Penalized Regression Methods

-Applied Analytics-

APM Chapter 6.4 & ISL Chapter 6.2

Lecturer: Darren Homrighausen, PhD

Preamble:

- Penalized methods allow us to exchange some bias for lower variance
- We discuss some penalized regression procedures, in particular ridge regression, lasso, and the elastic net

Least squares

Minimize the training/apparent error (SSE in the regression application) over all β:

$$SSE = \sum_{i=1}^{n}(Y_i - (\beta_0 + \sum_{j=1}^{p}\beta_j x_j))^2$$

When there are two features, this looks like searching over all candidate β_1, β_2:

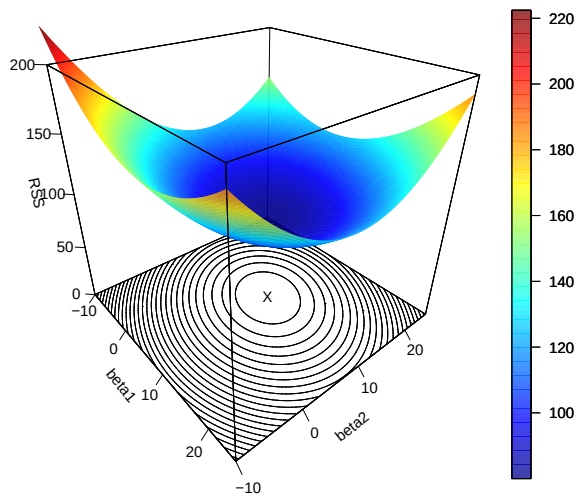


Figure: The "X" corresponds to $\hat{\beta}_{LS}$

Ridge regression

Ridge regression adds a penalty term given by the sum of the squared entries in

$$SSE_{L_2} = \sum_{i=1}^{n}(Y_i - (\beta_0 + \sum_{j=1}^{p}\beta_j x_j))^2 + \lambda \sum_{j=1}^{p}\beta_j^2$$

where the strength of this penalty term is the size of λ, which must be nonnegative

Observe:

- $\lambda = 0$ directly corresponds to multiple linear regression
- larger values of λ correspond to more penalty put on $\sum_{j=1}^{p}\beta_j^2$

Note that there is an equivalent way of thinking about ridge regression which is instructive

Ridge regression

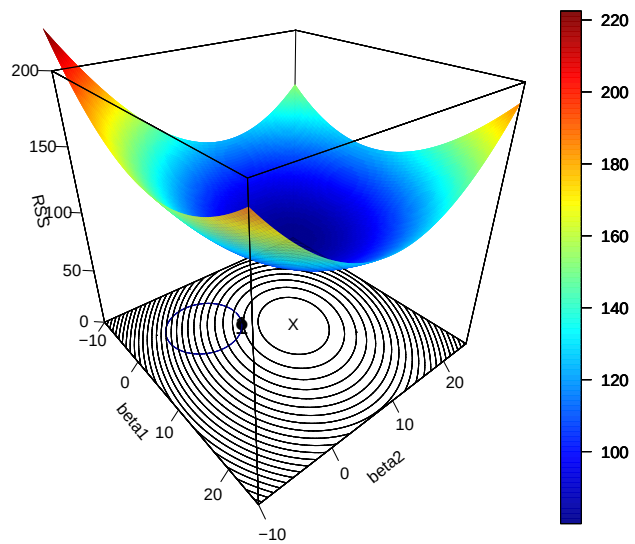
Reminder: The equation for a circle is:

$$x^2 + y^2 = r^2$$

This creates a circle centered at (0,0) with radius r

The penalty term $\sum_{j=1}^{p} \beta_j^2$ is a circle!

