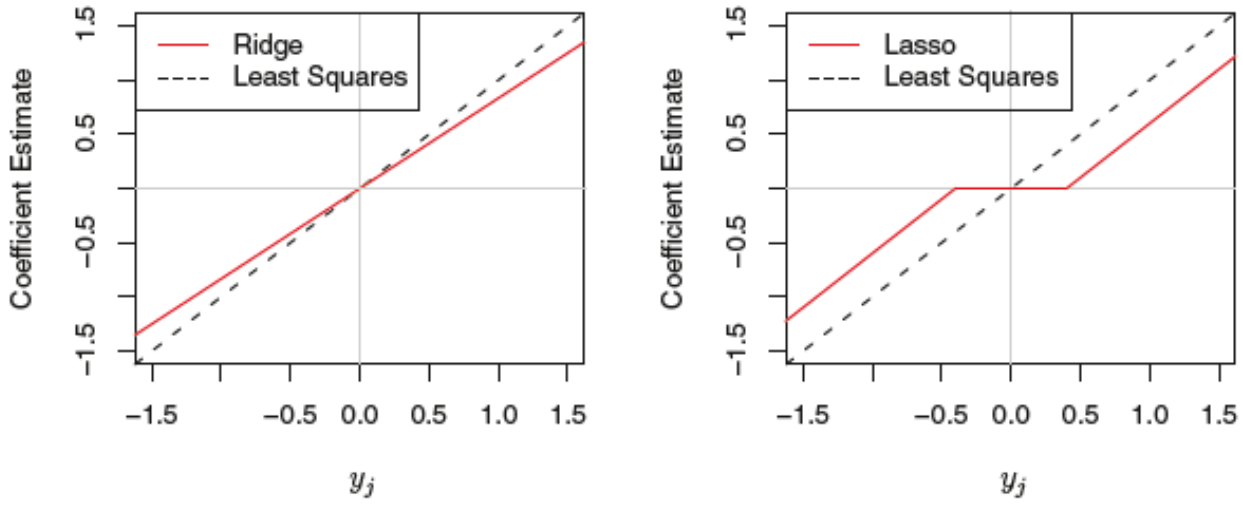
The solution $\hat{\beta}_j^{\text{OLS}} = y_j$ (that is, perfect fit, $R^2 = 1$.)

In this setting, the ridge regression estimates take the form

$$\hat{\beta}_j^{Ridge} = \frac{\hat{\beta}^{\text{OLS}}}{(1+\lambda)}, \qquad \hat{\beta}_j^{Lasso} = \begin{cases} \hat{\beta}^{\text{OLS}} - \lambda/2 & \text{if } \hat{\beta}^{\text{OLS}} > \lambda/2 \\ \hat{\beta}^{\text{OLS}} + \lambda/2 & \text{if } \hat{\beta}^{\text{OLS}} < -\lambda/2 \\ 0 & \text{if } \left|\hat{\beta}^{\text{OLS}}\right| \leq \lambda/2 \end{cases}$$

