

## 6.2 Shrinkage Methods

Lauren Perry

## Shrinkage Methods

- ▶ Subset selection methods use least squares to fit a linear model that contains a subset of the predictors.
- ▶ Instead, we can fit a model using all  $p$  predictors, with some of the coefficients shrunk toward zero.
- ▶ It turns out that shrinking the coefficient estimates can significantly reduce their variance.

## Least Squares

Recall: least squares regression estimates the coefficients by minimizing

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

## Ridge Regression

In *ridge regression*, we instead minimize

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

where  $\lambda > 0$  is a *tuning parameter*

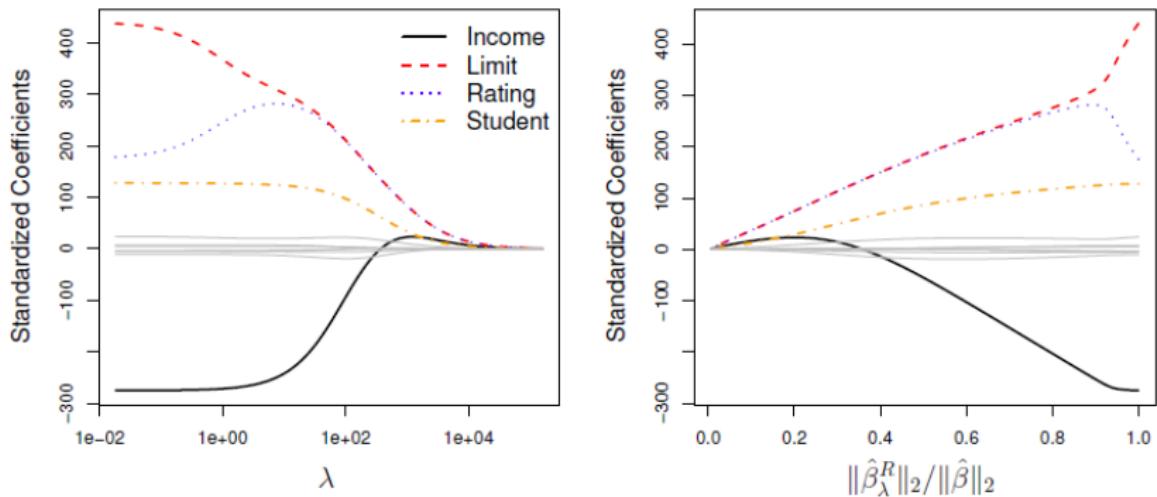
## Ridge Regression

- ▶ Ridge regression still finds parameters which fit the data well by making RSS small.
- ▶ However, it now has to balance that with the *shrinkage penalty*  $\lambda \sum_{j=1}^p \beta_j^2$ 
  - ▶ This term will be small when all the  $\beta$ s are close to 0.

## Ridge Regression

- ▶ Balancing RSS with the penalty term shrinks the coefficients toward 0.
  - ▶ Less useful coefficients will have values closer to 0.
- ▶ The tuning parameter  $\lambda$  controls the relative impact of these two terms.
  - ▶ We use cross-validation to select  $\lambda$

# Ridge Regression



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

## More on Previous Figure

- ▶ LHS shows ridge regression coefficient estimates plotted as a function of  $\lambda$
- ▶ RHS shows coefficient estimates as a function of

$$\|\hat{\beta}_\lambda^R\|_2 / \|\hat{\beta}\|_2$$

- ▶  $\hat{\beta}$  is the least squares coefs;  $\hat{\beta}_\lambda^R$  the ridge regression coefs
- ▶  $\|\beta\|_2$  is the  $l_2$  norm ("L-2") of a vector  $\|\beta\|_2 = \sqrt{\sum_{j=1}^p \beta_j^2}$

## Predictor Scaling

- ▶ Least squares estimates are *scale equivariant*
  - ▶ Multiplying  $X_j$  by a constant  $c$  scales the coef estimates by  $1/c$  (and so  $X_j \hat{\beta}_j$  will not change)
- ▶ Ridge regression coefficients are not
  - ▶ Coef estimates can change dramatically depending on variable scale.
  - ▶ To deal with this, we often standardize the predictors using

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

## Example

```
data("Credit")
x <- model.matrix(Balance ~ ., Credit)[, -1]
y <- Credit$Balance

library(glmnet)
grid <- 10^seq(10, -2, length = 100)
ridge.mod <- glmnet(x, y, alpha = 0, lambda = grid)
```

- ▶ glmnet will create a grid automatically, or we can create our own.
- ▶ For ridge regression, glmnet automatically scales the predictor variables.
- ▶ You will see more details in the lab.