This translates geometrically to a constraint set



Ridge regression and multicollinearity

Reminder: Multicollinearity was when two or more features are correlated, which leads to the fitted model becoming very 'tipped'

This tipping becomes apparent when we have two features that should have the same sign on the estimated coefficients, but the values are large and the signs are opposite

Example: Predict blood pressure via weight & body surface area

```
Y = blood$BP
weight = blood$Weight #persons weight
bsa = blood$BSA

outBoth = lm(Y~bsa+weight)
summary(outBoth)
outBSA = lm(Y~bsa)
summary(outBSA)
outWeight = lm(Y~weight)
summary(outWeight)
```

Ridge regression and multicollinearity

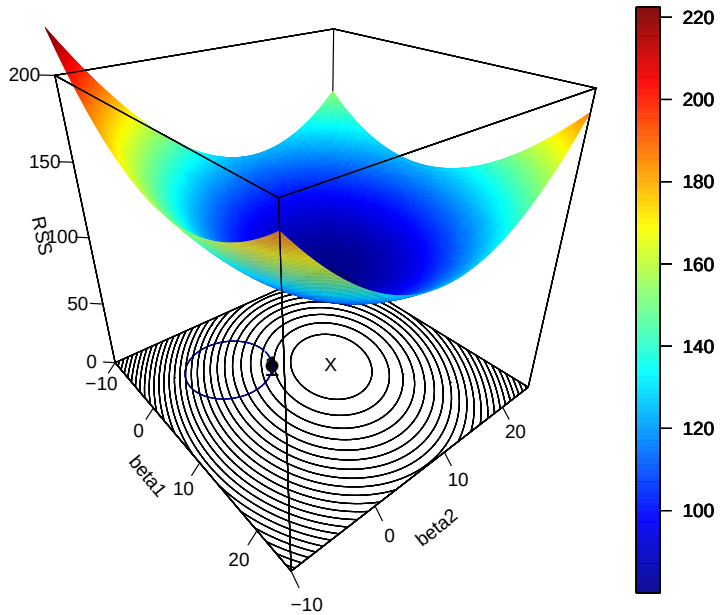
```
lm(formula = Y ~ bsa + weight)
            Estimate Std. Error t value Pr(>|t|)
(Intercept)    5.653      9.392   0.602    0.555
bsa           11.663     12.125   0.962    0.350
weight        -4.793      6.232  -0.769    0.452

lm(formula = Y ~ bsa)
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   2.7971     8.5284   0.328    0.747
bsa           2.3389     0.1792  13.052 1.29e-10 ***

lm(formula = Y ~ weight)
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  2.20531    8.66333   0.255    0.802
weight       1.20093    0.09297  12.917 1.53e-10 ***
```

Ridge regression and multicollinearity

Ridge prevents this 'cancellation' by imposing the circle constraint



In fact, we would estimate

```
weight 0.574093
bsa    1.146265
```

(This is at a particular value of λ. We will discuss this choice later)

Penalized Methods and Rescaling

Least squares is invariant to rescaling

Example: Let's multiply our design matrix by a factor of 10 to get $\tilde{\mathbb{X}} = 10\mathbb{X}$. Then:

$$\tilde{\beta}_{\text{LS}} = (\tilde{\mathbb{X}}^\top \tilde{\mathbb{X}})^{-1} \tilde{\mathbb{X}}^\top Y = \frac{1}{10}(\mathbb{X}^\top \mathbb{X})^{-1} \mathbb{X}^\top Y = \frac{\hat{\beta}_{\text{LS}}}{10}$$

So, multiplying our data by ten just results in our estimates being reduced by one tenth.

Hence, any prediction is left unchanged:

$$\tilde{\mathbb{X}} \tilde{\beta}_{\text{LS}} = \mathbb{X} \hat{\beta}_{\text{LS}}$$

This means, for instance, if we have a feature measured in miles, then we will get the "same" answer if we change it to kilometers

Penalized methods

Penalized methods are not invariant to rescaling

Though there are many choices, usually we standardize the feature matrix to have

- Zero (sample) mean and
- (sample) standard deviation 1.