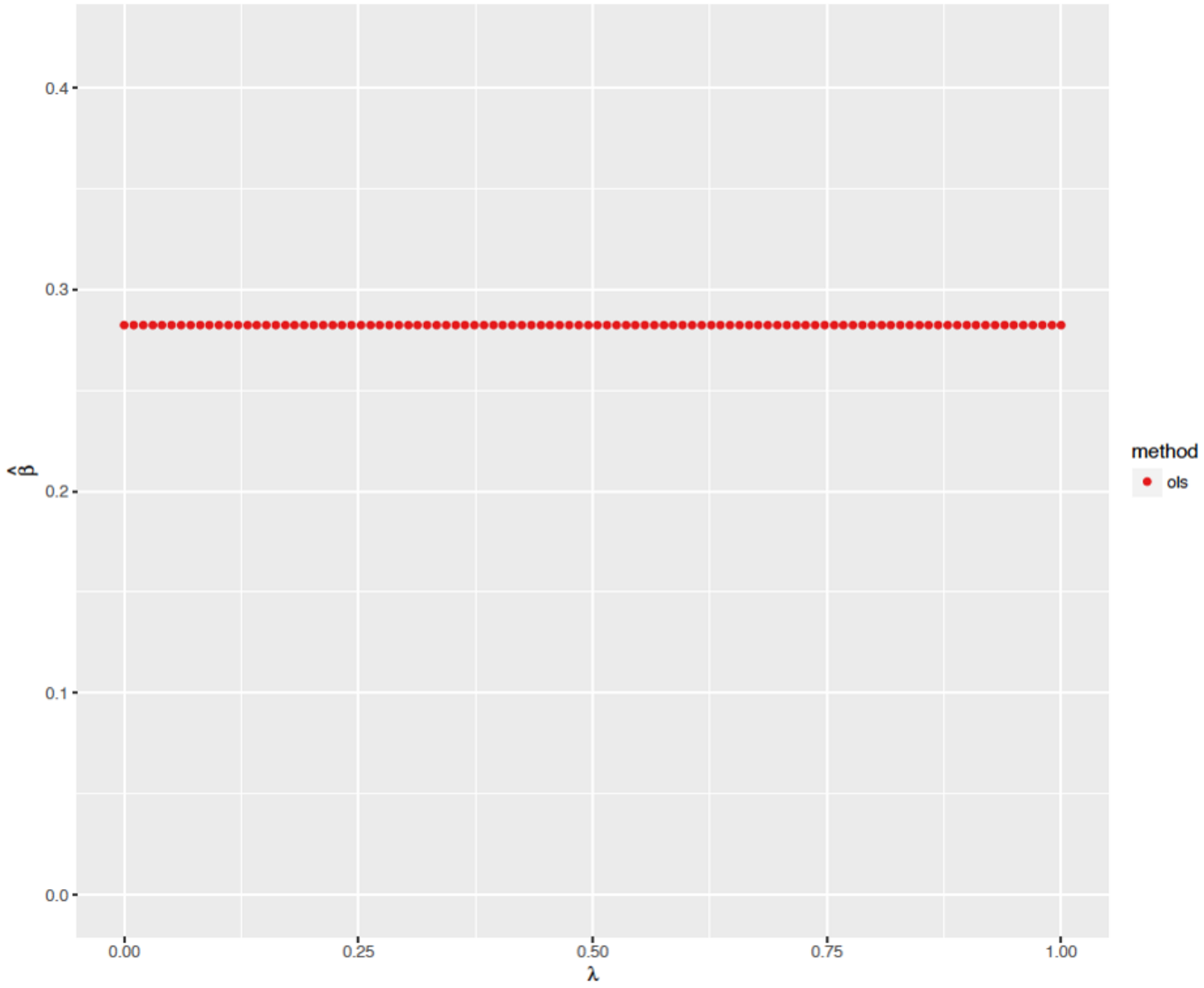
Two types of shrinkage

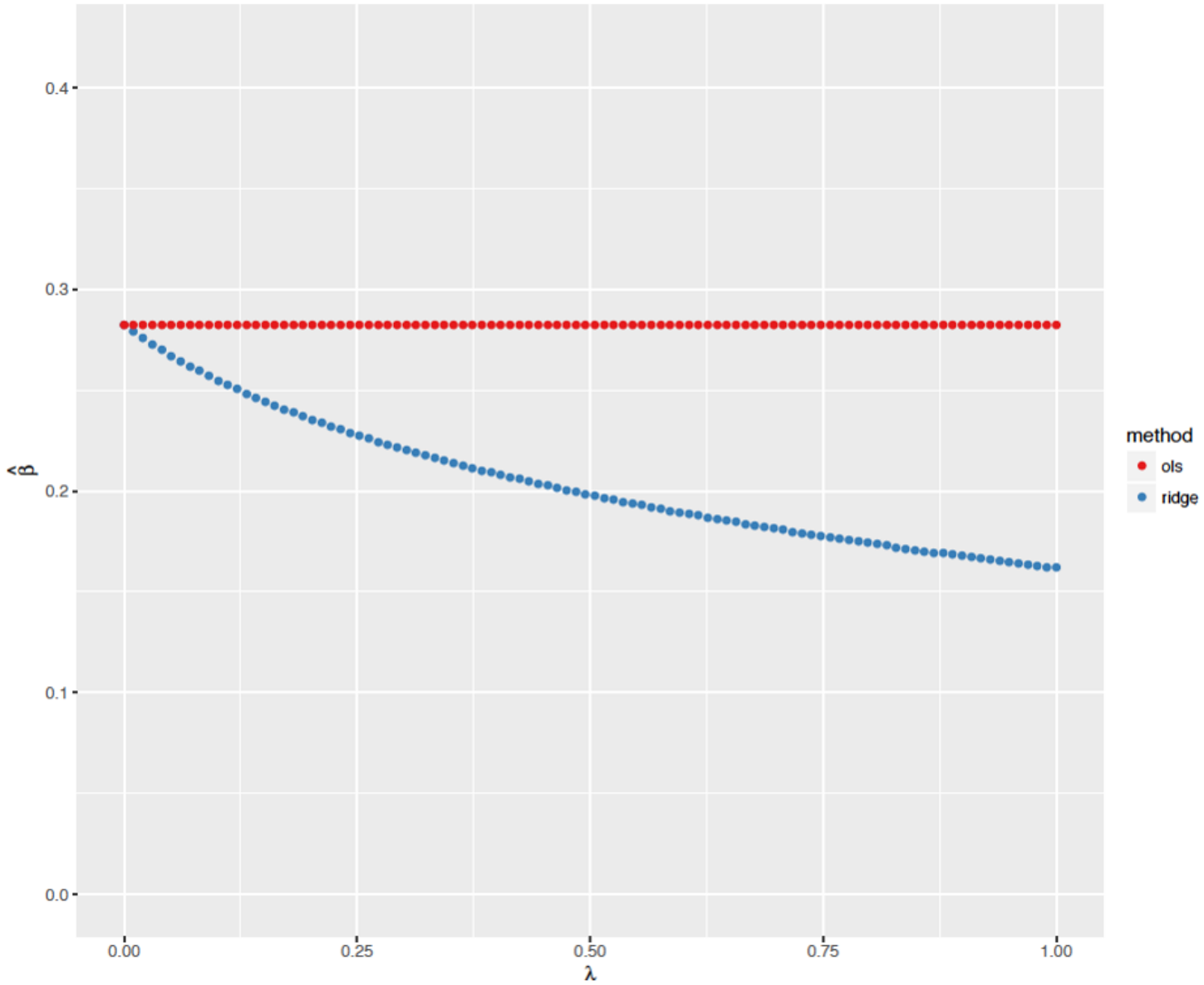
Ridge shrinks proportionally, whereas lasso shrinks by a similar amount, and sufficiently small coefficients are shrunken all the way to zero. The lasso type of shrinkage is called "soft-thresholding."