## Example

Selecting an arbitrary value of  $\lambda$ ,

```
ridge.mod$lambda[50]
```

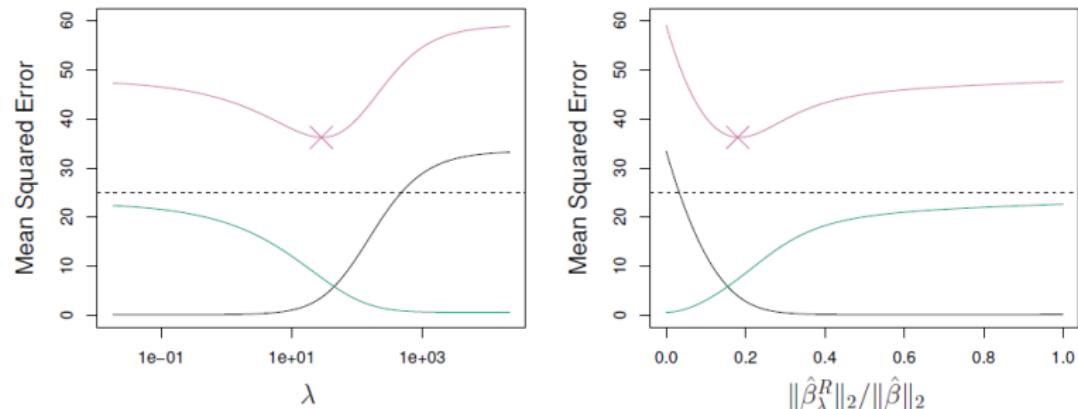
```
## [1] 11497.57
```

```
coef(ridge.mod)[,50]
```

	(Intercept)	Income	Limit	Rating
##	443.882328225	0.207297549	0.006255511	0.093529429
##	Age	Education	OwnYes	StudentYes
##	-0.007423108	-0.036491175	0.727221775	15.224286108
##	RegionSouth	RegionWest		
##	-0.088984491	-0.323636287		

# Why Ridge Regression?

The bias variance tradeoff strikes again



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

## Ridge Regression

Disadvantage:

- ▶ Although it shrinks coefficients *toward* zero, it is unable to set coefficients equal to 0.
- ▶ So all  $p$  predictors will always be included in the model.

## The Lasso

- ▶ More recent alternative to ridge regression that is able to shrink coefficients *to* 0.
- ▶ Now we minimize

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

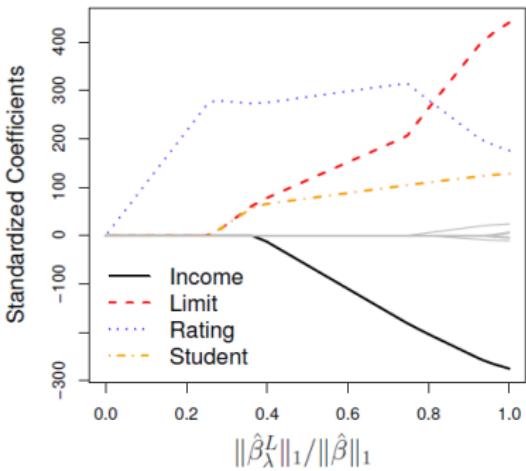
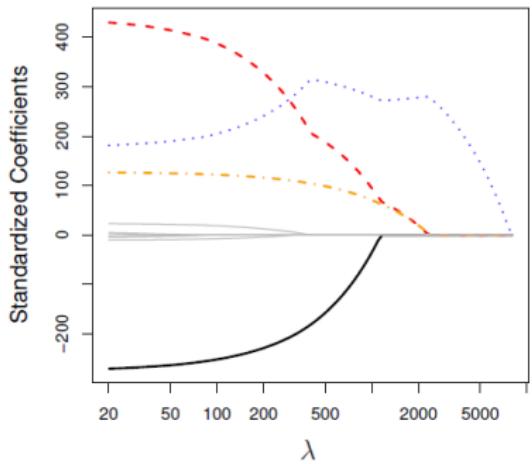
- ▶ This uses an  $l_1$  ("L-1") penalty instead of an  $l_2$  penalty:

$$||\beta||_1 = \sum_{j=1}^p |\beta_j|$$

## The Lasso

- ▶ The  $l_1$  penalty forces some coefficient estimates to exactly 0 when  $\lambda$  is sufficiently large.
- ▶ So the Lasso is a variable selection method.
- ▶ We say the Lasso yields *sparse* models since they involve only a subset of the variables.
- ▶ We again use cross-validation to select good values for  $\lambda$ .