This look like:

$$x_j \leftarrow \frac{(x_j - \text{mean}(x_j))}{\text{sd}(x_j)}$$

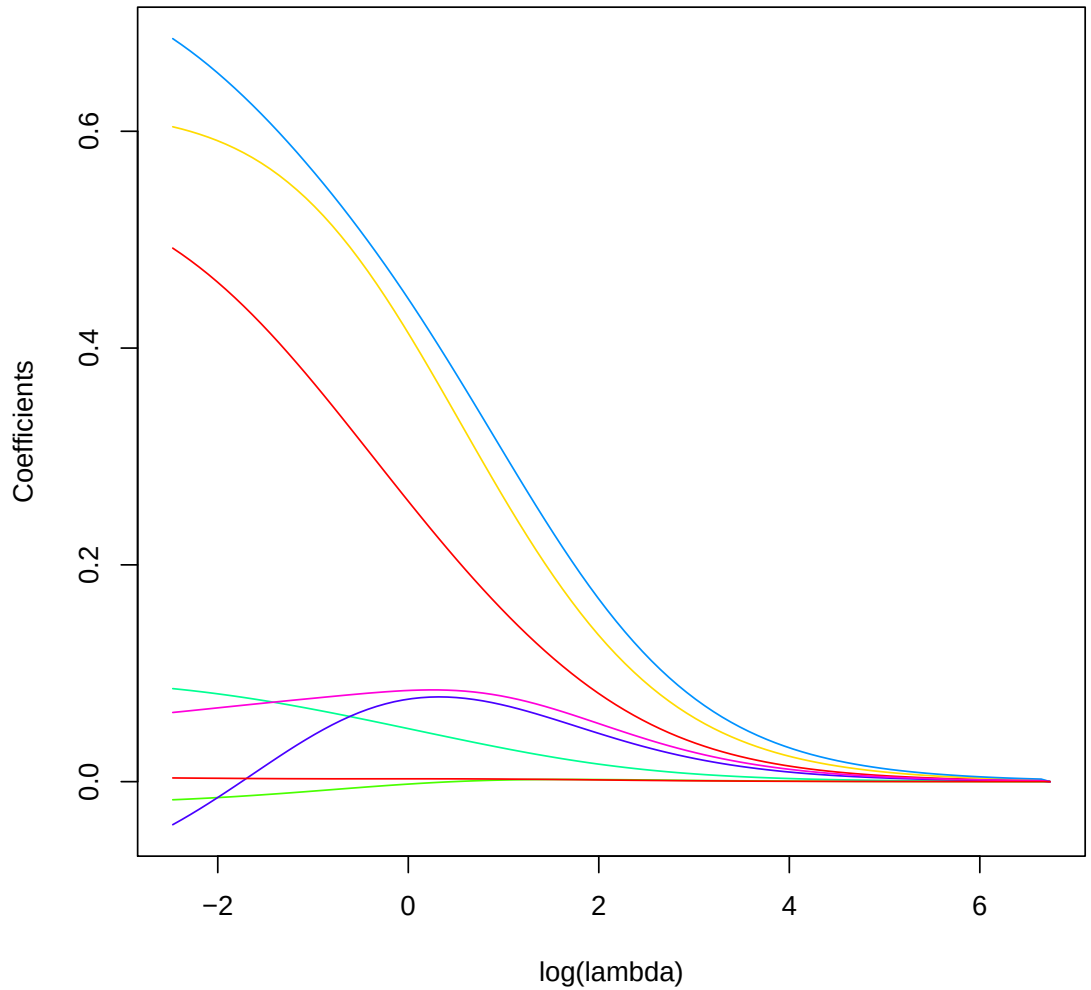
```
X %>% preProcess(method = c('center', 'scale')) %>%
  predict(newdata = X)
```

Note that you do not typically want to standardize dummy variables

(They are already 'scale-free', afterall)

Ridge regression coefficient path

We can look at all these solutions via the coefficient path



Lasso

Lasso

Lasso is technically an acronym from 'least angle selection and shrinkage operator'

It minimizes a related criterion to ridge:

$$SSE_{L_1} = \sum_{i=1}^{n}(Y_i - (\beta_0 + \sum_{j=1}^{p}\beta_j x_j))^2 + \lambda \sum_{j=1}^{p}|\beta_j|$$

The lasso superficially looks a lot like ridge regression

However, the subtle change from $\sum_{j=1}^{p}\beta_j^2$ to $\sum_{j=1}^{p}|\beta_j|$ makes a big difference