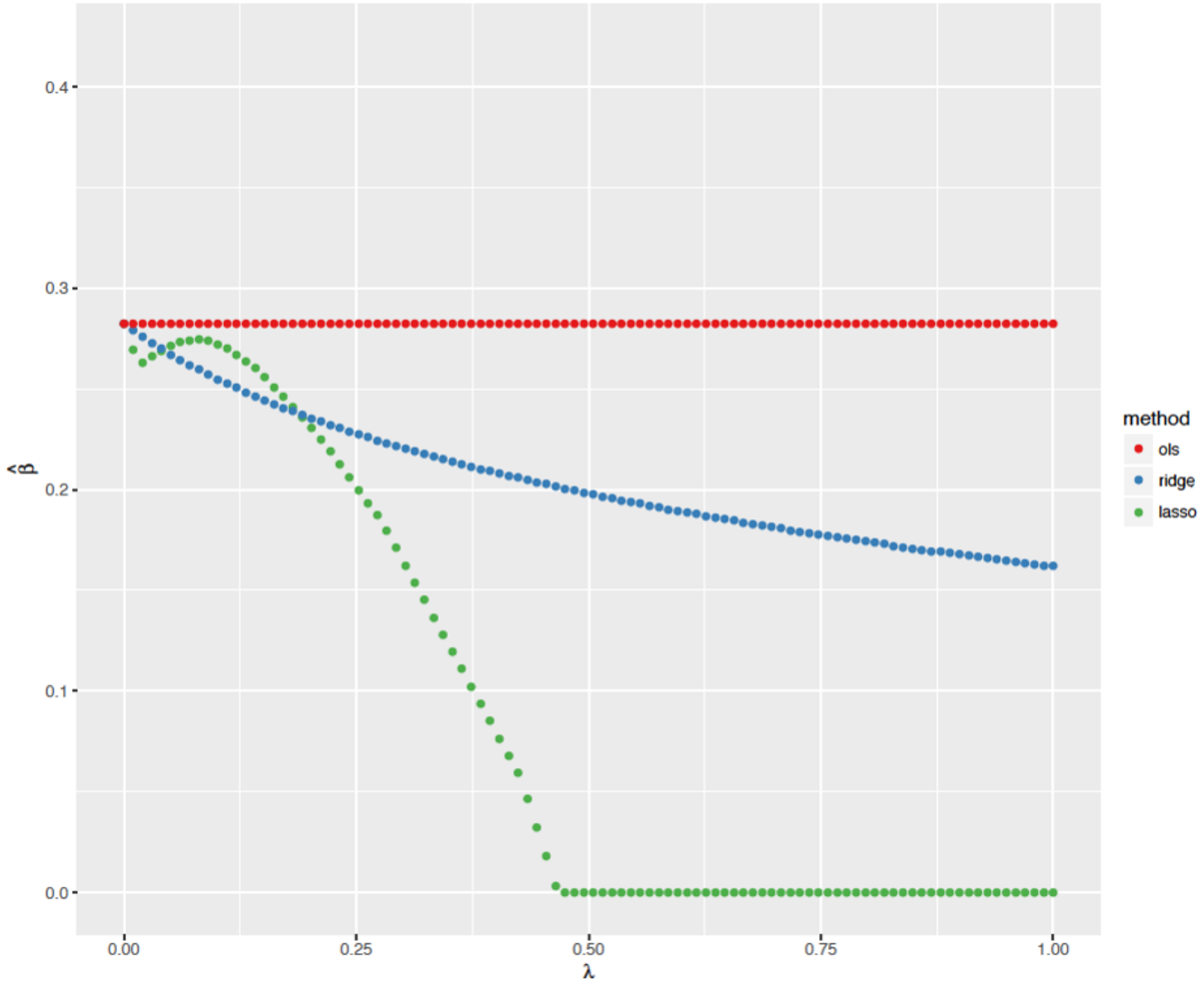


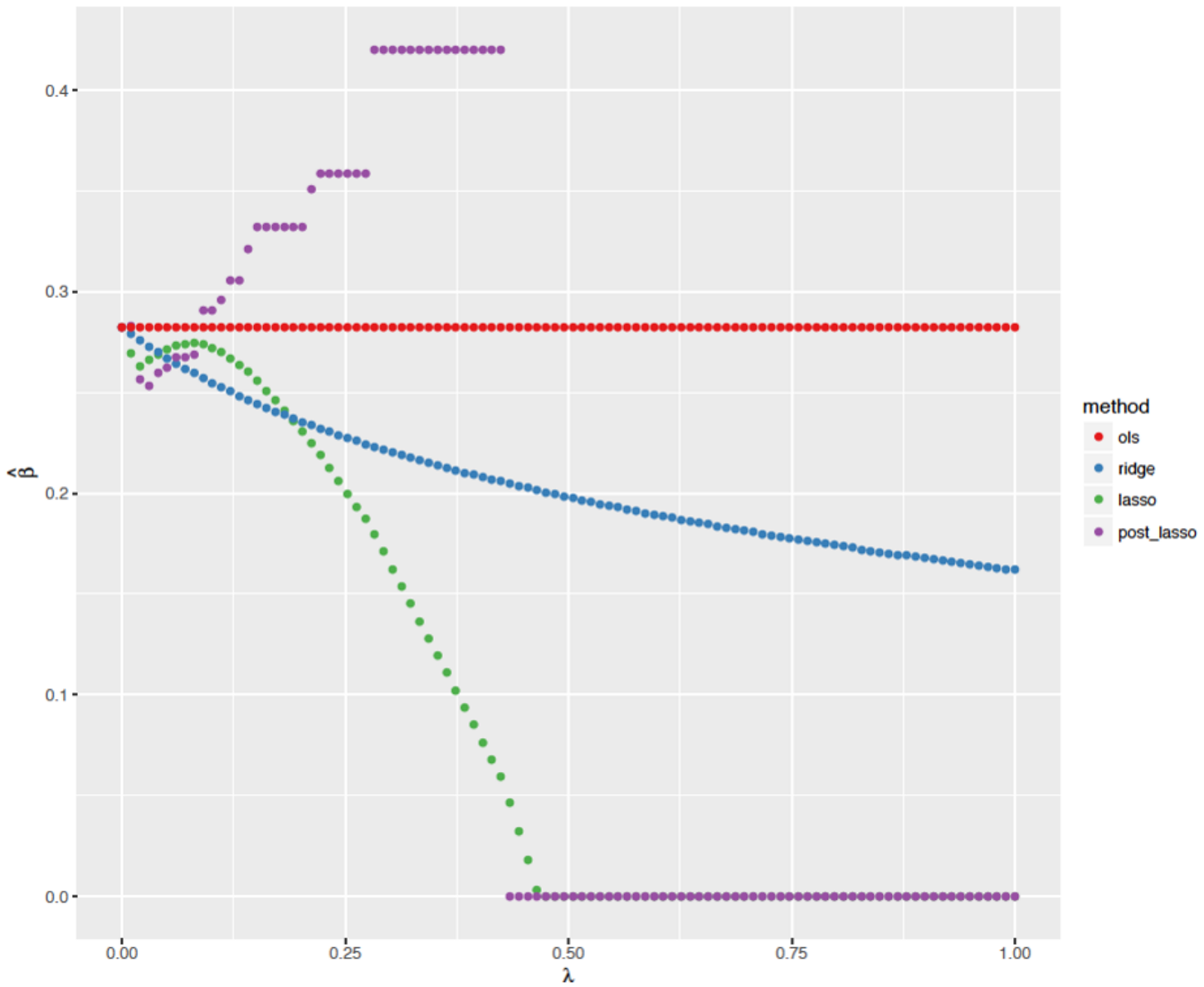
Source: James et al. (2017)

Main takeaway: We can't just use ridge/lasso coefficients for statistical inference (unless you're a Bayesian...)


0.4
0.3
0.2
0.1
0.0
β̂
0.00
0.25
0.50
0.75
1.00
λ
method
ols


0.4
0.3
0.2
0.1
0.0
$\hat{\beta}$
0.00
0.25
0.50
0.75
1.00
λ
method
ols
ridge


0.4
0.3
0.2
0.1
0.0
$\hat{\beta}$
0.00
0.25
0.50
0.75
1.00
λ
method
ols
ridge
lasso


0.4
0.3
0.2
0.1
0.0
$\hat{\beta}$
0.00
0.25
0.50
0.75
1.00
λ
method
ols
ridge
lasso
post_lasso

Shortcomings of linear models and what comes next

- Linear models can be very flexible but need lots of work (and memory and computing power) when it comes to explicitly adding polynomials, interactions, etc.
- Out next set of models, trees, random forests, are better equipped for these tasks, but come with a cost when it comes to interpretability.

Dimensionality reduction

Next time

`slides::end()`

 Source code

References

Hoerl, A. E., and Kennard, R. W. 1970. Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics*, 12(1), 55-67.

Taddy, M. 2019. *Business Data Science: Combining Machine Learning and Economics to Optimize, Automate, and Accelerate Business Decisions* . McGraw-Hill Education. Kindle Edition.

Tibshirani, R. 1996. Regression shrinkage and selection via the lasso. *J . Roy. Statist. Soc. B*, 58(1), 267-288.