## Example: Credit Dataset



## Variable Selection and Lasso

Why does the  $l_1$  penalty force coefficients to 0 (and the  $l_2$  penalty does not)?

Ridge regression minimizes

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

with the constraint that

$$\sum_{j=1}^p \beta_j^2 \leq s$$

## Variable Selection and Lasso

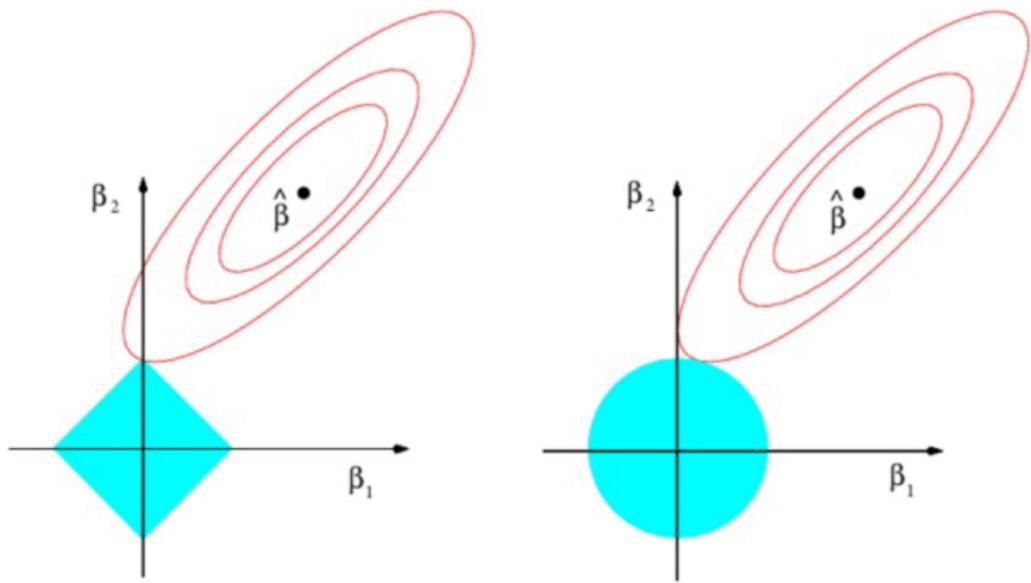
Lasso minimizes

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

with the constraint that

$$\sum_{j=1}^p |\beta_j| \leq s$$

## Variable Selection and Lasso



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

## Variable Selection and Lasso

- ▶ Each ellipses represents a contour on which all points have the same RSS value.
- ▶ The blue areas are the constraint regions.
- ▶ The ridge/lasso estimates are where the ellipses meets the constraint region.
  - ▶ With ridge, this generally does not occur on an axis.
  - ▶ With lasso, this is likely to occur at an axis, and so at least one of the coefficient estimates will be zero.

## Example

```
lasso.mod <- glmnet(x, y, alpha = 1, lambda = grid)
```

- ▶ The only difference from ridge regression is that, to get `glmnet` to do a ridge regression, we set `alpha=1`.

## Example

Selecting an arbitrary value of  $\lambda$ ,

```
lasso.mod$lambda[50]
```

```
## [1] 11497.57
```

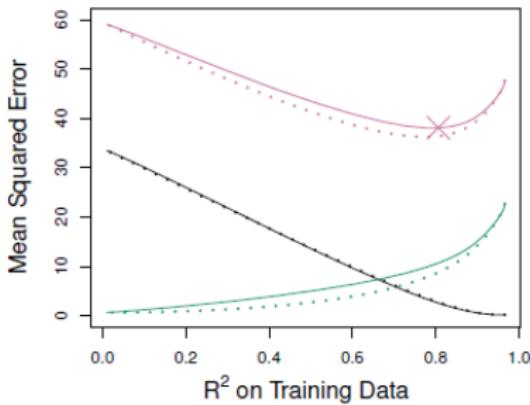
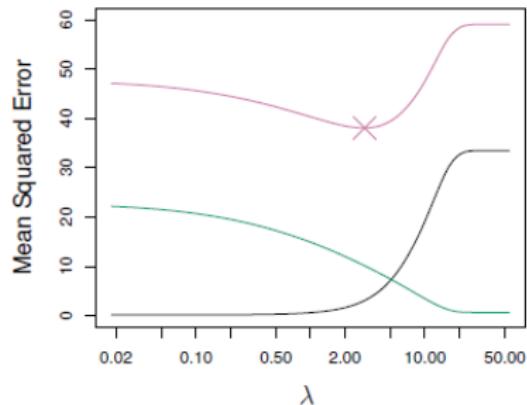
```
coef(lasso.mod)[,50]
```

	## (Intercept)	Income	Limit	Rating	Ca
##	520.015	0.000	0.000	0.000	0.
##	Education	OwnYes	StudentYes	MarriedYes	RegionS
##	0.000	0.000	0.000	0.000	0.