Lasso regression

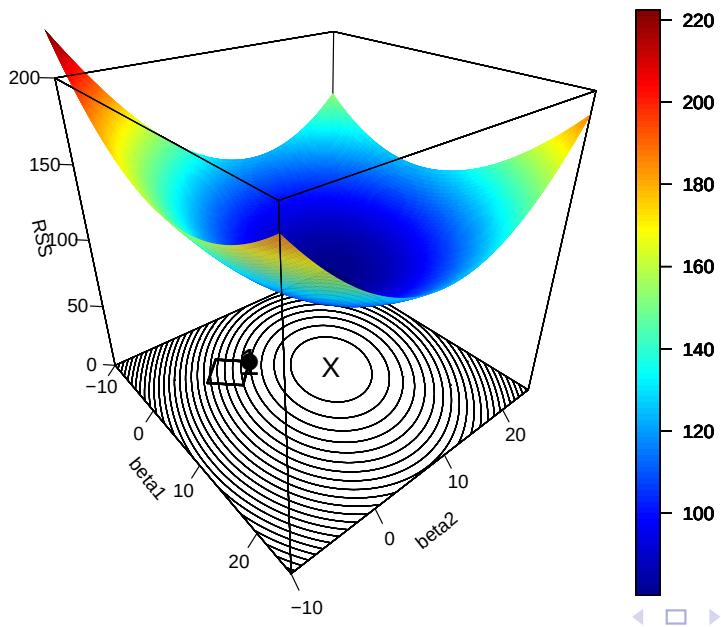
Reminder: The equation for a diamond is

$$|x| + |y| = r$$

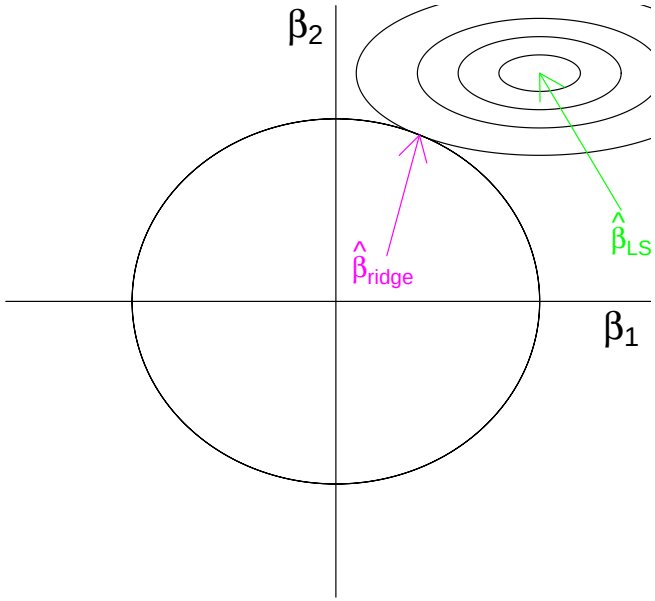
This creates a diamond centered at (0,0) with radius r

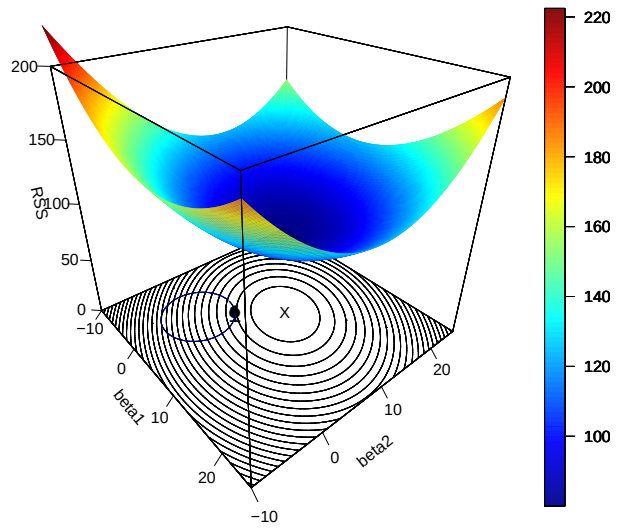
The penalty term $\sum_{j=1}^{p} |\beta_j|$ is a diamond!

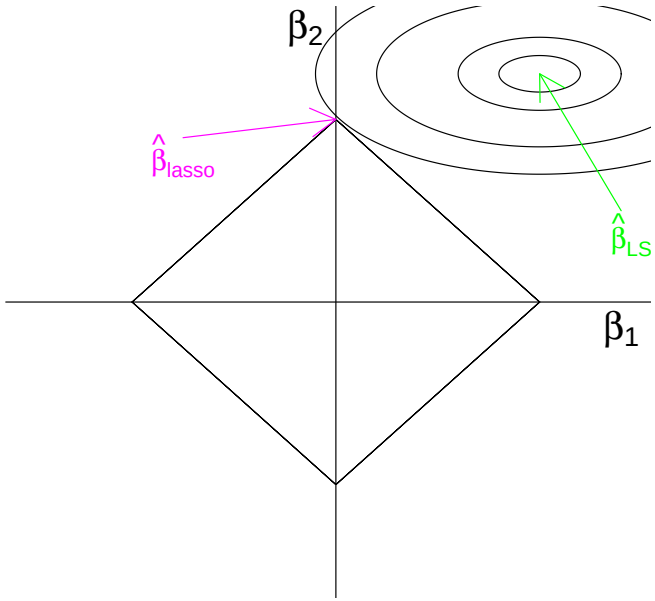
This translates geometrically to a constraint set:

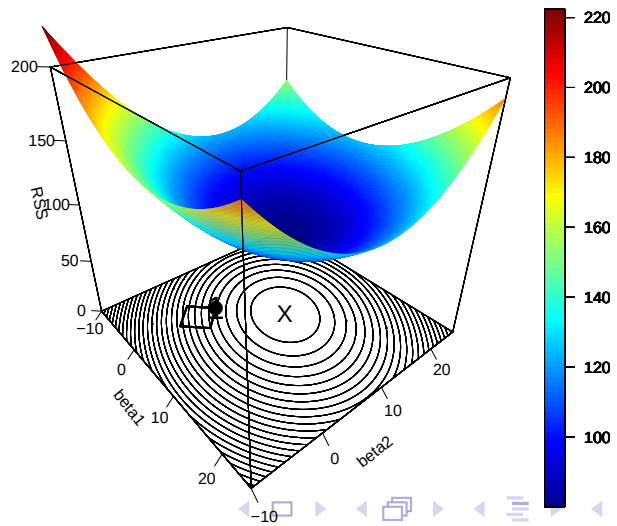


Ridge compared with lasso



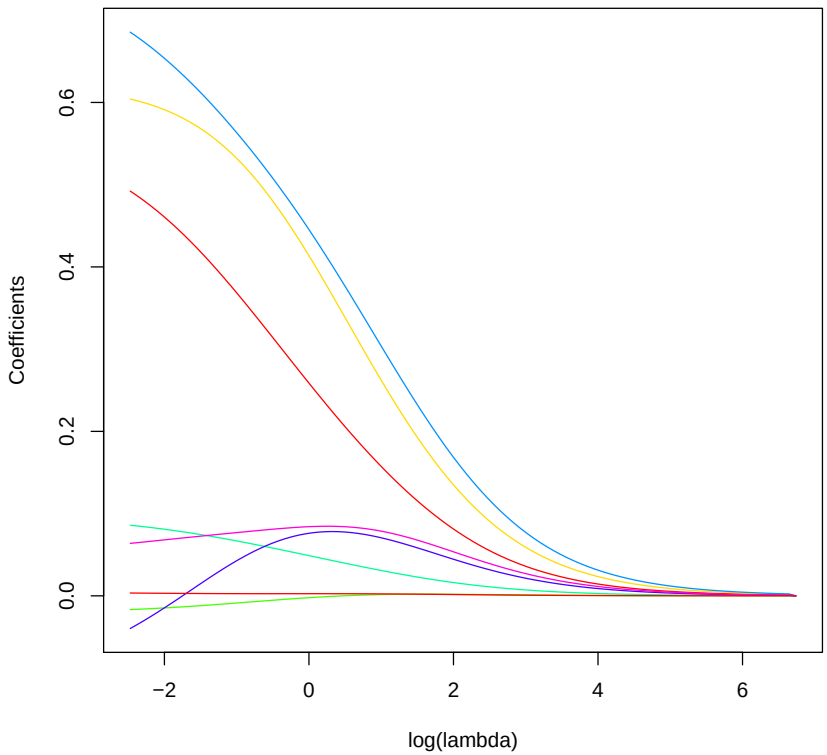




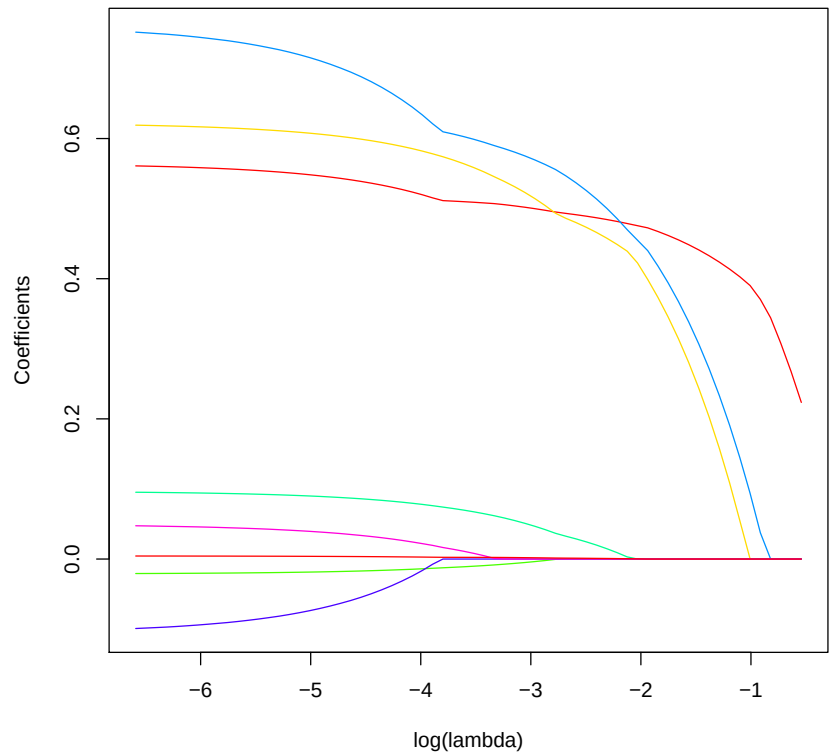


Coefficient paths for ridge and lasso

The 'corners' of the diamond forces some of the coefficient estimates to be exactly equal to zero



Ridge



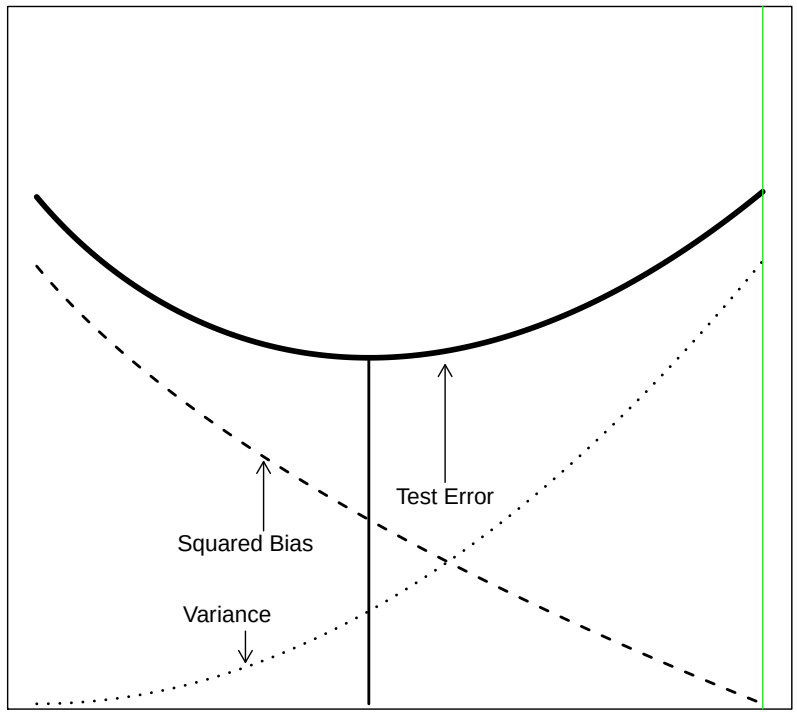
Lasso

Choosing λ

Choosing λ

With penalized methods, there isn't 'a' solution, rather an infinite number of solutions