## Ridge or Lasso?

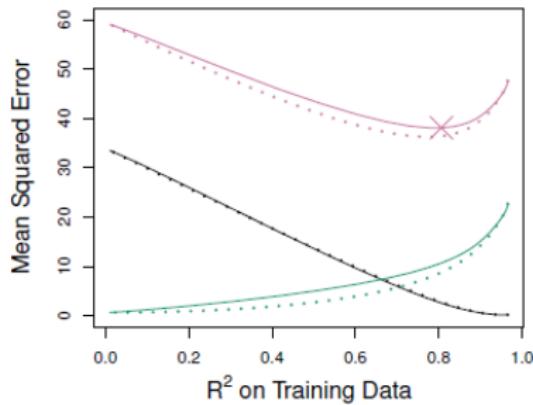
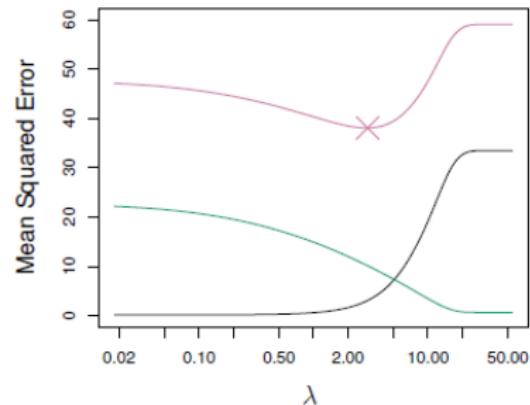
- ▶ Lasso has at least one clear advantage over ridge regression: variable selection.
- ▶ Is Lasso always better? We consider variance, squared bias, and MSE of both methods.

# Ridge or Lasso?



- ▶ Simulation setting: all 45 predictors are related to the response.
- ▶ LHS: squared bias (black), variance (green), and test MSE (purple) for the lasso on simulated data
- ▶ RHS: comparison between lasso (solid) and ridge (dotted)

# Ridge or Lasso?



- ▶ Simulation setting: only 2 of the 45 predictors are related to the response.
- ▶ LHS: squared bias (black), variance (green), and test MSE (purple) for the lasso on simulated data
- ▶ RHS: comparison between lasso (solid) and ridge (dotted)

## Ridge or Lasso?

So which is better? It depends.

- ▶ In general, ridge regression will perform better if all coefficients are related to the response
- ▶ ... and lasso will perform better if relatively few are related to the response.

but in practice we don't know this.

- ▶ We can use cross-validation to determine which approach to use for a particular dataset.

## A Simple Special Case

We build intuition by considering the following:

- ▶ Let  $n = p$  and  $X$  be the identity matrix  $I_n$ .
- ▶ We will build our model without an intercept.
- ▶ Then the least squares problem minimizes

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

- ▶ The least squares solution is given by

$$\hat{\beta}_j = y_j$$

## A Simple Special Case

Ridge regression amounts to minimizing

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

and lasso to minimizing

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

## A Simple Special Case

Here, the ridge estimates take the form

$$\hat{\beta}_j^R = y_j / (1 + \lambda)$$

and the lasso estimates take the form

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

## A Simple Special Case

- ▶ Notice that ridge shrinks all coefficients by the same proportion.
- ▶ but lasso shrinks each by a constant amount *or* sets the coefficient to 0.
- ▶ A more realistic  $X$  matrix results in a more complex setup, but the ideas are *basically* the same.

## Selecting the Tuning Parameter

- ▶ One challenge of ridge regression and lasso is that  $\lambda$  is user-defined.
- ▶ Cross-validation provides a straightforward way to tackle this problem.
  - ▶ Choose a grid of  $\lambda$  values and compute the CV error rate at each.
  - ▶ Select the tuning parameter for which the cross-validation error is smallest.
  - ▶ Fit the model using all available observations using the selected tuning parameter.

## Example

```
set.seed(1)
cv.out <- cv.glmnet(x, y, alpha = 0)
plot(cv.out)
bestlam <- cv.out$lambda.min
bestlam
log(bestlam)
```

- ▶ Here, we run a ridge regression, using cross-validation to select  $\lambda$ .
- ▶ This can be done fairly simply in R!
- ▶ Again, you will spend some time with this in the Ch 6 lab.

# Example