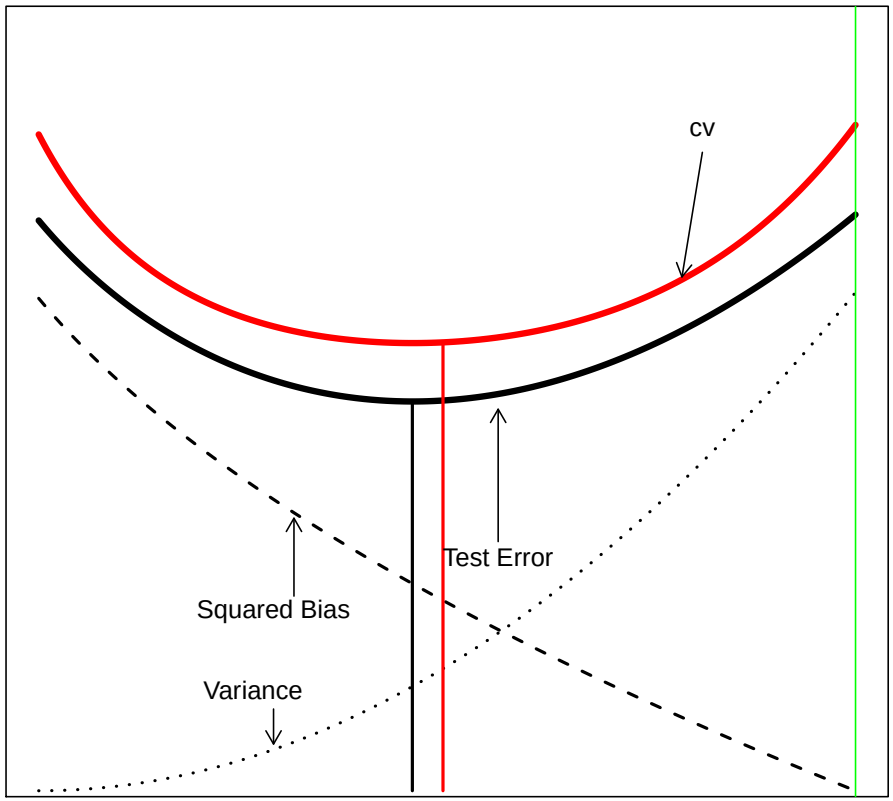
We need to choose a good value of λ



Ridge Regression: The tuning parameter

We can choose λ via the K-fold cross-validation estimator of the test error

Think of CV as a function of λ, and pick its minimizer: $\hat{\lambda}$



Choosing λ

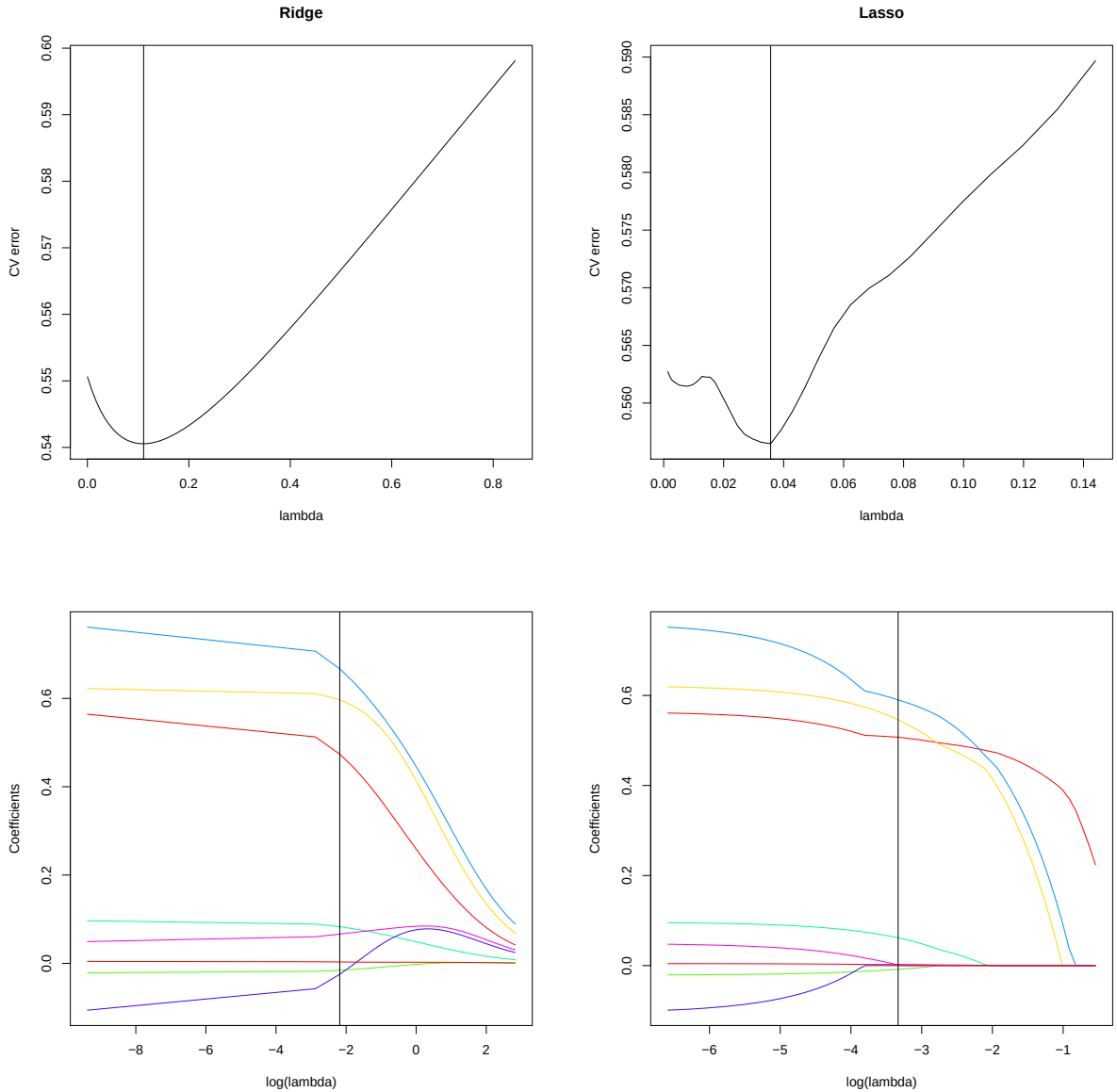


Figure: Vertical line at minimum CV tuning parameter

Elastic net

Elastic net

It turns out that both ridge and lasso are special cases of the elastic net

$$SSE_{Enet} = \sum_{i=1}^{n}(Y_i - (\beta_0 + \sum_{j=1}^{p}\beta_j x_j))^2 + \lambda(\alpha\sum_{j=1}^{p}|\beta_j| + (1-\alpha)\sum_{j=1}^{p}\beta_j^2)$$

(This is equivalent to the definition in the book, but I used more conventional notation)

- $\alpha = 1$ is lasso
- $\alpha = 0$ is ridge

Adding a second tuning parameter allows for more model flexibility, but now we have to set two tuning parameters

When would you choose each method?

- Ridge regression
  - The 'high dimensional' regime
  - If you're interested mainly in predictions
  - There are a large number of highly correlated features
  - If you are severely computationally constrained
- Lasso
  - If you're interested mainly in identifying which features are related to the supervisor
- Elastic net
  - If you're interested mainly in identifying which features are related to the supervisor
  - There are a large number of highly correlated features

Example

Example: With two highly correlated features measured on the same scale, like high school GPA and University GPA. Then:

- Ridge would estimate about the same coefficient on the two features and it would be a bit less than half that estimated when using each feature on its own
- The lasso will choose one of the features and remove the other
- The elastic net will either set both of the coefficients to zero or estimate both of the coefficients

Postamble:

- Penalized methods allow us to exchange some bias for lower variance

  (The least squares estimator has the lowest bias, but can have very large variance; especially when the features are correlated)

- We discuss some penalized regression procedures, in particular ridge regression, lasso, and the elastic net

  (The most general method is the elastic net. We can thing of lasso, ridge, and least squares as all being special cases)